MapReduce 服务

故障排除

文档版本 01

发布日期 2024-12-18

华为云计算技术有限公司

版权所有 © 华为云计算技术有限公司 2024。 保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明

HUAWEI和其他华为商标均为华为技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目 录

1 账号密码类 1
1.1 如何重置或修改 Manager（admin）密码? 1
1.2 用户名过长导致下载认证凭据失败 1
2 账号权限类 3
2.1 获取 MRS 集群主机列表接口时提示用户无权限 3
2.2 查看 MRS 集群详情失败 4
3 Web 页面访问类 5
3.1 无法访问 MRS 集群 Manager 页面如何处理? 5
3.2 登录集群 Manager 常见异常问题 7
3.2.1 访问 MRS Manager 页面报错“502 Bad Gateway” 7
3.2.2 访问 Manager 页面报错“请求 VPC 错误” 8
3.2.3 通过专线访问 Manager 页面方式出现 503 报错 9
3.2.4 登录 Manager 集群页面时报错“You have no right to access the page” 9
3.2.5 登录 Manager 时界面报错“认证信息无效” 10
3.2.6 Manager 界面超时退出后无法正常登录 11
3.2.7 升级 Python 后无法登录 Manager 页面 11
3.2.8 修改集群域名后无法登录 Manager 页面 12
3.2.9 登录 Manager 界面后页面空白无内容 13
3.2.10 集群节点内安装了原生 Kerberos 导致登录集群失败 14
3.2.11 macOS 使用浏览器访问 MRS Manager 15
3.2.12 Manager 界面登录用户被锁如何解锁 17
3.2.13 Manager 页面加载卡顿 18
3.3 访问 MRS 组件 Web UI 界面常见异常问题 19
3.3.1 使用 admin 用户访问组件的 WebUI 界面报错或部分功能不可用 19
3.3.2 访问组件 WebUI 页面报 500 异常 20
3.3.3 【HBase WebUI】无法从 HBase WebUI 界面跳转到 RegionServer WebUI 21
3.3.4 【HDFS WebUI】访问 HDFS WebUI 页面报错“重定向次数过多” 21
3.3.5 【HDFS WebUI】使用 IE 浏览器访问 HDFS WebUI 界面失败 22
3.3.6 【Hue WebUI】登录 Hue WebUI 报错无权限 22
3.3.7 【Hue WebUI】Hue WebUI 页面无法访问 23
3.3.8 【Hue WebUI】访问 Hue WebUI 页面报错“Proxy Error” 24
3.3.9 【Hue WebUI】集群未安装 Hive 服务时 Hue 原生页面无法正常显示 25

3.3.10 【Hue WedUI】Hue（主）无法打开 web 网页.....26
3.3.11 【Ranger WebUI】新用户修改完密码后无法登录 Ranger WebUI 界面.....26
3.3.12 【Tez WebUI】访问 Tez WebUI 界面报错 404.....26
3.3.13 【Spark WebUI】为什么从 Yarn Web UI 页面无法跳转到 Spark WebUI 界面.....27
3.3.14 【Spark WebUI】HistoryServer 缓存的应用被回收导致此类应用页面访问时出错.....28
3.3.15 【Spark WebUI】访问 JobHistory 中某个应用的原生页面时页面显示错误.....29
3.3.16 【Spark WebUI】使用 IE 浏览器访问 Spark2x WebUI 界面失败.....29
3.3.17 【Yarn WebUI】无法访问 Yarn WebUI.....30

4 API 使用类.....31

4.1 调用 API 接口创建集群失败.....31

5 集群管理类.....32

5.1 缩容 Task 节点失败.....32
5.2 如何处理集群内部 OBS 证书过期.....33
5.3 MRS 集群更换磁盘（适用于 2.x 及之前）.....34
5.4 MRS 集群更换磁盘（适用于 3.x）.....37
5.5 MRS 备份任务执行失败.....40
5.6 Core 节点出现 df 显示的容量和 du 显示的容量不一致.....41
5.7 如何解除网络 ACL 的关联子网.....41
5.8 修改节点主机名后导致 MRS 集群状态异常.....42
5.9 进程被终止如何定位原因.....42
5.10 MRS 配置跨集群互信失败.....44
5.11 MRS 集群节点使用 pip3 安装 Python 时提示网络不可达.....46
5.12 开源 confluent-kafka-go 连接 MRS 安全集群.....46
5.13 MRS 集群周期备份任务执行失败.....48
5.14 MRS 集群客户端无法下载.....49
5.15 开启 Kerberos 认证的 MRS 集群提交 Flink 作业报错.....49
5.16 通过 Hive beeline 命令行执行插入命令的时候报错.....51
5.17 MRS 集群节点如何进行操作系统漏洞升级.....52
5.18 使用 CDM 服务迁移数据至 MRS HDFS 时任务运行失败.....53
5.19 MRS 集群频繁产生节点间心跳中断告警.....54
5.20 PMS 进程占用内存高.....56
5.21 Knox 进程占用内存高.....57
5.22 安全集群外的客户端访问 HBase 很慢.....58
5.23 作业无法提交如何定位？.....58
5.24 HBase 日志文件过大导致 OS 盘空间不足.....61
5.25 HDFS 日志文件过大导致 OS 盘空间不足.....62
5.26 MRS 集群节点规格升级异常.....62
5.27 Manager 页面新建的租户删除失败.....64
5.28 MRS 集群切换 VPC 后集群状态异常不可用.....65
5.29 MRS 管理控制台上提交作业异常处理.....67
5.30 生成 HA 证书时报错“symbol xxx not defined in file libcrypto.so.1.1”.....67
5.31 MRS 集群 Core 节点扩容成功后部分实例启动失败.....68

6 使用 Alluixo 71
6.1 Alluixo 在 HA 模式下出现 Does not contain a valid host:port authority 报错 71
7 使用 ClickHouse 72
7.1 ZooKeeper 上数据错乱导致 ClickHouse 启动失败问题 72
7.2 ClickHouse 消费 Kafka 数据异常 74
8 使用 DBservice 76
8.1 DBServer 实例状态异常 76
8.2 DBServer 实例一直处于 Restoring 状态 77
8.3 DBService 默认端口 20050 或 20051 被占用 78
8.4 /tmp 目录权限不对导致 DBserver 实例状态一直处于 Restoring 78
8.5 DBService 备份任务执行失败 80
8.6 DBService 状态正常，组件无法连接 DBService 80
8.7 DBServer 启动失败 81
8.8 浮动 IP 不通导致 DBService 备份失败 82
8.9 DBService 配置文件丢失导致启动失败 84
9 使用 Flink 87
9.1 Flink 客户端执行命令报错“Error while parsing YAML configuration file : security.kerberos.login.keytab” 87
9.2 Flink 客户端执行命令报错“Error while parsing YAML configuration file : security.kerberos.login.principal:pippo” 88
9.3 Flink 客户端执行命令报错“Could not connect to the leading JobManager” 89
9.4 使用不同用户执行 yarn-session 创建 Flink 集群失败 90
9.5 Flink 业务程序无法读取 NFS 盘上的文件 91
9.6 自定义 Flink log4j 日志输出级别不生效 92
10 使用 Flume 93
10.1 Flume 向 Spark Streaming 提交作业后报类找不到错误 93
10.2 Flume 客户端安装失败 94
10.3 Flume 客户端无法连接服务端 94
10.4 Flume 数据写入组件失败 95
10.5 Flume 服务端进程故障 96
10.6 Flume 数据采集慢 96
10.7 Flume 启动失败 97
11 使用 HBase 99
11.1 连接到 HBase 响应慢 99
11.2 HBase 用户认证失败 99
11.3 端口被占用导致 RegionServer 启动失败 100
11.4 节点剩余内存不足导致 HBase 启动失败 101
11.5 HDFS 性能差导致 HBase 服务不可用告警 101
11.6 参数不合理导致 HBase 启动失败 102
11.7 残留进程导致 Regionsever 启动失败 103
11.8 HDFS 上设置配额导致 HBase 启动失败 103

11.9 HBase version 文件损坏导致启动失败 104
11.10 无业务情况下，RegionServer 占用 CPU 高 105
11.11 HBase 启动失败，RegionServer 日志中提示“FileNotFoundException” 106
11.12 HBase 启动后原生页面显示 RegionServer 个数多于实际个数 108
11.13 RegionServer 实例异常，处于 Restoring 状态 109
11.14 新安装的集群 HBase 启动失败 109
11.15 acl 表目录丢失导致 HBase 启动失败 110
11.16 集群上下电之后 HBase 启动失败 111
11.17 文件块过大导致 HBase 数据导入失败 112
11.18 使用 Phoenix 创建 HBase 表后，向索引表中加载数据报错 113
11.19 在 MRS 集群客户端无法执行 **hbase shell** 命令 114
11.20 HBase shell 客户端在使用中有 INFO 信息打印在控制台导致显示混乱 115
11.21 RegionServer 剩余内存不足导致 HBase 服务启动失败 116
11.22 集群扩容之后新节点 HRegionServer 启动失败 117
11.23 HBase 文件丢失导致 Region 状态长时间为 RIT 117

12 使用 HDFS 119

12.1 修改 RPC 端口后 HDFS NameNode 实例都变为备状态 119
12.2 通过公网 IP 连接使用 HDFS 客户端报错 120
12.3 使用 Python 远程连接 HDFS 的端口失败 120
12.4 HDFS 容量达到 100%导致上层服务 HBase、Spark 等不可用 121
12.5 启动 HDFS 和 Yarn 服务报错“Permission denied” 122
12.6 HDFS 用户可在其他用户目录下创建或删除文件 123
12.7 HDFS 的 DataNode 一直显示退服中 124
12.8 内存不足导致 HDFS NameNode 启动失败 126
12.9 ntpdate 修改时间导致 HDFS 出现大量丢块 127
12.10 DataNode 概率性出现 CPU 占用接近 100%导致节点丢失 129
12.11 单 NameNode 长期故障时如何使用客户端手动 checkpoint 130
12.12 HDFS 文件读写时报错“Failed to place enough replicas” 132
12.13 文件最大打开句柄数设置太小导致读写文件异常 132
12.14 HDFS 客户端写文件 close 失败 133
12.15 文件错误导致上传文件到 HDFS 失败 135
12.16 界面配置 dfs.blocksize 后上传数据，block 大小未改变 136
12.17 读取 HDFS 文件失败，报错“FileNotFoundException” 137
12.18 HDFS 写文件失败，报错“item limit of xxx is exceeded” 137
12.19 调整 HDFS SHDFShell 客户端日志级别 138
12.20 HDFS 读文件失败报错“No common protection layer” 138
12.21 HDFS 目录配额不足导致写文件失败 139
12.22 执行 balance 失败报错“Source and target differ in block-size” 140
12.23 查询或者删除 HDFS 文件失败 141
12.24 非 HDFS 数据残留导致数据分布不均衡 142
12.25 HDFS 客户端安装在数据节点导致数据分布不均衡 143
12.26 节点内 DataNode 磁盘使用率不均衡 143

12.27 执行 balance 常见问题定位方法 144
12.28 HDFS 显示磁盘空间不足，其实还有 10%磁盘空间 145
12.29 使用普通集群 Core 节点上安装的 HDFS 客户端时报错“error creating DomainSocket” 146
12.30 集群外节点安装客户端上传 HDFS 文件失败 146
12.31 HDFS 写并发较大时报副本不足 147
12.32 HDFS 客户端无法删除超长目录 148
12.33 集群外节点访问 MRS HDFS 报错 149
12.34 NameNode 节点存在 ALM-12027 主机 PID 使用率超过阈值告警 150
12.35 集群出现 ALM-14012 Journalnode 数据不同步告警 152
12.36 由于 HDFS 块丢失导致 DataNode 退服失败 152
12.37 使用 distcp 命令拷贝空文件夹报错 154

13 使用 Hive 155

13.1 Hive 常见日志说明 155
13.2 Hive 服务启动失败 156
13.3 安全集群执行 set 命令的时候报错“Cannot modify xxx at runtime” 157
13.4 提交 Hive 任务时如何指定队列？ 158
13.5 怎么通过客户端设置 Map/Reduce 内存？ 158
13.6 如何在导入 Hive 表时指定输出的文件压缩格式？ 159
13.7 Hive 表 desc 描述过长导致无法完整显示 159
13.8 Hive 表中增加分区列后再插入数据显示为 NULL 160
13.9 集群中创建的新用户执行查询 Hive 操作时无权限 161
13.10 执行 SQL 提交任务到指定队列时报错 163
13.11 执行 load data inpath 命令报错 163
13.12 执行 load data local inpath 命令报错 164
13.13 执行 create external table 命令报错 165
13.14 在 beeline 客户端执行 dfs -put 命令报错 165
13.15 执行 set role admin 命令报无权限 166
13.16 在 beeline 客户端创建 UDF 时报错 167
13.17 Hive 服务状态为故障如何处理 167
13.18 Hive 服务健康状态和 Hive 实例健康状态的区别 167
13.19 Shell 客户端连接提示“authentication failed” 168
13.20 客户端提示访问 ZooKeeper 失败 169
13.21 使用 UDF 函数时提示“Invalid function” 170
13.22 Hive 服务状态为 Unknown 170
13.23 HiveServer 或者 MetaStore 实例的健康状态为 unknown 171
13.24 HiveServer 或者 MetaStore 实例的健康状态为 Concerning 171
13.25 TEXTFILE 类型文件使用 ARC4 压缩时查询结果乱码 172
13.26 Hive 任务运行过程中失败，重试成功 172
13.27 执行 select 语句时报错“Execution Error return code 2” 173
13.28 有大量分区时执行 drop partition 操作失败 174
13.29 执行 join 操作时 localtask 启动失败 175
13.30 修改 hostname 后导致 WebHCat 启动失败 176

13.31 集群修改域名后 Hive 样例程序运行报错 177
13.32 DBService 超过最大连接数后导致 Hive MetaStore 异常 178
13.33 beeline 客户端报错“Failed to execute session hooks: over max connections” 179
13.34 beeline 客户端报错“OutOfMemoryError” 180
13.35 输入文件数超出设置限制导致任务执行失败 181
13.36 Hive 任务执行中报栈内存溢出导致任务执行失败 183
13.37 对同一张表或分区并发写数据导致任务失败 184
13.38 Hive 任务失败，报没有 HDFS 目录的权限 184
13.39 Load 数据到 Hive 表失败 185
13.40 参考 Hive JDBC 代码用例开发的业务应用运行失败 186
13.41 HiveServer 和 HiveHCat 进程故障 187
13.42 MRS Hive 连接 ZooKeeper 时报错“ConnectionLoss for hiveserver2” 188
13.43 Hive 执行 insert into 语句报错 189
13.44 增加 Hive 表字段超时 190
13.45 重启 Hive 服务失败 192
13.46 Hive 分区数过多导致删除表失败 193
13.47 Hive 执行 msck repair table 命令时报错 194
13.48 Hive 执行 insert into 命令报用户权限不足 194
13.49 在 Hive 中 drop 表后如何完全释放磁盘空间 195
13.50 JSON 表中有破损数据导致 Hive 查询异常 195
13.51 Hive 客户端执行 SQL 报错连接超时 197
13.52 WebHCat 健康状态异常导致启动失败 198
13.53 mapred-default.xml 文件解析异常导致 WebHCat 启动失败 199
13.54 MetaStore 动态分区数超出阈值导致 SQL 报错 200

14 使用 Hue 201

14.1 Hue 界面中查看到未知 Job 在运行 201
14.2 使用 IE 浏览器在 Hue 中执行 HQL 失败 201
14.3 Hue WebUI 界面访问失败 202
14.4 Hue 界面无法加载 HBase 表 203
14.5 Hue 中的输入框输入中文会出现混乱 204
14.6 Hue 上执行 Impala SQL 查询中文报错 205

15 使用 Impala 206

15.1 连接 impala-shell 失败 206
15.2 创建 Kudu 表报错 207
15.3 Impala 客户端安装 Python2 207

16 使用 Kafka 211

16.1 运行 Kafka 客户端获取 Topic 时报错 211
16.2 安全集群中使用 Python3.x 对接 Kafka 211
16.3 Flume 正常连接 Kafka 后发送消息失败 213
16.4 Producer 发送数据失败，报错“NullPointerException” 215
16.5 Producer 发送数据失败，报错“TOPIC_AUTHORIZATION_FAILED” 217

16.6 Producer 偶现发送数据失败，日志提示“Too many open files in system” 219
16.7 Consumer 初始化成功但是无法从 Kafka 中获取指定 Topic 消息 221
16.8 Consumer 消费数据失败，Consumer 一直处于等待状态 225
16.9 SparkStreaming 消费 Kafka 消息失败，提示“Error getting partition metadata” 227
16.10 新建集群 Consumer 消费数据失败，提示“GROUP_COORDINATOR_NOT_AVAILABLE” 229
16.11 SparkStreaming 消费 Kafka 消息失败，提示“Couldn't find leader offsets” 230
16.12 Consumer 消费数据失败，提示“SchemaException: Error reading field” 232
16.13 Kafka Consumer 消费数据丢失 233
16.14 账号锁定导致启动 Kafka 组件失败 233
16.15 Kafka Broker 上报进程异常，日志提示“IllegalArgumentException” 234
16.16 执行 Kafka Topic 删除操作时发现无法删除 235
16.17 执行 Kafka Topic 删除操作时提示“AdminOperationException” 239
16.18 执行 Kafka Topic 创建操作时发现无法创建提示“NoAuthException” 241
16.19 执行 Kafka Topic 设置 ACL 操作失败，提示“NoAuthException” 243
16.20 执行 Kafka Topic 创建操作时发现无法创建提示“NoNode for /brokers/ids” 245
16.21 执行 Kakfa Topic 创建操作时发现无法创建提示“replication factor larger than available brokers” 246
16.22 Consumer 消费数据时存在重复消费现象 247
16.23 执行 Kafka Topic 创建操作时发现 Partition 的 Leader 显示为 none 249
16.24 Kafka 安全使用说明 250
16.25 如何获取 Kafka Consumer Offset 信息 254
16.26 如何针对 Topic 进行配置增加和删除 256
16.27 如何读取“__consumer_offsets”内部 topic 的内容 257
16.28 如何配置 Kafka 客户端 shell 命令日志 258
16.29 如何获取 Topic 的分布信息 259
16.30 Kafka 高可靠使用说明 260
16.31 使用 Kafka Shell 命令无法操作 Kafka 集群 263
16.32 Kafka 生产者写入单条记录过长问题 264
16.33 Kakfa 消费者读取单条记录过长问题 265
16.34 Kafka 集群节点内多磁盘数据量占用高 265
16.35 Kafka 连接 ZooKeeper 客户端时发生断链 268

17 使用 Oozie 269

17.1 并发提交大量 oozie 任务时，任务一直没有运行 269
17.2 Oozie 调度 HiveSQL 作业报错处理 269
17.3 在 MRS 集群外客户端提交不了 Oozie 任务或两个小时才提交成功 270

18 使用 Presto 272

18.1 配置 sql-standard-with-group 创建 schema 失败报 Access Denied 272
18.2 Presto 的 coordinator 进程无法正常启动 273
18.3 Presto 查询 Kudu 表时系统报错找不到表 275
18.4 Presto 查询 Hive 表无数据 276
18.5 MRS Presto 查询报错“The node may have crashed or be under too much load” 277
18.6 MRS 集群如何使用公网访问 Presto 277

19 使用 Spark 280
19.1 运行 Spark 应用时修改 split 值报错 280
19.2 提交 Spark 任务时提示参数格式错误 281
19.3 磁盘容量不足导致 Spark、Hive 和 Yarn 服务不可用 282
19.4 引入 jar 包不正确导致 Spark 任务无法运行 282
19.5 Spark 任务由于内存不够或提交作业时未添加 Jar 包，作业卡住 283
19.6 提交 Spark 任务时报错“ClassNotFoundException” 284
19.7 提交 Spark 任务时 Driver 端提示运行内存超限 285
19.8 Yarn-cluster 模式下提交 Spark 任务报“Can't get the Kerberos realm”异常 286
19.9 JDK 版本不匹配导致启动 spark-sql 和 spark-shell 失败 287
19.10 Yarn-client 模式提交 Spark 任务时 ApplicationMaster 尝试启动两次失败 288
19.11 提交 Spark 任务时连接 ResourceManager 异常 289
19.12 DataArts Studio 调度 Spark 作业失败 289
19.13 通过 API 提交 Spark 作业后作业状态为 error 290
19.14 MRS 集群反复上报 43006 告警 291
19.15 在 spark-beeline 中创建或删除表失败 291
19.16 集群外节点提交 Spark 作业时报错无法连接 Driver 293
19.17 运行 Spark 任务发现大量 shuffle 结果丢失 294
19.18 JDBCServer 长时间运行导致磁盘空间不足 294
19.19 spark-shell 执行 SQL 跨文件系统 load 数据到 Hive 表失败 296
19.20 Spark 任务提交失败 296
19.21 Spark 任务运行失败 297
19.22 JDBCServer 连接失败 298
19.23 查看 Spark 任务日志失败 298
19.24 Spark Streaming 任务提交问题 299
19.25 Spark 连接其他服务认证问题 299
19.26 Spark 连接 Kafka 认证错误 300
19.27 SparkSQL 读取 ORC 表报错 300
19.28 Spark WebUI 页面上 stderr 和 stdout 无法跳转到日志页面 301
19.29 spark-beeline 查询 Hive 视图报错 304
19.30 Spark Streaming 任务失败但是 Yarn WebUI 中显示状态为成功 305

20 使用 Sqoop 306
20.1 Sqoop 如何连接 MySQL 306
20.2 Sqoop 读取 MySQL 数据到 HBase 时报 HBaseAdmin.<init>方法找不到异常 307
20.3 通过 Hue 创建 Sqoop 任务将数据从 HBase 导入 HDFS 时报错 308
20.4 通过 Sqoop 从 Hive 导出数据到 MySQL 8.0 时报数据格式错误 312
20.5 通过 sqoop import 命令从 PgSQL 导出数据到 Hive 时报错 313
20.6 通过 Sqoop 读取 MySQL 数据并写 parquet 文件到 OBS 时失败 314
20.7 通过 Sqoop 迁移数据库数据时报错 315

21 使用 Storm 317
21.1 Storm WebUI 页面中 events 超链接地址无效 317
21.2 提交 Storm 拓扑失败排查思路 318

21.3 提交 Storm 拓扑失败，提示 Failed to check principle for keytab.....320
21.4 提交 Storm 拓扑后 Worker 日志为空.....321
21.5 提交 Storm 拓扑后 Worker 运行异常，日志提示 Failed to bind to XXX.....322
21.6 使用 jstack 命令查看进程堆栈提示“well-known file is not secure”.....324
21.7 使用 Storm-JDBC 插件开发 Oracle 写入 Bolt 时发现数据无法写入.....326
21.8 Storm UI 查看信息时显示 Internal Server Error.....328

22 使用 Ranger.....329

22.1 Hive 启用 Ranger 鉴权后，在 Hue 页面能查看到没有权限的表和库.....329

23 使用 Yarn.....331

23.1 集群启动 Yarn 后产生大量作业占用资源.....331
23.2 通过客户端 hadoop jar 命令提交任务后返回“GC overhead”报错.....332
23.3 Yarn 汇聚日志过大导致节点磁盘被占满.....333
23.4 MapReduce 任务异常，临时文件未删除.....335
23.5 Yarn 客户端的端口信息错误导致提交任务后报错 connection refused.....336
23.6 通过 Yarn WebUI 查看作业日志时提示“Could not access logs page!”.....337
23.7 通过 Yarn WebUI 查看队列信息时提示“ERROR 500”错误.....338
23.8 通过 Yarn WebUI 页面查看作业日志提示“ERROR 500”错误.....338
23.9 使用 Yarn 客户端命令查询历史作业报错.....339
23.10 TimelineServer 目录文件数量到达上限.....340

24 使用 ZooKeeper.....342

24.1 通过 MRS 客户端连接 ZooKeeper 报错.....342
24.2 由于主备 Master 节点时间未同步导致 ZooKeeper 服务不可用.....343

25 存算分离类.....345

25.1 没有/tmp 目录权限导致执行访问 OBS 的作业报错.....345
25.2 使用 Hadoop 客户端删除 OBS 上数据时报.Trash 目录没有权限错误.....346
25.3 由于 MRS 集群节点的 NTP 时间不同步导致集群访问 OBS 时鉴权失败.....347

1 账号密码类

1.1 如何重置或修改 Manager（admin）密码?

- 已知旧密码修改admin密码请参考**修改admin密码**
- 忘记旧密码重置admin密码请参考**修改组件运行用户密码**

1.2 用户名过长导致下载认证凭据失败

用户问题

MRS 3.0.2~MRS 3.1.0版本集群，当用户名超过20位时（添加用户时最长限制为32位），下载Keytab文件会下载失败，状态代码：400 Bad Request。

问题现象

MRS 3.0.2~MRS 3.1.0版本集群，当用户名超过20位时（添加用户时最长限制为32位），下载Keytab文件会下载失败，状态代码：400 Bad Request。

原因分析

需要在主Master节点的“/opt/Bigdata/om-server_*/apache-tomcat-*/webapps/web/WEB-INF/validate”路径下，修改validate-common-config.xml、validate-rule-session.xml、 validate-rule-user.xml三个配置文件。

处理步骤

步骤1 以**omm**用户登录主Master节点的“/opt/Bigdata/om-server_*/apache-tomcat-*/webapps/web/WEB-INF/validate”路径。

cd /opt/Bigdata/om-server_*/apache-tomcat-*/webapps/web/WEB-INF/validate

步骤2 修改validate-common-config.xml文件。

vi validate-common-config.xml

将用户名的“maxLength”参数的值从“32”修改为“64”：

```
<!-- 用户名 -->
<validators alias="USER_NAME">
      <validator name="RANGE_LENGTH_VALIDATOR" minLength="3"
         maxLength="64" />
      <validator name="REGEXP_VALIDATOR" rule="^[_a-zA-Z0-9\- ]+$"
</validators>
```

步骤3 修改validate-rule-session.xml文件。

vi validate-rule-session.xml

将“下载当前用户凭据”的参数“rule”的值从“20”改为“64”：

```
 <!-- 下载当前用户凭据 -->
 <param_validator url="/api/v2/session/user/keytab/download" method="get"
errorHandler="com.huawei.bigdata.om.web.api.validate.SpecialValidatorErrorHandler" dataPattern="form">
            <!-- 参数名：文件名 -->
            <!--校验规则：userName_13位数字_keytab.tar；区分大小写-->
            <parameter name="file_name" required="true" errorKey="13-4000005"
errorMessage="RESID_OM_API_SESSION_0013">
                  <validator name="REGEXP_VALIDATOR" rule="[\-\w ]{3,64}_\d{13}_keytab\.tar"
caseSensitive="true" />
 </parameter>
```

步骤4 修改validate-rule-user.xml文件。

vi validate-rule-user.xml

将“下载当前用户凭据”的参数“rule”的值从“20”改为“64”：

```
<!--下载用户凭据  -->
   <param_validator url="/api/v2/permission/users/keytab/download" method="get"
errorHandler="com.huawei.bigdata.om.web.api.validate.SpecialValidatorErrorHandler" dataPattern="form">
          <!--必输；userName_13位数字_keytab.tar；区分大小写-->
          <parameter name="file_name" required="true" errorKey="12-4000005"
errorMessage="RESID_OM_API_AUTHORITY_0005">
                <validator name="REGEXP_VALIDATOR" rule="[\-\w ]{3,64}_\d{13}_keytab\.tar"
caseSensitive="true" />
          </parameter>
    </param_validator>
```

步骤5 重启Tomcat，并等待启动成功。

1. 以**omm**用户执行以下命令，查询出Tomcat进程的PID号。

   ps -ef|grep apache-tomcat

2. 使用**kill -9** *PID*命令强制停止查询出来的Tomcat进程，例如：

   kill -9 1203

3. 执行以下命令进行重启。

   sh ${BIGDATA_HOME}/om-server/tomcat/bin/startup.sh

步骤6 重新下载认证凭据。

----结束

2 账号权限类

2.1 获取 MRS 集群主机列表接口时提示用户无权限

用户问题

使用AK/SK获取MRS集群主机列表/v1.1/<project_id>/clusters/<cluster_id>/hosts接口时，提示用户无权限“**User do not have right to access cluster**”。

原因分析

使用AK/SK获取MRS集群主机列表时，请求头中未填充project_id等参数导致云服务解析token的project_id与集群的project_id不符合。

处理步骤

在使用AK/SK认证方式调用接口前，请参见**表2-1**收集对应信息，详细的签名方法和SDK使用方法请参见**AK/SK签名指南**。

在构造API请求前，需要获取以下信息，包括组成请求URL的Endpoint和URI，用于签名和认证的AK/SK，用于区分租户的项目ID。

表 2-1 信息收集项

Endpoint	地区与终端节点，即云服务在不同Region有不同的访问域名。 获取方式请参考**终端节点及区域**。
Project_Id	项目ID，在大多数API接口调用时需要配置在uri中，用以识别不同的项目。
AK/SK	访问密钥对，包含密钥ID与密钥。AK/SK用于对请求内容进行签名。
URI	API接口的调用路径及参数。 请参考各云服务的详细接口章节获取，MRS的接口请参考**API概览**。

X-Domain-Id	账号ID，用途： ● Token认证方式下获取Token。 ● AK/SK认证方式下，全局服务的接口调用，需在请求消息头中配置X-Domain-Id。
X-Project-Id	子项目ID，在多项目场景中使用。如果云服务资源创建在子项目中，AK/SK认证方式下，操作该资源的接口调用需要在请求消息头中携带X-Project-Id。

说明

表2-1中参数的获取方式请参见AK/SK签名指南。

2.2 查看 MRS 集群详情失败

问题现象

在MRS管理控制台，单击集群名称，进入MRS集群详情页面失败。

原因分析

1. 用户MRS集群选的是企业项目A（包含MRS FULLACCESS和ECS FULLACCESS权限）。
2. VPC选的是企业项目B。
3. 安全组选的是企业项目A。
4. IAM这边加入的组没有任何权限设置。
5. 经过分析用户的VPC企业项目B里缺少vpc readonly权限导致了该报错的发生。

处理步骤

给VPC所在的企业项目添加VPC只读权限后问题解决。

3 Web 页面访问类

3.1 无法访问 MRS 集群 Manager 页面如何处理?

问题现象

MRS集群创建完成后，无法通过管理控制台访问集群Manager管理页面。

排查思路

1. 检查集群状态：Manager页面无法正常访问，请首先确认MRS集群状态是否正常，如果集群已关闭或者正在创建中，或集群正在发生主备倒换，将无法访问Manager。
2. 检查用户权限：当前用户是否有访问Manager的权限，“ReadOnlyAccess”表示MRS的只读权限，拥有该权限的用户仅能查看MRS的资源，无法访问Manager页面。
3. 检查是否绑定弹性IP：如果需要从本地访问Manager，需要对集群绑定弹性IP且弹性IP可用才可访问。
4. 检查安全组：MRS集群对应的安全组规则中，需要对当前用户开放9022端口。安全组规则中添加的用户本地IP是否发生变化。
5. 检查浏览器：检查本地浏览器是否正常，例如是否配置了内部网络代理、是否添加了会拦截用户Token的安全设置。
6. 检查Manager：EIP、安全组等相关网络配置正常的情况下，检查Manager是否运行正常。
7. 检查用户密码：登录Manager使用的用户密码错误或失效。

处理步骤

步骤1 登录MRS管理控制台页面，选择“现有集群”，查看MRS集群运行状态是否正常。

- 是，执行**步骤3**。
- 否，如果MRS集群状态异常或正在创建中，或集群正在发生主备倒换，请先排查集群故障、或等待集群创建完成、或集群主备倒换完成后再尝试访问Manager。

步骤2 检查当前用户的权限配置是否正常，是否有访问Manager的权限。

具体请参考**创建MRS操作用户**配置用户。

请勿添加MRS ReadOnlyAccess权限，拥有该权限的用户仅能查看MRS的资源，不能访问Manager。

步骤3 在集群列表中单击需要访问的集群名称，进入集群信息页面。

步骤4 检查弹性公网IP是否正常。

单击“集群管理页面 ”后的“前往 Manager”，在“弹性公网IP”下拉框中选择可用的弹性公网IP。

说明

- 如果没有弹性公网IP，可先单击“管理弹性公网IP”创建弹性公网IP，然后在弹性公网IP下拉框中选择对应的弹性公网IP。
- 如果已创建弹性公网IP，但在绑定时无法找到，可能是由于该弹性公网IP被其他集群绑定，请先在弹性公网IP界面解绑，然后再为当前集群绑定。

步骤5 检查安全组设置是否正常。

弹性IP绑定成功后，如果仍旧无法访问Manager，需要检查当前集群对应的安全组规则中是否将9022端口放开。

1. 在“安全组”中选择待添加的安全组规则所在安全组。

   说明

   安全组名称可在集群的“概览”界面的“安全组”查看。

2. 添加安全组规则，可单击“添加安全组规则”后的“管理安全组规则”进入当前集群绑定的安全组基本信息界面，单击“入方向规则”，查看规则列表中是否有源地址为当前用户访问集群的IP地址、协议端口为“TCP：9022”且策略为“允许”的安全组规则；若没有则需单击“添加规则”放开9022端口。

   例如当前用户访问集群的地址为10.x.x.x，则添加的安全组规则为：

   图 3-1 配置安全组

   说明

   安全组中需要放开的IP地址为用户访问MRS集群的公网IP地址，如果当前网络为内部局域网环境，请务必确认安全组中配置的IP地址是否为公网出口IP。

3. 单击MRS集群“概览”页面的“添加安全组规则”检查配置的IP是否变更，如果变更则修改后重新访问Manager。

步骤6 弹性公网IP及安全组均正常的情况下，重新访问Manager，查看浏览器是否可以打开Manager登录页面。

输入用户名及对应密码，单击“登录”进入Manager页面。

说明

- 如需使用管理员用户admin登录，密码为创建集群时用户自行设置的密码。
- 如果使用管理员创建好的用户名密码登录，首次登录后需要重新设置密码。
- 如果登录报错“认证信息无效”，可能是用户密码输入错误或者密码过期，请检查并输入正确密码，或参考**如何重置或修改Manager（admin）密码?** 章节修改密码。
- 检查浏览器是否配置了内部网络代理、是否添加了会拦截用户Token的安全设置。
- 如果多次刷新均无法显示Manager登录页面，可尝试清除浏览器缓存或者更换一个浏览器查看。

登录集群Manager后仍旧出现的页面空白或异常等现象，可参考**登录集群Manager常见异常问题**进行处理。

步骤7 进入集群Manager界面后，用户也可以访问MRS集群的组件Web UI界面。

在Manager界面，选择“集群 > 服务 > *组件名称*”，在概览界面可登录组件的WebUI界面。

例如，选择“集群 > 服务 > HDFS”，在概览界面单击“NameNode WebUI”后的超链接即可进入HDFS WebUI界面。

访问组件WebUI界面出现的页面空白或异常现象，可参考**访问MRS组件Web UI界面常见异常问题**进行处理。

----结束

3.2 登录集群 Manager 常见异常问题

3.2.1 访问 MRS Manager 页面报错“502 Bad Gateway”

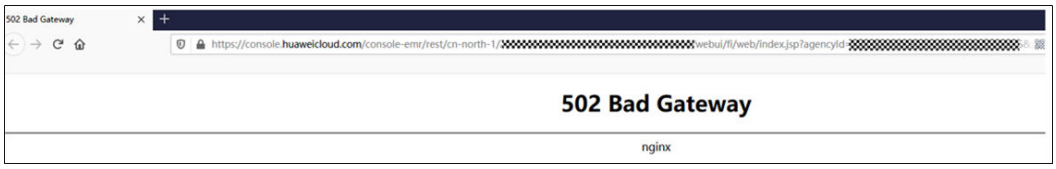
用户问题

访问集群管理页面地址打开显示“502 Bad Gateway”错误。

问题现象

未开启Kerberos认证的集群，访问MRS Manager页面报错如下图：

图 3-2 MRS Manager 页面报错



原因分析

由于MRS的域名由console-emr变更为mrs导致普通集群访问MRS Manager的链接有误。

处理步骤

步骤1 以**root**用户登录到所有Master节点。

步骤2 进入**/opt/knox/conf/**目录，找到ext.properties文件。

```
[root@node-[redacted] ~]# cd /opt/knox/conf/
[root@node-[redacted] conf]# ls
descriptors                gateway-site.xml          krb5JAASLogin.conf     shared-providers        user.keytab
ext.properties             knoxcli-log4j.properties  ldap-log4j.properties  shell-log4j.properties  users.ldif
gateway-log4j.properties   krb5.conf                  README                 topologies
```

步骤3 修改所有Master节点中的ext.properties配置中的“console-emr”为 “mrs”。

```
# Split by comma, exclude those cookies in request
exclude_request_cookies=newCluster,preCluster

# Console context path
console_context=console-emr
```

所有master节点都

改为：mrs

步骤4 进入**/opt/knox/bin/** 目录，并执行**su - omm**命令切换到**omm**用户。

步骤5 执行脚本**restart-knox.sh**重启knox服务。

步骤6 再次访问MRS Manager页面即可正常跳转。

----结束

3.2.2 访问 Manager 页面报错“请求 VPC 错误”

用户问题

通过管理控制台前往Manager，报错“请求VPC错误”。

原因分析

- 当前用户没有VPC的访问权限。
- IAM子用户所在的组有互斥权限导致访问Manager异常。
- 当前用户没有添加安全组中IP的权限，或安全组中添加的IP发生变化。

处理步骤

步骤1 配置用户权限。

具体请参考**创建MRS操作用户**创建用户，或添加用户权限（请勿添加MRS ReadOnlyAccess权限，拥有该权限的用户仅能查看MRS的资源。）。

说明

默认情况下，管理员创建的IAM用户没有任何权限，需要将其加入用户组，并给用户组授予策略或角色，才能使得用户组中的用户获得对应的权限，这一过程称为授权。授权后，用户就可以基于被授予的权限对云服务进行操作。

步骤2 在MRS管理控制台“现有集群” 列表，单击对应集群名称，进入集群信息页面。检查安全组中是否正确配置了本地IP。

- MRS 3.x及之后版本：请参考“用户指南 > 访问集群Manager> 访问FusionInsight Manager（MRS 3.x及之后版本）”中“通过弹性IP访问FusionInsight Manager”部分。

- MRS 3.x之前版本：请参考“用户指南 > 访问集群Manager> 访问MRS Manager（MRS 2.x及之前版本）”中“通过弹性公网IP访问Manager”部分。

----结束

3.2.3 通过专线访问 Manager 页面方式出现 503 报错

问题现象

通过专线方式访问Manager界面，出现503报错：当前无法使用此页面”。

原因分析

随着任务量的增大或者任务并发较高，有可能会导致Executor内存不足，进而导致访问Manager异常。

处理步骤

1. 通过**root**用户登录Master1或Master2其中任意一节点，执行以下命令切换到**omm**用户。

   su - omm

2. 执行以下命令，修改“catalina.sh”脚本。

   vim /opt/executor/bin/catalina.sh

   搜索“JAVA_OPTS” ，找到类似JAVA_OPTS=“-Xms1024m -Xmx4096m”的配置，将其值根据需要进行修改，保存修改。

3. 重启manager-executor进程，该进程在Master1和Master2节点上为主备部署，同一时刻只有一个节点上运行该进程，首先确认节点上是否有该进程，如果有该进程则进行重启操作。

   a. 分别登录Master1节点和Master2节点，执行以下命令确认当前节点是否存在该进程。当有输出时则进程存在。

      ps -ef | grep "/opt/executor" | grep -v grep

   b. 重启命令如下。

      sh /opt/executor/bin/shutdown.shsh /opt/executor/bin/startup.sh

4. 重新访问Manager页面。

3.2.4 登录 Manager 集群页面时报错“You have no right to access the page”

问题现象

访问集群管理页面报错：

```
You have no right to access the page.
```

原因分析

- 当前登录的用户没有访问Manager的权限。
- “/srv/BigData/dbdata_om”所在的分区磁盘空间被占满。

处理步骤

步骤1 使用**admin**用户登录到FusionInsight Manager用户管理界面，检查当前报错用户的权限。

步骤2 用户至少具有Manager的以下权限之一才可以访问。

角色名	来源	描述	创建时间	操作
Manager_administrator	OMS	Manager管理员：具有Manager和租户资源的所有权限。	--	修改 删除
Manager_auditor	OMS	Manager审计员：具有查看和管理Manager系统审计日志的权限。	--	修改 删除
Manager_operator	OMS	Manager操作员：具有查看和管理集群的权限。	--	修改 删除
Manager_tenant	OMS	Manager租户管理页面查看角色：具有查看租户管理页面的权限。	--	修改 删除
Manager_viewer	OMS	Manager查看员：具有查看集群和用户权限信息的权限。	--	修改 删除

步骤3 使用**root**或**omm**用户登录到主OMS节点。

步骤4 检查“/srv/BigData/dbdata_om”目录所在的分区是否已满。

步骤5 清理冗余的数据。

----结束

3.2.5 登录 Manager 时界面报错“认证信息无效”

问题现象

登录MRS集群的Manager界面时，输入用户名及密码信息后登录失败，界面提示“认证信息无效”。

原因分析

- 当前输入的用户名或密码信息错误。
- 当前登录用户的密码已过期。

处理步骤

步骤1 如果用户密码输入错误，请检查并输入正确密码。

MRS集群默认管理用户为admin，密码为创建集群时用户设置。

或参考**如何重置或修改Manager（admin）密码?** 章节重置用户密码。

步骤2 集群的用户密码过期或遗忘后，也可使用管理员登录Manager初始化该用户的密码。

1. 登录FusionInsight Manager。
2. 选择“系统 > 权限 > 用户”。
3. 在要初始化密码用户所在行，选择“更多 > 初始化密码”。在弹出窗口中输入当前登录的管理员用户密码确认身份，单击“确定”，在确认对话框单击“确定”后按照界面提示输入密码信息。

   用户密码重置后，登录Manager时需要再次修改密码。

----结束

3.2.6 Manager 界面超时退出后无法正常登录

现象描述

MRS Manager超时退出后偶现无法正常登录访问页面。

可能原因

超时后页面的js及css文件未加载成功。

操作步骤

步骤1 清除浏览器缓存。

步骤2 刷新当前页面即可正常登录及访问。

----结束

3.2.7 升级 Python 后无法登录 Manager 页面

用户问题

升级Python后，登录不进去MRS Manager页面。

问题现象

自行升级Python后，使用admin账号且密码正确的情况下登录不进去MRS Manager页面。

原因分析

用户升级Python版本到Python3.x的过程中，修改了openssl的文件目录权限，导致LdapServer服务无法正常启动，从而引起登录认证失败。

处理步骤

步骤1 以**root**用户登录集群的Master节点。

步骤2 执行**chmod 755 /usr/bin/openssl**命令，修改**/usr/bin/openssl**的文件目录权限为755。

步骤3 执行**su omm**命令，切换到omm用户。

步骤4 执行**openssl**命令，查看是否能够进入openssl模式。

如果能够成功进入，则表示权限修改成功，如果不能进入，则表示权限未修改成功。

如果权限未修改成功，请检查执行的命令是否正确，或者联系运维人员。

步骤5 权限修改成功后会重启LdapServer服务，请等待LdapServer服务重启成功后，重新登录MRS Manager。

----结束

建议与总结

自行安装的软件建议和系统的分开，系统软件升级可能造成兼容性问题。

3.2.8 修改集群域名后无法登录 Manager 页面

问题现象

用户修改域名后，通过console页面无法登录MRS Manager页面，或者登录MRS Manager页面异常。

原因分析

用户修改域名后，没有刷新executor用户的keytab文件，导致executor进程认证失败后不断循环认证，导致了acs进程内存溢出。

处理步骤

步骤1 重启acs进程。

1. 使用**root**用户登录主管理节点（即MRS集群详情页面“节点管理”页签下实心五角星所在的Master节点）。
2. 执行如下命令重启进程：

   su - omm

   ps -ef|grep =acs （查找acs进程PID）

   kill -9 *PID*（PID替换为实际的ID，结束acs进程）
3. 等待几分钟后执行命令**ps -ef|grep =acs**查询进程是否已经自动启动。

步骤2 替换executor用户的keytab文件。

1. 登录MRS Manager页面，选择“系统 > 用户”，在executor用户所在的“操作”列，单击“下载认证凭据”，解压后获取keytab文件。
2. 使用**root**用户登录主管理节点，将获取的keytab替换“/opt/executor/webapps/executor/WEB-INF/classes/user.keytab”文件。

步骤3 替换knox用户的keytab和conf文件。

1. 登录MRS Manager页面，选择“系统 > 用户”，在knox用户所在的“操作”列，单击“下载认证凭据”，解压后获取keytab和conf文件。
2. 使用root用户登录主管理节点，将获取的keytab替换“/opt/knox/conf/user.keytab”文件。
3. 修改/opt/knox/conf/krb5JAASLogin.conf中的principal的值，把域名修改为更改后的域名。
4. 将获取的krb5.conf替换“/opt/knox/conf/krb5.conf”文件。

步骤4 备份原有客户端目录

mv *{客户端目录}* **/opt/client_init**

步骤5 参考**更新客户端**章节，重新安装客户端。

步骤6 使用**root**用户登录主备管理节点，执行如下命令，重启knox进程。

su - omm

ps -ef | grep gateway | grep -v grep （查找knox进程PID）

kill -9 *PID*（PID替换为实际的ID，结束knox进程）

/opt/knox/bin/restart-knox.sh （启动knox进程）

步骤7 使用**root**用户登录主备管理节点，执行如下命令，重启executor进程。

su - omm

netstat -anp |grep 8181 |grep LISTEN （查找executor进程PID）

kill -9 *PID*（PID替换为实际的ID，结束executor进程）

/opt/executor/bin/startup.sh （启动executor进程）

----结束

3.2.9 登录 Manager 界面后页面空白无内容

用户问题

登录到FusionInsight Manager界面后，页面空白不显示。

问题现象

登录到FusionInsight Manager界面后，页面空白不显示。

原因分析

- Manager无法登录，需要清除浏览器缓存。
- Master节点的“/opt/Bigdata/tmp”目录权限被修改。

处理步骤

清除浏览器缓存

步骤1 切换至浏览器窗口（以Chrome为例），通过键盘按下“Ctrl+Shift+Delete”弹出“清除浏览数据”对话框。

步骤2 勾选待清除的浏览记录，单击“清除数据”，完成浏览器缓存清理。

步骤3 检查是否可以访问Manager。

- 是，操作结束。
- 否，执行**步骤4**。

修改“/opt/Bigdata/tmp”目录权限

步骤4 以**root**用户依次登录Master1、Master2节点。

步骤5 查看Master1、Master2节点的“/opt/Bigdata/tmp”路径权限是否被修改。（权限默认为770）

步骤6 如果被修改请执行以下命令恢复“/opt/Bigdata/tmp”路径权限。

chmod 1770 /opt/Bigdata/tmp

步骤7 登录主OMS节点，执行以下命令重启OMS。

sh ${BIGDATA_HOME}/om-server/om/sbin/restart-oms.sh

提示以下信息表示命令执行成功：

```
start HA successfully.
```

步骤8 等待一段时间后重新登录Manager。

----结束

3.2.10 集群节点内安装了原生 Kerberos 导致登录集群失败

问题现象

MRS集群创建成功后，登录集群Manager页面或者修改用户密码失败。

原因分析

1. 登录到集群的Kerberos服务节点，查看“/var/log/Bigdata/kerberos/krb5kdc.log”。
2. 登录到主、备OMS节点，查看“/var/log/Bigdata/okerberos/oms-krb5kdc.log”。

   发现有大量的认证失败，同时有 “<unknown client> for <unknown server>”错误提示。
3. 通过排查，发现是由于节点安装了原生的Kerberos导致认证异常。

   执行**rpm -qa | grep krb5**，观察是否有**krb5-workstation-***1.10.3-33.el6.x86_64*。

处理步骤

步骤1 执行以下命令卸载“krb5-workstation”软件和依赖的软件：

rpm -e *krb5-workstation-1.10.3-33.el6.x86_64*

步骤2 在Kerberos服务节点上使用**kill -9**命令结束Kerbeors的进程，等待其重启即可。

----结束

3.2.11 macOS 使用浏览器访问 MRS Manager

问题现象

在Mac电脑上无法使用Chrome浏览器访问MRS Manager。

原因分析

由于Mac电脑的证书限制导致无法访问。

处理步骤

步骤1 复制MRS Manager的登录链接，使用Chrome浏览器访问。

显示MRS Manager登录页面，此时Chrome浏览器暂时还不能登录MRS Manager页面，请继续操作后续步骤。

步骤2 查找证书。

单击浏览器左上角，单击“证书无效”所在行的。

步骤3 获取证书。

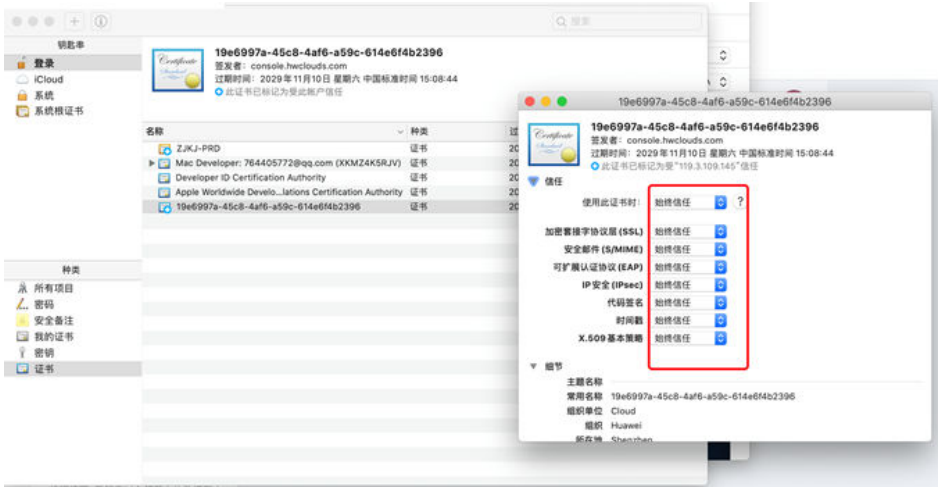
选中该证书拖动到电脑桌面，即可获得证书。

步骤4 在Mac应用中选择钥匙串访问。

步骤5 “钥匙串”选择“登录”，“种类”选择“证书”。

步骤6 将**步骤3**获得的证书拖动到钥匙串列表中。

步骤7 双击证书将该证书设置为“始终信任”并关闭弹窗。



步骤8 在出现的确认框中输入登录密码，此时在Chrome中即可访问MRS Manager。

----结束

3.2.12 Manager 界面登录用户被锁如何解锁

问题现象

1. Manager登录时显示：这个账户已经被锁住了，请联系系统管理员来重新获得访问权限。
2. Manager界面登录用户被锁，导致作业无法提交并无法进入Manager页面。
3. 通过后台提交作业执行**kinit** *用户名*操作时提示如下异常：
   kinit: Clients credentials have been revoked while getting initial credentials

原因分析

用户登录Manager账户密码错误次数超过限制，账户被锁。

处理步骤

步骤1 登录kadmin后台管理控制台。

1. 以**root**用户登录安装了客户端的节点，执行以下命令，切换到客户端目录，例如“/opt/client”。

   cd /opt/client
2. 执行以下命令，配置环境变量。

   source bigdata_env
3. 执行以下命令，输入kadmin/admin用户密码后进入kadmin控制台。

   kadmin -p kadmin/admin

   说明

   3.x及之后版本默认密码：Admin@123。3.x之前版本默认密码：KAdmin@123。
   首次登录后会提示该密码过期，请按照提示修改密码并妥善保存。

步骤2 执行以下命令 ，查看被锁用户数据密码错误次数“Failed password attempts”参数值是否大于或等于5，如果是，则证明账户被锁。

getprinc *被锁用户名*

步骤3 如果账户被锁，执行以下命令解锁用户后，重新登录Manager。

modprinc -unlock *被锁用户名*

----结束

3.2.13 Manager 页面加载卡顿

用户问题

MRS集群Manager页面加载卡顿。

问题现象

登录MRS集群的Manager页面，单击一个按钮需要等一分钟后才能加载到数据，部分按钮单击后卡顿无法访问。

原因分析

查看“/var/log/Bigdata/omm/oms/pms/scriptlog/pms_script.log”，“/var/log/Bigdata/omm/oms/pms/pms.log”日志内容类似如下：

```
# java.lang.OutOfMemoryError: GC overhead limit exceeded
# -XX:OnOutOfMemoryError="echo [ERROR]: JVM OutMemory >> /var/log/Bigdata/omm/oms/pms/scriptlog/pms_script.log;kill -9 %p"
#   Executing /bin/sh -c "echo [ERROR]: JVM OutMemory >> /var/log/Bigdata/omm/oms/pms/scriptlog/pms_script.log"...
[ERROR]: JVM OutMemory
#   Executing /bin/sh -c "kill -9 25000"...
/opt/Bigdata/om-server_8.1.0.1/OMS/workspace/bin/omm_s_pm_ctl.sh: line 334: 25000 Killed                  java -Dprocess.na
sd -Xms1024m -Xmx1024m -Xss256k -XX:OnOutOfMemoryError="echo [ERROR]: JVM OutMemory >> /var/log/Bigdata/omm/oms/pms/scriptl
s_script.log;kill -9 %p" -XX:ErrorFile=/var/log/Bigdata/omm/oms/pms/pmsdump/pms_error_211129223734.log -Djava.net.preferIPv
k=true -Djava.net.preferIPv6Addresses=false -Dbase.dir=/opt/Bigdata/om-server_8.1.0.1/OMS/workspace/lib -cp :/opt/Bigdata/o
ver_8.1.0.1/OMS/workspace/lib/snakeyaml-1.27.jar:/opt/Bigdata/om-server_8.1.0.1/OMS/workspace/lib/c3p0-0.9.5.4.jar:/opt/Big
om-server_8.1.0.1/OMS/workspace/lib/quartz-2.3.2.jar:/opt/Bigdata/om-server_8.1.0.1/OMS/workspace/lib/commons-collections4-
ar:/opt/Bigdata/om-server_8.1.0.1/OMS/workspace/lib/commons-io-2.8.0.jar:/opt/Bigdata/om-server_8.1.0.1/OMS/workspace/lib/g
20.0-jre.jar:/opt/Bigdata/om-server_8.1.0.1/OMS/workspace/lib/manager-pms-8.1.0-310013-All.jar:/opt/Bigdata/om-server_8.1.0
S/workspace/lib/manager-pms-8.1.0-310013-EXTERN.jar:/opt/Bigdata/om-server_8.1.0.1/OMS/workspace/lib/kafka-clients-2.4.0-ha
```

发现日志中有内存溢出的现象，判断是由于PMS内存不足导致Manager界面出现卡顿。

处理步骤

步骤1 以**omm**用户登录主管理节点。

步骤2 执行以下命令打开application.properties文件。

vi ${BIGDATA_HOME}/om-server_*/OMS/workspace0/conf/pms/application.properties

步骤3 将PMS进程的内存配置参数“pms.mem”根据实际情况调大，保存并退出。

```
...
pms.mem=800m
```

修改的新值的增长量不能大于操作系统可用剩余内存。

步骤4 执行以下命令查找PMS进程的PID，并终止PMS进程，等待进程重新启动。

ps -ef | grep pms

kill -9 *PID*

步骤5 登录Manager页面，查看卡顿情况消失。

----结束

3.3 访问 MRS 组件 Web UI 界面常见异常问题

3.3.1 使用 admin 用户访问组件的 WebUI 界面报错或部分功能不可用

问：

访问HDFS、Hue、Yarn、Flink、HetuEngine等组件的WebUI界面报错，或部分功能不可用怎么办？

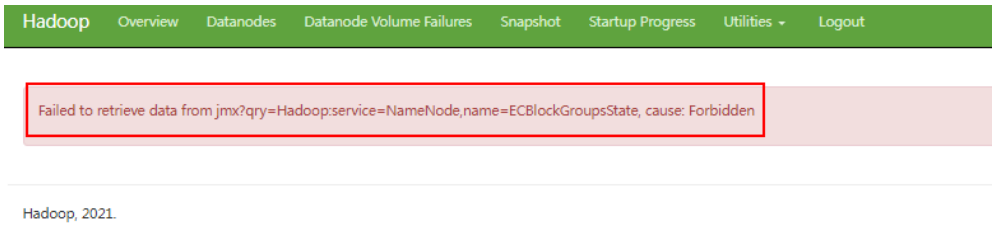
答：

访问HDFS、Hue、Yarn、Flink、HetuEngine等组件的WebUI的用户不具备对应组件的管理权限，导致界面报错或部分功能不可用。

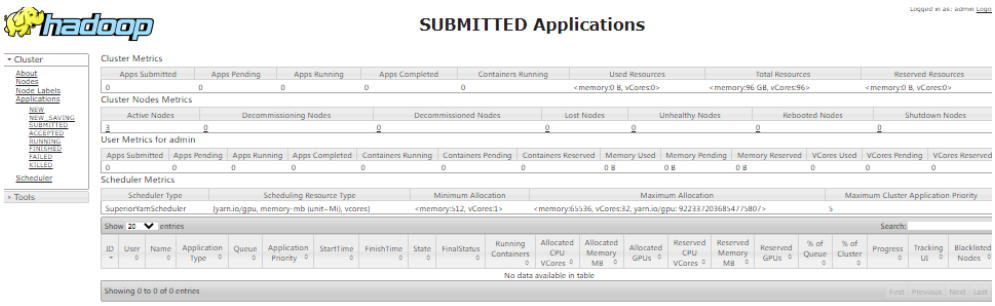
例如：

- 使用当前用户登录Flink WebUI后，部分内容不能正常显示，且没有权限创建应用、创建集群连接、创建数据连接等：

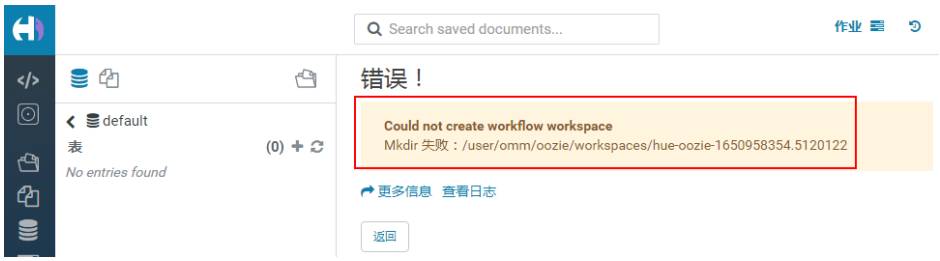
- 使用当前用户访问HDFS WebUI报错：Failed to retrieve data from /jmx?qry=java.lang:type=Memory, cause: Forbidden



- 使用当前用户访问Yarn WebUI界面，无法查看作业信息：



- 使用当前用户登录Hue WebUI后，在界面左侧导航栏单击 ，选择“Workflow”后报错：



建议使用新建的具有对于组件管理权限的用户访问，此时可以参考创建用户章节，创建一个业务用户，例如创建一个具有HDFS管理权限的用户登录并访问HDFS WebUI界面。

3.3.2 访问组件 WebUI 页面报 500 异常

用户问题

访问组件WebUI页面报500异常。

问题现象

访问组件WebUI页面报500异常。报错如下：

```
HTTP ERROR 500 java.io.IOException: Service connectivity error.
```

原因分析

出现该异常可能都是由于Knox线程数不足。

处理步骤

步骤1 以**root**用户分别登录OMS主节点和备节点。

步骤2 在主备OMS节点的gateway-site.xml文件中增加以下配置，适当增加线程池。

vi /opt/knox/conf/gateway-site.xml

```
<property>
<name>gateway.httpclient.maxConnections</name>
<value>64</value>
</property>
```

步骤3 以**omm**用户登录主OMS节点，执行以下命令重启knox进程。

sh /opt/knox/bin/restart-knox.sh

步骤4 等待五分钟，查看是否可以正常访问组件WebUI页面。

- 是，处理完毕。
- 否，请联系运维人员进行处理。

----结束

3.3.3 【HBase WebUI】无法从 HBase WebUI 界面跳转到 RegionServer WebUI

问题现象

MRS 1.9.3版本集群，在HBase WebUI界面单击“Region Servers”区域的“Base Status”页签中的“ServerName”列的任一RegionServer名称无法跳转到对应信息页。

操作步骤

步骤1 以**root**用户登录主OMS所在节点，执行以下命令获取RegionServer对应的URL。

cat /opt/Bigdata/Apache-httpd*/conf/whitelist.txt |grep RegionServer

```
[root@node-master1GGLT ~]# cat /opt/Bigdata/Apache-httpd*/conf/whitelist.txt |grep RegionServer
/HBase/RegionServer/44 https://node-ana-coreqacr0001.mrs-11ty.com:16030
/HBase/RegionServer/45 https://node-ana-coreqacr0002.mrs-11ty.com:16030
[root@node-master1GGLT ~]#
```

步骤2 修改HBase WebUI的HMaser的访问地址，将“/HMaster/*ID*/master-status”修改为“/RegionServer/*ID*/rs-status”，即可直接访问RegionServer信息页。其中，“/RegionServer/*ID*/rs-status”中的“*ID*”为**步骤1**查看到的RegionServer主机名称对应的ID。

例如，若查看到的RegionServer的URL为“/HBase/RegionServer/44”，则访问地址为“https://100.94.xxx.xxx:9022/component/HBase/RegionServer/44/rs-status”。

----结束

3.3.4 【HDFS WebUI】访问 HDFS WebUI 页面报错“重定向次数过多”

问题现象

通过Manager页面跳转HDFS WebUI页面偶现失败，报错“重定向次数过多”，此时其他组件的WebUI能正常访问。

原因分析

Knox为双活组件，运行过程中出现主备之间数据不同步问题导致。

处理步骤

步骤1 任意通过其他组件页面跳转对应WebUI页面，等待跳转成功。

步骤2 然后重新访问HDFS WebUI页面。

----结束

3.3.5 【HDFS WebUI】使用 IE 浏览器访问 HDFS WebUI 界面失败

用户问题

通过IE浏览器访问HDFS的WebUI界面，偶尔出现访问失败情况。

问题现象

访问页面失败，浏览器无法显示此页，如下图所示：

原因分析

部分低版本IE浏览器处理SSL握手有问题导致访问失败。

处理步骤

重新刷新页面或者推荐使用Google Chrome浏览器访问。

3.3.6 【Hue WebUI】登录 Hue WebUI 报错无权限

现象描述

开启Kerberos认证的集群，使用**admin**用户登录Hue WebUI，报错“无权限或无法连接到hive server，请检查当前用户的hive权限以及hue的hive配置是否正确。”或“No available hive server.....”

原因分析

对于开启Kerberos认证的集群，**admin**用户不具有操作Hue WebUI的权限。可以创建一个“人机”用户（例如**hueuser**），并加入“hive”、“hadoop”、“supergroup”用户组和“System_administrator”角色，主组为“hive”。

处理步骤

步骤1 登录FusionInsight Manager。

步骤2 选择“系统 > 权限 > 用户”。

步骤3 在用户列表上方，单击“添加用户”，并配置如下信息，其他参数保持默认。

- 用户名：自定义，例如**hueuser**。
- 用户类型：选择“人机”。
- 密码、确认新密码：根据界面要求输入自定义密码。
- 用户组：单击“添加”，选择“hive”、“hadoop”、“supergroup”，单击“确定”，
- 主组：选择“hive”。
- 角色：单击“添加”，选择“System_administrator”。

步骤4 单击“确定”完成用户创建。

步骤5 以新创建的用户（例如**hueuser**）重新访问Hue WebUI页面。

首次登录需要修改密码。

----结束

3.3.7 【Hue WebUI】Hue WebUI 页面无法访问

现象描述

Hue服务组件的Hue WebUI页面打不开。

原因分析

- 端口未对外开放。
- 浮动IP配置错误。
- 一次性查询大量的数据，导致查询耗时长，占用了Hue中Hive的链接，导致Hue访问出现异常。

排查思路

1. 在浏览器所在的本地机器，通过命令行执行**telnet**指令，查看端口是否开放。
2. 在Hue所在节点执行**ifconfig**，查看配置的浮动IP是否生效。
3. 恢复该故障大概需要10分钟。
4. 检查是否一次性查询了大量的数据。

处理步骤

步骤1 登录FusionInsight Manager管理界面。

步骤2 选择“集群 > *待操作集群的名称* > 服务 > Hue > 实例”，查看Hue服务主备是否正常。

- 是，执行**步骤7**。
- 否，执行**步骤3**。

步骤3 单击“配置”，检查Hue配置项“HUE_FLOAT_IP”和“HTTP_PORT”是否正确。

- 是，执行**步骤4**。
- 否，修改正确的Hue服务浮动IP地址与端口。

步骤4 在业务网络执行**telnet** *HUE节点IP HTTP_PORT*或者**telnet** *HUE_FLOAT_IP HTTP_PORT*，检查端口是否对外开放。

- 是，检查端口是否正确，重新尝试登录页面。
- 否，执行**步骤5**。

步骤5 选择“集群 > *待操作集群的名称* > 服务 > Hue”，单击“停止服务”。

步骤6 以**root**用户登录管理节点，执行**ping** *HUE_FLOAT_IP*，检查是否可以ping通。

- 是，浮动IP已被使用，请重新配置其他IP地址。
- 否，执行**步骤7**。

步骤7 检查是否一次性查询了大量的数据。

- 是，优化Hue查询，避免一次性查询大量数据。
- 否，执行**步骤8**。

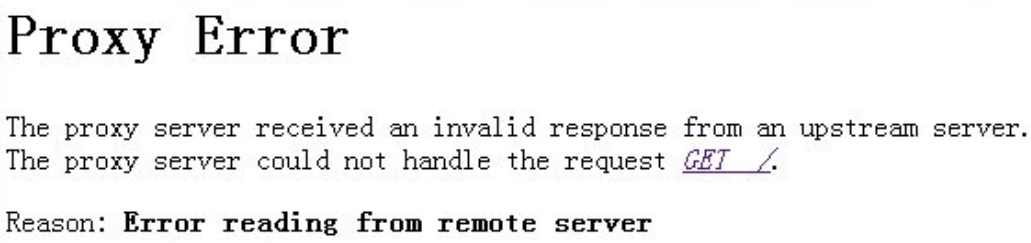
步骤8 联系运维人员。

----结束

3.3.8 【Hue WebUI】访问 Hue WebUI 页面报错“Proxy Error”

现象描述

打开Hue页面的时候提示“Proxy Error”。



原因分析

- 网络故障（如：错包率高、高延迟等）导致Httpd没有接收到报文或者接收到了错误报文。

- 时钟跳变或其他因素的影响导致Httpd服务与Hue服务认证关系失效。

排查思路

1. 排查是否发生过网络故障或时钟跳变。
2. 排查当前是否发生过或仍存在Hue服务相关告警。

处理步骤

步骤1 登录FusionInsight Manager检查是否存在网络故障、时钟跳变或Hue服务相关的告警，若存在问题则联系管理员进行处理，重试打开Hue页面确认是否正常。

- 是，处理结束。
- 否，执行**步骤2**。

步骤2 获取当前工作的Httpd服务器地址，为打开Hue页面URL中主机地址，即如果打开Hue的Url为“https://192.168.0.1:21201”，Httpd服务地址则为“192.168.0.1”。

步骤3 登录Httpd服务器，排查“/var/log/Bigdata/httpd/error_log”中的日志。

步骤4 确认“error_log”中是否存在“AH01102: error reading status line from remote server”和“AH00898: Error reading from remote server”字样的报错（可以多试几次打开Hue页面以确认情况）。

- 是，通过FusionInsight Manager重启Hue服务，执行**步骤5**。
- 否，执行**步骤6**。

步骤5 重启Hue服务后，再次打开Hue页面。

- 是，处理结束。
- 否，执行**步骤6**。

步骤6 联系运维。

----结束

3.3.9 【Hue WebUI】集群未安装 Hive 服务时 Hue 原生页面无法正常显示

用户问题

集群没有安装Hive服务时，Hue服务原生页面显示空白。

原因分析

MRS 3.x版本存在Hue依赖Hive组件。

处理步骤

如果出现此情况，首先需要检查当前集群是否安装了Hive组件，如果没有，需要安装Hive。

3.3.10 【Hue WedUI】Hue（主）无法打开 web 网页

问题背景与现象

访问Hue（主）的WebUI界面提示如下：

```
Service Unavailable
The server is temporarily unable to service your request due to maintenance downtime or capacity problems. Please try again later.
```

原因分析

- Hue配置过期。
- MRS 2.0.1及之后版本的单Master节点集群中，Hue服务需要手动修改配置。

解决办法

- Hue配置过期，重启Hue服务即可。
- 在MRS 2.0.1及之后版本，单Master节点的集群Hue服务需要手动修改配置。
  a. 登录Master节点。
  b. 执行**hostname -i**获取本机IP。
  c. 执行如下命令获取“HUE_FLOAT_IP”的地址：
     grep "HUE_FLOAT_IP" ${BIGDATA_HOME}/*MRS*_Current/1_*/etc*/ENV_VARS，
     其中*MRS*以实际文件名为准。
  d. 比较本机IP和“HUE_FLOAT_IP”的值是否相同，若不相同，请修改“HUE_FLOAT_IP”的值为本机IP。
  e. 重启Hue服务。

3.3.11 【Ranger WebUI】新用户修改完密码后无法登录 Ranger WebUI 界面

问题

使用新建用户登录Ranger页面，为什么在修改完密码后登录报401错误?

回答

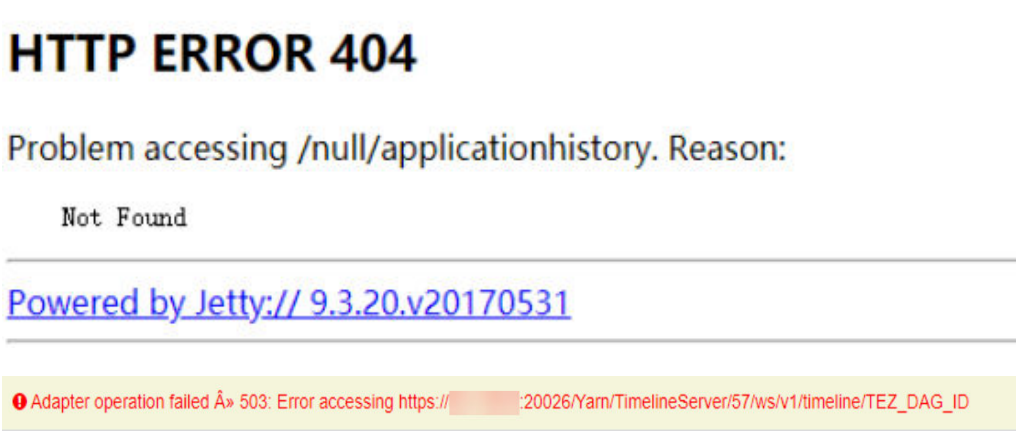
由于UserSync同步用户数据有时间周期，默认是5分钟，因此在Manager上新创建的用户在用户同步成功前无法登录Ranger，因为Ranger的DB里暂时还没有该用户信息，需要等待同步周期所设置的时间后再尝试登录。

未开启Kerberos认证时，由于Ranger并不从Manager同步用户数据，因此，仅有admin用户可以登录Ranger，暂时不支持其他用户登录。

3.3.12 【Tez WebUI】访问 Tez WebUI 界面报错 404

用户问题

登录Manager界面，跳转Tez WebUI界面，显示404异常或503异常：



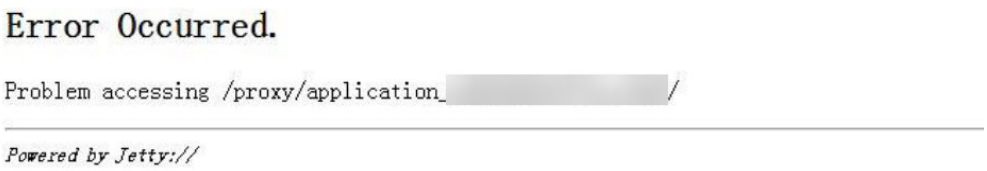
回答

Tez WebUI依赖Yarn的TimelineServer实例，需要预先安装TimelineServer，且处于良好状态。

3.3.13 【Spark WebUI】为什么从 Yarn Web UI 页面无法跳转到 Spark WebUI 界面

用户问题

在客户端采用yarn-client模式运行Spark应用，然后从Yarn的页面打开该应用的Web UI界面，出现下面的错误：



从Yarn ResourceManager的日志看到：

```
2021-07-21 16:35:27,099 | INFO  | Socket Reader #1 for port xxx | Auth successful for mapred/hadoop.<系统域名>@<系统域名> (auth:KERBEROS) | Server.java:1388
2021-07-21 16:35:27,105 | INFO  | abc6016381123-1178290888-1015 | admin is accessing unchecked http://10.120.169.53:23011 which is the app master GUI of application_1468986660719_0045 owned by spark | WebAppProxyServlet.java:393
2021-07-21 16:36:02,843 | INFO  | Socket Reader #1 for port xxx | Auth successful for hive/hadoop.<系统域名>@<系统域名> (auth:KERBEROS) | Server.java:1388
2021-07-21 16:36:02,851 | INFO  | Socket Reader #1 for port xx | Auth successful for hive/hadoop.<系统域名>@<系统域名> (auth:KERBEROS) | Server.java:1388
2021-07-21 16:36:12,163 | WARN  | abc6016381123-1178290888-1015 | /proxy/application_1468986660719_0045/: java.net.ConnectException: Connection timed out | Slf4jLog.java:76
2021-07-21 16:37:03,918 | INFO  | Socket Reader #1 for port xxx | Auth successful for hive/hadoop.<系统域名>@<系统域名> (auth:KERBEROS) | Server.java:1388
2021-07-21 16:37:03,926 | INFO  | Socket Reader #1 for port xxx | Auth successful for hive/hadoop.<系统域名>@<系统域名> (auth:KERBEROS) | Server.java:1388
2021-07-21 16:37:11,956 | INFO  | AsyncDispatcher event handler | Updating application attempt appattempt_1468986660719_0045_000001 with final state: FINISHING,
and exit status: -1000 | RMAppAttemptImpl.java:1253
```

原因分析

打开FusionInsight Manager页面，看到Yarn服务的业务IP地址为192网段。

从Yarn的日志看到，Yarn读取的Spark Web UI地址为http://10.10.169.53:23011，是10网段的IP地址。由于192网段的IP和10网段的IP不能互通，所以导致访问Spark Web UI界面失败。

处理步骤

步骤1 登录10.10.169.53客户端机器。

步骤2 修改“/etc/hosts”文件，将10.120.169.53更改为相对应的192网段的IP地址。

步骤3 修改再重新运行Spark应用，这时就可以打开Spark Web UI界面。

----结束

3.3.14 【Spark WebUI】HistoryServer 缓存的应用被回收导致此类应用页面访问时出错

用户问题

在History Server页面中访问某个Spark应用的页面时，发现访问时出错。

查看相应的HistoryServer日志后，发现有“FileNotFound”异常，相关日志如下所示：

```
2020-11-22 23:58:03,694 | WARN  | [qtp55429210-232] | /history/application_1479662594976_0001/stages/stage/ | org.sparkproject.jetty.servlet.ServletHandler.doHandle(ServletHandler.java:628)
java.io.FileNotFoundException: ${BIGDATA_HOME}/tmp/spark/jobHistoryTemp/blockmgr-5f1f6aca-2303-4290-9845-88fa94d78480/09/temp_shuffle_11f82aaf-e226-46dc-b1f0-002751557694 (No such file or directory)
```

原因分析

在History Server页面加载Task个数较多的Spark应用时，由于无法把全部的数据放入内存中，导致数据溢出到磁盘时，会产生前缀为“temp_shuffle”的文件。

HistoryServer默认会缓存50个Spark应用（由配置项“spark.history.retainedApplications”决定），当内存中的Spark应用个数超过这个数值时，HistoryServer会回收最先缓存的Spark应用，同时会清理掉相应的“temp_shuffle”文件。

当用户正在查看即将被回收的Spark应用时，可能会出现找不到“temp_shuffle”文件的错误，从而导致当前页面无法访问。

处理步骤

如果遇到上述问题，可参考以下两种方法解决。

- 重新访问这个Spark应用的HistoryServer页面，即可查看到正确的页面信息。
- 如果用户场景需要同时访问50个以上的Spark应用时，需要调大“spark.history.retainedApplications”参数的值。

  请登录FusionInsight Manager管理界面，选择“集群 > 服务 > Spark2x > 配置”，单击“全部配置”，在左侧的导航列表中，单击“JobHistory2x > 界面”，配置如下参数。

表 3-1 参数说明

参数	描述	默认值
spark.history.retainedApplications	HistoryServer缓存的Spark应用数，当需要缓存的应用个数超过此参数值时，HistoryServer会回收最先缓存的Spark应用。	50

3.3.15 【Spark WebUI】访问 JobHistory 中某个应用的原生页面时页面显示错误

用户问题

提交一个Spark应用，包含单个Job百万个Task。应用结束后，在JobHistory中访问该应用的WebUI页面，浏览器会等待较长时间才跳转到应用WebUI页面，若10分钟内无法跳转，则页面会显示Proxy Error信息。

图 3-3 错误信息样例

原因分析

在JobHistory界面中跳转到某个应用的WebUI页面时，JobHistory需要回放该应用的Event log，若应用包含的事件日志较大，则回放时间较长，浏览器需要较长时间的等待。

当前浏览器访问JobHistory WebUI页面需经过httpd代理，代理的超时时间是10分钟，因此，如果JobHistory在10分钟内无法完成Event log的解析并返回，httpd会主动向浏览器返回Proxy Error信息。

解决方法

由于当前JobHistory开启了本地磁盘缓存功能，访问应用时，会将应用的Event log的解析结果缓存到本地磁盘中，第二次访问时，能加快响应速度。因此，出现此种情况时，仅需稍作等待，重新访问原来的链接即可，此时不会再出现需要长时间等待的现象。

3.3.16 【Spark WebUI】使用 IE 浏览器访问 Spark2x WebUI 界面失败

用户问题

通过IE浏览器访问Spark的WebUI界面，偶尔出现访问失败情况。

问题现象

访问页面失败，浏览器无法显示此页，如下图所示：

原因分析

部分低版本IE浏览器处理SSL握手有问题导致访问失败。

处理步骤

重新刷新页面或者推荐使用Google Chrome浏览器访问。

3.3.17 【Yarn WebUI】无法访问 Yarn WebUI

问题现象

集群扩容到300节点后，无法访问Yarn WebUI界面。

原因分析

可能是由于集群节点较多时，NodeManager数据增加，但是未修改实例的内存，导致ResourceManager进程的垃圾回收时间过长，影响ResourceManager进程正常提供服务，在访问YARN的原生界面时异常。

此时建议修改实例的内存。

处理步骤

步骤1 登录FusionInsight Manager，选择“集群 > 服务 > Yarn > 配置 > 全部配置 > ResourceManager > 系统”。将“GC_OPTS”参数根据实际情况调大。

说明

集群中的NodeManager实例数量和ResourceManager内存大小的对应关系参考如下：

- 集群中的NodeManager实例数量达到100，ResourceManager实例的JVM参数建议配置为：-Xms4G -Xmx4G -XX:NewSize=512M -XX:MaxNewSize=1G
- 集群中的NodeManager实例数量达到200，ResourceManager实例的JVM参数建议配置为：-Xms6G -Xmx6G -XX:NewSize=512M -XX:MaxNewSize=1G
- 集群中的NodeManager实例数量达到500，ResourceManager实例的JVM参数建议配置为：-Xms10G -Xmx10G -XX:NewSize=1G -XX:MaxNewSize=2G
- 集群中的NodeManager实例数量达到1000，ResourceManager实例的JVM参数建议配置为：-Xms20G -Xmx20G -XX:NewSize=1G -XX:MaxNewSize=2G

步骤2 保存配置，并重启该ResourceManager实例。

步骤3 重新访问Yarn WebUI界面，如果依旧无法访问，请联系运维人员进行处理。

----结束

4 API 使用类

4.1 调用 API 接口创建集群失败

用户问题

按照参考文档创建集群并执行作业，调用创建集群接口：POST /v1.1/{project_id}/run-job-flow返回500。

问题现象

调用创建集群接口：POST /v1.1/{project_id}/run-job-flow返回500。

原因分析

用户没有创建集群的权限。

处理步骤

检查是否拥有创建MRS集群的权限，可参考文档权限管理。

5 集群管理类

5.1 缩容 Task 节点失败

问题现象

在MRS 2.x集群详情页面调整集群Task节点，将Task节点调整成0个，最终缩容失败，提示如下：

```
This operation is not allowed because the number of instances of NodeManager will be less than the minimum configuration after scale-in, which may cause data loss.
```

原因分析

用户将Core节点的NodeManager服务停止了，导致在检查Task节点退服过程中发现Task如果全部退订，则将没有NodeManager，则Yarn服务就不可用，而MRS判断剩余的NodeManager必须大于等于1才能退服Task节点。

处理步骤

步骤1 进入Yarn服务实例页面：

- MRS 3.*x*之前版本，单击集群名称，登录集群详情页面，选择“组件管理 > Yarn > 实例”。

  说明

  如果集群详情页面没有“组件管理”页签，请先完成IAM用户同步（在集群详情页的“概览”页签，单击“IAM用户同步”右侧的“同步”进行IAM用户同步）。

- MRS 3.x及后续版本，登录FusionInsight Manager。然后选择“集群 > *待操作的集群名*称 > 服务 > Yarn > 实例”。

步骤2 勾选Core节点的NodeManager实例，选择“更多 > 启动实例”。

步骤3 在集群列表页面缩容Task节点。

1. 单击集群名称进入集群详情页面，选择“节点管理”。
2. 在Task节点组所在行的“操作”列单击“缩容”。

图 5-1 缩容 Task

3. 单击“确定”并在弹出框选择“是”。

步骤4 等缩容成功后，若不想用Core节点的NodeManager再将其停止。

----结束

建议与总结

Core节点的NodeManager通常不会将其停止，建议不要随意变更集群部署结构。

5.2 如何处理集群内部 OBS 证书过期

用户问题

用户在MRS集群中访问OBS服务过程中出现证书过期问题。

问题现象

MRS集群产生“ALM-12054 证书文件失效”或“ALM-12055 证书文件即将过期”告警，且告警详情中触发告警的证书为OBS证书。

图 5-2 OBS 证书即将过期告警

告警ID:	12055	告警名称:	证书文件即将过期
告警级别:	次要	来源:	OMS
产生时间:	2023/05/08 20:44:40 GMT+08:00	清除时间:	--
对象:	controller	是否自动清除:	是
告警状态:	未清除	告警原因:	证书即将过期
序列号:	118	附加信息:	Detail=Certificate availability time < 30 days
定位信息:	来源=OMS;服务名=controller;角色名=cacerts(Alias name: obs. i.com);主机名 = node-master1BhJL		

图 5-3 OBS 证书失效告警

告警ID:	12054	告警名称:	证书文件失效
告警级别:	重要	来源:	OMS
产生时间:	2024/05/08 18:26:08 GMT+08:00	清除时间:	2023/05/08 20:44:37 GMT+08:00
对象:	controller	是否自动清除:	是
告警状态:	自动清除	告警原因:	证书过期
序列号:	103	附加信息:	Detail=Certificate has expired
定位信息:	来源=OMS;服务名=controller;角色名=cacerts(Alias name: obs. .com);主机名 = node-master1BhJL		

原因分析

OBS系统生成的证书有有效期限制，到达有效期后，证书文件失效，因此产生告警。

处理步骤

步骤1 查询MRS集群上的OBS证书信息。

使用**root**用户登录MRS集群的主OMS节点，执行以下命令查询是否存在OBS相关证书。

keytool -v -list -keystore *${JAVA_HOME}*/jre/lib/security/cacerts -protected 2> /dev/null|grep -E "Alias name*|Valid from*" | grep obs

如下示例表示存在OBS证书：

```
Alias name: obs.example.com
```

- 若不存在证书，则无需进行后续操作，等待告警恢复。
- 若存在证书，则执行**步骤2**。

说明

- *${java_home}*为集群JDK目录，MRS 3.x版本中，替换为“/opt/Bigdata/common/runtime0/jdk1.8*”，MRS 3.x之前版本中，替换为“/opt/Bigdata/jdk”。
- MRS 3.x版本集群若按照该指导执行后依然出现证书过期告警，请将*${JAVA_HOME}*替换为“*客户端安装目录*/JDK/jdk”，再次执行该指导。

步骤2 删除OBS证书。

在主OMS节点中，执行以下命令删除**步骤1**中查询到的OBS证书。

obs_url=$(keytool -v -list -keystore *${JAVA_HOME}*/jre/lib/security/cacerts -protected 2> /dev/null|grep -E "Alias name*|Valid from*" | grep obs | cut -d ':' -f 2 | awk '$1=$1')

jdk_cacert="*${JAVA_HOME}*/jre/lib/security/cacerts"

keytool -delete -alias ${obs_url} -keystore ${jdk_cacert} -storepass changeit

步骤3 执行以下命令，确认OBS证书已不存在，若仍存在，则继续执行**步骤2**。

keytool -v -list -keystore *${JAVA_HOME}*/jre/lib/security/cacerts -protected 2> /dev/null|grep -E "Alias name*|Valid from*" | grep obs

----结束

5.3 MRS 集群更换磁盘（适用于 2.x 及之前）

用户问题

磁盘无法访问。

问题现象

客户创建本地盘系列MRS集群，其中1个Core节点的磁盘存在硬件损坏，导致读取文件失败。

原因分析

磁盘硬件故障。

处理步骤

说明

该指导适用于MRS 3.x之前版本分析集群，如需为流式集群或混合集群更换磁盘，请联系华为云技术支持处理。

步骤1 登录**MRS Manager**。

步骤2 选择“主机管理”并单击需要更换磁盘的主机的“主机名称”，在“角色”列表中单击RegionServer，选择“更多 > 退服”。

步骤3 选择“主机管理”并单击需要更换磁盘的主机的“主机名称”，在“角色”列表中单击DataNode，选择“更多 > 退服”。

步骤4 选择“主机管理”并单击需要更换磁盘的主机的“主机名称”，在“角色”列表中单击NodeManager，选择“更多 > 退服”。

说明

该主机下若还有其他实例，请参考该步骤方式进行退服。

步骤5 执行**vim /etc/fstab**命令编辑注释旧磁盘的挂载点。

图 5-4 注释旧磁盘的挂载点

```
[root@node-ana-coregexX0001 ~]# vim /etc/fstab

devpts   /dev/pts          devpts   mode=0620,gid=5 0 0
proc     /proc             proc     defaults        0 0
sysfs    /sys              sysfs    noauto          0 0
debugfs  /sys/kernel/debug debugfs  noauto          0 0
tmpfs    /run              tmpfs    noauto          0 0
/dev/disk/by-label/ROOT / ext4 defaults,noatime 1 1
#UUID=0f871b41-61e0-4f7f-af54-a03a1bfb3753 /srv/BigData/hadoop/data1 ext4 defaults,noatime,nodiratime 1 0
~
~
```

步骤6 如果旧磁盘仍可访问，迁移旧磁盘上（例如：/srv/BigData/hadoop/data1/）的用户自有数据。

步骤7 登录MRS管理控制台。

步骤8 在集群详情页面，选择“节点管理”。

步骤9 单击待更换磁盘的“节点名称”进入弹性云服务器管理控制台，单击“关机”。

步骤10 联系华为云支持人员在后台更换磁盘。

步骤11 在弹性云服务器管理控制台，单击“开机”，将已更换磁盘的节点开机。

步骤12 执行**fdisk -l**命令，查看新增磁盘。

步骤13 使用**cat /etc/fstab**获取盘符。

图 5-5 获取盘符

```
[omm@node-master1dGom ~]$ cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Wed Feb 27 06:58:49 2019
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=b13ee9c8-0ef0-4159-9b90-fc47bde0d464 /                       ext4    defaults,noatime        1 1
UUID=029408e0-71a6-4f73-b817-42d7049b7595 /srv/BigData ext4 defaults,noatime,nodiratime 1 0
UUID=f9cb8844-dabf-4a69-aff4-587de2fc4d7c /srv/BigData1 ext4 defaults,noatime,nodiratime 1 0
UUID=876e73be-1f80-4466-92b7-01d7c68bbb1b /srv/BigData2 ext4 defaults,noatime,nodiratime 1 0
UUID=0d5fce7f-afd0-420a-b1bb-e5500a1851cd /srv/BigData3 ext4 defaults,noatime,nodiratime 1 0
```

步骤14 使用对应的盘符对新磁盘进行格式化。

例如：**mkfs.ext4 /dev/sdh**

步骤15 执行如下命令挂载新磁盘。

mount *新磁盘 挂载点*

例如：**mount /dev/sdh /srv/BigData/hadoop/data1**

步骤16 执行如下命令为新磁盘增加omm用户权限。

chown omm:wheel *挂载点*

例如：**chown -R omm:wheel /srv/BigData/hadoop/data1**

步骤17 在fstab文件中新增新磁盘UUID信息。

1. 使用**blkid**命令查看新磁盘的UUID。

```
[root@node-ana-coreKpoT0003 ~]# blkid
/dev/vda1: LABEL="ROOT" UUID="2aa97872-11ec-422e-9513-0f28b925ad5e" TYPE="ext4"
/dev/vdb: UUID="e5f652c3-f9af-427f-89da-f2545618688d" TYPE="ext4"
[root@node-ana-coreKpoT0003 ~]#
```

2. 打开“/etc/fstab”文件，新增如下信息：

```
UUID=新盘UUID /srv/BigData/hadoop/data1 ext4 defaults,noatime,nodiratime 1 0
```

步骤18 （可选）执行如下命令新建日志目录。

mkdir -p /srv/BigData/Bigdata

chown omm:ficommon /srv/BigData/Bigdata

chmod 770 /srv/BigData/Bigdata

说明

执行如下命令确认Bigdata日志软链接目录是否已存在，若存在则忽略该步骤。

ll /var/log

步骤19 登录**MRS Manager**。

步骤20 选择“主机管理”并单击更换了磁盘的主机的“主机名称”，在“角色”列表中单击RegionServer，选择“更多 > 入服”。

步骤21 选择“主机管理”并单击更换了磁盘的主机的“主机名称”，在“角色”列表中单击DataNode，选择“更多 > 入服”。

步骤22 选择“主机管理”并单击更换了磁盘的主机的“主机名称”，在“角色”列表中单击NodeManager，选择“更多 > 入服”。

说明

该主机下若还有其他实例，请参考该步骤方式进行入服。

步骤23 选择“服务管理 > HDFS”，在“服务状态”页签的“HDFS概述”模块查看“丢失块数”是否为“0”。

- “丢失块数”是为“0”，则操作完成。
- “丢失块数”不为“0”，请联系华为云支持人员进行处理。

----结束

5.4 MRS 集群更换磁盘（适用于 3.x）

用户问题

磁盘无法访问。

问题现象

客户创建本地盘系列MRS集群，其中1个Core节点的磁盘存在硬件损坏，导致读取文件失败。

原因分析

磁盘硬件故障。

处理步骤

说明

该指导适用于本地盘系列（d/i/ir/ki系列）MRS集群，针对Core、Task类型节点的磁盘存在硬件故障。

Kafka组件不支持更换磁盘，如果存储Kafka数据的节点故障，请联系华为云技术支持处理。

步骤1 登录**FusionInsight Manager**。

步骤2 选择“主机”并单击故障主机的“主机名称”，在“实例”列表中单击DataNode，选择“更多 > 退服”。

说明

- 该主机下若存在DataNode、NodeManager、RegionServer和ClickHouseServer实例，请参考该步骤进行退服操作；
- MRS 3.1.2版本之后支持退服ClickHouseServer角色实例。

步骤3 选择“主机”并勾选故障主机“主机名称”前的复选框，选择“更多 > 停止所有实例”。

步骤4 执行**vim /etc/fstab**命令编辑注释旧磁盘的挂载点。

图 5-6 注释旧磁盘的挂载点

```
[root@node-ana-coreXZYb0001 ~]# vim /etc/fstab


#
# /etc/fstab
# Created by anaconda on Sat Feb 27 07:10:42 2021
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=c89eca08-5da4-43de-add0-4bb58e820d78 /                       ext4    defaults,errors=panic,noatime        1 1
UUID=4b16f96b-6d16-4dbe-9517-9f63423f9f6e /tmp  ext4   defaults,noatime,nodiratime,errors=panic 1 0
UUID=e539a0fd-a639-41dc-aa88-5fdc0e4bb7b3 /var ext4    defaults,noatime,nodiratime,errors=panic 1 0
UUID=51ba7a26-67de-4762-8bea-85fc004065c2 /srv/BigData ext4   defaults,noatime,nodiratime 1 0
UUID=03ba5f78-d188-4e6b-b640-1915b858183a /var/log  ext4   defaults,noatime,nodiratime,errors=panic 1 0
# UUID=91c84554-22eb-4130-a7a1-5ceaf03c8c06 /srv/BigData/data1 ext4 defaults,noatime,nodiratime,nodev 1 0
```

步骤5 如果旧磁盘仍可访问，迁移旧磁盘上（例如：/srv/BigData/data1/）的用户自有数据。

cp -r ***旧磁盘挂载点 临时数据保存目录***

例如：**cp -r /srv/BigData/data1 /tmp/**

步骤6 登录MRS管理控制台。

步骤7 在集群详情页面，选择“节点管理”。

步骤8 单击待更换磁盘的“节点名称”进入弹性云服务器管理控制台，单击“关机”。

步骤9 联系华为云支持人员在后台更换磁盘。

步骤10 在弹性云服务器管理控制台，单击“开机”，将已更换磁盘的节点开机。

步骤11 初始化Linux数据盘。

具体步骤可参见初始化数据盘对新磁盘进行初始化操作。

步骤12 执行**lsblk**命令，查看新增磁盘分区信息。

图 5-7 查看新增磁盘（分区）

```
[root@ecs-fcq ~]# lsblk
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda       8:0    0  1.7T  0 disk
sdb       8:16   0  1.7T  0 disk
sdc       8:32   0  1.7T  0 disk
└sdc1     8:33   0  1.7T  0 part
sdd       8:48   0  1.7T  0 disk
└sdd1     8:49   0  1.7T  0 part
```

步骤13 使用**df -TH**获取文件系统类型。

图 5-8 获取文件系统类型

```
[root@node-ana-coreWQaI0001 ~]# df -TH
Filesystem     Type      Size  Used Avail Use% Mounted on
/dev/vda1      ext4      233G   44G  179G  20% /
devtmpfs       devtmpfs   34G     0   34G   0% /dev
tmpfs          tmpfs      34G     0   34G   0% /dev/shm
tmpfs          tmpfs      34G  9.3M   34G   1% /run
tmpfs          tmpfs      34G     0   34G   0% /sys/fs/cgroup
/dev/vda5      ext4       11G   40M   10G   1% /tmp
/dev/vda7      ext4       64G  152M   60G   1% /srv/BigData
/dev/vda6      ext4       11G  1.2G  8.9G  12% /var
/dev/vda8      ext4      190G  211M  180G   1% /var/log
/dev/sdc1      ext4      1.8T  1.4G  1.8T   1% /srv/BigData/data2
tmpfs          tmpfs     6.8G     0  6.8G   0% /run/user/2000
tmpfs          tmpfs     6.8G     0  6.8G   0% /run/user/0
[root@node-ana-coreWQaI0001 ~]#
```

步骤14 使用对应的文件系统类型对新磁盘（分区）进行格式化。

例如：**mkfs.ext4 /dev/sdd1**

步骤15 执行如下命令挂载新磁盘。

mount *新磁盘 挂载点*

例如：**mount /dev/sdd1 /srv/BigData/data1**

说明

如果挂载不上，请执行如下命令重载配置后重新挂载。

systemctl daemon-reload

步骤16 执行如下命令为新磁盘增加omm用户权限。

chown omm:wheel *挂载点*

例如：**chown -R omm:wheel /srv/BigData/data1**

步骤17 将旧磁盘上（例如：/srv/BigData/data1/）的用户自有数据迁移到新磁盘上。

cp -r*临时数据保存目录 新磁盘挂载点*

例如：**cp -r /tmp/data1/* /srv/BigData/data1/**

步骤18 在fstab文件中新增新磁盘UUID信息。

1. 使用**blkid**命令查看新磁盘的UUID。

```
[root@node-ana-coreWQaI0001 ~]# blkid
/dev/vda6: UUID="e539a0fd-a639-41dc-aa88-5fdc0e4bb7b3" TYPE="ext4"
/dev/vda1: UUID="c89eca08-5da4-43de-add0-4bb58e820d78" TYPE="ext4"
/dev/vda5: UUID="4b16f96b-6d16-4dbe-9517-9f63423f9f6e" TYPE="ext4"
/dev/vda7: UUID="51ba7a26-67de-4762-8bea-85fc004065c2" TYPE="ext4"
/dev/vda8: UUID="03ba5f78-d188-4e6b-b640-1915b858183a" TYPE="ext4"
/dev/sda1: UUID="02a09811-ae36-4140-abad-e5ef935e54e0" TYPE="ext4" PARTLABEL="logical" PARTUUID="1bd64663-42e1-4bdf-9ece-4b5b793
cf799"
/dev/sdc1: UUID="570ceafe-4505-462a-a358-e12408969d7f" TYPE="ext4" PARTLABEL="logical" PARTUUID="ac309415-3294-47c4-b009-ae39fc7
2f62e"
/dev/sdd1: UUID="7f377c8b-e1b9-423e-b7d2-a60e1d58c3eb" TYPE="ext4" PARTLABEL="logical" PARTUUID="7f0254ea-306c-46ae-b358-8e30456
e5120"
/dev/sdb1: UUID="67133dc9-da39-4561-9353-602257347cc1" TYPE="ext4" PARTLABEL="logical" PARTUUID="2004ff81-e343-4f41-bfe8-889b4bd
30960"
[root@node-ana-coreWQaI0001 ~]#
```

2. 打开“/etc/fstab”文件，新增如下信息：

```
UUID=新盘UUID /srv/BigData/data1 ext4 defaults,noatime,nodiratime,nodev 1 0
```

步骤19 登录**FusionInsight Manager**。

步骤20 选择“主机”并单击需要入服主机的“主机名称”，在“实例”列表中单击DataNode，选择“更多 > 入服”。

说明

- 该主机下若存在DataNode、NodeManager、RegionServer和ClickHouseServer实例，请参考该步骤进行入服操作；
- MRS 3.1.2版本之后支持入服ClickHouseServer角色实例。

步骤21 选择“主机”，并勾选故障主机“主机名称”前的复选框，选择“更多 > 启动所有实例”。

步骤22 选择“集群 > HDFS”，在“概览”页签的“基本信息”模块查看“丢失块数”是否为“0”。

- “丢失块数”是为“0”，则操作完成。
- “丢失块数”不为“0”，请联系华为云支持人员进行处理。

----结束

5.5 MRS 备份任务执行失败

问题现象

MRS的备份任务总是执行失败。

原因分析

备份目录软链接到系统盘，系统盘满了之后备份任务失败。

处理步骤

步骤1 检查备份目录是否软链接到系统盘。

1. 以**root**用户登录集群主备Master节点。
2. 执行**df -h**命令查看磁盘情况，检查系统盘的存储情况。
3. 执行 **ll /srv/BigData/LocalBackup**命令， 查看备份目录是否软连接到**/opt/Bigdata/LocalBackup**。

   检查备份文件是否软链接到系统盘且系统盘空间是否足够。如果软链接到系统盘且系统盘空间不足，请执行**步骤2**。如果否，说明不是由于系统盘空间不足导致，请联系技术服务。

步骤2 将历史备份数据移到数据盘的新目录中。

1. 以**root**用户登录Master节点。
2. 执行**su - omm**命令，切换到omm用户。
3. 执行**rm -rf /srv/BigData/LocalBackup**命令，删除备份目录软连接。
4. 执行**mkdir -p /srv/BigData/LocalBackup**命令，创建备份目录。
5. 执行**mv /opt/Bigdata/LocalBackup/* /srv/BigData/LocalBackup/**命令，将历史备份数据移到新目录。

步骤3 重新执行备份任务。

----结束

5.6 Core 节点出现 df 显示的容量和 du 显示的容量不一致

问题现象

Core节点出现df显示的容量和du显示的容量不一致：

分别使用命令**df -h** 和命令**du -sh /srv/BigData/hadoop/data1/**查询得到的/srv/BigData/hadoop/data1/目录磁盘占用量相差较大（大于10 GB）。

原因分析

使用命令**lsof |grep deleted**可以查询到此目录下有大量log文件处于deleted状态。

出现此问题的一种情况是长时间运行某些Spark任务，任务中的一些Container一直运行，并且持续产生日志；Spark的executor在打印日志的时候使用了log4j的日志滚动功能，将日志输出到stdout文件下；而Container同时也会监控这个文件，导致此文件被两个进程同时监控。当其中一个进程按照配置滚动的时候，删除了最早的日志文件，但是另一个进程依旧占用此文件句柄，从而产生了deleted状态的文件。

处理步骤

将Spark的executor日志输出目录修改成其他名称：

步骤1 打开日志配置文件，默认在“*<客户端安装目录>*/Spark/spark/conf/log4j-executor.properties”。

步骤2 将日志输出文件改名。

例如：

```
log4j.appender.sparklog.File = ${spark.yarn.app.container.log.dir}/stdout
```

改为：

```
log4j.appender.sparklog.File = ${spark.yarn.app.container.log.dir}/stdout.log
```

步骤3 保存退出。

步骤4 重新提交任务。

----结束

5.7 如何解除网络 ACL 的关联子网

操作场景

用户可根据自身网络需求，解除网络ACL与子网的关联关系。

操作步骤

步骤1 登录管理控制台。

步骤2 在服务列表中单击“网络 > 虚拟私有云”。

步骤3 在左侧导航栏单击“访问控制 > 网络ACL”。

步骤4 在右侧在“网络ACL”列表区域，选择网络ACL的名称列，单击您需要修改的网络ACL名称进入详情页面。

步骤5 在详情页面，单击“关联子网”页签。

步骤6 在“关联子网”页签详情区域，选择对应子网的“操作”列，单击“取消关联”。

步骤7 单击“确认”。

----结束

5.8 修改节点主机名后导致 MRS 集群状态异常

问题现象

修改节点主机名后，MRS集群状态异常。

原因分析

修改节点hostname导致兼容性问题和故障。

处理步骤

步骤1 以**root**用户登录集群的任意节点。

步骤2 在集群节点中执行**cat /etc/hosts**命令，查看各个节点的hostname值，根据此值来配置newhostname变量值 。

步骤3 在hostname被修改的节点上执行**sudo hostnamectl set-hostname ${newhostname}**命令，恢复正确的hostname。

说明

${newhostname}：表示新的hostname取值。

步骤4 修改完成后，重新登录修改的节点，查看修改后的hostname是否生效。

----结束

5.9 进程被终止如何定位原因

问题背景与现象

在某环境出现DataNode异常重启，且确认此时未从页面做重启DataNode的操作，需要定位是什么进程终止了DataNode服务端进程。

原因分析

常见的进程被异常终止有2种原因：

- **Java进程OOM被终止**

  一般Java进程都会配置OOM Killer，当检测到OOM会自动终止进程，OOM日志通常被打印到out日志中，此时可以看运行日志（如DataNode的日志路径

为 /var/log/Bigdata/hdfs/dn/hadoop-omm-datanode-*主机名*.log ），看是否有OutOfMemory内存溢出的打印。

- **被其他进程终止，或者人为终止。**

  排查DataNode运行日志（ /var/log/Bigdata/hdfs/dn/hadoop-omm-datanode-*主机名*.log ），是先收到“RECEIVED SIGNAL 15”再健康检查失败。

  即如下示例中DataNode先于 11:04:48被终止，然后过2分钟，于11:06:52启动。

```
2018-12-06 11:04：48,433 | ERROR | SIGTERM handler | RECEIVED SIGNAL 15: SIGTERM | LogAdapter.java:69
2018-12-06 11:04:48,436 | INFO  | Thread-1 | SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down DataNode at 192-168-235-85/192.168.235.85
************************************************************/ | LogAdapter.java:45
2018-12-06 11:06:52,744 | INFO  | main | STARTUP_MSG:
```

  以上日志说明，DataNode先被其他进程关闭，然后健康检查失败，2分钟后，被NodeAgent启动DataNode进程。

处理步骤

打开操作系统审计日志，给审计日志增加记录**kill**命令的规则，即可定位是何进程发送的命令。

操作影响

- 打印审计日志，会消耗一定操作系统性能，经过分析仅影响不到1%。
- 打印审计日志，会占用一定磁盘空间。该日志打印量不大，MB级别，且默认配置有老化机制和检测磁盘剩余空间机制，不会占满磁盘。

定位方法

在DataNode进程可能发生重启的所有节点，分别执行以下操作。

步骤1 以**root**用户登录节点，执行**service auditd status**命令，确认该服务状态。

```
Checking for service auditd  running
```

如果该服务未启动，执行**service auditd restart**命令重启该服务（无影响，耗时不到1秒）。

```
Shutting down auditd done
Starting auditd done
```

步骤2 审计日志临时增加**kill**命令审计规则。

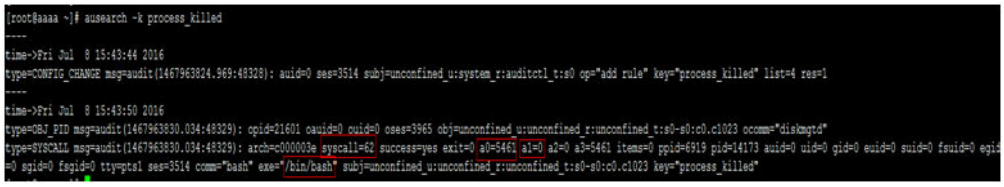
增加规则：

auditctl -a exit,always -F arch=b64 -S kill -S tkill -S tgkill -F a1!=0 -k process_killed

查看规则：

auditctl -l

步骤3 当进程有异常被终止后，使用**ausearch -k process_killed**命令，可以查询终止历史。



说明

a0是被终止进程的PID（16进制），a1是**kill**命令的信号量。

----结束

验证方法

步骤1 从MRS页面重启该节点一个实例，如DataNode。

步骤2 执行**ausearch -k process_killed**命令，确认是否有日志打印。

例如以下命令ausearch -k process_killed |grep “.sh”，可以看到是hdfs-daemon-ada* 脚本，关闭的DataNode进程。

```
[root@8-5-148-6 Bigdata]# ausearch -k process_killed | grep ".sh"
type=SYSCALL msg=audit(1481027370.223:22639542): arch=c000003e syscall=62 success=yes exit=0 a0=78dc a1=f a2=0 a3=78dc items=0 ppid=28873 pid=28880 auid=2000 uid=2000 gid=10 euid=2000 suid=2000 fsuid=2000 egid=10 sgid=10 fsgid=10 tty=(n
one) ses=19 comm="hdfs-daemon-ada" exe="/bin/bash" subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 key="process_killed"
type=SYSCALL msg=audit(1481027370.223:22639541): arch=c000003e syscall=62 success=yes exit=0 a0=78dc a1=0 a2=0 a3=7fffa7da5690 items=0 ppid=28873 pid=28880 auid=2000 uid=2000 gid=10 euid=2000 suid=2000 fsuid=2000 egid=10 sgid=10 fsgid=1
0 tty=(none) ses=19 comm="hdfs-daemon-ada" exe="/bin/bash" subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 key="process_killed"
type=SYSCALL msg=audit(1481027375.225:22639998): arch=c000003e syscall=62 success=no exit=-3 a0=78dc a1=0 a2=0 a3=78dc items=0 ppid=28873 pid=28880 auid=2000 uid=2000 gid=10 euid=2000 suid=2000 fsuid=2000 egid=10 sgid=10 fsgid=10 tty=(n
one) ses=19 comm="hdfs-daemon-ada" exe="/bin/bash" subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 key="process_killed"
[root@8-5-148-6 Bigdata]#
```

----结束

停止审计kill命令方法

步骤1 执行**service auditd restart**命令，即会清理临时增加的**kill**命令审计日志。

步骤2 执行**auditctl -l**命令，如果没有相关信息，即说明已清理该规则。

----结束

5.10 MRS 配置跨集群互信失败

问题背景与现象

MRS 1.8.2之前的版本和MRS 1.8.2及之后的版本的两个集群之间跨集群互信不通。

原因分析

配置跨集群互信后，互信的两个集群均会增加用户“krbtgt/*本集群域名@外部集群域名*”、“krbtgt/*外部集群域名@本集群域名*”，由于两个版本集群的用户默认密码不一致导致跨集群互信配置失败。

处理步骤

- **若未配置互信场景：**
  a. 配置互信前，登录MRS 1.8.2及之后版本的集群上的Master节点。
  b. 修改所有Master节点上的“/opt/Bigdata/om-0.0.1/sbin/addRealm.sh”脚本中的add_cross_realm_princ方法中的**local cross_realm_account_pwd="${DEFAULT_CROSS_REALM_PWD}"**为**local cross_realm_account_pwd="${DEFAULT _PWD}"**（位于脚本的1001行）。

     说明

     为MRS 1.8.2及之后版本的集群上的所有Master节点执行步骤**a**~**b**。
  c. 再参考**配置跨集群互信**配置。
  d. 检查互信是否成功。

- 成功，则配置结束。
- 失败，请刷新客户端配置后再检查互信是否成功，若依然不成功请联系运维人员处理。

- **若已经配置互信场景：**

a. 登录MRS 1.8.2及之后版本的集群上的Master节点。

b. 执行**/home/omm/kerberos/bin/kadmin -p kadmin/admin**并输入密码Kerberos客户端。

c. 执行listprincs并按回车，查询用户信息。

```
Last login: Thu Nov 14 12:53:41 2019 from 10.190.111.196

[root@node-master1EnmC ~]# /home/omm/kerberos/bin/kadmin -p kadmin/admin
Authenticating as principal kadmin/admin with password.
Password for kadmin/admin@8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM:
kadmin:  listprincs
K/M@8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM
krbtgt/8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM@8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM
kadmin/admin@8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM
kadmin/changepw@8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM
kadmin/history@8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM
backup/manager@8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM
admin@8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM
oms/manager@8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM
executor@8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM
knox@8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM
hdfs@8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM
hdfs/hadoop.8af4b8e5_a6fe_4b60_9bdd_efd84c8e024c.com@8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM
zookeeper/hadoop.8af4b8e5_a6fe_4b60_9bdd_efd84c8e024c.com@8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM
zkcli/hadoop.8af4b8e5_a6fe_4b60_9bdd_efd84c8e024c.com@8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM
mapred@8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM
mapred/hadoop.8af4b8e5_a6fe_4b60_9bdd_efd84c8e024c.com@8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM
mr_zk/hadoop.8af4b8e5_a6fe_4b60_9bdd_efd84c8e024c.com@8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM
check_ker_M@8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM
HTTP/node-ana-coreoeyi@8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM
HTTP/node-master1enmc@8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM
spark_zk/hadoop.8af4b8e5_a6fe_4b60_9bdd_efd84c8e024c.com@8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM
krbtgt/8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM@2A30110C_BBFA_468C_A802_4808C6B10B2A.COM
krbtgt/2A30110C_BBFA_468C_A802_4808C6B10B2A.COM@8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM
zlu_jj@8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM
test_jj@8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM
krbtgt/8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM@1353EEF8_DBFB_4ECA_8669_BC0B7700A8AB.COM
krbtgt/1353EEF8_DBFB_4ECA_8669_BC0B7700A8AB.COM@8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM
kadmin:
```

d. 执行**delprinc**命令删除用户“krbtgt/*本集群域名@外部集群域名*”、“krbtgt/*外部集群域名@本集群域名*”。

```
kadmin:
kadmin:  delprinc krbtgt/8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM@1353EEF8_DBFB_4ECA_8669_BC0B7700A8AB.COM
Are you sure you want to delete the principal "krbtgt/8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM@1353EEF8_DBFB_4ECA_8669_BC0B7700A8AB.COM"? (yes/no): yes
Principal "krbtgt/8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM@1353EEF8_DBFB_4ECA_8669_BC0B7700A8AB.COM" deleted.
Make sure that you have removed this principal from all ACLs before reusing.
kadmin:  delprinc krbtgt/1353EEF8_DBFB_4ECA_8669_BC0B7700A8AB.COM@8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM
Are you sure you want to delete the principal "krbtgt/1353EEF8_DBFB_4ECA_8669_BC0B7700A8AB.COM@8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM"? (yes/no): yes
Principal "krbtgt/1353EEF8_DBFB_4ECA_8669_BC0B7700A8AB.COM@8AF4B8E5_A6FE_4B60_9BDD_EFD84C8E024C.COM" deleted.
Make sure that you have removed this principal from all ACLs before reusing.
kadmin:
```

e. 使用**quit**命令退出Kerberos客户端。

f. 修改Master节点上的/opt/Bigdata/om-0.0.1/sbin/addRealm.sh脚本中的add_cross_realm_princ方法中的**local cross_realm_account_pwd="${DEFAULT_CROSS_REALM_PWD}"**为**local cross_realm_account_pwd="${DEFAULT _PWD}"**（位于脚本的1001行）。

g. 登录MRS Manager页面，选择“服务管理”。

h. 选择“更多 > 同步配置”。

i. 在弹出窗口勾选“重启配置过期的服务或实例。”，并单击“确定”，重启配置过期的服务。

同步配置时会调用addRealm.sh脚本添加krbtgt用户。

说明

为MRS 1.8.2及之后版本的集群上的所有Master节点执行步骤a~i。

j. 检查互信是否成功，若依然不成功，请联系运维人员处理。

5.11 MRS 集群节点使用 pip3 安装 Python 时提示网络不可达

问题现象

执行**pip3 install**安装Python时报错网络不可达。

具体如下图所示：

```
[root@node-master1Diqn base]# pip3 install openpyxl
Collecting openpyxl
  Retrying (Retry(total=4, connect=None, read=None, redirect=None)) after connection broken by 'NewConnection
Error('<pip._vendor.requests.packages.urllib3.connection.VerifiedHTTPSConnection object at 0x7f5ed31044e0>: F
ailed to establish a new connection: [Errno 101] Network is unreachable',)': /simple/openpyxl/
```

原因分析

用户未给Master节点绑定弹性公网IP，造成报错的发生。

处理步骤

步骤1 登录MRS服务管理控制台。

步骤2 选择“现有集群”，选中当前安装出问题的集群并单击集群名称，进入集群基本信息页面。

步骤3 在“节点管理”页签单击Master节点组中某一Master节点名称，登录到弹性云服务器管理控制台。

步骤4 选择“弹性公网IP”页签，单击“绑定弹性公网IP”为弹性云服务器绑定一个弹性公网IP。

步骤5 登录Master节点执行**pip3 install**命令安装Python。

----结束

5.12 开源 confluent-kafka-go 连接 MRS 安全集群

用户问题

开源confluent-kafka-go如何连接MRS的安全集群?

问题现象

开源confluent-kafka-go连接MRS的安全集群失败。

原因分析

confluent-kafka-go依赖的库librdkafka默认将broker所在hostname作为了server principle的一部分来使用，导致认证失败。

处理步骤

librdkafka具体修改步骤：

1. librdkafka源码地址：**https://github.com/edenhill/librdkafka**。
2. 在src/rdkafka_conf.c文件中增加sasl.kerberos.service.name配置项。
```
 "Kerberos principal name that Kafka runs as.",          .sdef = "kafka" },         { _RK_GLOBAL,
"sasl.kerberos.principal", _RK_C_STR,           _RK(sasl.principal),           "This client´s Kerberos
principal name.",           .sdef = "kafkaclient" }, +        { _RK_GLOBAL, "sasl.kerberos.domain.name",
_RK_C_STR, +          _RK(sasl.domain_name), +          "This cluster´s Kerberos domain name.",
+          .sdef = "hadoop.hadoop.com" },   #ifndef _MSC_VER          { _RK_GLOBAL,
"sasl.kerberos.kinit.cmd", _RK_C_STR,           _RK(sasl.kinit_cmd),           "Full kerberos kinit
command string, %{config.prop.name} is replaced "           "by corresponding config object value, %
{broker.name} returns the "           "broker´s hostname.", -          .sdef = "kinit -S \"%
{sasl.kerberos.service.name}/%{broker.name}\" " +          .sdef = "kinit -S \"%
{sasl.kerberos.service.name}/%{sasl.kerberos.domain.name}\" "           "-k -t \"%{sasl.kerberos.keytab}
\" %{sasl.kerberos.principal}" },          { _RK_GLOBAL, "sasl.kerberos.keytab", _RK_C_STR,
_RK(sasl.keytab),           "Path to Kerberos keytab file. Uses system default if not set."
"**NOTE**: This is not automatically used but must be added to the "           "template in
sasl.kerberos.kinit.cmd as "
```
3. 在src/rdkafka_conf.h文件中增加domain_name字段。
```
--- src\rdkafka_conf.h        2017-10-17 11:20:56.000000000 +0800 +++ src\rdkafka_conf.h
2017-10-25 16:26:34.000000000 +0800 @@ -118,12 +118,13 @@          struct {                  const
struct rd_kafka_sasl_provider *provider;                  char *principal;                  char
*mechanisms;                  char *service_name; +                              char *domain_name;
char *kinit_cmd;                  char *keytab;                  int   relogin_min_time;                  char
*username;                  char *password; #if WITH_SASL_SCRAM
```
4. 在src/rdkafka_sasl_cyrus.c文件中将hostname替换成domainName。
```
--- src\rdkafka_sasl.c        2017-10-17 11:20:56.000000000 +0800 +++ src\rdkafka_sasl.c
2017-10-25 16:09:38.000000000 +0800 @@ -192,13 +192,14 @@                              rk-
>rk_conf.sasl.mechanisms,                              rk->rk_conf.api_version_request ?
"" :                              ": try api.version.request=true");                  return -1;          } -
rd_strdupa(&hostname, rktrans->rktrans_rkb->rkb_nodename); +        //rd_strdupa(&hostname,
rktrans->rktrans_rkb->rkb_nodename); +                  rd_strdupa(&hostname, rk-
>rk_conf.sasl.domain_name);          if ((t = strchr(hostname, ´:´)))                  *t = ´\0´;  /* remove
":port" */
```
5. 重新编译librdkafka（请确保已安装libsasl2-dev），具体步骤参考**https://github.com/edenhill/librdkafka/tree/v0.11.1**。

   ./configure make make install
6. 使用客户端时增加如下配置项。
```
"security.protocol": "SASL_PLAINTEXT",
"sasl.kerberos.service.name": "kafka",
"sasl.kerberos.keytab": "/opt/nemon/user.keytab",
"sasl.kerberos.principal": "nemon@HADOOP.COM",
"sasl.kerberos.domain.name": "hadoop.hadoop.com",
```

说明

MRS 2.1.x及之前版本：

- sasl.kerberos.keytab：可通过在MRS Manager界面选择“系统设置 > 用户管理”，在对应用户所在行的“操作”列选择“更多 >下载认证凭据”，保存后解压得到用户的user.keytab文件。
- sasl.kerberos.principal：请填写实际用户名。
- sasl.kerberos.domain.name：domain的命名规则为hadoop. toLowerCase(realm)，假设集群的域名（default_realm）为example.com时，domain的值为hadoop.example.com。可通过MRS Manager界面选择“服务管理> KrbServer > > 服务配置 > 全部配置”，搜索并查看default_realm的值。

MRS 3.x及后续版本：

- sasl.kerberos.keytab：可通过在FusionInsight Manager界面选择“系统 > 权限 > 用户”，在对应用户所在行的“操作”列选择“更多 >下载认证凭据”，保存后解压得到用户的user.keytab文件。
- sasl.kerberos.principal：请填写实际用户名。
- sasl.kerberos.domain.name：domain的命名规则为hadoop. toLowerCase(realm)，假设集群的域名（default_realm）为example.com时，domain的值为hadoop.example.com。可通过FusionInsight Manager界面选择“集群 > 服务 > KrbServer > > 配置 > 全部配置”，搜索并查看default_realm的值。

5.13 MRS 集群周期备份任务执行失败

问题现象

MRS 1.7.2版本集群周期备份故障，在MRS Manager页面的“告警管理中”出现“ALM-12034 周期备份任务失败”告警。

原因分析

集群在周期性的备份中，由于备份产生了软连接，将/srv/BigData/LocalBackup连接到了/opt/Bigdata/LocalBackup，会使用到根目录磁盘空间，因为根目录磁盘空间不足，导致无法写入备份文件，导致集群周期备份失败。

处理步骤

1. 分别登录主、备Master节点。
2. 执行**cd /srv/BigData/**命令进入到备份文件所在目录。
3. 执行**unlink LocalBackup**命令删除LocalBackup软连接。
4. 执行**mkdir -p LocalBackup**命令创建LocalBackup目录。
5. 执行**chown -R omm:wheel LocalBackup**命令修改文件所属用户、群组。
6. 执行**chmod 700 LocalBackup**命令修改文件读写权限。
7. 登录MRS Manager页面重新执行周期备份。

5.14 MRS 集群客户端无法下载

问题现象

通过Master节点给MRS集群外的一台远端节点下载MRS集群客户端进行使用，但是一直提示网络或者参数有问题。

原因分析

- 可能是两台主机处于不同VPC网络中。
- 密码填写错误。
- 远端主机开启了防火墙。

处理步骤

- 两台主机处于不同VPC网络中
  放开远端主机的22端口。
- 密码填写错误
  请检查密码是否正确，密码中不能有特殊符号。
- 远端主机开启防火墙
  先将这个MRS集群客户端下载到服务器端主机，然后通过Linux提供的**scp**命令复制到远端主机。

5.15 开启 Kerberos 认证的 MRS 集群提交 Flink 作业报错

用户问题

用户在开启Kerberos认证的MRS集群中提交Flink作业报错。

问题现象

用户提交Flink样例程序**./flink run /opt/client/Flink/flink/examples/streaming/WordCount.jar**时报错：unable to establish the security context。

原因分析

1. MRS集群开启了Kerberos认证但是无法提交作业，所以首先检查权限配置问题，检查发现未正确配置“/opt/client/Flink/flink/conf/flink-conf.yaml”中的参数。

图 5-9 flink-conf.yaml 配置

```
query.proxy.query-threads: 0
security.kerberos.login.keytab: /data01/fiClient/mrs_210_new/keytab/flinkuser/user.keytab
high-availability.zookeeper.client.retry-wait: 5000
taskmanager.runtime.hashjoin-bloom-filters: false
akka.lookup.timeout: 10 s
blob.fetch.backlog: 1000
fs.obs.socket.send.buffer: 65536
blob.service.ssl.enabled: true
blob.fetch.num-concurrent: 50
query.server.query-threads: 0
akka.watch.heartbeat.interval: 10 s
security.kerberos.login.use-ticket-cache: false
fs.obs.max.total.tasks: 20
fs.obs.fast.upload: true
yarn.application-attempts: 3
akka.ask.timeout: 10 s
taskmanager.debug.memory.log-interval: 0
fs.obs.buffer.max.range: 136000
fs.obs.threads.read.core: 500
security.ssl.protocol: TLSv1.2
restart-strategy.fixed-delay.attempts: 3
akka.throughput: 15
taskmanager.network.netty.server.numThreads: 3
taskmanager.rpc.port: 32326-32390
akka.startup-timeout: 60 s
task.checkpoint.alignment.max-size: -1
taskmanager.runtime.max-fan: 128
web.ssl.enabled: false
containerized.heap-cutoff-ratio: 0.25
taskmanager.runtime.sort-spilling-threshold: 0.8
security.ssl.internal.keystore: /mrs_210_new/Flink/flink/conf//flink.keystore
fs.obs.endpoint: obs.cn-north-5.myhuaweicloud.com
```

keytab文件路径

此处配置为相对路径，若为绝对路径每个节点都要配置。

2. 修改并刷新配置后，重新提交作业出现作业可以提交但报“log4j:ERROR setFile(null,true) call failed”的错误。

图 5-10 log4j 报错

```
        at org.apache.flink.runtime.entrypoint.ClusterEntrypoint.<clinit>(ClusterEntrypoint.java:98)
log4j:ERROR setFile(null,true) call failed.
```

3. 查看log4j发现用户将“log4j.properties”文件改成了“log4g-cli.properties”（“log4j.properties”的名字是固定的不可随意修改）导致报错。

图 5-11 查看 log4j

```
[emr@kwetemr00005 flink]$ cd conf/
[emr@kwetemr00005 conf]$ ll
total 52
-rwxr-xr-x 1 emr emr  727 Apr 15 10:26 ca.cer
-rwxr-xr-x 1 emr emr 2075 Apr 15 10:26 ca.keystore
-rwxr-xr-x 1 emr emr  358 Apr 15 08:43 client.properties
-rwxr-xr-x 1 emr emr  534 Apr 15 08:43 core-site.xml
-rwxr-xr-x 1 emr emr  773 Apr 15 10:26 flink.cer
-rwxr-xr-x 1 emr emr 5564 Apr 20 10:12 flink-conf.yaml
-rwxr-xr-x 1 emr emr  965 Apr 15 10:26 flink.csr
-rwxr-xr-x 1 emr emr 1739 Apr 15 08:43 hdfs-site.xml
-rwxr-xr-x 1 emr emr 1113 Apr 15 10:47 log4j-cli.properties
-rwxr-xr-x 1 emr emr  976 Apr 15 08:43 log4j.properties
-rwxr-xr-x 1 emr emr  709 Apr 15 08:43 log4j-yarn-session.properties
drwxrwxr-x 2 emr emr   50 Apr 17 09:50 ssl
-rwxr-xr-x 1 emr emr 1953 Apr 15 08:43 yarn-site.xml
```

4. 修改后可以正常提交作业。

图 5-12 提交作业正常

```
[omm@node-master1FtpV flink]$ flink run -m yarn-cluster /opt/client/Flink/flink/examples/streaming/WordCount.jar
2020-04-17 10:23:13,988 | WARN  | [main] | Error while trying to split key and value in configuration file /opt/client/Flink/flink/conf/flink-conf.yaml:60: "security.ssl.internal.keysto
che.flink.configuration.GlobalConfiguration (GlobalConfiguration.java:179)
2020-04-17 10:23:16,844 | WARN  | [main] | The short-circuit local reads feature cannot be used because libhadoop cannot be loaded. | org.apache.hadoop.hdfs.shortcircuit.DomainSocketFac
ketFactory.java:116)
2020-04-17 10:23:16,844 | WARN  | [main] | The short-circuit local reads feature cannot be used because libhadoop cannot be loaded. | org.apache.hadoop.hdfs.shortcircuit.DomainSocketFac
ketFactory.java:116)
Starting execution of program
Executing WordCount example with default input data set.
Use --input to specify file input.
Printing result to stdout. Use --output to specify output path.
2020-04-17 10:23:28,548 | WARN  | [main] | Error while trying to split key and value in configuration file /opt/client/Flink/flink/conf/flink-conf.yaml:60: "security.ssl.internal.keysto
che.flink.configuration.GlobalConfiguration (GlobalConfiguration.java:179)
Program execution finished
Job with JobID 98cadfdf8e3d0eaa35ba42f2ce55d356 has finished.
```

处理步骤

步骤1 判断用户是在集群外还是集群内使用客户端提交作业。

1. 若在集群内使用客户端，切换到omm用户提交作业。
2. 若在集群外使用客户端，则要使用root用户提交作业。

步骤2 检查“flink-conf.yaml”文件各参数是否配置正确。

步骤3 对于开启Kerberos认证的集群配置项包括Kerberos的keytab、principal等。

- 从KDC服务器上下载用户keytab，并将keytab放到Flink客户端所在主机的某个文件夹下(例如/home/flinkuser/keytab)。
- 在“${FLINK_HOME}/conf/flink-conf.yaml”上配置：
  a. keytab路径（注意配置参数前面有空格）：
  ```
  security.kerberos.login.keytab:  /home/flinkuser/keytab/uer.keytab
  ```
  b. principal名(即开发用户名)：
  ```
  security.kerberos.login.principal:flinkuser
  ```

步骤4 重新正确提交作业**./flink run /opt/client/Flink/flink/examples/streaming/WordCount.jar**，验证是否可以提交作业。

- 若可以提交作业则说明权限认证没有问题，就可以去检查其他错误，本例中是修改了log4j.properties的名称，还原后可以正常提交作业。
- 若提交作业失败，请提交工单进行处理。

----结束

参考信息

Flink使用请参考**从零开始使用Flink**。

5.16 通过 Hive beeline 命令行执行插入命令的时候报错

问题现象

在MRS Hive的beeline中执行**insert into**插入语句时系统报以下错误：

```
Mapping run in Tez on Hive transactional table fails when data volume is high with error:
"org.apache.hadoop.hive.ql.lockmgr.LockException Reason: Transaction... already aborted, Hive SQL state
[42000]."
```

原因分析

对于Join操作，由于集群配置不理想和Tez资源设置不合理导致该问题。

处理步骤

可以在beeline上设置配置参数进行解决。

步骤1 设置以下属性以优化性能（建议在集群级别进行更改）

- 设置hive.auto.convert.sortmerge.join = true
- 设置hive.optimize.bucketmapjoin = true

- 设置hive.optimize.bucketmapjoin.sortedmerge = true

步骤2 更改以下内容以调整Tez的资源。

- 设置hive.tez.container.size = {与YARN容器相同的大小}
- 将hive.tez.container.size设置为与YARN容器大小“yarn.scheduler.minimum-allocation-mb”相同或更小的值（例如设置为二分之一或四分之一的值），但不要超过“yarn.scheduler.maximum-allocation-mb”参数值。

----结束

5.17 MRS 集群节点如何进行操作系统漏洞升级

用户问题

EulerOS系统底层存在漏洞时，MRS集群如何进行操作系统漏洞升级？

问题现象

在使用安全软件测试集群时，发现有EulerOS系统底层存在漏洞。

原因分析

MRS集群内服务部署在EulerOS系统中，因此需要进行漏洞升级。

处理步骤

说明

修复漏洞前请确认是否开启了主机安全服务（Host Security Service，简称HSS），如果已开启，需要先暂时关闭HSS服务对MRS集群的监测，漏洞修复完成后重新开启HSS服务。

步骤1 登录MRS服务控制台。

步骤2 单击集群名称进入集群详情页面，并选择“节点管理”。

步骤3 在Core节点组中勾选任意一个Core节点，单击“节点操作 > 停止所有角色”。

步骤4 登录Core节点后台，参考**如何使用自动化工具配置华为云镜像源(x86_64和ARM)？** 配置yum源。

步骤5 使用**uname -r**或**rpm -qa |grep kernel**命令，查询并记录当前节点内核版本。

步骤6 执行**yum update -y --skip-broken --setopt=protected_multilib=false**命令更新补丁。

步骤7 完成更新后查询内核版本，并执行**rpm -e** *旧内核版本*命令删除旧内核版本。

步骤8 在集群详情页，选择“节点管理”。

步骤9 在Core节点组中单击已更新补丁的Core名称，进入弹性云服务器管理控制台。

步骤10 在页面右上角单击“重启”，重启Core节点。

步骤11 重启完成后，在集群详情页的“节点管理”的Core节点组中勾选Core节点，单击“节点操作 > 启动所有角色”。

步骤12 重复**步骤1**~**步骤11**的操作，升级其他Core节点。

步骤13 所有Core节点升级完成后，参考**步骤1**~**步骤11**的操作先升级备Master节点，再升级主Master节点。

----结束

5.18 使用 CDM 服务迁移数据至 MRS HDFS 时任务运行失败

用户问题

使用CDM从旧的集群迁移数据至新集群的HDFS过程失败。

问题现象

使用CDM从源HDFS导入目的端HDFS，发现目的端MRS集群故障，NameNode无法启动。

查看日志发现在启动过程中存在“Java heap space”报错，需要修改NameNode的JVM参数。

图 5-13 故障日志

```
2020-08-27 11:44:18,327 | INFO  | main | 0.02999999932944774 6% max memory 486.4 MB = 149.4 KB | LightWeightGSet.java:397
2020-08-27 11:44:18,328 | INFO  | main | capacity      = 2^14 = 16384 entries | LightWeightGSet.java:402
2020-08-27 11:44:18,330 | INFO  | main | Using INode attribute provider:                    .adapter.hdfs.plugin.HWINodeAttributeProvider | FSNamesystem.java:914
2020-08-27 11:44:18,337 | INFO  | main | Lock on /srv/BigData/namenode/in_use.lock acquired by nodename 6565@node-master2jGRz | Storage.java:905
2020-08-27 11:44:18,637 | INFO  | main | Planning to load image: FSImageFile(file=/srv/BigData/namenode/current/fsimage_0000000000010002506, cpktTxId=0000000000010002506) | FSImage.java:808
2020-08-27 11:44:19,173 | INFO  | main | Enable the erasure coding policy RS-6-3-1024k | ErasureCodingPolicyManager.java:410
2020-08-27 11:44:19,175 | INFO  | pool-12-thread-1 | Loading 1048576 INodes. | FSImageFormatPBINode.java:336
2020-08-27 11:44:19,175 | INFO  | pool-12-thread-2 | Loading 946367 INodes. | FSImageFormatPBINode.java:336
2020-08-27 11:45:33,594 | WARN  | qtp1966124444-31-acceptor-0@62fa7d99-ServerConnector@20b2475a{HTTP/1.1,[http/1.1]}{node-master2jGRz:9870} |  | AbstractConnector.java:544
java.lang.OutOfMemoryError: Java heap space
2020-08-27 11:45:33,601 | INFO  | main | Loaded FSImage in 74 seconds. | FSImageFormatProtobuf.java:205
2020-08-27 11:45:33,601 | INFO  | main | Loaded image for txid 10002506 from /srv/BigData/namenode/current/fsimage_0000000000010002506 | FSImage.java:985
2020-08-27 11:45:36,045 | INFO  | main | Reading org.apache.hadoop.hdfs.server.namenode.RedundantEditLogInputStream@3a94964 expecting start txid #10002507 | FSImage.java:920
```

原因分析

使用CDM迁移数据的过程中，HDFS的数据量过大，导致在合并元数据时发生堆栈异常。

处理步骤

步骤1 进入HDFS服务配置页面：

- MRS 3.x之前版本，单击集群名称，登录集群详情页面，选择“组件管理 > HDFS > 服务配置”，单击“基础配置”下拉菜单，选择“全部配置”。

  说明

  如果集群详情页面没有“组件管理”页签，请先完成IAM用户同步（在集群详情页的“概览”页签，单击“IAM用户同步”右侧的“同步”进行IAM用户同步）。

- MRS 3.x及后续版本，登录FusionInsight Manager，然后选择“集群 > *待操作的集群名称* > 服务 > HDFS > 配置 > 全部配置”。

步骤2 搜索并修改“HDFS > NameNode” 中的“GC_OPTS”参数，将其中的“-Xms512M -Xmx512M”两个参数的值根据实际情况调整为较大的值。

步骤3 保存配置并重启受影响的服务或实例。

----结束

5.19 MRS 集群频繁产生节点间心跳中断告警

问题现象

MRS集群频繁上报Manager主备节点间心跳中断、DBService主备节点间心跳中断、节点故障等告警，偶尔会造成Hive不可用，影响上层业务。

原因分析

1. 在出现告警时间点发现虚拟机发生了重启，告警发生的原因是因虚拟机重启导致的。

```
[omm@node-master1yqIY nodeagent]$ last
omm      pts/0        100.125.0.70      Thu Sep 24 10:33   still logged in
omm      pts/1        100.125.0.70      Thu Sep 24 09:26 - 09:47  (00:20)
omm      pts/0        100.125.0.70      Thu Sep 24 09:22 - 10:21  (00:59)
omm      pts/1        100.125.0.70      Wed Sep 23 17:32 - 17:37  (00:05)
root     pts/0        10.203.216.102    Wed Sep 23 17:13 - 18:35  (01:21)
omm      pts/0        100.125.0.70      Wed Sep 23 16:55 - 16:56  (00:00)
omm      pts/0        100.125.0.70      Wed Sep 23 16:20 - 16:25  (00:05)
reboot   system boot  4.19.36-vhulk190 Wed Sep 23 16:06   still running
root     pts/1        10.203.216.102    Tue Sep 22 19:13 - 19:48  (00:34)
omm      pts/0        100.125.0.70      Tue Sep 22 19:08 - 20:03  (00:54)
root     pts/0        10.203.216.102    Tue Sep 22 17:03 - 17:52  (00:48)
omm      pts/1        100.125.0.70      Tue Sep 22 15:55 - 16:00  (00:05)
```

```
[omm@node-master2WbYp ~]$ last
omm      pts/0        10.80.0.56        Thu Sep 24 11:00   still logged in
omm      pts/0        10.80.0.56        Thu Sep 24 09:24 - 10:21  (00:56)
omm      pts/0        10.80.0.56        Wed Sep 23 17:32 - 17:37  (00:05)
omm      pts/0        10.80.0.56        Tue Sep 22 19:15 - 19:15  (00:00)
omm      pts/0        10.80.0.56        Tue Sep 22 15:57 - 16:21  (00:23)
omm      pts/0        10.80.0.56        Tue Sep 22 15:23 - 15:35  (00:12)
omm      pts/0        10.80.0.56        Tue Sep 22 15:07 - 15:12  (00:05)
omm      pts/0        10.80.0.56        Tue Sep 22 14:21 - 14:26  (00:05)
omm      pts/0        10.80.0.56        Mon Sep 21 10:57 - 11:06  (00:09)
omm      pts/0        10.80.0.56        Mon Sep 21 10:42 - 10:56  (00:14)
omm      pts/0        10.80.0.56        Thu Sep 17 16:05 - 16:15  (00:10)
omm      pts/0        10.80.0.56        Wed Sep 16 20:52 - 20:58  (00:06)
reboot   system boot  4.19.36-vhulk190 Wed Sep 16 18:05   still running
omm      pts/0        10.80.0.56        Wed Sep 16 15:43 - 16:10  (00:26)
omm      pts/0        10.80.0.56        Wed Sep 16 14:35 - 14:53  (00:17)
omm      pts/0        10.80.0.56        Wed Sep 16 14:33 - 14:33  (00:00)
omm      pts/0        10.80.0.56        Wed Sep 16 14:11 - 14:29  (00:17)
omm      pts/0        10.80.0.56        Wed Sep 16 14:02 - 14:09  (00:06)
omm      pts/0        10.80.0.56        Wed Sep 16 11:56 - 12:04  (00:08)
omm      pts/0        10.80.0.56        Wed Sep 16 11:26 - 11:31  (00:04)
omm      pts/0        10.80.0.56        Wed Sep 16 11:09 - 11:24  (00:15)
root     pts/0        10.203.230.193    Mon Sep 14 15:54 - 16:30  (00:35)
root     pts/0        10.203.172.29     Fri Sep 11 17:15 - 17:45  (00:30)
root     pts/0        10.203.172.29     Fri Sep 11 16:53 - 17:12  (00:19)
root     tty1                           Fri Sep 11 16:23 - 17:25  (01:01)
reboot   system boot  4.19.36-vhulk190 Fri Sep 11 10:07   still running
reboot   system boot  4.19.36-vhulk190 Thu Aug 27 16:41   still running
root     tty1                           Thu Aug 20 09:46 - 10:17  (00:30)
reboot   system boot  4.19.36-vhulk190 Wed Aug 19 17:48   still running
reboot   system boot  4.19.36-vhulk190 Wed Aug 19 17:46   still running
```

2. 经OS定位虚拟机发生重启的原因是节点没有可用的内存，系统发生内存溢出触发了oom-killer，当进程处于被调用的状态会使进程处于disk sleep状态，最终导致虚拟机发生重启。

```
mem info:
[344766.903734] MemTotal:       32397404 kB
MemFree:          160404 kB
MemAvailable:      31668 kB
Buffers:            2172 kB
Cached:          2768904 kB
SwapCached:            0 kB
Active:         30328872 kB
Inactive:        1035844 kB
Active(anon):   30320852 kB
Inactive(anon):  1004376 kB
Active(file):       8020 kB
Inactive(file):    31468 kB
Unevictable:           0 kB
Mlocked:               0 kB
[344766.903738] SwapTotal:             0 kB
SwapFree:              0 kB
```

总内存

用户态使用

```
[344766.904470] 20444     1 212684K    104K    S (sleeping) /sbin/agetty -o -p -- \u --noclear tty1 linux
[344766.904474] 15011  9241 845712K   1948K    S (sleeping) gaussdb: wal sender process REPLICATION node-masterlyqIY(30753) s
[344766.904477] 20384  9241 866276K 326020K  D (disk sleep) gaussdb: OMM OMM localhost(35218) PARSE
[344766.904480] 20389  9241 867524K 326732K  D (disk sleep) gaussdb: OMM OMM localhost(35222) PARSE
[344766.904484] 29384     1 253256K   1852K    S (sleeping) /usr/sbin/sssd -D
[344766.904487] 29453 29384 253144K   2620K     R (running) /usr/libexec/sssd/sssd_be --domain implicit_files --uid 0 --gid 0 --logger=journald
[344766.904491] 29454 29384 258292K   4004K    S (sleeping) /usr/libexec/sssd/sssd_be --domain default --uid 0 --gid 0 --logger=journald
[344766.904494] 29512 29384 283272K   2112K    S (sleeping) /usr/libexec/sssd/sssd_nss --uid 0 --gid 0 --logger=journald
[344766.904498] 29513 29384 243880K   1680K  D (disk sleep) /usr/libexec/sssd/sssd_pam --uid 0 --gid 0 --logger=journald
[344766.904501] 29527     1 5500276K 323624K    S (sleeping) /opt/Bigdata/jdk1.8.0_212//bin/java -cp
/opt/Bigdata/MRS_2.1.0/1_21_JDBCServer/etc/:/opt/Bigdata/security/:/opt/Bigdata/MRS_2.1.0/install/FusionInsight-Spark-2.3.2/spark/sbin/../jars/* -Dlog4j
-Djava.security.auth.login.config=/o
[344766.904505]  7855  9241 846668K  23736K    S (sleeping) gaussdb: OMM OMM localhost(46200) idle
[344766.904509] 25941  9241 859332K 323464K  D (disk sleep) gaussdb: OMM OMM localhost(48556) idle
[344766.904512] 25951  9241 857892K 319088K  D (disk sleep) gaussdb: OMM OMM localhost(48558) PARSE
[344766.904516] 26004  9241 867192K 324348K  D (disk sleep) gaussdb: OMM OMM localhost(48562) idle
[344766.904519] 26108  9241 857940K 323228K  D (disk sleep) gaussdb: OMM OMM localhost(48564) PARSE
[344766.904523] 26156  9241 858120K 324052K  D (disk sleep) gaussdb: OMM OMM localhost(48570) PARSE
[344766.904527] 26165  9241 866212K 322884K  D (disk sleep) gaussdb: OMM OMM localhost(48576) PARSE
[344766.904531] 26172  9241 858180K 322896K  D (disk sleep) gaussdb: OMM OMM localhost(48578) PARSE
[344766.904534] 26212  9241 857932K 323148K  D (disk sleep) gaussdb: OMM OMM localhost(48580) PARSE
[344766.904538] 26309  9241 859160K 321728K  D (disk sleep) gaussdb: OMM OMM localhost(48582) PARSE
[344766.904541] 26362  9241 866236K 322212K  D (disk sleep) gaussdb: OMM OMM localhost(48584) PARSE
[344766.904545] 26389  9241 866408K 323184K  D (disk sleep) gaussdb: OMM OMM localhost(48588) PARSE
[344766.904548] 26399  9241 857844K 321616K  D (disk sleep) gaussdb: OMM OMM localhost(48592) PARSE
[344766.904551] 26404  9241 858044K 322592K  D (disk sleep) gaussdb: OMM OMM localhost(48596) PARSE
[344766.904555] 26415  9241 857756K 322528K  D (disk sleep) gaussdb: OMM OMM localhost(48600) PARSE
[344766.904558] 26450  9241 858768K 323668K  D (disk sleep) gaussdb: OMM OMM localhost(48606) PARSE
[344766.904562] 26482  9241 858072K 323340K  D (disk sleep) gaussdb: OMM OMM localhost(48608) PARSE
[344766.904565] 26608  9241 858024K 322904K  D (disk sleep) gaussdb: OMM OMM localhost(48610) PARSE
[344766.904568] 27449  9241 866276K 323472K  D (disk sleep) gaussdb: OMM OMM localhost(48632) PARSE
[344766.904573] 30030     1 387064K  17424K     R (running) /opt/Bigdata/MRS_2.1.0/install/FusionInsight-Hue-3.11.0/hue/build/env/bin/python2.7
/opt/Bigdata/MRS_2.1.0/install/FusionInsight-Hue-3.11.0/hue/build/env/bin/supervisor -p /opt/Bigdata/MRS_2.1.0/install/FusionInsight-Hue-3.11.0/hue/cnf/
```

```
[344766.904726]   874  4953    1484K      8K  D (disk sleep) /bin/sh /opt/Bigdata/nodeagent/bin/scriptLauncher.sh /opt/Bigdata/MRS_2.1.0/install/dbservice/sb
[344766.904729]   875 26044    1488K     12K  D (disk sleep) /bin/sh /opt/Bigdata/nodeagent/bin/scriptLauncher.sh
/opt/Bigdata/MRS_2.1.0/install/FusionInsight-Hadoop-3.1.1/hadoop/sbin/yarn-resourcemanager-check.sh
[344766.904732]   876 10755 7522420K  670728K  D (disk sleep) /opt/Bigdata/jdk1.8.0_212//bin/java -Dprocess.name=nodeagent
-Dbeetle.application.home.path=/opt/Bigdata/security/config -Dsun.rmi.transport.tcp.responseTimeout=60000 -Djava.library.path=/opt/Bigdata/nodeagent/lib
-XX:ErrorFile=/var/log/Bigdata/nodeagent
[344766.904735]   878 17629 8616200K 1124612K  D (disk sleep) /opt/Bigdata/jdk1.8.0_212//bin/java -Djava.security.egd=file:/dev/./urandom -Dprocess.name=contr
-Dstack.conf.dir= -Dcontroller.home=/opt/Bigdata/om-0.0.1 -Dbeetle.application.home.path=/opt/Bigdata/om-0.0.1/etc/om -Dorg.terracotta.quartz.skipUpdate
[344766.904738]   879  7057    1484K      8K  D (disk sleep) /bin/sh /opt/Bigdata/nodeagent/bin/scriptLauncher.sh
/opt/Bigdata/MRS_2.1.0/install/FusionInsight-Flume-1.6.0/flume/bin/flume-check-service.sh
[344766.904741]   880  2535    1488K     12K  D (disk sleep) /bin/sh /opt/Bigdata/nodeagent/bin/scriptLauncher.sh /usr/bin/head -1  /opt/Bigdata/tmp//hadoop-
[344766.904744]   881  9760 7522420K  670728K  D (disk sleep) /opt/Bigdata/jdk1.8.0_212//bin/java -Dprocess.name=nodeagent
-Dbeetle.application.home.path=/opt/Bigdata/security/config -Dsun.rmi.transport.tcp.responseTimeout=60000 -Djava.library.path=/opt/Bigdata/nodeagent/lib
-XX:ErrorFile=/var/log/Bigdata/nodeagent
[344766.904746]   882  3895 7522420K  670728K  D (disk sleep) /opt/Bigdata/jdk1.8.0_212//bin/java -Dprocess.name=nodeagent
-Dbeetle.application.home.path=/opt/Bigdata/security/config -Dsun.rmi.transport.tcp.responseTimeout=60000 -Djava.library.path=/opt/Bigdata/nodeagent/lib
-XX:ErrorFile=/var/log/Bigdata/nodeagent
[344766.904748]   883  3665 7522420K  670728K  D (disk sleep) /opt/Bigdata/jdk1.8.0_212//bin/java -Dprocess.name=nodeagent
-Dbeetle.application.home.path=/opt/Bigdata/security/config -Dsun.rmi.transport.tcp.responseTimeout=60000 -Djava.library.path=/opt/Bigdata/nodeagent/lib
-XX:ErrorFile=/var/log/Bigdata/nodeagent
[344766.904751]   885  8623 7522420K  670728K  D (disk sleep) /opt/Bigdata/jdk1.8.0_212//bin/java -Dprocess.name=nodeagent
-Dbeetle.application.home.path=/opt/Bigdata/security/config -Dsun.rmi.transport.tcp.responseTimeout=60000 -Djava.library.path=/opt/Bigdata/nodeagent/lib
-XX:ErrorFile=/var/log/Bigdata/nodeagent
[344766.904753]   886  5536 7522420K  670728K  D (disk sleep) /opt/Bigdata/jdk1.8.0_212//bin/java -Dprocess.name=nodeagent
-Dbeetle.application.home.path=/opt/Bigdata/security/config -Dsun.rmi.transport.tcp.responseTimeout=60000 -Djava.library.path=/opt/Bigdata/nodeagent/lib
-XX:ErrorFile=/var/log/Bigdata/nodeagent
[344766.904754] Mem-Info:
[344766.904757] active anon:7580213 inactive anon:251094 isolated anon:0
```

3. 查看占用的内存进程，发现占用内存都是正常的业务进程。

结论：虚拟机内存不能满足服务需求。

处理步骤

- 建议扩大节点内存。
- 建议关闭不需要的服务。

5.20 PMS 进程占用内存高

用户问题

主Master节点内存使用率高如何处理?

问题现象

主Master节点内存使用率高，且用**top -c**命令查询的内存占用量高的是如下idle的进程。

```
12180 ommdba    20   0 1395492 1.180g 1.082g S   0.0  3.8  23:14.29 gaussdb: OMM OMM localhost(60598) idle
14828 ommdba    20   0 1395904 1.180g 1.081g S   0.0  3.8  23:17.08 gaussdb: OMM OMM localhost(60698) idle
15016 ommdba    20   0 1395840 1.180g 1.081g S   0.0  3.8  23:11.19 gaussdb: OMM OMM localhost(60824) idle
14943 ommdba    20   0 1395900 1.180g 1.081g S   0.0  3.8  23:14.76 gaussdb: OMM OMM localhost(60764) idle
14908 ommdba    20   0 1395840 1.180g 1.081g S   0.0  3.8  23:15.18 gaussdb: OMM OMM localhost(60738) idle
14953 ommdba    20   0 1395824 1.180g 1.081g S   0.0  3.8  23:15.96 gaussdb: OMM OMM localhost(60770) idle
14995 ommdba    20   0 1395560 1.180g 1.081g S   0.0  3.8  23:13.28 gaussdb: OMM OMM localhost(60812) idle
15062 ommdba    20   0 1395820 1.180g 1.081g S   0.0  3.8  23:16.12 gaussdb: OMM OMM localhost(60868) idle
15064 ommdba    20   0 1395512 1.180g 1.081g S   0.0  3.8  23:13.33 gaussdb: OMM OMM localhost(60870) idle
14973 ommdba    20   0 1395528 1.180g 1.081g S   0.0  3.8  23:12.74 gaussdb: OMM OMM localhost(60790) idle
14835 ommdba    20   0 1395536 1.180g 1.081g S   0.0  3.8  23:17.39 gaussdb: OMM OMM localhost(60704) idle
14822 ommdba    20   0 1395524 1.180g 1.081g S   0.0  3.8  23:13.80 gaussdb: OMM OMM localhost(60692) idle
14991 ommdba    20   0 1395808 1.180g 1.081g S   0.0  3.8  23:17.96 gaussdb: OMM OMM localhost(60808) idle
14975 ommdba    20   0 1395812 1.180g 1.081g S   0.0  3.8  23:12.57 gaussdb: OMM OMM localhost(60792) idle
15038 ommdba    20   0 1395520 1.180g 1.081g S   0.0  3.8  23:12.75 gaussdb: OMM OMM localhost(60846) idle
14919 ommdba    20   0 1395540 1.180g 1.081g S   0.0  3.8  23:11.58 gaussdb: OMM OMM localhost(60744) idle
14832 ommdba    20   0 1395476 1.180g 1.081g S   0.0  3.8  23:13.11 gaussdb: OMM OMM localhost(60702) idle
14989 ommdba    20   0 1395500 1.180g 1.081g S   0.0  3.8  23:15.63 gaussdb: OMM OMM localhost(60806) idle
14979 ommdba    20   0 1395448 1.180g 1.081g S   0.0  3.8  23:13.17 gaussdb: OMM OMM localhost(60796) idle
15047 ommdba    20   0 1395512 1.180g 1.081g S   0.0  3.8  23:12.10 gaussdb: OMM OMM localhost(60854) idle
14977 ommdba    20   0 1395496 1.180g 1.081g S   0.0  3.8  23:16.90 gaussdb: OMM OMM localhost(60794) idle
15028 ommdba    20   0 1395800 1.180g 1.081g S   0.0  3.8  23:09.35 gaussdb: OMM OMM localhost(60836) idle
```

原因分析

- PostgreSQL缓存：除了常见的执行计划缓存、数据缓存，PostgreSQL为了提高生成执行计划的效率，还提供了catalog，relation等缓存机制。长连接场景下这些缓存中的某些缓存是不会主动释放的，因此可能导致长连接占用大量的内存不释放。
- PMS是MRS的监控进程，此进程会经常创建表分区或者新表，由于PostgreSQL会缓存当前会话访问过的对象的元数据，且PMS的数据库连接池连接会长时间存在，所以连接占用的内存会逐渐上升。

处理步骤

步骤1 以**root**用户登录主Master节点。

步骤2 执行如下命令查询PMS进程号。

ps -ef | grep =pmsd |grep -v grep

步骤3 执行如下命令关闭PMS进程，其中**PID**为**步骤2**中获取的PMS进程号。

kill -9 *PID*

步骤4 等待PMS进程自动启动。

PMS启动需要2~3分钟。PMS是监控进程，重启不影响大数据业务。

----结束

5.21 Knox 进程占用内存高

用户问题

knox进程占用内存高。

问题现象

主Master节点内存使用率高，用**top -c**命令查看到占用内存较高的进程中有knox进程，且此进程占用内存超过4 GB。

原因分析

knox进程没有单独配置内存，进程会自动根据系统内存大小按照比例划分可用内存，导致knox占用内存大。

处理步骤

步骤1 以**root**用户分别登录Master节点。

步骤2 打开文件“/opt/knox/bin/gateway.sh”，查找APP_MEM_OPTS，并设置该参数的值为：“-Xms3072m -Xmx4096m”。

步骤3 登录Manager页面，在主机列表页面找到主Master节点的IP地址（即主机名称前带有实心五角星的节点），并登录该节点后台。

步骤4 执行如下命令重启进程。

su - omm

sh /opt/knox/bin/restart-knox.sh

----结束

5.22 安全集群外的客户端访问 HBase 很慢

问题现象

用户创建了MRS安全集群，在集群外节点安装了集群客户端，并使用客户端命令**hbase shell**访问HBase，发现访问HBase非常慢。

原因分析

安全集群需要进行Kerberos认证，需要在客户端节点的hosts中配置信息，访问速度才不会受到影响。

处理步骤

将集群节点上的hosts文件内容复制到安装客户端节点的hosts文件中。

5.23 作业无法提交如何定位?

问题背景与现象

通过DataArts Studio或者在MRS管理控制台无法提交作业。

问题影响

作业无法提交，业务中断。

处理步骤

步骤1 确认异常来源。

查看作业日志中收到的错误码，确认错误码是属于APIG还是MRS。

- 若是公共APIG的错误码（APIG的错误码是APIGW开头），联系公共APIG维护人员。
- 若是MRS侧错误，继续下一步。

步骤2 排查服务和进程运行状态等基本情况。

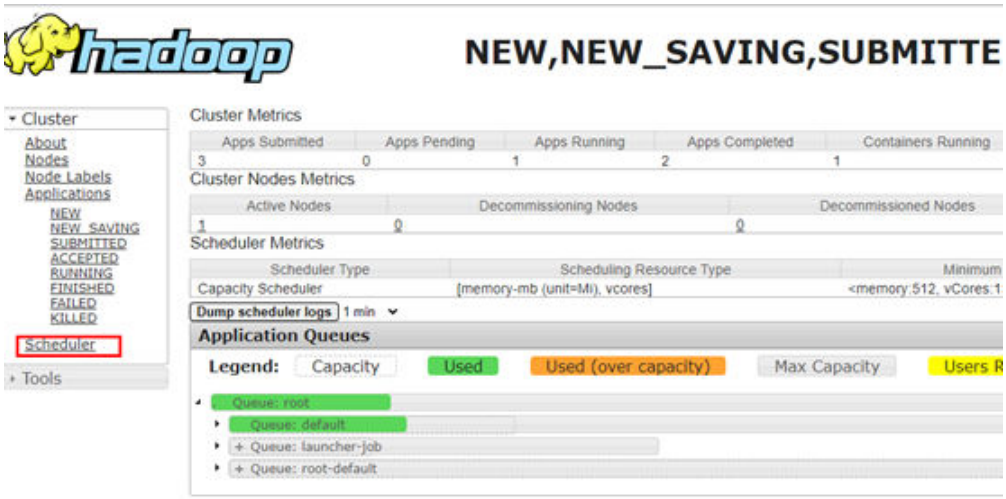
1. 登录Manager界面确认是否有服务故障，如果有作业相关服务故障或者底层基础服务故障，需要解决故障。
2. 查看是否有严重告警。
3. 登录主Master节点。
4. 执行如下命令查看OMS状态是否正常，主OMS节点executor和knox进程是否正常。knox是双主模式，executor是单主模式。

   /opt/Bigdata/om-0.0.1/sbin/status-oms.sh
5. 以**omm**用户执行**jmap -heap** *PID*检查knox和executor进程内存使用情况，如果多次执行查看到老生代内存使用率为99.9%说明有内存溢出。
   - 查询executor进程PID：**netstat -anp | grep 8181 | grep LISTEN**
   - 查询knox进程PID：**ps -ef|grep knox | grep -v grep**

   如果内存溢出，需要现在执行**jmap -dump:format=b,file=/home/omm/temp.bin** *PID*，导出内存信息后重启进程进行恢复。
6. 查看Yarn的原生界面，确认队列资源情况，以及任务是否提交到了yarn上。

   Yarn的原生界面：在集群详情页选择“组件管理 > Yarn > ResourceManager WebUI > ResourceManager (主)”。

   图 5-14 Yarn 界面队列资源情况



步骤3 排查任务提交失败点。

1. 登录MRS控制台，单击集群名称进入集群详情页面。
2. 选择“作业管理”页签，单击作业所在行“操作”列的“查看日志”。
3. 若没有日志或者日志信息不详细，则在“作业名称/ID”列复制作业ID。
4. 在主OMS节点执行如下命令确认任务请求是否下发到了knox，如果请求没有到knox则可能是knox出了问题，需要尝试重启knox进行恢复。

   grep "mrsjob" /var/log/Bigdata/knox/logs/gateway-audit.log | tail -10

5. 进入executor的日志中搜索作业ID，查看报错信息。

   日志路径：/var/log/Bigdata/executor/logs/exe.log

6. 修改“/opt/executor/webapps/executor/WEB-INF/classes/log4j.properties”文件开启executor的debug日志，提交测试任务，查看executor的日志并确认作业提交过程中的报错。

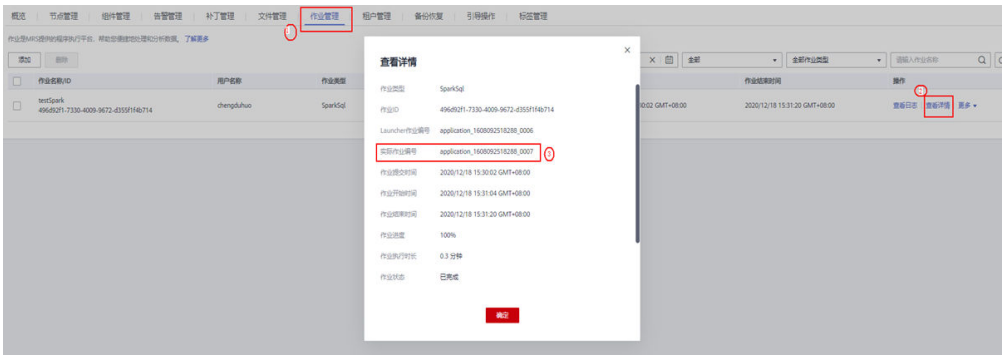
   日志路径：/var/log/Bigdata/executor/logs/exe.log

7. 如果当前任务在exeutor中出错，执行如下命令打印executor的jstack信息，确认线程当前执行状态。

   jstack PID > xxx.log

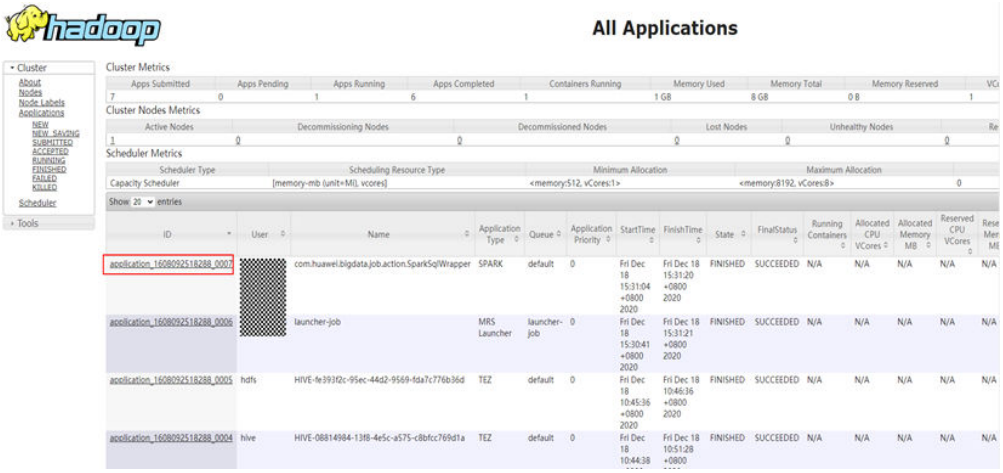
8. 在集群详情页面选择“作业管理”页签，单击作业所在行“操作”列的“查看详情”，获取“实际作业编号”applicationID。

   图 5-15 作业详情



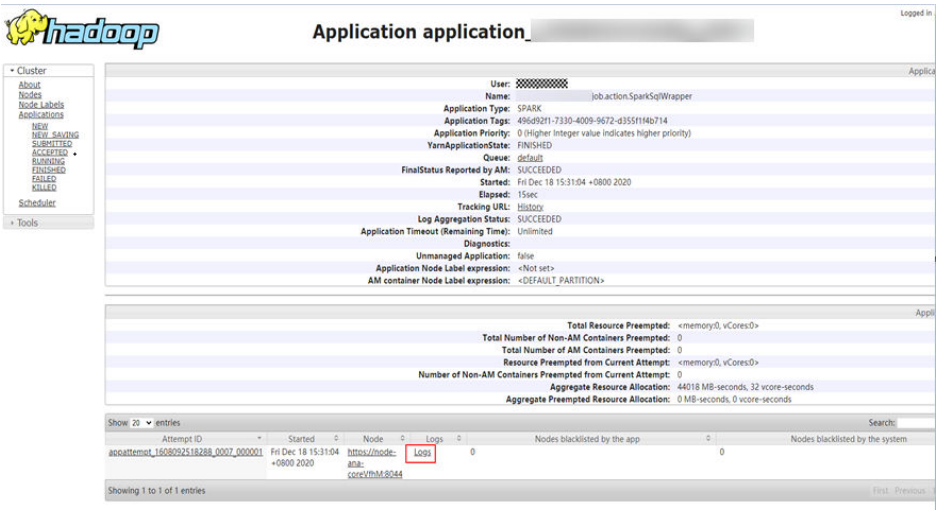
9. 在集群详情页选择“组件管理 > Yarn > ResourceManager WebUI > ResourceManager (主)”进去Yarn的原生界面，单击applicationID。

   图 5-16 Yarn 的 Applications



10. 在任务详情页面查看日志。

图 5-17 任务日志



----结束

5.24 HBase 日志文件过大导致 OS 盘空间不足

用户问题

OS盘/var/log分区空间不足。

问题现象

“/var/log/Bigdata/hbase/*/hbase-omm-*.out”日志文件过大，造成OS盘/var/log分区空间不足。

原因分析

在HBase长时间运行场景下，操作系统会把JVM创建的“/tmp/.java_pid*”文件定期清理。因为HBase的内存监控使用了JVM的jinfo命令，而jinfo依赖“/tmp/.java_pid*”文件，当该文件不存在时，jinfo会执行**kill -3**将jstack信息打印到.out日志文件里，从而导致.out日志文件过大。

处理步骤

在每个HBase实例的节点上部署定期清理.out日志文件的定时任务。后台登录HBase的实例节点，在crontab -e中添加每天0点清理.out日志的定时任务。

crontab -e

00 00 * * * for file in `ls /var/log/Bigdata/hbase/*/hbase-omm-*.out`; do echo "" > $file; done

说明

如果.out大文件出现比较频繁，可以每天清理多次或者调整操作系统的自动清理策略。

5.25 HDFS 日志文件过大导致 OS 盘空间不足

用户问题

OS盘/var/log分区空间不足。

问题现象

“/var/log/Bigdata/hdfs/*/hdfs-omm-*.out”日志文件过大，造成OS盘/var/log分区空间不足。

原因分析

在HDFS长时间运行场景下，操作系统会把JVM创建的“/tmp/.java_pid*”文件定期清理。因为HDFS的内存监控使用了JVM的jinfo命令，而jinfo依赖“/tmp/.java_pid*”文件，当该文件不存在时，jinfo会执行**kill -3**将jstack信息打印到.out日志文件里，从而导致.out日志文件过大。

处理步骤

在每个HDFS实例的节点上部署定期清理.out日志文件的定时任务。后台登录HDFS的实例节点，在crontab -e中添加每天0点清理.out日志的定时任务。

crontab -e

00 00 * * * for file in `ls /var/log/Bigdata/hdfs/*/hdfs-omm-*.out`; do echo "" > $file; done

说明

如果.out大文件出现比较频繁，可以每天清理多次或者调整操作系统的自动清理策略。

5.26 MRS 集群节点规格升级异常

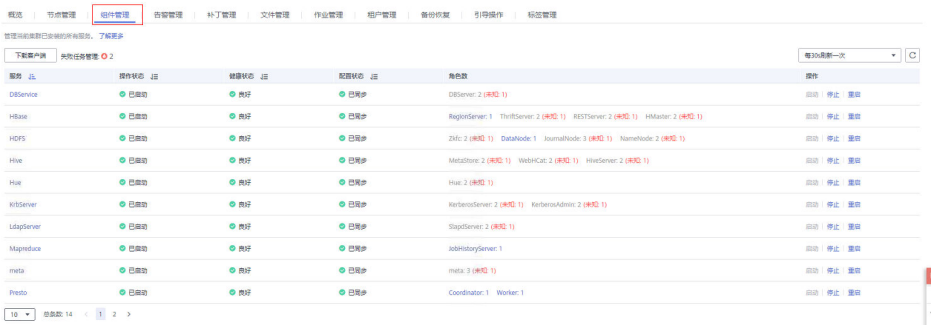
问题现象

升级节点规格可能存在规格升级成功后组件启动失败或组件异常导致节点升级后出现异常的情况，现象如下：

- 升级失败的节点规格已经升级成功。

- 失败任务管理中有新增的升级规格失败任务。

- 若IAM用户已同步，可在组件管理观察到存在异常角色。



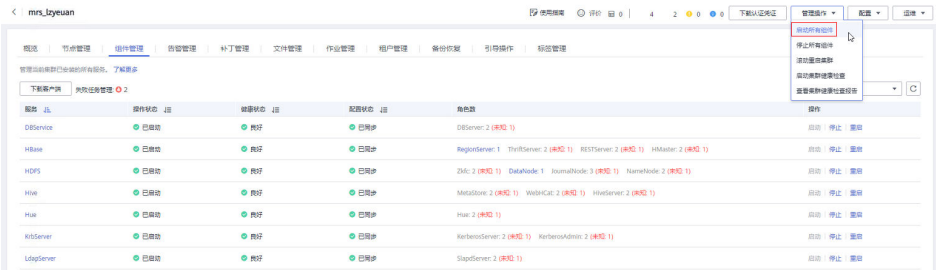
若未同步，可在集群Manager页面观察到存在异常角色。

原因分析

需解决异常，等集群恢复正常后继续升级节点。

处理步骤

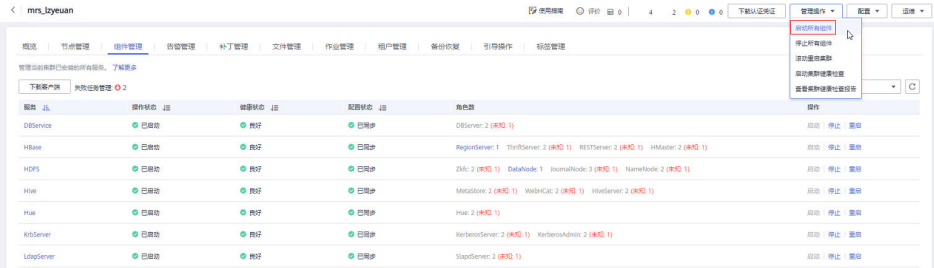
- 方式一：
  a. 进入集群组件管理页，查看服务健康状态与角色状态，若存在“未启动”的角色，进入服务。



  b. 勾选未开启的实例，单击“更多”，选择“启动实例”。

说明

若异常角色较多，可在管理操作处选择启动所有组件。



- 方式二：

  进入集群manager页面，查看服务，是否存在“未启动”实例，若存在，启动实例。

- 若存在其他情况导致服务异常无法解决，请联系技术服务协助处理。

5.27 Manager 页面新建的租户删除失败

问题现象

在FusionInsight Manager的“租户资源”页面添加租户后，删除租户时，报“删除租户角色失败”。

原因分析

在创建租户时会生成对应的角色，执行删除租户操作时会首先删除对应的角色。此时如果支持权限配置的组件状态异常，则会导致删除这个角色对应的资源权限失败。

处理步骤

步骤1 登录FusionInsight Manager，选择“系统 > 权限 > 角色”。

步骤2 单击“添加角色”，在“配置资源权限”中单击集群名称，确认可配置资源权限的组件。

步骤3 选择“集群 > 服务”，查看可配置资源权限的组件的运行状态是否都为“良好”。

步骤4 如果不为“良好”，请启动或者修复组件，直至状态为“良好”。

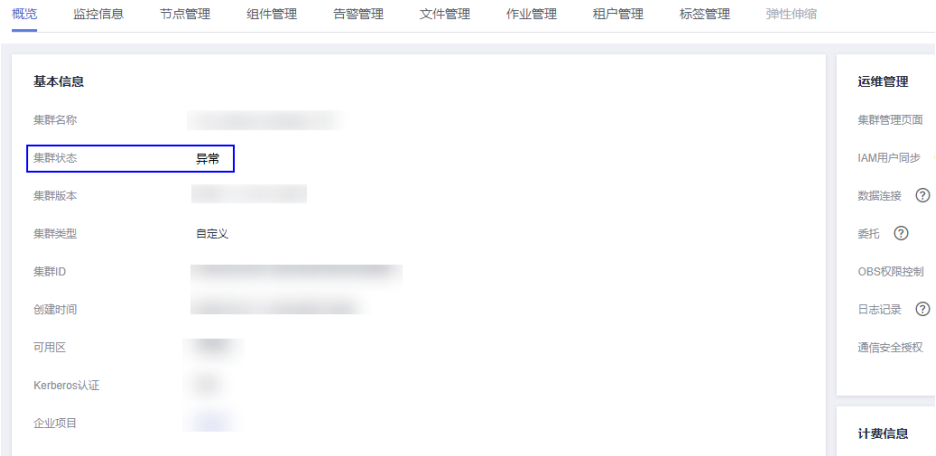
步骤5 再次执行删除租户操作。

----结束

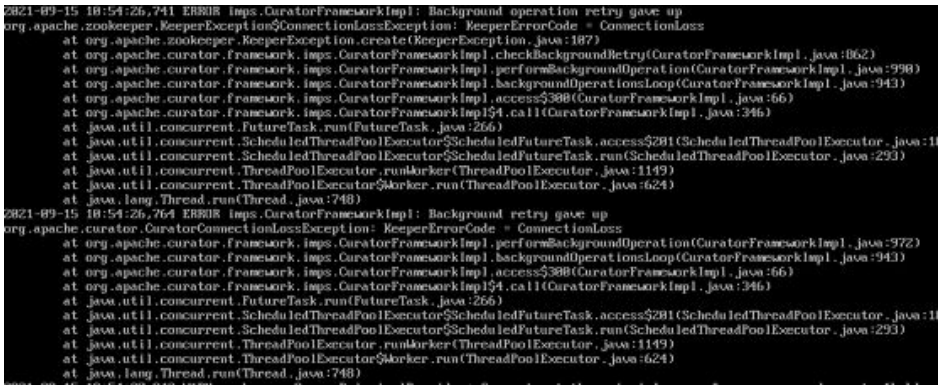
5.28 MRS 集群切换 VPC 后集群状态异常不可用

问题现象

客户MRS集群，在ECS侧将所有节点的VPC切换后集群状态异常。



所有服务不可用，其中Hive beeline报错如下：



原因分析

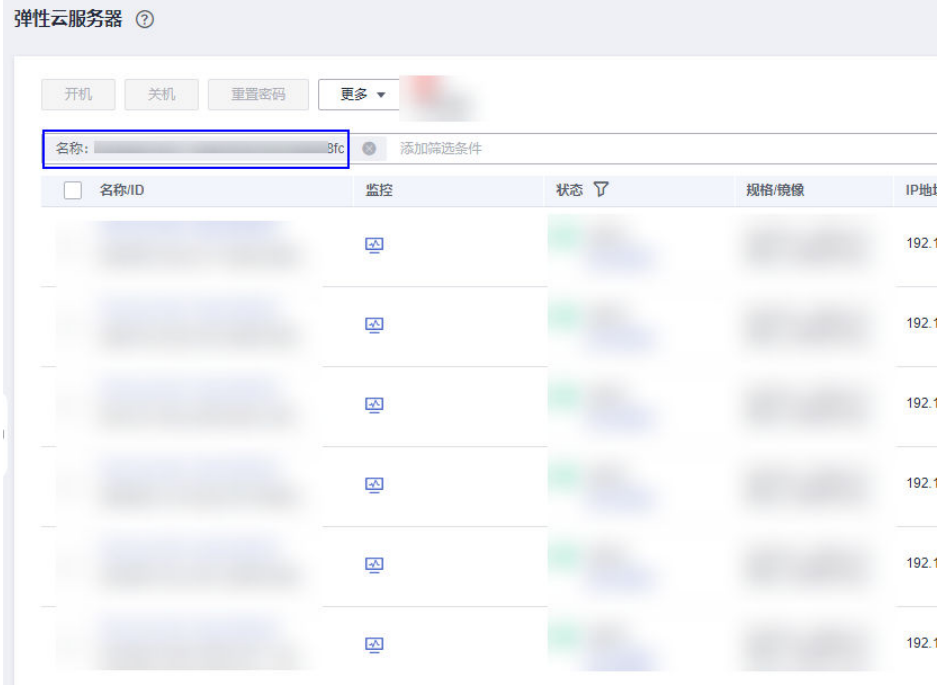
MRS不支持切换VPC，切换VPC后，节点的内网IP变化，但是配置文件和数据库还是原有的IP，导致集群通信等功能异常，集群状态也会异常。所以要恢复集群，就需要将节点切回到原来的VPC，并且IP和hosts文件中的IP主机映射一一对应。

处理步骤

步骤1 登录集群Master1节点，执行**ifconfig**查看到切换后的VPC。 执行命令**cat /etc/hosts**，查看host文件中记录切换VPC之前的IP。

步骤2 登录MRS管理控制台，在MRS集群概览页面查看“集群ID”和“虚拟私有云”名称。

步骤3 登录ECS管理控制台，在搜索栏选择“名称”输入MRS集群ID，搜索MRS集群的所有节点。



步骤4 在MRS集群节点的操作列选择“更多 > 网络设置 > 切换VPC”，切换节点VPC。

说明

- 每个节点都需要切换VPC。
- 切换VPC时“虚拟私有云”、“子网”、“安全组”需要和集群初始的配置一致。
- “私有IP地址”选择“现在创建”，填写**步骤1**中查询对应节点的IP。

步骤5 切换成功后，单击节点名称，需要在节点的“弹性网卡”中重启“源/目的检查”。

步骤6 将虚拟IP重新绑定到集群的Master节点上，操作如下：

1. 登录MRS管理控制台，进入MRS集群，在“概览”页面单击“前往Manager”后的，“访问方式”选择“专线访问”，记录集群浮动IP。查看“默认生效子网”后的子网，并记录。
2. 登录VPC管理控制台，选择“虚拟私有云 > 子网”，搜索MRS集群的子网。
3. 单击子网名称，进入子网详情页面，单击“IP地址管理”，搜索MRS集群浮动IP。
4. MRS集群浮动IP“操作”列的“绑定服务器”，选择MRS集群的Master节点，绑定成功后如下图：

步骤7 等待集群恢复。

----结束

5.29 MRS 管理控制台上提交作业异常处理

用户问题

MRS控制台作业管理提交作业“状态”为“已接受”，“执行结果”为“未定”，作业未成功提交到Yarn上，如下图所示：

原因分析

由于管理控制台的作业管理功能是由集群管理模块Executor来负责调度执行，因此作业未提交到Yarn上，根因需要查看Executor。而管理控制台的作业管理功能，正常情况添加作业后会自动在Yarn上启动两个任务，一个提交到launcher-job队列，该队列为辅助作业队列。另外一个是作业实际执行的队列，如默认的default队列。

查看Executor日志（主Master节点的“/var/log/executor/exe.log”），发现是由于提交Flink作业用户的密码改变或者过期导致下载用户keytab认证文件失败，最终导致作业未提交到launcher-job队列。

处理步骤

步骤1 重置提交作业用户的密码。

登录Manager页面，选择“系统设置 > 用户管理”。在提交作业的IAM用户的操作列，选择“更多 > 初始化密码”，根据界面提示操作。初始化完成后需要使用该用户登录一次MRS Manager。

步骤2 登录MRS管理控制台，进入MRS集群，在“概览”页面，单击“IAM用户同步”右侧的“单击同步”。

步骤3 IAM同步完成，作业管理添加配置作业提交作业即可正常。

----结束

5.30 生成 HA 证书时报错“symbol xxx not defined in file libcrypto.so.1.1”

用户问题

更换HA证书时，执行**sh ${OMS_RUN_PATH}/workspace/ha/module/hacom/script/gen-cert.sh --root-ca --country=CN --state=***state* **--city=***city* **--company=***company* **--organize=***organize* **--common-name=***commonname* **--**

email=*集群用户邮箱*命令在主管理节点“${OMS_RUN_PATH}/workspace0/ha/local/cert”目录生成“root-ca.crt”和“root-ca.pem”时，发生以下报错：

```
openssl: relocation error: openssl: symbol BIO_new_dgram_sctp version OPENSSL_1_1_0 not defined in file libcrypto.so.1.1 with link time referencecreate server private key failed.
```

原因分析

- 用户可能修改了openssl不是系统默认的/usr/bin/openssl。
- 动态库依赖libcrypto.so.1.1无法找到。
- 如果集群为3.2.0及之前版本，可能在执行操作前执行了配置环境变量的命令（例如**source bigdata_env**），或修改了环境变量。3.2.0之后版本已修复该问题。

处理步骤

步骤1 以**omm**用户通过主管理节点IP登录主管理节点。

步骤2 执行以下命令查看执行结果是否为“/usr/bin/openssl”。如果不是请修改openssl为系统默认的/usr/bin/openssl。

which openssl

步骤3 执行以下命令查看动态库依赖。

ldd /usr/bin/openssl

例如执行后结果如下：

```
[omm@xxx ~]$ ldd /usr/bin/openssl
        linux-vdso.so.1 (0x0000ffffaf016000)
        libssl.so.1.1 => /usr/lib64/libssl.so.1.1 (0x0000ffffaee7a000)
        libcrypto.so.1.1 => /usr/lib64/libcrypto.so.1.1 (0x0000ffffaebc2000)
        libz.so.1 => /usr/lib64/libz.so.1 (0x0000ffffaeb91000)
        libdl.so.2 => /usr/lib64/libdl.so.2 (0x0000ffffaeb70000)
        libpthread.so.0 => /usr/lib64/libpthread.so.0 (0x0000ffffaeb3b000)
        libc.so.6 => /usr/lib64/libc.so.6 (0x0000ffffae9c5000)
        /lib/ld-linux-aarch64.so.1 (0x0000ffffaefe8000)
```

查看执行结果中libcrypto.so.1.1的指向是否有值，如果为not found请执行以下命令加载。

echo $LD_LIBRARY_PATH

步骤4 查看系统库环境变量里是否加载了非系统的openssl相关的库。如果是，请修改为系统的openssl相关的库。

步骤5 如果仍旧无法解决，请联系支持人员。

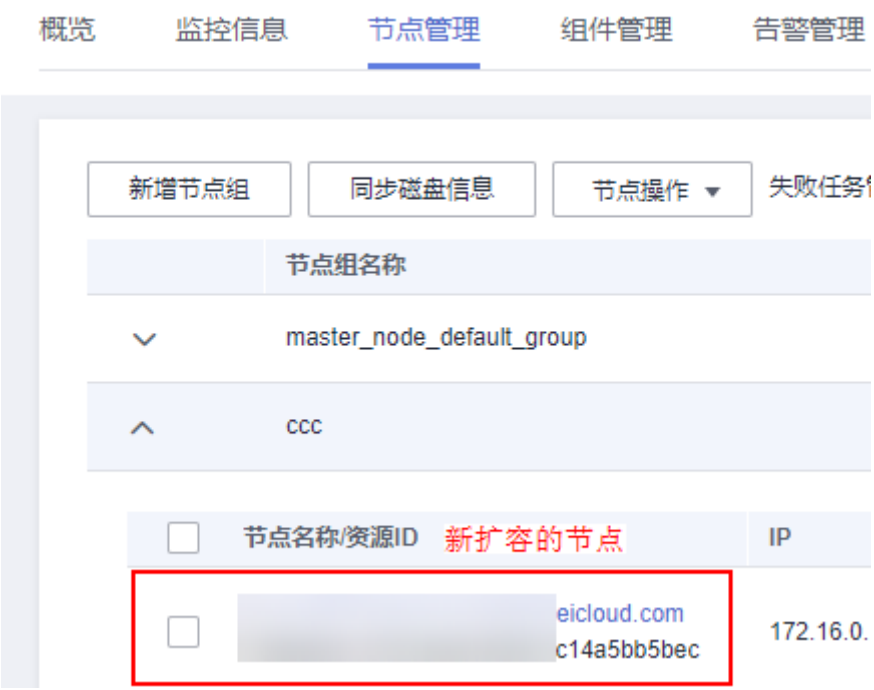
----结束

5.31 MRS 集群 Core 节点扩容成功后部分实例启动失败

问题现象

Core节点扩容可能存在扩容节点完成，但节点上存在启动失败的实例的情况，现象如下：

1. Core节点已经扩容完成，节点管理页面可以看到新扩容的节点：



2. 任务管理中有添加节点的任务失败或部分成功：

3. 若IAM用户已同步，可在组件管理观察到存在未启动角色。
4. 若未同步，可在集群Manager页面观察到存在未启动角色。

处理步骤

场景一：添加节点任务在安装组件前失败

步骤1 如果MRS集群为按需购买集群：

1. 登录MRS服务控制台。
2. 选择“现有集群”，单击集群名称进入集群详情页面。
3. 单击页面上面的 按钮，在“任务名”列，单击Core节点扩容任务。
4. 记录校验请求参数里的所有节点。
5. 选择“节点管理”页签，单击**步骤1.4**记录的节点，并单击右上角的“关机”，按界面提示操作进行关机。

6. 参考**缩容集群**缩容节点。

步骤2 如果MRS集群为包周期集群：参考**退订包周期集群指定节点**章节对异常节点进行退订。

----结束

场景二：添加节点任务在安装组件后失败

步骤1 登录MRS控制台。

步骤2 选择“现有集群”，单击集群名称进入集群详情页面。

步骤3 在“概览”页签单击“IAM用户同步”右侧的“单击同步”进行IAM用户同步。

步骤4 单击“组件管理”，查看各服务的角色操作状态，如果存在“未启动”的角色，单击“更多 > 启动实例”，启动该实例。

步骤5 如果启动失败，根据任务列表里中任务的报错信息进行修复再尝试重启。

说明

- 如果异常角色较多，可单击右上角的“管理操作”选择启动所有组件。
- 如果存在其他情况导致服务异常无法解决，请联系技术服务协助处理。
- 也可通过集群Manager页面进行启动实例操作，具体请参考**实例管理概述**。

----结束

6 使用 Alluixo

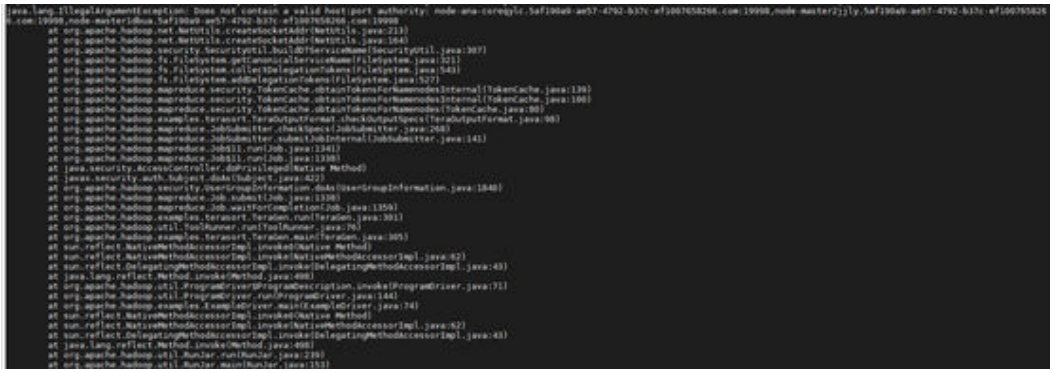
6.1 Alluixo 在 HA 模式下出现 Does not contain a valid host:port authority 报错

用户问题

安全集群Alluixo在HA模式下出现Does not contain a valid host:port authority的报错，如何处理?

问题现象

安全集群中，Alluixo在HA模式下出现Does not contain a valid host:port authority的报错。



原因分析

org.apache.hadoop.security.SecurityUtil.buildDTServiceName不支持在uri中填写多个alluxiomaster的地址。

处理步骤

使用alluxio:///或者alluxio://<主AlluxioMaster的ip或hostname>:19998/进行访问。

7 使用 ClickHouse

7.1 ZooKeeper 上数据错乱导致 ClickHouse 启动失败问题

问题现象

ClickHouse集群中某实例节点启动失败，该实例节点启动日志中有如下类似报错信息：

```
2021.03.15 21:01:19.816593 [ 11111 ] {} <Error> Application: DB::Exception:
The local set of parts of table DEFAULT.lineorder doesn't look like the set ofdoesn't look like the set of
parts in ZooKeeper: 59.99 million rows of 59.99 million total rows in
filesystem are suspicious. There are 30 unexpected parts with 59986052 rows
(14 of them is not just-written with 59986052 rows), 0 missing parts (with 0
blocks).: Cannot attach table `DEFAULT`.`lineorder` from metadata file
...
: while loading database
```

原因分析

使用ClickHouse过程中，ClickHouse实例异常场景下，重复创建集群ReplicatedMergeTree引擎表，后续又进行删除表等操作导致ZooKeeper上的数据异常，致使ClickHouse启动失败。

解决办法

步骤1 备份问题节点数据库下所有表数据到其他目录。

- 备份表数据：

  MRS 3.0.5及之前版本

  cd /srv/BigData/data1/clickhouse/data/*数据库名*

  mkdir -p *备份目录***/data1**

  mv *{表名}* *备份目录***/data1/**

  MRS 3.1.0及之后版本

  head -1 /srv/BigData/data1/clickhouse_path/metadata/*库名***|***表名***.sql | awk -F ' ' '{print $5}' | sed "s/'//g"** *获取到目标的UUID*

  cd /srv/BigData/data1/clickhouse/store/*{UUID前3个字符}*

  mkdir -p *备份目录***/data1**

  mv *{UUID}* *备份目录***/data1/**

说明

如果存在多磁盘的场景，需要对**data1**到**dataN**的磁盘数据都执行相同的备份操作。

- 备份元数据信息：

  cd /srv/BigData/data1/clickhouse_path/metadata/库名

  mv *表名*.**sql** *备份目录*

例如，下面是备份default数据库下的表lineorder数据到**/home/backup**目录下。

cd /srv/BigData/data1/clickhouse/data/default

mkdir -p /home/backup/data1

mv lineorder /home/backup/data1

cd /srv/BigData/data1/clickhouse_path/metadata

mv lineorder.sql /home/backup

步骤2 登录MRS Manager页面，选择“集群 > 服务 > ClickHouse > 实例”，选择对应的实例节点，单击“启动实例”，完成实例启动。

步骤3 实例启动成功后，使用ClickHouse客户端登录问题节点。命令中如果携带认证密码信息可能存在安全风险，在执行命令前建议关闭系统的history命令记录功能，避免信息泄露。

clickhouse client --host *clickhouse实例IP* **--user** *用户名* **--password** *密码*

步骤4 执行以下命令获取当前表所在的ZooKeeper路径**zookeeper_path**和对应节点所在的副本编号**replica_num**。

SELECT zookeeper_path FROM system.replicas WHERE database = '*数据库名***' AND table = '***表名***';**

SELECT replica_num,host_name FROM system.clusters;

步骤5 执行以下命令连接ZooKeeper命令行界面。

zkCli.sh -server *ZooKeeper所在节点的IP*.**2181**

获取ZooKeeper的IP地址，具体请参考**如何获取ZooKeeper地址?**

步骤6 找到对应故障节点表数据对应的ZooKeeper路径。

ls *zookeeper_path*/**replicas/**/*replica_num*

说明

zookeeper_path为**步骤4**中查询到的**zookeeper_path**值。

replica_num为**步骤4**中节点主机对应的副本编号**replica_num**的值。

步骤7 执行以下命令，删除ZooKeeper上的副本数据。

deleteall *zookeeper_path*/**replicas/**/*replica_num*

步骤8 使用ClickHouse客户端登录问题节点，重新执行create创建集群ReplicatedMergeTree引擎表。

clickhouse client --host *clickhouse实例IP* **--multiline --user** *用户名* **--password** *密码*

CREATE TABLE *数据库名.表名* **ON CLUSTER** *集群名*

...

ENGINE = ReplicatedMergeTree ...

其他副本节点有如下提示表已经存在的报错信息，属于正常现象，可以忽略。

```
Received exception from server (version 20.8.7):
Code: 57. DB::Exception: Received from x.x.x.x:9000. DB::Exception:
There was an error on [x.x.x.x:9000]: Code: 57, e.displayText() =
DB::Exception: Table DEFAULT.lineorder already exists. (version 20.8.11.17
(official build)).
```

建表成功后问题节点上表数据会自动进行同步，数据恢复完成。

----结束

7.2 ClickHouse 消费 Kafka 数据异常

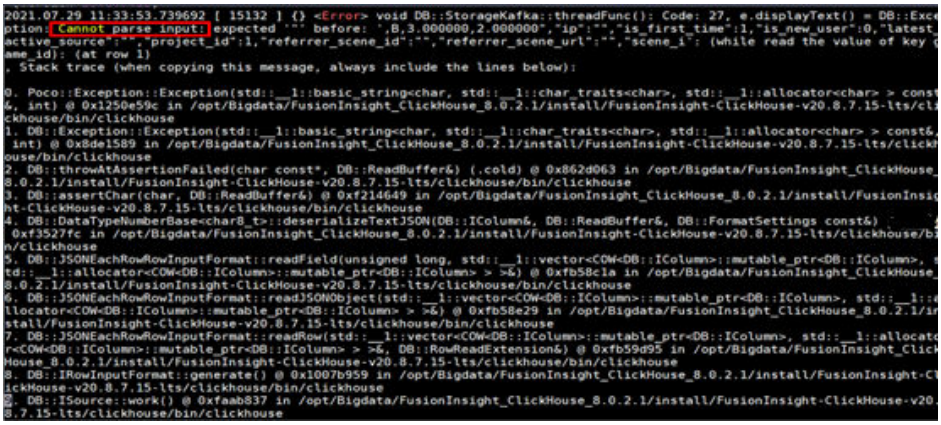
问题现象

用户在ClickHouse集群创建Kafka引擎表test.user_log_kafka消费Kafka数据，查看Kafka监控发现凌晨开始出现消息堆积，数据一直没有被消费。

原因分析

Kafka出现消息堆积，说明ClickHouse消费数据时出现异常，需要查看ClickHouse的日志。

1. 登录MRS集群，进入ClickHouse实例所在的节点，查看“/var/log/Bigdata/clickhouse”目录下的“clickhouse-server.log”日志文件，发现以下报错：



2. 进入到其他ClickHouse节点也发现了同样的报错日志，由此可知Kafka消息堆积是因为ClickHouse解析Kafka数据时出现异常。

解决办法

使用以下命令修改表的“kafka_skip_broken_messages”属性：

```
# ALTER test.user_log MODIFY SETTINGS kafka_skip_broken_messages=10000
```

说明

- 10000可以根据数据中脏数据的比例进行调整。
- kafka_skip_broken_messages：Kafka消息解析器对每个块的架构不兼容消息的容忍度，默认值：0。

  例如：kafka_skip_broken_messages = N，则引擎会跳过N条无法解析的Kafka消息。

8 使用 DBservice

8.1 DBServer 实例状态异常

问题背景与现象

DBServer实例状态一直是concerning。

图 8-1 DBServer 实例状态

角色	主机名	管理IP	业务IP	机架	操作状态	健康状态
DBServer(主)	node-master2JOcB	192.168.5.133	192.168.5.133	/default/rack9bdf	已启动	良好
DBServer(备)	node-master1DEdJ	192.168.5.42	192.168.5.42	/default/rack9bdf	已启动	恢复中

原因分析

数据目录下文件或目录的权限不对，GaussDB要求文件权限至少是600，目录权限至少为700。

图 8-2 目录权限列表

```
omm@ 192-168-234-176:/srv/BigData/dbdata_service> ll
total 4
drwx------ 19 omm wheel 4096 Dec 14 10:15 data
```


版权所有 © 华为云计算技术有限公司

图 8-3 文件权限列表

```
omm@192-168-234-176:/srv/BigData/dbdata_service/data> ll
total 128
drwx------ 6 omm wheel  4096 Dec  9 15:47 base
-rw------- 1 omm wheel   922 Dec  9 15:34 dblink.conf
-rw------- 1 omm wheel    16 Dec 14 10:15 gaussdb.state
drwx------ 2 omm wheel  4096 Dec 14 10:17 global
drwx------ 2 omm wheel  4096 Dec 11 00:00 pg_audit
drwx------ 2 omm wheel  4096 Dec 14 10:15 pg_blackbox
drwx------ 2 omm wheel  4096 Dec  9 15:34 pg_clog
drwx------ 2 omm wheel  4096 Dec 14 10:15 pg_confile_backup
-rw------- 1 omm wheel  1024 Dec  9 15:34 pg_ctl.lock
-rw------- 1 omm wheel  4245 Dec  9 15:47 pg_hba.conf
-rw------- 1 omm wheel  1024 Dec  9 15:47 pg_hba.conf.lock
-rw------- 1 omm wheel  1636 Dec  9 15:34 pg_ident.conf
drwx------ 2 omm wheel  4096 Dec  9 15:38 pg_log
drwx------ 4 omm wheel  4096 Dec  9 15:34 pg_multixact
drwx------ 2 omm wheel  4096 Dec 14 10:15 pg_notify
drwx------ 2 omm wheel  4096 Dec  9 15:34 pg_serial
drwx------ 2 omm wheel  4096 Dec  9 15:34 pg_snapshots
drwx------ 2 omm wheel  4096 Dec 14 11:56 pg_stat_tmp
drwx------ 2 omm wheel  4096 Dec  9 15:34 pg_subtrans
drwx------ 2 omm wheel  4096 Dec  9 15:34 pg_tblspc
drwx------ 2 omm wheel  4096 Dec  9 15:34 pg_twophase
-rw------- 1 omm wheel     4 Dec  9 15:34 PG_VERSION
drwx------ 2 omm wheel  4096 Dec  9 15:34 pg_wallet
drwx------ 3 omm wheel  4096 Dec  9 15:39 pg_xlog
-rw------- 1 omm wheel 13309 Dec 14 10:15 postgresql.conf
-rw------- 1 omm wheel  1024 Dec  9 15:34 postgresql.conf.lock
-rw------- 1 omm wheel   105 Dec 14 10:15 postmaster.opts
-rw------- 1 omm wheel    96 Dec 14 10:15 postmaster.pid
```

解决办法

步骤1 按照图8-2和图8-3的权限列表，修改相应文件和目录的权限。

步骤2 重启相应的DBServer实例。

----结束

8.2 DBServer 实例一直处于 Restoring 状态

问题背景与现象

DBServer实例状态一直是Restoring状态，重启之后仍然不恢复。

原因分析

1. DBService组件会对“${BIGDATA_HOME}/MRS_*XXX*/install/dbservice/ha/module/harm/plugin/script/gsDB/.startGS.fail”这个文件监控。其中*XXX*是产品版本号。
2. 如果这个文件中的值大于3就会启动失败，NodeAgent会一直尝试重启该实例，此时仍会失败而且这个值每启动失败一次就会加1。

解决办法

步骤1 登录Manager管理界面。

步骤2 停止该DBServer实例。

步骤3 使用**omm**用户登录到DBServer实例异常的节点。

步骤4 修改“${BIGDATA_HOME}/MRS_*XXX*/install/dbservice/ha/module/harm/plugin/script/gsDB/.startGS.fail”配置文件中的值为0。其中*XXX*是产品版本号。

步骤5 启动该DBServer实例。

----结束

8.3 DBService 默认端口 20050 或 20051 被占用

问题背景与现象

执行DBService服务重启操作时，DBService服务启动失败，打印的错误日志中出现20050或20051端口被占用等信息。

原因分析

1. 由于DBService使用的默认端口20050或20051被其他进程占用。
2. DBService进程没有停止成功，使用的端口未释放。

解决办法

该解决办法以20051端口被占用为例，20050端口被占用的解决办法与该办法类似。

步骤1 以**root**用户登录DBService安装报错的节点主机，执行命令：**netstat -nap | grep 20051**查看占用20051端口的进程。

步骤2 使用**kill**命令强制终止使用20051端口的进程。

步骤3 约2分钟后，再次执行命令：**netstat -nap | grep 20051**，查看是否还有进程占用该端口。

步骤4 确认占用该端口进程所属的服务，并修改为其他端口。

步骤5 分别在“/tmp”和“/var/run/MRS-DBService/”目录下执行**find . -name "*20051*"**命令，将搜索到的文件全部删除。

步骤6 登录Manager，重启DBService服务。

----结束

8.4 /tmp 目录权限不对导致 DBserver 实例状态一直处于 Restoring

问题背景与现象

DBServer实例状态一直是Restoring状态，重启之后仍然不恢复。

原因分析

1. 查看“/var/log/Bigdata/dbservice/healthCheck/dbservice_processCheck.log”，可以看到gaussdb异常。

   图 8-4 gaussdb 异常

```
[2019-07-22 10:57:00] ERROR: [:108]: Host 192.168.5.42 gaussdb status is Exception.
[2019-07-22 10:57:00] ERROR: [:154]: Check DBService health failed.
[2019-07-22 10:57:10] INFO: [:84]: check host:192.168.5.42 DBService health.
[2019-07-22 10:57:10] INFO: [:99]: Host 192.168.5.42 floatip status is Normal
Normal.
[2019-07-22 10:57:10] ERROR: [:108]: Host 192.168.5.42 gaussdb status is Exception.
[2019-07-22 10:57:10] ERROR: [:154]: Check DBService health failed.
[2019-07-22 10:57:20] INFO: [:84]: check host:192.168.5.42 DBService health.
[2019-07-22 10:57:20] INFO: [:99]: Host 192.168.5.42 floatip status is Normal
Normal.
[2019-07-22 10:57:20] ERROR: [:108]: Host 192.168.5.42 gaussdb status is Exception.
[2019-07-22 10:57:20] ERROR: [:154]: Check DBService health failed.
[2019-07-22 10:57:30] INFO: [:84]: check host:192.168.5.42 DBService health.
[2019-07-22 10:57:31] INFO: [:99]: Host 192.168.5.42 floatip status is Normal
Normal.
[2019-07-22 10:57:31] ERROR: [:108]: Host 192.168.5.42 gaussdb status is Exception.
[2019-07-22 10:57:31] ERROR: [:154]: Check DBService health failed.
[2019-07-22 10:57:41] INFO: [:84]: check host:192.168.5.42 DBService health.
[2019-07-22 10:57:41] INFO: [:99]: Host 192.168.5.42 floatip status is Normal
```

2. 检查发现“/tmp”权限不对。

   图 8-5 /tmp 权限

```
[root@node-master1DEdJ DB]# ll / -rlth
total 76K
drwxr-xr-x.   2 root root 4.0K Dec 12  2016 mnt
drwxr-xr-x.   2 root root 4.0K Dec 12  2016 media
drwxr-xr-x.  13 root root 4.0K Jul 15 16:25 usr
-rwxr-xr-x.   1 root root 3.8K Jul 15 16:25 README
-rwxr-xr-x.   1 root root    0 Jul 15 16:25 OTC_EulerOS_2.x86_64-0.9.1-20170904-0513
lrwxrwxrwx.   1 root root    8 Jul 15 16:26 sbin -> usr/sbin
lrwxrwxrwx.   1 root root    9 Jul 15 16:26 lib64 -> usr/lib64
lrwxrwxrwx.   1 root root    7 Jul 15 16:26 lib -> usr/lib
lrwxrwxrwx.   1 root root    7 Jul 15 16:26 bin -> usr/bin
drwxr-xr-x.   3 root root 4.0K Jul 15 16:29 srv
drwxr-xr-x.   7 root root 4.0K Jul 15 16:39 CloudResetPwdUpdateAgent
drwxr-xr-x.   7 root root 4.0K Jul 15 16:39 CloudrResetPwdAgent
drwx------.   2 root root  16K Jul 15 16:46 lost+found
dr-xr-xr-x. 236 root root    0 Jul 19 17:36 proc
dr-xr-xr-x.   4 root root 4.0K Jul 19 17:37 boot
dr-xr-xr-x.  13 root root    0 Jul 19 17:37 sys
drwxr-xr-x.  19 root root 4.0K Jul 19 17:37 var
drwxr-xr-x.  19 root root 3.0K Jul 19 17:37 dev
drwxr-xr-x.   2 root root 4.0K Jul 19 17:38 tmpdir
drwxr-xr-x.   7 root root 4.0K Jul 19 17:38 opt
-rw-------.   1 root root    0 Jul 19 17:39 install_os_optimization.log
drwxr-xr-x.   6 root root 4.0K Jul 19 17:54 home
drwxr-xr-x.  86 root root 4.0K Jul 19 17:54 etc
drwxr-xr-x.  30 root root  960 Jul 22 10:49 run
drwx------.  23 root root 4.0K Jul 22 11:42 tmp
drwx------.   5 root root 4.0K Jul 22 11:50 root
```

解决办法

步骤1 修改/tmp的权限。

chmod 1777 /tmp

步骤2 等待实例状态恢复。

----结束

8.5 DBService 备份任务执行失败

问题背景与现象

查看备份文件路径中没有DBService的备份文件。

ls /srv/BigData/LocalBackup/*default_20190720222358***/ -rlth**

图 8-6 查看备份文件

```
drwx------. 2 omm wheel 4096 Aug  5 09:00 LdapServer_20190805090027
drwx------. 2 omm wheel 4096 Aug  5 10:00 LdapServer_20190805100027
drwx------. 2 omm wheel 4096 Aug  5 09:00 NameNode_20190805090027
drwx------. 2 omm wheel 4096 Aug  5 10:00 NameNode_20190805100027
drwx------. 2 omm wheel 4096 Aug  5 09:01 OMS_20190805090027
drwx------. 2 omm wheel 4096 Aug  5 10:01 OMS_20190805100027
```

原因分析

- 查看DBService的备份日志/var/log/Bigdata/dbservice/scriptlog/backup.log，其实备份已经成功，只是上传至OMS节点时失败。

```
[2017-05-18 02:00:54] INFO: [dbservice_backup.sh:528]: Backup file had been saved to V100R002C50SPC205_DBSERVICE_20170518020051.tar.gz
[2017-05-18 02:00:54] DEBUG: [dbservice_backup.sh:570]: uploadScript:/opt/huawei/Bigdata/dbserviceSPC200/sbin/scp_upload.sh, omsFloatIP:192.168.1.2,
srcPath:/opt/huawei/Bigdata/dbserviceSPC200/bak.
[2017-05-18 02:00:54] INFO: [dbservice_backup.sh:587]: Begin to upload file.
Warning: Permanently added '                (ECDSA) to the list of known hosts.
Authorized users only. All activity may be monitored and reported.
ssh: connect to host                port 22: Connection refused
[2017-05-18 02:00:55] ERROR: [dbservice_backup.sh:615]: Upload file(/opt/huawei/Bigdata/dbserviceSPC200/bak) failed.
[2017-05-18 02:00:55] ERROR: [dbservice_backup.sh:555]: Scp backupfile to oms error.
[2017-05-18 02:00:55] ERROR: [dbservice_backup.sh:926]: main: auto backup failed.
[2017-05-18 02:00:55] INFO: [dbservice_backup.sh:929]: main: start create flag file.
[2017-05-18 02:00:55] INFO: [dbservice_backup.sh:750]: Send Alarm(AlarmID:[27002] Category:[0] LocationInfo:[DBService;DBServer;hadoopc1h2]) successful.
                                                                                                                                    1954,1
```

- 失败原因是由于ssh不通。

```
omm@hadoopc1h2:/opt/huawei/Bigdata/dbserviceSPC200/sbin> ssh  hadoopc1h1
Warning: Permanently added 'hadoopc1h1,                ' (ECDSA) to the list of known hosts.
Authorized users only. All activity may be monitored and reported.
Last login: Thu May 18 20:18:45 2017 from
omm@hadoopc1h1:~> ssh
Warning: Permanently added '                ' (ECDSA) to the list of known hosts.
Authorized users only. All activity may be monitored and reported.
Last login: Mon Apr 10 10:50:23 2017 from                2
omm@hadoopc1h2:~> exit
logout
Connection to                closed.
omm@hadoopc1h1:~> ssh
ssh: connect to host                port 22: Connection refused
```

解决办法

步骤1 网络问题，联系网络工程师处理。

步骤2 网络问题解决之后重新备份即可。

----结束

8.6 DBService 状态正常，组件无法连接 DBService

问题背景与现象

上层组件连接DBService失败，检查DBService组件状态正常，两个实例状态也正常。

图 8-7 DBService 状态

Role	Host Name	OM IP	Business IP	Rack	Operating Status	Health Status	Configuration Status
DBServer(Active)	189-120-85-102	[redacted]	[redacted]	/default/rack0	Started	Good	Synchronized
DBServer(Standby)	189-120-85-141	[redacted]	[redacted]	/default/rack0	Started	Good	Synchronized

原因分析

1. 上层组件是通过dbservice.floatip连接的DBService。
2. 在DBServer所在节点执行命令netstat -anp | grep 20051，发现DBService的Gauss进程在启动时并未绑定floatip，只监测了127.0.0.1的本地IP。

解决办法

步骤1 重新启动DBService服务。

步骤2 启动完成之后在主DBServer节点执行**netstat -anp | grep 20051**命令检查是否绑定了dbservice.floatip。

----结束

8.7 DBServer 启动失败

问题背景与现象

DBService组件启动失败，重启还是失败，实例状态一直为正在恢复状态。

图 8-8 DBService 的状态

角色	主机名	管理IP	业务IP	机架	操作状态	健康状态
DBServer(主)	node-master2JOc8	192.168.5.133	192.168.5.133	/default/rack9bdf	已启动	良好
DBServer(备)	node-master1DEdJ	192.168.5.42	192.168.5.42	/default/rack9bdf	已启动	恢复中

原因分析

1. 查看DBService的日志/var/log/Bigdata/dbservice/DB/gs_ctl-current.log，报如下错误。

```
OCATION:  PostmasterMain, postmaster.c:798
OG:  Starting SelectConfigFiles (postmaster.c:1049)
2017-09-23 15:19:03.591 CST]  gaussmaster 922216  LOG:  Starting checkDataDir (postmaster.c:1068)
2017-09-23 15:19:03.591 CST]  gaussmaster 922216  LOG:  Starting ChangeToDataDir (postmaster.c:1074)
2017-09-23 15:19:03.591 CST]  gaussmaster 922216  LOG:  Starting CheckDateTokenTables (postmaster.c:1120)
2017-09-23 15:19:03.591 CST]  gaussmaster 922216  LOG:  Starting CreateDataDirLockFile (postmaster.c:1151)
2017-09-23 15:19:03.596 CST]  gaussmaster 922216  LOG:  Starting pgaudit_agent_init (postmaster.c:1169)
2017-09-23 15:19:03.596 CST]  gaussmaster 922216  LOG:  Starting process_shared_preload_libraries (postmaster.c:1178)
2017-09-23 15:19:03.597 CST]  gaussmaster 922216  LOG:  could not bind IPv4 socket at the 0 time: ??????????? (pqcomm.c:562)
2017-09-23 15:19:03.597 CST]  gaussmaster 922216  HINT:  Is another postmaster already running on port 20051? If not, wait a few seconds and retry.
2017-09-23 15:19:03.698 CST]  gaussmaster 922216  LOG:  could not bind IPv4 socket at the 1 time: ??????????? (pqcomm.c:562)
2017-09-23 15:19:03.698 CST]  gaussmaster 922216  HINT:  Is another postmaster already running on port 20051? If not, wait a few seconds and retry.
2017-09-23 15:19:03.798 CST]  gaussmaster 922216  LOG:  could not bind IPv4 socket at the 2 time: ??????????? (pqcomm.c:562)
2017-09-23 15:19:03.798 CST]  gaussmaster 922216  HINT:  Is another postmaster already running on port 20051? If not, wait a few seconds and retry.
2017-09-23 15:19:03.898 CST]  gaussmaster 922216  WARNING:  could not create listen socket for "192.168.5.162" (postmaster.c:1235)
2017-09-23 15:19:03.898 CST]  gaussmaster 922216  LOG:  discard audit data: could not create lock file "/tmp/.s.PGSQL.20051.lock": ???? (pgaudit.c:1961)
2017-09-23 15:19:03.898 CST]  gaussmaster 922216  FATAL:  could not create lock file "/tmp/.s.PGSQL.20051.lock": ???? (miscinit.c:854)
```

2. 检查发现/tmp权限不正确，正确的权限应该为777。

```
mm@hadoopc1h2:/var/log/Bigdata/dbservice/DB> ll /
otal 100
drwxr-xr-x   2 root   root    4096 Aug  6  2016 bin
drwxr-xr-x   3 root   root    4096 Aug  6  2016 boot
drwxr-xr-x  17 root   root    5080 Sep 20 11:30 dev
drwxr-xr-x   3 httpd  common     0 Sep 20 11:20 ecmramfs
drwxr-xr-x  71 root   root    4096 Sep 22 02:40 etc
-rw-r-----   1 root   root       0 Sep 11 08:25 fsck_corrected_
drwxr-xr-x   9 root   root    4096 Sep 18 14:39 home
drwxr-xr-x  12 root   root    4096 Sep 14  2016 lib
drwxr-xr-x   8 root   root   12288 Sep 14  2016 lib64
drwx------   2 root   root   16384 Aug  7  2016 lost+found
drwxr-xr-x   2 root   root    4096 May  5  2010 media
drwxr-xr-x   2 root   root    4096 May  5  2010 mnt
drwxr-xr-x  19 root   root    4096 Jun 30 10:04 opt
dr-xr-xr-x 424 root   root       0 Sep 20 19:18 proc
drwx------   5 root   root    4096 Sep 23 10:21 root
drwxrwxr-x   4 root   root    4096 Aug  7  2016 rrdtool
drwxr-xr-x   3 root   root   12288 Sep 14  2016 sbin
drwxr-xr-x   2 root   root    4096 May  5  2010 selinux
drwxrwxrwx  10 root   root    4096 Nov 15  2016 srv
drwxr-xr-x  12 root   root       0 Sep 20 11:19 sys
lrwxrwxrwx   1 root   root       1 Aug  7  2016 target -> /
drwxr-xr-x   6 root   root    4096 Sep 23 15:19 tmp
drwxr-xr-x  13 root   root    4096 Apr 22  2014 usr
```

解决办法

步骤1 修改/tmp权限为777。

步骤2 重新启动DBService组件。

----结束

8.8 浮动 IP 不通导致 DBService 备份失败

问题背景与现象

在默认备份default中DBService备份失败，其他备份（NameNode、LdapServer、OMS备份）成功。

原因分析

1. 查看DBService的备份页面错误信息，有如下错误信息提示：

```
Clear temporary files at backup checkpoint DBService_test_DBService_DBService_20180326155921
that fialed last time.
Temporyary files at backup checkpoint DBService_test_DBService_DBService20180326155921 that
failed last time are cleared successfully.
```

```
Start executing the backup task.
The backup of configuration DBService is started.
Check the backup available disk space.
Backup initialization succeeded for configuration DBService.
Clear temporary files at backup checkpoint DBService_test_DBService_DBService_20180326155921 that failed last time.
Temporary files at backup checkpoint DBService_test_DBService_DBService_20180326155921 that failed last time are cleared successfully.
Checkpoint DBService_test_DBService_DBService_20180326162235 is verified successfully before backup.
Temporary files are cleared successfully before backup checkpoint DBService_test_DBService_DBService_20180326162235.
Prestart backup succeeded for checkpoint DBService_test_DBService_DBService_20180326162235.
The snapshot is created successfully for checkpoint DBService_test_DBService_DBService_20180326162235 before backup.
Backup is being performed for checkpoint DBService_test_DBService_DBService_20180326162235.
Backup execution failed. Task ID: 2
Detail: DBService backup task failed, please view details in logs.
Temporary files are cleared successfully after backup checkpoint DBService_test_DBService_DBService_20180326162235.
checkpoint DBService_test_DBService_DBService_20180326162235 is deleted successfully after backup failure.
Failed to backup configuration DBService.
```

2. 查看/var/log/Bigdata/dbservice/scriptlog/backup.log文件，发现日志停止打印，并没有备份相关信息。
3. 查看主OMS节点 /var/log/Bigdata/controller/backupplugin.log日志发现如下错误信息：

```
result error is ssh:connect to host 172.16.4.200 port 22 : Connection refused (172.16.4.200是DBService的浮动IP)
DBService backup failed.
```

```
2018-03-27 07:00:35,758 INFO  [pool-1-thread-5] Create adapter from com.huawei.bigdata.om.backup.MetadataPluginAdapter success.
com.huawei.bigdata.om.backup.plugin.AbstractBackupRecoveryPlugin.initializePluginAdapter(AbstractBackupRecoveryPlugin.java:92)
2018-03-27 07:00:35,759 INFO  [pool-1-thread-5] floatIp is 172.16.4.200. com.huawei.bigdata.om.dbservice.backup.BackupRecoveryPlugin.getFloatIp(BackupRecoveryPlugin.java:233)
2018-03-27 07:00:35,759 INFO  [pool-1-thread-5] cmd is ssh 172.16.4.200 /opt/huawei/Bigdata/FusionInsight_V100R002C60U20/dbservice/sbin/dbservice_backup.sh -b -d
/srv/BigData/LocalBackup/default_20180326213206/DBService_20180327070010. com.huawei.bigdata.om.dbservice.backup.BackupRecoveryPlugin.startBackup(BackupRecoveryPlugin.java:166)
2018-03-27 07:00:35,759 INFO  [pool-1-thread-5] create task taskId is 6. com.huawei.bigdata.om.dbservice.backup.BackupRecoveryPlugin.startBackup(BackupRecoveryPlugin.java:169)
2018-03-27 07:00:35,760 INFO  [pool-1-thread-5] startBackup result OperateResult(errorCode:RUNNING, result:6, detailInfo:, packageName:null).
com.huawei.bigdata.om.backup.BackupPluginContainerHandler.startBackup(BackupPluginContainerHandler.java:246)
2018-03-27 07:00:35,760 INFO  [Thread-132] Executing the command with arguments and env, timeout: 900000
com.huawei.bigdata.om.controller.api.extern.monitor.script.LinuxScriptExecutionHandler.logMessage(LinuxScriptExecutionHandler.java:64)
2018-03-27 07:00:35,863 INFO  [Thread-132] Execute command : /opt/huawei/Bigdata/om-0.0.1/sbin/scriptLauncher.sh ssh 172.16.4.200
/opt/huawei/Bigdata/FusionInsight_V100R002C60U20/dbservice/sbin/dbservice_backup.sh -b -d /srv/BigData/LocalBackup/default_20180326213206/DBService_20180327070010.
com.huawei.bigdata.om.dbservice.backup.BackupTask.run(BackupTask.java:48)
2018-03-27 07:00:35,863 INFO  [Thread-132] result status is 255. com.huawei.bigdata.om.dbservice.backup.BackupTask.run(BackupTask.java:49)
2018-03-27 07:00:35,863 INFO  [Thread-132] result output is . com.huawei.bigdata.om.dbservice.backup.BackupTask.run(BackupTask.java:50)
2018-03-27 07:00:35,863 INFO  [Thread-132] result erro is ssh: connect to host 172.16.4.200 port 22: Connection refused
. com.huawei.bigdata.om.dbservice.backup.BackupTask.run(BackupTask.java:51)
2018-03-27 07:00:35,863 ERROR [Thread-132] DBService backup failed. com.huawei.bigdata.om.dbservice.backup.BackupTask.run(BackupTask.java:64)
2018-03-27 07:00:40,868 INFO  [pool-1-thread-5] query backup taskId is 6. com.huawei.bigdata.om.dbservice.backup.BackupRecoveryPlugin.getBackupProgress(BackupRecoveryPlugin.java:247)
```

解决办法

步骤1 登录DBService主节点（绑定有DBService浮动IP的master节点）。

```
[root@node-master1cuEb ~]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.223  netmaDBService主节点oadcast 192.168.2.255
        ether fa:16:3e:eb:7e:74  txqueuelen 1000  (Ethernet)
        RX packets 125672126  bytes 35833339919 (33.3 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 111023825  bytes 33326544401 (31.0 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0:DBS: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.206  netmask 255.255.255.0  broadcast 192.168.2.255
        ether fa:16:3e:eb:7e:74  txqueuelen 1000  (Ethernet)

eth0:FI_HUE: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.197  netmask 255.255.255.0  broadcast 192.168.2.255
        ether fa:16:3e:eb:7e:74  txqueuelen 1000  (Ethernet)
```

步骤2 检查 /etc/ssh/sshd_config文件中ListenAddress配置项，添加DBService浮动IP到ListenAdress或者注释掉ListenAdress配置项。

步骤3 执行如下命令重启sshd服务。

service sshd restart

步骤4 观察下次备份DBServie是否备份成功。

----结束

8.9 DBService 配置文件丢失导致启动失败

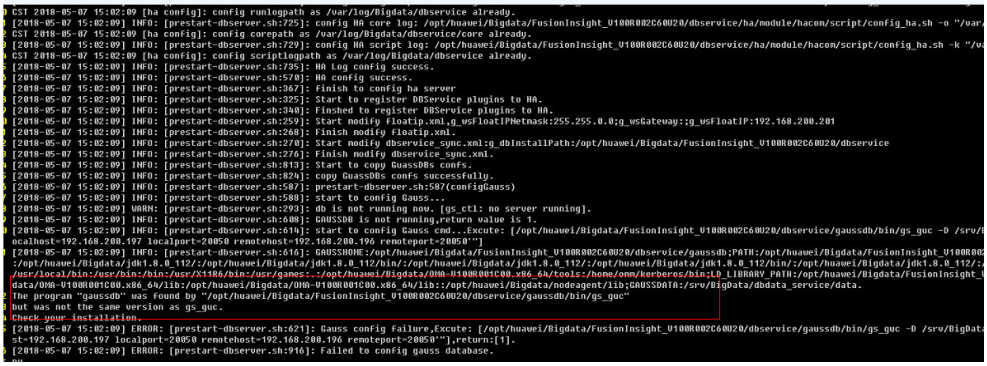
问题背景与现象

节点异常下电，重启备DBService失败。

原因分析

1. 查看/var/log/Bigdata/dbservice/DB/gaussdb.log日志没有内容。
2. 查看/var/log/Bigdata/dbservice/scriptlog/preStartDBService.log日志，发现如下信息，判断为配置信息丢失。

```
The program "gaussdb" was found by "
/opt/Bigdata/MRS_xxx/install/dbservice/gaussdb/bin/gs_guc)
But not was not the same version as gs_guc.
Check your installation.
```



3. 比对主备DBServer节点/srv/BigData/dbdata_service/data目录下的配置文件发现差距比较大。

```
omm@hadoopc1h3:/srv/BigData/dbdata_service/data> ll
total 128
-rw------- 1 omm wheel     4 May  8 09:54 PG_VERSION
drwx------ 2 omm wheel  4096 May  8 09:54 bak
drwx------ 7 omm wheel  4096 May  8 09:54 base
-rw------- 1 omm wheel   922 May  8 09:54 dblink.conf
-rw------- 1 omm wheel    16 May  8 09:59 gaussdb.state
drwx------ 2 omm wheel  4096 May  8 09:58 global
drwx------ 2 omm wheel  4096 May  8 09:54 pg_audit
drwx------ 2 omm wheel  4096 May  8 09:58 pg_blackbox
drwx------ 2 omm wheel  4096 May  8 09:54 pg_clog
drwx------ 2 omm wheel  4096 May  8 09:58 pg_confile_backup
-rw------- 1 omm wheel     0 May  8 09:54 pg_ctl.lock
-rw------- 1 omm wheel  4287 May 18  2017 pg_hba.conf
-rw------- 1 omm wheel  1024 May  8 09:54 pg_hba.conf.lock
-rw------- 1 omm wheel  1636 May  8 09:54 pg_ident.conf
drwx------ 2 omm wheel  4096 May  8 09:54 pg_log
drwx------ 4 omm wheel  4096 May  8 09:54 pg_multixact
drwx------ 2 omm wheel  4096 May  8 09:58 pg_notify
drwx------ 2 omm wheel  4096 May  8 09:54 pg_serial
drwx------ 2 omm wheel  4096 May  8 09:54 pg_snapshots
drwx------ 2 omm wheel  4096 May  8 09:58 pg_stat_tmp
drwx------ 2 omm wheel  4096 May  8 09:54 pg_subtrans
drwx------ 2 omm wheel  4096 May  8 09:54 pg_tblspc
drwx------ 2 omm wheel  4096 May  8 09:54 pg_twophase
drwx------ 2 omm wheel  4096 May  8 09:54 pg_wallet
drwx------ 3 omm wheel  4096 May  8 09:54 pg_xlog
-rw------- 1 omm wheel 15277 May  8 09:59 postgresql.conf
-rw------- 1 omm wheel  1024 May  8 09:54 postgresql.conf.lock
-rw------- 1 omm wheel   134 May  8 09:59 postmaster.opts
-rw------- 1 omm wheel   127 May  8 09:58 postmaster.pid
```

```
omm@hadoopc1h3:/srv/BigData/dbdata_service> cd data_bak/
omm@hadoopc1h3:/srv/BigData/dbdata_service/data_bak> ll
total 64
-rw------- 1 omm wheel   202 Feb 11 10:43 backup_label
-rw------- 1 omm wheel     0 Feb 11 10:42 build_completed.start
-rw------- 1 omm wheel    16 Apr 28 17:32 gaussdb.state
-rw------- 1 omm wheel     7 Apr 28 17:32 gs_build.pid
drwx------ 2 omm wheel  4096 Feb 11 10:44 pg_audit
drwx------ 2 omm wheel  4096 Feb 11 10:41 pg_blackbox
drwx------ 2 omm wheel  4096 Feb 11 10:09 pg_confile_backup
-rw------- 1 omm wheel     0 Apr 28 17:32 pg_ctl.lock
-rw------- 1 omm wheel  4287 May 18  2017 pg_hba.conf
drwx------ 2 omm wheel  4096 Feb 11 10:43 pg_notify
drwx------ 2 omm wheel  4096 Feb 11 10:43 pg_xlog
-rw------- 1 omm wheel 15155 May  7 15:33 postgresql.conf
-rw------- 1 omm wheel  1024 May  7 15:33 postgresql.conf.lock
-rw------- 1 omm wheel   134 Feb 11 10:42 postmaster.opts
```

解决办法

步骤1 把主节点**/srv/BigData/dbdata_service/data**的内容拷贝到备节点，保持文件权限和属组与主节点一样。

步骤2 修改postgresql.conf配置信息，localhost修改成本节点IP，remotehost修改成对端节点IP。

```
#------------------------------------------------------------------------------
# CUSTOMIZED OPTIONS
#------------------------------------------------------------------------------

# Add settings for extensions here
max_files_per_process = 300
unix_socket_directory = '/var/run/FusionInsight-DBService'
replconninfo1 = 'localhost=192.168.200.197 localport=20050 remotehost=192.168.200.196 remoteport=20050'
"postgresql.conf" 382L, 15277C
```

步骤3 登录Manager页面重启备DBServer节点。

----结束

9 使用 Flink

9.1 Flink 客户端执行命令报错“Error while parsing YAML configuration file : security.kerberos.login.keytab”

问题背景与现象

客户端安装成功，执行客户端命令例如yarn-session.sh时报错，提示如下：

```
[root@host01 bin]# yarn-session.sh
2018-10-25 01:22:06,454 | ERROR | [main] | Error while trying to split key and value in configuration
file /opt/flinkclient/Flink/flink/conf/flink-conf.yaml:80: "security.kerberos.login.keytab: " |
org.apache.flink.configuration.GlobalConfiguration (GlobalConfiguration.java:160)
Exception in thread "main" org.apache.flink.configuration.IllegalConfigurationException: Error while
parsing YAML configuration file :80: "security.kerberos.login.keytab: "
```

原因分析

在安全集群环境下，Flink需要进行安全认证。当前客户端未进行相关安全认证设置。

1. Flink整个系统有两种认证方式：
   - 使用kerberos认证：Flink yarn client、Yarn Resource Manager、JobManager、HDFS、TaskManager、Kafka和Zookeeper。
   - 使用YARN内部的认证机制：Yarn Resource Manager与Application Master（简称AM）。
2. 如果用户安装安全集群需要使用kerberos认证和security cookie认证。根据日志提示，发现配置文件中“security.kerberos.login.keytab :”配置项错误，未进行安全配置。

解决办法

步骤1 从MRS上下载用户的keytab认证文件，并放置到Flink客户端所在节点的某个目录下。

步骤2 在“flink-conf.yaml”文件中配置：

1. keytab路径。
   ```
   security.kerberos.login.keytab: /home/flinkuser/keytab/abc222.keytab
   ```

说明

- “/home/flinkuser/keytab/abc222.keytab”表示的是用户目录，为**步骤1**中放置目录。
- 请确保客户端用户具备对应目录权限。

2. principal名。
```
security.kerberos.login.principal: abc222
```
3. 对于HA模式，如果配置了ZooKeeper，还需要设置ZooKeeper Kerberos认证相关的配置。
```
zookeeper.sasl.disable: false
security.kerberos.login.contexts: Client
```
4. 如果用户对于Kafka Client和Kafka Broker之间也需要做Kerberos认证，配置如下：
```
security.kerberos.login.contexts: Client,KafkaClient
```

----结束

9.2 Flink 客户端执行命令报错“Error while parsing YAML configuration file : security.kerberos.login.principal:pippo”

问题背景与现象

客户端安装成功，执行客户端命令例如yarn-session.sh时报错，提示如下：

```
[root@host01 bin]# yarn-session.sh
2018-10-25 19:27:01,397 | ERROR | [main] | Error while trying to split key and value in configuration
file /opt/flinkclient/Flink/flink/conf/flink-conf.yaml:81: "security.kerberos.login.principal:pippo " |
org.apache.flink.configuration.GlobalConfiguration (GlobalConfiguration.java:160)
Exception in thread "main" org.apache.flink.configuration.IllegalConfigurationException: Error while
parsing YAML configuration file :81: "security.kerberos.login.principal:pippo "
    at org.apache.flink.configuration.GlobalConfiguration.loadYAMLResource(GlobalConfiguration.java:161)
    at org.apache.flink.configuration.GlobalConfiguration.loadConfiguration(GlobalConfiguration.java:112)
    at org.apache.flink.configuration.GlobalConfiguration.loadConfiguration(GlobalConfiguration.java:79)
    at org.apache.flink.yarn.cli.FlinkYarnSessionCli.main(FlinkYarnSessionCli.java:482)
```

原因分析

配置文件“flink-conf.yaml”中配置项格式错误。

```
security.kerberos.login.principal:pippo
```

解决办法

修改“flink-conf.yaml”中配置，配置项名称和值之间存在一个空格。

```
security.kerberos.login.principal: pippo
```

9.3 Flink 客户端执行命令报错“Could not connect to the leading JobManager”

问题背景与现象

创建Fllink集群，执行yarn-session.sh命令卡住一段时间后报错：

```
2018-09-20 22:51:16,842 | WARN  | [main] | Unable to get ClusterClient status from Application Client | org.apache.flink.yarn.YarnClusterClient (YarnClusterClient.java:253)
org.apache.flink.util.FlinkException: Could not connect to the leading JobManager. Please check that the JobManager is running.
    at org.apache.flink.client.program.ClusterClient.getJobManagerGateway(ClusterClient.java:861)
    at org.apache.flink.yarn.YarnClusterClient.getClusterStatus(YarnClusterClient.java:248)
    at org.apache.flink.yarn.YarnClusterClient.waitForClusterToBeReady(YarnClusterClient.java:516)
    at org.apache.flink.yarn.cli.FlinkYarnSessionCli.run(FlinkYarnSessionCli.java:717)
    at org.apache.flink.yarn.cli.FlinkYarnSessionCli$1.call(FlinkYarnSessionCli.java:514)
    at org.apache.flink.yarn.cli.FlinkYarnSessionCli$1.call(FlinkYarnSessionCli.java:511)
    at java.security.AccessController.doPrivileged(Native Method)
    at javax.security.auth.Subject.doAs(Subject.java:422)
    at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1729)
    at org.apache.flink.runtime.security.HadoopSecurityContext.runSecured(HadoopSecurityContext.java:41)
    at org.apache.flink.yarn.cli.FlinkYarnSessionCli.main(FlinkYarnSessionCli.java:511)
Caused by: org.apache.flink.runtime.leaderretrieval.LeaderRetrievalException: Could not retrieve the leader gateway.
    at org.apache.flink.runtime.util.LeaderRetrievalUtils.retrieveLeaderGateway(LeaderRetrievalUtils.java:79)
    at org.apache.flink.client.program.ClusterClient.getJobManagerGateway(ClusterClient.java:856)
    ... 10 common frames omitted
Caused by: java.util.concurrent.TimeoutException: Futures timed out after [10000 milliseconds]
```

可能原因

Flink开启了SSL通信加密，却没有正确的配置SSL证书。

解决办法

针对MRS 2.x及之前版本，操作如下：

方法1：

关闭Flink SSL通信加密，修改客户端配置文件“conf/flink-conf.yaml”。

```
security.ssl.internal.enabled: false
```

方法2：

开启Flink SSL通信加密，security.ssl.internal.enabled保持默认。

正确配置SSL：

- 配置keystore或truststore文件路径为**相对路径**时，Flink Client执行命令的目录需要可以直接访问该相对路径。

  ```
  security.ssl.internal.keystore: ssl/flink.keystore
  security.ssl.internal.truststore: ssl/flink.truststore
  ```

  在Flink的CLI yarn-session.sh命令中增加“-t”选项来传输keystore和truststore文件到各个执行节点。

  yarn-session.sh -t ssl/ 2

- 配置keystore或truststore文件路径为**绝对路径**时，需要在Flink Client以及各个节点的该绝对路径上放置keystore或truststore文件。

```
security.ssl.internal.keystore: /opt/client/Flink/flink/conf/flink.keystore
security.ssl.internal.truststore: /opt/client/Flink/flink/conf/flink.truststore
```

针对MRS3.x及之后版本，操作如下：

方法1：

关闭Flink SSL通信加密，修改客户端配置文件“conf/flink-conf.yaml”。

```
security.ssl.enabled: false
```

方法2：

开启Flink SSL通信加密，security.ssl.enabled 保持默认。

正确配置SSL：

- 配置keystore或truststore文件路径为**相对路径**时，Flink Client执行命令的目录需要可以直接访问该相对路径。

  ```
  security.ssl.keystore: ssl/flink.keystore
  security.ssl.truststore: ssl/flink.truststore
  ```

  在Flink的CLI yarn-session.sh命令中增加“-t”选项来传输keystore和truststore文件到各个执行节点。

  ```
  yarn-session.sh -t ssl/ 2
  ```

- 配置keystore或truststore文件路径为**绝对路径**时，需要在Flink Client以及各个节点的该绝对路径上放置keystore或truststore文件。

  ```
  security.ssl.keystore: /opt/client/Flink/flink/conf/flink.keystore
  security.ssl.truststore: /opt/client/Flink/flink/conf/flink.truststore
  ```

9.4 使用不同用户执行 yarn-session 创建 Flink 集群失败

问题背景与现象

使用Flink过程中，具有两个相同权限用户testuser和bdpuser。

使用用户testuser创建Flink集群正常，但是切换至bdpuser用户创建Fllink集群时，执行yarn-session.sh命令报错：

```
2019-01-02 14:28:09,098 | ERROR | [main] | Ensure path threw exception |
org.apache.flink.shaded.curator.org.apache.curator.framework.imps.CuratorFrameworkImpl
(CuratorFrameworkImpl.java:566)
org.apache.flink.shaded.zookeeper.org.apache.zookeeper.KeeperException$NoAuthException:
KeeperErrorCode = NoAuth for /flink/application_1545397824912_0022
```

可能原因

高可用配置项未修改。

由于在Flink的配置文件中，“high-availability.zookeeper.client.acl”默认为“creator”，仅创建者有权限访问，新用户无法访问ZooKeeper上的目录导致yarn-session.sh执行失败。

解决办法

步骤1 修改客户端配置文件“conf/flink-conf.yaml”中配置项“high-availability.zookeeper.path.root”，例如：

```
high-availability.zookeeper.path.root:  flink2
```

步骤2 重新提交Flink任务。

----结束

9.5 Flink 业务程序无法读取 NFS 盘上的文件

用户问题

Flink业务程序无法读取集群节点挂载的NFS盘上的文件。

问题现象

用户开发的Flink业务程序中需要读取用户定义的配置文件，该配置文件放在NFS盘上，NFS盘是挂载在集群节点上的，集群的所有节点均可以访问该盘。用户提交Flink程序后，业务代码访问不到客户自定义的配置文件，导致业务程序启动失败。

原因分析

该问题的根因是NFS盘上的根目录权限不足，导致Flink程序启动后无法访问该目录。

MRS的Flink任务是在YARN运行，当集群未启用Kerberos认证时，在YARN上运行任务的用户为yarn_user。用户自定义的配置文件如果在任务启动之后使用，则文件以及文件的父目录（NFS上的文件所在的父目录，非集群节点上的软连接），必须允许yarn_user可以访问，否则程序中无法获取文件内容。当集群为启用Kerberos认证的集群时，则文件的权限必须允许提交程序的用户访问。

处理步骤

步骤1 以**root**用户登录集群的Master节点。

步骤2 执行如下命令查看用户自定义配置文件所在父目录的权限。

ll *<文件所在路径的父目录路径>*

步骤3 进入NFS盘待访问文件所在目录，修改用户自定义配置文件所在父目录的权限为755。

chmod 755 -R /*<文件所在路径的父目录路径>*

步骤4 确认Core或者Task节点是否可以访问到该配置文件。

1. 以**root**用户登录Core/Task节点。
   如果当前集群已启用Kerberos认证，请以**root**用户登录Core节点。
2. 执行**su - yarn_user**命令切换到yarn_user用户。
   如果当前集群已启用Kerberos认证，请执行 **su -** *提交作业的用户*命令切换用户。
3. 执行如下命令查看用户权限，文件所在路径请使用该文件的绝对路径。
   ll *<文件所在路径>*

----结束

建议与总结

当用户提交的任务中要访问自定义的配置文件时，特别是挂载NFS盘时，除了确认文件的权限之外，还要确认文件所在父目录的权限是否正确。NFS盘挂载到MRS集群节点上，一般会新建软连接到NFS目录，这个时候需要查看NFS上的目录权限是否正确。

9.6 自定义 Flink log4j 日志输出级别不生效

用户问题

MRS 3.1.0集群自定义Flink log4j日志级别不生效。

问题现象

1. 在使用MRS 3.1.0集群Flink数据分析时，将“$Flink_HOME/conf”目录下的“log4j.properties”文件中日志级别修改为INFO级别日志。
2. 任务正常提交后，运行控制台未打印出INFO级别日志，输出的日志级别还是ERROR级别。

原因分析

修改“$Flink_HOME/conf”目录下的“log4j.properties”文件，控制的是JobManager和TaskManager的算子内的日志输出，输出的日志会打印到对应的yarn contain中，可以在Yarn WebUI查看对应日志。

MRS 3.1.0及之后版本的Flink 1.12.0版本开始默认的日志框架是log4j2，配置的方式跟之前log4j的方式有区别，使用如log4j日志规则不会生效。

处理步骤

Log4j2详细日志规格配置参考开源官方文档：**http://logging.apache.org/log4j/2.x/manual/configuration.html#Properties**。

10 使用 Flume

10.1 Flume 向 Spark Streaming 提交作业后报类找不到错误

用户问题

Flume向Spark Streaming提交作业，提交到集群后报类找不到的错误。

问题现象

Spark Streaming代码打成jar包提交到集群后报类找不到错误，通过以下两种方式依然不生效。

1. 在提交Spark作业的时候使用**--jars** 命令引用类所在的jar包。
2. 将类所在的jar包引入Spark Streaming的jar包。

原因分析

执行Spark作业时无法加载部分jar，导致找不到class。

处理步骤

步骤1 使用 **--jars** 加载flume-ng-sdk-{version}.jar依赖包。

步骤2 同时修改“spark-default.conf”中两个配置项。

spark.driver.extraClassPath=$PWD/*:{加上原来配置的值}

spark.executor.extraClassPath =$PWD/*

步骤3 作业运行成功。如果还有报错，则需要排查还有哪个jar没有加载，再次执行步骤1和步骤2。

----结束

10.2 Flume 客户端安装失败

问题现象

安装Flume客户端失败，提示JAVA_HOME is null或flume has been installed。

```
CST 2016-08-31 17:02:51 [flume-client install]: JAVA_HOME is null in current user,please install the JDK and set the JAVA_HOME
CST 2016-08-31 17:02:51 [flume-client install]: check environment failed.
CST 2016-08-31 17:02:51 [flume-client install]: check param failed.
CST 2016-08-31 17:02:51 [flume-client install]: install flume client failed.
```

```
CST 2016-08-31 17:03:58 [flume-client install]: flume has been installed
CST 2016-08-31 17:03:58 [flume-client install]: check path failed.
CST 2016-08-31 17:03:58 [flume-client install]: check param failed.
CST 2016-08-31 17:03:58 [flume-client install]: install flume client failed.
```

原因分析

- Flume客户端安装时会检查环境变量，如果没有可用的JAVA，会报错并退出安装。
- 指定的目录下已经安装有Flume客户端。

解决办法

步骤1 如果报JAVA_HOME is null错误，需要使用命令：

export JAVA_HOME=*java路径*

设置JAVA_HOME，重新运行安装脚本。

步骤2 如果指定的目录下已经安装有Flume客户端，需要先卸载已经存在的Flume客户端，或指定其他目录安装。

----结束

10.3 Flume 客户端无法连接服务端

问题现象

安装Flume客户端并设置Avro Sink与服务端通信，发现无法连接Flume服务端。

原因分析

1. 服务端配置错误，监测端口启动失败，例如服务端Avro Source配置了错误的IP，或者已经被占用了的端口。

   查看Flume运行日志：

   ```
   2016-08-31 17:28:42,092 | ERROR | [lifecycleSupervisor-1-9] |  Unable to start EventDrivenSourceRunner: { source:Avro source avro_source: { bindAddress: 10.120.205.7, port: 21154 } } - Exception follows.  | org.apache.flume.lifecycle.LifecycleSupervisor $MonitorRunnable.run(LifecycleSupervisor.java:253)
   java.lang.RuntimeException: org.jboss.netty.channel.ChannelException: Failed to bind to: /192.168.205.7:21154
   ```

2. 若采用了加密传输，证书或密码错误。

```
2016-08-31 17:15:59,593 | ERROR | [conf-file-poller-0] |  Source avro_source has been removed due to an error during configuration  | org.apache.flume.node.AbstractConfigurationProvider.loadSources(AbstractConfigurationProvider.java:388)
org.apache.flume.FlumeException: Avro source configured with invalid keystore: /opt/Bigdata/MRS_XXX/install/FusionInsight-Flume-1.9.0/flume/conf/flume_sChat.jks
```

3. 客户端与服务端通信异常。

```
PING 192.168.85.55 (10.120.85.55) 56(84) bytes of data.
From 192.168.85.50 icmp_seq=1 Destination Host Unreachable
From 192.168.85.50 icmp_seq=2 Destination Host Unreachable
From 192.168.85.50 icmp_seq=3 Destination Host Unreachable
From 192.168.85.50 icmp_seq=4 Destination Host Unreachable
```

解决办法

步骤1 设置为正确的IP，必须为本机的IP，如果端口被占用，重新配置一个空闲的端口。

步骤2 配置正确的证书路径。

步骤3 联系网络管理员，恢复网络。

----结束

10.4 Flume 数据写入组件失败

问题现象

Flume进程启动后，Flume数据无法写入到对应组件（本示例以服务端写入到HDFS为例）。

原因分析

1. HDFS未启动或故障。

   查看Flume运行日志：

```
2019-02-26 11:16:33,564 | ERROR | [SinkRunner-PollingRunner-DefaultSinkProcessor] |  opreation the hdfs file errors.  | org.apache.flume.sink.hdfs.HDFSEventSink.process(HDFSEventSink.java:414)
2019-02-26 11:16:33,747 | WARN  | [hdfs-CCCC-call-runner-4] |  A failover has occurred since the start of call #32795 ClientNamenodeProtocolTranslatorPB.getFileInfo over 192-168-13-88/192.168.13.88:25000  | org.apache.hadoop.io.retry.RetryInvocationHandler$ProxyDescriptor.failover(RetryInvocationHandler.java:220)
2019-02-26 11:16:33,748 | ERROR | [hdfs-CCCC-call-runner-4] |  execute hdfs error. {}  | org.apache.flume.sink.hdfs.HDFSEventSink$3.call(HDFSEventSink.java:744)
java.net.ConnectException: Call From 192-168-12-221/192.168.12.221 to 192-168-13-88:25000 failed on connection exception: java.net.ConnectException: Connection refused; For more details see:  http://wiki.apache.org/hadoop/ConnectionRefused
```

2. HDFS Sink未启动。

   查看Flume运行日志，发现“ flume current metrics”中并没有Sink信息：

```
2019-02-26 11:46:05,501 | INFO  | [pool-22-thread-1] |  flume current metrics:{"CHANNEL.BBBB":{"ChannelCapacity":"10000","ChannelFillPercentage":"0.0","Type":"CHANNEL","ChannelStoreSize":"0","EventProcessTimedelta":"0","EventTakeSuccessCount":"0","ChannelSize":"0","EventTakeAttemptCount":"0","StartTime":"1551152734999","EventPutAttemptCount":"0","EventPutSuccessCount":"0","StopTime":"0"},"SOURCE.AAAA":{"AppendBatchAcceptedCount":"0","EventAcceptedCount":"0","AppendReceivedCount":"0","MonTime":"0","StartTime":"1551152735503","AppendBatchReceivedCount":"0","EventReceivedCount":"0","Type":"SOURCE","TotalFilesCount":"1001","SizeAcceptedCount":"0","UpdateTime":"605410241202740","AppendAcceptedCount":"0","OpenConnectionCount":"0","MovedFilesCount":"1001","StopTime":"0"}}  | org.apache.flume.node.Application.getRestartComps(Application.java:467)
```

解决办法

步骤1 若Flume数据写入的组件未启动，启动对应组件；若组件异常，请联系服务技术支持。

步骤2 Sink未启动，检查配置文件是否配置正确，若配置错误，则正确修改配置文件后重启Flume进程，如果配置正确，则查看日志错误信息，根据具体错误信息制定解决办法。

----结束

10.5 Flume 服务端进程故障

问题现象

Flume运行一段时间后，Manager界面Flume实例显示运行状态“故障”。

原因分析

Flume文件或文件夹权限异常，重启后Manager界面提示如下信息：

```
[2019-02-26 13:38:02]RoleInstance prepare to start failure [{ScriptExecutionResult=ScriptExecutionResult [exitCode=126, output=, errMsg=sh: line 1: /opt/Bigdata/MRS_XXX/install/FusionInsight-Flume-1.9.0/flume/bin/flume-manage.sh: Permission denied
```

解决办法

与运行正常的Flume节点进行文件和文件夹权限对比，更改错误文件或文件夹权限。

10.6 Flume 数据采集慢

问题现象

Flume启动后，Flume数据采集慢。

原因分析

1. Flume堆内存设置不合理，导致Flume进程一直处于频繁GC。查看Flume运行日志：
```
2019-02-26T13:06:20.666+0800: 1085673.512: [Full GC:[CMS: 3849339k->3843458K(3853568K), 2.5817610 secs] 4153654K->3843458K(4160256K), [CMS Perm : 27335K->27335K(45592K),2.5820080 SECS] [Times: user=2.63, sys0.00, real=2.59 secs]
```
2. 用户业务配置的Spooldir source的deletePolicy策略是立即删除（immediate）。

解决办法

步骤1 适当调大堆内存（xmx）的值。

步骤2 将Spooldir source的deletePolicy策略更改为永不删除（never）。

----结束

10.7 Flume 启动失败

问题现象

安装Flume服务或重启Flume服务失败。

原因分析

1. Flume堆内存设置的值大于机器剩余内存，查看Flume启动日志：
```
[CST 2019-02-26 13:31:43][INFO] [[checkMemoryValidity:124]] [GC_OPTS is invalid:
Xmx(40960000MB) is bigger than the free memory(56118MB) in system.] [9928]
```
2. Flume文件或文件夹权限异常，界面或后台会提示如下信息：
```
[2019-02-26 13:38:02]RoleInstance prepare to start failure
[{ScriptExecutionResult=ScriptExecutionResult [exitCode=126, output=, errMsg=sh: line 1: /opt/Bigdata/
MRS_XXX/install/FusionInsight-Flume-1.9.0/flume/bin/flume-manage.sh: Permission denied
```
3. JAVA_HOME配置错误，查看Flume agent启动日志：
```
Info: Sourcing environment configuration script /opt/FlumeClient/fusioninsight-flume-1.9.0/conf/
flume-env.sh
+ '[' -n '' ']'
+ exec /tmp/MRS-Client/MRS_Flume_ClientConfig/JDK/jdk-8u18/bin/java '-
XX:OnOutOfMemoryError=bash /opt/FlumeClient/fusioninsight-flume-1.9.0/bin/
out_memory_error.sh /opt/FlumeClient/fusioninsight-flume-1.9.0/conf %p' -Xms2G -Xmx4G -
XX:CMSFullGCsBeforeCompaction=1 -XX:+UseConcMarkSweepGC -XX:+CMSParallelRemarkEnabled -
XX:+UseCMSCompactAtFullCollection -Dkerberos.domain.name=hadoop.hadoop.com -verbose:gc -
XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=10 -XX:GCLogFileSize=1M -XX:+PrintGCDetails -
XX:+PrintGCDateStamps -Xloggc:/var/log/Bigdata//flume-client-1/flume/flume-root-20190226134231-
%p-gc.log -Dproc_org.apache.flume.node.Application -Dproc_name=client -Dproc_conf_file=/opt/
FlumeClient/fusioninsight-flume-1.9.0/conf/properties.properties -Djava.security.krb5.conf=/opt/
FlumeClient/fusioninsight-flume-1.9.0/conf//krb5.conf -Djava.security.auth.login.config=/opt/
FlumeClient/fusioninsight-flume-1.9.0/conf//jaas.conf -Dzookeeper.server.principal=zookeeper/
hadoop.hadoop.com -Dzookeeper.request.timeout=120000 -Dflume.instance.id=884174180 -
Dflume.agent.name=clientName1 -Dflume.role=client -Dlog4j.configuration.watch=true -
Dlog4j.configuration=log4j.properties -Dflume_log_dir=/var/log/Bigdata//flume-client-1/flume/ -
Dflume.service.id=flume-client-1 -Dbeetle.application.home.path=/opt/FlumeClient/fusioninsight-
flume-1.9.0/conf/service -Dflume.called.from.service -Dflume.conf.dir=/opt/FlumeClient/fusioninsight-
flume-1.9.0/conf -Dflume.metric.conf.dir=/opt/FlumeClient/fusioninsight-flume-1.9.0/conf -
Dflume.script.home=/opt/FlumeClient/fusioninsight-flume-1.9.0/bin -cp '/opt/FlumeClient/
fusioninsight-flume-1.9.0/conf:/opt/FlumeClient/fusioninsight-flume-1.9.0/lib/*:/opt/FlumeClient/
fusioninsight-flume-1.9.0/conf/service/' -Djava.library.path=/opt/FlumeClient/fusioninsight-flume-1.9.0/
plugins.d/native/native org.apache.flume.node.Application --conf-file /opt/FlumeClient/fusioninsight-
flume-1.9.0/conf/properties.properties --name client
/opt/FlumeClient/fusioninsight-flume-1.9.0/bin/flume-ng: line 233: /tmp/FusionInsight-Client/Flume/
FusionInsight_Flume_ClientConfig/JDK/jdk-8u18/bin/java: No such file or directory
```

解决办法

步骤1 适当调大堆内存（xmx）的值。

步骤2 与正常启动Flume的节点进行文件和文件夹权限对比，更改错误文件或文件夹权限。

步骤3 重新配置JAVA_HOME。

客户端替换“${install_home}/fusioninsight-flume-*flume组件版本号*/conf/ENV_VARS文件中JAVA_HOME”的值，服务端替换“etc”目录下“ENV_VARS”文件中“JAVA_HOME”的值。

其中“JAVA_HOME”的值可通过登录正常启动Flume的节点，执行**echo ${JAVA_HOME}**获取。

说明

${install_home}为Flume客户端的安装路径。

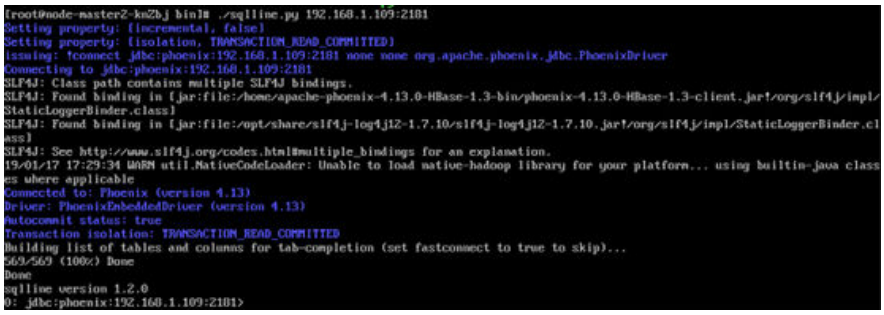
----结束

11 使用 HBase

11.1 连接到 HBase 响应慢

问题现象

在相同的VPC下，外部集群通过Phoenix连接到HBase时，响应较慢。



原因分析

客户配置了DNS服务，由于客户端连接到HBase先通过DNS来解析服务器端，所以导致响应慢。

处理步骤

步骤1 以**root**用户登录Master节点。

步骤2 执行**vi /etc/resolv.conf**，打开resolv.conf文件，注释掉DNS服务器地址。

例如，#1.1.1.1。

----结束

11.2 HBase 用户认证失败

用户问题

HBase用户认证失败。

问题现象

客户侧HBase用户认证失败，报错信息如下：

```
2019-05-13 10:53:09,975 ERROR [localhost-startStop-1] xxxConfig.LoginUtil: login failed with hbaseuser
and /usr/local/linoseyc/hbase-tomcat/webapps/bigdata_hbase/WEB-INF/classes/user.keytab.
2019-05-13 10:53:09,975 ERROR [localhost-startStop-1] xxxConfig.LoginUtil: perhaps cause 1 is (wrong
password) keytab file and user not match, you can kinit -k -t keytab user in client server to check.
2019-05-13 10:53:09,975 ERROR [localhost-startStop-1] xxxConfig.LoginUtil: perhaps cause 2 is (clock
skew) time of local server and remote server not match, please check ntp to remote server.
2019-05-13 10:53:09,975 ERROR [localhost-startStop-1] xxxConfig.LoginUtil: perhaps cause 3 is (aes256 not
support) aes256 not support by default jdk/jre, need copy local_policy.jar and US_export_policy.jar from
remote server in path ${BIGDATA_HOME}/jdk/jre/lib/security.
```

原因分析

用户使用的JDK中的jar包与MRS服务认证的jar包版本不一致。

处理步骤

步骤1 以**root**用户登录集群Master1节点。

步骤2 执行如下命令，查看MRS服务认证的jar包。

ll /opt/share/local_policy/local_policy.jar

ll /opt/Bigdata/jdk{version}/jre/lib/security/local_policy.jar

步骤3 将步骤2中的jar包下载到本地。

步骤4 将下载的jar包替换到本地JDK目录/opt/Bigdata/jdk/jre/lib/security。

步骤5 执行**cd** *客户端安装目录*/**HBase/hbase/bin**命令，进入到HBase的bin目录。

步骤6 执行**sh start-hbase.sh**命令，重启HBase组件。

----结束

11.3 端口被占用导致 RegionServer 启动失败

问题现象

Manager页面监控发现RegionServer状态为Restoring。

原因分析

1. 通过查看RegionServer日志（/var/log/Bigdata/hbase/rs/hbase-omm-xxx.log）。
2. 使用**lsof -i:21302**（MRS1.7.X及以后端口号是16020）查看到pid，然后根据pid查看到相应的进程，发现RegionServer的端口被DFSZkFailoverController占用。
3. 查看“/proc/sys/net/ipv4/ip_local_port_range”显示为“9000 65500”，临时端口范围与MRS产品端口范围重叠，因为安装时未进行preinstall操作。

解决办法

步骤1 执行**kill -9** *DFSZkFailoverController的pid*， 使得其重启后绑定其它端口，然后重启Restoring的RegionServer。

----结束

11.4 节点剩余内存不足导致 HBase 启动失败

问题现象

HBase的RegionServer服务一直是Restoring状态。

原因分析

1. 查看RegionServer的日志（“/var/log/Bigdata/hbase/rs/hbase-omm-XXX.out”），发现显示以下打印信息：
   There is insufficient memory for the Java Runtime Environment to continue.
2. 使用**free**指令查看，该节点确实没有足够内存。

解决办法

步骤1 现场进行排查内存不足原因，确认是否有某些进程占用过多内存，或者由于服务器自身内存不足。

----结束

11.5 HDFS 性能差导致 HBase 服务不可用告警

问题现象

HBase组件不断上报服务不可用告警。

原因分析

该问题多半为HDFS性能较慢，导致健康检查超时，从而导致监控告警。可通过以下方式判断：

1. 首先查看HMaster日志（“/var/log/Bigdata/hbase/hm/hbase-omm-xxx.log”），确认HMaster日志中没有频繁打印“system pause”或“jvm”等GC相关信息。
2. 然后可以通过下列三种方式确认原因为HDFS性能慢造成告警产生。
   a. 使用客户端验证，通过**hbase shell**进入hbase命令行后，执行**list**验证需要运行多久。
   b. 开启HDFS的debug日志，然后查看下层目录很多的路径（**hadoop fs -ls /XXX/XXX**），验证需要运行多久。
   c. 打印HMaster进程jstack：

      su - omm

      jps

jstack pid

3. 如下图所示，Jstack显示一直卡在DFSClient.listPaths。

图 11-1 异常

```
java.lang.Thread.State: WAITING (on object monitor)
     at java.lang.Object.wait(Native Method)
     at java.lang.Object.wait(Object.java:503)
     at org.apache.hadoop.ipc.Client.call(Client.java:1396)
     - locked <0x00000000b9268a38> (a org.apache.hadoop.ipc.Client$Call)
     at org.apache.hadoop.ipc.Client.call(Client.java:1363)
     at org.apache.hadoop.ipc.ProtobufRpcEngine$Invoker.invoke(ProtobufRpcEngine.java:206)
     at com.sun.proxy.$Proxy13.getListing(Unknown Source)
     at org.apache.hadoop.hdfs.protocolPB.ClientNamenodeProtocolTranslatorPB.getListing(ClientNamenodeProtocolTra
     at sun.reflect.GeneratedMethodAccessor24.invoke(Unknown Source)
     at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
     at java.lang.reflect.Method.invoke(Method.java:606)
     at org.apache.hadoop.io.retry.RetryInvocationHandler.invokeMethod(RetryInvocationHandler.java:187)
     at org.apache.hadoop.io.retry.RetryInvocationHandler.invoke(RetryInvocationHandler.java:102)
     at com.sun.proxy.$Proxy14.getListing(Unknown Source)
     at sun.reflect.GeneratedMethodAccessor24.invoke(Unknown Source)
     at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
     at java.lang.reflect.Method.invoke(Method.java:606)
     at org.apache.hadoop.hbase.fs.HFileSystem$1.invoke(HFileSystem.java:294)
     at com.sun.proxy.$Proxy17.getListing(Unknown Source)
     at org.apache.hadoop.hdfs.DFSClient.listPaths(DFSClient.java:1767)
     at org.apache.hadoop.hdfs.DFSClient.listPaths(DFSClient.java:1750)
     at org.apache.hadoop.hdfs.DistributedFileSystem.listStatusInternal(DistributedFileSystem.java:691)
     at org.apache.hadoop.hdfs.DistributedFileSystem.access$600(DistributedFileSystem.java:102)
     at org.apache.hadoop.hdfs.DistributedFileSystem$15.doCall(DistributedFileSystem.java:753)
     at org.apache.hadoop.hdfs.DistributedFileSystem$15.doCall(DistributedFileSystem.java:749)
     at org.apache.hadoop.fs.FileSystemLinkResolver.resolve(FileSystemLinkResolver.java:81)
     at org.apache.hadoop.hdfs.DistributedFileSystem.listStatus(DistributedFileSystem.java:749)
     at org.apache.hadoop.fs.FileSystem.listStatus(FileSystem.java:1483)
```

解决办法

步骤1 如果确认是HDFS性能慢导致告警，需要排除是否为旧版本中Impala运行导致HDFS性能慢或者是否为集群最初部署时JournalNode部署不正确（部署过多，大于3个）。

----结束

11.6 参数不合理导致 HBase 启动失败

问题现象

修改部分参数后，无法正常启动HBase。

原因分析

1. 查看HMaster日志（/var/log/Bigdata/hbase/hm/hbase-omm-xxx.log）显示，hbase.regionserver.global.memstore.size + hfile.block.cache.size总和大于0.8导致启动不成功，因此需要调整参数配置值总和低于0.8。

```
ava HotSpot(TM) 64-Bit Server VM warning: ignoring option PermSize=128M; support was removed in 8.0
ava HotSpot(TM) 64-Bit Server VM warning: ignoring option MaxPermSize=128M; support was removed in 8.0
ava HotSpot(TM) 64-Bit Server VM warning: UseCMSCompactAtFullCollection is deprecated and will likely be removed in a future release.
ava HotSpot(TM) 64-Bit Server VM warning: CMSFullGCsBeforeCompaction is deprecated and will likely be removed in a future release.
NFO: Watching file:/opt/huawei/Bigdata/etc/1_14_RegionServer/log4j.properties for changes with interval : 60000
xception in thread "main" java.lang.RuntimeException: Current heap configuration for MemStore and BlockCache exceeds the threshold required for successful cluster operation. The combined value cannot exceed
0.8. Please check the settings for hbase.regionserver.global.memstore.size and hfile.block.cache.size in your configuration. hbase.regionserver.global.memstore.size is 0.6 hfile.block.cache.size is 0.25
        at org.apache.hadoop.hbase.io.util.HeapMemorySizeUtil.checkForClusterFreeMemoryLimit(HeapMemorySizeUtil.java:64)
        at org.apache.hadoop.hbase.HBaseConfiguration.addHbaseResources(HBaseConfiguration.java:82)
        at org.apache.hadoop.hbase.HBaseConfiguration.create(HBaseConfiguration.java:96)
        at org.apache.hadoop.hbase.regionserver.HRegionServer.main(HRegionServer.java:2663)
```

2. 查看HMaster和RegionServer的out日志（/var/log/Bigdata/hbase/hm/hbase-omm-xxx.out/var/log/Bigdata/hbase/rs/hbase-omm-xxx.out），提示Unrecognized VM option。

```
Unrecognized VM option
Error: Could not create the Java Virtual Machine.
Error: A fatal exception has occurred. Program will exit.
```

检查GC_OPTS相关参数存在多余空格，如-D sun.rmi.dgc.server.gcInterval=0x7FFFFFFFFFFFFFE。

解决办法

步骤1 针对memstore、cache修改配置参数后，重启HBase服务成功。

步骤2 针对GC_OPTS配置错误，修改参数后重启HBase服务成功。

----结束

11.7 残留进程导致 Regionsever 启动失败

问题现象

HBase服务启动失败，健康检查报错。

原因分析

查看启动HBase服务时manager页面的详细打印信息，提示the previous process is not quit。

解决办法

步骤1 登录节点，后台通过执行**ps -ef | grep HRegionServer**发现确实存在一个残留的进程。

步骤2 确认进程可以终止后，使用kill命令终止该进程（如果kill无法终止该进程，需要使用kill -9来强制终止该进程）。

步骤3 重新启动HBase服务成功。

----结束

11.8 HDFS 上设置配额导致 HBase 启动失败

问题现象

HBase启动失败。

原因分析

查看HMaster日志信息（“/var/log/Bigdata/hbase/hm/hbase-omm-xxx.log”），出现如下异常，The DiskSpace quota of /hbase is exceeded。

```
Cause:
org.apache.hadoop.hdfs.protocol.DSQuotaExceededException: The DiskSpace quota of /hbase is exceeded: quota=29240.3g diskspace consumed=37945.7g
        at org.apache.hadoop.hdfs.server.namenode.INodeDirectoryWithQuota.verifyQuota(INodeDirectoryWithQuota.java:159)
        at org.apache.hadoop.hdfs.server.namenode.FSDirectory.verifyQuota(FSDirectory.java:1643)
        at org.apache.hadoop.hdfs.server.namenode.FSDirectory.updateCount(FSDirectory.java:1978)
        at org.apache.hadoop.hdfs.server.namenode.FSDirectory.addChild(FSDirectory.java:1745)
        at org.apache.hadoop.hdfs.server.namenode.FSDirectory.addChild(FSDirectory.java:1762)
        at org.apache.hadoop.hdfs.server.namenode.FSDirectory.unprotectedMkdir(FSDirectory.java:1581)
        at org.apache.hadoop.hdfs.server.namenode.FSDirectory.mkdirs(FSDirectory.java:1537)
        at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.mkdirsInternal(FSNamesystem.java:2768)
        at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.mkdirs(FSNamesystem.java:2721)
        at org.apache.hadoop.hdfs.server.namenode.NameNodeRpcServer.mkdirs(NameNodeRpcServer.java:641)
        at org.apache.hadoop.hdfs.protocolPB.ClientNamenodeProtocolServerSideTranslatorPB.mkdirs(ClientNamenodeProtocolServerSideTranslatorPB.java:416)
        at org.apache.hadoop.hdfs.protocol.proto.ClientNamenodeProtocolProtos$ClientNamenodeProtocol$2.callBlockingMethod(ClientNamenodeProtocolProtos.j
        at org.apache.hadoop.ipc.ProtobufRpcEngine$Server$ProtoBufRpcInvoker.call(ProtobufRpcEngine.java:427)
        at org.apache.hadoop.ipc.RPC$Server.call(RPC.java:925)
        at org.apache.hadoop.ipc.Server$Handler$1.run(Server.java:1710)
        at org.apache.hadoop.ipc.Server$Handler$1.run(Server.java:1706)
        at java.security.AccessController.doPrivileged(Native Method)
        at javax.security.auth.Subject.doAs(Subject.java:415)
        at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1232)
        at org.apache.hadoop.ipc.Server$Handler.run(Server.java:1704)

        at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)
        at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:57)
        at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:45)
        at java.lang.reflect.Constructor.newInstance(Constructor.java:525)
        at org.apache.hadoop.ipc.RemoteException.instantiateException(RemoteException.java:90)
        at org.apache.hadoop.ipc.RemoteException.unwrapRemoteException(RemoteException.java:57)
        at org.apache.hadoop.hdfs.DFSClient.primitiveMkdir(DFSClient.java:1868)
        at org.apache.hadoop.hdfs.DFSClient.mkdirs(DFSClient.java:1837)
        at org.apache.hadoop.hdfs.DistributedFileSystem.mkdirs(DistributedFileSystem.java:483)
        at org.apache.hadoop.fs.FileSystem.mkdirs(FileSystem.java:1728)
        at org.apache.hadoop.hbase.regionserver.wal.HLog.<init>(HLog.java:413)
        at org.apache.hadoop.hbase.regionserver.wal.HLog.<init>(HLog.java:367)
        at org.apache.hadoop.hbase.regionserver.HRegionServer.instantiateHLog(HRegionServer.java:1348)
        at org.apache.hadoop.hbase.regionserver.HRegionServer.setupWALAndReplication(HRegionServer.java:1337)
        at org.apache.hadoop.hbase.regionserver.HRegionServer.handleReportForDutyResponse(HRegionServer.java:1048)
        at org.apache.hadoop.hbase.regionserver.HRegionServer.run(HRegionServer.java:714)
        at java.lang.Thread.run(Thread.java:722)
```

解决办法

步骤1 通过后台使用**df -h**命令查看数据盘目录空间已满，因此需要删除无用的数据来进行应急恢复。

步骤2 后续需要扩容节点来解决数据目录空间不足问题。

----结束

11.9 HBase version 文件损坏导致启动失败

问题背景

HBase启动失败。

原因分析

1. HBase启动时会读取hbase.version文件，但是日志显示读取存在异常。

```
2019-07-27 15:30:18,692 | ERROR | master/node-master1r2W8:16000:becomeActiveMaster | Failed to become active master | org.slf4j.helpers.MarkerIgnoringBase.error(MarkerIgnoringBase.java:159)
org.apache.hadoop.hbase.util.FileSystemVersionException: HBase file layout needs to be upgraded. You have version null and I want version 8. Consult http://hbase.apache.org/book.html for furth
er information about upgrading HBase. Is your hbase.rootdir valid? If so, you may need to run 'hbase hbck -fixVersionFile'.
        at org.apache.hadoop.hbase.util.FSUtils.checkVersion(FSUtils.java:509)
        at org.apache.hadoop.hbase.master.MasterFileSystem.checkRootDir(MasterFileSystem.java:271)
        at org.apache.hadoop.hbase.master.MasterFileSystem.createInitialFileSystemLayout(MasterFileSystem.java:151)
        at org.apache.hadoop.hbase.master.MasterFileSystem.<init>(MasterFileSystem.java:122)
        at org.apache.hadoop.hbase.master.HMaster.finishActiveMasterInitialization(HMaster.java:869)
        at org.apache.hadoop.hbase.master.HMaster.startActiveMasterManager(HMaster.java:2297)
```

2. 通过**hadoop fs -cat /hbase/hbase.version**命令发现文件不能正常查看，该文件损坏。

解决办法

步骤1 执行**hbase hbck -fixVersionFile**命令修复文件。

步骤2 如**步骤1**不能解决，从同版本的其他集群中获取hbase.version文件上传进行替换。

步骤3 重新启动HBase服务。

----结束

11.10 无业务情况下，RegionServer 占用 CPU 高

问题背景

无业务情况下，RegionServer占用CPU较高。

原因分析

1. 通过**top**命令获取RegionServer的进程使用CPU情况信息，查看CPU使用率高的进程号。
2. 根据RegionServer的进程编号，获取该进程下线程使用CPU情况。

   top -H -p <PID>（根据实际RegionServer的进程ID进行替换），具体如下图所示，发现部分线程CPU使用率均达到80%。

```
PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
75706 omm     20   0 6879444   1.0g  25612 S  90.4  1.6   0:00.00 java
75716 omm     20   0 6879444   1.0g  25612 S  90.4  1.6   0:04.74 java
75720 omm     20   0 6879444   1.0g  25612 S  88.6  1.6   0:01.93 java
75721 omm     20   0 6879444   1.0g  25612 S  86.8  1.6   0:01.99 java
75722 omm     20   0 6879444   1.0g  25612 S  86.8  1.6   0:01.94 java
75723 omm     20   0 6879444   1.0g  25612 S  86.8  1.6   0:01.96 java
75724 omm     20   0 6879444   1.0g  25612 S  86.8  1.6   0:01.97 java
75725 omm     20   0 6879444   1.0g  25612 S  81.5  1.6   0:02.06 java
75726 omm     20   0 6879444   1.0g  25612 S  79.7  1.6   0:02.01 java
75727 omm     20   0 6879444   1.0g  25612 S  79.7  1.6   0:01.95 java
75728 omm     20   0 6879444   1.0g  25612 S  78.0  1.6   0:01.99 java
```

3. 根据RegionServer的进程编号，获取线程堆栈信息。

   jstack 12345 >allstack.txt（根据实际RegionServer的进程ID进行替换）
4. 将需要的线程ID转换为16进制格式：

   printf "%x\n" 30648

   输出结果TID为77b8。
5. 根据输出16进制TID，在线程堆栈中进行查找，发现在执行compaction操作。

```
"regionserver/ahbd-hbase-dat1/12.2.168.1:21302-longCompactions-1482676601478" #1641 prio=5 os_prio=0 tid=0x00007fa614563000 nid=0x77b8 runnable [0x0
   java.lang.Thread.State: RUNNABLE
    at org.apache.hadoop.io.compress.snappy.SnappyCompressor.compressBytesDirect(Native Method)
    at org.apache.hadoop.io.compress.snappy.SnappyCompressor.compress(SnappyCompressor.java:228)
    at org.apache.hadoop.io.compress.BlockCompressorStream.compress(BlockCompressorStream.java:149)
    at org.apache.hadoop.io.compress.BlockCompressorStream.finish(BlockCompressorStream.java:142)
    at org.apache.hadoop.hbase.io.encoding.HFileBlockDefaultEncodingContext.compressAfterEncoding(HFileBlockDefaultEncodingContext.java:219)
    at org.apache.hadoop.hbase.io.encoding.HFileBlockDefaultEncodingContext.compressAndEncrypt(HFileBlockDefaultEncodingContext.java:132)
    at org.apache.hadoop.hbase.io.hfile.HFileBlock$Writer.finishBlock(HFileBlock.java:989)
    at org.apache.hadoop.hbase.io.hfile.HFileBlock$Writer.ensureBlockReady(HFileBlock.java:961)
    at org.apache.hadoop.hbase.io.hfile.HFileBlock$Writer.finishBlockAndWriteHeaderAndData(HFileBlock.java:1077)
```

6. 对其它线程执行相同操作，发现均为compactions线程。

```
"regionserver/ahbd-hbase-dat1/12.2.168.1:21302-longCompactions-1482676601473" #1629 prio=5 os_prio=0 tid=0x00007fa61454d800 nid=0x77a0 runnable
   java.lang.Thread.State: RUNNABLE
    at org.apache.hadoop.hdfs.DFSOutputStream.writeChunk(DFSOutputStream.java:425)
    - locked <0x000000020276ba38> (a org.apache.hadoop.hdfs.DFSOutputStream)
    at org.apache.hadoop.fs.FSOutputSummer.writeChecksumChunks(FSOutputSummer.java:214)
    at org.apache.hadoop.fs.FSOutputSummer.flushBuffer(FSOutputSummer.java:165)
    - locked <0x000000020276ba38> (a org.apache.hadoop.hdfs.DFSOutputStream)
    at org.apache.hadoop.fs.FSOutputSummer.flushBuffer(FSOutputSummer.java:146)
    - eliminated <0x000000020276ba38> (a org.apache.hadoop.hdfs.DFSOutputStream)
    at org.apache.hadoop.fs.FSOutputSummer.write1(FSOutputSummer.java:137)
    at org.apache.hadoop.fs.FSOutputSummer.write(FSOutputSummer.java:112)
    - locked <0x000000020276ba38> (a org.apache.hadoop.hdfs.DFSOutputStream)
    at org.apache.hadoop.fs.FSDataOutputStream$PositionCache.write(FSDataOutputStream.java:58)
    at java.io.DataOutputStream.write(DataOutputStream.java:107)
    - locked <0x00000004de9535c8> (a org.apache.hadoop.hdfs.client.HdfsDataOutputStream)
    at java.io.FilterOutputStream.write(FilterOutputStream.java:97)
```

解决办法

属于正常现象。

发现消耗CPU较高线程均为HBase的compaction，其中部分线程调用Snappy压缩处理，部分线程调用HDFS读写数据。当前每个Region数据量和数据文件多，且采用Snappy压缩算法，因此执行compaction时会使用大量CPU导致CPU较高。

定位办法

步骤1 使用**top**命令查看CPU使用率高的进程号。

步骤2 查看此进程中占用CPU高的线程。

使用命令**top -H -p <PID>**即可打印出某进程<PID>下的线程的CPU耗时信息。

一般某个进程如果出现问题，是因为某个线程出现问题了，获取查询到的占用CPU最高的线程号。

或者使用命令**ps -mp <PID> -o THREAD,tid,time | sort -rn**。

观察回显可以得到CPU最高的线程号。

步骤3 获取出现问题的线程的堆栈。

java问题使用jstack工具是最有效，最可靠的。

到java/bin目录下有jstack工具，获取进程堆栈，并输出到本地文件。

jstack <PID> > allstack.txt

获取线程堆栈，并输出到本地文件。

步骤4 将需要的线程ID转换为16进制格式。

printf "%x\n" <PID>

回显结果为线程ID，即TID。

步骤5 使用命令获得TID,并输出到本地文件。

jstack <PID> | grep <TID> > Onestack.txt

如果只是在命令行窗口查看，可以使用命名：

jstack <PID> | grep <TID> -A 30

-A 30意思是显示30行。

----结束

11.11 HBase 启动失败，RegionServer 日志中提示“FileNotFoundException”

问题背景

HBase启动失败，RegionServer一直处于Restoring状态。

原因分析

1. 查看RegionServer的日志（/var/log/Bigdata/hbase/rs/hbase-omm-XXX.log），发现显示以下打印信息：

```
 | ERROR | RS_OPEN_REGION-ab-dn01:21302-2 | ABORTING region server ab-
dn01,21302,1487663269375: The coprocessor
org.apache.kylin.storage.hbase.cube.v2.coprocessor.endpoint.CubeVisitService threw
java.io.FileNotFoundException: File does not exist: hdfs://hacluster/kylin/kylin_metadata/coprocessor/
kylin-coprocessor-1.6.0-SNAPSHOT-0.jar |
org.apache.hadoop.hbase.regionserver.HRegionServer.abort(HRegionServer.java:2123)
java.io.FileNotFoundException: File does not exist: hdfs://hacluster/kylin/kylin_metadata/coprocessor/
kylin-coprocessor-1.6.0-SNAPSHOT-0.jar
at org.apache.hadoop.hdfs.DistributedFileSystem$25.doCall(DistributedFileSystem.java:1399)
at org.apache.hadoop.hdfs.DistributedFileSystem$25.doCall(DistributedFileSystem.java:1391)
at org.apache.hadoop.fs.FileSystemLinkResolver.resolve(FileSystemLinkResolver.java:81)
at org.apache.hadoop.hdfs.DistributedFileSystem.getFileStatus(DistributedFileSystem.java:1391)
at org.apache.hadoop.fs.FileUtil.copy(FileUtil.java:340)
at org.apache.hadoop.fs.FileUtil.copy(FileUtil.java:292)
at org.apache.hadoop.fs.FileSystem.copyToLocalFile(FileSystem.java:2038)
at org.apache.hadoop.fs.FileSystem.copyToLocalFile(FileSystem.java:2007)
at org.apache.hadoop.fs.FileSystem.copyToLocalFile(FileSystem.java:1983)
at org.apache.hadoop.hbase.util.CoprocessorClassLoader.init(CoprocessorClassLoader.java:168)
at
org.apache.hadoop.hbase.util.CoprocessorClassLoader.getClassLoader(CoprocessorClassLoader.java:250)
at org.apache.hadoop.hbase.coprocessor.CoprocessorHost.load(CoprocessorHost.java:224)
at
org.apache.hadoop.hbase.regionserver.RegionCoprocessorHost.loadTableCoprocessors(RegionCoprocess
orHost.java:365)
at
org.apache.hadoop.hbase.regionserver.RegionCoprocessorHost.<init>(RegionCoprocessorHost.java:227)
at org.apache.hadoop.hbase.regionserver.HRegion.<init>(HRegion.java:783)
at org.apache.hadoop.hbase.regionserver.HRegion.<init>(HRegion.java:689)
at sun.reflect.GeneratedConstructorAccessor22.newInstance(Unknown Source)
at
sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:4
5)
at java.lang.reflect.Constructor.newInstance(Constructor.java:423)
at org.apache.hadoop.hbase.regionserver.HRegion.newHRegion(HRegion.java:6312)
at org.apache.hadoop.hbase.regionserver.HRegion.openHRegion(HRegion.java:6622)
at org.apache.hadoop.hbase.regionserver.HRegion.openHRegion(HRegion.java:6594)
at org.apache.hadoop.hbase.regionserver.HRegion.openHRegion(HRegion.java:6550)
at org.apache.hadoop.hbase.regionserver.HRegion.openHRegion(HRegion.java:6501)
at
org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler.openRegion(OpenRegionHandler.ja
va:363)
at
org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler.process(OpenRegionHandler.java:12
9)
at org.apache.hadoop.hbase.executor.EventHandler.run(EventHandler.java:129)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)
at java.lang.Thread.run(Thread.java:745)
```

2. 使用客户端**hdfs**命令查看，如下文件不存在。

   hdfs://hacluster/kylin/kylin_metadata/coprocessor/kylin-coprocessor-1.6.0-SNAPSHOT-0.jar

3. HBase在配置协处理器时，一定要保证对应的jar包路径没有问题，否则HBase会无法启动。

解决办法

使用Kylin对接MRS，确保Kylin相关jar包存在。

11.12 HBase 启动后原生页面显示 RegionServer 个数多于实际个数

问题背景

HBase启动后，HMaster原生页面显示RegionServer个数多于实际RegionServer个数。

查看HMaster原生页面，显示有4个RegionServer在线，如下图示：

ServerName	Start time	Requests Per Second	Num. Regions
controller-192-168-1-1,21302,1494933959261	Tue May 16 19:25:59 CST 2017	0	19
controller-192-168-1-2,21302,1494933957536	Tue May 16 19:25:57 CST 2017	0	24
controller-192-168-1-3,21302,1494933958592	Tue May 16 19:25:58 CST 2017	0	16
eth0,21302,1494933958592	Tue May 16 19:25:58 CST 2017	0	0
Total:4		0	59

原因分析

如下图可以看出，第三行hostname为controller-192-168-1-3节点和第四行hostname为eth0节点为同一RegionServer上报的信息，登录相应节点，查看/etc/hosts文件，发现，对应同一ip，配置两个hostname。如下：

```
# special IPv6 addresses
::1             localhost ipv6-localhost ipv6-loopback

fe00::0         ipv6-localnet

ff00::0         ipv6-mcastprefix
ff02::1         ipv6-allnodes
ff02::2         ipv6-allrouters
ff02::3         ipv6-allhosts
11.1.1.3        eth2 eth2
#192.168.1.3     eth0 eth0
192.168.2.3     eth1 eth1
10.130.87.37    eth3 eth3
192.168.1.102 controller
1.1.1.1 hadoop.hadoop.com
192.168.1.2 controller-192-168-1-2
192.168.1.1 controller-192-168-1-1
192.168.1.3 controller-192-168-1-3
```

解决办法

登录RegionServer所在节点，修改/etc/hosts文件，同一ip只能对应同一hostname。

11.13 RegionServer 实例异常，处于 Restoring 状态

问题背景

HBase启动失败，RegionServer一直处于Restoring状态。

原因分析

查看异常的RegionServer实例的运行日志（/var/log/Bigdata/hbase/rs/hbase-omm-XXX.log），发现显示以下打印信息ClockOutOfSyncException...，Reported time is too far out of sync with master

```
2017-09-18 11:16:23,636 | FATAL | regionserver21302 | Master rejected startup because clock is out of sync |
org.apache.hadoop.hbase.regionserver.HRegionServer.reportForDuty(HRegionServer.java:2059)
org.apache.hadoop.hbase.ClockOutOfSyncException: org.apache.hadoop.hbase.ClockOutOfSyncException:
Server nl-bi-fi-datanode-24-65,21302,1505726180086 has been rejected; Reported time is too far out of
sync with master.  Time difference of 152109ms > max allowed of 30000ms
at org.apache.hadoop.hbase.master.ServerManager.checkClockSkew(ServerManager.java:354)
...

...
2017-09-18 11:16:23,858 | ERROR | main | Region server exiting |
org.apache.hadoop.hbase.regionserver.HRegionServerCommandLine.start(HRegionServerCommandLine.java:
70)
java.lang.RuntimeException: HRegionServer Aborted
```

该日志说明异常的RegionServer实例和HMaster实例的时差大于允许的时差值30s（由参数hbase.regionserver.maxclockskew控制，默认30000ms），导致RegionServer实例异常。

解决办法

调整异常节点时间，确保节点间时差小于30s。

11.14 新安装的集群 HBase 启动失败

问题背景

新安装的集群HBase启动失败，查看RegionServer日志报如下错误：

```
2018-02-24 16:53:03,863 | ERROR | regionserver/host3/187.6.71.69:21302 | Master passed us a different
hostname to use; was=host3, but now=187-6-71-69 |
org.apache.hadoop.hbase.regionserver.HRegionServer.handleReportForDutyResponse(HRegionServer.java:138
6)
```

原因分析

/etc/hosts中同一个IP地址配置了多个主机名映射关系。

解决办法

步骤1 修改/etc/host中IP与主机名的映射关系，配置正确。

步骤2 重新启动HBase组件。

----结束

11.15 acl 表目录丢失导致 HBase 启动失败

问题背景与现象

集群HBase启动失败

原因分析

1. 查看HBase的HMaster日志，报如下错误：

```
2018-04-10 09:14:05,616 | INFO  | ftn-ies-301-a-fi03:21300.activeMasterManager | Entered into preCreateTable. | org.apache.hadoop.hbase.index.coprocessor.mas
le(IndexMasterObserver.java:103)
2018-04-10 09:14:05,616 | INFO  | ftn-ies-301-a-fi03:21300.activeMasterManager | Exiting from preCreateTable. | org.apache.hadoop.hbase.index.coprocessor.mas
le(IndexMasterObserver.java:159)
2018-04-10 09:14:05,617 | INFO  | ftn-ies-301-a-fi03:21300.activeMasterManager | Client=null/null create 'hbase:acl', {NAME => 'l', BLOOMFILTER => 'NONE', VE
KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COMPRESSION => 'NONE', CACHE_DATA_IN_L1 => 'true', MIN_VERSIONS => '0', BLOCK
, REPLICATION_SCOPE => '0'} | org.apache.hadoop.hbase.master.HMaster.createTable(HMaster.java:1876)
2018-04-10 09:14:05,653 | ERROR | ftn-ies-301-a-fi03:21300.activeMasterManager | Exception occurred while creating the table hbase:acl | org.apache.hadoop.hb
er.java:1909)
org.apache.hadoop.hbase.TableExistsException: hbase:acl
        at org.apache.hadoop.hbase.master.handler.CreateTableHandler.checkAndSetEnablingTable(CreateTableHandler.java:172)
        at org.apache.hadoop.hbase.master.handler.CreateTableHandler.prepare(CreateTableHandler.java:140)
        at org.apache.hadoop.hbase.master.HMaster.createTable(HMaster.java:1905)
        at org.apache.hadoop.hbase.security.access.AccessControlLists.createACLTable(AccessControlLists.java:128)
        at org.apache.hadoop.hbase.security.access.AccessController.postStartMaster(AccessController.java:1416)
        at org.apache.hadoop.hbase.master.MasterCoprocessorHost$62.call(MasterCoprocessorHost.java:769)
        at org.apache.hadoop.hbase.master.MasterCoprocessorHost.execOperation(MasterCoprocessorHost.java:1315)
        at org.apache.hadoop.hbase.master.MasterCoprocessorHost.postStartMaster(MasterCoprocessorHost.java:765)
        at org.apache.hadoop.hbase.master.HMaster.finishActiveMasterInitialization(HMaster.java:933)
        at org.apache.hadoop.hbase.master.HMaster.access$900(HMaster.java:190)
        at org.apache.hadoop.hbase.master.HMaster$3.run(HMaster.java:2081)
        at java.lang.Thread.run(Thread.java:745)
2018-04-10 09:14:05,656 | ERROR | ftn-ies-301-a-fi03:21300.activeMasterManager | Coprocessor postStartMaster() hook failed | org.apache.hadoop.hbase.master.H
ion(HMaster.java:935)
org.apache.hadoop.hbase.TableExistsException: hbase:acl
        at org.apache.hadoop.hbase.master.handler.CreateTableHandler.checkAndSetEnablingTable(CreateTableHandler.java:172)
        at org.apache.hadoop.hbase.master.handler.CreateTableHandler.prepare(CreateTableHandler.java:140)
        at org.apache.hadoop.hbase.master.HMaster.createTable(HMaster.java:1905)
        at org.apache.hadoop.hbase.security.access.AccessControlLists.createACLTable(AccessControlLists.java:128)
        at org.apache.hadoop.hbase.security.access.AccessController.postStartMaster(AccessController.java:1416)
        at org.apache.hadoop.hbase.master.MasterCoprocessorHost$62.call(MasterCoprocessorHost.java:769)
        at org.apache.hadoop.hbase.master.MasterCoprocessorHost.execOperation(MasterCoprocessorHost.java:1315)
```

2. 检查HDFS上HBase的路径发现acl表路径丢失。

Browse Directory

/hbase/data/hbase Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwx------	hbase	supergroup	0 B	Thu Mar 15 21:30:29 2018	0	0 B	meta
drwx------	hbase	supergroup	0 B	Thu Mar 15 21:30:36 2018	0	0 B	namespace

解决办法

步骤1 停止HBase组件。

步骤2 在HBase客户端使用hbase用户登录认证，执行如下命令。

例如：

```
hadoop03:~ # source /opt/client/bigdata_env
hadoop03:~ # kinit hbase
Password for hbase@HADOOP.COM:
hadoop03:~ # hbase zkcli
```

步骤3 删除zk中acl表信息。

例如：

```
[zk: hadoop01:24002,hadoop02:24002,hadoop03:24002(CONNECTED) 0] deleteall /hbase/table/hbase:acl
[zk: hadoop01:24002,hadoop02:24002,hadoop03:24002(CONNECTED) 0] deleteall /hbase/table-lock/
hbase:acl
```

步骤4 启动HBase组件。

----结束

11.16 集群上下电之后 HBase 启动失败

问题背景与现象

集群的ECS关机重启后，HBase启动失败。

原因分析

查看HMaster的运行日志，发现有报大量的如下错误：

```
2018-03-26 11:10:54,185 | INFO  | hadoopc1h3,21300,1522031630949_splitLogManager__ChoreService_1 |
total tasks = 1 unassigned = 0 tasks={/hbase/splitWAL/WALs%2Fhadoopc1h1%2C213
02%2C1520214023667-splitting
%2Fhadoopc1h1%252C21302%252C1520214023667.default.1520584926990=last_update =
1522033841041 last_version = 34255 cur_worker_name = hadoopc1h3,21302,
1520943011826 status = in_progress incarnation = 3 resubmits = 3 batch = installed = 1 done = 0 error = 0}
| org.apache.hadoop.hbase.master.SplitLogManager$TimeoutMonitor.chore
(SplitLogManager.java:745)
2018-03-26 11:11:00,185 | INFO  | hadoopc1h3,21300,1522031630949_splitLogManager__ChoreService_1 |
total tasks = 1 unassigned = 0 tasks={/hbase/splitWAL/WALs%2Fhadoopc1h1%2C213
02%2C1520214023667-splitting
%2Fhadoopc1h1%252C21302%252C1520214023667.default.1520584926990=last_update =
1522033841041 last_version = 34255 cur_worker_name = hadoopc1h3,21302,
1520943011826 status = in_progress incarnation = 3 resubmits = 3 batch = installed = 1 done = 0 error = 0}
| org.apache.hadoop.hbase.master.SplitLogManager$TimeoutMonitor.chore
(SplitLogManager.java:745)
2018-03-26 11:11:06,185 | INFO  | hadoopc1h3,21300,1522031630949_splitLogManager__ChoreService_1 |
total tasks = 1 unassigned = 0 tasks={/hbase/splitWAL/WALs%2Fhadoopc1h1%2C213
02%2C1520214023667-splitting
%2Fhadoopc1h1%252C21302%252C1520214023667.default.1520584926990=last_update =
1522033841041 last_version = 34255 cur_worker_name = hadoopc1h3,21302,
1520943011826 status = in_progress incarnation = 3 resubmits = 3 batch = installed = 1 done = 0 error = 0}
| org.apache.hadoop.hbase.master.SplitLogManager$TimeoutMonitor.chore
(SplitLogManager.java:745)
2018-03-26 11:11:10,787 | INFO  | RpcServer.reader=9,bindAddress=hadoopc1h3,port=21300 | Kerberos
principal name is hbase/hadoop.hadoop.com@HADOOP.COM | org.apache.hadoop.hbase
.ipc.RpcServer$Connection.readPreamble(RpcServer.java:1532)
2018-03-26 11:11:12,185 | INFO  | hadoopc1h3,21300,1522031630949_splitLogManager__ChoreService_1 |
total tasks = 1 unassigned = 0 tasks={/hbase/splitWAL/WALs%2Fhadoopc1h1%2C213
02%2C1520214023667-splitting
%2Fhadoopc1h1%252C21302%252C1520214023667.default.1520584926990=last_update =
1522033841041 last_version = 34255 cur_worker_name = hadoopc1h3,21302,
1520943011826 status = in_progress incarnation = 3 resubmits = 3 batch = installed = 1 done = 0 error = 0}
| org.apache.hadoop.hbase.master.SplitLogManager$TimeoutMonitor.chore
(SplitLogManager.java:745)
2018-03-26 11:11:18,185 | INFO  | hadoopc1h3,21300,1522031630949_splitLogManager__ChoreService_1 |
total tasks = 1 unassigned = 0 tasks={/hbase/splitWAL/WALs%2Fhadoopc1h1%2C213
02%2C1520214023667-splitting
%2Fhadoopc1h1%252C21302%252C1520214023667.default.1520584926990=last_update =
1522033841041 last_version = 34255 cur_worker_name = hadoopc1h3,21302,
1520943011826 status = in_progress incarnation = 3 resubmits = 3 batch = installed = 1 done = 0 error = 0}
| org.apache.hadoop.hbase.master.SplitLogManager$TimeoutMonitor.chore
(SplitLogManager.java:745)
```

节点上下电，RegionServer的wal分裂失败导致。

解决办法

步骤1 停止HBase组件。

步骤2 通过**hdfs fsck**命令检查/hbase/WALs文件的健康状态。

hdfs fsck /hbase/WALs

输出如下表示文件都正常，如果有异常则需要先处理异常的文件，再执行后面的操作。

```
The filesystem under path '/hbase/WALs' is HEALTHY
```

步骤3 备份/hbase/WALs文件。

hdfs dfs -mv /hbase/WALs /hbase/WALs_old

步骤4 新建/hbase/WALs目录。

hdfs dfs -mkdir /hbase/WALs

必须保证路径权限是hbase:hadoop。

步骤5 启动HBase组件。

----结束

11.17 文件块过大导致 HBase 数据导入失败

问题现象

导入数据到hbase报错：NotServingRegionException。

原因分析

当一个block size大于2G时，hdfs在seek的时候会出现读取异常，持续频繁写入regionserver时出现了full gc，且时间比较久，导致hmaster与regionserver之间的心跳异常，然后hmaster把regionserver标记为dead状态，强制重启了Regionserver，重启后触发servercrash机制开始回滚wal日志。现在这个splitwal的文件已经达到将近2.1G，且其仅有一个block块，导致hdfs seek异常，引起splitwal失败，regionserver检测到当前这个wal日志还需要split，又会触发splitwal日志的机制进行回滚，就这样在split与split失败之间不停循环，导致无法上线该regionserver节点上的region，最后出现查询该RS上某一个region时会报region not online的异常。

处理步骤

步骤1 进入HBase服务页面：

- MRS 3.*x*之前版本，单击集群名称，登录集群详情页面，选择“组件管理 > HBase”。

  说明

  如果集群详情页面没有“组件管理”页签，请先完成IAM用户同步（在集群详情页的“概览”页签，单击“IAM用户同步”右侧的“同步”进行IAM用户同步）。

- MRS 3.x及后续版本，登录FusionInsight Manager。然后选择“集群 > *待操作的集群名称* > 服务 > HBase”。

步骤2 在“HMaster Web UI”右侧，单击“HMaster (主)”进入HBase Web UI界面。

步骤3 在“Procedures”页签查看问题节点。

步骤4 以root用户登录问题节并执行**hdfs dfs -ls**命令查看所有块信息。

步骤5 执行**hdfs dfs -mkdir**命令新建目录用于存放问题块。

步骤6 执行**hdfs dfs -mv**将问题块转移至新建目录位置。

----结束

建议与总结

以下两点可供参考：

- 数据块损坏，通过**hdfs fsck /tmp -files -blocks -racks**命令检查block数据块健康信息。
- region正在分裂时对数据的操作会抛NotServingRegionException异常。

11.18 使用 Phoenix 创建 HBase 表后，向索引表中加载数据报错

问题背景与现象

使用Phoenix创建HBase表后，使用命令向索引表中加载数据报错：

- MRS 2.x及之前版本：Mutable secondary indexes must have the hbase.regionserver.wal.codec property set to org.apache.hadoop.hbase.regionserver.wal.IndexedWALEditCodec in the hbase-sites.xml of every region server. tableName=MY_INDEX (state=42Y88,code=1029)

```
Error: ERROR 1029 (42Y88): Mutable secondary indexes must have the hbase.regionserver.wal.codec property set to org.apache.hadoop.hbase.regionserver.wal.IndexedWA
LEditCodec in the hbase-sites.xml of every region server. tableName=MY_INDEX (state=42Y88,code=1029)
java.sql.SQLException: ERROR 1029 (42Y88): Mutable secondary indexes must have the hbase.regionserver.wal.codec property set to org.apache.hadoop.hbase.regionserv
er.wal.IndexedWALEditCodec in the hbase-sites.xml of every region server. tableName=MY_INDEX
        at org.apache.phoenix.exception.SQLExceptionCode$Factory$1.newException(SQLExceptionCode.java:498)
        at org.apache.phoenix.exception.SQLExceptionInfo.buildException(SQLExceptionInfo.java:150)
        at org.apache.phoenix.schema.MetaDataClient.createIndex(MetaDataClient.java:1534)
        at org.apache.phoenix.compile.CreateIndexCompiler$1.execute(CreateIndexCompiler.java:85)
        at org.apache.phoenix.jdbc.PhoenixStatement$2.call(PhoenixStatement.java:410)
        at org.apache.phoenix.jdbc.PhoenixStatement$2.call(PhoenixStatement.java:393)
        at org.apache.phoenix.call.CallRunner.run(CallRunner.java:53)
        at org.apache.phoenix.jdbc.PhoenixStatement.executeMutation(PhoenixStatement.java:392)
        at org.apache.phoenix.jdbc.PhoenixStatement.executeMutation(PhoenixStatement.java:380)
        at org.apache.phoenix.jdbc.PhoenixStatement.execute(PhoenixStatement.java:1829)
        at sqlline.Commands.execute(Commands.java:822)
        at sqlline.Commands.sql(Commands.java:732)
        at sqlline.SqlLine.dispatch(SqlLine.java:813)
        at sqlline.SqlLine.begin(SqlLine.java:686)
        at sqlline.SqlLine.start(SqlLine.java:398)
        at sqlline.SqlLine.main(SqlLine.java:291)
0: jdbc:phoenix:node-master1GlxJ,node-ana-cor>
```

- MRS 3.x及之后版本：Exception in thread "main" java.io.IOException: Retry attempted 10 times without completing, bailing out

```
2022-04-17 20:24:37,157 INFO  [main] tool.LoadIncrementalHFiles: Split occurred while grouping HFiles, retry attempt 10 with 1 files remaining to group or split
2022-04-17 20:24:37,170 ERROR [main] tool.LoadIncrementalHFiles: ------------------------------------------------
Bulk load aborted with some files not yet loaded:
------------------------------------------------
  hdfs://hacluster/tmp/3cdc0475-3867-4d9f-a774-87bc6759ee77/ANALYSIS.USER_IDENTIFICATION/f/36b9e9618d784ccf9d982ce46eba4b76

Exception in thread "main" java.io.IOException: Retry attempted 10 times without completing, bailing out
        at org.apache.hadoop.hbase.tool.LoadIncrementalHFiles.performBulkLoad(LoadIncrementalHFiles.java:460)
        at org.apache.hadoop.hbase.tool.LoadIncrementalHFiles.doBulkLoad(LoadIncrementalHFiles.java:379)
        at org.apache.hadoop.hbase.tool.LoadIncrementalHFiles.doBulkLoad(LoadIncrementalHFiles.java:293)
        at org.apache.phoenix.mapreduce.AbstractBulkLoadTool.completebulkload(AbstractBulkLoadTool.java:389)
        at org.apache.phoenix.mapreduce.AbstractBulkLoadTool.submitJob(AbstractBulkLoadTool.java:343)
        at org.apache.phoenix.mapreduce.AbstractBulkLoadTool.loadData(AbstractBulkLoadTool.java:279)
        at org.apache.phoenix.mapreduce.AbstractBulkLoadTool.run(AbstractBulkLoadTool.java:180)
        at org.apache.hadoop.util.ToolRunner.run(ToolRunner.java:76)
        at org.apache.hadoop.util.ToolRunner.run(ToolRunner.java:90)
        at org.apache.phoenix.mapreduce.JsonBulkLoadTool.main(JsonBulkLoadTool.java:51)
[root@node-master1Hypi ~]#
```

处理步骤

步骤1 MRS 2.x及之前版本，操作步骤如下：

1. 使用**admin**用户登录MRS Manager界面，选择“服务管理 > HBase > 服务配置”，将“参数类别”的“基础配置”切换为“全部配置”，选择“HMaster >

自定义”，给参数“hbase.hmaster.config.expandor”新增名称为“hbase.regionserver.wal.codec”，值为“org.apache.hadoop.hbase.regionserver.wal.IndexedWALEditCodec”的配置项。

2. 选择“RegionServer > 自定义”，给参数“hbase.regionserver.config.expandor”新增名称为“hbase.regionserver.wal.codec”，值为“org.apache.hadoop.hbase.regionserver.wal.IndexedWALEditCodec”的配置项，单击“保存配置”，输入当前用户密码，单击“确定”，保存配置。
3. 单击“服务状态”，选择“更多 > 重启服务”，输入当前用户密码，单击“确定”，重启HBase服务。

步骤2 MRS 3.x及之后版本，操作步骤如下：

1. 使用**admin**用户登录FusionInsight Manager，选择“集群 > 服务 > HBase > 配置 > 全部配置 > RegionServer > 自定义”，给参数“hbase.regionserver.config.expandor”新增名称为“hbase.regionserver.wal.codec”，值为“org.apache.hadoop.hbase.regionserver.wal.IndexedWALEditCodec”的配置项。
2. 选择“HMaster > 自定义”，给参数“hbase.hmaster.config.expandor”新增名称为“hbase.regionserver.wal.codec”，值为“org.apache.hadoop.hbase.regionserver.wal.IndexedWALEditCodec”的配置项。
3. 单击“保存”，在弹出的对话框中单击“确定”，保存配置。
4. 单击“概览”，选择“更多 > 重启服务”，输入当前用户密码，单击“确定”，重启HBase服务。

----结束

11.19 在 MRS 集群客户端无法执行 hbase shell 命令

用户问题

在MRS集群客户端无法执行**hbase shell**命令。

原因分析

- 执行**hbase shell**命令前未配置环境变量。
- 当前MRS集群未安装HBase客户端。

处理步骤

步骤1 使用**root**用户登录安装客户端的节点，切换到客户端安装目录，查看是否安装了HBase客户端。

- 是，执行**步骤2**。
- 否，下载并安装HBase客户端，详细操作请参考**安装客户端**章节。

步骤2 执行以下命令配置环境变量。

source bigdata_env

步骤3 如果当前集群已启用Kerberos认证，执行以下命令认证当前用户，当前用户需要具有创建HBase表的权限。如果当前集群未启用Kerberos认证，则无需执行此命令。

kinit *MRS集群用户*

步骤4 执行HBase组件的客户端命令。

hbase shell

----结束

11.20 HBase shell 客户端在使用中有 INFO 信息打印在控制台导致显示混乱

用户问题

在使用HBase shell中会有INFO信息打印在控制台，导致显示混乱，用户在输入命令前需要按回车符清理控制台。如下图所示：

```
hbase:013:0>
hbase:014:0>
hbase:015:0>
hbase:016:0> 2022-07-04 09:49:45,712 INFO  [ReadOnlyZKClient-[illegible]h:2181,no[illegible]:2181,nod[illegible]181@0x284990de] zookeeper.ZooKeeper: Connection: 0x1400001b098c5f4d closed
2022-07-04 09:49:45,712 INFO  [ReadOnlyZKClient-[illegible]90de-EventThread] zookeeper.ClientCnxn: EventThread shut down for connection: 0x1400001b098c5f4d

hbase:017:0>
```

原因分析

- HBase客户端中默认日志打印设置为“INFO,console”，所以在使用期间会有INFO日志输出到控制台，影响HBase shell窗口的显示。
- HBase客户端命令繁多，例如：**hbase shell**、**hbase hbck**、**hbase org.apache.hadoop.hbase.mapreduce.RowCounter**等，且后续还会增加。部分命令的输出为INFO打印，如果直接把INFO关闭会导致部分命令输出结果丢失。例如：RowCounter输出结果为INFO类型：

```
2022-06-08 18:01:52,490 INFO  [main] mapreduce.Job: Running job: job_1654133272184_0001
2022-06-08 18:02:03,713 INFO  [main] mapreduce.Job: Job job_1654133272184_0001 running in uber mode : false
2022-06-08 18:02:03,714 INFO  [main] mapreduce.Job:  map 0% reduce 0%
2022-06-08 18:04:55,491 INFO  [main] mapreduce.Job:  map 100% reduce 0%
2022-06-08 18:04:55,496 INFO  [main] mapreduce.Job: Job job_1654133272184_0001 completed successfully
2022-06-08 18:04:55,581 INFO  [main] mapreduce.Job: te-milliseconds taken by all map tasks=694030336
        Map-Reduce Framework
                Map input records=24000000
                Map output records=0
                Input split bytes=114
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=396
                CPU time spent (ms)=66770
                Physical memory (bytes) snapshot=468553728
                Virtual memory (bytes) snapshot=5754421248
                Total committed heap usage (bytes)=168820736
                Peak Map Physical memory (bytes)=513437696
                Peak Map Virtual memory (bytes)=5989408768
        HBaseCounters
                BYTES_IN_REMOTE_RESULTS=0
                BYTES_IN_RESULTS=2277333228
                MILLIS_BETWEEN_NEXTS=164002
                NOT_SERVING_REGION_EXCEPTION=0
                NUM_SCANNER_RESTARTS=0
                NUM_SCAN_RESULTS_STALE=0
                REGIONS_SCANNED=1
                REMOTE_RPC_CALLS=0
                REMOTE_RPC_RETRIES=0
                ROWS_FILTERED=0
                ROWS_SCANNED=24000000
                RPC_CALLS=240001
                RPC_RETRIES=0
        org.apache.hadoop.hbase.mapreduce.RowCounter$RowCounterMapper$Counters
                ROWS=24000000
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=0
[root@[illegible] opt]#
```

处理步骤

步骤1 使用**root**用户登录安装HBase客户端的节点。

步骤2 在“*HBase客户端安装目录*/HBase/component_env”文件中添加如下信息：

export HBASE_ROOT_LOGGER=INFO,RFA

把日志输出到日志文件中，后期如果使用**hbase org.apache.hadoop.hbase.mapreduce.RowCounter**等命令，执行结果请在日志文件“*HBase客户端安装目录*/HBase/hbase/logs/hbase.log”中查看。

步骤3 切换到HBase客户端安装目录，执行以下命令使配置生效。

cd *HBase客户端安装目录*

source HBase/component_env

----结束

11.21 RegionServer 剩余内存不足导致 HBase 服务启动失败

用户问题

RegionServer剩余内存不足导致HBase服务启动失败。

原因分析

RegionServer启动时节点剩余内存不足，导致无法启动实例。排查步骤如下：

1. 登录Master节点，到“/var/log/Bigdata”查找HBase相关日志，HMaster的日志中报错“connect regionserver timeout ”。
2. 登录到**1**中HMaster连接不上的RegionServer节点，到“/var/log/Bigdata”查找HBase相关日志，RegionServer报错“error=’Cannot allocate memory’(errno=12)”。
3. 根据**2**报错判断由于RegionServer内存不足导致RegionServer启动失败。

处理步骤

步骤1 登录报错的RegionServer节点，执行以下命令查看节点剩余内存：

free -g

步骤2 执行**top**命令查看节点内存使用情况。

步骤3 根据**top**提示结束内存占用多的进程（内存占用多并且非MRS自身组件的进程），并重新启动HBase服务。

说明

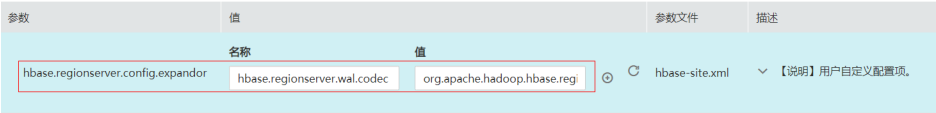
集群的Core节点除了MRS组件运行占用外，Yarn上的作业还会被分配到节点运行，占用节点内存。若是由于Yarn作业占用内存多导致组件无法正常启动时，建议扩容Core节点。

----结束

11.22 集群扩容之后新节点 HRegionServer 启动失败

问题现象

- 集群扩容完成之后，新节点的HRegionserver启动失败，一直处于异常状态，无法正常提供服务。
- 登录故障RegionServer所在节点，**jps**查看RegionServer进程没有启动，在Manager上手动重启故障的RegionServer实例失败，查看对应RegionServer节点/var/log/Bigdata/hbase/rs/hbase-omm-regionserver-node-ana-coreqRvt.log日志，有报错信息“ClassNotFound：org.apache.hadoop.hbase.regionserver.wal.IndexedWALEditCodec”。
- Manager上，可以看到RegionServer有如下自定义配置：



原因分析

用户配置了Phoenix的索引功能，由于新节点没有Phoenix对应的jar包，导致找不到类，启动失败。

处理步骤

步骤1 登录到正常的RegionServer节点，执行以下命令。

grep -Rn ‘org.apache.hadoop.hbase.regionserver.wal.IndexedWALEditCodec’ /opt/Bigdata/MRS_Current/1_16_RegionServer/

查询出/opt/Bigdata/MRS_Current/1_16_RegionServer/install/hbase/lib目录下面有两个Phoenix包（phoenix-4.14.1-server.jar和phoenix-core-4.14.1.jar）包含这个类。

步骤2 使用**scp**命令将上述两个包拷贝到异常RegionServer节点上相同目录下，重启故障的RegionServer，恢复正常。

----结束

11.23 HBase 文件丢失导致 Region 状态长时间为 RIT

用户问题

数据迁移到新集群后，HBase的Region状态为RIT或FAILED_OPEN。

原因分析

1. 登录到集群的Core节点查看RegionServer日志，报错：
   - “Region‘3b3ae24c65fc5094bc2acfebaa7a56de’”打开失败，Region状态被设置为FAILED_OPEN。

– “Region‘b7b3faab86527b88a92f2a248a54d3dc’”下的一个文件找不到报错FileNotFoundException。

2. 在HBase客户端中执行**hbase hbck**命令检查Region，报错：

```
ERROR: Found lingering reference file hdfs://hacluster/hbase/news_user_actions/
3b3ae24c65fc5094bc2acfebaa7a56de/meta/
0f47cda55fa44cf9aa2599079894aed6.b7b3faab86527b88a92f2a248a54d3dc
```

从引用文件可以看出，‘3b3ae24c65fc5094bc2acfebaa7a56de’Region是‘ b7b3faab86527b88a92f2a248a54d3dc’的子Region，子Region引用了父Region的文件，父Region的文件在迁移过程中因为未知原因丢失了，导致子Region找不到引用文件。

这种异常并不能通过简单的重试可解决，所以Region会长时间处于RIT状态。

处理步骤

步骤1 在HDFS客户端执行以下命令删除2查看到的残留文件：

hdfs dfs -rmr *残留文件所在的HDFS路径*

例如：

hdfs dfs -rmr */hbase/news_user_actions/3b3ae24c65fc5094bc2acfebaa7a56de/meta/0f47cda55fa44cf9aa2599079894aed6.b7b3faab86527b88a92f2a248a54d3dc*

----结束

12 使用 HDFS

12.1 修改 RPC 端口后 HDFS NameNode 实例都变为备状态

用户问题

通过页面更改NameNode的RPC端口，随后重启HDFS服务，出现所有NameNode一直是备状态，导致集群异常。

问题现象

所有NameNode都是备状态，导致集群异常。

原因分析

集群安装启动后，如果修改NameNode的RPC端口，则需要重新格式化Zkfc服务来更新zookeeper上的节点信息。

处理步骤

步骤1 登录Manager，停止HDFS服务。

说明

在停止HDFS时，建议不要停止相关服务。

步骤2 停止成功后，登录到被修改了RPC端口的Master节点。

说明

如果两个Master节点都被修改了RPC端口，则只需登录其中一个修改即可。

步骤3 执行**su - omm**命令切换到**omm**用户。

说明

如果是安全集群，需要执行**kinit hdfs**命令进行认证。

步骤4 执行如下命令，将环境变量脚本加载到环境中。

cd ${BIGDATA_HOME}/*MRS_X.X.X/1_8_***Zkfc/etc**

source ${BIGDATA_HOME}/*MRS_X.X.X*/install/FusionInsight-Hadoop-*3.1.1*/hadoop/sbin/exportENV_VARS.sh

说明

命令中的“*MRS_X.X.X*”和“1_8”根据实际版本而定。

步骤5 加载完成后，执行如下命令，格式化Zkfc。

cd ${HADOOP_HOME}/bin

./hdfs zkfc -formatZK

步骤6 格式化成功后，在Manager页面“重启”HDFS服务。

说明

如果更改了NameNode的RPC端口，则之前安装的所有客户端都需要刷新配置文件。

----结束

12.2 通过公网 IP 连接使用 HDFS 客户端报错

问题现象

通过公网IP连接主机，不能使用HDFS客户端，运行**hdfs**命令后提示“-bash: hdfs: command not found”。

原因分析

用户登录Master节点执行命令之前，未设置环境变量。

处理步骤

步骤1 以**root**用户登录任意一个Master节点。

步骤2 执行**source** *客户端安装目录*/**bigdata_env**命令，设置环境变量。

步骤3 执行**hdfs**命令即可成功使用HDFS客户端。

----结束

12.3 使用 Python 远程连接 HDFS 的端口失败

用户问题

使用Python远程连接HDFS的端口失败，如何解决?

问题现象

用户使用Python远程连接HDFS的50070端口失败。

原因分析

HDFS开源3.0.0以下版本的默认端口为50070，3.0.0及以上的默认端口为9870。用户使用的端口和HDFS版本不匹配导致连接端口失败。

步骤1 登录集群的主Master节点。

步骤2 执行**su - omm**命令，切换到omm用户。

步骤3 执行**/opt/Bigdata/om-0.0.1/sbin/queryVersion.sh**或者**sh ${BIGDATA_HOME}/om-server/om/sbin/queryVersion.sh**命令，查看集群中的HDFS版本号。

根据版本号确认开源组件的端口号，查询开源组件的端口号可参考**开源组件端口列表**，获取对应版本的HDFS端口号。

步骤4 执行**netstat -anp|grep** *${port}*命令，查看组件的默认端口号是否存在。

如果不存在，说明用户修改了默认的端口号。请修改为默认端口，再重新连接HDFS。

如果存在，请联系技术服务。

说明

- ${ port }：表示与组件版本相对应的组件默认端口号。
- 如果用户修改了默认端口号，请使用修改后的端口号连接HDFS。不建议修改默认端口号。

----结束

12.4 HDFS 容量达到 100%导致上层服务 HBase、Spark 等不可用

用户问题

集群的HDFS容量使用达到100%，HDFS服务状态为只读，导致上层服务HBase、Spark等上报服务不可用告警。

问题现象

HDFS使用容量100%，磁盘容量只使用85%左右，HDFS服务状态为只读，导致上层服务HBase、Spark等上报服务不可用。

原因分析

当前NodeManager和DataNode共数据盘使用，MRS默认预留15%的数据磁盘空间给非HDFS使用，可通过HDFS参数“dfs.datanode.du.reserved.percentage”修改百分比来控制具体的磁盘占比。

当HDFS磁盘使用100%之后，可通过降低“dfs.datanode.du.reserved.percentage”百分比来恢复业务，再进行磁盘扩容。

处理步骤

步骤1 登录集群任意Master节点。

步骤2 执行**source /opt/client/bigdata_env**命令，初始化环境变量。

说明

如果是安全集群，则需要执行**kinit -kt <keytab file> <pricipal name>** 进行认证。

步骤3 执行**hdfs dfs -put ./startDetail.log /tmp**命令，测试HDFS写文件失败。

```
19/05/12 10:07:32 WARN hdfs.DataStreamer: DataStreamer Exception
org.apache.hadoop.ipc.RemoteException(java.io.IOException): File /tmp/startDetail.log._COPYING_ could
only be replicated to 0 nodes instead of minReplication (=1).  There are 3 datanode(s) running and no
node(s) are excluded in this operation.
```

步骤4 执行**hdfs dfsadmin -report**命令，检查HDFS使用容量，发现已经达到100%。

```
Configured Capacity: 5389790579100 (4.90 TB)
Present Capacity: 5067618628404 (4.61 TB)
DFS Remaining: 133350196 (127.17 MB)
DFS Used: 5067485278208 (4.61 TB)
DFS Used%: 100.00%
Under replicated blocks: 10
Blocks with corrupt replicas: 0
Missing blocks: 0
Missing blocks (with replication factor 1): 0
Pending deletion blocks: 0
```

步骤5 当HDFS使用容量达到100%时，通过HDFS参数“dfs.datanode.du.reserved.percentage”修改百分比来控制具体的磁盘占比。

1. 登录Manager进入服务配置页面。
   - MRS Manager界面操作入口：登录MRS Manager，依次选择 “服务管理 > HDFS> 配置”。
   - FusionInsight Manager界面操作入口：登录FusionInsight Manager，选择“集群 > 服务 >HDFS > 配置”。
2. 选择“全部配置”，在搜索框中搜索“dfs.datanode.du.reserved.percentage”。
3. 修改此参数的取值为“10”。

步骤6 修改完成后，扩容Core节点的磁盘个数。

----结束

12.5 启动 HDFS 和 Yarn 服务报错“Permission denied”

用户问题

启动HDFS和Yarn时报错。

问题现象

无法启动HDFS、Yarn服务组件，报错内容：/dev/null Permission denied。

```
[2018-11-16 08:52:57] Start service 'ServiceName: Yarn'.
[2018-11-16 08:52:57] Start role 'ROLE[name: ResourceManager]'.
[2018-11-16 08:52:57] Start role 'ROLE[name: NodeManager]'.
[2018-11-16 08:52:57] Start role instance 'ResourceManager#192.168.0.23@node-master2-CMCgr' .
[2018-11-16 08:52:57] Start role instance 'ResourceManager#192.168.0.59@node-master1-bdWZs' .
[2018-11-16 08:52:57] Start role instance 'NodeManager#192.168.0.37@node-core-gKPas' .
[2018-11-16 08:52:57] Start role instance 'NodeManager#192.168.0.137@node-core-qFOXF' .
[2018-11-16 08:52:57] Start role instance 'NodeManager#192.168.0.135@node-core-nDKml' .
[2018-11-16 08:52:57] Start the role instance for 'ROLE[name: ResourceManager]' successfully.
[2018-11-16 08:52:57] Start the role instance for 'ROLE[name: ResourceManager]' successfully.
[2018-11-16 08:52:57] Start the role instance for 'ROLE[name: NodeManager]' successfully.
[2018-11-16 08:52:57] Start the role instance for 'ROLE[name: NodeManager]' successfully.
[2018-11-16 08:52:57] Start the role for 'ServiceName: Yarn' successfully.
Fail to prepare to start role instance 'NodeManager#192.168.0.135@node-core-
nDKml'[ScriptExecutionResult=ScriptExecutionResult [exitCode=1, output=, errMsg=/etc/bashrc: line 84: /dev/null:
Permission denied
```

原因分析

用户修改了虚机系统的“/dev/null”的权限值为775。

```
70  cd ..
71  ll
72  chmod -R 775 /dev/
73  ll
74  chmod -r 775 dbdata_om/
75  ll
76  chmod -r 770 dbdata_om/
77  ll
78  chmod -r 777 dbdata_om/
79  ll
80  cd ..
81  ll
```

处理步骤

步骤1 以**root**用户登录集群的任意一个Master节点。

步骤2 登录成功后，执行**chmod 666 /dev/null**命令，修改“/dev/null”的权限值为“666”。

步骤3 执行**ls -al /dev/null**命令，查看修改的“/dev/null”权限值是否为“666”，如果不是，需要修改为“666”。

步骤4 修改成功后，重新启动HDFS和Yarn组件。

----结束

12.6 HDFS 用户可在其他用户目录下创建或删除文件

问题现象

在使用MRS服务时，某个用户可以在其他用户的HDFS目录下面删除或者创建文件。

原因分析

客户配置的用户具有ficommon组的权限，所以可以对HDFS任意操作。需要移除用户的ficommon组权限。

处理步骤

步骤1 以**root**用户登录集群的Master节点。

步骤2 执行**id** *${用户名}*命令，显示用户组信息，确认是否有“ficommon”组权限。

如果存在“ficommon”组权限，请执行**步骤3**。如果不存在，请联系技术服务。

说明

用户名为出现HDFS权限设置问题的用户。

步骤3 执行**gpasswd -d** *${用户名}* **ficommon**命令，删除该用户的“ficommon”组权限。

步骤4 执行成功后，登录Manager修改参数。

MRS Manager界面操作（适用MRS 3.x之前版本）：

1. 登录MRS Manager，在MRS Manager页面，选择“服务管理 > HDFS > 服务配置”。
2. “参数类别”选择“全部配置”，在搜索框中输入“dfs.permissions.enabled”，修改为“true”。
3. 修改完成后，单击“保存配置”，并重启HDFS服务。

FusionInsight Manager界面操作（适用MRS 3.x及之后版本）：

1. 登录FusionInsight Manager。选择“集群 > 服务 > HDFS > 配置 > 全部配置”。
2. 在搜索框中输入“dfs.permissions.enabled”，修改为“true”。
3. 修改完成后，单击“保存”，并重启HDFS服务。

MRS集群详情页操作（适用MRS 1.9.2及之后版本）：

1. 登录MRS控制台，选择“组件管理 > HDFS > 服务配置”。
2. “参数类别”选择“全部配置”，在搜索框中输入“dfs.permissions.enabled”，修改为“true”。
3. 修改完成后，单击“保存配置”，并重启HDFS服务。

----结束

12.7 HDFS 的 DataNode 一直显示退服中

用户问题

HDFS的DataNode一直显示退服中。

问题现象

HDFS的某个DataNode退服（或者对Core节点进行缩容）任务失败，但是DataNode在任务失败后一直处于退服中的状态。

原因分析

在对HDFS的某个DataNode进行退服（或者对core节点进行缩容）过程中，因为Master节点重启或者nodeagent进程意外退出等情况出现，使得退服（或缩容）任务失败，并且没有进行黑名单清理。此时DataNode节点会一直处于退服中的状态，需要人工介入进行黑名单清理。

处理步骤

步骤1 进入服务实例界面。

MRS Manager界面操作：

登录MRS Manager，在MRS Manager页面，选择“服务管理 > HDFS > 实例”。

FusionInsight Manager界面操作：

对于MRS 3.x及后续版本集群：也可登录FusionInsight Manager。选择“集群 > 服务 > HDFS > 实例”。

也可登录MRS控制台，选择“组件管理 > HDFS > 实例”。

步骤2 查看HDFS服务实例状态，找到一直处于退服中的DataNode，复制这个DataNode的IP地址。

步骤3 登录Master1节点的后台，执行**cd ${BIGDATA_HOME}/MRS_*/1_*_NameNode/etc/**命令进入黑名单目录。

步骤4 执行**sed -i "/^*IP*$/d" excludeHosts**命令清理黑名单中的故障DataNode信息，该命令中IP替换为**步骤2**中查询到的故障DataNode的IP地址，其中不能有空格。

步骤5 如果有两个Master节点，请在Master2节点上同样执行**步骤3**和**步骤4**。

步骤6 在Master1节点执行如下命令初始化环境变量。

source *客户端安装目录*/**bigdata_env**

步骤7 如果当前集群已启用Kerberos认证，执行以下命令认证当前用户。如果当前集群未启用Kerberos认证，则无需执行此命令。

kinit *具有HDFS操作权限的业务用户*

步骤8 在Master1节点执行如下命令刷新HDFS的黑名单。

hdfs dfsadmin -refreshNodes

步骤9 执行**hdfs dfsadmin -report**命令来查看各个DataNode的状态，确认中查到的IP对应的DataNode已经恢复为“Normal”状态。

图 12-1 DataNode 的状态

```
Name: [redacted]:9866 (node-ana-coreoYfm)
Hostname: node-ana-coreoYfm
Rack: /default/rack0
Decommission Status : Normal
Configured Capacity: 105554829312 (98.31 GB)
DFS Used: 1225715740 (1.14 GB)
Non DFS Used: 3045261284 (2.84 GB)
DFS Remaining: 95361495372 (88.81 GB)
DFS Used%: 1.16%
DFS Remaining%: 90.34%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 10
Last contact: Thu Aug 15 15:53:17 CST 2019
Last Block Report: Thu Aug 15 12:12:46 CST 2019
Num of Blocks: 974
```

步骤10 进入服务实例界面。

MRS Manager界面操作：

登录MRS Manager，在MRS Manager页面，选择“服务管理 > HDFS > 实例”。

FusionInsight Manager界面操作：

对于MRS 3.x及后续版本集群：可登录FusionInsight Manager。选择“集群 > 服务 > HDFS > 实例”。

登录MRS控制台，选择“组件管理 > HDFS > 实例”。

步骤11 勾选一直处于退服中的DataNode实例，单击“更多 > 重启实例”。

步骤12 等待重启完成，确认DataNode是否恢复正常。

----结束

建议与总结

尽量不要在退服（或缩容）过程中重启节点等高危操作。

参考信息

无

12.8 内存不足导致 HDFS NameNode 启动失败

问题背景与现象

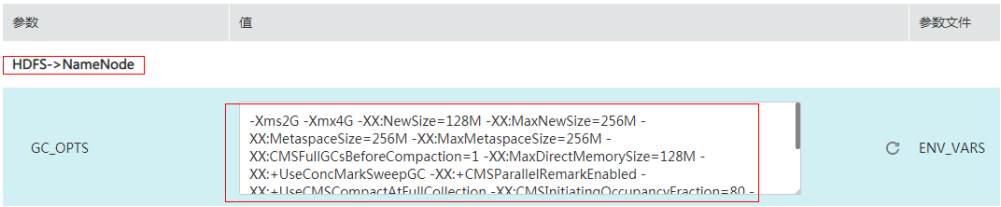
场景一：重启HDFS服务后，HDFS的状态为Bad，且NameNode实例状态异常，并且很久没有退出安全模式。

场景二：NameNode启动时，启动超时后启动失败，原生页面无法打开。

原因分析

1. 在NameNode运行日志（/var/log/Bigdata/hdfs/nn/hadoop-omm-namenode-XXX.log）中搜索“WARN”，可以看到有大量时间在垃圾回收，如下例中耗时较长63s。
```
2017-01-22 14:52:32,641 | WARN  | org.apache.hadoop.util.JvmPauseMonitor$Monitor@1b39fd82 |
Detected pause in JVM or host machine (eg GC): pause of approximately 63750ms
GC pool 'ParNew' had collection(s): count=1 time=0ms
GC pool 'ConcurrentMarkSweep' had collection(s): count=1 time=63924ms | JvmPauseMonitor.java:189
```
2. 分析NameNode日志“/var/log/Bigdata/hdfs/nn/hadoop-omm-namendoe-XXX.log”，可以看到NameNode在等待块上报，且总的Block个数过多，如下例中是3629万。
```
2017-01-22 14:52:32,641 | INFO  | IPC Server handler 8 on 25000 | STATE* Safe mode ON.
The reported blocks 29715437 needs additional 6542184 blocks to reach the threshold 0.9990 of total
blocks 36293915.
```
3. 打开Manager页面，查看NameNode的GC_OPTS参数配置如下：

图 12-2 查看 NameNode 的 GC_OPTS 参数配置



4. NameNode内存配置和数据量对应关系参考**表12-1**。

表 12-1 NameNode 内存配置和数据量对应关系

文件对象数量	参考值
10,000,000	“-Xms6G -Xmx6G -XX:NewSize=512M -XX:MaxNewSize=512M”
20,000,000	“-Xms12G -Xmx12G -XX:NewSize=1G -XX:MaxNewSize=1G”
50,000,000	“-Xms32G -Xmx32G -XX:NewSize=2G -XX:MaxNewSize=3G”
100,000,000	“-Xms64G -Xmx64G -XX:NewSize=4G -XX:MaxNewSize=6G”
200,000,000	“-Xms96G -Xmx96G -XX:NewSize=8G -XX:MaxNewSize=9G”
300,000,000	“-Xms164G -Xmx164G -XX:NewSize=12G -XX:MaxNewSize=12G”

解决办法

步骤1 按照规格修改NameNode的内存参数，如这里3600万block，将内存参数调整为“-Xms32G -Xmx32G -XX:NewSize=2G -XX:MaxNewSize=3G”。

步骤2 重启一个NameNode，确认该NameNode可以正常启动。

步骤3 重启另一个NameNode，确认页面状态恢复。

----结束

12.9 ntpdate 修改时间导致 HDFS 出现大量丢块

问题背景与现象

1. 用ntpdate修改了集群时间，修改时未停止集群，修改后HDFS进入安全模式，无法启动。
2. 退出安全模式后启动，hfck检查丢了大概1 TB数据。

原因分析

1. 查看NameNode原生页面发现有大量的块丢失。

图 12-3 块丢失

```
There are 41491 missing blocks. The following files may be corrupted:

blk_1090519588 /user/etlhadoop/struct_data/uds_data/PRS/20180130/DCM_PRS_PDWTMDTL_S_000_input/1/ccw-20180130-pdwtmdtl-
023_022.bin_7
blk_1090519796 /user/etlhadoop/struct_data/uds_data/GCM/20180130/DCM_GCM_PNDLTA200211_H_output/1/part-m-00010
blk_1090520189 /user/hive/warehouse/prs_mc.db/dcm_prs_pdwtmdtl_s/pt_dt=2018-01-30/part-m-00004
blk_1082131961 /user/hive/warehouse/cas_mc.db/dcm_cas_nthpatel_h/end_dt=2017-12-31/000004_0
blk_1082132310 /user/hive/warehouse/crl_mc.db/dcm_crl_ecs_tb2045_s/pt_dt=2017-12-31/000005_0
blk_1082132604 /user/hive/warehouse/crl_mc.db/dcm_crl_ecs_tb2045_s/pt_dt=2017-12-31/000040_0
blk_1090521279 /user/hive/warehouse/gcm_mc.db/dcm_gcm_pndlta200211_h/end_dt=2018-01-30/000006_0
blk_1090521284 /user/hive/warehouse/gcm_mc.db/dcm_gcm_pndlta200211_h/end_dt=2018-01-30/000012_0
blk_1090521427 /user/hive/warehouse/pis_mc.db/dcm_pis_lthpcdtl_h/end_dt=2018-01-30/000080_0
blk_1090521473 /user/hive/warehouse/pis_mc.db/dcm_pis_lthpcdtl_h/end_dt=2018-01-30/000016_0
blk_1082133176 /user/hive/warehouse/cas_mc.db/dcm_cas_kffpbat_s/pt_dt=2017-12-31/part-m-00006
blk_1090522261 /user/etlhadoop/struct_data/uds_data/ECS/20180130/DCM_ECS_TB1170_S_000_input/1/ciw-20180130-hdwb1171-022_032.bin_16
blk_1090522656 /user/etlhadoop/struct_data/uds_data/ECS/20180130/DCM_ECS_TB1170_S_output/1/part-m-00007
blk_1090522747 /user/hive/warehouse/gcm_mc.db/dcm_gcm_zassure_change_detail_s/pt_dt=2018-01-31/000002_0
blk_1082134372 /user/hive/warehouse/bcs_mc.db/dcm_bcs_bthrsism_h/pt_dt=2017-12-31/part-m-00006
blk_1090523585 /user/hive/warehouse/ecs_mc.db/dcm_ecs_tb1170_s/pt_dt=2018-01-30/000002_0
blk_1090523811 /user/hive/warehouse/nae_mc.db/dcm_nae_nfpjnl_s/pt_dt=2018-01-30/part-m-00005
blk_1082135337 /user/hive/warehouse/bcs_mc.db/dcm_bcs_bthrsism_h/pt_dt=2017-12-31/part-m-00022
blk_1090524043 /user/hive/warehouse/nae_mc.db/dcm_nae_nfpjnl_s/pt_dt=2018-01-30/part-m-00016
blk_1082136206 /user/hive/warehouse/bcs_mc.db/dcm_bcs_bthrsism_h/pt_dt=2017-12-31/part-m-00038
blk_1090525355 /user/hive/warehouse/bdsp_bcas_act.db/bcs_jzcs_detail/pt_dt=2017-11-30/000006_0
blk_1090526191 /user/hive/warehouse/bdsp_bcas_act.db/bcs_jzcs_detail/pt_dt=2017-11-30/000008_0
blk_1090526995 /user/hive/warehouse/bdsp_bcas_act.db/bcs_jzcs_detail/pt_dt=2017-11-30/000014_0
blk_1082140552 /user/hive/warehouse/cc8_mc.db/m01_cc8_corp_cust_mgr/pt_dt=2017-12-31/000001_0
blk_1090529399 /user/hive/warehouse/bdsp_bcas_act.db/bcs_jzcs_middle_t/pt_dt=2017-11-30/000017_0
blk_1090529420 /user/hive/warehouse/bdsp_bcas_act.db/bcs_jzcs_middle_t/pt_dt=2017-11-30/000014_0
blk_1082141596 /user/hive/warehouse/asa_mc.db/t80_asa_bcas_agt_stat/pt_dt=2017-12-31/000032_0
blk_1082141631 /user/hive/warehouse/asa_mc.db/t80_asa_bcas_agt_stat/pt_dt=2017-12-31/000003_0
blk_1082142345 /user/hive/warehouse/sum_mc.db/c00_prod_level_overview_h/pt_dt=2017-12-31/000000_0_copy_1514441582192
blk_1090531076
/user/etlhadoop/struct_data/uds_data/GCM/20180131/DCM_GCM_DEDUN_STOP_PARA_S_000_input/1/CMA_DEDUN_STOP_PARA0111800000-011-
20180131.BIN_11_UTF
blk_1090531330 /user/hive/warehouse/gcc_mc.db/dcm_gcc_zcorp_motor_info_s/pt_dt=2018-01-31/000011_0
blk_1090531342 /user/hive/warehouse/gcc_mc.db/dcm_gcc_zcorp_motor_info_s/pt_dt=2018-01-31/000002_0
blk_1090531494
/user/etlhadoop/struct_data/uds_data/GCM/20180131/DCM_GCM_ZMORTGAGE_PROJECT_STAT_S_000_input/1/CMA_ZMORTGAGE_PROJECT_STAT0050100000-
```

2. 查看原生页面Datanode Information发现显示的DataNode节点数和实际的相差10个节点。

图 12-4 查看 DataNode 节点数

Hadoop | Overview | Datanodes | Datanode Volume Failures | Snapshot | Startup Progress | Utilities | Logout

Summary

Security is on.

Safemode is off.

14442 files and directories, 13907 blocks = 28349 total filesystem object(s).

Heap Memory used 495.63 MB of 1.99 GB Heap Memory. Max Heap Memory is 3.98 GB.

Non Heap Memory used 104.5 MB of 107.94 MB Commited Non Heap Memory. Max Non Heap Memory is 1.36 GB.

Configured Capacity:	112.09 GB
DFS Used:	15.33 GB (13.68%)
Non DFS Used:	18.56 GB
DFS Remaining:	78.2 GB (69.77%)
Block Pool Used:	15.33 GB (13.68%)
DataNodes usages% (Min/Median/Max/stdDev):	13.56% / 13.73% / 13.73% / 0.08%
Live Nodes	3 (Decommissioned: 0)
Dead Nodes	0 (Decommissioned: 0)
Decommissioning Nodes	0

3. 查看DateNode运行日志“/var/log/Bigdata/hdfs/dn/hadoop-omm-datanode-主机名.log”，发现如下错误信息。

   重要错误信息Clock skew too great

图 12-5 DateNode 运行日志错误

```
    at org.apache.hadoop.ipc.Client.call(Client.java:1486)
    at org.apache.hadoop.ipc.Client.call(Client.java:1447)
    at org.apache.hadoop.ipc.ProtobufRpcEngine$Invoker.invoke(ProtobufRpcEngine.java:229)
    at com.sun.proxy.$Proxy14.versionRequest(Unknown Source)
    at org.apache.hadoop.hdfs.protocolPB.DatanodeProtocolClientSideTranslatorPB.versionRequest(DatanodeProtocolClientSideTranslatorPB.java:273)
    at org.apache.hadoop.hdfs.server.datanode.BPServiceActor.retrieveNamespaceInfo(BPServiceActor.java:187)
    at org.apache.hadoop.hdfs.server.datanode.BPServiceActor.connectToNNAndHandshake(BPServiceActor.java:237)
    at org.apache.hadoop.hdfs.server.datanode.BPServiceActor.run(BPServiceActor.java:689)
    at java.lang.Thread.run(Thread.java:745)
Caused by: GSSException: No valid credentials provided (Mechanism level: Clock skew too great (37))
    at sun.security.jgss.krb5.Krb5Context.initSecContext(Krb5Context.java:770)
    at sun.security.jgss.GSSContextImpl.initSecContext(GSSContextImpl.java:248)
    at sun.security.jgss.GSSContextImpl.initSecContext(GSSContextImpl.java:179)
    at com.sun.security.sasl.gsskerb.GssKrb5Client.evaluateChallenge(GssKrb5Client.java:192)
    ... 20 more
Caused by: KrbException: Clock skew too great (37)
    at sun.security.krb5.KrbKdcRep.check(KrbKdcRep.java:88)
    at sun.security.krb5.KrbTgsRep.<init>(KrbTgsRep.java:87)
    at sun.security.krb5.KrbTgsReq.getReply(KrbTgsReq.java:259)
    at sun.security.krb5.KrbTgsReq.sendAndGetCreds(KrbTgsReq.java:270)
    at sun.security.krb5.internal.CredentialsUtil.serviceCreds(CredentialsUtil.java:302)
    at sun.security.krb5.internal.CredentialsUtil.acquireServiceCreds(CredentialsUtil.java:120)
    at sun.security.krb5.Credentials.acquireServiceCreds(Credentials.java:458)
    at sun.security.jgss.krb5.Krb5Context.initSecContext(Krb5Context.java:693)
```

解决办法

步骤1 修改在原生页面查看不到的10个数据节点的时间。

步骤2 在Manager页面重启对应的DataNode实例。

----结束

12.10 DataNode 概率性出现 CPU 占用接近 100%导致节点丢失

问题背景与现象

DataNode概率性出现CPU占用接近100%，导致节点丢失（ssh连得很慢或者连不上）。

图 12-6 DataNode 出现 CPU 占用接近 100%

```
  PID USER      PR  NI  VIRT  RES  SHR S  %CPU %MEM    TIME+  COMMAND
60636 omm       20   0 9445m 1.7g  16m S   299  1.3  1952:06 javc.exec -Dproc_datanode -outfile /var/log/Bigdata/hdfs/dn/javc.out -errfile /var/log/Bigdata/hdfs/dn/javc.err -pidfil
02428 ossadm    20   0 18116 3784 1828 R   155  0.0  1:17.63 /opt/tsp/manager/rtsp/python/bin/python /opt/tsp/manager/agent-1.3.10.200/tools/pyscript/sysappctrl.pyc -cmd status -te
02410 ossadm    20   0 55016 8048 2836 R   155  0.0  1:59.80 /opt/tsp/manager/rtsp/python/bin/python /opt/tsp/manager/agent-1.3.10.200/tools/pyscript/watchdog.pyc -cmd status
02412 ossadm    20   0 36752 5912 2340 R   155  0.0  1:50.32 /opt/tsp/manager/rtsp/python/bin/python /opt/tsp/manager/agent-1.3.10.200/tools/pyscript/sysappctrl.pyc -cmd procinfo -
02484 omm       20   0 12800 1476 1124 R   155  0.0  0:10.73 /bin/bash -c                    /jdk1.7.0_80//bin/java -server -Xmx1024m -Djava.io.tmpdir=/export/data1/yarn/nm/localdi
02341 ossadm    20   0 57760 8688 3000 R   139  0.0  3:29.41 /opt/tsp/manager/rtsp/python/bin/python /opt/tsp/manager/agent/tools/pyscript/syscollector.pyc sys /opt/tsp/manager/var
02531 omm       20   0 11176  640  468 R   106  0.0  0:04.19 -bash -c echo $OMS_RUN_PATH
```

原因分析

1. DataNode有许多写失败的日志。

图 12-7 DataNode 写失败的日志

```
2015-08-31 11:29:34,184 | ERROR | DataXceiver for client DFSClient_NONMAPREDUCE_1675952887_23 at /192.168.8.40:44514 [Receiving block
BP-125271511-192.168.8.29-1440656260530:blk_1074766997_1034914] | TSP21:25009:DataXceiver error processing WRITE_BLOCK operation  src:
/192.168.8.40:44514 dst: /192.168.8.64:25009 | DataXceiver.java:258
java.io.IOException: Premature EOF from inputStream
        at org.apache.hadoop.io.IOUtils.readFully(IOUtils.java:194)
        at org.apache.hadoop.hdfs.protocol.datatransfer.PacketReceiver.doReadFully(PacketReceiver.java:213)
        at org.apache.hadoop.hdfs.protocol.datatransfer.PacketReceiver.doRead(PacketReceiver.java:134)
        at org.apache.hadoop.hdfs.protocol.datatransfer.PacketReceiver.receiveNextPacket(PacketReceiver.java:109)
        at org.apache.hadoop.hdfs.server.datanode.BlockReceiver.receivePacket(BlockReceiver.java:446)
        at org.apache.hadoop.hdfs.server.datanode.BlockReceiver.receiveBlock(BlockReceiver.java:707)
        at org.apache.hadoop.hdfs.server.datanode.DataXceiver.writeBlock(DataXceiver.java:748)
        at org.apache.hadoop.hdfs.protocol.datatransfer.Receiver.opWriteBlock(Receiver.java:124)
        at org.apache.hadoop.hdfs.protocol.datatransfer.Receiver.processOp(Receiver.java:71)
        at org.apache.hadoop.hdfs.server.datanode.DataXceiver.run(DataXceiver.java:240)
        at java.lang.Thread.run(Thread.java:745)
2015-08-31 11:29:35,147 | INFO  | DataXceiver for client DFSClient_NONMAPREDUCE_-402997805_1 at /192.168.8.30:59449 [Sending block BP-
125271511-192.168.8.29-1440656260530:blk_1074181856_446655] | src: /192.168.8.64:25009, dest: /192.168.8.30:59449, bytes: 16826, op:
HDFS_READ, cliID: DFSClient_NONMAPREDUCE_-402997805_1, offset: 0, srvID: 9d1d30a5-046d-438b-83c9-2c6c54c6bd12, blockid: BP-125271511-
192.168.8.29-1440656260530:blk_1074181856_446655, duration: 78832 | BlockSender.java:738
2015-08-31 11:29:35,269 | INFO  | org.apache.hadoop.util.JvmPauseMonitor$Monitor@551bd2a0 | Detected pause in JVM or host machine (eg
GC): pause of approximately 7480ms
No GCs detected | JvmPauseMonitor.java:172
2015-08-31 11:29:36,985 | INFO  | org.apache.hadoop.util.JvmPauseMonitor$Monitor@551bd2a0 | Detected pause in JVM or host machine (eg
GC): pause of approximately 1215ms
No GCs detected | JvmPauseMonitor.java:172
2015-08-31 11:29:43,067 | INFO  | DataXceiver for client DFSClient_NONMAPREDUCE_1675952887_23 at /192.168.8.33:35530 [Receiving block
BP-125271511-192.168.8.29-1440656260530:blk_1074767006_1034923] | Exception for BP-125271511-192.168.8.29-
1440656260530:blk_1074767006_1034923 | BlockReceiver.java:742
java.io.IOException: Premature EOF from inputStream
```

2. 短时间内写入大量文件导致这种情况，因此DataNode内存不足。

图 12-8 写入大量文件导致 DataNode 内存不足

```
Line 153101: 2015-08-31 11:24:29,313 | INFO  | org.apache.hadoop.util.JvmPauseMonitor$Monitor@551bd2a0 | Detected pause in JVM or host machine (eg GC): pause of approximately 1199ms
Line 153132: 2015-08-31 11:24:42,689 | WARN  | org.apache.hadoop.util.JvmPauseMonitor$Monitor@551bd2a0 | Detected pause in JVM or host machine (eg GC): pause of approximately 11273ms
Line 153135: 2015-08-31 11:24:45,810 | INFO  | org.apache.hadoop.util.JvmPauseMonitor$Monitor@551bd2a0 | Detected pause in JVM or host machine (eg GC): pause of approximately 1005ms
Line 153138: 2015-08-31 11:24:49,801 | INFO  | org.apache.hadoop.util.JvmPauseMonitor$Monitor@551bd2a0 | Detected pause in JVM or host machine (eg GC): pause of approximately 1067ms
Line 153155: 2015-08-31 11:25:10,167 | WARN  | org.apache.hadoop.util.JvmPauseMonitor$Monitor@551bd2a0 | Detected pause in JVM or host machine (eg GC): pause of approximately 12323ms
```

解决办法

步骤1 检查DataNode内存配置，以及机器剩余内存是否充足。

步骤2 增加DataNode内存，并重启DataNode。

----结束

12.11 单 NameNode 长期故障时如何使用客户端手动 checkpoint

问题背景与现象

在备NameNode长期异常的情况下，会积攒大量的editlog，此时如果重启HDFS或者主NameNode，主NameNode会读取大量的未合并的editlog，导致耗时启动较长，甚至启动失败。

原因分析

备NameNode会周期性做合并editlog，生成fsimage文件的过程叫做checkpoint。备NameNode在新生成fsimage后，会将fsimage传递到主NameNode。

说明

由于“备NameNode会周期性做合并editlog”，因此当备NameNode异常时，无法合并editlog，因此主NameNode在下次启动的时候，需要加载较多editlog，需要大量内存，并且耗时较长。

合并元数据的周期由以下参数确定，即如果NameNode运行30分钟或者HDFS操作100万次，均会执行checkpoint。

- dfs.namenode.checkpoint.period：checkpoint周期，默认1800s。
- dfs.namenode.checkpoint.txns：执行指定操作次数后执行checkpoint，默认1000000。

解决办法

在重启前，主动执行异常checkpoint合并主NameNode的元数据。

步骤1 停止业务。

步骤2 获取主NameNode的主机名。

步骤3 在客户端执行如下命令：

source /opt/client/bigdata_env

kinit *组件用户*

说明：“/opt/client”需要改为实际客户端的安装路径。

步骤4 执行如下命令，让主NameNode进入安全模式，其中linux22换为主NameNode的主机名。

hdfs dfsadmin -fs linux22:25000 -safemode enter

```
linux16:/opt/fi_client #  hdfs dfsadmin -fs linux22:25000   -safemode enter
17/04/26 18:38:30 WARN fs.FileSystem: "linux22:25000" is a deprecated filesystem name. Use "hdfs://linux22:25000/" instead.
17/04/26 18:38:32 INFO hdfs.PeerCache: SocketCache disabled.
Safe mode is ON
```

步骤5 执行如下命令，在主NameNode，合并editlog。

hdfs dfsadmin -fs linux22:25000 -saveNamespace

```
linux16:/opt/fi_client #  hdfs dfsadmin -fs linux22:25000   -saveNamespace
17/04/26 18:38:54 WARN fs.FileSystem: "linux22:25000" is a deprecated filesystem name. Use "hdfs://linux22:25000/" instead.
17/04/26 18:38:56 INFO hdfs.PeerCache: SocketCache disabled.
Save namespace successful
```

步骤6 执行如下命令，让主NameNode离开安全模式。

hdfs dfsadmin -fs linux22:25000 -safemode leave

```
linux16:/opt/fi_client #  hdfs dfsadmin -fs linux22:25000  -safemode leave
17/04/26 18:39:07 WARN fs.FileSystem: "linux22:25000" is a deprecated filesystem name. Use "hdfs://linux22:25000/" instead.
17/04/26 18:39:09 INFO hdfs.PeerCache: SocketCache disabled.
Safe mode is OFF
```

步骤7 检查是否真的合并完成。

cd /srv/BigData/namenode/current

检查先产生的fsimage是否是当前时间的，若是则表示已经合并完成

```
-rw------- 1 omm wheel 1048576 Apr 26 18:43 edits_inprogress_0000000000002083018
-rw------- 1 omm wheel  736657 Apr 26 15:46 fsimage_0000000000002071390
-rw------- 1 omm wheel      62 Apr 26 15:46 fsimage_0000000000002071390.md5
-rw------- 1 omm wheel  736657 Apr 26 16:46 fsimage_0000000000002075405
-rw------- 1 omm wheel      62 Apr 26 16:46 fsimage_0000000000002075405.md5
-rw------- 1 omm wheel  736410 Apr 26 17:46 fsimage_0000000000002079398
-rw------- 1 omm wheel      62 Apr 26 17:46 fsimage_0000000000002079398.md5
-rw------- 1 omm wheel       8 Apr 26 18:42 seen_txid
linux-20:/srv/BigData/namenode/current #
linux-20:/srv/BigData/namenode/current #
```

----结束

12.12 HDFS 文件读写时报错“Failed to place enough replicas”

问题背景与现象

当用户在HDFS上执行写操作时，出现“Failed to place enough replicas:expected…”信息。

原因分析

- DataNode的数据接收器不可用。

  此时DataNode会有如下日志：

```
2016-03-17 18:51:44,721 | WARN |
org.apache.hadoop.hdfs.server.datanode.DataXceiverServer@5386659f |
hadoopc1h2:25009:DataXceiverServer: | DataXceiverServer.java:158
java.io.IOException: Xceiver count 4097 exceeds the limit of concurrent xcievers: 4096
at org.apache.hadoop.hdfs.server.datanode.DataXceiverServer.run(DataXceiverServer.java:140)
at java.lang.Thread.run(Thread.java:745)
```

- DataNode的磁盘空间不足。
- DataNode的心跳有延迟。

解决办法

- 如果DataNode的数据接收器不可用，通过在Manager页面，增加HDFS参数“dfs.datanode.max.transfer.threads”的值解决。
- 如果没有足够的硬盘空间或者CPU，试着增加新的数据节点或确保资源是可用的（磁盘空间或CPU）。
- 如果网络问题，确保网络是可用的。

12.13 文件最大打开句柄数设置太小导致读写文件异常

问题背景与现象

文件最大打开句柄数设置太小，导致文件句柄不足。写文件到HDFS很慢，或者写文件失败。

原因分析

1. DataNode日志“/var/log/Bigdata/hdfs/dn/hadoop-omm-datanode-XXX.log”，存在异常提示java.io.IOException: Too many open files。

```
2016-05-19 17:18:59,126 | WARN  |
org.apache.hadoop.hdfs.server.datanode.DataXceiverServer@142ff9fa |
YSDN12:25009:DataXceiverServer:  |
org.apache.hadoop.hdfs.server.datanode.DataXceiverServer.run(DataXceiverServer.java:160)
java.io.IOException: Too many open files
at sun.nio.ch.ServerSocketChannelImpl.accept0(Native Method)
at sun.nio.ch.ServerSocketChannelImpl.accept(ServerSocketChannelImpl.java:241)
at sun.nio.ch.ServerSocketAdaptor.accept(ServerSocketAdaptor.java:100)
at org.apache.hadoop.hdfs.net.TcpPeerServer.accept(TcpPeerServer.java:134)
at org.apache.hadoop.hdfs.server.datanode.DataXceiverServer.run(DataXceiverServer.java:137)
at java.lang.Thread.run(Thread.java:745)
```

2. 如果某个DataNode日志中打印“Too many open files”，说明该节点文件句柄不足，导致打开文件句柄失败，然后就会重试往其他DataNode节点写数据，最终表现为写文件很慢或者写文件失败。

解决办法

步骤1 执行**ulimit -a**命令查看有问题节点文件句柄数最多设置是多少，如果很小，建议修改成640000。

图 12-9 查看文件句柄数

```
[omm@189-39-150-167 ~]$ ulimit -a
core file size          (blocks, -c) 0
data seg size           (kbytes, -d) unlimited
scheduling priority             (-e) 0
file size               (blocks, -f) unlimited
pending signals                 (-i) 256551
max locked memory       (kbytes, -l) 64
max memory size         (kbytes, -m) unlimited
open files                      (-n) 640000
pipe size            (512 bytes, -p) 8
POSIX message queues     (bytes, -q) 819200
real-time priority              (-r) 0
stack size              (kbytes, -s) 10240
cpu time               (seconds, -t) unlimited
max user processes              (-u) 60000
virtual memory          (kbytes, -v) unlimited
file locks                      (-x) unlimited
```

步骤2 执行**vi /etc/security/limits.d/90-nofile.conf**命令编辑文件，修改文件句柄数设置。如果没有这个文件，可以新建一个文件，并按照下图内容修改。

图 12-10 修改文件句柄数

```
*          hard    nofile    640000
*          soft    nofile    640000
~
```

步骤3 重新打开一个终端窗口，用**ulimit -a**命令查看是否修改成功，如果没有，请重新按照上述步骤重新修改。

步骤4 从Manager页面重启DataNode实例。

----结束

12.14 HDFS 客户端写文件 close 失败

问题背景与现象

HDFS客户端写文件close失败，客户端提示数据块没有足够副本数。

客户端日志：

```
2015-05-27 19:00:52.811 [pool-2-thread-3] ERROR: /tsp/nedata/collect/UGW/ugwufdr/
20150527/10/6_20150527105000_20150527105500_SR5S14_1432723806338_128_11.pkg.tmp143272380633
8 close hdfs sequence file fail (SequenceFileInfoChannel.java:444)
java.io.IOException: Unable to close file because the last block does not have enough number of replicas.
```

```
at org.apache.hadoop.hdfs.DFSOutputStream.completeFile(DFSOutputStream.java:2160)
at org.apache.hadoop.hdfs.DFSOutputStream.close(DFSOutputStream.java:2128)
at org.apache.hadoop.fs.FSDataOutputStream$PositionCache.close(FSDataOutputStream.java:70)
at org.apache.hadoop.fs.FSDataOutputStream.close(FSDataOutputStream.java:103)
at com.huawei.pai.collect2.stream.SequenceFileInfoChannel.close(SequenceFileInfoChannel.java:433)
at com.huawei.pai.collect2.stream.SequenceFileWriterToolChannel
$FileCloseTask.call(SequenceFileWriterToolChannel.java:804)
at com.huawei.pai.collect2.stream.SequenceFileWriterToolChannel
$FileCloseTask.call(SequenceFileWriterToolChannel.java:792)
at java.util.concurrent.FutureTask.run(FutureTask.java:262)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1145)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:615)
at java.lang.Thread.run(Thread.java:745)
```

原因分析

1. HDFS客户端开始写Block。

   例如：HDFS客户端是在2015-05-27 18:50:24,232开始写/20150527/10/6_20150527105000_20150527105500_SR5S14_1432723806338_128_11.pkg.tmp1432723806338的。其中分配的块是blk_1099105501_25370893。

   ```
   2015-05-27 18:50:24,232 | INFO  | IPC Server handler 30 on 25000 | BLOCK* allocateBlock: /
   20150527/10/6_20150527105000_20150527105500_SR5S14_1432723806338_128_11.pkg.tmp1432723
   806338. BP-1803470917-192.168.57.33-1428597734132
   blk_1099105501_25370893{blockUCState=UNDER_CONSTRUCTION, primaryNodeIndex=-1,
   replicas=[ReplicaUnderConstruction[[DISK]DS-b2d7b7d0-f410-4958-8eba-6deecbca2f87:NORMAL|
   RBW], ReplicaUnderConstruction[[DISK]DS-76bd80e7-ad58-49c6-bf2c-03f91caf750f:NORMAL|RBW]]}
   |
   org.apache.hadoop.hdfs.server.namenode.FSNamesystem.saveAllocatedBlock(FSNamesystem.java:3166
   )
   ```

2. 写完之后HDFS客户端调用了fsync。

   ```
   2015-05-27 19:00:22,717 | INFO  | IPC Server handler 22 on 25000 | BLOCK* fsync:
   20150527/10/6_20150527105000_20150527105500_SR5S14_1432723806338_128_11.pkg.tmp1432723
   806338 for DFSClient_NONMAPREDUCE_-120525246_15 |
   org.apache.hadoop.hdfs.server.namenode.FSNamesystem.fsync(FSNamesystem.java:3805)
   ```

3. HDFS客户端调用close关闭文件，NameNode收到客户端的close请求之后就会检查最后一个块的完成状态，只有当有足够的DataNode上报了块完成才可用关闭文件，检查块完成的状态是通过checkFileProgress函数检查的，打印如下：

   ```
   2015-05-27 19:00:27,603 | INFO  | IPC Server handler 44 on 25000 | BLOCK* checkFileProgress:
   blk_1099105501_25370893{blockUCState=COMMITTED, primaryNodeIndex=-1,
   replicas=[ReplicaUnderConstruction[[DISK]DS-ef5fd3c9-5088-4813-ae9a-34a0714ec3a3:NORMAL|
   RBW], ReplicaUnderConstruction[[DISK]DS-f863e30f-ce5b-48cc-9cca-72f64c558adc:NORMAL|RBW]]}
   has not reached minimal replication 1 |
   org.apache.hadoop.hdfs.server.namenode.FSNamesystem.checkFileProgress(FSNamesystem.java:3197)
   2015-05-27 19:00:28,005 | INFO  | IPC Server handler 45 on 25000 | BLOCK* checkFileProgress:
   blk_1099105501_25370893{blockUCState=COMMITTED, primaryNodeIndex=-1,
   replicas=[ReplicaUnderConstruction[[DISK]DS-ef5fd3c9-5088-4813-ae9a-34a0714ec3a3:NORMAL|
   RBW], ReplicaUnderConstruction[[DISK]DS-f863e30f-ce5b-48cc-9cca-72f64c558adc:NORMAL|RBW]]}
   has not reached minimal replication 1 |
   org.apache.hadoop.hdfs.server.namenode.FSNamesystem.checkFileProgress(FSNamesystem.java:3197)
   2015-05-27 19:00:28,806 | INFO  | IPC Server handler 63 on 25000 | BLOCK* checkFileProgress:
   blk_1099105501_25370893{blockUCState=COMMITTED, primaryNodeIndex=-1,
   replicas=[ReplicaUnderConstruction[[DISK]DS-ef5fd3c9-5088-4813-ae9a-34a0714ec3a3:NORMAL|
   RBW], ReplicaUnderConstruction[[DISK]DS-f863e30f-ce5b-48cc-9cca-72f64c558adc:NORMAL|RBW]]}
   has not reached minimal replication 1 |
   org.apache.hadoop.hdfs.server.namenode.FSNamesystem.checkFileProgress(FSNamesystem.java:3197)
   2015-05-27 19:00:30,408 | INFO  | IPC Server handler 43 on 25000 | BLOCK* checkFileProgress:
   blk_1099105501_25370893{blockUCState=COMMITTED, primaryNodeIndex=-1,
   replicas=[ReplicaUnderConstruction[[DISK]DS-ef5fd3c9-5088-4813-ae9a-34a0714ec3a3:NORMAL|
   RBW], ReplicaUnderConstruction[[DISK]DS-f863e30f-ce5b-48cc-9cca-72f64c558adc:NORMAL|RBW]]}
   has not reached minimal replication 1 |
   org.apache.hadoop.hdfs.server.namenode.FSNamesystem.checkFileProgress(FSNamesystem.java:3197)
   2015-05-27 19:00:33,610 | INFO  | IPC Server handler 37 on 25000 | BLOCK* checkFileProgress:
   blk_1099105501_25370893{blockUCState=COMMITTED, primaryNodeIndex=-1,
   replicas=[ReplicaUnderConstruction[[DISK]DS-ef5fd3c9-5088-4813-ae9a-34a0714ec3a3:NORMAL|
   ```

```
RBW], ReplicaUnderConstruction[[DISK]DS-f863e30f-ce5b-48cc-9cca-72f64c558adc:NORMAL|RBW]]}
has not reached minimal replication 1 |
org.apache.hadoop.hdfs.server.namenode.FSNamesystem.checkFileProgress(FSNamesystem.java:3197)
2015-05-27 19:00:40,011 | INFO  | IPC Server handler 37 on 25000 | BLOCK* checkFileProgress:
blk_1099105501_25370893{blockUCState=COMMITTED, primaryNodeIndex=-1,
replicas=[ReplicaUnderConstruction[[DISK]DS-ef5fd3c9-5088-4813-ae9a-34a0714ec3a3:NORMAL|
RBW], ReplicaUnderConstruction[[DISK]DS-f863e30f-ce5b-48cc-9cca-72f64c558adc:NORMAL|RBW]]}
has not reached minimal replication 1 |
org.apache.hadoop.hdfs.server.namenode.FSNamesystem.checkFileProgress(FSNamesystem.java:3197)
```

4. NameNode打印了多次checkFileProgress是由于HDFS客户端多次尝试close文件，但是由于当前状态不满足要求，导致close失败， HDFS客户端retry的次数是由参数dfs.client.block.write.locateFollowingBlock.retries决定的，该参数默认是5，所以在NameNode的日志中看到了6次checkFileProgress打印。
5. 但是再过0.5s之后，DataNode就上报块已经成功写入。

```
2015-05-27 19:00:40,608 | INFO  | IPC Server handler 60 on 25000 | BLOCK* addStoredBlock:
blockMap updated: 192.168.10.21:25009 is added to
blk_1099105501_25370893{blockUCState=COMMITTED, primaryNodeIndex=-1,
replicas=[ReplicaUnderConstruction[[DISK]DS-ef5fd3c9-5088-4813-ae9a-34a0714ec3a3:NORMAL|
RBW], ReplicaUnderConstruction[[DISK]DS-f863e30f-ce5b-48cc-9cca-72f64c558adc:NORMAL|RBW]]}
size 11837530 |
org.apache.hadoop.hdfs.server.blockmanagement.BlockManager.logAddStoredBlock(BlockManager.java
:2393)
2015-05-27 19:00:48,297 | INFO  | IPC Server handler 37 on 25000 | BLOCK* addStoredBlock:
blockMap updated: 192.168.10.10:25009 is added to blk_1099105501_25370893 size 11837530 |
org.apache.hadoop.hdfs.server.blockmanagement.BlockManager.logAddStoredBlock(BlockManager.java
:2393)
```

6. DataNode上报块写成功通知延迟的原因可能有：网络瓶颈导致、CPU瓶颈导致。
7. 如果此时再次调用close或者close的retry的次数增多，那么close都将返回成功。建议适当增大参数dfs.client.block.write.locateFollowingBlock.retries的值，默认值为5次，尝试的时间间隔为400ms、800ms、1600ms、3200ms、6400ms，12800ms，那么close函数最多需要25.2秒才能返回。

解决办法

可以通过调整客户端参数dfs.client.block.write.locateFollowingBlock.retries的值来增加retry的次数，可以将值设置为6，那么中间睡眠等待的时间为400ms、800ms、1600ms、3200ms、6400ms、12800ms，也就是说close函数最多要50.8秒才能返回。

备注说明

一般出现上述现象，说明集群负载很大，通过调整参数只是临时规避这个问题，建议还是降低集群负载。

例如：避免把所有CPU都分配MapReduce跑任务。

12.15 文件错误导致上传文件到 HDFS 失败

问题背景与现象

用**hadoop dfs -put**把本地文件拷贝到HDFS上，有报错。

上传部分文件后，报错失败，从NameNode原生页面看，临时文件大小不再变化。

原因分析

1. 查看NameNode日志“/var/log/Bigdata/hdfs/nn/hadoop-omm-namenode-主机名.log”，发现该文件一直在被尝试写，直到最终失败。
```
2015-07-13 10:05:07,847 | WARN  | org.apache.hadoop.hdfs.server.namenode.LeaseManager
$Monitor@36fea922 | DIR* NameSystem.internalReleaseLease: Failed to release lease for file /hive/
order/OS_ORDER._8.txt._COPYING_. Committed blocks are waiting to be minimally replicated. Try
again later. | FSNamesystem.java:3936
2015-07-13 10:05:07,847 | ERROR | org.apache.hadoop.hdfs.server.namenode.LeaseManager
$Monitor@36fea922 | Cannot release the path /hive/order/OS_ORDER._8.txt._COPYING_ in the lease
[Lease.  Holder: DFSClient_NONMAPREDUCE_-1872896146_1, pendingcreates: 1] |
LeaseManager.java:459
org.apache.hadoop.hdfs.protocol.AlreadyBeingCreatedException: DIR*
NameSystem.internalReleaseLease: Failed to release lease for file /hive/order/
OS_ORDER._8.txt._COPYING_. Committed blocks are waiting to be minimally replicated. Try again
later.
at FSNamesystem.internalReleaseLease(FSNamesystem.java:3937)
```
2. 根因分析：被上传的文件损坏，因此会上传失败。
3. 验证办法：cp或者scp被拷贝的文件，也会失败，确认文件本身已损坏。

解决办法

步骤1 文件本身损坏造成的此问题，采用正常文件进行上传。

----结束

12.16 界面配置 dfs.blocksize 后上传数据，block 大小未改变

问题背景与现象

界面配置“dfs.blocksize”，将其设置为268435456，上传数据，block大小还是原来的大小。

原因分析

HDFS客户端的“hdfs-site.xml”文件中的dfs.blocksize大小没有更改，以客户端配置为准。

解决办法

步骤1 确保“dfs.blocksize”为512的倍数。

步骤2 重新下载安装客户端或者更改客户端配置。

步骤3 dfs.blocksize是客户端配置，以客户端为准。若客户端不配置，以服务端为准。

----结束

12.17 读取 HDFS 文件失败，报错“FileNotFoundException”

问题背景与现象

有MapReduce任务所有map任务均成功，但reduce任务失败，查看日志发现报异常“FileNotFoundException...No lease on...File does not exist”。

```
Error: org.apache.hadoop.ipc.RemoteException(java.io.FileNotFoundException): No lease on /user/sparkhive/warehouse/daas/dsp/output/_temporary/1/_temporary/attempt_1479799053892_17075_r_000007_0/part-r-00007 (inode 6501287): File does not exist. Holder DFSClient_attempt_1479799053892_17075_r_000007_0_-1463597952_1 does not have any open files.
at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.checkLease(FSNamesystem.java:3350)
at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.completeFileInternal(FSNamesystem.java:3442)
at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.completeFile(FSNamesystem.java:3409)
at org.apache.hadoop.hdfs.server.namenode.NameNodeRpcServer.complete(NameNodeRpcServer.java:789)
```

原因分析

FileNotFoundException...No lease on...File does not exist，该日志说明文件在操作的过程中被删除了。

1. 搜索HDFS的NameNode的审计日志（Active NameNode的/var/log/Bigdata/audit/hdfs/nn/hdfs-audit-namenode.log）搜索文件名，确认文件的创建时间。
2. 搜索文件创建到出现异常时间范围的NameNode的审计日志，搜索该文件是否被删除或者移动到其他目录。
3. 如果该文件没有被删除或者移动，可能是该文件的父目录，或者更上层目录被删除或者移动，需要继续搜索上层目录。如本样例中，是文件的父目录被删除。

```
2017-05-31 02:04:08,286 | INFO  | IPC Server handler 30 on 25000 | allowed=true ugi=appUser@HADOOP.COM (auth:TOKEN) ip=/192.168.1.22	cmd=delete	src=/user/sparkhive/warehouse/daas/dsp/output/_temporary	dst=null	perm=null proto=rpc | FSNamesystem.java:8189
```

说明

- 如上日志说明：192.168.1.22 节点的appUser用户删除了/user/sparkhive/warehouse/daas/dsp/output/_temporary。
- 可以使用**zgrep "文件名" *.zip**命令搜索zip包的内容。

解决办法

步骤1 需要排查业务，确认为何该文件或者文件的父目录被删除。

----结束

12.18 HDFS 写文件失败，报错“item limit of xxx is exceeded”

问题背景与现象

客户端或者上层组件日志报往HDFS的某目录写文件失败，报错为：

The directory item limit of /tmp is exceeded: limit=5 items=5。

原因分析

1. 查看客户端或者NameNode运行日志“/var/log/Bigdata/hdfs/nn/hadoop-omm-namenode-XXX.log”存在异常提示The directory item limit of /tmp is exceeded:。该错误的含义为/tmp目录的文件数超过1048576的限制。
```
2018-03-14 11:18:21,625 | WARN  | IPC Server handler 62 on 25000 | DIR* NameSystem.startFile: /tmp/test.txt The directory item limit of /tmp is exceeded: limit=1048576 items=1048577 | FSNamesystem.java:2334
```
2. 该限制是dfs.namenode.fs-limits.max-directory-items参数，定义单个目录下不含递归的最大目录数或者文件数，默认值1048576，取值范围1～6400000。

解决办法

步骤1 确认该目录不含递归拥有100万以上文件目录是否正常，如果正常，可以将HDFS的参数dfs.namenode.fs-limits.max-directory-items调大并且重启HDFS NameNode生效。

步骤2 如果该目录下拥有100万文件不正常，需要清理不需要的文件。

----结束

12.19 调整 HDFS SHDFShell 客户端日志级别

- **临时调整**，关闭该shell客户端窗口后，日志会还原为默认值。
  a. 执行**export HADOOP_ROOT_LOGGER**命令可以调整客户端日志级别。
  b. 执行**export HADOOP_ROOT_LOGGER=***日志级别***,console**，可以调整shell客户端的日志级别。

     export HADOOP_ROOT_LOGGER=DEBUG,console，调整为DEBUG。

     export HADOOP_ROOT_LOGGER=ERROR,console，调整为ERROR。
- **永久调整**
  a. 在HDFS客户端环境变量配置文件“/opt/client/HDFS/component_env”（其中“/opt/client”需要改为实际客户端路径）增加“export HADOOP_ROOT_LOGGER=*日志级别*,console”。
  b. 执行**source /opt/client/bigdata_env**。
  c. 重新执行客户端命令。

12.20 HDFS 读文件失败报错“No common protection layer”

问题背景与现象

shell客户端或者其他客户端操作HDFS失败，报“**No common protection layer between client and server**”。

在集群外的机器，执行任意hadoop命令，如**hadoop fs -ls /**均失败，最底层的报错为"**No common protection layer between client and server**"。

```
2017-05-13 19:14:19,060 | ERROR | [pool-1-thread-1] |  Server startup failure  | org.apache.sqoop.core.SqoopServer.initializeServer(SqoopServer.java:69)
org.apache.sqoop.common.SqoopException: MAPRED_EXEC_0028:Failed to operate HDFS - Failed to get the file /user/loader/etl_dirty_data_dir status
```

```
        at org.apache.sqoop.job.mr.HDFSClient.fileExist(HDFSClient.java:85)
...
        at java.lang.Thread.run(Thread.java:745)
Caused by: java.io.IOException: Failed on local exception: java.io.IOException: Couldn't setup connection for loader/hadoop@HADOOP.COM to loader37/10.162.0.37:25000; Host Details : local host is: "loader37/10.162.0.37"; destination host is: "loader37":25000;
        at org.apache.hadoop.net.NetUtils.wrapException(NetUtils.java:776)
...
        ... 10 more
Caused by: java.io.IOException: Couldn't setup connection for loader/hadoop@HADOOP.COM to loader37/10.162.0.37:25000
        at org.apache.hadoop.ipc.Client$Connection$1.run(Client.java:674
        ... 28 more
Caused by: javax.security.sasl.SaslException: No common protection layer between client and server
        at com.sun.security.sasl.gsskerb.GssKrb5Client.doFinalHandshake(GssKrb5Client.java:251)
...
        at org.apache.hadoop.ipc.Client$Connection.setupIOstreams(Client.java:720)
```

原因分析

1. HDFS的客户端和服务端数据传输走的rpc协议，该协议有多种加密方式，由hadoop.rpc.protection参数控制。
2. 如果客户端和服务端的hadoop.rpc.protection参数的配置值不一样，即会报**No common protection layer between client and server**错误。

说明

hadoop.rpc.protection参数表示数据可通过以下任一方式在节点间进行传输。

- privacy：指数据在鉴权及加密后再传输。这种方式会降低性能。
- authentication：指数据在鉴权后直接传输，不加密。这种方式能保证性能但存在安全风险。
- integrity：指数据直接传输，即不加密也不鉴权。 为保证数据安全，请谨慎使用这种方式。

解决办法

步骤1 重新下载客户端，如果是应用程序，更新应用程序中的配置文件。

----结束

12.21 HDFS 目录配额不足导致写文件失败

问题背景与现象

给某目录设置quota后，往目录中写文件失败，出现如下问题“**The DiskSpace quota of** /tmp/tquota2 is **exceeded**”。

```
[omm@189-39-150-115 client]$ hdfs dfs -put switchuser.py  /tmp/tquota2
put: The DiskSpace quota of /tmp/tquota2 is exceeded: quota = 157286400 B = 150 MB but diskspace consumed = 402653184 B = 384 MB
```

可能原因

目录配置的剩余的空间小于写文件实际需要的空间。

原因分析

1. HDFS支持设置某目录的配额，即限制某目录下的文件最多占用空间大小，例如如下命令是设置“/tmp/tquota”目录最多写入150MB的文件（文件大小*副本数）。

   hadoop dfsadmin -setSpaceQuota 150M /tmp/tquota2

2. 使用如下命令可以查看目录设置的配额情况，SPACE_QUOTA是设置的空间配额，REM_SPACE_QUOTA是当前剩余的空间配额。

   hdfs dfs -count -q -h -v /tmp/tquota2

   图 12-11 查看目录设置的配额

```
hdfs dfs -count -q -h -v /tmp/tquota2
QUOTA REM_QUOTA SPACE_QUOTA REM_SPACE_QUOTA  DIR_COUNT   FILE_COUNT CONTENT_SIZE PATHNAME
none  inf       150M        150M                 1            0            0  /tmp/tquota2
```

3. 日志分析，如下日志说明写入文件需要消耗384M，但是当前的空间配额是150M，因此空间不足。写文件前，需要的剩余空间是：块大小*副本数，128M*3副本=384M。

```
[omm@189-39-150-115 client]$
[omm@189-39-150-115 client]$ hdfs dfs -put switchuser.py  /tmp/tquota2
put: The DiskSpace quota of /tmp/tquota2 is exceeded: quota = 157286400 B = 150 MB but diskspace consumed = 402653184 B = 384 MB
```

解决办法

步骤1 增加配额大小，即重新设置目录的配额大小。

hadoop dfsadmin -setSpaceQuota 150G /*目录名*

步骤2 清空配额。

hdfs dfsadmin -clrSpaceQuota /*目录名*

----结束

12.22 执行 balance 失败报错“Source and target differ in block-size”

问题背景与现象

执行distcp跨集群拷贝文件时，出现部分文件拷贝失败“ Source and target differ in block-size. Use -pb to preserve block-sizes during copy. ”

```
Caused by: java.io.IOException: Check-sum mismatch between hdfs://10.180.144.7:25000/kylin/kylin_default_instance_prod/parquet/f2e72874-f01c-45ff-b219-207f3a5b3fcb/c769cd2d-575a-4459-837b-a19dd7b20c27/339114721280/0.parquettar and hdfs://10.180.180.194:25000/kylin/kylin_default_instance_prod/parquet/f2e72874-f01c-45ff-b219-207f3a5b3fcb/.distcp.tmp.attempt_1523424430246_0004_m_000019_2. Source and target differ in block-size. Use -pb to preserve block-sizes during copy. Alternatively, skip checksum-checks altogether, using -skipCrc. (NOTE: By skipping checksums, one runs the risk of masking data-corruption during file-transfer.)        at org.apache.hadoop.tools.mapred.RetriableFileCopyCommand.compareCheckSums(RetriableFileCopyCommand.java:214)
```

可能原因

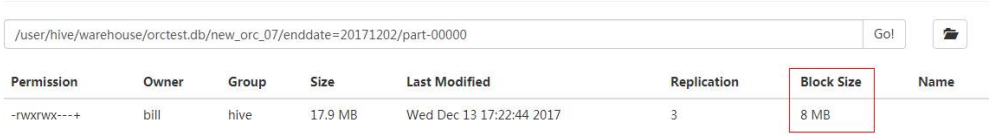
distcp默认拷贝文件时不记录原block大小导致在原文件block.size不是128M时校验失败，需要在distcp命令增加-pb参数。

原因分析

1. HDFS在写的时候有设置块大小，默认128M，某些组件或者业务程序写入的文件可能不是128M，如8M。

```
<name>dfs.blocksize</name>
<value>134217728</value>
```

图 12-12 某些组件或者业务程序写入的文件大小



2. distcp 从源集群读文件后写入新集群，默认是使用的MapReduce任务中的dfs.blocksize，默认128M。
3. 在distcp写完文件后，会基于块的物理大小做校验，因为该文件在新旧集群中block.size不一致，因此拆分大小不一致，导致校验失败。

   如以上文件，在旧集群是17.9/8MB = 3个block，在新集群17.9/128M = 1个block，因此实际在磁盘的物理大小因分割而导致校验失败。

解决办法

distcp时，增加**-pb**参数。该参数作用为distcp时候保留block大小，确保新集群写入文件blocksize和老集群一致。

图 12-13 distcp 时保留 block 大小

```
[root@189-39-235-118 clientu10]#hadoop distcp -pb  hdfs://haclusterX/user hdfs://hacluster/tmp/test
```

12.23 查询或者删除 HDFS 文件失败

问题背景与现象

使用HDFS的shell客户端查询或者删除文件失败，父目录可以看见此文件（不可见字符）。

图 12-14 父目录文件列表

```
drwxrwx---+  - datalab90020_639_w hive          0 2018-04-10 01:44 /user/hive/warehouse/datalake_dwi_barpsit.db/v_tp_mp_aut_input_tmp
drwxrwx---+  - datalab90020_639_w hive          0 2018-04-10 16:45 /user/hive/warehouse/datalake_dwi_barpsit.db/v_tp_mp_aut_input_tmp2
[root@dggtsp355-or FusionInsight_client]# hadoop fs -ls /user/hive/warehouse/datalake_dwi_barpsit.db
Found 4 items
drwxrwxr-x   - datalab90020_639_w hive          0 2018-04-11 12:05 /user/hive/warehouse/datalake_dwi_barpsit.db/bak_v_tp_mp_aut_input
drwxrwx---+  - datalab90020_639_w hive          0 2018-04-11 11:16 /user/hive/warehouse/datalake_dwi_barpsit.db/v_tp_mp_aut_input
drwxrwx---+  - datalab90020_639_w hive          0 2018-04-10 01:44 /user/hive/warehouse/datalake_dwi_barpsit.db/v_tp_mp_aut_input_tmp
drwxrwx---+  - datalab90020_639_w hive          0 2018-04-10 16:45 /user/hive/warehouse/datalake_dwi_barpsit.db/v_tp_mp_aut_input_tmp2
[root@dggtsp355-or FusionInsight_client]# hadoop fs -rm -r /user/hive/warehouse/datalake_dwi_barpsit.db/v_tp_mp_aut_input
rm: `/user/hive/warehouse/datalake_dwi_barpsit.db/v_tp_mp_aut_input': No such file or directory
[root@dggtsp355-or FusionInsight_client]# hdfs dfs -rm -r /user/hive/warehouse/datalake_dwi_barpsit.db/v_tp_mp_aut_input
rm: `/user/hive/warehouse/datalake_dwi_barpsit.db/v_tp_mp_aut_input': No such file or directory
[root@dggtsp355-or FusionInsight_client]#
[root@dggtsp355-or FusionInsight_client]#
[root@dggtsp355-or FusionInsight_client]# hdfs dfs -ls /user/hive/warehouse/datalake_dwi_barpsit.db/v_tp_mp_aut_input
ls: `/user/hive/warehouse/datalake_dwi_barpsit.db/v_tp_mp_aut_input': No such file or directory
[root@dggtsp355-or FusionInsight_client]#
[root@dggtsp355-or FusionInsight_client]#
```

原因分析

可能是该文件写入时有异常，写入了不可见字符。可以将该文件名重定向写入本地文本中，使用**vi**命令打开。

hdfs dfs -ls *父目录* **> /tmp/t.txt**

vi /tmp/t.txt

然后输入命令“**:set list**”将文件名的不可见字符显示出来。如这里显示出文件名中包含“**^M**”不可见字符。

图 12-15 显示不可见字符

```
drwxrwx---+  - datalab90020_639_w hive          0 2018-04-11 11:16 /user/hive/warehouse/datalake_dwi_barpsit.db/v_tp_mp_aut_input^M$
~
~
~
```

解决办法

步骤1 使用shell命令读到文本中记录的该文件名，确认如下命令输出的是该文件在HDFS中的全路径。

cat /tmp/t.txt |awk '{print $8}'

图 12-16 文件路径

```
drwxrwx---+  - datalab90020_639_w hive          0 2018-04-11 11:16 /user/hive/warehouse/datalake_dwi_barpsit.db/v_tp_mp_aut_input
drwxrwx---+  - datalab90020_639_w hive          0 2018-04-10 01:44 /user/hive/warehouse/datalake_dwi_barpsit.db/v_tp_mp_aut_input_tmp
drwxrwx---+  - datalab90020_639_w hive          0 2018-04-10 16:45 /user/hive/warehouse/datalake_dwi_barpsit.db/v_tp_mp_aut_input_tmp2
[root@dggtsp355-or FusionInsight_client]# cat /tmp/t.txt |awk '{print $8}'

/user/hive/warehouse/datalake_dwi_barpsit.db/v_tp_mp_aut_input
[root@dggtsp355-or FusionInsight_client]# hadoop fs -rm -r $(cat /tmp/t.txt |awk '{print $8}')
' to trash at: hdfs://hacluster/user/datalab90020_639_w/.Trash/Current/user/hive/warehouse/datalake_dwi_barpsit.db/v_tp_mp_aut_input
' to trash at: hdfs://hacluster/user/datalab90020_639_w/.Trash/Currentrehouse/datalake_dwi_barpsit.db/v_tp_mp_aut_input
[root@dggtsp355-or FusionInsight_client]# hdfs dfs -ls  /user/hive/warehouse/datalake_dwi_barpsit.db
Found 2 items
drwxrwx---+  - datalab90020_639_w hive          0 2018-04-10 01:44 /user/hive/warehouse/datalake_dwi_barpsit.db/v_tp_mp_aut_input_tmp
drwxrwx---+  - datalab90020_639_w hive          0 2018-04-10 16:45 /user/hive/warehouse/datalake_dwi_barpsit.db/v_tp_mp_aut_input_tmp2
[root@dggtsp355-or FusionInsight_client]#
```

步骤2 使用如下命令删除该文件。

hdfs dfs -rm $(cat /tmp/t.txt |awk '{print $8}')

步骤3 查看确认该文件已被删除。

hdfs dfs -ls *父目录*

----结束

12.24 非 HDFS 数据残留导致数据分布不均衡

问题背景与现象

数据出现不均衡，某磁盘过满而其他磁盘未写满。

HDFS DataNode数据存储目录配置为“/export/data1/dfs--/export/data12/dfs”，看到的现象是大量数据都是存储到了“/export/data1/dfs”，其他盘的数据比较均衡。

原因分析

磁盘为卸载重装，有一个目录在上次卸载时未卸载干净，即添加的磁盘，未格式化，残留历史垃圾数据。

解决办法

手动清理未卸载干净的数据。

12.25 HDFS 客户端安装在数据节点导致数据分布不均衡

问题背景与现象

HDFS的DataNode数据分布不均匀，在某节点上磁盘使用率很高，甚至达到100%，其他节点空闲很多。

原因分析

客户端安装在该节点，根据HDFS数据副本机制，第一个副本会存放在本地机器，最终导致节点磁盘被占满，而其他节点空闲很多。

解决办法

步骤1 针对已有不平衡的数据，执行balance脚本均衡数据。

/opt/client/HDFS/hadoop/sbin/start-balancer.sh -threshold 10

其中“/opt/client”是实际的客户端安装目录。

步骤2 针对新写入数据，将客户端安装在没有安装DataNode的节点。

----结束

12.26 节点内 DataNode 磁盘使用率不均衡

问题背景与现象

单个节点内DataNode的各磁盘使用率不均匀。

例如：

```
189-39-235-71:~ # df -h
Filesystem  Size  Used Avail Use% Mounted on
/dev/xvda  360G  92G   250G  28% /
/dev/xvdb  700G  900G   200G  78% /srv/BigData/hadoop/data1
/dev/xvdc  700G  900G   200G  78% /srv/BigData/hadoop/data2
/dev/xvdd  700G  900G   200G  78% /srv/BigData/hadoop/data3
/dev/xvde  700G  900G   200G  78% /srv/BigData/hadoop/data4
/dev/xvdf  10G   900G   890G  2% /srv/BigData/hadoop/data5
189-39-235-71:~ #
```

可能原因

部分磁盘故障，更换为新盘，因此新盘使用率低。

增加了磁盘个数，如原先4个数据盘，现扩容为5个数据盘。

原因分析

DataNode节点内写block磁盘时，有两种策略“轮询”和“优先写剩余磁盘空间多的磁盘”，默认是“轮询”。

参数说明：dfs.datanode.fsdataset.volume.choosing.policy

可选值：

- 轮询：org.apache.hadoop.hdfs.server.datanode.fsdataset.RoundRobinVolumeChoosingPolicy
- 优先写剩余空间多的磁盘：org.apache.hadoop.hdfs.server.datanode.fsdataset.AvailableSpaceVolumeChoosingPolicy

解决办法

将DataNode选择磁盘策略的参数dfs.datanode.fsdataset.volume.choosing.policy的值改为：org.apache.hadoop.hdfs.server.datanode.fsdataset.AvailableSpaceVolumeChoosingPolicy，保存并重启受影响的服务或实例。

让DataNode根据磁盘剩余空间大小，优先选择磁盘剩余空间多的节点存储数据副本。

说明

- 针对新写入到本DataNode的数据会优先写磁盘剩余空间多的磁盘。
- 部分磁盘使用率较高，依赖业务逐渐删除在HDFS中的数据（老化数据）来逐渐降低。

12.27 执行 balance 常见问题定位方法

问题 1：报没权限（Access denied）执行 balance

问题详细：执行start-balancer.sh，“hadoop-root-balancer-主机名.out”日志显示“Access denied for user test1. Superuser privilege is required”

```
cat /opt/client/HDFS/hadoop/logs/hadoop-root-balancer-host2.out
Time Stamp              Iteration#  Bytes Already Moved  Bytes Left To Move  Bytes Being Moved
INFO: Watching file:/opt/client/HDFS/hadoop/etc/hadoop/log4j.properties for changes with interval : 60000
org.apache.hadoop.ipc.RemoteException(org.apache.hadoop.security.AccessControlException): Access denied
for user test1.
Superuser privilege is required
at
org.apache.hadoop.hdfs.server.namenode.FSPermissionChecker.checkSuperuserPrivilege(FSPermissionChecker
.java:122)
at
org.apache.hadoop.hdfs.server.namenode.FSNamesystem.checkSuperuserPrivilege(FSNamesystem.java:5916)
```

问题根因：

执行balance需要使用管理员账户

解决方法

- 安全版本
  使用hdfs或者其他属于supergroup组的用户认证后，执行balance
- 普通版本
  执行HDFS的balance命令前，需要在客户端执行su - hdfs命令。

问题 2：执行 balance 失败，/system/balancer.id 文件异常

问题详细：

在HDFS客户端启动一个Balance进程，该进程被异常停止后，再次执行Balance操作，操作会失败。

```
org.apache.hadoop.ipc.RemoteException(org.apache.hadoop.protocol.RecoveryInProgressException): Failed to APPEND_FILE /system/balancer.id for DFSClient because lease recovery is in progress. Try again later.
```

问题根因：

通常，HDFS执行Balance操作结束后，会自动释放“/system/balancer.id”文件，可再次正常执行Balance。

但在上述场景中，由于第一次的Balance操作是被异常停止的，所以第二次进行Balance操作时，“/system/balancer.id”文件仍然存在，则会触发append /system/balancer.id操作，进而导致Balance操作失败。

解决方法

方法1：等待硬租期超过1小时后，原有客户端释放租约，再执行第二次Balance操作。

方法2：删除HDFS中的“/system/balancer.id”文件，再执行下次Balance操作。

12.28 HDFS 显示磁盘空间不足，其实还有 10%磁盘空间

问题背景与现象

1. 出现“HDFS磁盘空间使用率超过阈值”告警。
2. 查看HDFS页面，查看磁盘空间使用率非常高。

原因分析

HDFS中配置了dfs.datanode.du.reserved.percentage参数：每个磁盘的保留空间所占磁盘百分比。DataNode会保留这么多可用空间，以备其他组件如Yarn的NodeManager运行计算时，或者预留升级时使用。

因为预留了10%的磁盘，当磁盘使用率达到90%的时候，HDFS的DataNode即会认为没有可用磁盘空间。

解决办法

步骤1 扩容，在HDFS DataNode磁盘到80%，即需要及时扩容，请参考**扩容集群**。扩容完成后需进行数据均衡，请参考**配置HDFS DataNode数据均衡**。

步骤2 如不能及时扩容，需要删除HDFS中的不需要数据，释放磁盘空间。

----结束

12.29 使用普通集群 Core 节点上安装的 HDFS 客户端时报错“error creating DomainSocket”

用户问题

普通集群在Core节点新建用户安装使用HDFS客户端报错。

问题现象

普通集群在Core节点新建用户安装使用客户端报错如下：

```
2020-03-14 19:16:17,166 WARN shortcircuit.DomainSocketFactory: error creating DomainSocket
java.net.ConnectException: connect(2) error: Permission denied when trying to connect to '/var/run/MRS-HDFS/dn_socket'
at org.apache.hadoop.net.unix.DomainSocket.connect0(Native Method)
at org.apache.hadoop.net.unix.DomainSocket.connect(DomainSocket.java:256)
at org.apache.hadoop.hdfs.shortcircuit.DomainSocketFactory.createSocket(DomainSocketFactory.java:168)
at org.apache.hadoop.hdfs.client.impl.BlockReaderFactory.nextDomainPeer(BlockReaderFactory.java:799)
...
```

原因分析

用户使用 **useradd** 命令来创建用户，此用户默认用户组不包含“ficommmon”用户组，导致在使用hdfs的get命令的时候出现上述报错。

处理步骤

使用命令**usermod -a -G ficommon username** 为用户添加用户组“ficommon”。

12.30 集群外节点安装客户端上传 HDFS 文件失败

用户问题

集群外节点安装客户端使用**hdfs**命令上传文件失败。

问题现象

在集群节点上安装客户端，在该客户端使用**hdfs**命令上传一个文件，报如下错误：

图 12-17 上传文件报错

```
[root@gywa02 bin]# hadoop fs -put test.txt /tmp/input
2020-07-31 18:12:27,533 INFO obs.OBSFileSystem: This Filesystem GC-ful, clear resource.
2020-07-31 18:12:31,757 INFO hdfs.DataStreamer: Exception in createBlockOutputStream blk_1073774851_34031
java.net.NoRouteToHostException: No route to host
        at sun.nio.ch.SocketChannelImpl.checkConnect(Native Method)
        at sun.nio.ch.SocketChannelImpl.finishConnect(SocketChannelImpl.java:717)
        at org.apache.hadoop.net.SocketIOWithTimeout.connect(SocketIOWithTimeout.java:206)
        at org.apache.hadoop.net.NetUtils.connect(NetUtils.java:531)
        at org.apache.hadoop.hdfs.DataStreamer.createSocketForPipeline(DataStreamer.java:255)
        at org.apache.hadoop.hdfs.DataStreamer.createBlockOutputStream(DataStreamer.java:1789)
        at org.apache.hadoop.hdfs.DataStreamer.nextBlockOutputStream(DataStreamer.java:1743)
        at org.apache.hadoop.hdfs.DataStreamer.run(DataStreamer.java:718)
2020-07-31 18:12:31,759 WARN hdfs.DataStreamer: Abandoning BP-1721849101-192.168.0.86-1595473704426:blk_1073774851_34031
2020-07-31 18:12:31,800 WARN hdfs.DataStreamer: Excluding datanode DatanodeInfoWithStorage[192.168.0.157:9866,DS-592b7049-b4af-4bba-a184-1e1928a9028b,DISK]
2020-07-31 18:12:34,860 INFO hdfs.DataStreamer: Exception in createBlockOutputStream blk_1073774852_34032
java.net.NoRouteToHostException: No route to host
        at sun.nio.ch.SocketChannelImpl.checkConnect(Native Method)
        at sun.nio.ch.SocketChannelImpl.finishConnect(SocketChannelImpl.java:717)
        at org.apache.hadoop.net.SocketIOWithTimeout.connect(SocketIOWithTimeout.java:206)
        at org.apache.hadoop.net.NetUtils.connect(NetUtils.java:531)
        at org.apache.hadoop.hdfs.DataStreamer.createSocketForPipeline(DataStreamer.java:255)
        at org.apache.hadoop.hdfs.DataStreamer.createBlockOutputStream(DataStreamer.java:1789)
        at org.apache.hadoop.hdfs.DataStreamer.nextBlockOutputStream(DataStreamer.java:1743)
        at org.apache.hadoop.hdfs.DataStreamer.run(DataStreamer.java:718)
2020-07-31 18:12:34,860 WARN hdfs.DataStreamer: Abandoning BP-1721849101-192.168.0.86-1595473704426:blk_1073774852_34032
2020-07-31 18:12:34,899 WARN hdfs.DataStreamer: Excluding datanode DatanodeInfoWithStorage[192.168.0.189:9866,DS-5bee1b3a-4453-4d86-a632-262cb67c0bdb,DISK]
2020-07-31 18:12:37,948 INFO hdfs.DataStreamer: Exception in createBlockOutputStream blk_1073774853_34033
java.net.NoRouteToHostException: No route to host
        at sun.nio.ch.SocketChannelImpl.checkConnect(Native Method)
        at sun.nio.ch.SocketChannelImpl.finishConnect(SocketChannelImpl.java:717)
        at org.apache.hadoop.net.SocketIOWithTimeout.connect(SocketIOWithTimeout.java:206)
        at org.apache.hadoop.net.NetUtils.connect(NetUtils.java:531)
        at org.apache.hadoop.hdfs.DataStreamer.createSocketForPipeline(DataStreamer.java:255)
        at org.apache.hadoop.hdfs.DataStreamer.createBlockOutputStream(DataStreamer.java:1789)
        at org.apache.hadoop.hdfs.DataStreamer.nextBlockOutputStream(DataStreamer.java:1743)
        at org.apache.hadoop.hdfs.DataStreamer.run(DataStreamer.java:718)
2020-07-31 18:12:37,948 WARN hdfs.DataStreamer: Abandoning BP-1721849101-192.168.0.86-1595473704426:blk_1073774853_34033
2020-07-31 18:12:37,988 WARN hdfs.DataStreamer: Excluding datanode DatanodeInfoWithStorage[192.168.0.174:9866,DS-fa34f00b-2c03-4d0e-ad6e-3a2555735cbd,DISK]
2020-07-31 18:12:38,034 WARN hdfs.DataStreamer: DataStreamer Exception
org.apache.hadoop.ipc.RemoteException(java.io.IOException): File /tmp/input/test.txt._COPYING_ could only be written to 0 of the 1 minReplication nodes. There are 3 da
        at org.apache.hadoop.hdfs.server.blockmanagement.BlockManager.chooseTarget4NewBlock(BlockManager.java:2223)
        at org.apache.hadoop.hdfs.server.namenode.FSDirWriteFileOp.chooseTargetForNewBlock(FSDirWriteFileOp.java:346)
        at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.getAdditionalBlock(FSNamesystem.java:2727)
        at org.apache.hadoop.hdfs.server.namenode.NameNodeRpcServer.addBlock(NameNodeRpcServer.java:879)
        at org.apache.hadoop.hdfs.protocolPB.ClientNamenodeProtocolServerSideTranslatorPB.addBlock(ClientNamenodeProtocolServerSideTranslatorPB.java:596)
        at org.apache.hadoop.hdfs.protocol.proto.ClientNamenodeProtocolProtos$ClientNamenodeProtocol$2.callBlockingMethod(ClientNamenodeProtocolProtos.java)
        at org.apache.hadoop.ipc.ProtobufRpcEngine$Server$ProtoBufRpcInvoker.call(ProtobufRpcEngine.java:530)
        at org.apache.hadoop.ipc.RPC$Server.call(RPC.java:1036)
```

原因分析

从错误截图可以看到报错是no route to host，且报错信息里面有192.168的ip，也即客户端节点到集群的DN节点的内网路由不通，导致上传文件失败。

处理步骤

在客户端节点的客户端目录下，找到HDFS的客户端配置目录hdfs-site.xml文件，在配置文件中增加配置项dfs.client.use.datanode.hostname，并将该配置设置为true。

12.31 HDFS 写并发较大时报副本不足

问题背景与现象

用户运行作业时写文件到HDFS，偶现写文件失败的情况。

操作日志如下：

```
105 | INFO  | IPC Server handler 23 on 25000 | IPC Server handler 23 on 25000, call
org.apache.hadoop.hdfs.protocol.ClientProtocol.addBlock from 192.168.1.96:47728 Call#1461167 Retry#0 |
Server.java:2278
java.io.IOException: File /hive/warehouse/000000_0.835bf64f-4103 could only be replicated to 0 nodes
instead of minReplication (=1).  There are 3 datanode(s) running and 3 node(s) are excluded in this
operation.
```

原因分析

- HDFS写文件的预约机制：无论文件是10 MB还是1 GB，开始写的每个块都会被预约128 MB。如果需要写入一个10 MB的文件，HDFS会预约一个块来写，当文件写完后，这个块只占实际大小10 MB，释放多余预约的118 MB空间。如果需要写入一个1 GB的文件，HDFS还是会预约一个块来写，这个块写完后再开启下一个块，文件写完后，实际占用1 GB磁盘，释放多余预约的空间。

- 该异常通常是因为业务写文件的并发量太高，预约写Block的磁盘空间不足，导致写文件失败。

解决办法

步骤1 登录HDFS的WebUI页面，进入DataNode的JMX页面。

1. 在HDFS原生界面，选择Datanodes页面。
2. 找到对应的DataNode节点，单击Http Address地址进入DataNode详情。
3. 将url的“datanode.html”改为“jmx”就能获取到DataNode的JMX信息。

步骤2 搜索“XceiverCount”指标，当该指标的值*Block块的大小超过DataNode磁盘的容量，就说明预约写Block的磁盘空间不足。

步骤3 发生该问题，通常有以下两种方法来解决：

方法一：降低业务的并发度。

方法二：减少业务写文件的数目，将多个文件合并成一个文件来写。

----结束

12.32 HDFS 客户端无法删除超长目录

问题背景与现象

执行**hadoop fs -rm -r -f obs:**//*<obs_path>*命令，删除OBS超长目录出现如下报错：

```
2022-02-28 17:12:45,605 INFO internal.RestStorageService: OkHttp cost 19 ms to apply http request
2022-02-28 17:12:45,606 WARN internal.RestStorageService: Request failed, Response code: 400; Request
ID: 0000017F3F9A8545401491602FC8CAD9; Request path: http://wordcount01-fcq.obs.example.com/user
%2Froot%2F.Trash%2FCurrent
%2Ftest1%2F12345678901234567890123456789012345678901234567890123456789012345678901234567
8901234567890123456789012345678901234567890123456789012345678901234567890123456789012345
6789012345678901234567890123456789012345678901234567890123456789012345678901234567890123
4567890123456789012345678901234567890123456789012345678901234567890123456789012345678901
2345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567
8901234567890123456789012345678901234567890123456789012345678901234567890123456789012345
6789012345678901234567890123456789012345678901234567890123456789012345678901234567890123
4567890123456789012345678901234567890123456789012345678901234567890123456789012345678901
2345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567
8901234567890123456789012345678901234567890123456
2022-02-28 17:12:45,606 WARN services.AbstractClient: Storage|1|HTTP+XML|getObjectMetadata||||
2022-02-28 17:12:45|2022-02-28 17:12:45|||400|
2022-02-28 17:12:45,607 INFO log.AccessLogger: 2022-02-28 17:12:45 605|
com.obs.services.internal.RestStorageService|executeRequest|560|OkHttp cost 19 ms to apply http request
2022-02-28 17:12:45 606|com.obs.services.internal.RestStorageService|handleThrowable|221|Request failed,
Response code: 400; Request ID: 0000017F3F9A8545401491602FC8CAD9; Request path: http://wordcount01-
fcq.obs.example.com/user%2Froot%2F.Trash%2FCurrent
%2Ftest1%2F12345678901234567890123456789012345678901234567890123456789012345678901234567
8901234567890123456789012345678901234567890123456789012345678901234567890123456789012345
6789012345678901234567890123456789012345678901234567890123456789012345678901234567890123
4567890123456789012345678901234567890123456789012345678901234567890123456789012345678901
2345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567
8901234567890123456789012345678901234567890123456789012345678901234567890123456789012345
6789012345678901234567890123456789012345678901234567890123456789012345678901234567890123
4567890123456789012345678901234567890123456789012345678901234567890123456789012345678901
2345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567
```

```
8901234567890123456789012345678901234567890123456789012345678901234567890123456
2022-02-28 17:12:45 606|com.obs.services.AbstractClient|doActionWithResult|404|Storage|1|HTTP+XML|
getObjectMetadata||||2022-02-28 17:12:45|2022-02-28 17:12:45|||400|
```

原因分析

使用**rm**命令从HDFS中删除某些内容时，文件或目录不会立即被清除，它们将被移动到回收站Current目录（/user/*${username}*/.Trash/current）中。

解决办法

使用skipTrash命令可以跳过HDFS回收站，直接删除。使用前先设置HDFS客户端配置项“dfs.client.skipTrash.enabled=true”。

步骤1 以**root**用户登录集群任一Master节点。

步骤2 执行如下命令编辑HDFS用到的“hdfs-site.xml”文件。

vim *客户单安装目录*/**HDFS/hadoop/etc/hadoop/hdfs-site.xml**

步骤3 在hdfs-site.xml文件中增加如下内容。

```
<property>
<name>dfs.client.skipTrash.enabled</name>
<value>true</value>
</property>
```

步骤4 执行以下命令直接删除OBS超长目录。

hadoop fs -rm -r -f -skipTrash obs://*<obs_path>*

步骤5 登录集群其他Master节点，执行**步骤2**~**步骤4**，直到集群所有Master节点操作完成。

----结束

12.33 集群外节点访问 MRS HDFS 报错

问题背景与现象

集群外节点访问MRS HDFS的时候报错：Class org.apache.hadoop.hdfs.server.namenode.ha.AdaptiveFailoverProxyProvider not found。

```
java.lang.RuntimeException: java.lang.RuntimeException: java.lang.ClassNotFoundException: Class org.apache.hadoop.hdfs.server.namenode.ha.AdaptiveFailoverProxyProvider not found
        at org.apache.hadoop.conf.Configuration.getClass(Configuration.java:2696)
        at org.apache.hadoop.hdfs.NameNodeProxiesClient.getFailoverProxyProviderClass(NameNodeProxiesClient.java:296)
        at org.apache.hadoop.hdfs.NameNodeProxiesClient.createFailoverProxyProvider(NameNodeProxiesClient.java:237)
        at org.apache.hadoop.hdfs.NameNodeProxiesClient.createFailoverProxyProvider(NameNodeProxiesClient.java:225)
        at org.apache.hadoop.hdfs.NameNodeProxiesClient.createProxyWithClientProtocol(NameNodeProxiesClient.java:135)
        at org.apache.hadoop.hdfs.DFSClient.<init>(DFSClient.java:358)
        at org.apache.hadoop.hdfs.DFSClient.<init>(DFSClient.java:295)
        at org.apache.hadoop.hdfs.DistributedFileSystem.initialize(DistributedFileSystem.java:186)
        at org.apache.hadoop.fs.FileSystem.createFileSystem(FileSystem.java:3459)
        at org.apache.hadoop.fs.FileSystem.access$200(FileSystem.java:125)
        at org.apache.hadoop.fs.FileSystem$Cache.getInternal(FileSystem.java:3512)
        at org.apache.hadoop.fs.FileSystem$Cache.get(FileSystem.java:3480)
        at org.apache.hadoop.fs.FileSystem.get(FileSystem.java:490)
        at org.apache.hadoop.fs.FileSystem.get(FileSystem.java:239)
        at org.apache.hadoop.fs.FileSystem.get(FileSystem.java:474)
        at org.apache.hadoop.fs.Path.getFileSystem(Path.java:371)
        at org.apache.hadoop.fs.shell.PathData.expandAsGlob(PathData.java:329)
        at org.apache.hadoop.fs.shell.Command.expandArgument(Command.java:249)
        at org.apache.hadoop.fs.shell.Command.expandArguments(Command.java:232)
        at org.apache.hadoop.fs.shell.FsCommand.processRawArguments(FsCommand.java:106)
        at org.apache.hadoop.fs.shell.Command.run(Command.java:176)
        at org.apache.hadoop.fs.FsShell.run(FsShell.java:344)
        at org.apache.hadoop.util.ToolRunner.run(ToolRunner.java:76)
        at org.apache.hadoop.util.ToolRunner.run(ToolRunner.java:90)
        at org.apache.hadoop.fs.FsShell.main(FsShell.java:411)
Caused by: java.lang.RuntimeException: java.lang.ClassNotFoundException: Class org.apache.hadoop.hdfs.server.namenode.ha.AdaptiveFailoverProxyProvider not found
        at org.apache.hadoop.conf.Configuration.getClass(Configuration.java:2664)
        at org.apache.hadoop.conf.Configuration.getClass(Configuration.java:2688)
        ... 24 more
Caused by: java.lang.ClassNotFoundException: Class org.apache.hadoop.hdfs.server.namenode.ha.AdaptiveFailoverProxyProvider not found
        at org.apache.hadoop.conf.Configuration.getClassByName(Configuration.java:2568)
        at org.apache.hadoop.conf.Configuration.getClass(Configuration.java:2662)
        ... 25 more
```

原因分析

出现这个报错可能的场景有：

- 开源HDFS客户端访问MRS集群的HDFS时报错。
- 使用jar包连接MRS集群的HDFS（包括提交任务时连接HDFS）时报错。

解决办法

方法一：

步骤1 找到命令或者jar包使用的HDFS配置文件hdfs-site.xml。

步骤2 修改“dfs.client.failover.proxy.provider.hacluster”参数配置项如下。

```
<property>
<name>dfs.client.failover.proxy.provider.hacluster</name>
<value>org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider</value>
</property>
```

说明

也可以将上述配置项删除。

步骤3 保存文件后，重新访问MRS HDFS。

----结束

方法二：

步骤1 从Maven库下载MRS集群版本对应的hadoop-plugins版本。

下载路径：https://repo.huaweicloud.com/repository/maven/huaweicloudsdk/com/huawei/mrs/hadoop-plugins/

步骤2 将下载的jar包添加到命令或者jar包的依赖中。

----结束

12.34 NameNode 节点存在 ALM-12027 主机 PID 使用率超过阈值告警

问题背景与现象

3.1.2及之前的3.x版本集群，NameNode节点存在ALM-12027主机PID使用率超过阈值告警，节点Java进程可能出现“unable to create new native thread”报错。

原因分析

1. 使用以下命令统计节点进程的线程数并排序。

   ps -efT | awk '{print $2}' |sort -n |uniq -c |sort -n

   执行后结果如下：

```
   64 15310
54533 2346
  275 2983
  504 32139
  323 33641
  113 38454
  105 41630
  174 43672
  101 44918
   66 55187
    1 55315
   76 7057
```

2. 查看启动线程数最多的进程，案例中进程2346为NameNode进程，启动了5.4万线程，且持续增长。
3. 多次打印对应进程的jstack日志，根据jstack日志信息发现，NameNode存在大量线程处于WAITING，且长期不释放。

```
Line 200:   at java.util.concurrent.locks.LockSupport.park(LockSupport.java:175)
Line 215:   at java.util.concurrent.locks.LockSupport.park(LockSupport.java:175)
Line 230:   at java.util.concurrent.locks.LockSupport.park(LockSupport.java:175)
Line 245:   at java.util.concurrent.locks.LockSupport.park(LockSupport.java:175)
Line 260:   at java.util.concurrent.locks.LockSupport.park(LockSupport.java:175)
Line 275:   at java.util.concurrent.locks.LockSupport.park(LockSupport.java:175)

"Thread-121224" #653415 daemon prio=5 os_prio=0 tid=0x00007fb25b753000 nid=0x50ea waiting on condition [0x00007fb16a287000]
   java.lang.Thread.State: WAITING (parking)
    at sun.misc.Unsafe.park(Native Method)
    - parking to wait for  <0x000000073f5e8838> (a java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject)
    at java.util.concurrent.locks.LockSupport.park(LockSupport.java:175)
    at java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.await(AbstractQueuedSynchronizer.java:2045)
    at java.util.concurrent.LinkedBlockingQueue.take(LinkedBlockingQueue.java:442)
    at java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1074)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1134)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at java.lang.Thread.run(Thread.java:748)

   Locked ownable synchronizers:
    - None
```

结合以上问题分析如下：NameNode存在内置机制，根据WARN日志信息自动开启DEBUG日志，在环境中由于选择副本失败，导致一直启动Debug日志，不停地修改log4j，修改组件的log4j后进程会自动加载该配置文件，此时就会有新的线程自动产生，长时间后就会触发该告警。

出现这种情况时，将内置机制关闭，禁止自动修改日志级别即可恢复。

解决办法

步骤1 分别登录到集群主备NameNode节点，执行以下命令备份脚本。

cd $BIGDATA_HOME/FusionInsight_Current/*_*_NameNode/install/hadoop/sbin

cp hdfs-namenode-period-check.sh /tmp

步骤2 在主备NameNode节点编辑hdfs-namenode-period-check.sh文件。

vi hdfs-namenode-period-check.sh

在main方法中将“checkBlockplacementLog”注释掉，例如：

```
main()
{
    Log $INFO "start period check"
    checkHaState
    checkDefaultFS
    checkAutoBalancer
    checkFsMonitorDirectory
    checkAutoMover
```

```
    checkAutoDatamove
    checkAutoNodeLabelrefresh
    checkJournalNodeSync
    checkCheckpoint
    checkCleanAcls
    checkSsdMonitor
    checkOperationCollecter
    checkMapReduceDistributedCache
    #checkBlockplacementLog
    checkAutoDiskBalancer
}
```

步骤3 保存文件后，登录Manager，选择“集群 > 服务 > HDFS > 实例”，勾选所有NameNode实例，选择“更多 > 重启实例”。

----结束

12.35 集群出现 ALM-14012 Journalnode 数据不同步告警

问题背景与现象

MRS集群出现ALM-14012 Journalnode数据不同步告警。

原因分析

1. 登录告警节点，查找日志路径“/var/log/Bigdata/hdfs/nn”下Journalnode实例的startDetail.log日志信息，发现Journalnode实例停止过。
2. 分别查看告警节点和其他JournalNode节点的“/srv/BigData/journalnode/hacluster/current”路径下最新的edits日志文件，发现告警节点与其他节点存在不同步的情况。

解决办法

步骤1 登录FusionInsight Manager，选择“集群 > 服务 > HDFS > 实例”，勾选告警发生节点对应的Journalnode实例，选择“更多 > 停止实例”。

步骤2 登录告警节点，将“/srv/BigData/journalnode/hacluster/current”目录下的所有文件移动到其他新建目录下（例如“/opt/test”），保持该目录下清空状态。

步骤3 登录FusionInsight Manager，选择“集群 > 服务 > HDFS > 实例”，勾选停止的Journalnode实例，单击“启动实例”。

步骤4 等待一段时间后，观察告警是否恢复。

----结束

12.36 由于 HDFS 块丢失导致 DataNode 退服失败

问题背景与现象

在退服DataNode过程中，一直提示退服失败。

原因分析

1. 查看退服失败报错日志，日志中显示总计1564个块，有一个块一直没法被备份。

```
[2021-02-25 09:52:46]NameNode#192.168.1.51, current underReplicatedBlocks: 1, total blocks: 1564.
[2021-02-25 09:52:50]NameNode#192.168.1.51, current underReplicatedBlocks: 1, total blocks: 1564.
[2021-02-25 09:52:55]NameNode#192.168.1.51, current underReplicatedBlocks: 1, total blocks: 1564.
[2021-02-25 09:52:59]NameNode#192.168.1.51, current underReplicatedBlocks: 1, total blocks: 1564.
[2021-02-25 09:53:03]NameNode#192.168.1.51, current underReplicatedBlocks: 1, total blocks: 1564.
[2021-02-25 09:53:07]NameNode#192.168.1.51, current underReplicatedBlocks: 1, total blocks: 1564.
[2021-02-25 09:53:11]NameNode#192.168.1.51, current underReplicatedBlocks: 1, total blocks: 1564.
[2021-02-25 09:53:16]NameNode#192.168.1.51, current underReplicatedBlocks: 1, total blocks: 1564.
[2021-02-25 09:53:20]NameNode#192.168.1.51, current underReplicatedBlocks: 1, total blocks: 1564.
[2021-02-25 09:53:20]Node 192.168.1.44 decommission failed, no data was decommissioned in 30 minutes.
[2021-02-25 09:53:20]192.168.1.44#DataNode cannot be decommissioned.
```

2. 登录集群Master节点，进入HDFS客户端，执行**hdfs fsck /**命令查看损坏的块，并记录文件路径。

   例如：/tmp/hive-scratch/omm/_tez_session_dir/*xxx*-resources/*xxx*.jar。

   并且HDFS状态为“CORRUPT”

```
Erasure Coded Block Groups:
 Total size:    0 B
 Total files:   0
 Total block groups (validated):        0
 Minimally erasure-coded block groups:  0
 Over-erasure-coded block groups:       0
 Under-erasure-coded block groups:      0
 Unsatisfactory placement block groups: 0
 Average block group size:      0.0
 Missing block groups:          0
 Corrupt block groups:          0
 Missing internal blocks:       0
FSCK ended at Thu Feb 25 17:36:56 CST 2021 in 2584 milliseconds


The filesystem under path '/' is CORRUPT
```

解决办法

步骤1 请确认该损坏的块是否可以删除。

- 是，执行**步骤2**。
- 否，请联系技术支持。

步骤2 执行以下命令进入HDFS客户端。

cd *HDFS客户端安装目录*

source bigdata_env

kinit *业务用户*

步骤3 执行以下命令删除之前记录的损坏的块。

hdfs dfs -rm -skipTrash */tmp/hive-scratch/omm/_tez_session_dir/xxx-resources/xxx.jar*

步骤4 执行命令查看HDFS状态是否恢复为“HEALTHY”。

hdfs fsck /

```
Erasure Coded Block Groups:
 Total size:    0 B
 Total files:   0
 Total block groups (validated):        0
 Minimally erasure-coded block groups:  0
 Over-erasure-coded block groups:       0
 Under-erasure-coded block groups:      0
 Unsatisfactory placement block groups: 0
 Average block group size:      0.0
 Missing block groups:          0
 Corrupt block groups:          0
 Missing internal blocks:       0
FSCK ended at Thu Feb 25 17:38:38 CST 2021 in 2555 milliseconds


The filesystem under path '/' is HEALTHY
```

步骤5 再次执行DataNode退服操作。

----结束

12.37 使用 distcp 命令拷贝空文件夹报错

问题背景与现象

通过MRS客户端使用以下distcp命令，无法从HDFS复制空文件夹到OBS。

hadoop distcp -Dfs.obs.endpoint=xxx

-Dfs.obs.access.key=*xxx* **-Dfs.obs.secret.key=***xxx* **-update hdfs://hacluster/blee obs://xxx/aaa**

原因分析

如果源端（例如“blee”）为空目录，且目的端（例如“aaa”）目录不存在，系统会自动创建出来“aaa”目录，但是不会在“aaa”目录下面再创建“blee”目录。

如果源端“blee”不是空目录，且目的端“aaa”目录不存在，系统会自动创建出来“aaa”目录，并在“aaa”目录下面再创建“blee”目录，进行文件迁移。

解决办法

- 进行迁移操作时，源端目录（例如“blee”）不建议为空。
- 如果源端目录为空，在执行迁移前，需要手动创建目的端目录，即手动创建“aaa”目录。

13 使用 Hive

13.1 Hive 常见日志说明

审计日志

对于审计日志来说，记录了某个时间点某个用户从哪个IP发起对HiveServer或者MetaStore的请求以及记录执行的语句是什么。

如下的HiveServer审计日志，表示在2016-02-01 14:51:22 用户user_chen向HiveServer发起了show tables请求，客户端IP为192.168.1.18。

```
2016-02-01 14:51:22,335 | INFO  | HiveServer2-Handler-Pool: Thread-37815 | UserN
ame=user_chen    UserIP=192.168.1.18     Time=2016/02/01 14:51:22         Opertai
on=ExecuteStatement     stmt={show tables}      Resource=       Result= Detail=
| org.apache.hive.service.cli.thrift.ThriftCLIService.logAuditEvent(ThriftCLISer
vice.java:350)
```

如下的MetaStore审计日志，表示在2016-01-29 11:31:15 用户hive向MetaStore发起shutdown请求，客户端ip为192.168.1.18。

```
2016-01-29 11:31:15,451 | INFO  | pool-6-thread-70648 | ugi=hive/hadoop.hadoop.c
om@HADOOP.COM    ip=/192.168.1.18        cmd=Shutting down the object store...
 | org.apache.hadoop.hive.metastore.HiveMetaStore$HMSHandler.logAuditEvent(HiveM
etaStore.java:375)
```

在遇到诸如下述类问题时，需要查看审计日志：

1. 如客户端发起请求，但是未得到响应。由于不确定任务是卡在客户端还是服务端，可以通过审计日志查看。如果审计日志根本没有相关信息，那么说明任务终止在客户端；如果审计日志有相关信息打印，那么就需要去运行日志里进一步定位原因。
2. 查看指定时间段的任务请求个数，可通过审计日志查看在指定时间段有多少个请求。

HiveServer 运行日志

HiveServer负责接收客户端请求（SQL语句），然后编译、执行（提交到YARN或运行Local MR）、与MetaStore交互获取元数据信息等。HiveServer运行日志记录了一个SQL完整的执行过程。

通常情况下，当遇到SQL语句运行失败，首先需要查看HiveServer运行日志。

MetaStore 运行日志

通常情况下，当遇到查看HiveServer运行日志时，如遇到MetaException或者连接MetaStore失败，则需要查看MetaStore运行日志。

GC 日志查看

HiveServer和MetaStore均有GC日志，当遇到GC问题可以查看GC日志以快速定位是否是GC导致。例如遇到HiveServer或MetaStore频繁重启就需要去看下对应的GC日志。

13.2 Hive 服务启动失败

Hive服务启动失败最常见的原因是metastore实例无法连接上DBservice，可以查看metastore日志中具体的错误信息。

可能原因

1. DBservice没有初始化好Hive的元数据库hivemeta。
2. DBservice的浮动IP配置有误，导致metastore节点IP无法正确连接浮动IP，或者是在与该ip建立互信的时候失败导致metastore启动失败。

处理步骤 1

步骤1 执行以下命令：

source /opt/Bigdata/MRS_*XXX*/install/dbservice/.dbservice_profile

gsql -h *DBservice浮动IP地址* **-p 20051 -d hivemeta -U** *hive* **-W** *hive用户密码*

步骤2 如果不能正确进入交互界面，说明数据库初始化失败。如果报如下错误说明在DBservice所在的节点的配置文件可能丢失了hivemeta的配置。

```
org.postgresql.util.PSQLException: FATAL: no pg_hba.conf entry for host "192.168.0.146", database "HIVEMETA"。
```

步骤3 编辑“/srv/BigData/dbdata_service/data/pg_hba.conf”，在文件最后面追加**host hivemeta hive 0.0.0.0/0 sha256**配置。

步骤4 执行**source /opt/Bigdata/MRS_*XXX*/install/dbservice/.dbservice_profile**命令配置环境变量。

步骤5 执行**gs_ctl -D $GAUSSDATA reload #**命令使修改后的配置生效。

----结束

处理步骤 2

DBservice的浮动IP配置需要同网段内没有被使用过的IP，也就是在配置前ping不通的IP，请修改DBService浮动IP配置。

13.3 安全集群执行 set 命令的时候报错“Cannot modify xxx at runtime”

问题现象

执行set命令时报以下错误：

```
0: jdbc:hive2://192.168.1.18:21066/> set mapred.job.queue.name=QueueA;
 Error: Error while processing statement: Cannot modify mapred.job.queue.name at list of params that are allowed to be modified at runtime (state=42000,code=1)
```

处理步骤

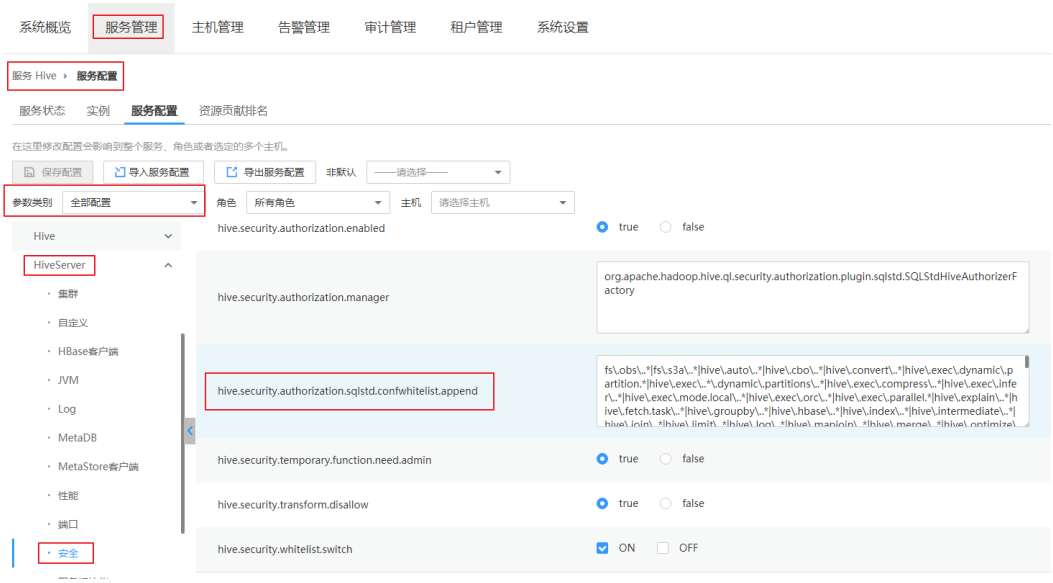
方案1：

步骤1 登录Manager界面，修改Hive参数。

- MRS Manager界面操作：登录MRS Manager页面，选择“服务管理 > Hive > 服务配置 > 全部配置 > HiveServer > 安全”。
- FusionInsight Manager界面操作：登录FusionInsight Manager页面，选择“集群 > 服务 > Hive > 配置 > 全部配置 > HiveServer > 安全”。

步骤2 将需要执行的命令参数添加到配置项“hive.security.authorization.sqlstd.confwhitelist.append”中。

步骤3 单击保存并重启HiveServer后即可。如下图所示：



----结束

方案2：

步骤1 登录Manager界面，修改Hive参数。

- MRS Manager界面操作：登录MRS Manager页面，选择“服务管理 > Hive > 服务配置 > 全部配置 > HiveServer > 安全”。

- FusionInsight Manager界面操作：登录FusionInsight Manager页面，选择“集群 > 服务 > Hive > 配置 > 全部配置 > HiveServer > 安全”。

步骤2 找到选项“hive.security.whitelist.switch”，选择“OFF”，单击保存并重启HiveServer即可。

----结束

13.4 提交 Hive 任务时如何指定队列?

问题现象

怎样在Hive提交任务的时候指定队列?

处理步骤

步骤1 在执行语句前通过如下参数设置任务队列，例如，提交任务至队列QueueA。

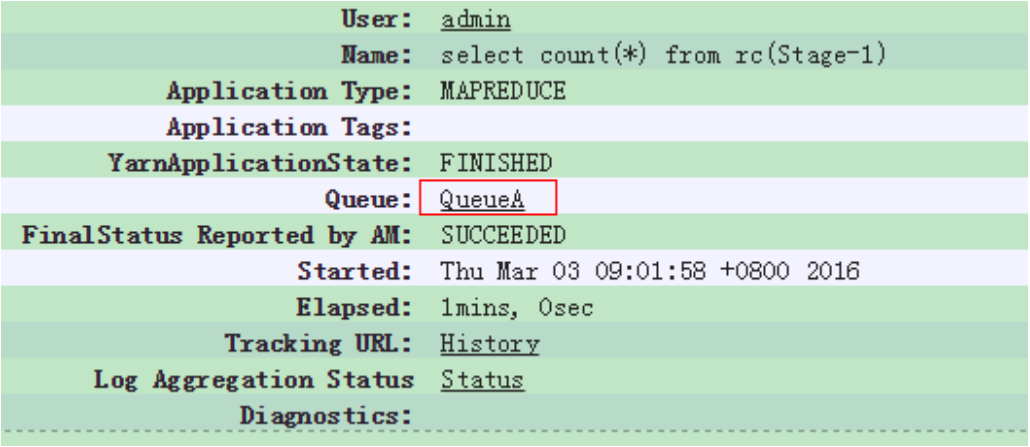
set mapred.job.queue.name=QueueA;

select count(*) from rc;

说明

队列的名称区分大小写，如写成queueA，Queuea均无效；且该队列为叶子队列，不支持提交任务到非叶子队列。

步骤2 提交任务后，可在Yarn页面看到，如下任务已经提交到队列QueueA。



----结束

13.5 怎么通过客户端设置 Map/Reduce 内存?

问题现象

客户端怎么设置Map/Reduce内存?

处理步骤

Hive在执行SQL语句前，可以通过**set**命令来设置Map/Reduce相关客户端参数。

以下为与Map/Reduce内存相关的参数：

```
set mapreduce.map.memory.mb=4096;// 每个Map Task需要的内存量
set mapreduce.map.java.opts=-Xmx3276M; // 每个Map Task 的JVM最大使用内存
set mapreduce.reduce.memory.mb=4096; // 每个Reduce Task需要的内存量
set mapreduce.reduce.java.opts=-Xmx3276M; // 每个Reduce Task 的JVM最大使用内存
set mapred.child.java.opts=-Xms1024M -Xmx3584M;//此参数为全局参数，即对Map和Reduce统一设置
```

说明

参数设置只对当前session有效。

13.6 如何在导入 Hive 表时指定输出的文件压缩格式?

问题现象

如何在导入Hive表时指定输出的文件压缩格式?

处理步骤

当前Hive支持以下几种压缩格式：

```
org.apache.hadoop.io.compress.BZip2Codec
org.apache.hadoop.io.compress.Lz4Codec
org.apache.hadoop.io.compress.DeflateCodec
org.apache.hadoop.io.compress.SnappyCodec
org.apache.hadoop.io.compress.GzipCodec
```

- 如需要全局设置，即对所有表都进行压缩，可以在Manager页面对Hive的服务配置参数进行如下全局配置：
  - hive.exec.compress.output设置为true
  - mapreduce.output.fileoutputformat.compress.codec设置为org.apache.hadoop.io.compress.BZip2Codec

    说明

    hive.exec.compress.output参数必须设置为true，才能使下边的参数选项生效。
- 如需在session级设置，只需要在执行命令前增加如下设置即可：

  set hive.exec.compress.output=true;

  set mapreduce.output.fileoutputformat.compress.codec=org.apache.hadoop.io.compress.SnappyCodec;

13.7 Hive 表 desc 描述过长导致无法完整显示

问题现象

desc描述表过长时，如何让描述显示完整?

处理步骤

步骤1 启动Hive的beeline时，设置参数maxWidth=20000即可，例如：

```
[root@192-168-1-18 logs]# beeline --maxWidth=20000
scan complete in 3ms
Connecting to
......
Beeline version 1.1.0 by Apache Hive
```

步骤2 （可选）通过**beeline -help**命令查看关于客户端显示的设置。如下：

```
 -u <database url>              the JDBC URL to connect to
 -n <username>                  the username to connect as
 -p <password>                  the password to connect as
 -d <driver class>              the driver class to use
 -i <init file>                 script file for initialization
 -e <query>                     query that should be executed
 -f <exec file>                 script file that should be executed
 --hiveconf property=value      Use value for given property
 --color=[true/false]           control whether color is used for display
 --showHeader=[true/false]      show column names in query results
 --headerInterval=ROWS;         the interval between which heades are displayed
 --fastConnect=[true/false]     skip building table/column list for tab-completion
 --autoCommit=[true/false]      enable/disable automatic transaction commit
 --verbose=[true/false]         show verbose error messages and debug info
 --showWarnings=[true/false]    display connection warnings
 --showNestedErrs=[true/false]  display nested errors
 --numberFormat=[pattern]       format numbers using DecimalFormat pattern
 --force=[true/false]           continue running script even after errors
 --maxWidth=MAXWIDTH            the maximum width of the terminal
 --maxColumnWidth=MAXCOLWIDTH   the maximum width to use when displaying columns
 --silent=[true/false]          be more silent
 --autosave=[true/false]        automatically save preferences
 --outputformat=[table/vertical/csv2/tsv2/dsv/csv/tsv]  format mode for result display
                                Note that csv, and tsv are deprecated - use csv2, tsv2 instead
 --truncateTable=[true/false]   truncate table column when it exceeds length
 --delimiterForDSV=DELIMITER    specify the delimiter for delimiter-separated values output format
(default: |)
 --isolation=LEVEL              set the transaction isolation level
 --nullemptystring=[true/false] set to true to get historic behavior of printing null as empty string
 --socketTimeOut=n              socket connection timeout interval, in second. The default value is 300.
```

----结束

13.8 Hive 表中增加分区列后再插入数据显示为 NULL

问题现象

1. 执行如下命令创建表
```
create table test_table(
col1 string,
col2 string
)
PARTITIONED BY(p1 string)
STORED AS orc tblproperties('orc.compress'='SNAPPY');
```
2. 修改表结构，添加分区并插入数据
```
alter table test_table add partition(p1='a');
insert into test_table partition(p1='a') select col1,col2 from temp_table;
```
3. 修改表结构，添加列并插入数据
```
alter table test_table add columns(col3 string);
insert into test_table partition(p1='a') select col1,col2,col3 from temp_table;
```
4. 查询test_table表数据，返回结果中列col3的值全为NULL
```
select * from test_table where p1='a'
```

5. 新添加表分区，并插入数据
```
alter table test_table add partition(p1='b');
insert into test_table partition(p1='b') select col1,col2,col3 from temp_table;
```
6. 查询test_table表数据，返回结果中列col3有不为NULL的值
```
select * from test_table where p1='b'
```

原因分析

在alter table时默认选项为RESTRICT，RESTRICT只会更改元数据，不会修改此操作之前创建的partition的表结构，而只会修改之后创建的新的partition，所以查询时旧的partition中的值全为NULL。

处理步骤

add column时加入cascade关键字即可，例如：
```
alter table test_table add columns(col3 string) cascade;
```

13.9 集群中创建的新用户执行查询 Hive 操作时无权限

问题现象

创建了新用户，但是执行查询的时候报无权限的错。

```
Error: Error while compiling statement: FAILED: HiveAccessControlException Permission denied: Principal [name=hive, type=USER] does not have following privileges for operation QUERY [[SELECT] on Object [type=TABLE_OR_VIEW, name=default.t1]] (state=42000,code=40000)
```

原因分析

创建的新用户没有Hive组件的操作权限。

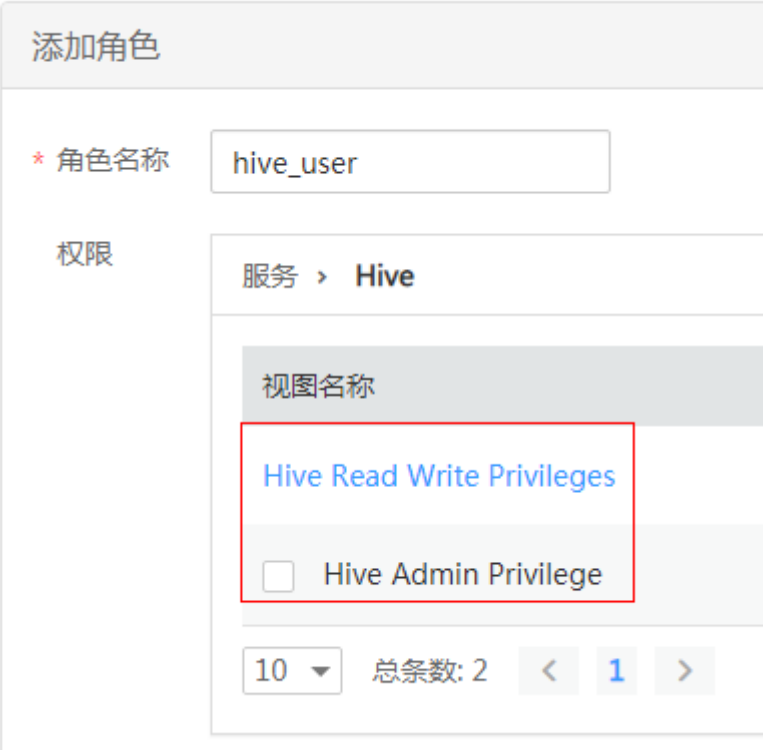
解决方案

MRS Manager界面操作：

步骤1 登录MRS Manager页面，选择“系统配置 > 角色管理 > 添加角色”。

步骤2 输入角色名称。

步骤3 在“权限”区域选择Hive，出现Hive管理员权限和Hive表的读写权限。



步骤4 选择“Hive Read Write Privileges”Hive表的读写权限，此时显示列Hive中的所有数据库。

步骤5 勾选角色需要的权限并单击“确定”完成角色创建。

步骤6 在MRS Manager页面，选择“系统配置 > 用户管理”。

步骤7 在已创建的新用户对应的“操作”列单击“修改”。

步骤8 单击“选择添加的用户组”，如需使用Hive服务，必须添加Hive组。

步骤9 单击“选择并绑定角色”，勾选**步骤5**中已创建的角色。

步骤10 单击“确定”完成用户权限的配置。

----结束

FusionInsight Manager界面操作：

步骤1 登录FusionInsight Manager。选择“系统 > 权限 > 角色”。

步骤2 单击“添加角色”，输入“角色名称”和“描述”。

步骤3 设置角色“配置资源权限”，选择“Hive读写权限”Hive表的读写权限，此时显示列Hive中的所有数据库。

步骤4 勾选角色需要的权限并单击“确定”完成角色创建。

步骤5 在FusionInsight Manager页面，选择“系统 > 权限 > 用户”。

步骤6 在已创建的新用户对应的“操作”列单击“修改”。

步骤7 单击“用户组”右侧的“添加”，如需使用Hive服务，必须添加Hive组。

步骤8 单击“角色”右侧的“添加”，勾选4中已创建的角色。

步骤9 单击“确定”完成用户权限的配置。

----结束

13.10 执行 SQL 提交任务到指定队列时报错

问题现象

执行SQL提交任务到Yarn报如下错误：

```
Failed to submit application_1475400939788_0033 to YARN :
org.apache.hadoop.security.AccessControlException: User newtest cannot submit applications to queue
root.QueueA
```

原因分析

当前登录的用户无YARN队列提交权限。

解决方案

用户无YARN队列提交权限，需要赋予YARN相应队列的提交权限。在Manager页面的“系统 >权限 > 用户”中给用户绑定队列提交权限的角色。

13.11 执行 load data inpath 命令报错

问题现象

执行load data inpath报如下错误：

- 错误1：
```
HiveAccessControlException Permission denied. Principal [name=user1, type=USER] does not have
following privileges on Object [type=DFS_URI, name=hdfs://hacluster/tmp/input/mapdata] for
operation LOAD : [OBJECT OWNERSHIP]
```
- 错误2：
```
HiveAccessControlException Permission denied. Principal [name=user1, type=USER] does not have
following privileges on Object [type=DFS_URI, name=hdfs://hacluster/tmp/input/mapdata] for
operation LOAD : [INSERT, DELETE]
```
- 错误3：
```
SemanticException [Error 10028]: Line 1:17 Path is not legal ''file:///tmp/input/mapdata'': Move from:
file:/tmp/input/mapdata to: hdfs://hacluster/user/hive/warehouse/tmp1 is not valid. Please check that
values for params "default.fs.name" and "hive.metastore.warehouse.dir" do not conflict.
```

原因分析

当前登录的用户不具备操作此目录的权限或者文件目录格式不正确。

解决方案

Hive对load data inpath命令有如下权限要求，请对照下述要求是否满足：

- 文件的owner需要为执行命令的用户。
- 当前用户需要对该文件有读、写权限。

- 当前用户需要对该文件的目录有执行权限。
- 由于load操作会将该文件移动到表对应的目录中，所以要求当前用户需要对表的对应目录有写权限。
- 要求文件的格式与表指定的存储格式相同。如创建表时指定stored as rcfile，但是文件格式为txt，则不符合要求。
- 文件必须是HDFS上的文件，不可以用file://的形式指定本地文件系统上的文件。
- 文件名不能以下横线（_）或点（.）开头，以这些开头的文件会被忽略。

  如下所示，如果用户test_hive load数据，正确的权限如下：

```
[root@192-168-1-18 duan]# hdfs dfs -ls /tmp/input2
16/03/21 14:45:07 INFO hdfs.PeerCache: SocketCache disabled.
Found 1 items
-rw-r--r--   3 test_hive hive          6 2016-03-21 14:44 /tmp/input2/input.txt
```

13.12 执行 load data local inpath 命令报错

问题现象

执行load data local inpath报如下错误：

- 错误1：

```
HiveAccessControlException Permission denied. Principal [name=user1, type=USER] does not have following privileges on Object [type=LOCAL_URI, name=file:/tmp/input/mapdata] for operation LOAD : [SELECT, INSERT, DELETE]
```

- 错误2：

```
HiveAccessControlException Permission denied. Principal [name=user1, type=USER] does not have following privileges on Object [type=LOCAL_URI, name=file:/tmp/input/mapdata] for operation LOAD : [OBJECT OWNERSHIP]
```

- 错误3：

```
SemanticException Line 1:23 Invalid path ''/tmp/input/mapdata'': No files matching path file:/tmp/input/mapdata
```

原因分析

当前登录的用户不具备操作此目录的权限或者在HiveServer所在节点上没有此目录。

解决方案

说明

通常不建议使用本地文件加载数据到hive表。 建议先将本地文件放入HDFS，然后从集群中加载数据。

Hive对load data local inpath命令有如下权限要求，请对照下述要求是否满足：

- 由于所有的命令都是发送到主HiveServer上去执行的，所以要求此文件在HiveServer节点上。
- HiveServer进程是以操作系统上的omm用户启动的，所以要求omm用户对此文件有读权限，对此文件的目录有读、执行权限。
- 文件的owner需要为执行命令的用户。
- 当前用户需要对该文件有读、写权限。
- 要求文件的格式与表指定的存储格式相同。如创建表时指定stored as rcfile，但是文件格式为txt，则不符合要求。

- 文件名不能以下横线（_）或点（.）开头，以这些开头的文件会被忽略。

13.13 执行 create external table 命令报错

问题现象

执行命令：create external table xx(xx int) stored as textfile location '/tmp/aaa/aaa'，报以下错误：

```
Permission denied. Principal [name=fantasy, type=USER] does not have following privileges on Object [type=DFS_URI, name=/tmp/aaa/aaa] for operation CREATETABLE : [SELECT, INSERT, DELETE, OBJECT OWNERSHIP] (state=42000,code=40000)
```

原因分析

当前登录的用户不具备该目录或者其父目录的读写权限。创建外部表时，会判断当前用户对指定的目录以及该目录下其它目录和文件是否有读写权限，如果该目录不存在，会去判断其父目录，依次类推。如果一直不满足就会报权限不足。而不是报指定的目录不存在。

解决方案

请确认当前用户为路径“/tmp/aaa/aaa”的owner有读写权限，如果该路径不存在，确认对其父路径有读写权限。

13.14 在 beeline 客户端执行 dfs -put 命令报错

问题现象

执行命令：

dfs -put /opt/kv1.txt /tmp/kv1.txt

报以下错误：

```
Permission denied. Principal [name=admin, type=USER] does not have following privileges onObject[type=COMMAND_PARAMS,name=[-put, /opt/kv1.txt, /tmp/kv1.txt]] for operation DFS : [ADMIN PRIVILEGE] (state=,code=1)
```

原因分析

当前登录的用户不具备操作此命令的权限。

解决方案

如果登录的当前用户具有admin角色，请用set role admin来切换成admin角色操作。如果不具备admin角色，在Manager页面中给用户绑定对应角色的权限。

13.15 执行 set role admin 命令报无权限

问题现象

执行命令：

set role admin

报下述错误：

```
0: jdbc:hive2://192.168.42.26:21066/> set role admin;
Error: Error while processing statement: FAILED: Execution Error, return code 1 from
org.apache.hadoop.hive.ql.exec.DDLTask. dmp_B doesn't belong to role admin (state=08S01,code=1)
```

原因分析

当前登录的用户不具有Hive的admin角色的权限。

解决方案

步骤1 登录Manager。

- MRS 3.x之前版本，执行**步骤7**。
- MRS 3.x及之后版本，选择“集群 > 服务 > Hive”，在服务“概览”页面右上角单击“更多”，查看“启用Ranger鉴权”是否置灰。
  - 是，执行**步骤2**。
  - 否，执行**步骤7**。

步骤2 选择“集群 > 服务 > Ranger”，单击“基本信息”区域中的“RangerAdmin”，进入Ranger WebUI界面。

步骤3 单击右上角用户名后，选择“Log Out”，退出当前用户后使用**rangeradmin**用户登录。

步骤4 在首页中单击“Settings”，选择“Roles”。

步骤5 单击“Role Name”为“admin”的角色，在“Users”区域，单击“Select User”，选择指定用户名。

步骤6 单击Add Users按钮，在对应用户名所在行勾选“Is Role Admin”，单击“Save”保存配置，操作结束。

步骤7 选择“系统 > 权限 > 角色”，添加一个拥有Hive管理员权限的角色。

步骤8 在FusionInsight Manager页面，选择“系统 > 权限 > 用户 ”。

步骤9 在指定用户对应的“操作”列单击“修改”。

步骤10 为用户绑定拥有Hive管理员权限的角色，并单击“确定”完成权限添加。

----结束

13.16 在 beeline 客户端创建 UDF 时报错

问题现象

执行命令：

create function fn_test3 as 'test.MyUDF' using jar 'hdfs:///tmp/udf2/MyUDF.jar'

报以下错误：

```
Error: Error while compiling statement: FAILED: HiveAccessControlException Permission denied: Principal [name=admin, type=USER] does not have following privileges for operation CREATEFUNCTION [[ADMIN PRIVILEGE] on Object [type=DATABASE, name=default], [ADMIN PRIVILEGE] on Object [type=FUNCTION, name=default.fn_test3]] (state=42000,code=40000)
```

原因分析

Hive中创建永久函数需要特殊的role admin。

解决方案

在执行语句前执行**set role admin命令**即可解决。

13.17 Hive 服务状态为故障如何处理

可能原因

- DBservice服务不可用会导致Hive服务状态为Bad。
- HDFS服务不可用会导致Hive服务状态为Bad。
- ZooKeeper服务不可用会导致Hive服务状态为Bad。
- LDAP/KrbServer服务不可用会导致Hive服务状态为Bad。
- MetaStore实例不可用会导致Hive服务状态为Bad。

解决方案

- DBservice服务不可用请参考**ALM-27001 DBService服务不可用**。
- HDFS服务不可用请参考**ALM-14000 HDFS服务不可用**。
- ZooKeeper服务不可用请参考**ALM-13000 ZooKeeper服务不可用**。
- LDAP/KrbServer服务不可用请参考**ALM-25000 LdapServer服务不可用**/**ALM-25500 KrbServer服务不可用**。
- MetaStore实例不可用请参考**ALM-16004 Hive服务不可用**。

13.18 Hive 服务健康状态和 Hive 实例健康状态的区别

问题现象

Hive服务健康状态和Hive实例健康状态的区别是什么？

解决方案

Hive服务的健康状态（也就是在Manager界面看到的健康状态）有Good，Bad，Partially Healthy，Unknown四种状态，四种状态除了取决于Hive本身服务的可用性（会用简单的SQL来检测Hive服务的可用性），还取决于Hive服务所依赖的其他组件的服务状态。

Hive实例分为Hiveserver和Metastore两种，健康状态有Good，Concerning，Unknown三种状态，这三种状态是通过jmx通信来判定，与实例通信正常时为Good，通信异常时为Concerning，无法通信时为Unknown。

13.19 Shell 客户端连接提示“authentication failed”

问题现象

安全集群中，HiveServer服务正常的情况下，Shell客户端中执行beeline命令失败，界面提示“authentication failed”，如下：

```
Debug is true storeKey false useTicketCache true useKeyTab false doNotPrompt false ticketCache is null
isInitiator true KeyTab is null refreshKrb5Config is false principal is null tryFirstPass is false useFirstPass is
false storePass is false clearPass is false
Acquire TGT from Cache
Credentials are no longer valid
Principal is null
null credentials from Ticket Cache
[Krb5LoginModule] authentication failed
No password provided
```

可能原因

- 客户端用户没有进行安全认证
- kerberos认证超期

解决方案

步骤1 登录Hive客户端所在节点。

步骤2 执行**source** *集群客户端安装目录*/**bigdata_env**命令。

可通过**klist**命令查看本地是否有有效票据，如下信息表明票据在16年12月24日14:11:42生效，将在16年12月25日14:11:40失效。此期间内可以使用该票据，其他时间则该票据无效。

```
klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: xxx@HADOOP.COM
Valid starting     Expires            Service principal
12/24/16 14:11:42  12/25/16 14:11:40  krbtgt/HADOOP.COM@HADOOP.COM
```

步骤3 执行**kinit** *username*进行认证，然后再使用客户端。

----结束

13.20 客户端提示访问 ZooKeeper 失败

问题现象

安全集群中，HiveServer服务正常的情况下，通过jdbc接口连接HiveServer执行sql时报出ZooKeeper认证异常"The ZooKeeper client is AuthFailed"，如下：

```
14/05/19 10:52:00 WARN utils.HAClientUtilDummyWatcher: The ZooKeeper client is AuthFailed
 14/05/19 10:52:00 INFO utils.HiveHAClientUtil: Exception thrown while reading data from znode.The possible reason may be connectionless. This is recoverable. Retrying..
 14/05/19 10:52:16 WARN utils.HAClientUtilDummyWatcher: The ZooKeeper client is AuthFailed
 14/05/19 10:52:32 WARN utils.HAClientUtilDummyWatcher: The ZooKeeper client is AuthFailed
 14/05/19 10:52:32 ERROR st.BasicTestCase: Exception: Could not establish connection to active hiveserver
 java.sql.SQLException: Could not establish connection to active hiveserver
```

或者报出无法读取"Hiveserver2 configs from ZooKeeper"，如下：

```
Exception in thread "main" java.sql.SQLException: org.apache.hive.jdbc.ZooKeeperHiveClientException: Unable to read HiveServer2 configs from ZooKeeper
at org.apache.hive.jdbc.HiveConnection.<init>(HiveConnection.java:144)
at org.apache.hive.jdbc.HiveDriver.connect(HiveDriver.java:105)
at java.sql.DriverManager.getConnection(DriverManager.java:664)
at java.sql.DriverManager.getConnection(DriverManager.java:247)
at JDBCExample.main(JDBCExample.java:82)
Caused by: org.apache.hive.jdbc.ZooKeeperHiveClientException: Unable to read HiveServer2 configs from ZooKeeper
at
org.apache.hive.jdbc.ZooKeeperHiveClientHelper.configureConnParams(ZooKeeperHiveClientHelper.java:100)
at org.apache.hive.jdbc.Utils.configureConnParams(Utils.java:509)
at org.apache.hive.jdbc.Utils.parseURL(Utils.java:429)
at org.apache.hive.jdbc.HiveConnection.<init>(HiveConnection.java:142)
... 4 more
Caused by: org.apache.zookeeper.KeeperException$ConnectionLossException: KeeperErrorCode = ConnectionLoss for /hiveserver2
at org.apache.zookeeper.KeeperException.create(KeeperException.java:99)
at org.apache.zookeeper.KeeperException.create(KeeperException.java:51)
at org.apache.zookeeper.ZooKeeper.getChildren(ZooKeeper.java:2374)
at org.apache.curator.framework.imps.GetChildrenBuilderImpl$3.call(GetChildrenBuilderImpl.java:214)
at org.apache.curator.framework.imps.GetChildrenBuilderImpl$3.call(GetChildrenBuilderImpl.java:203)
at org.apache.curator.RetryLo，op.callWithRetry(RetryLoop.java:107)
at
org.apache.curator.framework.imps.GetChildrenBuilderImpl.pathInForeground(GetChildrenBuilderImpl.java:200)
at org.apache.curator.framework.imps.GetChildrenBuilderImpl.forPath(GetChildrenBuilderImpl.java:191)
at org.apache.curator.framework.imps.GetChildrenBuilderImpl.forPath(GetChildrenBuilderImpl.java:38)
```

可能原因

- 客户端连接HiveServer时，HiveServer的地址是从ZooKeeper中自动获取，当ZooKeeper连接认证异常时，无法从ZooKeeper中获取正确的HiveServer地址。
- 在连接zookeeper认证时，需要客户端传入krb5.conf，principal，keytab等相关信息。认证失败有如下几种：
  - user.keytab路径写错。
  - user.principal写错。
  - 集群做过切换域名操作但客户端拼接url时使用旧的principal。
  - 有防火墙相关设置，导致客户端本身无法通过kerberos认证，Kerberos需要开放的端口有21730(TCP)、21731(TCP/UDP)、21732(TCP/UDP)。

解决方案

步骤1 确保用户可以正常读取客户端节点相关路径下的user.keytab文件。

步骤2 确保用户的user.principal与指定的keytab文件对应。

可通过**klist -kt keytabpath/user.keytab**查看。

步骤3 如果集群有做过切换域名操作，需要保证url中使用的principal字段是新域名。

如默认为hive/hadoop.hadoop.com@HADOOP.COM，当集群有切换域名的操作时，该字段需要进行相关修改。如域名为abc.com时，则此处应填写hive/hadoop.abc.com@ABC.COM。

步骤4 确保可以正常的认证连接HiveServer。

在客户端执行以下命令

source *客户端安装目录*/**bigdata_env**

kinit *username*

然后再使用客户端执行**beeline**，确保可以正常运行。

----结束

13.21 使用 UDF 函数时提示“Invalid function”

问题现象

在Hive客户端中使用Spark创建UDF函数时，报出"ERROR 10011","invalid function"的异常，如下：

```
Error: Error while compiling statement: FAILED: SemanticException [Error 10011]: Line 1:7 Invalid function 'test_udf' (state=42000,code=10011)
```

在多个HiveServer之间使用UDF也存在上述问题。例如，在HiveServer1中使用HiverServer2创建的UDF，如果不及时同步元数据信息，连接HiveServer1的客户端也会提示上述错误信息。

可能原因

多个HiveServer之间或者Hive与Spark之间共用的元数据未同步，导致不同HiveServer实例内存数据不一致，造成UDF不生效。

解决方案

需要将新建的UDF信息同步到HiveServer中，执行reload function操作即可。

13.22 Hive 服务状态为 Unknown

问题现象

Hive服务状态为unknown。

可能原因

Hive服务停止。

解决方案

尝试重启Hive服务。

13.23 HiveServer 或者 MetaStore 实例的健康状态为 unknown

问题现象

HiveServer或者MetaStore实例的健康状态为unknown。

可能原因

HiveServer或者MetaStore实例被停止。

解决方案

重启HiveServer或者MetaStore实例。

13.24 HiveServer 或者 MetaStore 实例的健康状态为 Concerning

问题现象

HiveServer或者MetaStore实例的健康状态为Concerning。

可能原因

HiveServer或者MetaStore实例在启动的时候发生异常，无法正常启动。如，当修改MetaStore/HiveServer GC参数时，可通过查看对应进程的启动日志，如hiveserver.out(hadoop-omm-jar-192-168-1-18.out)文件排查。 如下异常：

```
Error: Could not find or load main class Xmx2048M
```

说明java虚拟机启动时，将Xmx2048M作为java进程的启动参数而不是JVM的启动参数了，如下将符号‘-’误删掉。

```
METASTORE_GC_OPTS=Xms1024M Xmx2048M -DIgnoreReplayReqDetect
-XX\:CMSFullGCsBeforeCompaction\=1 -XX\:+UseConcMarkSweepGC
-XX\:+CMSParallelRemarkEnabled -XX\:+UseCMSCompactAtFullCollection
-XX\:+ExplicitGCInvokesConcurrent -server -XX\:MetaspaceSize\=128M
-XX\:MaxMetaspaceSize\=256M
```

解决方案

因此遇到此类异常应该检查最近的变更项，以确认是否配置有误。

```
METASTORE_GC_OPTS=Xms1024M -Xmx2048M -DIgnoreReplayReqDetect
-XX\:CMSFullGCsBeforeCompaction\=1 -XX\:+UseConcMarkSweepGC
```

```
-XX\:+CMSParallelRemarkEnabled -XX\:+UseCMSCompactAtFullCollection
-XX\:+ExplicitGCInvokesConcurrent -server -XX\:MetaspaceSize\=128M
-XX\:MaxMetaspaceSize\=256M
```

13.25 TEXTFILE 类型文件使用 ARC4 压缩时查询结果乱码

问题现象

Hive查询结果表做压缩存储（ARC4），对结果表做select * 查询时返回结果为乱码。

可能原因

Hive默认压缩格式不是ARC4格式或者未开启输出压缩。

解决方案

步骤1 在select结果乱码时，在beeline中进行如下设置。

set mapreduce.output.fileoutputformat.compress.codec=org.apache.hadoop.io.encryption.arc4.ARC4BlockCodec;

set hive.exec.compress.output=true;

步骤2 使用块解压的方式先将表导入一个新表中。

insert overwrite table tbl_result select * from tbl_source;

步骤3 再进行查询。

select * from tbl_result;

----结束

13.26 Hive 任务运行过程中失败，重试成功

问题现象

当Hive任务在正常运行时失败，在客户端报出错误，类似的错误打印：

```
Error:Invalid OperationHandler:OperationHander [opType=EXECUTE_STATEMENT,getHandleIdentifier()=XXX]
(state=,code=0)
```

而此任务提交到Yarn上的MapReduce任务运行成功。

```
0: jdbc:hive2://189.120.204.104:21066/> select count(*) from test1;
INFO  : Number of reduce tasks determined at compile time: 1
INFO  : In order to change the average load for a reducer (in bytes):
INFO  :   set hive.exec.reducers.bytes.per.reducer=<number>
INFO  : In order to limit the maximum number of reducers:
INFO  :   set hive.exec.reducers.max=<number>
INFO  : In order to set a constant number of reducers:
INFO  :   set mapreduce.job.reduces=<number>
INFO  : number of splits:1
INFO  : Submitting tokens for job: job_1484563934624_0003
INFO  : Kind: HDFS_DELEGATION_TOKEN, Service: ha-hdfs:hacluster, Ident: (HDFS_DELEGATION_TOKEN token 7 for admin)
INFO  : Kind: HIVE_DELEGATION_TOKEN, Service: HiveServer2ImpersonationToken, Ident: 00 05 61 64 6d 69 6e 05 61 64 6d 69 6e 21 68 69 76 65 2f 68 61 64 6f 6f 70 2e 68
 85 ce e4 8a 01 59 ce 92 52 e4 8e 07 d8 0c
INFO  : The url to track the job: https://189-120-204-104:26001/proxy/application_1484563934624_0003/
INFO  : Starting Job = job_1484563934624_0003, Tracking URL = https://189-120-204-104:26001/proxy/application_1484563934624_0003/
INFO  : Kill Command = /opt        3igdata/FusionInsight-Hive-1.1.0/hadoop/bin/hadoop job  -kill job_1484563934624_0003
INFO  : Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
INFO  : 2017-01-17 11:46:12,579 Stage-1 map = 0%,  reduce = 0%
INFO  : 2017-01-17 11:46:23,243 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.32 sec
Error: Invalid OperationHandle: OperationHandle [opType=EXECUTE_STATEMENT, getHandleIdentifier()=386323de-df1a-4299-826e-96368d4baf80] (state=,code=0)
0: jdbc:hive2://189.120.204.215:21066/>
```

原因分析

出错的集群有两个HiveServer实例，首先查看其中一个HiveServer日志发现里面的报错与客户端中的错误一样均是Error:Invalid OperationHandler，查看另一个HiveServer发现在出错的时间段此实例有如下类似START_UP的打印，说明那段时间进程被停止过，后来又启动成功，提交的任务本来连接的是重启过的HiveServer实例，当这个实例被停止后，任务进程连接到另一个健康的HiveServer上导致报错。

```
2017-02-15 14:40:11,309 | INFO  | main | STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting HiveServer2
STARTUP_MSG:   host = XXX-120-85-154/XXX.120.85.154
STARTUP_MSG:   args = []
STARTUP_MSG:   version = 1.3.0
```

解决办法

重新提交一次任务即可，保证在任务执行期间不手动重启HiveServer进程。

13.27 执行 select 语句时报错“Execution Error return code 2”

问题现象

执行语句select count(*) from XXX;时客户端报错：

```
Error:Error while processing statement :FAILED:Execution Error,return code 2 from ...
```

这个报错return code2说明是在执行mapreduce任务期间报错导致任务失败。

```
0: jdbc:hive2://134.160.37.21:21066/> select count(*) from src.gn_data_info_gz where day_id='18' and timenap='10';
INFO  : Number of reduce tasks determined at compile time: 1
INFO  : In order to change the average load for a reducer (in bytes):
INFO  :   set hive.exec.reducers.bytes.per.reducer=<number>
INFO  : In order to limit the maximum number of reducers:
INFO  :   set hive.exec.reducers.max=<number>
INFO  : In order to set a constant number of reducers:
INFO  :   set mapreduce.job.reduces=<number>
INFO  : number of splits:496
INFO  : Submitting tokens for job: job_1482323187492_57815
INFO  : Kind: HDFS_DELEGATION_TOKEN, Service: ha-hdfs:hacluster, Ident: (HDFS_DELEGATION_TOKEN token 1083940 for boncusermm)
INFO  : Kind: HIVE_DELEGATION_TOKEN, Service: HiveServer2ImpersonationToken, Ident: 00 0a 62 6f 6e 63 75 73 65 72 6d 6d 0a 62 6f 6e 63 75 73 65 72 6d 6d 21 68 69
74 55 8a 01 59 d4 b5 f8 55 8d 02 50 ae 8e 03 65
INFO  : The url to track the job: https://hnocno3:26001/proxy/application_1482323187492_57815/
INFO  : Starting Job = job_1482323187492_57815, Tracking URL = https://hnocno3:26001/proxy/application_1482323187492_57815/
INFO  : Kill Command = /opt/        Bigdata/FusionInsight_V100R002C60U10/FusionInsight-Hive-1.3.0/hive-1.3.0/bin/..//../hadoop/bin/hadoop job  -kill job_148232318
INFO  : Hadoop job information for Stage-1: number of mappers: 496; number of reducers: 1
INFO  : 2017-01-18 16:21:00,906 Stage-1 map = 0%,  reduce = 0%
INFO  : 2017-01-18 16:21:18,957 Stage-1 map = 1%,  reduce = 0%, Cumulative CPU 50.53 sec
INFO  : 2017-01-18 16:21:32,526 Stage-1 map = 2%,  reduce = 0%, Cumulative CPU 416.23 sec
INFO  : 2017-01-18 16:21:35,035 Stage-1 map = 5%,  reduce = 0%, Cumulative CPU 1421.09 sec
INFO  : 2017-01-18 16:21:36,331 Stage-1 map = 7%,  reduce = 0%, Cumulative CPU 2159.35 sec
INFO  : 2017-01-18 16:21:37,810 Stage-1 map = 9%,  reduce = 0%, Cumulative CPU 2548.77 sec
INFO  : 2017-01-18 16:21:39,126 Stage-1 map = 15%,  reduce = 0%, Cumulative CPU 3264.95 sec
INFO  : 2017-01-18 16:21:40,509 Stage-1 map = 20%,  reduce = 0%, Cumulative CPU 3621.79 sec
INFO  : 2017-01-18 16:21:41,710 Stage-1 map = 26%,  reduce = 0%, Cumulative CPU 3913.79 sec
INFO  : 2017-01-18 16:21:42,890 Stage-1 map = 32%,  reduce = 0%, Cumulative CPU 4202.18 sec
INFO  : 2017-01-18 16:21:44,037 Stage-1 map = 41%,  reduce = 0%, Cumulative CPU 4595.63 sec
INFO  : 2017-01-18 16:21:45,119 Stage-1 map = 49%,  reduce = 0%, Cumulative CPU 4822.15 sec
INFO  : 2017-01-18 16:21:46,213 Stage-1 map = 57%,  reduce = 0%, Cumulative CPU 5107.44 sec
INFO  : 2017-01-18 16:21:47,300 Stage-1 map = 68%,  reduce = 0%, Cumulative CPU 5405.71 sec
INFO  : 2017-01-18 16:21:48,407 Stage-1 map = 76%,  reduce = 0%, Cumulative CPU 5611.73 sec
INFO  : 2017-01-18 16:21:49,483 Stage-1 map = 85%,  reduce = 0%, Cumulative CPU 5804.64 sec
INFO  : 2017-01-18 16:21:50,565 Stage-1 map = 92%,  reduce = 0%, Cumulative CPU 5958.81 sec
INFO  : 2017-01-18 16:21:51,641 Stage-1 map = 96%,  reduce = 0%, Cumulative CPU 6041.06 sec
INFO  : 2017-01-18 16:21:52,744 Stage-1 map = 98%,  reduce = 0%, Cumulative CPU 6073.82 sec
INFO  : 2017-01-18 16:22:08,352 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 6078.4 sec
INFO  : MapReduce Total cumulative CPU time: 0 days 1 hours 41 minutes 18 seconds 400 msec
ERROR : Ended Job = job_1482323187492_57815 with errors
Error: Error while processing statement: FAILED: Execution Error, return code 2 from org.apache.hadoop.hive.ql.exec.mr.MapRedTask (state=08S01,code=2)
0: jdbc:hive2://134.160.37.21:21066/>
```

原因分析

1. 进入Yarn原生页面查看MapReduce任务的日志看到报错是无法识别到压缩方式导致错误，看文件后缀是gzip压缩，堆栈却报出是zlib方式。

```
2017-01-18 16:22:07,566 INFO [main] org.apache.hadoop.hive.ql.exec.MapOperator: 4 Close done
2017-01-18 16:22:07,572 WARN [main] org.apache.hadoop.mapred.YarnChild: Exception running child : java.io.IOException: java.io.IOException: unknown compression method
        at org.apache.hadoop.hive.io.HiveIOExceptionHandlerChain.handleRecordReaderNextException(HiveIOExceptionHandlerChain.java:121)
        at org.apache.hadoop.hive.io.HiveIOExceptionHandlerUtil.handleRecordReaderNextException(HiveIOExceptionHandlerUtil.java:77)
        at org.apache.hadoop.hive.ql.io.HiveContextAwareRecordReader.doNext(HiveContextAwareRecordReader.java:355)
        at org.apache.hadoop.hive.ql.io.HiveRecordReader.doNext(HiveRecordReader.java:79)
        at org.apache.hadoop.hive.ql.io.HiveRecordReader.doNext(HiveRecordReader.java:33)
        at org.apache.hadoop.hive.ql.io.HiveContextAwareRecordReader.next(HiveContextAwareRecordReader.java:116)
        at org.apache.hadoop.mapred.MapTask$TrackedRecordReader.moveToNext(MapTask.java:199)
        at org.apache.hadoop.mapred.MapTask$TrackedRecordReader.next(MapTask.java:185)
        at org.apache.hadoop.mapred.MapRunner.run(MapRunner.java:52)
        at org.apache.hadoop.mapred.MapTask.runOldMapper(MapTask.java:453)
        at org.apache.hadoop.mapred.MapTask.run(MapTask.java:343)
        at org.apache.hadoop.mapred.YarnChild$2.run(YarnChild.java:180)
        at java.security.AccessController.doPrivileged(Native Method)
        at javax.security.auth.Subject.doAs(Subject.java:422)
        at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1726)
        at org.apache.hadoop.mapred.YarnChild.main(YarnChild.java:174)
Caused by: java.io.IOException: unknown compression method
        at org.apache.hadoop.io.compress.zlib.ZlibDecompressor.inflateBytesDirect(Native Method)
        at org.apache.hadoop.io.compress.zlib.ZlibDecompressor.decompress(ZlibDecompressor.java:225)
        at org.apache.hadoop.io.compress.DecompressorStream.decompress(DecompressorStream.java:91)
        at org.apache.hadoop.io.compress.DecompressorStream.read(DecompressorStream.java:85)
        at java.io.InputStream.read(InputStream.java:101)
        at org.apache.hadoop.util.LineReader.fillBuffer(LineReader.java:180)
        at org.apache.hadoop.util.LineReader.readDefaultLine(LineReader.java:216)
        at org.apache.hadoop.util.LineReader.readLine(LineReader.java:174)
        at org.apache.hadoop.mapred.LineRecordReader.next(LineRecordReader.java:248)
        at org.apache.hadoop.mapred.LineRecordReader.next(LineRecordReader.java:48)
        at org.apache.hadoop.hive.ql.io.HiveContextAwareRecordReader.doNext(HiveContextAwareRecordReader.java:350)
        ... 13 more

2017-01-18 16:22:07,576 INFO [main] org.apache.hadoop.mapred.Task: Runnning cleanup for the task
```

2. 因此怀疑此语句查询的表对应的HDFS上的文件有问题，Map日志中打印出了解析的对应的文件名，将其从HDFS上下载到本地，看到是gz结尾的文件，使用**tar**命令解压报错，格式不正确无法解压。使用**file**命令查看文件属性发现此文件来自于FAT系统的压缩而非UNIX。

```
[root@hn0dn01 ~]# ls -l *.txt.gz
-rw-r--r-- 1 root root 101966463 Jan 18 20:13 20170118095958920074010 1.txt.gz
-rw-r--r-- 1 root root  90448283 Jan 18 19:55 20170118104000000074002 0.txt.gz
[root@hn0dn01 ~]# file 20170118095958920074010 1.txt.gz
20170118095958920074010 1.txt.gz: gzip compressed data, was "20170118095958920074010 1.txt", from Unix, last modified: Wed Jan 18 09:59:52 2017
[root@hn0dn01 ~]# file  20170118104000000074002 0.txt.gz
20170118104000000074002 0.txt.gz: gzip compressed data, from FAT filesystem (MS-DOS, OS/2, NT)
[root@hn0dn01 ~]# tar -zxvf  20170118104000000074002 0.txt.gz
tar: This does not look like a tar archive
tar: Skipping to next header

gzip: stdin: decompression OK, trailing garbage ignored
tar: Child returned status 2
tar: Error is not recoverable: exiting now
[root@hn0dn01 ~]#
```

解决办法

将格式不正确的文件移除HDFS目录或者替换为正确的格式的文件。

13.28 有大量分区时执行 drop partition 操作失败

问题背景与现象

执行drop partitions操作，执行异常：

```
MetaStoreClient lost connection. Attempting to reconnect. |
org.apache.hadoop.hive.metastore.RetryingMetaStoreClient.invoke(RetryingMetaStoreClient.java:187)
org.apache.thrift.transport.TTransportException
at org.apache.thrift.transport.TIOStreamTransport.read(TIOStreamTransport.java:132)
at org.apache.thrift.transport.TTransport.xxx(TTransport.java:86)
at org.apache.thrift.transport.TSaslTransport.readLength(TSaslTransport.java:376)
at org.apache.thrift.transport.TSaslTransport.readFrame(TSaslTransport.java:453)
at org.apache.thrift.transport.TSaslTransport.read(TSaslTransport.java:435)
...
```

查看对应MetaStore日志，有StackOverFlow异常：

```
2017-04-22 01:00:58,834 | ERROR | pool-6-thread-208 | java.lang.StackOverflowError
at org.datanucleus.store.rdbms.sql.SQLText.toSQL(SQLText.java:330)
at org.datanucleus.store.rdbms.sql.SQLText.toSQL(SQLText.java:339)
at org.datanucleus.store.rdbms.sql.SQLText.toSQL(SQLText.java:339)
at org.datanucleus.store.rdbms.sql.SQLText.toSQL(SQLText.java:339)
at org.datanucleus.store.rdbms.sql.SQLText.toSQL(SQLText.java:339)
```

原因分析

drop partition的处理逻辑是将找到所有满足条件的分区，将其拼接起来，最后统一删除。由于分区数过多，拼删元数据堆栈较深，出现StackOverFlow异常。

解决办法

分批次删除分区。

13.29 执行 join 操作时 localtask 启动失败

问题背景与现象

1. 执行join等操作，数据量较小时，会启动localtask执行，执行过程会报错：
```
jdbc:hive2://10.*.*.*:21066/> select a.name ,b.gender from student a join student1 b on (a.name =
b.name);
ERROR : Execution failed with exit status: 1
ERROR : Obtaining error information
ERROR :
Task failed!
Task ID:
  Stage-4
...
Error: Error while processing statement: FAILED: Execution Error, return code 1 from
org.apache.hadoop.hive.ql.exec.mr.MapredLocalTask (state=08S01,code=1)
...
```
2. 查看对应HiveServer日志，发现是启动localtask失败：
```
2018-04-25 16:37:19,296 | ERROR | HiveServer2-Background-Pool: Thread-79 | Execution failed with
exit status: 1 | org.apache.hadoop.hive.ql.session.SessionState
$LogHelper.printError(SessionState.java:1016)
2018-04-25 16:37:19,296 | ERROR | HiveServer2-Background-Pool: Thread-79 | Obtaining error
information | org.apache.hadoop.hive.ql.session.SessionState
$LogHelper.printError(SessionState.java:1016)
2018-04-25 16:37:19,297 | ERROR | HiveServer2-Background-Pool: Thread-79 |
Task failed!
Task ID:
  Stage-4
Logs:
 | org.apache.hadoop.hive.ql.session.SessionState$LogHelper.printError(SessionState.java:1016)
2018-04-25 16:37:19,297 | ERROR | HiveServer2-Background-Pool: Thread-79 | /var/log/Bigdata/hive/
hiveserver/hive.log | org.apache.hadoop.hive.ql.session.SessionState
$LogHelper.printError(SessionState.java:1016)
2018-04-25 16:37:19,297 | ERROR | HiveServer2-Background-Pool: Thread-79 | Execution failed with
exit status: 1 |
org.apache.hadoop.hive.ql.exec.mr.MapredLocalTask.executeInChildVM(MapredLocalTask.java:342)
2018-04-25 16:37:19,309 | ERROR | HiveServer2-Background-Pool: Thread-79 | FAILED: Execution
Error, return code 1 from org.apache.hadoop.hive.ql.exec.mr.MapredLocalTask |
org.apache.hadoop.hive.ql.session.SessionState$LogHelper.printError(SessionState.java:1016)
...
2018-04-25 16:37:36,438 | ERROR | HiveServer2-Background-Pool: Thread-88 | Error running hive
query:  | org.apache.hive.service.cli.operation.SQLOperation$1$1.run(SQLOperation.java:248)
org.apache.hive.service.cli.HiveSQLException: Error while processing statement: FAILED: Execution
Error, return code 1 from org.apache.hadoop.hive.ql.exec.mr.MapredLocalTask
        at org.apache.hive.service.cli.operation.Operation.toSQLException(Operation.java:339)
        at org.apache.hive.service.cli.operation.SQLOperation.runQuery(SQLOperation.java:169)
        at org.apache.hive.service.cli.operation.SQLOperation.access$200(SQLOperation.java:75)
        at org.apache.hive.service.cli.operation.SQLOperation$1$1.run(SQLOperation.java:245)
        at java.security.AccessController.doPrivileged(Native Method)
        at javax.security.auth.Subject.doAs(Subject.java:422)
        at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1710)
        at org.apache.hive.service.cli.operation.SQLOperation$1.run(SQLOperation.java:258)
        at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
        at java.util.concurrent.FutureTask.run(FutureTask.java:266)
```

```
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)
        at java.lang.Thread.run(Thread.java:745)
```

3. 查看对应HiveServer日志目录（/var/log/Bigdata/hive/hiveserver）下的hs_err_pid_*****.log，发现有内存不够的错误：

```
# There is insufficient memory for the Java Runtime Environment to continue.
# Native memory allocation (mmap) failed to map 20776943616 bytes for committing reserved memory.
 ...
```

原因分析

Hive在执行join操作，数据量小时会生成MapJoin，执行MapJoin时会生成localtask任务，localtask启动的jvm内存继承了父进程的内存。

当有多个join执行的时候，启动多个localtask，如果机器内存不够，就会导致启动localtask失败。

解决办法

步骤1 进入Hive服务配置页面：

- MRS 3.x之前版本，单击集群名称，登录集群详情页面，选择“组件管理 > Hive > 服务配置”，单击“基础配置”下拉菜单，选择“全部配置”。

  说明

  如果集群详情页面没有“组件管理”页签，请先完成IAM用户同步（在集群详情页的“概览”页签，单击“IAM用户同步”右侧的“同步”进行IAM用户同步）。

- MRS 3.x及后续版本，登录FusionInsight Manager，然后选择“集群 > *待操作的集群名称* > 服务 > Hive > 配置 > 全部配置”。

步骤2 搜索“hive.auto.convert.join”参数并修改hive的配置hive.auto.convert.join为false，保存配置并重启服务。

该参数修改后会对业务性能有一定影响。继续执行后续步骤可不影响业务性能。

步骤3 搜索“HIVE_GC_OPTS”参数并修改，把Xms调小，具体要根据业务评估，最小设置为Xmx的一半，修改完后保存配置并重启服务。

----结束

13.30 修改 hostname 后导致 WebHCat 启动失败

问题背景与现象

用户修改hostname导致WebHCat启动失败。

查看对应节点WebHCat启动日志（ /var/log/Bigdata/hive/webhcat/hive.log），发现报如下错误：

```
org.apache.hadoop.security.authentication.client.AuthenticationException: GSSException: No valid credentials provided (Mechanism level: Server not found in Kerberos database (7)
    at org.apache.hadoop.hive.om.utils.WebHCatAuthenticator.doSpnegoSequence(WebHCatAuthenticator.java:302)
    at org.apache.hadoop.hive.om.utils.WebHCatAuthenticator.authenticate(WebHCatAuthenticator.java:149)
    at org.apache.hadoop.hive.om.monitor.WebHCatHealthChecker.renewToken(WebHCatHealthChecker.java:186)
    at org.apache.hadoop.hive.om.monitor.WebHCatHealthChecker.checkWebHCat(WebHCatHealthChecker.java:119)
    at org.apache.hadoop.hive.om.monitor.WebHCatHealthChecker.run(WebHCatHealthChecker.java:168)
    at java.lang.Thread.run(Thread.java:745)
Caused by: GSSException: No valid credentials provided (Mechanism level: Server not found in Kerberos database (7) - UNKNOWN_SERVER)
    at sun.security.jgss.krb5.Krb5Context.initSecContext(Krb5Context.java:770)
    at sun.security.jgss.GSSContextImpl.initSecContext(GSSContextImpl.java:248)
    at sun.security.jgss.GSSContextImpl.initSecContext(GSSContextImpl.java:179)
    at org.apache.hadoop.hive.om.utils.WebHCatAuthenticator$1.run(WebHCatAuthenticator.java:277)
    at org.apache.hadoop.hive.om.utils.WebHCatAuthenticator$1.run(WebHCatAuthenticator.java:253)
    at java.security.AccessController.doPrivileged(Native Method)
    at javax.security.auth.Subject.doAs(Subject.java:422)
    at org.apache.hadoop.hive.om.utils.WebHCatAuthenticator.doSpnegoSequence(WebHCatAuthenticator.java:253)
    ... 5 more
Caused by: KrbException: Server not found in Kerberos database (7) - UNKNOWN_SERVER
    at sun.security.krb5.KrbTgsRep.<init>(KrbTgsRep.java:73)
    at sun.security.krb5.KrbTgsReq.getReply(KrbTgsReq.java:251)
    at sun.security.krb5.KrbTgsReq.sendAndGetCreds(KrbTgsReq.java:262)
    at sun.security.krb5.internal.CredentialsUtil.serviceCreds(CredentialsUtil.java:308)
    at sun.security.krb5.internal.CredentialsUtil.acquireServiceCreds(CredentialsUtil.java:126)
    at sun.security.krb5.Credentials.acquireServiceCreds(Credentials.java:458)
    at sun.security.jgss.krb5.Krb5Context.initSecContext(Krb5Context.java:693)
    ... 12 more
Caused by: KrbException: Identifier doesn't match expected value (906)
    at sun.security.krb5.internal.KDCRep.init(KDCRep.java:140)
    at sun.security.krb5.internal.TGSRep.init(TGSRep.java:65)
    at sun.security.krb5.internal.TGSRep.<init>(TGSRep.java:60)
    at sun.security.krb5.KrbTgsRep.<init>(KrbTgsRep.java:55)
```

原因分析

1. MRS WebHCat角色的服务端账户中涉及到hostname，如果安装完集群后再修改hostname，就会导致启动失败。
2. “/etc/hosts”中配置了一对多或者多对一的主机名和IP对应关系，导致在执行**hostname**和**hostname -i**获取不到正确的IP和hostname。

解决办法

步骤1 将修改了节点的hostname全部修改为集群初始的hostname。

步骤2 排查WebHCat所在节点的“/etc/hosts”是否配置正确。

步骤3 重启WebHCat。

----结束

13.31 集群修改域名后 Hive 样例程序运行报错

问题背景与现象

Hive的二次开发代码样例运行报No rules applied to ****的错误：

```
/AdHocClient/user.keytab
java.io.IOException: Login failure for platformUser@ADHOC.COM from keytab user.keytab: javax.security.auth.login.LoginException: java.lang.IllegalArgumentException: I
egal principal name platformUser@ADHOC.COM: org.apache.hadoop.security.authentication.util.KerberosName$NoMatchingRule: No rules applied to platformUser@ADHOC.COM
        at org.apache.hadoop.security.UserGroupInformation.loginUserFromKeytab(UserGroupInformation.java:979)
        at com.huawei.adhoc.connector.factory.LoginUtil.loginHadoop(LoginUtil.java:311)
        at com.huawei.adhoc.connector.factory.LoginUtil.login(LoginUtil.java:134)
        at com.huawei.adhoc.connector.factory.C70ConnectorFactory.getConnection(C70ConnectorFactory.java:92)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at com.huawei.adhoc.jdbc.connection.util.GetConnectionHolder70.run(ConnectionUtil.java:238)
        at java.lang.Thread.run(Thread.java:745)
Caused by: javax.security.auth.login.LoginException: java.lang.IllegalArgumentException: Illegal principal name platformUser@ADHOC.COM: org.apache.hadoop.security.auth
tication.util.KerberosName$NoMatchingRule: No rules applied to platformUser@ADHOC.COM
        at org.apache.hadoop.security.UserGroupInformation$HadoopLoginModule.commit(UserGroupInformation.java:202)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at javax.security.auth.login.LoginContext.invoke(LoginContext.java:755)
        at javax.security.auth.login.LoginContext.access$000(LoginContext.java:195)
```

原因分析

1. Hive的二次开发样例代码会加载core-site.xml，此文件默认是通过classload加载，所以使用的时候要把此配置文件放到启动程序的classpath路径下面。
2. 如果修改了集群的域名，那么core-site.xml将发生变化，需要下载最新的core-site.xml并放入到打包hive二次开发样例代码进程的classpath路径下面。

解决办法

步骤1 下载集群Hive最新的客户端，获取最新的“core-site.xml”。

步骤2 将core-site.xml放入到打包Hive二次开发样例代码进程的classpath路径下面。

----结束

13.32 DBService 超过最大连接数后导致 Hive MetaStore 异常

问题背景与现象

DBService默认最大连接数是300，如果当业务量比较大，导致连接DBService的最大连接数超过300时，MetaStore会出现异常，并报slots are reserved for non-replication superuser connections的错误：

```
2018-04-26 14:58:55,657 | ERROR | BoneCP-pool-watch-thread | Failed to acquire connection to
jdbc:postgresql://10.*.*.*:20051/hivemeta?socketTimeout=60. Sleeping for 1000 ms. Attempts left: 9 |
com.jolbox.bonecp.BoneCP.obtainInternalConnection(BoneCP.java:292)
org.postgresql.util.PSQLException: FATAL: remaining connection slots are reserved for non-replication
superuser connections
    at org.postgresql.core.v3.ConnectionFactoryImpl.readStartupMessages(ConnectionFactoryImpl.java:643)
    at org.postgresql.core.v3.ConnectionFactoryImpl.openConnectionImpl(ConnectionFactoryImpl.java:184)
    at org.postgresql.core.ConnectionFactory.openConnection(ConnectionFactory.java:64)
    at org.postgresql.jdbc2.AbstractJdbc2Connection.<init>(AbstractJdbc2Connection.java:124)
    at org.postgresql.jdbc3.AbstractJdbc3Connection.<init>(AbstractJdbc3Connection.java:28)
    at org.postgresql.jdbc3g.AbstractJdbc3gConnection.<init>(AbstractJdbc3gConnection.java:20)
    at org.postgresql.jdbc4.AbstractJdbc4Connection.<init>(AbstractJdbc4Connection.java:30)
    at org.postgresql.jdbc4.Jdbc4Connection.<init>(Jdbc4Connection.java:22)
    at org.postgresql.Driver.makeConnection(Driver.java:392)
    at org.postgresql.Driver.connect(Driver.java:266)
    at java.sql.DriverManager.getConnection(DriverManager.java:664)
    at java.sql.DriverManager.getConnection(DriverManager.java:208)
    at com.jolbox.bonecp.BoneCP.obtainRawInternalConnection(BoneCP.java:361)
    at com.jolbox.bonecp.BoneCP.obtainInternalConnection(BoneCP.java:269)
    at com.jolbox.bonecp.ConnectionHandle.<init>(ConnectionHandle.java:242)
    at com.jolbox.bonecp.PoolWatchThread.fillConnections(PoolWatchThread.java:115)
    at com.jolbox.bonecp.PoolWatchThread.run(PoolWatchThread.java:82)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)
    at java.lang.Thread.run(Thread.java:745)
```

原因分析

业务量大导致连接DBService的最大连接数超过了300，需要修改DBService的最大连接数。

解决办法

步骤1 进入DBService服务配置页面：

- MRS 3.x之前版本，单击集群名称，登录集群详情页面，选择“组件管理 > DBService > 服务配置”，单击“基础配置”下拉菜单，选择“全部配置”。

说明

如果集群详情页面没有“组件管理”页签，请先完成IAM用户同步（在集群详情页的“概览”页签，单击“IAM用户同步”右侧的“同步”进行IAM用户同步）。

- MRS 3.x及后续版本，登录FusionInsight Manager，然后选择“集群 > 服务 > DBService > 配置 > 全部配置”。

步骤2 搜索dbservice.database.max.connections配置项，并修改dbservice.database.max.connections配置的值到合适值，不能超过1000。

步骤3 保存配置，并重启受影响的服务或者实例。

步骤4 如果调整完还报超过最大连接数，需要排查业务代码，是否有连接泄露。

----结束

13.33 beeline 客户端报错“Failed to execute session hooks: over max connections”

问题背景与现象

HiveServer连接的最大连接数默认为200，当超过200时，beeline会报Failed to execute session hooks: over max connections：

```
beeline> [root@172-27-16-38 c70client]# beeline
Connecting to
jdbc:hive2://129.188.82.38:24002,129.188.82.36:24002,129.188.82.35:24002/;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=hiveserver2;sasl.qop=auth-conf;auth=KERBEROS;principal=hive/hadoop.hadoop.com@HADOOP.COM
Debug is  true storeKey false useTicketCache true useKeyTab false doNotPrompt false ticketCache is null isInitiator true KeyTab is null refreshKrb5Config is false principal is null tryFirstPass is false useFirstPass is false storePass is false clearPass is false
Acquire TGT from Cache
Principal is xxx@HADOOP.COM
Commit Succeeded

Error: Failed to execute session hooks: over max connections. (state=,code=0)
Beeline version 1.2.1 by Apache Hive
```

查看HiveServer日志（/var/log/Bigdata/hive/hiveserver/hive.log）报over max connections错误：

```
2018-05-03 04:31:56,728 | WARN  | HiveServer2-Handler-Pool: Thread-137 | Error opening session:  | org.apache.hive.service.cli.thrift.ThriftCLIService.OpenSession(ThriftCLIService.java:542)
org.apache.hive.service.cli.HiveSQLException: Failed to execute session hooks: over max connections.
    at org.apache.hive.service.cli.session.SessionManager.openSession(SessionManager.java:322)
    at org.apache.hive.service.cli.CLIService.openSessionWithImpersonation(CLIService.java:189)
    at org.apache.hive.service.cli.thrift.ThriftCLIService.getSessionHandle(ThriftCLIService.java:663)
    at org.apache.hive.service.cli.thrift.ThriftCLIService.OpenSession(ThriftCLIService.java:527)
    at org.apache.hive.service.cli.thrift.TCLIService$Processor$OpenSession.getResult(TCLIService.java:1257)
    at org.apache.hive.service.cli.thrift.TCLIService$Processor$OpenSession.getResult(TCLIService.java:1242)
    at org.apache.thrift.ProcessFunction.process(ProcessFunction.java:39)
    at org.apache.thrift.TBaseProcessor.process(TBaseProcessor.java:39)
    at org.apache.hadoop.hive.thrift.HadoopThriftAuthBridge$Server$TUGIAssumingProcessor.process(HadoopThriftAuthBridge.java:710)
    at org.apache.thrift.server.TThreadPoolServer$WorkerProcess.run(TThreadPoolServer.java:286)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)
    at java.lang.Thread.run(Thread.java:745)
Caused by: org.apache.hive.service.cli.HiveSQLException: over max connections.
    at org.apache.hadoop.hive.transporthook.SessionControllerTsaslTransportHook.checkTotalSessionNumber(SessionControllerTsaslTransportHook.java:208)
    at org.apache.hadoop.hive.transporthook.SessionControllerTsaslTransportHook.postOpen(SessionControllerTsaslTransportHook.java:163)
```

```
    at
org.apache.hadoop.hive.transporthook.SessionControllerTsaslTransportHook.run(SessionControllerTsaslTransp
ortHook.java:134)
    at org.apache.hive.service.cli.session.SessionManager.executeSessionHooks(SessionManager.java:432)
    at org.apache.hive.service.cli.session.SessionManager.openSession(SessionManager.java:314)
    ... 12 more
```

原因分析

业务量大导致连接HiveServer单个节点最大连接数超过了200，需要调大连接HiveServer实例的最大连接数。

解决办法

步骤1 进入Hive服务配置页面：

- MRS 3.x之前版本，单击集群名称，登录集群详情页面，选择“组件管理 > Hive > 服务配置”，单击“基础配置”下拉菜单，选择“全部配置”。

  说明

  如果集群详情页面没有“组件管理”页签，请先完成IAM用户同步（在集群详情页的“概览”页签，单击“IAM用户同步”右侧的“同步”进行IAM用户同步）。

- MRS 3.x及后续版本，登录FusionInsight Manager，然后选择“集群 > 服务 > Hive > 配置 > 全部配置”。

步骤2 搜索hive.server.session.control.maxconnections配置项，并修改hive.server.session.control.maxconnections配置的值到合适值，不能超过1000。

步骤3 保存配置并重启受影响的服务或者实例。

----结束

13.34 beeline 客户端报错“OutOfMemoryError”

问题背景与现象

beeline客户端查询大量数据时，报OutOFMemoryError：Java heap space，具体报错信息如下：

```
org.apache.thrift.TException: Error in calling method FetchResults
    at org.apache.hive.jdbc.HiveConnection$SynchronizedHandler.invoke(HiveConnection.java:1514)
    at com.sun.proxy.$Proxy4.FetchResults(Unknown Source)
    at org.apache.hive.jdbc.HiveQueryResultSet.next(HiveQueryResultSet.java:358)
    at org.apache.hive.beeline.BufferedRows.<init>(BufferedRows.java:42)
    at org.apache.hive.beeline.BeeLine.print(BeeLine.java:1856)
    at org.apache.hive.beeline.Commands.execute(Commands.java:873)
    at org.apache.hive.beeline.Commands.sql(Commands.java:714)
    at org.apache.hive.beeline.BeeLine.dispatch(BeeLine.java:1035)
    at org.apache.hive.beeline.BeeLine.execute(BeeLine.java:821)
    at org.apache.hive.beeline.BeeLine.begin(BeeLine.java:778)
    at org.apache.hive.beeline.BeeLine.mainWithInputRedirection(BeeLine.java:486)
    at org.apache.hive.beeline.BeeLine.main(BeeLine.java:469)
Caused by: java.lang.OutOfMemoryError: Java heap space
    at com.sun.crypto.provider.CipherCore.doFinal(CipherCore.java:959)
    at com.sun.crypto.provider.CipherCore.doFinal(CipherCore.java:824)
    at com.sun.crypto.provider.AESCipher.engineDoFinal(AESCipher.java:436)
    at javax.crypto.Cipher.doFinal(Cipher.java:2223)
    at sun.security.krb5.internal.crypto.dk.AesDkCrypto.decryptCTS(AesDkCrypto.java:414)
    at sun.security.krb5.internal.crypto.dk.AesDkCrypto.decryptRaw(AesDkCrypto.java:291)
    at sun.security.krb5.internal.crypto.Aes256.decryptRaw(Aes256.java:86)
```

```
    at sun.security.jgss.krb5.CipherHelper.aes256Decrypt(CipherHelper.java:1397)
    at sun.security.jgss.krb5.CipherHelper.decryptData(CipherHelper.java:576)
    at sun.security.jgss.krb5.WrapToken_v2.getData(WrapToken_v2.java:130)
    at sun.security.jgss.krb5.WrapToken_v2.getData(WrapToken_v2.java:105)
    at sun.security.jgss.krb5.Krb5Context.unwrap(Krb5Context.java:1058)
    at sun.security.jgss.GSSContextImpl.unwrap(GSSContextImpl.java:403)
    at com.sun.security.sasl.gsskerb.GssKrb5Base.unwrap(GssKrb5Base.java:77)
    at org.apache.thrift.transport.TSaslTransport$SaslParticipant.unwrap(TSaslTransport.java:559)
    at org.apache.thrift.transport.TSaslTransport.readFrame(TSaslTransport.java:462)
    at org.apache.thrift.transport.TSaslTransport.read(TSaslTransport.java:435)
    at org.apache.thrift.transport.TSaslClientTransport.read(TSaslClientTransport.java:37)
    at org.apache.thrift.transport.TTransport.xxx(TTransport.java:86)
    at org.apache.hadoop.hive.thrift.TFilterTransport.xxx(TFilterTransport.java:62)
    at org.apache.thrift.protocol.TBinaryProtocol.xxx(TBinaryProtocol.java:429)
    at org.apache.thrift.protocol.TBinaryProtocol.readI32(TBinaryProtocol.java:318)
    at org.apache.thrift.protocol.TBinaryProtocol.readMessageBegin(TBinaryProtocol.java:219)
    at org.apache.thrift.TServiceClient.receiveBase(TServiceClient.java:77)
    at org.apache.hive.service.cli.thrift.TCLIService$Client.recv_FetchResults(TCLIService.java:505)
    at org.apache.hive.service.cli.thrift.TCLIService$Client.FetchResults(TCLIService.java:492)
    at sun.reflect.GeneratedMethodAccessor2.invoke(Unknown Source)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:498)
    at org.apache.hive.jdbc.HiveConnection$SynchronizedHandler.invoke(HiveConnection.java:1506)
    at com.sun.proxy.$Proxy4.FetchResults(Unknown Source)
    at org.apache.hive.jdbc.HiveQueryResultSet.next(HiveQueryResultSet.java:358)
Error: Error retrieving next row (state=,code=0)
```

原因分析

- 客户查询大量数据，数据量过大。
- 客户在检索数据时使用**select * from table_name;**，进行全表查询，表内数据过多。
- beeline默认启动内存128M，查询时返回结果集过大，导致beeline无法承载导致。

解决办法

步骤1 执行**select count(*) from table_name;**前确认需要查询的数据量大小，确认是否需要在beeline中显示如此数量级的数据。

步骤2 如数量在一定范围内需要显示，请调整hive客户端的jvm参数， 在hive客户端目录/Hive下的component_env中添加export HIVE_OPTS=-Xmx1024M（具体数值请根据业务调整），并重新执行**source** *客户端目录*/**bigdata_env**配置环境变量。

----结束

13.35 输入文件数超出设置限制导致任务执行失败

问题背景与现象

Hive执行查询操作时报Job Submission failed with exception 'java.lang.RuntimeException(input file number exceeded the limits in the conf;input file num is: 2380435,max heap memory is: 16892035072,the limit conf is: 500000/4)'，此报错中具体数值根据实际情况会发生变化，具体报错信息如下：

```
ERROR : Job Submission failed with exception 'java.lang.RuntimeException(input file numbers exceeded the limits in the conf;
 input file num is: 2380435 ,
 max heap memory is: 16892035072 ,
 the limit conf is: 500000/4)'
```

```
java.lang.RuntimeException: input file numbers exceeded the limits in the conf;
 input file num is: 2380435 ,
 max heap memory is: 16892035072 ,
 the limit conf is: 500000/4
    at org.apache.hadoop.hive.ql.exec.mr.ExecDriver.checkFileNum(ExecDriver.java:545)
    at org.apache.hadoop.hive.ql.exec.mr.ExecDriver.execute(ExecDriver.java:430)
    at org.apache.hadoop.hive.ql.exec.mr.MapRedTask.execute(MapRedTask.java:137)
    at org.apache.hadoop.hive.ql.exec.Task.executeTask(Task.java:158)
    at org.apache.hadoop.hive.ql.exec.TaskRunner.runSequential(TaskRunner.java:101)
    at org.apache.hadoop.hive.ql.Driver.launchTask(Driver.java:1965)
    at org.apache.hadoop.hive.ql.Driver.execute(Driver.java:1723)
    at org.apache.hadoop.hive.ql.Driver.runInternal(Driver.java:1475)
    at org.apache.hadoop.hive.ql.Driver.run(Driver.java:1283)
    at org.apache.hadoop.hive.ql.Driver.run(Driver.java:1278)
    at org.apache.hive.service.cli.operation.SQLOperation.runQuery(SQLOperation.java:167)
    at org.apache.hive.service.cli.operation.SQLOperation.access$200(SQLOperation.java:75)
    at org.apache.hive.service.cli.operation.SQLOperation$1$1.run(SQLOperation.java:245)
    at java.security.AccessController.doPrivileged(Native Method)
    at javax.security.auth.Subject.doAs(Subject.java:422)
    at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1710)
    at org.apache.hive.service.cli.operation.SQLOperation$1.run(SQLOperation.java:258)
    at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
    at java.util.concurrent.FutureTask.run(FutureTask.java:266)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)
    at java.lang.Thread.run(Thread.java:745)

Error: Error while processing statement: FAILED: Execution Error, return code 1 from
org.apache.hadoop.hive.ql.exec.mr.MapRedTask (state=08S01,code=1)
```

原因分析

MapReduce任务提交前对输入文件数的检查策略：在提交的MapReduce任务中，允许的最大输入文件数和HiveServer最大堆内存的比值，例如500000/4（默认值），表示每4GB堆内存最大允许500000个输入文件。在输入的文件数超出此限制时则会发生此错误。

解决办法

步骤1 进入Hive服务配置页面：

- MRS 3.x之前版本，单击集群名称，登录集群详情页面，选择“组件管理 > Hive > 服务配置”，单击“基础配置”下拉菜单，选择“全部配置”。

  说明

  如果集群详情页面没有“组件管理”页签，请先完成IAM用户同步（在集群详情页的“概览”页签，单击“IAM用户同步”右侧的“同步”进行IAM用户同步）。

- MRS 3.x及后续版本，登录FusionInsight Manager，然后选择“集群 > 服务 > Hive > 配置 > 全部配置”。

步骤2 搜索hive.mapreduce.input.files2memory配置项，并修改hive.mapreduce.input.files2memory配置的值到合适值，根据实际内存和任务情况对此值进行调整。

步骤3 保存配置并重启受影响的服务或者实例。

步骤4 如调整后问题仍未解决，请根据业务情况调整HiveServer的GC参数至合理的值。

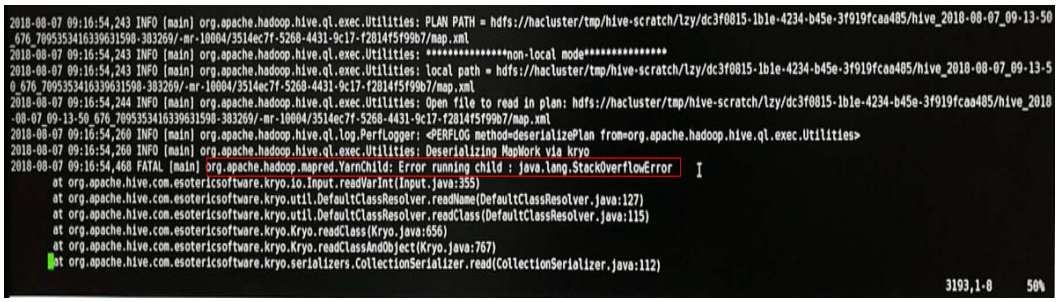
----结束

13.36 Hive 任务执行中报栈内存溢出导致任务执行失败

问题背景与现象

Hive执行查询操作时报错**Error running child : java.lang.StackOverflowError**，具体报错信息如下：

```
FATAL [main] org.apache.hadoop.mapred.YarnChild: Error running child : java.lang.StackOverflowError
at org.apache.hive.come.esotericsoftware.kryo.io.Input.readVarInt(Input.java:355)
at
org.apache.hive.come.esotericsoftware.kryo.util.DefautClassResolver.readName(DefautClassResolver.java:127)
at
org.apache.hive.come.esotericsoftware.kryo.util.DefautClassResolver.readClass(DefautClassResolver.java:115)
at org.apache.hive.come.esotericsoftware.kryo.Kryo.readClass(Kryo.java.656)
at org.apache.hive.come.esotericsoftware.kryo.kryo.readClassAnd0bject(Kryo.java:767)
at
org.apache.hive.come.esotericsoftware.kryo.serializers.collectionSerializer.read(CollectionSerializer.java:112)
```



原因分析

java.lang.StackOverflowError这是内存溢出错误的一种，即线程栈的溢出，方法调用层次过多（比如存在无限递归调用）或线程栈太小都会导致此报错。

解决办法

通过调整mapreduce阶段的map和reduce子进程JVM参数中的栈内存解决此问题，主要涉及参数为mapreduce.map.java.opts（调整map的栈内存）和mapreduce.reduce.java.opts（调整reduce的栈内存），调整方法如下（以mapreduce.map.java.opts参数为例）。

- 临时增加map内存（只针对此次beeline生效）：

  在beeline中执行如下命令**set mapreduce.map.java.opts=-Xss8G;**（具体数值请结合实际业务情况进行调整）。

- 永久增加map内存mapreduce.map.memory.mb和mapreduce.map.java.opts的值：

  a. 进入Hive服务配置页面：

     - MRS 3.x之前版本，单击集群名称，登录集群详情页面，选择“组件管理 > Hive > 服务配置”，单击“基础配置”下拉菜单，选择“全部配置”。

说明

如果集群详情页面没有“组件管理”页签，请先完成IAM用户同步（在集群详情页的“概览”页签，单击“IAM用户同步”右侧的“同步”进行IAM用户同步）。

- MRS 3.x及后续版本，登录FusionInsight Manager，然后选择“集群 > *待操作的集群名*称 > 服务 > Hive > 配置 > 全部配置”。

b. 在hiveserver自定义参数界面添加自定义参数mapreduce.map.java.opts及相应的值。

c. 保存配置并重启受影响的服务或者实例。

修改配置后需要保存，请注意参数在HiveServer自定义参数处修改，保存重启后生效（重启期间Hive服务不可用），请注意执行时间窗口。

13.37 对同一张表或分区并发写数据导致任务失败

问题背景与现象

Hive执行插入语句时，报错HDFS上文件或目录已存在或被清除，具体报错如下：

```
2019-03-18 14:34:23,016 | WARN  | HiveServer2-Background-Pool: Thread-1179606 | Failed to move to trash: hdfs://hacluster/user/hive/warehouse/frpdb.db/dw_fixed_cost_xn_temp5_f/000000_0; Force to delete it. | org.apache.hadoop.hive.common.FileUtils.moveToTrash(FileUtils.java:651)
2019-03-18 14:34:23,017 | INFO  | HiveServer2-Background-Pool: Thread-1179604 | Moved to trash: hdfs://hacluster/user/hive/warehouse/frpdb.db/dw_fixed_cost_xn_temp6_f/000000_0 | org.apache.hadoop.hive.common.FileUtils.moveToTrash(FileUtils.java:644)
2019-03-18 14:34:23,017 | ERROR | HiveServer2-Background-Pool: Thread-1179606 | Failed to delete hdfs://hacluster/user/hive/warehouse/frpdb.db/dw_fixed_cost_xn_temp5_f/000000_0 | org.apache.hadoop.hive.common.FileUtils.moveToTrash(FileUtils.java:660)
2019-03-18 14:34:23,017 | ERROR | HiveServer2-Background-Pool: Thread-1179606 | Failed with exception Destination directory hdfs://hacluster/user/hive/warehouse/frpdb.db/dw_fixed_cost_xn_temp5_f has not be cleaned up.
org.apache.hadoop.hive.ql.metadata.HiveException: Destination directory hdfs://hacluster/user/hive/warehouse/frpdb.db/dw_fixed_cost_xn_temp5_f has not be cleaned up.
at org.apache.hadoop.hive.ql.metadata.Hive.replaceFiles(Hive.java:2974)
at org.apache.hadoop.hive.ql.metadata.Hive.loadTable(Hive.java:1664)
at org.apache.hadoop.hive.ql.exec.MoveTask.execute(MoveTask.java:374)
at org.apache.hadoop.hive.ql.exec.Task.executeTask(Task.java:158)
at org.apache.hadoop.hive.ql.exec.TaskRunner.runSequentialTaskRunner(TaskRunner.java:101)
```

原因分析

1. 根据HiveServer的审计日志，确认该任务的开始时间和结束时间。
2. 在上述时间区间内，查找是否有对同一张表或分区进行插入数据的操作。
3. Hive不支持对同一张表或分区进行并发数据插入，这样会导致多个任务操作同一个数据临时目录，一个任务将另一个任务的数据移走，导致任务失败。

解决办法

修改业务逻辑，单线程插入数据到同一张表或分区。

13.38 Hive 任务失败，报没有 HDFS 目录的权限

问题背景与现象

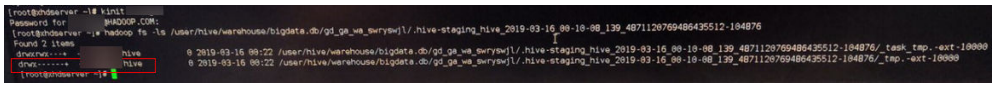
Hive任务报错，提示执行用户没有HDFS目录权限：

```
2019-04-09 17:49:19,845 | ERROR | HiveServer2-Background-Pool: Thread-3160445 | Job Submission failed
with exception 'org.apache.hadoop.security.AccessControlException(Permission denied: user=hive_quanxian,
access=READ_EXECUTE, inode="/user/hive/warehouse/bigdata.db/
gd_ga_wa_swryswjl":zhongao:hive:drwx------
at
org.apache.hadoop.hdfs.server.namenode.FSPermissionChecker.checkAccessAcl(FSPermissionChecker.java:426
)
at org.apache.hadoop.hdfs.server.namenode.FSPermissionChecker.check(FSPermissionChecker.java:329)
at
org.apache.hadoop.hdfs.server.namenode.FSPermissionChecker.checkSubAccess(FSPermissionChecker.java:30
0)
```

```
at
org.apache.hadoop.hdfs.server.namenode.FSPermissionChecker.checkPermission(FSPermissionChecker.java:24
1)
at
com.xxx.hadoop.adapter.hdfs.plugin.HWAccessControlEnforce.checkPermission(HWAccessControlEnforce.java:
69)
at
org.apache.hadoop.hdfs.server.namenode.FSPermissionChecker.checkPermission(FSPermissionChecker.java:19
0)
at org.apache.hadoop.hdfs.server.namenode.FSDirectory.checkPermission(FSDirectory.java:1910)
at org.apache.hadoop.hdfs.server.namenode.FSDirectory.checkPermission(FSDirectory.java:1894)
at
org.apache.hadoop.hdfs.server.namenode.FSDirStatAndListingOp.getContentSummary(FSDirStatAndListingO
p.java:135)
at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.getContentSummary(FSNamesystem.java:3983)
at
org.apache.hadoop.hdfs.server.namenode.NameNodeRpcServer.getContentSummary(NameNodeRpcServer.ja
va:1342)
at
org.apache.hadoop.hdfs.protocolPB.ClientNamenodeProtocolServerSideTranslatorPB.getContentSummary(Cli
entNamenodeProtocolServerSideTranslatorPB.java:925)
at org.apache.hadoop.hdfs.protocol.proto.ClientNamenodeProtocolProtos$ClientNamenodeProtocol
$2.callBlockingMethod(ClientNamenodeProtocolProtos.java)
at org.apache.hadoop.ipc.ProtobufRpcEngine$Server$ProtoBufRpcInvoker.call(ProtobufRpcEngine.java:616)
at org.apache.hadoop.ipc.RPC$Server.call(RPC.java:973)
at org.apache.hadoop.ipc.Server$Handler$1.run(Server.java:2260)
at org.apache.hadoop.ipc.Server$Handler$1.run(Server.java:2256)
at java.security.AccessController.doPrivileged(Native Method)
at javax.security.auth.Subject.doAs(Subject.java:422)
at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1781)
at org.apache.hadoop.ipc.Server$Handler.run(Server.java:2254)
)'
```

原因分析

1. 根据堆栈信息，可以看出在检查子目录的权限时失败。
   ```
   org.apache.hadoop.hdfs.server.namenode.FSPermissionChecker.checkSubAccess(FSPermissionChecker.ja
   va:300)
   ```
2. 检查HDFS上表目录下所有文件目录的权限，发现有一个目录权限为700（只有文件属主能够访问），确认存在异常目录。



解决办法

1. 确认该文件是否为手动异常导入，如不是数据文件或目录，删除该文件或目录。
2. 当无法删除时，建议修改文件或目录权限为770。

13.39 Load 数据到 Hive 表失败

问题背景与现象

用户在建表成功后，通过Load命令往此表导入数据，但导入操作中遇到如下问题：

```
.......
> LOAD DATA INPATH '/user/tester1/hive-data/data.txt' INTO TABLE employees_info;
Error: Error while compiling statement: FAILED: SemanticException Unable to load data to destination table.
Error: The file that you are trying to load does not match the file format of the destination table.
(state=42000,code=40000)
.........
```

原因分析

1. 经分析，发现在建表时没有指定存储格式，所以采用了缺省存储格式RCFile。
2. 在导入数据时，被导入数据格式是TEXTFILE格式，最终导致此问题。

解决办法

属于应用侧问题，解决办法有多种。只要保证表所指定存储格式和被导入数据格式是一致的，可以根据实际情况采用合适方法。

- 方法1：

  可以使用具有Hive表操作权限的用户在建表时指定存储格式，例如：

  CREATE TABLE IF NOT EXISTS employees_info(name STRING,age INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE;

  指定被导入数据格式为TEXTFILE。
- 方法2：

  被导入数据存储格式不能为TEXTFILE，而应为RCFile。

13.40 参考 Hive JDBC 代码用例开发的业务应用运行失败

问题背景与现象

用户参考Hive组件的“jdbc-examples”样例工程，完成其自身业务应用开发后，运行此应用失败。应用侧报如下异常：

```
.........

2017-05-11 14:33:52.174 ERROR   --- [          main] o.a.thrift.transport.TSaslTransport      : SASL negotiation failure
javax.security.sasl.SaslException: GSS initiate failed
at com.sun.security.sasl.gsskerb.GssKrb5Client.evaluateChallenge(Unknown Source)
at org.apache.thrift.transport.TSaslClientTransport.handleSaslStartMessage(TSaslClientTransport.java:94)
at org.apache.thrift.transport.TSaslTransport.open(TSaslTransport.java:271)
at org.apache.thrift.transport.TSaslClientTransport.open(TSaslClientTransport.java:37)
at org.apache.hadoop.hive.thrift.client.TUGIAssumingTransport$1.run(TUGIAssumingTransport.java:52)
at org.apache.hadoop.hive.thrift.client.TUGIAssumingTransport$1.run(TUGIAssumingTransport.java:49)
at java.security.AccessController.doPrivileged(Native Method)
at javax.security.auth.Subject.doAs(Unknown Source)
at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1711)
at org.apache.hadoop.hive.thrift.client.TUGIAssumingTransport.open(TUGIAssumingTransport.java:49)
at org.apache.hive.jdbc.HiveConnection.openTransport(HiveConnection.java:260)
at org.apache.hive.jdbc.HiveConnection.createClient(HiveConnection.java:213)
at org.apache.hive.jdbc.HiveConnection.<init>(HiveConnection.java:178)
at org.apache.hive.jdbc.HiveDriver.connect(HiveDriver.java:105)
at java.sql.DriverManager.getConnection(Unknown Source)
at java.sql.DriverManager.getConnection(Unknown Source)
at com.xxx.bigdata.hive.example.JDBCExample.main(JDBCExample.java:107)
Caused by: org.ietf.jgss.GSSException: No valid credentials provided (Mechanism level: Failed to find any Kerberos tgt)
at sun.security.jgss.krb5.Krb5InitCredential.getInstance(Unknown Source)
at sun.security.jgss.krb5.Krb5MechFactory.getCredentialElement(Unknown Source)
at sun.security.jgss.krb5.Krb5MechFactory.getMechanismContext(Unknown Source)
at sun.security.jgss.GSSManagerImpl.getMechanismContext(Unknown Source)
at sun.security.jgss.GSSContextImpl.initSecContext(Unknown Source)
at sun.security.jgss.GSSContextImpl.initSecContext(Unknown Source)
... 17 common frames omitted
......
```

原因分析

1. 初步分析怀疑是没有完成kerberos认证就去进行业务交互。
2. 深入分析日志发现日志上虽然有打印“com.xxx.bigdata.security.LoginUtil - Login success!!!!!!!!!!!!!!”，但没打印“org.apache.hadoop.security.UserGroupInformation : Login successful...”。

   分析代码，发现：

```
/*     */   @InterfaceAudience.Public
/*     */   @InterfaceStability.Evolving
/*     */   public static synchronized void loginUserFromKeytab(String user, String path)
/*     */     throws IOException
/*     */   {
/*  958 */     if (!isSecurityEnabled()) {
/*  959 */       return;
/*     */     }
......
```

3. 分析“isSecurityEnabled()”，发现是否要发起认证，还需要判断configuration中是否有配置“hadoop.security.authentication”为“kerberos”。

   本Hive业务应用确实没有正确设置此配置，所以被认为不需要做kerberos认证。

   分析Hive组件的“jdbc-examples”样例工程，不存在类似问题，是因为该工程的classpath路径下，存在core-site.xml配置文件，此配置文件上设置“hadoop.security.authentication”为“kerberos”。

解决办法

属于用户使用不当。对于本业务应用来说，若要解决此问题，可以参考如下几种办法：

- 方法1：

  直接参考Hive组件的“jdbc-examples”样例工程，将core-site.xml配置文件放在classpath路径下。

- 方法2：

  在代码中，显式加载配置文件core-site.xml，例如：

```
......
    conf = new Configuration();
    String userdir = System.getProperty("user.dir") + File.separator + "conf" + File.separator;
    conf.addResource(new Path(userdir + "core-site.xml"));
......
```

- 方法3：

  在代码中，设置“hadoop.security.authentication”为“kerberos”，例如：

```
......
    CONF = new Configuration();
    CONF.set("hadoop.security.authentication", "kerberos");
......
```

13.41 HiveServer 和 HiveHCat 进程故障

用户问题

客户集群HiveServer和WebHCat进程状态均为故障。

问题现象

客户MRS集群Master2节点上的HiveServer和WebHCat进程状态显示为故障，重启之后仍为故障状态。

原因分析

在Manager界面单独启动故障的HiveServer进程，登录后台查找hiveserver.out日志中对应时间点的报错，报错信息为：error parsing conf mapred-site.xml和Premature end of file。然后重启WebHCat也发现同样报错，原因即为解析mapred-site.xml文件错误。

处理步骤

1. 以**root**用户登录Master2节点。
2. 执行**find / -name 'mapred-site.xml'**命令获取mapred-site.xml文件所在位置。
   - HiveServer对应路径为“/opt/Bigdata/*集群版本*/1_13_HiveServer/etc/mapred-site.xml”。
   - WebHCat对应路径为“/opt/Bigdata/*集群版本*/1_13_WebHCat/etc/mapred-site.xml”。
3. 确认mapred-site.xml文件是否有异常，该案例中该配置文件内容为空导致解析失败。
4. 修复mapred-site.xml文件，将Master1节点上对应目录下的配置文件用**scp**命令拷贝到Master2节点对应目录替换原文件。
5. 执行**chown omm:wheel mapred-site.xml**命令更改所属组和用户。
6. 在Manager界面重启故障的HiveServer和WebHCat进程，恢复正常。

13.42 MRS Hive 连接 ZooKeeper 时报错“ConnectionLoss for hiveserver2”

问题现象

使用MRS 1.8集群的Hive 1.2.1通过Hive的JDBC接口连接MRS集群成功，但是使用MRS 1.9.0集群的Hive 2.3.2，通过Hive的JDBC接口连接MRS集群进行计算任务报错。

报错信息如下：

```
Caused by: org.apache.zookeeper.KeeperException$ConnectionLossException: KeeperErrorCode =
ConnectionLoss for /hiveserver2
```

原因分析

MRS 1.8版本和开源版本认证方式一样，在Hive的URL中直接将Kerberos认证的文件传进去进行认证。而在MRS 1.9版本中，此方法行不通，需要先进行Kerberos认证，并且需要获取一些别的配置信息，然后再进行URL拼接。

处理步骤

请参考官方MRS 1.9样例工程中hive-examples的认证连接，样例地址请参考**开发指南**。

建议与总结

在官方MRS 1.9的样例基础进行代码开发。

13.43 Hive 执行 insert into 语句报错

问题现象

使用MRS Hive执行一条SQL，有如下报错：

图 13-1 使用 MRS Hive 执行 SQL 报错

```
0_762_9950469685432585 54-19104/-local-10004/HashTable-Stage-7/MapJoin-mapfile121651--.hashtable
2020-06-02 17:10:02     Uploaded 1 File to: file:/opt/Bigdata/tmp/hivelocaltmp/3c3889d8-827f-4454-88aa-c47e57127d9d/hive_2020-06-02_17-08-50_762_9950469685432585 54-19104/-local-10
ashTable-Stage-7/MapJoin-mapfile121651--.hashtable (304884 bytes)
2020-06-02 17:10:02     End of local task; Time Taken: 5.211 sec.
Error: org.apache.hive.service.cli.HiveSQLException: Error while processing statement: FAILED: Execution Error, return code 1 from org.apache.hadoop.hive.ql.exec.ColumnStatsTask
        at org.apache.hive.service.cli.operation.Operation.toSQLException(Operation.java:380)
        at org.apache.hive.service.cli.operation.SQLOperation.runQuery(SQLOperation.java:268)
        at org.apache.hive.service.cli.operation.SQLOperation.access$800(SQLOperation.java:93)
        at org.apache.hive.service.cli.operation.SQLOperation$BackgroundWork$1.run(SQLOperation.java:379)
        at java.security.AccessController.doPrivileged(Native Method)
        at javax.security.auth.Subject.doAs(Subject.java:422)
        at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1840)
        at org.apache.hive.service.cli.operation.SQLOperation$BackgroundWork.run(SQLOperation.java:393)
        at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
        at java.util.concurrent.FutureTask.run(FutureTask.java:266)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        at java.lang.Thread.run(Thread.java:748) (state=08S01,code=1)
```

原因分析

1. 查看HiveServer日志，在对应时间点，有如下的报错信息。

   图 13-2 HiveServer 日志

```
    at org.apache.hadoop.hive.ql.Driver.run(Driver.java:1238)
    at org.apache.hadoop.hive.ql.Driver.run(Driver.java:1233)
    at org.apache.hive.service.cli.operation.SQLOperation.runQuery(SQLOperation.java:266)
    at org.apache.hive.service.cli.operation.SQLOperation.access$800(SQLOperation.java:93)
    at org.apache.hive.service.cli.operation.SQLOperation$BackgroundWork$1.run(SQLOperation.java:379)
    at java.security.AccessController.doPrivileged(Native Method)
    at javax.security.auth.Subject.doAs(Subject.java:422)
    at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1840)
    at org.apache.hive.service.cli.operation.SQLOperation$BackgroundWork.run(SQLOperation.java:393)
    at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
    at java.util.concurrent.FutureTask.run(FutureTask.java:266)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at java.lang.Thread.run(Thread.java:748)
  org.apache.hadoop.hive.ql.metadata.Hive.setPartitionColumnStatistics(Hive.java:3783)
2020-06-02 16:11:03,771 | ERROR | HiveServer2-Background-Pool: Thread-2440344 | Failed to run column stats task | org.apache.hadoop.hive.ql.exec.ColumnStatsTask.execute(ColumnStatsTask.java:433)
org.apache.hadoop.hive.ql.metadata.HiveException: org.apache.thrift.transport.TTransportException
    at org.apache.hadoop.hive.ql.metadata.Hive.setPartitionColumnStatistics(Hive.java:3783) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hadoop.hive.ql.exec.ColumnStatsTask.persistColumnStats(ColumnStatsTask.java:420) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hadoop.hive.ql.exec.ColumnStatsTask.execute(ColumnStatsTask.java:431) [hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hadoop.hive.ql.exec.Task.executeTask(Task.java:199) [hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hadoop.hive.ql.exec.TaskRunner.runSequential(TaskRunner.java:100) [hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hadoop.hive.ql.Driver.launchTask(Driver.java:2185) [hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hadoop.hive.ql.Driver.execute(Driver.java:1841) [hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hadoop.hive.ql.Driver.runInternal(Driver.java:1527) [hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hadoop.hive.ql.Driver.run(Driver.java:1238) [hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hadoop.hive.ql.Driver.run(Driver.java:1233) [hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hive.service.cli.operation.SQLOperation.runQuery(SQLOperation.java:266) [hive-service-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hive.service.cli.operation.SQLOperation.access$800(SQLOperation.java:93) [hive-service-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hive.service.cli.operation.SQLOperation$BackgroundWork$1.run(SQLOperation.java:379) [hive-service-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at java.security.AccessController.doPrivileged(Native Method) ~[?:1.8.0_232]
    at javax.security.auth.Subject.doAs(Subject.java:422) [?:1.8.0_232]
    at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1840) [hadoop-common-2.8.3-mrs-1.9.0.jar:?]
    at org.apache.hive.service.cli.operation.SQLOperation$BackgroundWork.run(SQLOperation.java:393) [hive-service-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511) [?:1.8.0_232]
    at java.util.concurrent.FutureTask.run(FutureTask.java:266) [?:1.8.0_232]
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149) [?:1.8.0_232]
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624) [?:1.8.0_232]
    at java.lang.Thread.run(Thread.java:748) [?:1.8.0_232]
Caused by: org.apache.thrift.transport.TTransportException
    at org.apache.thrift.transport.TIOStreamTransport.read(TIOStreamTransport.java:132) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.thrift.transport.TTransport.readAll(TTransport.java:86) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.thrift.transport.TSaslTransport.readLength(TSaslTransport.java:376) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.thrift.transport.TSaslTransport.readFrame(TSaslTransport.java:453) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.thrift.transport.TSaslTransport.read(TSaslTransport.java:435) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.thrift.transport.TSaslClientTransport.read(TSaslClientTransport.java:37) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.thrift.transport.TTransport.readAll(TTransport.java:86) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hadoop.hive.thrift.TFilterTransport.readAll(TFilterTransport.java:62) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.thrift.protocol.TBinaryProtocol.readAll(TBinaryProtocol.java:429) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.thrift.protocol.TBinaryProtocol.readI32(TBinaryProtocol.java:318) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.thrift.protocol.TBinaryProtocol.readMessageBegin(TBinaryProtocol.java:219) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.thrift.TServiceClient.receiveBase(TServiceClient.java:77) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hadoop.hive.metastore.api.ThriftHiveMetastore$Client.recv_set_aggr_stats_for(ThriftHiveMetastore.java:3586) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hadoop.hive.metastore.api.ThriftHiveMetastore$Client.set_aggr_stats_for(ThriftHiveMetastore.java:3573) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hadoop.hive.metastore.HiveMetaStoreClient.setPartitionColumnStatistics(HiveMetaStoreClient.java:1718) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hadoop.hive.ql.metadata.SessionHiveMetaStoreClient.setPartitionColumnStatistics(SessionHiveMetaStoreClient.java:355) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at sun.reflect.GeneratedMethodAccessor192.invoke(Unknown Source) ~[?:?]
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) ~[?:1.8.0_232]
    at java.lang.reflect.Method.invoke(Method.java:498) ~[?:1.8.0_232]
    at org.apache.hadoop.hive.metastore.RetryingMetaStoreClient.invoke(RetryingMetaStoreClient.java:173) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at com.sun.proxy.$Proxy35.setPartitionColumnStatistics(Unknown Source) ~[?:?]
    at sun.reflect.GeneratedMethodAccessor192.invoke(Unknown Source) ~[?:?]
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) ~[?:1.8.0_232]
    at java.lang.reflect.Method.invoke(Method.java:498) ~[?:1.8.0_232]
    at org.apache.hadoop.hive.metastore.HiveMetaStoreClient$SynchronizedHandler.invoke(HiveMetaStoreClient.java:2376) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at com.sun.proxy.$Proxy35.setPartitionColumnStatistics(Unknown Source) ~[?:?]
    at org.apache.hadoop.hive.ql.metadata.Hive.setPartitionColumnStatistics(Hive.java:3780) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    ... 21 more
```

2. 在如上报错信息中未发现重要信息，但从堆栈中发现metadata字样，怀疑报错是和MetaStore有关。

   图 13-3 堆栈中 metadata 字样

```
2020-06-02 16:11:03,771 | ERROR | HiveServer2-Background-Pool: Thread-2440344 | Failed to run column stats task | org.apache.hadoop.hive.ql.exec.ColumnStatsTask.execute(ColumnStatsTask.java:433)
org.apache.hadoop.hive.ql.metadata.HiveException: org.apache.thrift.transport.TTransportException
    at org.apache.hadoop.hive.ql.metadata.Hive.setPartitionColumnStatistics(Hive.java:3783) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
```

3. 查看MetaStore日志，发现如下报错。

图 13-4 MetaStore 日志

```
| org.apache.hadoop.hive.metastore.RetryingHMSHandler.invokeInternal(RetryingHMSHandler.java:204)
[2020-06-02 16:19:28,125] | ERROR | pool-12-thread-155 | Error occurred during processing of message. | org.apache.thrift.server.TThreadPoolServer$WorkerProcess.run(TThreadPoolServer.java:297)
org.datanucleus.exceptions.NucleusDataStoreException: Put request failed [UPDATE PARTITION_PARAMS SET PARAM_VALUE = ? WHERE PART_ID=? AND PARAM_KEY=?]
    at org.datanucleus.store.rdbms.scostore.JoinMapStore.put(JoinMapStore.java:318) ~[datanucleus-rdbms-4.1.19.jar:?]
    at org.datanucleus.store.types.wrappers.backed.Map.put(Map.java:653) ~[datanucleus-core-4.1.17.jar:?]
    at org.apache.hadoop.hive.common.StatsSetupConst.setColumnStatsState(StatsSetupConst.java:251) ~[hive-common-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hadoop.hive.metastore.ObjectStore.updatePartitionColumnStatistics(ObjectStore.java:7094) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at sun.reflect.GeneratedMethodAccessor99.invoke(Unknown Source) ~[?:?]
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) ~[?:1.8.0_232]
    at java.lang.reflect.Method.invoke(Method.java:498) ~[?:1.8.0_232]
    at org.apache.hadoop.hive.metastore.RawStoreProxy.invoke(RawStoreProxy.java:101) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at com.sun.proxy.$Proxy25.updatePartitionColumnStatistics(Unknown Source) ~[?:?]
    at org.apache.hadoop.hive.metastore.HiveMetaStore$HMSHandler.updatePartitonColStats(HiveMetaStore.java:5138) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hadoop.hive.metastore.HiveMetaStore$HMSHandler.set_aggr_stats_for(HiveMetaStore.java:6726) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at sun.reflect.GeneratedMethodAccessor75.invoke(Unknown Source) ~[?:?]
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) ~[?:1.8.0_232]
    at java.lang.reflect.Method.invoke(Method.java:498) ~[?:1.8.0_232]
    at org.apache.hadoop.hive.metastore.RetryingHMSHandler.invokeInternal(RetryingHMSHandler.java:148) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hadoop.hive.metastore.RetryingHMSHandler.invoke(RetryingHMSHandler.java:107) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at com.sun.proxy.$Proxy33.set_aggr_stats_for(Unknown Source) ~[?:?]
    at org.apache.hadoop.hive.metastore.api.ThriftHiveMetastore$Processor$set_aggr_stats_for.getResult(ThriftHiveMetastore.java:13239) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hadoop.hive.metastore.api.ThriftHiveMetastore$Processor$set_aggr_stats_for.getResult(ThriftHiveMetastore.java:13223) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.thrift.ProcessFunction.process(ProcessFunction.java:39) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.thrift.TBaseProcessor.process(TBaseProcessor.java:39) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hadoop.hive.thrift.HadoopThriftAuthBridge$Server$TUGIAssumingProcessor$1.run(HadoopThriftAuthBridge.java:594) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hadoop.hive.thrift.HadoopThriftAuthBridge$Server$TUGIAssumingProcessor$1.run(HadoopThriftAuthBridge.java:589) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at java.security.AccessController.doPrivileged(Native Method) ~[?:1.8.0_232]
    at javax.security.auth.Subject.doAs(Subject.java:422) ~[?:1.8.0_232]
    at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1840) ~[hadoop-common-2.8.3-mrs-1.9.0.jar:?]
    at org.apache.hadoop.hive.thrift.HadoopThriftAuthBridge$Server$TUGIAssumingProcessor.process(HadoopThriftAuthBridge.java:589) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.thrift.server.TThreadPoolServer$WorkerProcess.run(TThreadPoolServer.java:286) ~[hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149) [?:1.8.0_232]
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624) [?:1.8.0_232]
    at java.lang.Thread.run(Thread.java:748) [?:1.8.0_232]
Caused by: org.datanucleus.store.rdbms.exceptions.MappedDatastoreException: UPDATE PARTITION_PARAMS SET PARAM_VALUE = ? WHERE PART_ID=? AND PARAM_KEY=?
    at org.datanucleus.store.rdbms.scostore.JoinMapStore.internalUpdate(JoinMapStore.java:1020) ~[datanucleus-rdbms-4.1.19.jar:?]
    at org.datanucleus.store.rdbms.scostore.JoinMapStore.put(JoinMapStore.java:304) ~[datanucleus-rdbms-4.1.19.jar:?]
    ... 30 more
Caused by: org.postgresql.util.PSQLException: ERROR: value too long for type character varying(4000)
    at org.postgresql.core.v3.QueryExecutorImpl.receiveErrorResponse(QueryExecutorImpl.java:2199) ~[gsjdbc4-V100R003C10SPC115.jar:?]
    at org.postgresql.core.v3.QueryExecutorImpl.processResults(QueryExecutorImpl.java:1928) ~[gsjdbc4-V100R003C10SPC115.jar:?]
    at org.postgresql.core.v3.QueryExecutorImpl.execute(QueryExecutorImpl.java:348) ~[gsjdbc4-V100R003C10SPC115.jar:?]
    at org.postgresql.jdbc2.AbstractJdbc2Statement.execute(AbstractJdbc2Statement.java:545) ~[gsjdbc4-V100R003C10SPC115.jar:?]
    at org.postgresql.jdbc2.AbstractJdbc2Statement.executeWithFlags(AbstractJdbc2Statement.java:419) ~[gsjdbc4-V100R003C10SPC115.jar:?]
    at org.postgresql.jdbc2.AbstractJdbc2Statement.executeUpdate(AbstractJdbc2Statement.java:365) ~[gsjdbc4-V100R003C10SPC115.jar:?]
    at com.jolbox.bonecp.PreparedStatementHandle.executeUpdate(PreparedStatementHandle.java:205) ~[bonecp-0.8.0.RELEASE.jar:?]
    at org.datanucleus.store.rdbms.ParamLoggingPreparedStatement.executeUpdate(ParamLoggingPreparedStatement.java:393) ~[datanucleus-rdbms-4.1.19.jar:?]
    at org.datanucleus.store.rdbms.SQLController.executeStatementUpdate(SQLController.java:431) ~[datanucleus-rdbms-4.1.19.jar:?]
    at org.datanucleus.store.rdbms.scostore.JoinMapStore.internalUpdate(JoinMapStore.java:1010) ~[datanucleus-rdbms-4.1.19.jar:?]
    at org.datanucleus.store.rdbms.scostore.JoinMapStore.put(JoinMapStore.java:304) ~[datanucleus-rdbms-4.1.19.jar:?]
    ... 30 more
2020-06-02 16:19:28,125 | INFO | pool-12-thread-155 | 156: Cleaning up thread local RawStore... | org.apache.hadoop.hive.metastore.HiveMetaStore$HMSHandler.logInfo(HiveMetaStore.java:885)
```

查看如上错误的上下文，确定是本次执行SQL的报错，在报错信息里面发现如下内容：

```
Caused by: org.postgresql.util.PSQLException: ERROR: value too long for type character varying(4000)
```

确认是该条SQL对表的操作，所有列的字节长度超过4000的限制，导致SQL执行失败，需要修改该限制。

处理步骤

步骤1 以**root**用户登录集群任意一个Master节点，并执行**su - omm**命令切换到**omm**用户。

步骤2 执行如下命令登录数据库。命令中如果携带认证密码信息可能存在安全风险，在执行命令前建议关闭系统的history命令记录功能，避免信息泄露。

gsql -p 20051 -d hivemeta -U *username* **-W** *password*

步骤3 执行如下命令修改限制。

alter table PARTITION_PARAMS alter column PARAM_VALUE type varchar(6000);

----结束

13.44 增加 Hive 表字段超时

用户问题

增加Hive表字段报错。

问题现象

Hive对包含10000+分区的表执行**ALTER TABLE table_name ADD COLUMNS(column_name string) CASCADE;**，报错如下：

```
Timeout when executing method: alter_table_with_environment_context; 600525ms exceeds 600000ms
```

原因分析

1. MetaStore客户端连接超时，MRS默认MetaStore客户端和服务端连接的超时时间是600s，在Manager页面调大“hive.metastore.client.socket.timeout”为“3600s”。
2. 出现另一个报错：
```
Error: org.apache.hive.service.cli.HiveSQLException: Error while processing statement: FAILED: Execution Error, return code 1 from org.apache.hadoop.hive.ql.exec.DDLTask. Unable to alter table. java.net.SocketTimeoutException: Read timed out
```
   Metastore元数据JDBC连接超时，默认60ms。
3. 调大javax.jdo.option.ConnectionURL中socketTimeout=60000，仍然产生最初的报错：
```
Timeout when executing method: alter_table_with_environment_context;3600556ms exceeds 3600000ms
```
4. 尝试调大hive.metastore.batch.retrieve.max、hive.metastore.batch.retrieve.table.partition.max、dbservice.database.max.connections等参数均未能解决。
5. 怀疑是GaussDB的问题，因为增加字段会遍历每个分区执行getPartitionColumnStatistics和alterPartition。
6. 使用**omm**用户执行**gsql -p 20051 -U omm -W** *password* **-d hivemeta**登录Hive元数据库。命令中如果携带认证密码信息可能存在安全风险，在执行命令前建议关闭系统的history命令记录功能，避免信息泄露。
7. 执行**select * from pg_locks;**没有发现锁等待。
8. 执行**select * from pg_stat_activity;**发现进程执行时间较长。
```
SELECT 'org.apache.hadoop.hive.metastore.model.MPartitionColumnStatistics'AS NUCLEUS_TYPE,A0.AVG_COL_LEN,A0."COLUMN_NAME",A0.COLUMN_TYPE,A0.DB_NAME,A0.BIG_DECIMAL_HIGH_VALUE,A0.BIG_DECIMAL_LOW_VALUE,A0.DOUBLE_HIGH_VALUE,A0.DOUBLE_LOW_VALUE,A0.LAST_ANALYZED,A0.LONG_HIGH_VALUE,A0.LONG_LOW_VALUE,A0.MAX_COL_LEN,A0.NUM_DISTINCTS,A0.NUM_FALSES,A0.NUM_NULLS,A0.NUM_TRUES,A0.PARTITION_NAME,A0."TABLE_NAME",A0.CS_ID,A0.PARTITION_NAMEAS NUCORDER0 FROM PART_COL_STATS A0 WHERE A0."TABLE_NAME" = '$1' ANDA0.DB_NAME = '$2' AND A0.PARTITION_NAME = '$3' AND((((((A0."COLUMN_NAME" = '$4') OR (A0."COLUMN_NAME" ='$5')) OR (A0."COLUMN_NAME" = '$6')) OR (A0."COLUMN_NAME" ='$7')) OR (A0."COLUMN_NAME" = '$8')) OR (A0."COLUMN_NAME" ='$9')) ORDER BY NUCORDER0;
```

9. 执行**gs_guc reload -c log_min_duration_statement=100 -D /srv/BigData/dbdata_service/data/**开启SQL录制，发现**8**中语句执行时长700ms，而且因为有10000+分区，会触发执行10000+次命令。

10. 在SQL前加explain (analyze,verbose,timing,costs,buffers)分析执行计划，发现执行时需要全表扫描。

```
HIVEMETA=> explain (analyze,verbose,timing,costs,buffers) SELECT 'org.apache.hadoop.hive.metastore.model.MStorageDescriptor' AS NUCLEUS_TYPE,A0.INPUT_FORMAT,A0.IS_COMPRESSED,A0.IS_STOREDASSUBDIRECTORIES,A0.LOCATION,A0.NUM_BUCKETS,A0.OUTPUT_FORMAT,A0.SD_ID FROM SDS A0 WHERE A0.CD_ID = '463249' FETCH NEXT ROW ONLY;
                                                
```

问题现象

用户A在MRS节点后台上打开了Hive相关配置文件且未关闭，此时用户B在MRS Manager页面的“服务管理”中修改Hive配置项，保存配置并重启Hive服务，此时保存配置失败，并且Hive服务启动失败。

原因分析

由于用户B在MRS Manager页面修改配置时，配置文件被用户A在MRS节点后台打开，导致该配置文件不能被替换，最终导致Hive服务启动失败。

处理步骤

步骤1 用户需要首先手动关闭集群节点后台打开的Hive配置文件。

步骤2 在MRS Manager页面重新修改Hive的配置并保存配置。

步骤3 重启Hive服务。

----结束

13.46 Hive 分区数过多导致删除表失败

问题现象

Hive创建的二级分区表有两万多个分区，导致用户在执行**truncate table ${TableName},drop table ${TableName}**时失败。

原因分析

删除文件操作是单线程串行执行的，Hive分区数过多导致在元数据数据库会保存大量元数据信息，在执行删表语句时删除元数据就要用很长时间，最终在超时时间内删除不完，就会导致操作失败。

说明

超时时间可通过登录FusionInsight Manager，选择“集群 > 服务 > Hive > 配置 >全部配置 > MetaStore（角色） > 服务初始化”查看，“hive.metastore.client.socket.timeout”对应的值即为超时时间时长，在“描述”列可查看默认值。

处理步骤

步骤1 如果是内部表可以先通过**alter table ${TableName} set TBLPROPERTIES('EXTERNAL'='true')**来将内部表转成外部表，这样Hive删除的时候只删除元数据省去了删除HDFS数据的时间。

步骤2 如果要用相同的表名可以先将表结构用**show create table ${TableName}**来导出表结构，再用**ALTER TABLE ${TableName} RENAME TO ${new_table_name};**来将表重命名。这样就可以新建一个和原来一样表。

步骤3 执行**hdfs dfs -rm -r -f ${hdfs_path}**在HDFS上删除表数据。

步骤4 在Hive中用**alter table ${Table_Name} drop partition (${PartitionName}<’XXXX’, ${PartitionName}>’XXXX’);**删除分区(具体删除条件可灵活处理)，减少Temporyary文件数。

步骤5 删除分区少于一千个后，直接用**drop table ${TableName}**删掉表即可。

----结束

建议与总结

Hive分区虽然可以提高查询效率，但要避免分区不合理导致出现大量小文件的问题，要提前规划好分区策略。

13.47 Hive 执行 msck repair table 命令时报错

现象描述

Hive执行**msck repair table** *table_name*命令报错：

```
FAILED: Execution Error, return code 1 from org.apache.hadoop.hive.ql.exec.DDLTask (state=08S01,code=1)。
```

可能原因

查看HiveServer日志“/var/log/Bigdata/hive/hiveserver/hive.log”，发现目录名不符合分区格式。

```
2020-07-15 15:38:10,427 | WARN  | HiveServer2-Background-Pool: Thread-10905216 | Failed to run metacheck: | org.apache.hadoop.hive.ql.exec.DDLTask.msck(DDLTask.java:2023)
org.apache.hadoop.hive.ql.metadata.HiveException: Repair: Cannot add partition ads_marketing.t_marketing_telemarketing_order_list:dtime=2020-04-24 17%3A55%3A00 due to invalid characters in the name
    at org.apache.hadoop.hive.ql.exec.DDLTask.msck(DDLTask.java:1986) [hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hadoop.hive.ql.exec.DDLTask.execute(DDLTask.java:424) [hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hadoop.hive.ql.exec.Task.executeTask(Task.java:199) [hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hadoop.hive.ql.exec.TaskRunner.runSequential(TaskRunner.java:100) [hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hadoop.hive.ql.Driver.launchTask(Driver.java:2185) [hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hadoop.hive.ql.Driver.execute(Driver.java:1841) [hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hadoop.hive.ql.Driver.runInternal(Driver.java:1527) [hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hadoop.hive.ql.Driver.run(Driver.java:1238) [hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hadoop.hive.ql.Driver.run(Driver.java:1233) [hive-exec-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hive.service.cli.operation.SQLOperation.runQuery(SQLOperation.java:266) [hive-service-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hive.service.cli.operation.SQLOperation.access$800(SQLOperation.java:93) [hive-service-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at org.apache.hive.service.cli.operation.SQLOperation$BackgroundWork$1.run(SQLOperation.java:375) [hive-service-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at java.security.AccessController.doPrivileged(Native Method) ~[?:1.8.0_232]
    at javax.security.auth.Subject.doAs(Subject.java:422) [?:1.8.0_232]
    at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1840) [hadoop-common-2.8.3-mrs-1.9.0.jar:?]
    at org.apache.hive.service.cli.operation.SQLOperation$BackgroundWork.run(SQLOperation.java:393) [hive-service-2.3.3-mrs-1.9.0.jar:2.3.3-mrs-1.9.0]
    at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511) [?:1.8.0_232]
    at java.util.concurrent.FutureTask.run(FutureTask.java:266) [?:1.8.0_232]
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149) [?:1.8.0_232]
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624) [?:1.8.0_232]
    at java.lang.Thread.run(Thread.java:748) [?:1.8.0_232]
```

处理步骤

- 方法一：删除错误的文件或目录。
- 方法二：执行**set hive.msck.path.validation=skip;**，跳过无效的目录。

13.48 Hive 执行 insert into 命令报用户权限不足

用户问题

用户在Beeline命令行执行insert into命令报错：

```
INFO  : Concurrency mode is disabled, not creating a lock manager
Error: Error while processing statement: FAILED: Execution Error, return code 1 from org.apache.hadoop.hive.ql.exec.mr.MapRedTask. org.apache.hadoop.yarn.exceptions.YarnException: Failed to submit application_xxx to YARN : User xxx cannot submit application application_xxx to queue root.default. ACL check failed. (state=08S01,code=1)
```

原因分析

当前提交任务的用户不具备Yarn对应队列的权限。

处理步骤

步骤1 为当前用户配置Yarn队列的权限。

- 在Manager界面为当前用户绑定对应权限的角色，例如：default，详细操作请参考**修改用户信息**章节。
- 若Hive启用了Ranger鉴权，也可以参考**添加Yarn的Ranger访问权限策略**章节设置当前用户在指定Yarn队列提交任务的权限。

步骤2 权限配置生效后，重新执行命令。

----结束

13.49 在 Hive 中 drop 表后如何完全释放磁盘空间

用户问题

在Hive命令行执行**drop**表的操作后，通过命令**hdfs dfsadmin -report**查看磁盘空间，发现表没有删除。

原因分析

在Hive命令行执行**drop**表只删除了外部表的表结构，并没有删除该表存储在HDFS上的表数据。

处理步骤

步骤1 使用**root**用户登录安装客户端的节点，并认证用户。

cd *客户端安装目录*

source bigdata_env

kinit *组件业务用户*（未开启Kerberos认证的集群跳过此操作）

步骤2 执行以下命令删除存储在HDFS上的表。

hadoop fs -rm hdfs://hacluster/*表所在的具体路径*

----结束

13.50 JSON 表中有破损数据导致 Hive 查询异常

用户问题

在Hive客户端查询JSON数据，JSON表中有破损数据导致查询异常：

1. 在Hive客户端上使用默认开源的JSON序列化建表语句创建表：

```
create external table if not exists test (
name string
)
row format serde 'org.apache.hive.hcatalog.data.JsonSerDe'
```

stored as textfile

location '*hdfs://hacluster/user/hive/warehouse/database/table/jsondata***';**

2. 查询表出现异常。

```
INFO  : State: Compiling.
INFO  : Compiling command(queryId=omm_20221014092819_d396e069-8010-415c-a6e9-a7fa1191fb95): select * from test; Current sessionId=0139cc48-e14f-474e-9408-3920197d
156e
INFO  : hive.compile.auto.avoid.cbo=true
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Current sql is not contains insert syntax, not need record dest table flag
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:test.name, type:string, comment:null)], properties:null)
INFO  : Completed compiling command(queryId=omm_20221014092819_d396e069-8010-415c-a6e9-a7fa1191fb95); Time taken: 0.037 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : State: Executing.
INFO  : Executing command(queryId=omm_20221014092819_d396e069-8010-415c-a6e9-a7fa1191fb95): select * from test; Current sessionId=0139cc48-e14f-474e-9408-3920197d
156e
INFO  : Completed executing command(queryId=omm_20221014092819_d396e069-8010-415c-a6e9-a7fa1191fb95); Time taken: 0.0 seconds
INFO  : OK
INFO  : Concurrency mode is disabled, not creating a lock manager
Error: java.io.IOException: org.apache.hadoop.hive.serde2.SerDeException: org.codehaus.jackson.JsonParseException: Unexpected character ('1' (code 49)): was expec
ting a colon to separate field name and value
 at [Source: java.io.ByteArrayInputStream@1e5b799d; line: 1, column: 13] (state=,code=0)
```

说明

该章节内容适用于MRS 1.9.2之后版本。

原因分析

使用默认开源的JSON序列化语句创建Hive表无法将表中JSON破损数据过滤掉，导致查询异常。

处理步骤

步骤1 下载**json-serde-1.3.8-jar-with-dependencies.jar**到本地，并将该Jar上传到所有的HiveServer所在节点。

步骤2 以客户端安装用户，登录**步骤1**上传了Jar包并且安装了Hive和HDFS客户端的节点。

步骤3 执行以下命令认证用户。

cd *客户端安装目录*

source bigdata_env

kinit *具有Hive管理员权限的用户*（未开启Kerberos认证的集群跳过此操作）

步骤4 执行以下命令在HDFS上创建Jar包存放目录。

hdfs dfs -mkdir *Jar包存放在hdfs中的路径*

步骤5 执行以下命令将**步骤2**的Jar上传至HDFS中。

hdfs dfs -put *Jar包存放路径 Jar包存放在hdfs中的路径*

步骤6 执行以下命令让Hive在执行命令行任务时加载指定Jar包。

beeline

set role admin;（未开启Kerberos认证的集群跳过此操作）

add jar *Jar包存放在hdfs中的路径;*

步骤7 执行以下命令重新创建表：

create external table if not exists *test* **(**

name string

)

row format serde 'org.openx.data.jsonserde.JsonSerDe'

stored as textfile

location '*hdfs://hacluster/user/hive/warehouse/database/table/jsondata***';**

步骤8 执行以下命令修改表属性忽略破损的JSON数据：

ALTER TABLE *test* **SET SERDEPROPERTIES("ignore.malformed.json" = "true");**

设置该属性后，查询表时若表中有破损的JSON数据默认显示为“NULL”。

步骤9 执行以下命令查询表数据：

select * from *test***;**

数据查询成功则表示已忽略破损的JSON数据，其中，破损的JSON数据显示为“NULL”：

```
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Current sql is not contains insert syntax, not need record dest table flag
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:test.name, type:string, comment:null)], properties:null)
INFO  : Completed compiling command(queryId=omm_20221014093100_2ad45753-8e44-43f6-8931-7c8d0c95f420); Time taken: 0.053 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : State: Executing.
INFO  : Executing command(queryId=omm_20221014093100_2ad45753-8e44-43f6-8931-7c8d0c95f420): select * from test; Current sessionId=01
156e
INFO  : Completed executing command(queryId=omm_20221014093100_2ad45753-8e44-43f6-8931-7c8d0c95f420); Time taken: 0.0 seconds
INFO  : OK
INFO  : Concurrency mode is disabled, not creating a lock manager
+------------+
|     .name  |
+------------+
| 123        |
| NULL       |
+------------+
```

----结束

13.51 Hive 客户端执行 SQL 报错连接超时

现象描述

Hive客户端执行SQL失败，报错：Timed out waiting for a free available connection。

可能原因

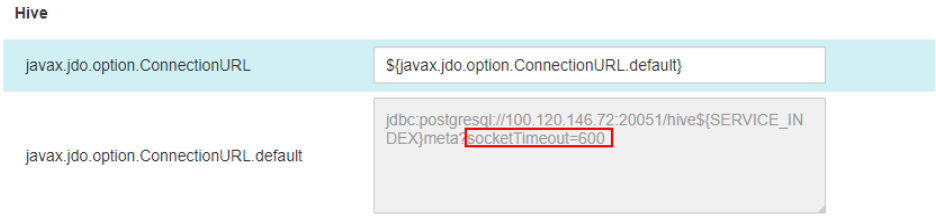
DBService连接较多，获取连接超时。

操作步骤

步骤1 客户端是否使用Spark-SQL客户端执行SQL。

- 是，检查连接的URL中超时参数，将其修改为600，执行**步骤7**。
- 否，执行**步骤2**。

步骤2 登录Manager页面，选择“集群 > 服务 > Hive > 配置 > 全部配置”，搜索“javax.jdo.option.ConnectionURL”，检查超时参数是否小于600。



说明

Hive、HiveServer、MetaStore、WebHCat中均有该参数，请确保它们的参数值一致。

- 是，执行**步骤3**。
- 否，执行**步骤7**。

步骤3 检查参数“javax.jdo.option.ConnectionURL”的值是否为“${javax.jdo.option.ConnectionURL.default}”。

- 是，执行**步骤4**。
- 否，修改URL中超时参数为600，单击“保存”，执行**步骤7**。

步骤4 单击“实例”，选择任意HiveServer实例，并使用**root**用户登录实例节点。

步骤5 打开配置文件“${BIGDATA_HOME}/FusionInsight_Current/*HiveServer/etc/hivemetastore-site.xml”，查找配置项“javax.jdo.option.ConnectionURL”，复制配置项值。

```
<property>
<name>javax.jdo.option.ConnectionURL</name>
<value>jdbc:postgresql://100.120.146.72:20051/hivemeta?socketTimeout=60</value>
</property>
<property>
```

步骤6 登录Manager页面，选择“集群 > 服务 > Hive > 配置 > 全部配置”，搜索“javax.jdo.option.ConnectionURL”，修改配置为**步骤5**中复制的URL，并修改超时参数为600，单击“保存”。

说明

Hive、HiveServer、MetaStore、WebHCat中均有该参数，请确保它们的参数值一致。

步骤7 选择“集群 > 服务 > Hive > 配置 > 全部配置”，搜索“maxConnectionsPerPartition”，检查是否小于100。

- 是，修改参数为100，单击“保存”，执行**步骤8**。
- 否，执行**步骤8**。

步骤8 若以上步骤有修改参数，选择“集群 > 服务 > Hive > 概览”，选择“更多 > 滚动重启服务”，未修改则无需执行此步骤。

----结束

13.52 WebHCat 健康状态异常导致启动失败

用户问题

WebHCat实例启动失败。

问题现象

在Manager页面上查看到WebHCat实例的健康状态为“故障”，并上报“ALM-12007 进程故障”告警，该告警的服务名称为“Hive”，实例名称为“WebHCat”。且重启Hive服务报错。

查看WebCat实例的日志“/var/log/Bigdata/hive/webhcat/webhcat.log”报错“Service not found in Kerberos database”和“Address already in use”。

处理步骤

步骤1 依次登录WebHCat实例所在节点检查“/etc/hosts”文件中的IP及主机名称映射关系是否正确。且“/etc/hostname”和“/etc/HOSTNAME”文件的WebHCat配置需与“/etc/hosts”保持一致，若不一致则需手动修改。

说明

WebHCat实例的IP地址及主机名称映射关系可登录FusionInsight Manager界面，选择“集群 > 服务 > Hive > 实例”查看。

步骤2 登录WebHCat实例所在节的任一节点，执行以下命令切换到**omm**用户。

su - omm

步骤3 执行以下命令查看是否存在WebHCat进程。

ps -ef|grep webhcat|grep -v grep

若存在，则需执行以下命令结束WebHCat进程：

kill -9 ${webhcat_pid}

步骤4 登录FusionInsight Manager，选择“集群 > 服务 > Hive > 实例”，勾选所有WebHCat实例，选择“更多 > 重启实例”，等待WebHCat重启成功即可。

----结束

13.53 mapred-default.xml 文件解析异常导致 WebHCat 启动失败

用户问题

MRS的Hive服务故障，重新启动后，Master2节点上的HiveServer和WebHCat进程启动失败，Master1节点进程正常。

原因分析

登录Master2节点，查看“/var/log/Bigdata/hive/hiveserver/hive.log”日志，发现HiveServer一直加载“/opt/Bigdata/*/*_HiveServer/etc/hive-site.xml”；查看HiveServer退出时的“/var/log/Bigdata/hive/hiveserver/hiveserver.out”日志，发现解析“mapred-default.xml”文件异常。

处理步骤

步骤1 登录Master2节点，使用以下命令查找“mapred-default.xml”所在路径：

find /opt/ -name 'mapred-default.xml'

查询到该配置文件在“/opt/Bigdata/*/*_WebHCat/etc/”目录下面，且该文件内容为空。

步骤2 登录到Master1节点，将“/opt/Bigdata/*/*_WebHCat/etc/mapred-default.xml”文件拷贝到Master2节点，并修改文件的属组为“omm:wheel”。

步骤3 登录Manager，重启异常的HiveServer和WebHCat实例。

----结束

13.54 MetaStore 动态分区数超出阈值导致 SQL 报错

现象描述

执行SparkSQL或HiveSQL命令时，系统报错如下：

```
Number of dynamic partitions created is 2001, which is more than 2000. To slove this try to set
hive.exec.max.dynamic.partitions to at least 2001
```

原因分析

Hive中会默认限制最大动态分区数，由“hive.exec.max.dynamic.partitions”参数控制（默认值为1000）。 如果超过此限制，Hive将不会创建新的动态分区。

处理步骤

- 调整上层业务，将动态分区数控制在“hive.exec.max.dynamic.partitions”参数值范围内。
- 执行**set hive.exec.max.dynamic.partitions =** *XXX*;命令调大“hive.exec.max.dynamic.partitions”参数值。

  SparkSQL中需要设置“spark.hadoop.hive.exec.max.dynamic.partitions”参数。

14 使用 Hue

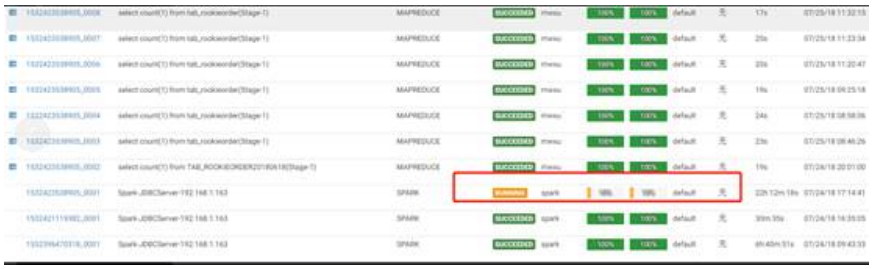
14.1 Hue 界面中查看到未知 Job 在运行

用户问题

用户查到Hue上有未知的Job在运行。

问题现象

MRS集群创建好后， Hue上查到有Job在运行 ，并且该Job并不是用户主动提交的。



原因分析

此Job为Spark服务启动之后，系统自身连接JDBC的一个默认任务，是常驻的。

处理步骤

非问题，无需处理。

14.2 使用 IE 浏览器在 Hue 中执行 HQL 失败

问题背景与现象

使用IE浏览器在Hue中访问Hive Editor并执行所有HQL失败，界面提示“There was an error with your query.”。

原因分析

IE浏览器存在功能问题，不支持在307重定向中处理含有form data的AJAX POST请求，建议更换兼容的浏览器。

解决办法

推荐使用Google Chrome浏览器21及以上版本。

14.3 Hue WebUI 界面访问失败

用户问题

访问Hue WebUI跳转到错误的页面。

问题现象

查看Hue WebUI报错如下：

```
503 Service Unavailable
The server is temporarily unable to service your requster due to maintenance downtime or capacity problems.Please try again later.
```

原因分析

Hue浮动IP故障。

处理步骤

步骤1 登录Manager页面，选择“集群 > 服务 > Hue > 配置”，搜索“HUE_FLOAT_IP”，查看Hue当前配置的浮动IP。

步骤2 连接到Hue主实例节点，检查主实例是否绑定浮动IP网卡。

ifconfig | grep -A2 HUE | grep *<浮动IP>*

- 是，结果不为空，主实例节点已绑定浮动IP，执行**步骤4**。
- 否，结果为空，主实例节点未绑定浮动IP，执行**步骤3**。

步骤3 连接到Hue主节点实例，绑定浮动IP网卡。本机网卡、浮动IP、掩码参考**ifconfig**中的本机IP对应的网卡。

ifconfig *<本机网卡>***:HUE** *<浮动IP>* **netmask** *<掩码>*

步骤4 连接到OMS主节点，尝试连接浮动IP，查看当前主机名是否为上述Hue主实例节点。

su - omm

ssh *<浮动ip>*

- 是，**浮动IP正常**，若Hue WebUI依然不能访问，请联系技术支持。
- 否，浮动IP被占用，执行**步骤5**。

步骤5 登录到集群内Hue主实例外的其他所有节点，执行以下命令，查看当前节点是否占用了Hue的浮动IP。

ifconfig | grep -B2 *<浮动ip>*

- 是，执行**ifconfig** *<网卡名称>* **down**解绑网卡，然后执行**步骤6**。
- 否，请联系技术支持。

步骤6 重复**步骤4**~**步骤5**，直到**步骤4**能够通过浮动IP连接到Hue主实例并成功访问Hue WebUI。

----结束

14.4 Hue 界面无法加载 HBase 表

用户问题

用户在Hue界面将hive数据导入hbase后，报检测不到hbase表的错误。

问题现象

Kerberos集群中，IAM子账户权限不足导致无法加载hbase表。

原因分析

IAM子账户权限不足。

处理步骤

MRS Manager界面操作：

步骤1 登录MRS Manager。

步骤2 选择“系统管理 > 用户管理”。

步骤3 在使用的用户所在行的单击“修改”。

步骤4 为用户添加supergroup组。

步骤5 单击“确定”完成修改操作。

----结束

FusionInsight Manager界面操作：

步骤1 登录FusionInsight Manager。

步骤2 选择“系统 > 权限 > 用户”。

步骤3 在使用的用户所在行单击“修改”。

步骤4 为用户添加supergroup组。

步骤5 单击“确定”完成修改操作。

----结束

建议与总结

如果是开启Kerberos认证的集群，页面出现No data available优先排查权限问题。

14.5 Hue 中的输入框输入中文会出现混乱

用户问题

Hue中的输入框输入中文会出现混乱。

问题现象

Hue的输入框中输入中文时，会出现混乱，第一次输入不了的情况，也即输入的中文会首先出现英文，中文并未输入，再次输入会带上之前的内容，示例如下：

原因分析

Hue对中文的处理存在混乱的情况，需要进行修正。

处理步骤

该方案仅适用于MRS 2.x及之前版本集群。

步骤1 登录MRS Manager页面，选择“服务管理 > Hue > 实例”，查询Hue实例所在的节点的IP。

步骤2 使用**root**用户远程连接工具登录Hue实例所在节点的机器，并执行如下命令切换到omm用户。

su - omm

步骤3 执行如下的命令，切换到如下的目录。

cd /opt/Bigdata/MRS_Current/1_*_Hue/install/hue/build/static/desktop/js/ace

步骤4 执行如下命令找到ace.js的hash文件。

ll ace.*.js

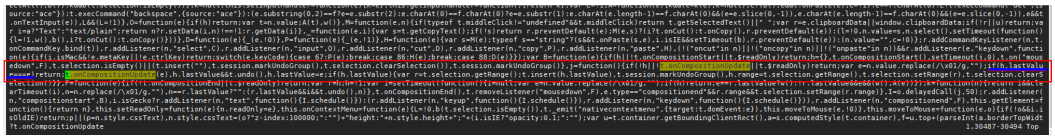
示例结果如下：

```
[omm@node-master1LIgg ace]$ ll ace.*.js
-rwxr-xr-x. 2 omm ficommon 347250 Jan 19 10:03 ace.18f1745e9832.js
[omm@node-master1LIgg ace]$
```

步骤5 执行如下命令打开并编辑上述的ace.*.js文件，也即修改hash文件。

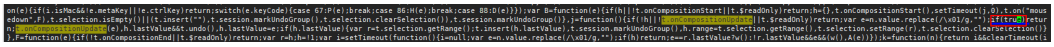
vim ace.18f1745e9832.js

步骤6 在ace.*.js文件中搜索“t.onCompositionUpdate”关键字。



步骤7 修改上图中红框中，蓝色下划线的内容，也即将蓝色下划线部分if (h.lastValue === e)修改为if (true) 。

修改后如下：



步骤8 保存修改。

Hue的所有实例节点均需要执行**步骤1**~**步骤8**进行修改。

步骤9 刷新页面，重新输入中文。

----结束

14.6 Hue 上执行 Impala SQL 查询中文报错

问题现象

Hue上执行Impala SQL，查询SQL结果中包含中文字符，出现报错“UnicodeDecodeError: 'utf-8' codec can't decode byte in position 0: unexpected end of data”。

原因分析

Hive中一个中文字符长度为“1”；而Impala中一个中文字符的长度为“3”，从而导致在Impala SQL中使用substr()，substring()，strleft()等函数截取中文字符时，不能把中文当做1个字符长度来处理，导致报编码问题。

处理步骤

步骤1 登录Impala客户端安装节点，执行以下命令。

cd *客户端安装目录*

source bigdata_env

步骤2 执行以下命令创建表。

impala-shell -d bigdata

步骤3 执行以下命令查询表数据。

select strleft(worker,3) from eier;

----结束

15 使用 Impala

15.1 连接 impala-shell 失败

用户问题

用户连接impala-shell失败。

问题现象

用户在“组件管理”页面修改任意组件的配置并重启服务后，连接impala-shell，会出现连接失败，报错no such file/directory。

```
[root@node-master1emdj etc]# pwd
/opt/Bigdata/MRS_2.1.0/1_7_KuduMaster/etc
[root@node-master1emdj etc]# impala-shell -i 192.168.0.73
shell-init: error retrieving current directory: getcwd: cannot access parent directories: No such file or directory
chdir: error retrieving current directory: getcwd: cannot access parent directories: No such file or directory
Traceback (most recent call last):
  File "/opt/client/Impala/impala/shell/impala_shell.py", line 38, in <module>
    from impala_client import (ImpalaClient, DisconnectedException, QueryStateException,
  File "/opt/client/Impala/impala/shell/lib/impala_client.py", line 20, in <module>
    import sasl
  File "build/bdist.linux-x86_64/egg/sasl/__init__.py", line 1, in <module>

  File "build/bdist.linux-x86_64/egg/sasl/saslwrapper.py", line 7, in <module>
  File "build/bdist.linux-x86_64/egg/_saslwrapper.py", line 7, in <module>
  File "build/bdist.linux-x86_64/egg/_saslwrapper.py", line 3, in __bootstrap__
  File "/usr/lib/python2.7/site-packages/setuptools-0.6c11-py2.7.egg/pkg_resources.py", line 2594, in <module>
    for comparator, version in req.specs:
  File "/usr/lib/python2.7/site-packages/setuptools-0.6c11-py2.7.egg/pkg_resources.py", line 425, in __init__

  File "/usr/lib/python2.7/site-packages/setuptools-0.6c11-py2.7.egg/pkg_resources.py", line 440, in add_entry
    `req`.  But, if there is an active distribution for the project and it
  File "/usr/lib/python2.7/site-packages/setuptools-0.6c11-py2.7.egg/pkg_resources.py", line 1688, in find_on_path
    return ()
  File "/usr/lib/python2.7/site-packages/setuptools-0.6c11-py2.7.egg/pkg_resources.py", line 1835, in _normalize_cached

  File "/usr/lib/python2.7/site-packages/setuptools-0.6c11-py2.7.egg/pkg_resources.py", line 1829, in normalize_path
    register_namespace_handler(object,null_ns_handler)
  File "/usr/lib64/python2.7/posixpath.py", line 368, in realpath
    return abspath(path)
  File "/usr/lib64/python2.7/posixpath.py", line 356, in abspath
    cwd = os.getcwd()
OSError: [Errno 2] No such file or directory
```

原因分析

修改服务配置并重启服务后，部分服务的目录结构会删除并重新创建，如服务的etc目录等。如果重启服务前所在的目录为etc或者其子目录，由于重启后目录重建，仍在原来目录执行impala-shell时会产生某些系统变量或者参数无法找到的情况，所以连接impala-shell连接失败。

处理步骤

任意切换到存在的目录，重新连接impala-shell即可。

15.2 创建 Kudu 表报错

用户问题

创建Kudu表报错。

问题现象

新建了集群，在创建表时，报错如下：

```
[Cloudera]ImpalaJDBCDriver ERROR processing query/statement. Error Code: 0, SQL state:
TStatus(statusCode:ERROR_STATUS, sqlState:HY000, errorMessage:AnalysisException: Table property
'kudu.master_addresses' is required when the impalad startup flag -kudu_master_hosts is not used.”
```

原因分析

用户未在impala sql中指定kudu.master_addresses地址导致报错。

处理步骤

参考**使用Impala操作Kudu**文档在创建Kudu表时指定“kudu.master_addresses”地址。

15.3 Impala 客户端安装 Python2

用户问题

运行Impala客户端会报类似如下错误信息：

```
[root@node-master1avIy ~]# impala-shell -i 192.168.128.49:21000
  File "/opt/client/Impala/impala/shell/impala_shell.py", line 1675
    except Exception, e:
                    ^
SyntaxError: invalid syntax
[root@node-master1avIy ~]#
```

原因分析

由于最新的MRS集群使用的是Euler 2.9及以上版本的操作系统，系统自带只Python3版本，而Impala客户端是基于Python2实现的，和Python3部分语法不兼容，运行Impala客户端会报错，所以需要手动安装Python2以解决客户端运行问题。

处理步骤

步骤1 使用**root**用户登录Impala所在节点，执行如下命令，确认当前系统上安装的Python版本：

python --version

```
[root@node-master2JgOY ~]# python --version
Python 3.7.4
```

步骤2 执行命令**yum install make**，查看yum是否可用。

- 如果yum install报如下错误，说明yum设置有问题，执行**步骤3**。

```
[root@node-master2JgOY ~]# yum install make
Error: There are no enabled repositories in "/etc/yum.repos.d", "/etc/yum/repos.d", "/etc/distro.repos.d".
```

- 如果没有报错，执行**步骤4**。

步骤3 执行命令**cat /etc/yum.repos.d/EulerOS-base.repo**，查看yum源和系统版本信息不匹配是否匹配，如果不匹配则修改，如下所示：

修改前：

```
[root@node-master1avIy ~]# cat /etc/yum.repos.d/EulerOS-base.repo
[base]
name=EulerOS-2.0SP2 base
baseurl=http://mirrors.my[redacted].com/euler/ict/site-euleros/euleros/repo/yum/2.2/os/x86_64/
enabled=1
gpgcheck=1
gpgkey=http://mirrors.my[redacted].com/euler/ict/site-euleros/euleros/repo/yum/2.2/os/RPM-GPG-KEY-EulerOS
[root@node-master1avIy ~]# uname -a
Linux node-master1avIy.mrs-mq7v.com 4.18.0-147.5.1.6.h541.eulerosv2r9.x86_64 #1 SMP Wed Aug 4 02:30:13 UTC
 x86_64 GNU/Linux
```

修改后：

```
[base]
name=EulerOS-2.0SP9 base
baseurl=http://mirrors.my[redacted].com/euler/ict/site-euleros/euleros/repo/yum/2.9/os/x86_64/
enabled=1
gpgcheck=1
gpgkey=http://mirrors.my[redacted].com/euler/ict/site-euleros/euleros/repo/yum/2.9/os/RPM-GPG-KEY-EulerOS
```

步骤4 执行如下命令，查看yum源上python2开头的软件。

yum list python2*

```
[root@node-master2JgOY ~]# yum list python2*
Last metadata expiration check: 0:02:36 ago on Thu 16 Dec 2021 10:05:52 AM CST.
Available Packages
python2.x86_64                                          2.7.16-16.eulerosv2r9
python2-debug.x86_64                                    2.7.16-16.eulerosv2r9
python2-devel.x86_64                                    2.7.16-16.eulerosv2r9
python2-help.noarch                                     2.7.16-16.eulerosv2r9
python2-pip.noarch                                      18.0-13.h2.eulerosv2r9
python2-setuptools.noarch                               40.4.3-4.h1.eulerosv2r9
python2-tkinter.x86_64                                  2.7.16-16.eulerosv2r9
python2-tools.x86_64                                    2.7.16-16.eulerosv2r9
```

步骤5 执行如下命令，安装Python2。

yum install python2

```
[root@node-master2JgOY ~]# yum install python2
Last metadata expiration check: 0:00:48 ago on Thu 16 Dec 2021 10:05:52 AM CST.
Error:
 Problem: problem with installed package python3-unversioned-command-3.7.4-7.h29.eulerosv2r9.x86_64
  - package python3-unversioned-command-3.7.4-7.h29.eulerosv2r9.x86_64 conflicts with python2 provided by python2-2.7.16-16.e
ulerosv2r9.x86_64
  - package python3-unversioned-command-3.7.4-7.h11.eulerosv2r9.x86_64 conflicts with python2 provided by python2-2.7.16-16.e
ulerosv2r9.x86_64
  - package python3-unversioned-command-3.7.4-7.h13.eulerosv2r9.x86_64 conflicts with python2 provided by python2-2.7.16-16.e
ulerosv2r9.x86_64
  - package python3-unversioned-command-3.7.4-7.h15.eulerosv2r9.x86_64 conflicts with python2 provided by python2-2.7.16-16.e
ulerosv2r9.x86_64
  - package python3-unversioned-command-3.7.4-7.h18.eulerosv2r9.x86_64 conflicts with python2 provided by python2-2.7.16-16.e
ulerosv2r9.x86_64
  - package python3-unversioned-command-3.7.4-7.h33.eulerosv2r9.x86_64 conflicts with python2 provided by python2-2.7.16-16.e
ulerosv2r9.x86_64
  - package python3-unversioned-command-3.7.4-7.h38.eulerosv2r9.x86_64 conflicts with python2 provided by python2-2.7.16-16.e
ulerosv2r9.x86_64
  - conflicting requests
(try to add '--allowerasing' to command line to replace conflicting packages or '--skip-broken' to skip uninstallable package
s or '--nobest' to use not only best candidate packages)
```

因为当前系统上已安装Python3，所有直接安装Python2会有上面的冲突提示。

可以选择--allowerasing或--skip-broken安装，例如：

yum install python2 --skip-broken

```
[root@node-master2JgOY ~]# yum install python2 --skip-broken
Last metadata expiration check: 0:34:08 ago on Thu 16 Dec 2021 10:05:52 AM CST.
Dependencies resolved.
==============================================================================================================================
 Package                             Architecture         Version                                Repository           Size
==============================================================================================================================
Installing:
 python2                             x86_64               2.7.16-16.eulerosv2r9                  base                6.4 M
Installing dependencies:
 libXft                              x86_64               2.3.2-13.eulerosv2r9                   base                 41 k
```

安装完成后，会自动将Python版本修改为Python2，如下所示：

```
Installed:
  libXft-2.3.2-13.eulerosv2r9.x86_64                          libXrender-0.9.10-10.eulerosv2r9.x86_64
  python2-2.7.16-16.eulerosv2r9.x86_64                        python2-debug-2.7.16-16.eulerosv2r9.x86_64
  python2-devel-2.7.16-16.eulerosv2r9.x86_64                  python2-help-2.7.16-16.eulerosv2r9.noarch
  python2-setuptools-40.4.3-4.h1.eulerosv2r9.noarch           python2-tkinter-2.7.16-16.eulerosv2r9.x86_64
  python2-tools-2.7.16-16.eulerosv2r9.x86_64                  python3-rpm-generators-9-1.eulerosv2r9.noarch
  tk-1:8.6.8-5.eulerosv2r9.x86_64

Complete!
[root@node-master2JgOY ~]# python --version
Python 2.7.16
```

如果Python2安装成功，但是显示的Python版本不对，需要执行以下命令手动给“/usr/bin/python2”创建软链接“/usr/bin/python”。

ln -s /usr/bin/python2 /usr/bin/python

步骤6 验证Impala客户端是否可用。

版权所有 © 华为云计算技术有限公司

```
[root@node-master1avIy ~]# impala-shell -i 192.168.128.49:21000
Starting Impala Shell without Kerberos authentication
Opened TCP connection to 192.168.128.49:21000
Connected to 192.168.128.49:21000
Server version: impalad version 3.4.0-RELEASE RELEASE (build eebadd34c1563cbf5825a4e4d361e7b3601f9827)
***********************************************************************************
Welcome to the Impala shell.
(Impala Shell v3.4.0-RELEASE (eebadd3) built on Thu Nov  4 11:29:54 CST 2021)

After running a query, type SUMMARY to see a summary of where time was spent.
***********************************************************************************
[192.168.128.49:21000] default> show databases;
Query: show databases
+------------------+--------------------------------------------+
| name             | comment                                    |
+------------------+--------------------------------------------+
| _impala_builtins | System database for Impala builtin functions |
| default          | Default Hive database                      |
+------------------+--------------------------------------------+
Fetched 2 row(s) in 0.16s
[192.168.128.49:21000] default>
```

----结束

16 使用 Kafka

16.1 运行 Kafka 客户端获取 Topic 时报错

问题现象

运行Kafka客户端获取Topic时报错，报错内容如下：

```
ERROR org.apache.kafka.common.errors.InvalidReplicationFactorException: Replication factor: 2 larger than available brokers: 0.
```

原因分析

由特殊字符导致获取的ZooKeeper地址变量错误。

处理步骤

步骤1 登录任意一个Master节点。

步骤2 执行**cat** *客户端安装目录*/**Kafka/kafka/config/server.properties |grep '^zookeeper.connect =**'命令，查看ZooKeeper地址的变量信息。

步骤3 重新运行Kafka获取Topic，其中从**步骤2**中获取的变量不要添加任何字符。

----结束

16.2 安全集群中使用 Python3.x 对接 Kafka

用户问题

通过Python3.x环境如何对接开启Kerberos认证的Kafka集群?

问题现象

客户想使用Python3.x的环境对接开启Kerberos认证的Kafka的集群。

处理步骤

步骤1 登录Master节点，执行如下命令，配置华为云欧拉镜像源。

wget http://mirrors.myhuaweicloud.com/repo/mirrors_source.sh && sh mirrors_source.sh

步骤2 执行如下命令，编译Python3.x。

yum groupinstall "Development tools" -y

yum -y install zlib zlib-devel

yum -y install bzip2 bzip2-devel

yum -y install ncurses ncurses-devel

yum -y install readline readline-devel

yum -y install openssl openssl-devel

yum -y install openssl-static

yum -y install xz lzma xz-devel

yum -y install sqlite sqlite-devel

yum -y install gdbm gdbm-devel

yum -y install tk tk-devel

yum -y install libffi libffi-devel

步骤3 编译成功后，执行如下命令，下载解压Python3.x的tgz包。

wget https://www.python.org/ftp/python/3.6.7/Python-3.6.7.tgz

tar -zxvf Python-3.6.7.tgz

cd Python-3.6.7

说明

Python3.x的tgz包也可以去Python官网下载。推荐使用Python-3.6.X版本，3.7版本无法使用rdd的take函数。

步骤4 执行如下命令，设置Python3.x的配置信息及编译安装，安装到/opt/Bigdata/python3目录下。

./configure --prefix=/opt/Bigdata/python3 --enable-shared CFLAGS=-fPIC

make && make install

说明

安装的目录可以自定义。

步骤5 执行如下命令，配置Python3.x变量。

echo "/opt/Bigdata/python3/lib" >> /etc/ld.so.conf

ldconfig

ln -s /opt/Bigdata/python3/bin/python3 /usr/bin/python3

ln -s /opt/Bigdata/python3/bin/pip3 /usr/bin/pip3

说明

配置变量的目录需要跟**步骤4**中安装的目录保持一致。

步骤6 配置成功后，执行如下命令，在Python3.x环境中安装Kafka组件。

cp /usr/include/gssapi/* /home/omm/kerberos/include/gssapi/

pip3 install kafka-python

pip3 install gssapi

步骤7 安装成功后，执行以下命令配置环境变量。

source *客户端安装目录*/**bigdata_env**

步骤8 执行以下命令认证当前用户。

kinit *Kafka用户*

说明

kinit命令使用的用户为登录Manager的用户，此用户需要具有Kafka用户组相关权限。

步骤9 执行Python3.x脚本样例。

脚本样例：

```
producer：
from kafka import KafkaProducer
producer = KafkaProducer(bootstrap_servers=["broker_ip:21007"],
security_protocol="SASL_PLAINTEXT",
sasl_mechanism="GSSAPI",
sasl_kerberos_service_name="kafka",
sasl_kerberos_domain_name="hadoop.hadoop.com")
for _ in range(100):
response = producer.send("test-topic", b"testmessage")
result = response.get(timeout=50)
print(result)

consumer：
from kafka import KafkaConsumer
consumer = KafkaConsumer("test-topic",
bootstrap_servers=["broker_ip:21007"],
group_id="test-group",
enable_auto_commit="true",
security_protocol="SASL_PLAINTEXT",
sasl_mechanism="GSSAPI",
sasl_kerberos_service_name="kafka",
sasl_kerberos_domain_name="hadoop.hadoop.com")
for message in consumer:
print(message)
```

----结束

16.3 Flume 正常连接 Kafka 后发送消息失败

问题现象

使用MRS版本安装集群，主要安装ZooKeeper、Flume、Kafka。

在使用Flume向Kafka发送数据功能时，发现Flume发送数据到Kafka失败。

可能原因

1. Kafka服务异常。
2. Flume连接Kafka地址错误，导致发送失败。
3. Flume发送超过Kafka大小限制的消息，导致发送失败。

原因分析

Flume发送数据到Kafka失败，可能原因是Flume侧问题或者Kafka侧问题。

1. Manager界面查看当前Kafka状态及监控指标。
   - MRS Manager界面操作：登录MRS Manager，选择 “服务管理 > Kafka”，查看当前Kafka状态，发现状态为良好，且监控指标内容显示正确。
   - FusionInsight Manager界面操作：登录FusionInsight Manager，选择“集群 > 服务 > Kafka”，查看当前Kafka状态，发现状态为良好，且监控指标内容显示正确。
2. 查看Flume日志，发现打印MessageSizeTooLargeException异常信息，如下所示：
   ```
   2016-02-26 14:55:19,126 | WARN  | [SinkRunner-PollingRunner-DefaultSinkProcessor] |  Produce request with correlation id 349829 failed due to [LOG,7]: kafka.common.MessageSizeTooLargeException  | kafka.utils.Logging$class.warn(Logging.scala:83)
   ```
   通过异常信息，发现当前Flume向Kafka写入的数据超过了Kafka服务端定义的消息的最大值。
3. 通过Manager查看Kafka服务端定义的消息的最大值。
   - MRS Manager界面操作入口：登录MRS Manager，依次选择 “服务管理 > Kafka> 配置”。
   - FusionInsight Manager界面操作入口：登录FusionInsight Manager，选择“集群 > 服务 > Kafka > 配置”。

   进入Kafka配置页面，参数类别选择全部配置，显示所有Kafka相关配置，在“搜索”中输入message.max.bytes进行检索。

   MRS中Kafka服务端默认可以接收的消息最大为1000012 bytes =977 KB。

解决办法

与用户确认，当前Flume发送数据确实存在超过1M的消息。因此，为了确保当前这些消息能够写入Kafka，需要调整Kafka服务端相关参数。

步骤1 修改message.max.bytes，使得message.max.bytes的值大于当前业务中消息最大值，使得Kafka服务端可以接收全部消息。

步骤2 修改replica.fetch.max.bytes，使得**replica.fetch.max.bytes >= message.max.bytes**，使得不同Broker上的Partition的Replica可以同步到全部消息。

- MRS Manager界面操作入口：登录MRS Manager，依次选择 “服务管理 > Kafka> 配置”。
- FusionInsight Manager界面操作入口：登录FusionInsight Manager，选择“集群 > 服务 > Kafka > 配置”。

进入Kafka配置页面，参数类别选择全部配置， 显示所有Kafka相关配置，在“搜索”中输入replica.fetch.max.bytes进行检索。

步骤3 单击“保存”，并重启Kafka服务，使得Kafka相关配置生效。

步骤4 修改消费者业务应用中fetch.message.max.bytes，使得fetch.message.max.bytes >= message.max.bytes，确保消费者可以消费到全部消息。

----结束

16.4 Producer 发送数据失败，报错“NullPointerException”

问题现象

使用MRS安装集群，主要安装ZooKeeper、Kafka。

在使用Producer向Kafka发送数据功能时，发现客户端抛出NullPointerException。

可能原因

1. Kafka服务异常。
2. 客户端Producer侧jaas文件和keytab文件错误。

原因分析

Producer发送数据到Kafka失败，可能原因客户端Producer侧问题或者Kafka侧问题。

1. 通过Manager页面查看Kafka服务状态及监控指标。
   - MRS Manager界面操作：登录MRS Manager，依次选择“服务管理 > Kafka”，查看当前Kafka状态，发现状态为良好，且监控指标内容显示正确。
   - FusionInsight Manager界面操作：登录FusionInsight Manager，选择“集群 > 服务 > Kafka”，查看当前Kafka状态，发现状态为良好，且监控指标内容显示正确。
2. 查看Producer客户端日志，发现打印NullPointerException异常信息，如图16-1所示。

图 16-1 Producer 客户端日志

```
[2016-12-06 02:04:05,906]-[schedule-C50D0717-4643-4D4E-9D6E-B940E4BD7159]-[kafka-producer-network-thread |
SZX1000161910-10.21.219.222-bigdata-producer-5]-[1005]-[org.apache.kafka.clients.producer.internals.Sender.run
thread:
java.lang.NullPointerException
    at org.apache.kafka.common.network.Selector.pollSelectionKeys(Selector.java:302)
    at org.apache.kafka.common.network.Selector.poll(Selector.java:283)
    at org.apache.kafka.clients.NetworkClient.poll(NetworkClient.java:260)
    at org.apache.kafka.clients.producer.internals.Sender.run(Sender.java:229)
    at org.apache.kafka.clients.producer.internals.Sender.run(Sender.java:134)
    at java.lang.Thread.run(Thread.java:745)
[2016-12-06 02:04:05,921]-[schedule-C50D0717-4643-4D4E-9D6E-B940E4BD7159]-[kafka-producer-network-thread |
SZX1000161910-10.21.219.222-bigdata-producer-3]-[1005]-[org.apache.kafka.clients.producer.internals.Sender.run
thread:
java.lang.NullPointerException
    at org.apache.kafka.common.network.Selector.pollSelectionKeys(Selector.java:302)
    at org.apache.kafka.common.network.Selector.poll(Selector.java:283)
    at org.apache.kafka.clients.NetworkClient.poll(NetworkClient.java:260)
    at org.apache.kafka.clients.producer.internals.Sender.run(Sender.java:229)
    at org.apache.kafka.clients.producer.internals.Sender.run(Sender.java:134)
    at java.lang.Thread.run(Thread.java:745)
```

或者日志中只有异常信息没有堆栈信息（只有NullPointerException无堆栈信息，出现这个问题是jdk的自我保护，相同堆栈打印太多，就会触发这个保护开关，后续不再打印堆栈），如图16-2所示。

图 16-2 异常信息

```
[2016-11-23 04:06:53,973] [kafka-producer-network-thread | producer-1] [ERROR] [org.apache.kafka.clients.producer.internals.Sender] (run:130)- Uncaught error in kafka producer I/O thread:
java.lang.NullPointerException
[2016-11-23 04:06:53,973] [kafka-producer-network-thread | producer-1] [ERROR] [org.apache.kafka.clients.producer.internals.Sender] (run:130)- Uncaught error in kafka producer I/O thread:
java.lang.NullPointerException
[2016-11-23 04:06:53,973] [kafka-producer-network-thread | producer-1] [ERROR] [org.apache.kafka.clients.producer.internals.Sender] (run:130)- Uncaught error in kafka producer I/O thread:
java.lang.NullPointerException
[2016-11-23 04:06:53,973] [kafka-producer-network-thread | producer-1] [ERROR] [org.apache.kafka.clients.producer.internals.Sender] (run:130)- Uncaught error in kafka producer I/O thread:
java.lang.NullPointerException
```

3. 查看Producer客户端日志，发现打印Failed to configure SaslClientAuthenticator异常信息，如图16-3所示。

图 16-3 异常日志信息

```
Caused by: org.apache.kafka.common.KafkaException: Failed to configure SaslClientAuthenticator
at org.apache.kafka.common.security.authenticator.SaslClientAuthenticator.configure(SaslClientAuthenticator.java:96)
at org.apache.kafka.common.network.SaslChannelBuilder.buildChannel(SaslChannelBuilder.java:89)
... 9 more
Caused by: org.apache.kafka.common.KafkaException: Failed to create SaslClient
at org.apache.kafka.common.security.authenticator.SaslClientAuthenticator.createSaslClient(SaslClientAuthenticator.java:112)
at org.apache.kafka.common.security.authenticator.SaslClientAuthenticator.configure(SaslClientAuthenticator.java:94)
... 10 more
Caused by: javax.security.sasl.SaslException: PLAIN: authorization ID and password must be specified
at com.sun.security.sasl.PlainClient.<init>(PlainClient.java:58)
at com.sun.security.sasl.ClientFactoryImpl.createSaslClient(ClientFactoryImpl.java:97)
at javax.security.sasl.Sasl.createSaslClient(Sasl.java:384)
at com.ibm.messagehub.login.MessageHubSaslClientFactory.createSaslClient(MessageHubSaslClientFactory.java:77)
at javax.security.sasl.Sasl.createSaslClient(Sasl.java:384)
at org.apache.kafka.common.security.authenticator.SaslClientAuthenticator$1.run(SaslClientAuthenticator.java:107)
at org.apache.kafka.common.security.authenticator.SaslClientAuthenticator$1.run(SaslClientAuthenticator.java:102)
at java.security.AccessController.doPrivileged(Native Method)
at javax.security.auth.Subject.doAs(Subject.java:422)
at org.apache.kafka.common.security.authenticator.SaslClientAuthenticator.createSaslClient(SaslClientAuthenticator.java:102)
... 11 more
```

4. 认证失败导致创建KafkaChannel失败，导致通过channel(key)方法获取的KafkaChannel为空，以至于疯狂打印NullPointerException，上述日志可以发现，认证失败的原因是用户密码不正确，密码不正确的原因可能是用户名不匹配导致。
5. 检查Jaas文件和Keytab文件，发现Jaas文件中配置使用的pricipal为stream。

图 16-4 检查 Jaas 文件

```
KafkaClient {
com.sun.security.auth.module.Krb5LoginModule required
debug=false
keyTab="/opt/client/user.keytab"
useTicketCache=false
storeKey=true
principal="stream@HADOOP.COM"
useKeyTab=true;
};
```

查看user.keytab文件，发现principal为zmk_kafka。

图 16-5 查看 user.keytab 文件

```
[root@8-5-148-6 client]# klist -kt user.keytab
Keytab name: FILE:user.keytab
KVNO Timestamp         Principal
---- ----------------- --------------------------------------------------------
   1 12/19/16 16:28:17 zmk_kafka@HADOOP.COM
   1 12/19/16 16:28:17 zmk_kafka@HADOOP.COM
```

发现jaas文件和user.keytab文件中principal不对应。

该情况是由于应用程序自动定时更新Jaas文件，但是有两个不同的进程在进行更新，一个进程写入正确的Principal而另一个却写入了错误的Principal，以至于程序时而正常，时而异常。

解决办法

步骤1 修改Jaas文件，确保使用的Principal在Keytab文件中存在。

----结束

16.5 Producer 发送数据失败，报错“TOPIC_AUTHORIZATION_FAILED”

问题现象

使用MRS安装集群，主要安装ZooKeeper、Kafka。

在使用Producer向Kafka发送数据功能时，发现客户端报错：TOPIC_AUTHORIZATION_FAILED。

可能原因

1. Kafka服务异常。
2. 客户端Producer侧采用非安全访问，服务端配置禁止访问。
3. 客户端Producer侧采用非安全访问，Kafka Topic设置ACL。

原因分析

Producer发送数据到Kafka失败，可能原因客户端Producer侧问题或者Kafka侧问题。

1. 查看Kafka服务状态：
   - MRS Manager界面操作：登录MRS Manager，依次选择“服务管理 > Kafka”，查看当前Kafka状态，发现状态为良好，且监控指标内容显示正确。
   - FusionInsight Manager界面操作：登录FusionInsight Manager，选择“集群 > 服务 > Kafka”，查看当前Kafka状态，发现状态为良好，且监控指标内容显示正确。
2. 查看Producer客户端日志，发现打印TOPIC_AUTHORIZATION_FAILED异常信息。

```
[root@10-10-144-2 client]# kafka-console-producer.sh --broker-list 10.5.144.2:9092 --topic test
1
[2017-01-24 16:58:36,671] WARN Error while fetching metadata with correlation id 0 :
{test=TOPIC_AUTHORIZATION_FAILED} (org.apache.kafka.clients.NetworkClient)
[2017-01-24 16:58:36,672] ERROR Error when sending message to topic test with key: null, value: 1
bytes with error: Not authorized to access topics: [test]
(org.apache.kafka.clients.producer.internals.ErrorLoggingCallback)
```

   Producer采用9092端口来访问Kafka，9092为非安全端口。
3. 通过Manager页面，查看当前Kafka集群配置，发现未设置自定义配置“allow.everyone.if.no.acl.found”=“false”。
   - MRS Manager界面操作入口：登录MRS Manager，依次选择“服务管理 > Kafka> 配置”。

– FusionInsight Manager界面操作入口：登录FusionInsight Manager，选择“集群 > 服务 > Kafka > 配置”。

4. 当ACL设置为false不允许采用9092来进行访问。
5. 查看Producer客户端日志，发现打印TOPIC_AUTHORIZATION_FAILED异常信息。

```
[root@10-10-144-2 client]# kafka-console-producer.sh --broker-list 10.5.144.2:21005 --topic test_acl
1
[2017-01-25 11:09:40,012] WARN Error while fetching metadata with correlation id 0 :
{test_acl=TOPIC_AUTHORIZATION_FAILED} (org.apache.kafka.clients.NetworkClient)
[2017-01-25 11:09:40,013] ERROR Error when sending message to topic test_acl with key: null, value:
1 bytes with error: Not authorized to access topics: [test_acl]
(org.apache.kafka.clients.producer.internals.ErrorLoggingCallback)
[2017-01-25 11:14:40,010] WARN Error while fetching metadata with correlation id 1 :
{test_acl=TOPIC_AUTHORIZATION_FAILED} (org.apache.kafka.clients.NetworkClient)
```

Producer采用21005端口来访问Kafka，21005为非安全端口。

6. 通过客户端命令查看topic的ACL权限设置信息。

```
[root@10-10-144-2 client]# kafka-acls.sh --authorizer-properties zookeeper.connect=10.5.144.2:24002/
kafka --list --topic topic_acl
Current ACLs for resource `Topic:topic_acl`:
 User:test_user has Allow permission for operations: Describe from hosts: *
User:test_user has Allow permission for operations: Write from hosts: *
```

Topic设置ACL，则不允许采用9092来访问。

7. 查看Producer客户端日志，发现打印TOPIC_AUTHORIZATION_FAILED异常信息。

```
[root@10-10-144-2 client]# kafka-console-producer.sh --broker-list 10.5.144.2:21007 --topic topic_acl
--producer.config /opt/client/Kafka/kafka/config/producer.properties
1
[2017-01-25 12:43:58,506] WARN Error while fetching metadata with correlation id 0 :
{topic_acl=TOPIC_AUTHORIZATION_FAILED} (org.apache.kafka.clients.NetworkClient)
[2017-01-25 12:43:58,507] ERROR Error when sending message to topic topic_acl with key: null,
value: 1 bytes with error: Not authorized to access topics: [topic_acl]
(org.apache.kafka.clients.producer.internals.ErrorLoggingCallback)
```

Producer采用21007端口来访问Kafka。

8. 通过客户端命令klist查询当前认证用户。

```
[root@10-10-144-2 client]# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: test@HADOOP.COM

Valid starting     Expires            Service principal
01/25/17 11:06:48  01/26/17 11:06:45  krbtgt/HADOOP.COM@HADOOP.COM
```

当前认证用户为**test**。

9. 通过客户端命令查看Topic的ACL权限设置信息。

```
[root@10-10-144-2 client]# kafka-acls.sh --authorizer-properties zookeeper.connect=10.5.144.2:2181/
kafka --list --topic topic_acl
Current ACLs for resource `Topic:topic_acl`:
 User:test_user has Allow permission for operations: Describe from hosts: *
User:test_user has Allow permission for operations: Write from hosts: *
```

Topic设置ACL，用户test_user具有producer权限。test无权限进行producer操作。

解决方法参考**步骤2**。

10. 通过SSH登录Kafka Broker：

通过**cd /var/log/Bigdata/kafka/broker**命令进入日志目录。

查看kafka-authorizer.log发现如下日志提示用户不属于kafka或者kafkaadmin组。

```
2017-01-25 13:26:33,648 | INFO  | [kafka-request-handler-0] | The principal is test, belongs to Group :
[hadoop, ficommon] | kafka.authorizer.logger (SimpleAclAuthorizer.scala:169)
```

```
2017-01-25 13:26:33,648 | WARN  | [kafka-request-handler-0] | The user is not belongs to kafka or kafkaadmin group, authorize failed! | kafka.authorizer.logger (SimpleAclAuthorizer.scala:170)
```

解决方法参考**步骤3**。

解决办法

步骤1 配置自定义配置“allow.everyone.if.no.acl.found”参数为“true”，重启Kafka服务。

步骤2 采用具有权限用户登录。

例如：

kinit test_user

或者赋予用户相关权限。

说明

需要使用Kafka管理员用户（属于kafkaadmin组）操作。

例如：

kafka-acls.sh --authorizer-properties zookeeper.connect=10.5.144.2:2181/kafka --topic topic_acl --producer --add --allow-principal User:test

```
[root@10-10-144-2 client]# kafka-acls.sh --authorizer-properties zookeeper.connect=8.5.144.2:2181/kafka --list --topic topic_acl
Current ACLs for resource `Topic:topic_acl`:
 User:test_user has Allow permission for operations: Describe from hosts: *
 User:test_user has Allow permission for operations: Write from hosts: *
  User:test has Allow permission for operations: Describe from hosts: *
  User:test has Allow permission for operations: Write from hosts: *
```

步骤3 用户加入**Kafka**组或者**Kafkaadmin**组。

----结束

16.6 Producer 偶现发送数据失败，日志提示“Too many open files in system”

问题背景与现象

在使用Producer向Kafka发送数据功能时，发现客户端发送失败。

图 16-6 Producer 发送数据失败

```
2018-05-08 23:29:16,856 | ERROR | [SinkRunner-PollingRunner-DefaultSinkProcessor] | Failed to publish events  | org.apache.flume.sink.
2018-05-08 23:29:16,856 | ERROR | [SinkRunner-PollingRunner-DefaultSinkProcessor] | Unable to deliver event. Exception follows.  | org
2018-05-08 23:29:16,986 | ERROR | [SinkRunner-PollingRunner-DefaultSinkProcessor] | Failed to publish events  | org.apache.flume.sink.
2018-05-08 23:29:16,987 | ERROR | [SinkRunner-PollingRunner-DefaultSinkProcessor] | Unable to deliver event. Exception follows.  | org
```

可能原因

1. Kafka服务异常。
2. 网络异常。
3. Kafka Topic异常。

原因分析

1. 查看Kafka服务状态：
   - MRS Manager界面操作：登录MRS Manager，依次选择“服务管理 > Kafka”，查看当前Kafka状态，发现状态为良好，且监控指标内容显示正确。
   - FusionInsight Manager界面操作：登录FusionInsight Manager，选择“集群 > 服务 > Kafka”，查看当前Kafka状态，发现状态为良好，且监控指标内容显示正确。
2. 查看SparkStreaming日志中提示错误的Topic信息。

   执行Kafka相关命令，获取Topic分布信息和副本同步信息，观察返回结果。

   kafka-topics.sh --describe --zookeeper <zk_host:port/chroot>

   如图16-7所示，发现对应Topic状态正常。所有Partition均存在正常Leader信息。

   图 16-7 Topic 状态

   ```
   Topic: STK6     Partition: 36   Leader: 3       Replicas: 3,5   Isr: 3,5
   Topic: STK6     Partition: 37   Leader: 4       Replicas: 4,6   Isr: 4,6
   Topic: STK6     Partition: 38   Leader: 5       Replicas: 5,7   Isr: 5,7
   Topic: STK6     Partition: 39   Leader: 6       Replicas: 6,8   Isr: 6,8
   Topic: STK6     Partition: 40   Leader: 7       Replicas: 7,9   Isr: 7,9
   Topic: STK6     Partition: 41   Leader: 8       Replicas: 8,1   Isr: 8,1
   Topic: STK6     Partition: 42   Leader: 9       Replicas: 9,2   Isr: 9,2
   Topic: STK6     Partition: 43   Leader: 1       Replicas: 1,3   Isr: 3,1
   Topic: STK6     Partition: 44   Leader: 2       Replicas: 2,4   Isr: 2,4
   Topic: STK6     Partition: 45   Leader: 3       Replicas: 3,6   Isr: 3,6
   Topic: STK6     Partition: 46   Leader: 4       Replicas: 4,7   Isr: 4,7
   Topic: STK6     Partition: 47   Leader: 5       Replicas: 5,8   Isr: 5
   Topic: STK6     Partition: 48   Leader: 6       Replicas: 6,9   Isr: 6,9
   Topic: STK6     Partition: 49   Leader: 7       Replicas: 7,1   Isr: 7,1
   Topic: STK6     Partition: 50   Leader: 8       Replicas: 8,2   Isr: 2,8
   Topic: STK6     Partition: 51   Leader: 9       Replicas: 9,3   Isr: 9,3
   Topic: STK6     Partition: 52   Leader: 1       Replicas: 1,4   Isr: 4,1
   Topic: STK6     Partition: 53   Leader: 2       Replicas: 2,5   Isr: 5,2
   Topic: STK6     Partition: 54   Leader: 3       Replicas: 3,7   Isr: 3,7
   Topic: STK6     Partition: 55   Leader: 4       Replicas: 4,8   Isr: 4,8
   Topic: STK6     Partition: 56   Leader: 5       Replicas: 5,9   Isr: 5,9
   Topic: STK6     Partition: 57   Leader: 6       Replicas: 6,1   Isr: 6,1
   Topic: STK6     Partition: 58   Leader: 7       Replicas: 7,2   Isr: 2,7
   ```

3. 通过telnet命令，查看是否可以连接Kafka。

   telnet *Kafka业务IP Kafka业务port*

   如果无法telnet成功，请检查网络安全组与ACL。
4. 通过SSH登录Kafka Broker。

   通过**cd /var/log/Bigdata/kafka/broker**命令进入日志目录。

   查看server.log发现如下日志抛出java.io.IOException: Too many open files in system。

   图 16-8 日志异常

   ```
   2018-05-08 23:05:00,061 | ERROR | [kafka-socket-acceptor-PLAINTEXT-21005] | Error while accepting connection | kafka.network.Accept
   java.io.IOException: Too many open files in system
       at sun.nio.ch.ServerSocketChannelImpl.accept0(Native Method)
       at sun.nio.ch.ServerSocketChannelImpl.accept(ServerSocketChannelImpl.java:422)
       at sun.nio.ch.ServerSocketChannelImpl.accept(ServerSocketChannelImpl.java:250)
       at kafka.network.Acceptor.accept(SocketServer.scala:336)
   ```

5. 通过**lsof**命令查看当前节点Kafka进程句柄使用情况，发现占用的句柄数达到了47万。

图 16-9 句柄数

```
omm@lf2-bi-sparkstream-71-24-8:/var/log/Bigdata/kafka/broker> jps
24338 Kafka
14630 MetricAgentMain
30713 NodeAgent
46973 Jps
omm@lf2-bi-sparkstream-71-24-8:/var/log/Bigdata/kafka/broker> lsof -p 24383|wc
0
omm@lf2-bi-sparkstream-71-24-8:/var/log/Bigdata/kafka/broker> lsof -p 24338|wc
473282
```

6. 排查业务代码，不停地创建新的Producer对象，未正常关闭。

解决办法

步骤1 停止当前应用，保证服务端句柄不再疯狂增加影响服务正常运行。

步骤2 优化应用代码，解决句柄泄露问题。

建议：全局尽量使用一个Producer对象，在使用完成之后主动调用close接口进行句柄关闭。

----结束

16.7 Consumer 初始化成功但是无法从 Kafka 中获取指定 Topic 消息

问题背景与现象

使用MRS安装集群，主要安装ZooKeeper、Flume、Kafka、Storm、Spark。

使用Storm、Spark、Flume或者自己编写consumer代码来消费Kafka中指定Topic的消息时，发现消费不到任何数据。

可能原因

1. Kafka服务异常。
2. Consumer中ZooKeeper相关连接地址配置错误，导致无法消费。
3. Consumer发生ConsumerRebalanceFailedException异常，导致无法消费。
4. Consumer发生网络导致的ClosedChannelException异常，导致无法消费。

原因分析

Storm、Spark、Flume或者自定义Consumer代码可以都称为Consumer。

1. 查看Kafka服务状态：
   - MRS Manager界面操作：登录MRS Manager，依次选择“服务管理 > Kafka”，查看当前Kafka状态，发现状态为良好，且监控指标内容显示正确。

– FusionInsight Manager界面操作：登录FusionInsight Manager，选择“集群 > 服务 > Kafka”，查看当前Kafka状态，发现状态为良好，且监控指标内容显示正确。

2. 通过Kafka Client，判断是否可以正常消费数据。

   假设客户端已经安装在/opt/client目录，test为需要消费的Topic名称，192.168.234.231为ZooKeeper的IP地址。

```
cd /opt/client
source bigdata_env
kinit admin
kafka-topics.sh --zookeeper 192.168.234.231:2181/kafka --describe --topic testkafka-console-consumer.sh --topic test --zookeeper 192.168.234.231:2181/kafka --from-beginning
```

   当可以消费到数据时，表示集群服务正常。

3. 查看Consumer相关配置，发现ZooKeeper连接地址错误。

   – Flume

```
server.sources.Source02.type=org.apache.flume.source.kafka.KafkaSource

server.sources.Source02.zookeeperConnect=192.168.234.231:2181
server.sources.Source02.topic = test
server.sources.Source02.groupId = test_01
```

   – Spark

```
val zkQuorum = "192.168.234.231:2181"
```

   – Storm

```
BrokerHosts brokerHosts = new ZKHosts("192.168.234.231:2181");
```

   – Consumer API

```
zookeeper.connect="192.168.234.231:2181"
```

   MRS中Kafka在ZooKeeper存储的ZNode是以/kafka为根路径，有别于开源。Kafka对应的ZooKeeper连接配置为192.168.234.231:2181/kafka。

   Consumer中配置为ZooKeeper连接配置为192.168.234.231:2181，导致无法正确获取Kafka中Topic相关信息。

   解决方法参考**步骤1**。

4. 查看Consumer相关日志，发现打印ConsumerRebalanceFailedException异常。

```
2016-02-03 15:55:32,557 | ERROR | [ZkClient-EventThread-75- 192.168.234.231:2181/kafka] |  Error handling event ZkEvent[New session event sent to kafka.consumer.ZookeeperConsumerConnector$ZKSessionExpireListener@34b41dfe]  | org.I0Itec.zkclient.ZkEventThread.run(ZkEventThread.java:77)
kafka.common.ConsumerRebalanceFailedException: pc-zjqbetl86-1454482884879-2ec95ed3 can't rebalance after 4 retries
at kafka.consumer.ZookeeperConsumerConnector$ZKRebalancerListener.syncedRebalance(ZookeeperConsumerConnector.scala:633)
at kafka.consumer.ZookeeperConsumerConnector$ZKSessionExpireListener.handleNewSession(ZookeeperConsumerConnector.scala:487)
at org.I0Itec.zkclient.ZkClient$4.run(ZkClient.java:472)
at org.I0Itec.zkclient.ZkEventThread.run(ZkEventThread.java:71)
```

   通过异常信息，发现当前Consumer没有在指定的重试次数内完成Rebalance，使得Consumer没有被分配Kafka Topic-Partition，则无法消费消息。

   解决方法参考**步骤3**。

5. 查看Consumer相关日志，发现打印Fetching topic metadata with correlation id 0 for topics [Set(test)] from broker [id:26,host:192-168-234-231,port:9092] failed错误信息和ClosedChannelException异常。

```
[2016-03-04 03:33:53,047] INFO Fetching metadata from broker id:26,host: 192-168-234-231,port:9092 with correlation id 0 for 1 topic(s) Set(test) (kafka.client.ClientUtils$)
[2016-03-04 03:33:55,614] INFO Connected to 192-168-234-231:21005 for producing (kafka.producer.SyncProducer)
[2016-03-04 03:33:55,614] INFO Disconnecting from 192-168-234-231:21005 (kafka.producer.SyncProducer)
[2016-03-04 03:33:55,615] WARN Fetching topic metadata with correlation id 0 for topics [Set(test)] from broker [id:26,host: 192-168-234-231,port:21005] failed (kafka.client.ClientUtils$)
```

```
java.nio.channels.ClosedChannelException
at kafka.network.BlockingChannel.send(BlockingChannel.scala:100)
at kafka.producer.SyncProducer.liftedTree1$1(SyncProducer.scala:73)
at kafka.producer.SyncProducer.kafka$producer$SyncProducer$$doSend(SyncProducer.scala:72)
at kafka.producer.SyncProducer.send(SyncProducer.scala:113)
at kafka.client.ClientUtils$.fetchTopicMetadata(ClientUtils.scala:58)
at kafka.client.ClientUtils$.fetchTopicMetadata(ClientUtils.scala:93)
at kafka.consumer.ConsumerFetcherManager
$LeaderFinderThread.doWork(ConsumerFetcherManager.scala:66)
at kafka.utils.ShutdownableThread.run(ShutdownableThread.scala:60)
[2016-03-04 03:33:55,615] INFO Disconnecting from 192-168-234-231:21005
(kafka.producer.SyncProducer)
```

通过异常信息，发现当前Consumer无法从Kafka Broker 192-168-234-231节点获取元数据，导致无法连接正确的Broker获取消息。

6. 检查网络是否存在问题，如果网络没有问题，检查是否配置主机和IP的对应关系
   - Linux

     执行**cat /etc/hosts**命令。

     ```
     127.0.0.1       localhost

     192.168.0.131 192-168-0-131
     192.168.0.51 192-168-0-51
     192.168.0.122 192-168-0-122
     ```

   - Windows

     打开“C:\Windows\System32\drivers\etc\hosts”。

     ```
     # For example:
     #
     #      192.168.94.97     rhino.acme.com          # source server
     #      192.168.63.10     x.acme.com              # x client host

     # localhost name resolution is handled within DNS itself.
     #       127.0.0.1       localhost
     #       ::1             localhost
     10.82.129.120 rms.[redacted].com # modified by IrmTool at 2015-01-18 17:55:13
     ```

     解决方法参考**步骤4**。

解决办法

步骤1 ZooKeeper连接地址配置错误。

步骤2 修改Consumer配置中的ZooKeeper连接地址信息，保证和MRS相一致。

- Flume

  ```
  server.sources.Source02.type=org.apache.flume.source.kafka.KafkaSource
  server.sources.Source02.zookeeperConnect=192.168.234.231:2181/kafka
  server.sources.Source02.topic = test
  server.sources.Source02.groupId = test_01
  ```

- Spark

  ```
  val zkQuorum = "192.168.234.231:2181/kafka"
  ```

- Storm

  ```
  BrokerHosts brokerHosts = new ZKHosts("192.168.234.231:2181/kafka");
  ```

- Consumer API

```
zookeeper.connect="192.168.234.231:2181/kafka"
```

步骤3 Rebalance异常。

同一个消费者组(consumer group)有多个consumer先后启动，就是一个消费者组内有多个consumer同时消费多个partition数据，consumer端也会有负载均衡（consumer个数小于partitions数量时）。

consumer实际上是靠存储在zk中的临时节点来表明针对哪个topic的哪个partition拥有读权限的。所在路径为：/consumers/consumer-group-xxx/owners/topic-xxx/x。

当触发负载均衡后，原来的consumer会重新计算并释放已占用的partitions，此过程需要一定的处理时间，新来的consumer抢占该partitions时很有可能会失败。

表 16-1 参数说明

名称	作用	默认值
rebalance.max.retries	Rebalance最大重试次数	4
rebalance.backoff.ms	Rebalance每次重试间隔	2000
zookeeper.session.timeout.ms	Zookeeper连接会话超时时间	15000

可以适当调大上述三个参数，可以参考如下数值：

```
zookeeper.session.timeout.ms = 45000
rebalance.max.retries = 10
rebalance.backoff.ms = 5000
```

参数设置应遵循：

rebalance.max.retries * rebalance.backoff.ms > zookeeper.session.timeout.ms

步骤4 网络异常。

在hosts文件中没有配置主机名和IP的对应关系，导致使用主机名进行访问时，无法获取信息。

步骤5 在hosts文件中添加对应的主机名和IP的对应关系。

- Linux

```
127.0.0.1       localhost


192.168.0.131 192-168-0-131
192.168.0.51 192-168-0-51
192.168.0.122 192-168-0-122
192.168.234.231 192-168-234-231
```

- Windows

```
# For example:
#
#      192.168.94.97     rhino.acme.com          # source server
#      192.168.63.10     x.acme.com              # x client host

# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1             localhost
10.82.129.120 rms       .com # modified by IrmTool at 2015-01-18 17:55:13
192.168.234.231 192-168-234-231
```

----结束

16.8 Consumer 消费数据失败，Consumer 一直处于等待状态

问题现象

使用MRS服务安装集群，主要安装ZooKeeper、Kafka。

在使用Consumer从Kafka消费数据时，发现客户端一直处于等待状态。

可能原因

1. Kafka服务异常。
2. 客户端Consumer侧采用非安全访问，服务端配置禁止访问。
3. 客户端Consumer侧采用非安全访问，Kafka Topic设置ACL。

原因分析

Consumer向Kafka消费数据失败，可能原因客户端Consumer侧问题或者Kafka侧问题。

1. 查看Kafka服务状态：
   - MRS Manager界面操作：登录MRS Manager，依次选择“服务管理 > Kafka”，查看当前Kafka状态，发现状态为良好，且监控指标内容显示正确。
   - FusionInsight Manager界面操作：登录FusionInsight Manager，选择“集群 > 服务 > Kafka”，查看当前Kafka状态，发现状态为良好，且监控指标内容显示正确。
2. 查看Consumer客户端日志发现频繁连接Broker节点和断链打印信息，如下所示。

```
[root@10-10-144-2 client]# kafka-console-consumer.sh --topic test --zookeeper 10.5.144.2:2181/kafka
--from-beginning
[2017-03-07 09:22:00,658] INFO Fetching metadata from broker BrokerEndPoint(1,10.5.144.2,9092)
with correlation id 26 for 1 topic(s) Set(test) (kafka.client.ClientUtils$)
[2017-03-07 09:22:00,659] INFO Connected to 10.5.144.2:9092 for producing
(kafka.producer.SyncProducer)
[2017-03-07 09:22:00,659] INFO Disconnecting from 10.5.144.2:9092 (kafka.producer.SyncProducer)
```

   Consumer采用9092端口来访问Kafka，9092为非安全端口。
3. 通过Manager页面查看当前Kafka集群配置，发现未配置自定义参数“allow.everyone.if.no.acl.found”=“false”。
   - MRS Manager界面操作入口：登录MRS Manager，依次选择“服务管理 > Kafka> 配置”。

– FusionInsight Manager界面操作入口：登录FusionInsight Manager，选择“集群 > 服务 > Kafka > 配置”。

4. 当ACL设置为false不允许采用9092来进行访问。
5. 查看Consumer客户端日志发现频繁连接Broker节点和断链打印信息，如下所示。

```
[root@10-10-144-2 client]# kafka-console-consumer.sh --topic test_acl --zookeeper 10.5.144.2:2181/kafka --from-beginning
[2017-03-07 09:49:16,992] INFO Fetching metadata from broker BrokerEndPoint(2,10.5.144.3,9092) with correlation id 16 for 1 topic(s) Set(topic_acl) (kafka.client.ClientUtils$)
[2017-03-07 09:49:16,993] INFO Connected to 10.5.144.3:9092 for producing (kafka.producer.SyncProducer)
[2017-03-07 09:49:16,994] INFO Disconnecting from 10.5.144.3:9092 (kafka.producer.SyncProducer)
```

Consumer采用21005端口来访问Kafka，21005为非安全端口。

6. 通过客户端命令查看topic的ACL权限设置信息。

```
[root@10-10-144-2 client]# kafka-acls.sh --authorizer-properties zookeeper.connect=10.5.144.2:2181/kafka --list --topic topic_acl
Current ACLs for resource `Topic:topic_acl`:
 User:test_user has Allow permission for operations: Describe from hosts: *
User:test_user has Allow permission for operations: Write from hosts: *
```

Topic设置ACL，则不允许采用9092来访问。

7. 查看Consumer客户端日志发现打印信息，如下所示。

```
[root@10-10-144-2 client]# kafka-console-consumer.sh --topic topic_acl --bootstrap-server 10.5.144.2:21007 --consumer.config /opt/client/Kafka/kafka/config/consumer.properties --from-beginning --new-consumer
[2017-03-07 10:19:18,478] INFO Kafka version : 0.9.0.0 (org.apache.kafka.common.utils.AppInfoParser)
[2017-03-07 10:19:18,478] INFO Kafka commitId : unknown (org.apache.kafka.common.utils.AppInfoParser)
```

Consumer采用21007端口来访问Kafka。

8. 通过客户端命令klist查询当前认证用户：

```
[root@10-10-144-2 client]# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: test@HADOOP.COM

Valid starting     Expires            Service principal
01/25/17 11:06:48  01/26/17 11:06:45  krbtgt/HADOOP.COM@HADOOP.COM
```

当前认证用户为**test**。

9. 通过客户端命令查看topic的acl权限设置信息。

```
[root@10-10-144-2 client]# kafka-acls.sh --authorizer-properties zookeeper.connect=10.5.144.2:24002/kafka --list --topic topic_acl
Current ACLs for resource `Topic:topic_acl`:
 User:test_user has Allow permission for operations: Describe from hosts: *
 User:test_user has Allow permission for operations: Write from hosts: *
 User:ttest_user has Allow permission for operations: Read from hosts: *
```

Topic设置acl，test无权限进行consumer操作。

解决方法参考**步骤2**。

10. 通过SSH登录Kafka Broker：

通过**cd /var/log/Bigdata/kafka/broker**命令进入日志目录。

查看kafka-authorizer.log发现如下日志提示用户不属于kafka或者kafkaadmin组。

```
2017-01-25 13:26:33,648 | INFO  | [kafka-request-handler-0] | The principal is test, belongs to Group : [hadoop, ficommon] | kafka.authorizer.logger (SimpleAclAuthorizer.scala:169)
2017-01-25 13:26:33,648 | WARN  | [kafka-request-handler-0] | The user is not belongs to kafka or kafkaadmin group, authorize failed! | kafka.authorizer.logger (SimpleAclAuthorizer.scala:170)
```

解决方法参考**步骤3**。

解决办法

步骤1 配置自定义参数“allow.everyone.if.no.acl.found”参数为“true”，重启Kafka服务。

步骤2 采用具有权限用户登录。

例如：

kinit test_user

或者赋予用户相关权限。

须知

需要使用Kafka管理员用户（属于kafkaadmin组）操作。

例如：

kafka-acls.sh --authorizer-properties zookeeper.connect=10.5.144.2:2181/kafka --topic topic_acl --consumer --add --allow-principal User:test --group test

```
[root@10-10-144-2 client]# kafka-acls.sh --authorizer-properties zookeeper.connect=8.5.144.2:2181/kafka --
list --topic topic_acl
Current ACLs for resource `Topic:topic_acl`:
 User:test_user has Allow permission for operations: Describe from hosts: *
 User:test_user has Allow permission for operations: Write from hosts: *
  User:test has Allow permission for operations: Describe from hosts: *
  User:test has Allow permission for operations: Write from hosts: *
  User:test has Allow permission for operations: Read from hosts: *
```

步骤3 用户加入**Kafka**组或者**Kafkaadmin**组。

----结束

16.9 SparkStreaming 消费 Kafka 消息失败，提示“Error getting partition metadata”

问题现象

使用SparkStreaming来消费Kafka中指定Topic的消息时，发现无法从Kafka中获取到数据。提示如下错误： Error getting partition metadata。

```
Exception in thread "main" org.apache.spark.SparkException:  Error getting partition metadata for
'testtopic'. Does the topic exist?
org.apache.spark.streaming.kafka.KafkaCluster$$anonfun$checkErrors$1.apply(KafkaCluster.scala:366)
org.apache.spark.streaming.kafka.KafkaCluster$$anonfun$checkErrors$1.apply(KafkaCluster.scala:366)
scala.util.Either.fold(Either.scala:97)
org.apache.spark.streaming.kafka.KafkaCluster$.checkErrors(KafkaCluster.scala:365)
org.apache.spark.streaming.kafka.KafkaUtils$.createDirectStream(KafkaUtils.scala:422)
com.xxx.bigdata.spark.examples.FemaleInfoCollectionPrint$.main(FemaleInfoCollectionPrint.scala:45)
com.xxx.bigdata.spark.examples.FemaleInfoCollectionPrint.main(FemaleInfoCollectionPrint.scala)
sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
java.lang.reflect.Method.invoke(Method.java:498)
org.apache.spark.deploy.SparkSubmit$.org$apache$spark$deploy$SparkSubmit$
$runMain(SparkSubmit.scala:762)
org.apache.spark.deploy.SparkSubmit$.doRunMain$1(SparkSubmit.scala:183)
org.apache.spark.deploy.SparkSubmit$.submit(SparkSubmit.scala:208)
```

```
org.apache.spark.deploy.SparkSubmit$.main(SparkSubmit.scala:123)
org.apache.spark.deploy.SparkSubmit.main(SparkSubmit.scala)
```

可能原因

1. Kafka服务异常。
2. 客户端Consumer侧采用非安全访问，服务端配置禁止访问。
3. 客户端Consumer侧采用非安全访问，Kafka Topic设置ACL。

原因分析

1. 查看Kafka服务状态：
   - MRS Manager界面操作：登录MRS Manager，依次选择“服务管理 > Kafka”，查看当前Kafka状态，发现状态为良好，且监控指标内容显示正确。
   - FusionInsight Manager界面操作：登录FusionInsight Manager，选择“集群 > 服务 > Kafka”，查看当前Kafka状态，发现状态为良好，且监控指标内容显示正确。
2. 通过Manager页面，查看当前Kafka集群配置，发现未配置“allow.everyone.if.no.acl.found”或配置为“false”。
3. 当ACL设置为false不允许采用Kafka非安全端口21005来进行访问。
4. 通过客户端命令查看topic的ACL权限设置信息：

   ```
   [root@10-10-144-2 client]# kafka-acls.sh --authorizer-properties zookeeper.connect=10.5.144.2:2181/kafka --list --topic topic_acl
   Current ACLs for resource `Topic:topic_acl`:
    User:test_user has Allow permission for operations: Describe from hosts: *
   User:test_user has Allow permission for operations: Write from hosts: *
   ```

   Topic设置ACL，则不允许采用Kafka非安全端口21005来访问。

解决办法

步骤1 修改或者添加自定义配置“allow.everyone.if.no.acl.found”参数为“true”，重启Kafka服务。

步骤2 删除Topic设置的ACL。

例如：

kinit test_user

须知

需要使用Kafka管理员用户（属于kafkaadmin组）操作。

例如：

kafka-acls.sh --authorizer-properties zookeeper.connect=10.5.144.2:2181/kafka --remove --allow-principal User:test_user --producer --topic topic_acl

kafka-acls.sh --authorizer-properties zookeeper.connect=10.5.144.2:2181/kafka --remove --allow-principal User:test_user --consumer --topic topic_acl --group test

----结束

16.10 新建集群 Consumer 消费数据失败，提示“GROUP_COORDINATOR_NOT_AVAILABLE”

问题背景与现象

新建Kafka集群，部署Broker节点数为2，使用Kafka客户端可以正常生产，但是无法正常消费。Consumer消费数据失败，提示GROUP_COORDINATOR_NOT_AVAILABLE，关键日志如下：

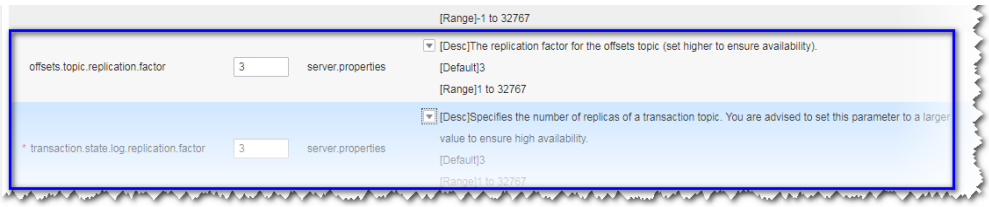
```
2018-05-12 10:58:42,561 | INFO | [kafka-request-handler-3] | [GroupCoordinator 2]: Preparing to restabilize group DemoConsumer with old generation 118 | kafka.coordinator.GroupCoordinator (Logging.scala:68)
2018-05-12 10:59:13,562 | INFO | [executor-Heartbeat] | [GroupCoordinator 2]: Preparing to restabilize group DemoConsumer with old generation 119 | kafka.coordinator.GroupCoordinator (Logging.scala:68)
```

可能原因

__consumer_offsets无法创建。

原因分析

1. 查看日志，发现大量__consumer_offset创建异常。
2. 查看集群发现当前Broker数量为2。
3. 查看__cosumer_offset topic要求副本为3，因此创建失败。



解决办法

可以将扩容至少3个流式core节点，或参考如下步骤修改服务配置参数。

步骤1 进入服务参数配置界面。

- MRS Manager界面操作：登录MRS Manager，选择“服务管理 > Kafka > 服务配置”，“参数类别”设置为“全部配置”。
- FusionInsight Manager界面操作：登录FusionInsight Manager。选择“集群 > 服务 > Kafka”，单击“配置”，选择“全部配置”。

步骤2 搜索并修改offsets.topic.replication.factor和transaction.state.log.replication.factor的值为2。

步骤3 保存配置，并重启配置过期的服务或者实例。

----结束

16.11 SparkStreaming 消费 Kafka 消息失败，提示“Couldn't find leader offsets”

问题背景与现象

使用SparkStreaming来消费Kafka中指定Topic的消息时，发现无法从Kafka中获取到数据。

提示如下错误： Couldn't find leader offsets。

```
2018-05-30 12:01:17,816 | INFO  | [Driver] | Reconnect due to socket error: java.net.SocketTimeoutException | kafka.utils.Logging$class.info(Logging.scala:68)
2018-05-30 12:01:47,859 | ERROR | [Driver] | User class threw exception: org.apache.spark.SparkException: java.net.SocketTimeoutException
org.apache.spark.SparkException: Couldn't find leader offsets for Set([STK6,57], [STK6,21]) | org.apache.spark.Logging$class.logError(Logging.scala:96)
org.apache.spark.SparkException: java.net.SocketTimeoutException
org.apache.spark.SparkException: Couldn't find leader offsets for Set([STK6,57], [STK6,21])
        at org.apache.spark.streaming.kafka.KafkaCluster$$anonfun$checkErrors$1.apply(KafkaCluster.scala:366)
        at org.apache.spark.streaming.kafka.KafkaCluster$$anonfun$checkErrors$1.apply(KafkaCluster.scala:366)
        at scala.util.Either.fold(Either.scala:97)
        at org.apache.spark.streaming.kafka.KafkaCluster$.checkErrors(KafkaCluster.scala:365)
        at org.apache.spark.streaming.kafka.KafkaUtils$.createDirectStream(KafkaUtils.scala:422)
        at org.apache.spark.streaming.kafka.KafkaUtils$.createDirectStream(KafkaUtils.scala:532)
        at org.apache.spark.streaming.kafka.KafkaUtils.createDirectStream(KafkaUtils.scala)
        at com.stk.bigdata.sparkstreaming.notify.SparkAlarmControlwz.main(SparkAlarmControlwz.java:194)
        at com.stk.bigdata.sparkstreaming.submit.SparkNotifyA.main(SparkNotifyA.java:14)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at org.apache.spark.deploy.yarn.ApplicationMaster$$anon$2.run(ApplicationMaster.scala:540)
2018-05-30 12:01:47,863 | INFO  | [Driver] | Final app status: FAILED, exitCode: 15, (reason: User class threw exception: org.apache.spark.SparkException: java.n
org.apache.spark.SparkException: Couldn't find leader offsets for Set([STK6,57], [STK6,21])) | org.apache.spark.Logging$class.logInfo(Logging.scala:59)
2018-05-30 12:01:47,866 | INFO  | [pool-1-thread-1] | Invoking stop() from shutdown hook | org.apache.spark.Logging$class.logInfo(Logging.scala:59)
```

可能原因

- Kafka服务异常。
- 网络异常。
- Kafka Topic异常。

原因分析

步骤1 通过Manager页面，查看Kafka集群当前状态，发现状态为“良好”，且监控指标内容显示正确。

步骤2 查看SparkStreaming日志中提示错误的Topic信息。

执行Kafka相关命令，获取Topic分布信息和副本同步信息，观察返回结果。

kafka-topics.sh --describe --zookeeper *<zk_host:port/chroot>* **--topic** *<topic name>*

如下所示，发现对应Topic状态正常。所有Partition均存在正常Leader信息。

图 16-10 Topic 分布信息和副本同步信息

```
Topic: STK6     Partition: 36   Leader: 3       Replicas: 3,5   Isr: 3,5
Topic: STK6     Partition: 37   Leader: 4       Replicas: 4,6   Isr: 4,6
Topic: STK6     Partition: 38   Leader: 5       Replicas: 5,7   Isr: 5,7
Topic: STK6     Partition: 39   Leader: 6       Replicas: 6,8   Isr: 6,8
Topic: STK6     Partition: 40   Leader: 7       Replicas: 7,9   Isr: 7,9
Topic: STK6     Partition: 41   Leader: 8       Replicas: 8,1   Isr: 8,1
Topic: STK6     Partition: 42   Leader: 9       Replicas: 9,2   Isr: 9,2
Topic: STK6     Partition: 43   Leader: 1       Replicas: 1,3   Isr: 3,1
Topic: STK6     Partition: 44   Leader: 2       Replicas: 2,4   Isr: 2,4
Topic: STK6     Partition: 45   Leader: 3       Replicas: 3,6   Isr: 3,6
Topic: STK6     Partition: 46   Leader: 4       Replicas: 4,7   Isr: 4,7
Topic: STK6     Partition: 47   Leader: 5       Replicas: 5,8   Isr: 5
Topic: STK6     Partition: 48   Leader: 6       Replicas: 6,9   Isr: 6,9
Topic: STK6     Partition: 49   Leader: 7       Replicas: 7,1   Isr: 7,1
Topic: STK6     Partition: 50   Leader: 8       Replicas: 8,2   Isr: 2,8
Topic: STK6     Partition: 51   Leader: 9       Replicas: 9,3   Isr: 9,3
Topic: STK6     Partition: 52   Leader: 1       Replicas: 1,4   Isr: 4,1
Topic: STK6     Partition: 53   Leader: 2       Replicas: 2,5   Isr: 5,2
Topic: STK6     Partition: 54   Leader: 3       Replicas: 3,7   Isr: 3,7
Topic: STK6     Partition: 55   Leader: 4       Replicas: 4,8   Isr: 4,8
Topic: STK6     Partition: 56   Leader: 5       Replicas: 5,9   Isr: 5,9
Topic: STK6     Partition: 57   Leader: 6       Replicas: 6,1   Isr: 6,1
Topic: STK6     Partition: 58   Leader: 7       Replicas: 7,2   Isr: 2,7
```

步骤3 检查客户端与Kafka集群网络是否连通，若网络不通协调网络组进行处理。

步骤4 通过SSH登录Kafka Broker。

通过**cd /var/log/Bigdata/kafka/broker**命令进入日志目录。

查看server.log发现如下日志抛出java.lang.OutOfMemoryError: Direct buffer memory。

```
2018-05-30 12:02:00,246 | ERROR | [kafka-network-thread-6-PLAINTEXT-3] | Processor got uncaught exception. | kafka.network.Processor (Logging.scala:103)
 java.lang.OutOfMemoryError: Direct buffer memory
 at java.nio.Bits.reserveMemory(Bits.java:694)
 at java.nio.DirectByteBuffer.<init>(DirectByteBuffer.java:123)
 at java.nio.ByteBuffer.allocateDirect(ByteBuffer.java:311)
 at sun.nio.ch.Util.getTemporaryDirectBuffer(Util.java:241)
 at sun.nio.ch.IOUtil.read(IOUtil.java:195)
 at sun.nio.ch.SocketChannelImpl.read(SocketChannelImpl.java:380)
```

at org.apache.kafka.common.network.PlaintextTransportLayer.read(PlaintextTransportLayer.java:110)

步骤5 通过Manager页面，查看当前Kafka集群配置。

- MRS Manager界面操作：登录MRS Manager，选择“服务管理 > Kafka > 服务配置”，“参数类别”设置为“全部配置”，发现“KAFKA_JVM_PERFORMANCE_OPTS”的中“-XX:MaxDirectMemorySize”值为“1G”。
- FusionInsight Manager界面操作：登录FusionInsight Manager。选择“集群 > 服务 > Kafka”，单击“配置”，选择“全部配置”，发现“KAFKA_JVM_PERFORMANCE_OPTS”的中“-XX:MaxDirectMemorySize”值为“1G”。

步骤6 直接内存配置过小导致报错，而且一旦直接内存溢出，该节点将无法处理新请求，会导致其他节点或者客户端访问超时失败。

----结束

解决办法

步骤1 登录到Manager，进入Kafka配置页面。

步骤2 选择“全部配置”，搜索并修改KAFKA_JVM_PERFORMANCE_OPTS的值。

步骤3 保存配置，并重启配置过期的服务或者实例。

----结束

16.12 Consumer 消费数据失败，提示“SchemaException: Error reading field”

问题背景与现象

Consumer来消费Kafka中指定Topic的消息时，发现无法从Kafka中获取到数据。

提示如下错误：

org.apache.kafka.common.protocol.types.SchemaException: Error reading field 'brokers': Error reading field 'host': Error reading string of length 28271, only 593 bytes available

```
Exception in thread "Thread-0" org.apache.kafka.common.protocol.types.SchemaException: Error reading
field 'brokers': Error reading field 'host': Error reading string of length 28271, only 593 bytes available
at org.apache.kafka.common.protocol.types.Schema.read(Schema.java:73)
at org.apache.kafka.clients.NetworkClient.parseResponse(NetworkClient.java:380)
at org.apache.kafka.clients.NetworkClient.handleCompletedReceives(NetworkClient.java:449)
at org.apache.kafka.clients.NetworkClient.poll(NetworkClient.java:269)
at
org.apache.kafka.clients.consumer.internals.ConsumerNetworkClient.clientPoll(ConsumerNetworkClient.java:
360)
at
org.apache.kafka.clients.consumer.internals.ConsumerNetworkClient.poll(ConsumerNetworkClient.java:224)
at
org.apache.kafka.clients.consumer.internals.ConsumerNetworkClient.poll(ConsumerNetworkClient.java:192)
at
org.apache.kafka.clients.consumer.internals.ConsumerNetworkClient.poll(ConsumerNetworkClient.java:163)
atorg.apache.kafka.clients.consumer.internals.AbstractCoordinator.ensureCoordinatorReady(AbstractCoordina
tor.java:179)
at org.apache.kafka.clients.consumer.KafkaConsumer.pollOnce(KafkaConsumer.java:973)
at org.apache.kafka.clients.consumer.KafkaConsumer.poll(KafkaConsumer.java:937)
at KafkaNew.Consumer$ConsumerThread.run(Consumer.java:40)
```

可能原因

客户端和服务端Jar版本不一致。

解决办法

修改Consumer应用程序中Kafka jar，确保和服务端保持一致。

16.13 Kafka Consumer 消费数据丢失

问题背景与现象

用户将消费完的数据存入数据库，发现数据与生产数据不一致，怀疑Kafka消费丢数据。

可能原因

- 业务代码原因
- Kafka生产数据写入异常
- Kafka消费数据异常

解决办法

Kafka排查：

步骤1 通过consumer-groups.sh来观察写入和消费的offerset的变化情况（生产一定数量的消息，客户端进行消费，观察offerset的变化）。

```
2019-04-08 14:23:25,341] WARN [Principal=null]: TGT renewal thread has been interrupted and will exit. (org.apache.kafka.common.security.kerberos.KerberosLogin)
root@SmiBigdataCN3 kafka]# ./bin/kafka-consumer-groups.sh --describe --bootstrap-server 10.3.1.49:21007 --new-consumer --group yhdsbsj --command-config config/consum
properties
ote: This will only show information about consumers that use the Java consumer API (non-ZooKeeper-based consumers).

OPIC                           PARTITION  CURRENT-OFFSET  LOG-END-OFFSET  LAG        CONSUMER-ID                                    HOST
ENT-ID
LW5JDGB                        0          290078541       290078541       0          consumer-1-7bb54edf-9cbb-4d58-989b-1b4e6607217e   /10.2.1.180
sumer-1
LW5JDGB                        1          281608671       281608671       0          consumer-1-7bb54edf-9cbb-4d58-989b-1b4e6607217e   /10.2.1.180
sumer-1
LW5JDGB                        2          283880519       283880519       0          consumer-1-7bb54edf-9cbb-4d58-989b-1b4e6607217e   /10.2.1.180
sumer-1
```

步骤2 新建一个消费组，用客户端进行消费，然后查看消费的消息。

new-consumer:

kafka-console-consumer.sh --topic <topic name> --bootstrap-server <IP1:PORT, IP2:PORT,...> --new-consumer --consumer.config <config file>

----结束

业务代码排查：

步骤1 查看客户端里有没有提交offerset的报错。

步骤2 如果没有报错把消费的API里加上打印消息，打印少量数据（只打印key即可），查看丢失的数据。

----结束

16.14 账号锁定导致启动 Kafka 组件失败

问题背景与现象

新创建的MRS集群，启动Kafka服务失败。

查看服务启动日志，显示认证失败。

```
/home/omm/kerberos/bin/kinit -k -t ${BIGDATA_HOME}/etc/2_15_ Broker /kafka.keytab kafka/
hadoop.hadoop.com -c ${BIGDATA_HOME}/etc/2_15_ Broker /11846 failed.
export key tab file for kafka/hadoop.hadoop.com failed.export and check keytab file failed, errMsg=]}] for
Broker #192.168.1.92@192-168-1-92.
[2015-07-11 02:34:33] RoleInstance started failure for ROLE[name: Broker].
[2015-07-11 02:34:34] Failed to complete the instances start operation. Current operation entities: [Broker
#192.168.1.92@192-168-1-92], Failure entites : [Broker #192.168.1.92@192-168-1-92].Operation
Failed.Failed to complete the instances start operation. Current operation entities:
[Broker#192.168.1.92@192-168-1-92], Failure entites: [Broker #192.168.1.92@192-168-1-92].
```

原因分析

查看Kerberos日志“/var/log/Bigdata/kerberos/krb5kdc.log”，发现有集群外的IP使用Kafka用户连接，导致多次认证失败，最终导致Kafka账号被锁定。

```
Jul 11 02:49:16 192-168-1-91 krb5kdc[1863](info): AS_REQ (2 etypes {18 17}) 192.168.1.93:
NEEDED_PREAUTH: kafka/hadoop.hadoop.com@HADOOP.COM for krbtgt/HADOOP.COM@HADOOP.COM,
Additional pre-authentication required
Jul 11 02:49:16 192-168-1-91 krb5kdc[1863](info): preauth (encrypted_timestamp) verify failure: Decrypt
integrity check failed
Jul 11 02:49:16 192-168-1-91 krb5kdc[1863](info): AS_REQ (2 etypes {18 17}) 192.168.1.93:
PREAUTH_FAILED: kafka/hadoop.hadoop.com@HADOOP.COM for krbtgt/HADOOP.COM@HADOOP.COM,
Decrypt integrity check failed
```

解决办法

步骤1 确认集群外连接Kafka的节点IP地址信息（如原因分析示例中的192.168.1.93）。

步骤2 登录集群外的节点，断开其对Kafka的认证。

步骤3 等待5分钟，此账号就会被解锁。

步骤4 重新尝试启动Kafka服务。

----结束

16.15 Kafka Broker 上报进程异常，日志提示“IllegalArgumentException”

问题背景与现象

使用Manager提示进程故障告警，查看告警进程为Kafka Broker。

可能原因

Broker配置异常。

原因分析

1. 在Manager页面，在告警页面得到主机信息。
2. 通过SSH登录Kafka Broker，执行**cd /var/log/Bigdata/kafka/broker**命令进入日志目录。

   查看server.log发现如下日志抛出IllegalArgumentException异常，提示java.lang.IllegalArgumentException: requirement failed: replica.fetch.max.bytes should be equal or greater than message.max.bytes。

```
2017-01-25 09:09:14,930 | FATAL | [main] | | kafka.Kafka$ (Logging.scala:113)
java.lang.IllegalArgumentException: requirement failed: replica.fetch.max.bytes should be equal or
```

```
greater than message.max.bytes
        at scala.Predef$.require(Predef.scala:233)
        at kafka.server.KafkaConfig.validateValues(KafkaConfig.scala:959)
        at kafka.server.KafkaConfig.<init>(KafkaConfig.scala:944)
        at kafka.server.KafkaConfig$.fromProps(KafkaConfig.scala:701)
        at kafka.server.KafkaConfig$.fromProps(KafkaConfig.scala:698)
        at kafka.server.KafkaServerStartable$.fromProps(KafkaServerStartable.scala:28)
        at kafka.Kafka$.main(Kafka.scala:60)
        at kafka.Kafka.main(Kafka.scala)
```

Kafka要求replica.fetch.max.bytes需要大于等于message.max.bytes。

3. 进入Kafka配置页面，选择“全部配置”， 显示所有Kafka相关配置，分别搜索message.max.bytes、

   replica.fetch.max.bytes进行检索，发现replica.fetch.max.bytes小于message.max.bytes。

解决办法

步骤1 登录Manager界面，进入Kafka配置页面。

- MRS 3.x之前的版本：登录MRS Manager，选择“服务管理 > Kafka > 配置 > 全部配置”。
- MRS 3.x及后续版本，登录FusionInsight Manager，选择“集群 > 服务 > Kafka > 配置 > 全部配置”。

步骤2 搜索并修改replica.fetch.max.bytes参数，使得replica.fetch.max.bytes的值大于等于message.max.bytes，使得不同Broker上的Partition的Replica可以同步到全部消息。

步骤3 保存配置，查看集群是否存在配置过期的服务，如果存在，需重启对应服务或角色实例使配置生效。

步骤4 修改消费者业务应用中fetch.message.max.bytes，使得fetch.message.max.bytes的值大于等于message.max.bytes，确保消费者可以消费到全部消息。

----结束

16.16 执行 Kafka Topic 删除操作时发现无法删除

问题背景与现象

在使用Kafka客户端命令删除Topic时，发现Topic无法被删除。

```
kafka-topics.sh --delete --topic test --zookeeper 192.168.234.231:2181/kafka
```

可能原因

- 客户端命令连接ZooKeeper地址错误。
- Kafka服务异常Kafka部分节点处于停止状态。
- Kafka服务端配置禁止删除。
- Kafka配置自动创建，且Producer未停止。

原因分析

1. 客户端命令，打印ZkTimeoutException异常。

```
[2016-03-09 10:41:45,773] WARN Can not get the principle name from server 192.168.234.231
(org.apache.zookeeper.ClientCnxn)
```

```
Exception in thread "main" org.I0Itec.zkclient.exception.ZkTimeoutException: Unable to connect to
zookeeper server within timeout: 30000
at org.I0Itec.zkclient.ZkClient.connect(ZkClient.java:880)
at org.I0Itec.zkclient.ZkClient.<init>(ZkClient.java:98)
at org.I0Itec.zkclient.ZkClient.<init>(ZkClient.java:84)
at kafka.admin.TopicCommand$.main(TopicCommand.scala:51)
at kafka.admin.TopicCommand.main(TopicCommand.scala)
```

解决方法参考**步骤1**。

2. 客户端查询命令。

kafka-topics.sh --list --zookeeper 192.168.0.122:2181/kafka

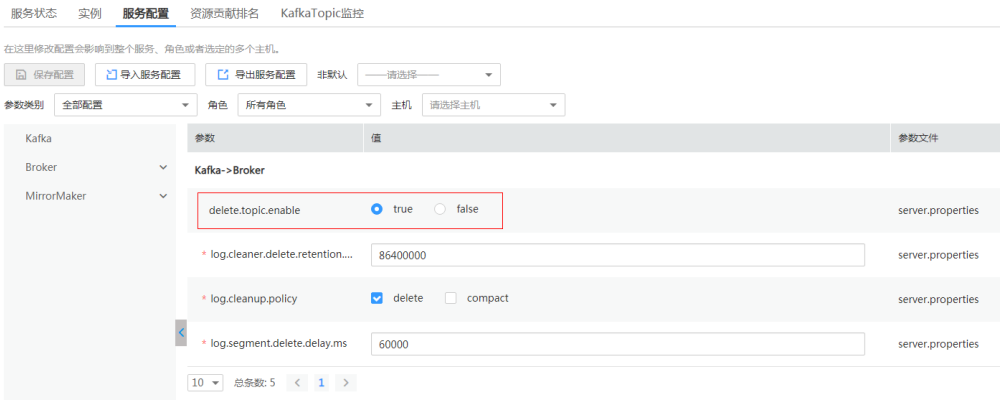
```
test - marked for deletion
```

通过Manager查看Kafka Broker实例的运行状态。

通过**cd /var/log/Bigdata/kafka/broker**命令进入RunningAsController节点日志目录，在controller.log发现ineligible for deletion: test。

```
2016-03-09 11:11:26,228 | INFO  | [main] | [Controller 1]: List of topics to be deleted:  |
kafka.controller.KafkaController (Logging.scala:68)
2016-03-09 11:11:26,229 | INFO  | [main] | [Controller 1]: List of topics ineligible for deletion: test |
kafka.controller.KafkaController (Logging.scala:68)
```

3. 通过Manager查询Broker删除Topic相关配置。



解决方法参考**步骤2**。

4. 客户端查询命令：

```
kafka-topics.sh --describe -topic test --zookeeper 192.168.0.122:2181/kafka
```

```
Topic:test      PartitionCount:10       ReplicationFactor:2     Configs:
        Topic: test     Partition: 0    Leader: -1      Replicas: 1,2   Isr:
        Topic: test     Partition: 1    Leader: -1      Replicas: 2,3   Isr:
        Topic: test     Partition: 2    Leader: -1      Replicas: 3,1   Isr:
        Topic: test     Partition: 3    Leader: -1      Replicas: 1,3   Isr:
        Topic: test     Partition: 4    Leader: -1      Replicas: 2,1   Isr:
        Topic: test     Partition: 5    Leader: -1      Replicas: 3,2   Isr:
        Topic: test     Partition: 6    Leader: -1      Replicas: 1,2   Isr:
        Topic: test     Partition: 7    Leader: -1      Replicas: 2,3   Isr:
        Topic: test     Partition: 8    Leader: -1      Replicas: 3,1   Isr:
        Topic: test     Partition: 9    Leader: -1      Replicas: 1,3   Isr:
```

进入RunningAsController节点日志目录，在controller.log发现marked ineligible for deletion。

```
2016-03-10 11:11:17,989 | INFO  | [delete-topics-thread-3] | [delete-topics-thread-3], Handling
deletion for topics test | kafka.controller.TopicDeletionManager$DeleteTopicsThread (Logging.scala:68)
2016-03-10 11:11:17,990 | INFO  | [delete-topics-thread-3] | [delete-topics-thread-3], Not retrying
deletion of topic test at this time since it is marked ineligible for deletion |
kafka.controller.TopicDeletionManager$DeleteTopicsThread (Logging.scala:68)
```

5. 通过Manager查询Broker状态。

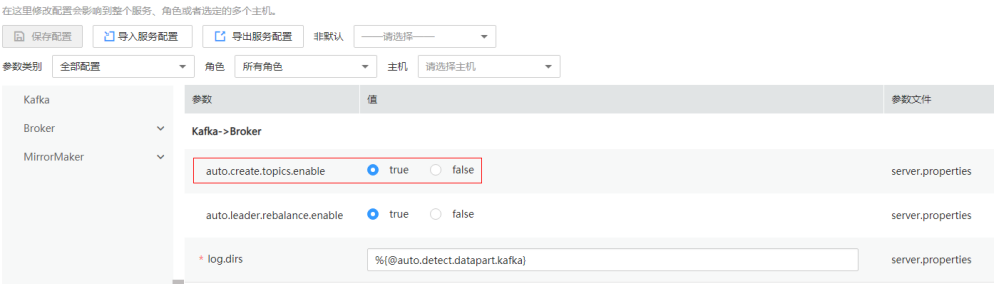
其中一个Broker处于停止或者故障状态。Topic进行删除必须保证该Topic的所有Partition所在的Broker必须处于正常状态。

解决方法参考**步骤3**。

6. 进入RunningAsController节点日志目录，在controller.log发现Deletion successfully，然后又出现New topics: [Set(test)]，表明被再次创建。

```
2016-03-10 11:33:35,208 | INFO  | [delete-topics-thread-3] | [delete-topics-thread-3], Deletion of topic test successfully completed | kafka.controller.TopicDeletionManager$DeleteTopicsThread (Logging.scala:68)

2016-03-10 11:33:38,501 | INFO  | [ZkClient-EventThread-19-192.168.0.122:2181,160.172.0.52:2181,160.172.0.51:2181/kafka] | [TopicChangeListener on Controller 3]: New topics: [Set(test)], deleted topics: [Set()], new partition replica assignment
```

7. 通过Manager查询Broker创建Topic相关配置。



经确认，对该Topic操作的应用没有停止。

解决方法参考**步骤4**。

解决办法

步骤1 ZooKeeper连接失败导致。

Kafka客户端连接ZooKeeper服务超时。检查客户端到ZooKeeper的网络连通性。

网络连接失败，通过Manager界面查看Zookeeper服务信息。

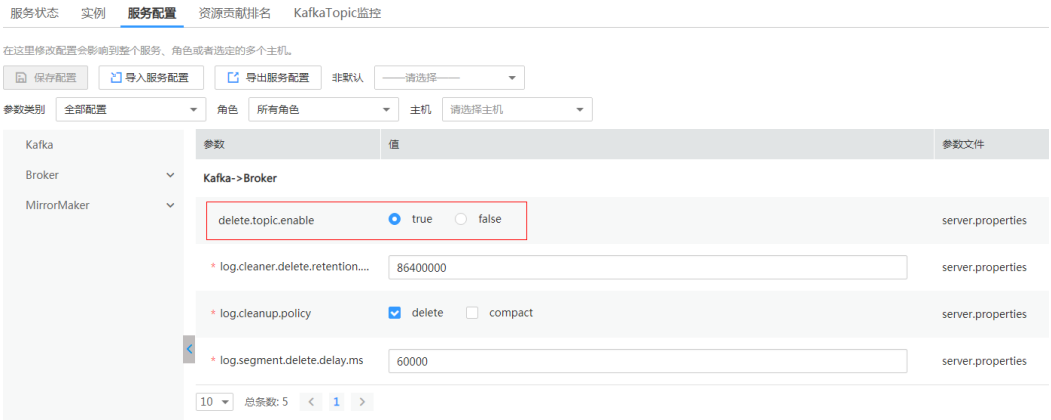
图 16-11 Zookeeper 服务信息

配置错误，修改客户端命令中ZooKeeper地址。

步骤2 Kafka服务端配置禁止删除。

通过Manager界面修改delete.topic.enable为true。保存配置并重启服务。

图 16-12 修改 delete.topic.enable



客户端查询命令，无Topic:test。

```
kafka-topics.sh --list --zookeeper 192.168.0.122:24002/kafka
```

进入RunningAsController节点日志目录，在controller.log发现Deletion of topic test successfully。

```
2016-03-10 10:39:40,665 | INFO  | [delete-topics-thread-3] | [Partition state machine on Controller 3]: Invoking state change to OfflinePartition for partitions [test,2],[test,15],[test,6],[test,16],[test,12],[test,7],[test,10],[test,13],[test,9],[test,19],[test,3],[test,5],[test,1],[test,0],[test,17],[test,8],[test,4],[test,11],[test,14],[test,18] | kafka.controller.PartitionStateMachine (Logging.scala:68)
2016-03-10 10:39:40,668 | INFO  | [delete-topics-thread-3] | [Partition state machine on Controller 3]: Invoking state change to NonExistentPartition for partitions [test,2],[test,15],[test,6],[test,16],[test,12],[test,7],[test,10],[test,13],[test,9],[test,19],[test,3],[test,5],[test,1],[test,0],[test,17],[test,8],[test,4],[test,11],[test,14],[test,18] | kafka.controller.PartitionStateMachine (Logging.scala:68)
2016-03-10 10:39:40,977 | INFO  | [delete-topics-thread-3] | [delete-topics-thread-3], Deletion of topic test successfully completed | kafka.controller.TopicDeletionManager$DeleteTopicsThread (Logging.scala:68)
```

步骤3 Kafka部分节点处于停止或者故障状态。

启动停止的Broker实例。

客户端查询命令，无Topic:test。

```
kafka-topics.sh --list --zookeeper 192.168.0.122:24002/kafka
```

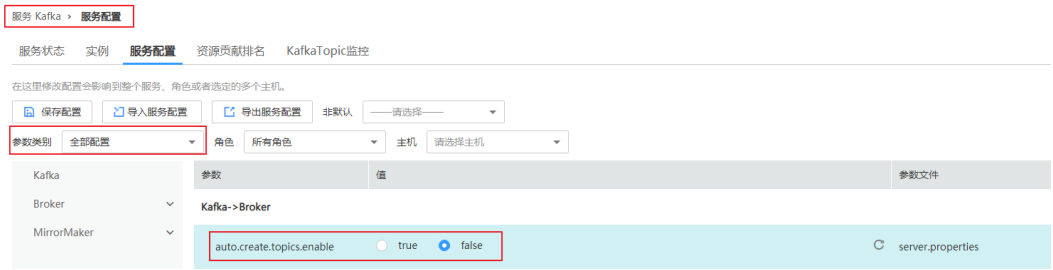
进入RunningAsController节点日志目录，在controller.log发现Deletion of topic test successfully。

```
2016-03-10 11:17:56,463 | INFO  | [delete-topics-thread-3] | [Partition state machine on Controller 3]: Invoking state change to NonExistentPartition for partitions [test,4],[test,1],[test,8],[test,2],[test,5],[test,9],[test,7],[test,6],[test,0],[test,3] | kafka.controller.PartitionStateMachine (Logging.scala:68)
2016-03-10 11:17:56,726 | INFO  | [delete-topics-thread-3] | [delete-topics-thread-3], Deletion of topic test successfully completed | kafka.controller.TopicDeletionManager$DeleteTopicsThread (Logging.scala:68)
```

步骤4 Kafka配置自动创建，且Producer未停止。

停止相关应用，通过Manager界面修改“auto.create.topics.enable”为“false”，保存配置并重启服务。

图 16-13 修改 auto.create.topics.enable



步骤5 再次执行delete操作。

----结束

16.17 执行 Kafka Topic 删除操作时提示“AdminOperationException”

问题背景与现象

在使用Kafka客户端命令设置Topic ACL权限时，发现Topic无法被设置。

```
kafka-topics.sh --delete --topic test4 --zookeeper 10.5.144.2:2181/kafka
```

提示错误ERROR kafka.admin.AdminOperationException: Error while deleting topic test4。

具体如下：

```
Error while executing topic command : Error while deleting topic test4
[2017-01-25 14:00:20,750] ERROR kafka.admin.AdminOperationException: Error while deleting topic test4
at kafka.admin.TopicCommand$$anonfun$deleteTopic$1.apply(TopicCommand.scala:177)
at kafka.admin.TopicCommand$$anonfun$deleteTopic$1.apply(TopicCommand.scala:162)
at scala.collection.mutable.ResizableArray$class.foreach(ResizableArray.scala:59)
at scala.collection.mutable.ArrayBuffer.foreach(ArrayBuffer.scala:47)
at kafka.admin.TopicCommand$.deleteTopic(TopicCommand.scala:162)
at kafka.admin.TopicCommand$.main(TopicCommand.scala:68)
at kafka.admin.TopicCommand.main(TopicCommand.scala)
 (kafka.admin.TopicCommand$)
```

可能原因

用户不属于**kafkaadmin**组，Kafka提供安全访问接口，kafkaamdin组用户才可以进行topic删除操作。

原因分析

1. 使用客户端命令，打印AdminOperationException异常。
2. 通过客户端命令**klist**查询当前认证用户：

```
[root@10-10-144-2 client]# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: test@HADOOP.COM

Valid starting     Expires            Service principal
01/25/17 11:06:48  01/26/17 11:06:45  krbtgt/HADOOP.COM@HADOOP.COM
```

如上例中当前认证用户为test。

3. 通过命令**id**查询用户组信息

```
[root@10-10-144-2 client]# id test
uid=20032(test) gid=10001(hadoop) groups=10001(hadoop),9998(ficommon),10003(kafka)
```

解决办法

MRS Manager界面操作：

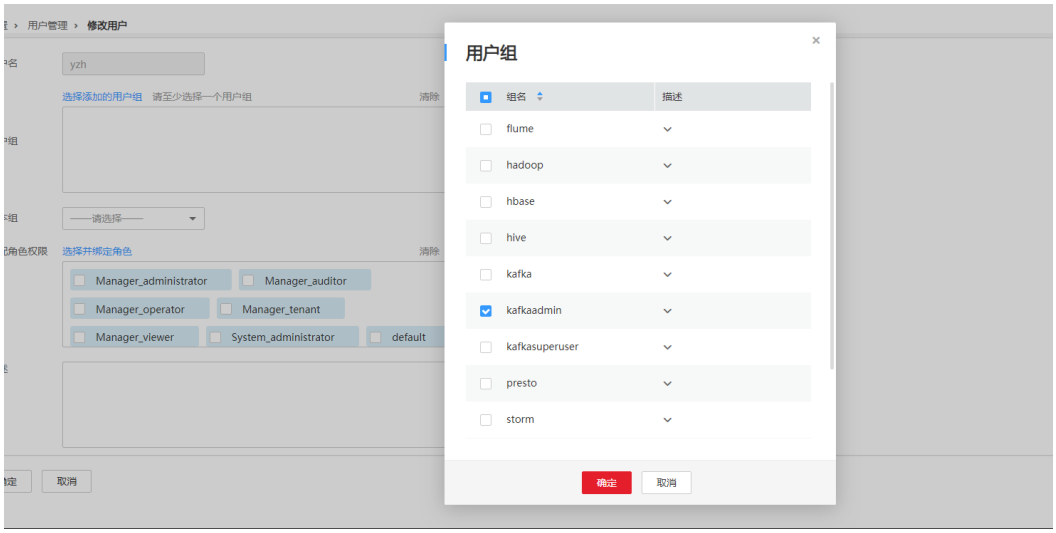
步骤1 登录MRS Manager。

步骤2 选择“系统设置 > 用户管理”。

步骤3 在操作用户对应的“操作”列，单击“修改”。

步骤4 为用户加入**kafkaadmin**组。单击“确定”完成修改操作。

图 16-14 修改用户信息



步骤5 通过命令**id**查询用户组信息。

```
[root@10-10-144-2 client]# id test
uid=20032(test) gid=10001(hadoop) groups=10001(hadoop),9998(ficommon),10002(kafkaadmin),
10003(kafka)
```

----结束

FusionInsight Manager界面操作：

步骤1 登录FusionInsight Manager。

步骤2 选择“系统 > 权限 > 用户”。

步骤3 在使用的用户所在行的单击“修改”。

步骤4 为用户添加**kafkaadmin**组。单击“确定”完成修改操作。

步骤5 通过命令**id**查询用户组信息。

```
[root@10-10-144-2 client]# id test
uid=20032(test) gid=10001(hadoop) groups=10001(hadoop),9998(ficommon),10002(kafkaadmin),
10003(kafka)
```

----结束

16.18 执行 Kafka Topic 创建操作时发现无法创建提示“NoAuthException”

问题背景与现象

在使用Kafka客户端命令创建Topic时，发现Topic无法被创建。

```
kafka-topics.sh --create --zookeeper 192.168.234.231:2181/kafka --replication-factor 1 --partitions 2 --topic test
```

提示错误NoAuthException，KeeperErrorCode = NoAuth for /config/topics。

具体如下：

```
Error while executing topic command org.apache.zookeeper.KeeperException$NoAuthException: KeeperErrorCode = NoAuth for /config/topics
org.I0Itec.zkclient.exception.ZkException: org.apache.zookeeper.KeeperException$NoAuthException: KeeperErrorCode = NoAuth for /config/topics
 at org.I0Itec.zkclient.exception.ZkException.create(ZkException.java:68)
 at org.I0Itec.zkclient.ZkClient.retryUntilConnected(ZkClient.java:685)
 at org.I0Itec.zkclient.ZkClient.create(ZkClient.java:304)
 at org.I0Itec.zkclient.ZkClient.createPersistent(ZkClient.java:213)
 at kafka.utils.ZkUtils$.createParentPath(ZkUtils.scala:215)
 at kafka.utils.ZkUtils$.updatePersistentPath(ZkUtils.scala:338)
 at kafka.admin.AdminUtils$.writeTopicConfig(AdminUtils.scala:247)
```

可能原因

用户不属于**kafkaadmin**组，Kafka提供安全访问接口，kafkaamdin组用户才可以进行topic删除操作。

原因分析

1. 使用客户端命令，打印NoAuthException异常。

```
Error while executing topic command org.apache.zookeeper.KeeperException$NoAuthException: KeeperErrorCode = NoAuth for /config/topics
org.I0Itec.zkclient.exception.ZkException: org.apache.zookeeper.KeeperException$NoAuthException: KeeperErrorCode = NoAuth for /config/topics
 at org.I0Itec.zkclient.exception.ZkException.create(ZkException.java:68)
 at org.I0Itec.zkclient.ZkClient.retryUntilConnected(ZkClient.java:685)
 at org.I0Itec.zkclient.ZkClient.create(ZkClient.java:304)
 at org.I0Itec.zkclient.ZkClient.createPersistent(ZkClient.java:213)
 at kafka.utils.ZkUtils$.createParentPath(ZkUtils.scala:215)
 at kafka.utils.ZkUtils$.updatePersistentPath(ZkUtils.scala:338)
 at kafka.admin.AdminUtils$.writeTopicConfig(AdminUtils.scala:247)
```

2. 通过客户端命令**klist**查询当前认证用户：

```
[root@10-10-144-2 client]# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: test@HADOOP.COM

Valid starting     Expires            Service principal
01/25/17 11:06:48  01/26/17 11:06:45  krbtgt/HADOOP.COM@HADOOP.COM
```

如上例中当前认证用户为**test**。

3. 通过命令**id**查询用户组信息。

```
[root@10-10-144-2 client]# id test
uid=20032(test) gid=10001(hadoop) groups=10001(hadoop),9998(ficommon),10003(kafka)
```

解决办法

MRS Manager界面操作：

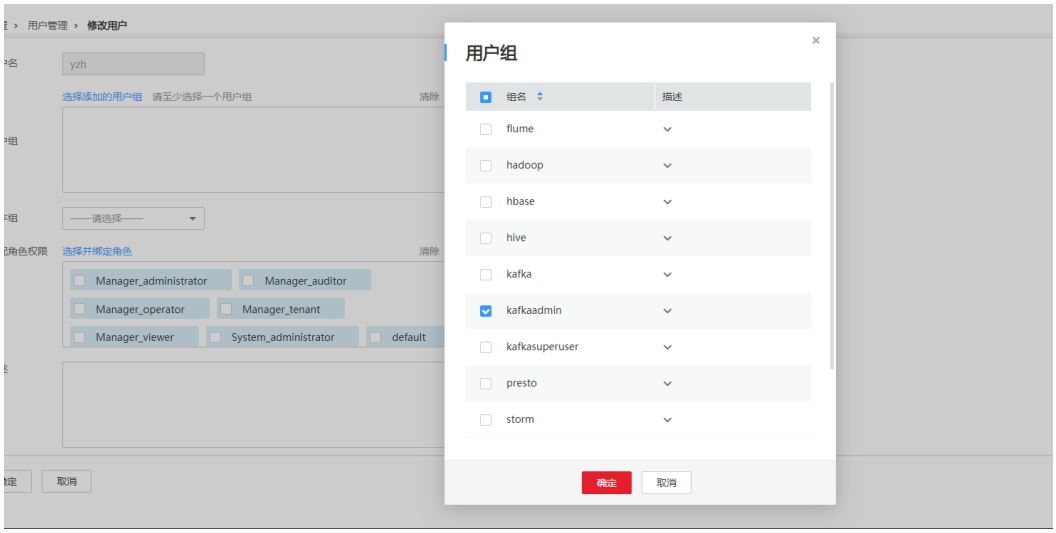
步骤1 登录MRS Manager。

步骤2 选择“系统设置 > 用户管理”。

步骤3 在操作用户对应的“操作”列，单击“修改”。

步骤4 为用户加入**kafkaadmin**组。

图 16-15 为用户加入 kafkaadmin 组



步骤5 通过命令**id**查询用户组信息。

```
[root@10-10-144-2 client]# id test
uid=20032(test) gid=10001(hadoop) groups=10001(hadoop),9998(ficommon),10002(kafkaadmin),
10003(kafka)
```

----结束

FusionInsight Manager界面操作：

步骤1 登录FusionInsight Manager。

步骤2 选择“系统 > 权限 > 用户”。

步骤3 在使用的用户所在行的单击“修改”。

步骤4 为用户添加**kafkaadmin**组。单击“确定”完成修改操作。

步骤5 通过命令**id**查询用户组信息。

```
[root@10-10-144-2 client]# id test
uid=20032(test) gid=10001(hadoop) groups=10001(hadoop),9998(ficommon),10002(kafkaadmin),
10003(kafka)
```

----结束

16.19 执行 Kafka Topic 设置 ACL 操作失败，提示“NoAuthException”

问题背景与现象

在使用Kafka客户端命令设置Topic ACL权限时，发现Topic无法被设置。

```
kafka-acls.sh --authorizer-properties zookeeper.connect=10.5.144.2:2181/kafka  --topic topic_acl --producer
--add --allow-principal User:test_acl
```

提示错误NoAuthException: KeeperErrorCode = NoAuth for /kafka-acl-changes/acl_changes_0000000002。

具体如下：

```
Error while executing ACL command: org.apache.zookeeper.KeeperException$NoAuthException:
KeeperErrorCode = NoAuth for /kafka-acl-changes/acl_changes_0000000002
org.I0Itec.zkclient.exception.ZkException: org.apache.zookeeper.KeeperException$NoAuthException:
KeeperErrorCode = NoAuth for /kafka-acl-changes/acl_changes_0000000002
at org.I0Itec.zkclient.exception.ZkException.create(ZkException.java:68)
at org.I0Itec.zkclient.ZkClient.retryUntilConnected(ZkClient.java:995)
at org.I0Itec.zkclient.ZkClient.delete(ZkClient.java:1038)
at kafka.utils.ZkUtils.deletePath(ZkUtils.scala:499)
at kafka.common.ZkNodeChangeNotificationListener$$anonfun$purgeObsoleteNotifications
$1.apply(ZkNodeChangeNotificationListener.scala:118)
at kafka.common.ZkNodeChangeNotificationListener$$anonfun$purgeObsoleteNotifications
$1.apply(ZkNodeChangeNotificationListener.scala:112)
at scala.collection.mutable.ResizableArray$class.foreach(ResizableArray.scala:59)
at scala.collection.mutable.ArrayBuffer.foreach(ArrayBuffer.scala:47)
at
kafka.common.ZkNodeChangeNotificationListener.purgeObsoleteNotifications(ZkNodeChangeNotificationLis
tener.scala:112)
at kafka.common.ZkNodeChangeNotificationListener.kafka$common$ZkNodeChangeNotificationListener$
$processNotifications(ZkNodeChangeNotificationListener.scala:97)
at
kafka.common.ZkNodeChangeNotificationListener.processAllNotifications(ZkNodeChangeNotificationListener
.scala:77)
at kafka.common.ZkNodeChangeNotificationListener.init(ZkNodeChangeNotificationListener.scala:65)
at kafka.security.auth.SimpleAclAuthorizer.configure(SimpleAclAuthorizer.scala:136)
at kafka.admin.AclCommand$.withAuthorizer(AclCommand.scala:73)
at kafka.admin.AclCommand$.addAcl(AclCommand.scala:80)
at kafka.admin.AclCommand$.main(AclCommand.scala:48)
at kafka.admin.AclCommand.main(AclCommand.scala)
Caused by: org.apache.zookeeper.KeeperException$NoAuthException: KeeperErrorCode = NoAuth for /kafka-
acl-changes/acl_changes_0000000002
at org.apache.zookeeper.KeeperException.create(KeeperException.java:117)
at org.apache.zookeeper.KeeperException.create(KeeperException.java:51)
at org.apache.zookeeper.ZooKeeper.delete(ZooKeeper.java:1416)
at org.I0Itec.zkclient.ZkConnection.delete(ZkConnection.java:104)
at org.I0Itec.zkclient.ZkClient$11.call(ZkClient.java:1042)
at org.I0Itec.zkclient.ZkClient.retryUntilConnected(ZkClient.java:985)
```

可能原因

用户不属于**kafkaadmin**组，Kafka提供安全访问接口，**kafkaamdin**组用户才可以进行设置操作。

原因分析

1. 使用客户端命令，打印NoAuthException异常。

2. 通过客户端命令**klist**查询当前认证用户：

```
[root@10-10-144-2 client]# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: test@HADOOP.COM

Valid starting     Expires            Service principal
01/25/17 11:06:48  01/26/17 11:06:45  krbtgt/HADOOP.COM@HADOOP.COM
```

如上例中当前认证用户为**test**。

3. 通过命令**id**查询用户组信息。

```
[root@10-10-144-2 client]# id test
uid=20032(test) gid=10001(hadoop) groups=10001(hadoop),9998(ficommon),10003(kafka)
```

解决办法

MRS Manager界面操作：

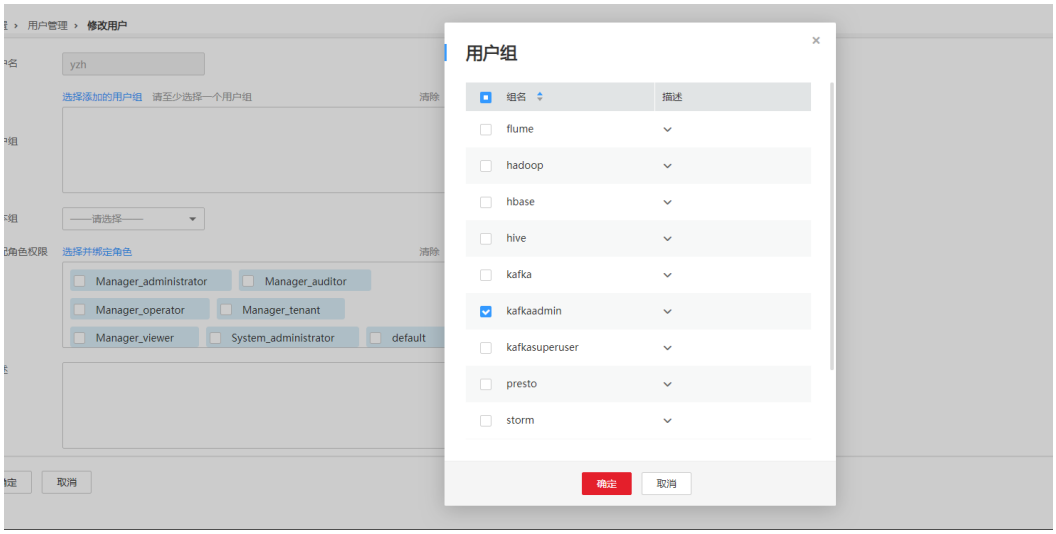
步骤1 登录MRS Manager。

步骤2 选择“系统设置 > 用户管理”。

步骤3 在操作用户对应的“操作”列，单击“修改”。

步骤4 为用户加入**kafkaadmin**组。

图 16-16 修改用户组



步骤5 通过命令**id**查询用户组信息。

```
[root@host1 client]# id test
uid=20032(test) gid=10001(hadoop) groups=10001(hadoop),9998(ficommon),10002(kafkaadmin),
10003(kafka)
```

----结束

FusionInsight Manager界面操作：

步骤1 登录FusionInsight Manager。

步骤2 选择“系统 > 权限 > 用户”。

步骤3 在使用的用户所在行的单击“修改”。

步骤4 为用户添加**kafkaadmin**组。单击“确定”完成修改操作。

步骤5 通过命令**id**查询用户组信息。

```
[root@10-10-144-2 client]# id test
uid=20032(test) gid=10001(hadoop) groups=10001(hadoop),9998(ficommon),10002(kafkaadmin),
10003(kafka)
```

----结束

16.20 执行 Kafka Topic 创建操作时发现无法创建提示“NoNode for /brokers/ids”

问题背景与现象

在使用Kafka客户端命令创建Topic时，发现Topic无法被创建。

```
kafka-topics.sh --create  --replication-factor 1 --partitions 2 --topic test --zookeeper
192.168.234.231:2181
```

提示错误NoNodeException: KeeperErrorCode = NoNode for /brokers/ids。

具体如下：

```
Error while executing topic command : org.apache.zookeeper.KeeperException$NoNodeException:
KeeperErrorCode = NoNode for /brokers/ids
[2017-09-17 16:35:28,520] ERROR org.I0Itec.zkclient.exception.ZkNoNodeException:
org.apache.zookeeper.KeeperException$NoNodeException: KeeperErrorCode = NoNode for /brokers/ids
    at org.I0Itec.zkclient.exception.ZkException.create(ZkException.java:47)
    at org.I0Itec.zkclient.ZkClient.retryUntilConnected(ZkClient.java:995)
    at org.I0Itec.zkclient.ZkClient.getChildren(ZkClient.java:675)
    at org.I0Itec.zkclient.ZkClient.getChildren(ZkClient.java:671)
    at kafka.utils.ZkUtils.getChildren(ZkUtils.scala:541)
    at kafka.utils.ZkUtils.getSortedBrokerList(ZkUtils.scala:176)
    at kafka.admin.AdminUtils$.createTopic(AdminUtils.scala:235)
    at kafka.admin.TopicCommand$.createTopic(TopicCommand.scala:105)
    at kafka.admin.TopicCommand$.main(TopicCommand.scala:60)
    at kafka.admin.TopicCommand.main(TopicCommand.scala)
Caused by: org.apache.zookeeper.KeeperException$NoNodeException: KeeperErrorCode = NoNode for /
brokers/ids
    at org.apache.zookeeper.KeeperException.create(KeeperException.java:115)
    at org.apache.zookeeper.KeeperException.create(KeeperException.java:51)
    at org.apache.zookeeper.ZooKeeper.getChildren(ZooKeeper.java:2256)
    at org.apache.zookeeper.ZooKeeper.getChildren(ZooKeeper.java:2284)
    at org.I0Itec.zkclient.ZkConnection.getChildren(ZkConnection.java:114)
    at org.I0Itec.zkclient.ZkClient$4.call(ZkClient.java:678)
    at org.I0Itec.zkclient.ZkClient$4.call(ZkClient.java:675)
    at org.I0Itec.zkclient.ZkClient.retryUntilConnected(ZkClient.java:985)
    ... 8 more
 (kafka.admin.TopicCommand$)
```

可能原因

- Kafka服务处于停止状态。
- 客户端命令中ZooKeeper地址参数配置错误。

原因分析

1. 使用客户端命令，打印NoNodeException异常。

```
Error while executing topic command : org.apache.zookeeper.KeeperException$NoNodeException:
KeeperErrorCode = NoNode for /brokers/ids
[2017-09-17 16:35:28,520] ERROR org.I0Itec.zkclient.exception.ZkNoNodeException:
org.apache.zookeeper.KeeperException$NoNodeException: KeeperErrorCode = NoNode for /brokers/ids
    at org.I0Itec.zkclient.exception.ZkException.create(ZkException.java:47)
```

```
    at org.I0Itec.zkclient.ZkClient.retryUntilConnected(ZkClient.java:995)
    at org.I0Itec.zkclient.ZkClient.getChildren(ZkClient.java:675)
    at org.I0Itec.zkclient.ZkClient.getChildren(ZkClient.java:671)
    at kafka.utils.ZkUtils.getChildren(ZkUtils.scala:541)
    at kafka.utils.ZkUtils.getSortedBrokerList(ZkUtils.scala:176)
    at kafka.admin.AdminUtils$.createTopic(AdminUtils.scala:235)
    at kafka.admin.TopicCommand$.createTopic(TopicCommand.scala:105)
    at kafka.admin.TopicCommand$.main(TopicCommand.scala:60)
    at kafka.admin.TopicCommand.main(TopicCommand.scala)
```

2. 通过Manager查看Kafka服务是否处于正常状态。
3. 检查客户端命令中ZooKeeper地址是否正确，访问ZooKeeper上所存放的Kafka信息，其路径（Znode）应该加上/kafka，发现配置中缺少/kafka：

```
[root@10-10-144-2 client]#
kafka-topics.sh --create  --replication-factor 1 --partitions 2 --topic test --zookeeper
192.168.234.231:2181
```

解决办法

步骤1 保证Kafka服务处于正常状态。

步骤2 创建命令中ZooKeeper地址信息需要添加/kafka。

```
[root@10-10-144-2 client]#
kafka-topics.sh --create  --replication-factor 1 --partitions 2 --topic test --zookeeper
192.168.234.231:2181/kafka
```

----结束

16.21 执行 Kakfa Topic 创建操作时发现无法创建提示“replication factor larger than available brokers”

问题背景与现象

在使用Kafka客户端命令创建Topic时，发现Topic无法被创建。

```
kafka-topics.sh --create  --replication-factor 2 --partitions 2 --topic test --zookeeper
192.168.234.231:2181
```

提示错误replication factor larger than available brokers。

具体如下：

```
Error while executing topic command : replication factor: 2 larger than available brokers: 0
[2017-09-17 16:44:12,396] ERROR kafka.admin.AdminOperationException: replication factor: 2 larger than
available brokers: 0
    at kafka.admin.AdminUtils$.assignReplicasToBrokers(AdminUtils.scala:117)
    at kafka.admin.AdminUtils$.createTopic(AdminUtils.scala:403)
    at kafka.admin.TopicCommand$.createTopic(TopicCommand.scala:110)
    at kafka.admin.TopicCommand$.main(TopicCommand.scala:61)
    at kafka.admin.TopicCommand.main(TopicCommand.scala)
 (kafka.admin.TopicCommand$)
```

可能原因

- Kafka服务处于停止状态。
- Kafka服务当前可用Broker小于设置的replication-factor。
- 客户端命令中ZooKeeper地址参数配置错误。

原因分析

1. 使用客户端命令，打印replication factor larger than available brokers异常。
```
Error while executing topic command : replication factor: 2 larger than available brokers: 0
[2017-09-17 16:44:12,396] ERROR kafka.admin.AdminOperationException: replication factor: 2 larger
than available brokers: 0
    at kafka.admin.AdminUtils$.assignReplicasToBrokers(AdminUtils.scala:117)
    at kafka.admin.AdminUtils$.createTopic(AdminUtils.scala:403)
    at kafka.admin.TopicCommand$.createTopic(TopicCommand.scala:110)
    at kafka.admin.TopicCommand$.main(TopicCommand.scala:61)
    at kafka.admin.TopicCommand.main(TopicCommand.scala)
 (kafka.admin.TopicCommand$)
```
2. 通过Manager参看Kafka服务是否处于正常状态，当前可用Broker是否小于设置的replication-factor。
3. 检查客户端命令中ZooKeeper地址是否正确，访问ZooKeeper上所存放的Kafka信息，其路径（Znode）应该加上/kafka，发现配置中缺少/kafka。
```
[root@10-10-144-2 client]#
kafka-topics.sh --create  --replication-factor 2 --partitions 2 --topic test --zookeeper
192.168.234.231:2181
```

解决办法

步骤1 保证Kafka服务处于正常状态，且可用Broker不小于设置的replication-factor。

步骤2 创建命令中ZooKeeper地址信息需要添加/kafka。
```
[root@10-10-144-2 client]#
kafka-topics.sh --create  --replication-factor 1 --partitions 2 --topic test --zookeeper
192.168.234.231:2181/kafka
```

----结束

16.22 Consumer 消费数据时存在重复消费现象

问题背景与现象

当数据量较大时会频繁地发生rebalance导致出现重复消费的情况，关键日志如下：

```
2018-05-12 10:58:42,561 | INFO | [kafka-request-handler-3] | [GroupCoordinator 2]: Preparing to restabilize
group DemoConsumer with old generation 118 | kafka.coordinator.GroupCoordinator (Logging.scala:68)
2018-05-12 10:58:43,245 | INFO | [kafka-request-handler-5] | [GroupCoordinator 2]: Stabilized group
DemoConsumer generation 119 | kafka.coordinator.GroupCoordinator (Logging.scala:68)
2018-05-12 10:58:43,560 | INFO | [kafka-request-handler-7] | [GroupCoordinator 2]: Assignment received
from leader for group DemoConsumer for generation 119 | kafka.coordinator.GroupCoordinator
(Logging.scala:68)
2018-05-12 10:59:13,562 | INFO | [executor-Heartbeat] | [GroupCoordinator 2]: Preparing to restabilize
group DemoConsumer with old generation 119 | kafka.coordinator.GroupCoordinator (Logging.scala:68)
2018-05-12 10:59:13,790 | INFO | [kafka-request-handler-3] | [GroupCoordinator 2]: Stabilized group
DemoConsumer generation 120 | kafka.coordinator.GroupCoordinator (Logging.scala:68)
2018-05-12 10:59:13,791 | INFO | [kafka-request-handler-0] | [GroupCoordinator 2]: Assignment received
from leader for group DemoConsumer for generation 120 | kafka.coordinator.GroupCoordinator
(Logging.scala:68)
2018-05-12 10:59:43,802 | INFO | [kafka-request-handler-2] | Rolled new log segment for
'__consumer_offsets-17' in 2 ms. | kafka.log.Log (Logging.scala:68)
2018-05-12 10:59:52,456 | INFO | [group-metadata-manager-0] | [Group Metadata Manager on Broker 2]:
Removed 0 expired offsets in 0 milliseconds. | kafka.coordinator.GroupMetadataManager (Logging.scala:68)
2018-05-12 11:00:49,772 | INFO | [kafka-scheduler-6] | Deleting segment 0 from log __consumer_offsets-17.
| kafka.log.Log (Logging.scala:68)
2018-05-12 11:00:49,773 | INFO | [kafka-scheduler-6] | Deleting index /srv/BigData/kafka/data4/kafka-logs/
__consumer_offsets-17/00000000000000000000.index.deleted | kafka.log.OffsetIndex (Logging.scala:68)
2018-05-12 11:00:49,773 | INFO | [kafka-scheduler-2] | Deleting segment 2147948547 from log
__consumer_offsets-17. | kafka.log.Log (Logging.scala:68)
2018-05-12 11:00:49,773 | INFO | [kafka-scheduler-4] | Deleting segment 4282404355 from log
```

```
__consumer_offsets-17. | kafka.log.Log (Logging.scala:68)
2018-05-12 11:00:49,775 | INFO | [kafka-scheduler-2] | Deleting index /srv/BigData/kafka/data4/kafka-logs/
__consumer_offsets-17/00000000002147948547.index.deleted | kafka.log.OffsetIndex (Logging.scala:68)
2018-05-12 11:00:49,775 | INFO | [kafka-scheduler-4] | Deleting index /srv/BigData/kafka/data4/kafka-logs/
__consumer_offsets-17/00000000004282404355.index.deleted | kafka.log.OffsetIndex (Logging.scala:68)
2018-05-12 11:00:50,533 | INFO | [kafka-scheduler-6] | Deleting segment 4283544095 from log
__consumer_offsets-17. | kafka.log.Log (Logging.scala:68)
2018-05-12 11:00:50,569 | INFO | [kafka-scheduler-6] | Deleting index /srv/BigData/kafka/data4/kafka-logs/
__consumer_offsets-17/00000000004283544095.index.deleted | kafka.log.OffsetIndex (Logging.scala:68)
2018-05-12 11:02:21,178 | INFO | [kafka-request-handler-2] | [GroupCoordinator 2]: Preparing to restabilize
group DemoConsumer with old generation 120 | kafka.coordinator.GroupCoordinator (Logging.scala:68)
2018-05-12 11:02:22,839 | INFO | [kafka-request-handler-4] | [GroupCoordinator 2]: Stabilized group
DemoConsumer generation 121 | kafka.coordinator.GroupCoordinator (Logging.scala:68)
2018-05-12 11:02:23,169 | INFO | [kafka-request-handler-1] | [GroupCoordinator 2]: Assignment received
from leader for group DemoConsumer for generation 121 | kafka.coordinator.GroupCoordinator
(Logging.scala:68)
2018-05-12 11:02:49,913 | INFO | [kafka-request-handler-6] | Rolled new log segment for
'__consumer_offsets-17' in 2 ms. | kafka.log.Log (Logging.scala:68)
```

其中Preparing to restabilize group DemoConsumer with old generation表示正在发生rebalance。

可能原因

参数设置不合理。

原因分析

原因：由于参数设置不当，数据量大时数据处理时间过长，导致频繁发生balance，此时offset无法正常提交，导致重复消费数据。

原理：每次poll的数据处理完后才提交offset，如果poll数据后的处理时长超出了session.timeout.ms的设置时长，此时发生rebalance导致本次消费失败，已经消费数据的offset无法正常提交，所以下次重新消费时还是在旧的offset消费数据，从而导致消费数据重复。

解决办法

建议用户在Manager页面调整以下服务参数：

- request.timeout.ms=100000
- session.timeout.ms=90000
- max.poll.records=50
- heartbeat.interval.ms=3000

其中：

- request.timeout.ms要比session.timeout.ms大10s。
- session.timeout.ms的大小设置要在服务端参数group.min.session.timeout.ms和group.max.session.timeout.ms之间。

以上参数可以根据实际情况进行适当的调整，特别是max.poll.records，这个参数是为了控制每次poll数据的records量，保证每次的处理时长尽量保持稳定。目的是为了保证poll数据以后的处理时间不要超过session.timeout.ms的时间。

参考信息

- poll之后的数据处理效率要高，不要阻塞下一次poll。

- poll方法和数据处理建议异步处理。

16.23 执行 Kafka Topic 创建操作时发现 Partition 的 Leader 显示为 none

问题背景与现象

在使用Kafka客户端命令创建Topic时，发现创建Topic Partition的Leader显示为none。

```
[root@10-10-144-2 client]#
kafka-topics.sh --create  --replication-factor 1 --partitions 2 --topic test --zookeeper 10.6.92.36:2181/kafka

Created topic "test".
```

```
[root@10-10-144-2 client]#
kafka-topics.sh --describe --zookeeper 10.6.92.36:2181/kafka

Topic:test      PartitionCount:2        ReplicationFactor:2     Configs:
        Topic: test     Partition: 0    Leader: none        Replicas: 2,3   Isr:
        Topic: test     Partition: 1    Leader: none        Replicas: 3,1   Isr:
```

可能原因

- Kafka服务处于停止状态。
- 找不到用户组信息。

原因分析

1. 查看Kafka服务状态：
   - MRS Manager界面操作：登录MRS Manager，依次选择“服务管理 > Kafka”，查看当前Kafka状态，发现状态为良好，且监控指标内容显示正确。
   - FusionInsight Manager界面操作：登录FusionInsight Manager，选择“集群 > 服务 > Kafka”，查看当前Kafka状态，发现状态为良好，且监控指标内容显示正确。
2. 在Kafka概览页面获取Controller节点信息。
3. 登录Controller所在节点，通过**cd /var/log/Bigdata/kafka/broker**命令进入节点日志目录，在state-change.log发现存在ZooKeeper权限异常，提示NoAuthException。

```
2018-05-31 09:20:42,436 | ERROR | [ZkClient-EventThread-34-10.6.92.36:24002,10.6.92.37:24002,10.6.92.38:24002/kafka] | Controller 4 epoch 6 initiated state change for partition [test,1] from NewPartition to OnlinePartition failed | state.change.logger (Logging.scala:103)

 org.I0Itec.zkclient.exception.ZkException: org.apache.zookeeper.KeeperException$NoAuthException: KeeperErrorCode = NoAuth for /brokers/topics/test/partitions
 at org.I0Itec.zkclient.exception.ZkException.create(ZkException.java:68)
 at org.I0Itec.zkclient.ZkClient.retryUntilConnected(ZkClient.java:1000)
 at org.I0Itec.zkclient.ZkClient.create(ZkClient.java:527)
 at org.I0Itec.zkclient.ZkClient.createPersistent(ZkClient.java:293)
```

4. 查看对应时间段ZooKeeper审计日志，权限异常。

```
2018-05-31 09:20:42,421 | ERROR | CommitProcWorkThread-1 | session=0xc3000007015d5a18 user=10.6.92.39,kafka/hadoop.hadoop.com@HADOOP.COM,kafka/
```

```
hadoop.hadoop.com@HADOOP.COM    ip=10.6.92.39    operation=create znode
target=ZooKeeperServer    znode=/kafka/brokers/topics/test/partitions/0/state    result=failure
2018-05-31 09:20:42,423 | ERROR | CommitProcWorkThread-1 | session=0xc3000007015d5a18
user=10.6.92.39,kafka/hadoop.hadoop.com@HADOOP.COM,kafka/
hadoop.hadoop.com@HADOOP.COM    ip=10.6.92.39    operation=create znode
target=ZooKeeperServer    znode=/kafka/brokers/topics/test/partitions/0    result=failure
2018-05-31 09:20:42,435 | ERROR | CommitProcWorkThread-1 | session=0xc3000007015d5a18
user=10.6.92.39,kafka/hadoop.hadoop.com@HADOOP.COM,kafka/
hadoop.hadoop.com@HADOOP.COM    ip=10.6.92.39    operation=create znode
target=ZooKeeperServer    znode=/kafka/brokers/topics/test/partitions    result=failure
2018-05-31 09:20:42,439 | ERROR | CommitProcWorkThread-1 | session=0xc3000007015d5a18
user=10.6.92.39,kafka/hadoop.hadoop.com@HADOOP.COM,kafka/
hadoop.hadoop.com@HADOOP.COM    ip=10.6.92.39    operation=create znode
target=ZooKeeperServer    znode=/kafka/brokers/topics/test/partitions/1/state    result=failure
2018-05-31 09:20:42,441 | ERROR | CommitProcWorkThread-1 | session=0xc3000007015d5a18
user=10.6.92.39,kafka/hadoop.hadoop.com@HADOOP.COM,kafka/
hadoop.hadoop.com@HADOOP.COM    ip=10.6.92.39    operation=create znode
target=ZooKeeperServer    znode=/kafka/brokers/topics/test/partitions/1    result=failure
2018-05-31 09:20:42,453 | ERROR | CommitProcWorkThread-1 | session=0xc3000007015d5a18
user=10.6.92.39,kafka/hadoop.hadoop.com@HADOOP.COM,kafka/
hadoop.hadoop.com@HADOOP.COM    ip=10.6.92.39    operation=create znode
target=ZooKeeperServer    znode=/kafka/brokers/topics/test/partitions    result=failure
```

5. 在ZooKeeper各个实例节点上执行**id -Gn kafka**命令，发现有一个节点无法查询用户组信息。

```
[root @bdpsit3ap03 ~]# id -Gn kafka
id:  kafka: No such user
[root @bdpsit3ap03 ~]#
```

6. MRS集群中的用户管理由LDAP服务管理提供，又依赖于操作系统的sssd或nscd服务，用户的建立到同步到sssd服务需要一定时间，如果此时用户没有生效，或者sssd版本存在bug的情况下，某些情况下在ZooKeeper节点会出现用户无效的情况，导致创建Topic异常。

解决办法

步骤1 重启sssd/nscd服务。

- Red Hat/Euler

  service sssd restart

- SUSE

  sevice nscd restart

步骤2 重启相关服务后，在节点通过**id** *username*命令查看相应用户信息是否已有效。

----结束

16.24 Kafka 安全使用说明

Kafka API 简单说明

- 新Producer API

  指org.apache.kafka.clients.producer.KafkaProducer中定义的接口，在使用“kafka-console-producer.sh”时，默认使用此API。

- 旧Producer API

  指kafka.producer.Producer中定义的接口，在使用“kafka-console-producer.sh”时，加“--old-producer”参数会调用此API。

- 新Consumer API

指org.apache.kafka.clients.consumer.KafkaConsumer中定义的接口，在使用“kafka-console-consumer.sh”时，加“--new-consumer”参数会调用此API。

- 旧Consumer API

  指kafka.consumer.ConsumerConnector中定义的接口，在使用“kafka-console-consumer.sh”时，默认使用此API。

说明

新Producer API和新Consumer API，在下文中统称为新API。

Kafka 访问协议说明

Kafka当前支持四种协议类型的访问：PLAINTEXT、SSL、SASL_PLAINTEXT、SASL_SSL。

Kafka服务启动时，默认会启动PLAINTEXT和SASL_PLAINTEXT两种协议类型的访问监测。可通过设置Kafka服务配置参数“ssl.mode.enable”为“true”，来启动SSL和SASL_SSL两种协议类型。

下表是四种协议类型的简单说明：

协议类型	说明	支持的API	默认端口
PLAINTEXT	支持无认证的明文访问	新API和旧API	9092
SASL_PLAINTEXT	支持Kerberos认证的明文访问	新API	21007
SSL	支持无认证的SSL加密访问	新API	9093
SASL_SSL	支持Kerberos认证的SSL加密访问	新API	21009

Topic 的 ACL 设置

Kafka支持安全访问，因此可以针对Topic进行ACL设置，从而控制不同的用户可以访问不同的Topic。Topic的权限信息，需要在Linux客户端上，使用“kafka-acls.sh”脚本进行查看和设置。

- 操作场景

  该任务指导Kafka管理员根据业务需求，为其他使用Kafka的系统用户授予相关Topic的特定权限。

  Kafka默认用户组信息表所示。

用户组名	描述
kafkaadmin	Kafka管理员用户组。添加入本组的用户，拥有所有Topic的创建，删除，授权及读写权限。
kafkasuperuser	添加入本组的用户，拥有所有Topic的读写权限。

用户组名	描述
kafka	Kafka普通用户组。添加入本组的用户，需要被kafkaadmin组用户授予特定Topic的读写权限，才能访问对应Topic。

- 前提条件
  a. 系统管理员已明确业务需求，并准备一个Kafka管理员用户（属于kafkaadmin组）。
  b. 已安装Kafka客户端。
- 操作步骤
  a. 以客户端安装用户，登录安装Kafka客户端的节点。
  b. 切换到Kafka客户端安装目录，例如“/opt/kafkaclient”。

     cd /opt/kafkaclient
  c. 执行以下命令，配置环境变量。

     source bigdata_env
  d. 执行以下命令，进行用户认证。（普通集群跳过此步骤）

     kinit *组件业务用户*
  e. 执行以下命令，切换到Kafka客户端安装目录。

     cd Kafka/kafka/bin
  f. 使用“kafka-acl.sh”进行用户授权常用命令如下：
     - 查看某Topic权限控制列表：

       ./kafka-acls.sh --authorizer-properties zookeeper.connect=<ZooKeeper集群业务IP:2181/kafka > --list --topic <Topic名称>
     - 添加给某用户Producer权限：

       ./kafka-acls.sh --authorizer-properties zookeeper.connect=<ZooKeeper集群业务IP:2181/kafka > --add --allow-principal User:<用户名> --producer --topic <Topic名称>
     - 删除某用户Producer权限：

       ./kafka-acls.sh --authorizer-properties zookeeper.connect=<ZooKeeper集群业务IP:2181/kafka > --remove --allow-principal User:<用户名> --producer --topic <Topic名称>
     - 添加给某用户Consumer权限：

       ./kafka-acls.sh --authorizer-properties zookeeper.connect=<ZooKeeper集群业务IP:2181/kafka > --add --allow-principal User:<用户名> --consumer --topic <Topic名称> --group <消费者组名称>
     - 删除某用户Consumer权限：

       ./kafka-acls.sh --authorizer-properties zookeeper.connect=<ZooKeeper集群业务IP:2181/kafka > --remove

--allow-principal User:<用户名> --consumer --topic <Topic名称> --group <消费者组名称>

说明

MRS 1.6.3及之前版本，无论集群是否开启Kerberos认证ZooKeeper默认端口号均为24002。MRS 1.6.3及之后版本，无论集群是否开启Kerberos认证ZooKeeper默认端口号均为2181。

针对不同的 Topic 访问场景，Kafka 新旧 API 使用说明

- 场景一：访问设置了ACL的Topic

使用的API	用户属组	客户端参数	服务端参数	访问的端口
新API	用户需满足以下条件之一即可： ● 属于系统管理员组 ● 属于kafkaadmin组 ● 属于kafkasuperuser组 ● 被授权的kafka组的用户	security.protocol=SASL_PLAINTEXT sasl.kerberos.service.name = kafka	-	sasl.port（默认21007）
		security.protocol=SASL_SSL sasl.kerberos.service.name = kafka	ssl.mode.enable配置为true	sasl-ssl.port（默认21009）
旧API	不涉及	不涉及	不涉及	不涉及

- 场景二：访问未设置ACL的Topic

使用的API	用户属组	客户端参数	服务端参数	访问的端口
新API	用户需满足以下条件之一： ● 属于系统管理员组 ● 属于kafkaadmin组 ● 属于kafkasuperuser组	security.protocol=SASL_PLAINTEXT sasl.kerberos.service.name = kafka	-	sasl.port（默认21007）
	用户属于kafka组		allow.everyone.if.no.acl.found配置为true	sasl.port（默认21007）

使用的API	用户属组	客户端参数	服务端参数	访问的端口
	用户需满足以下条件之一： ● 属于系统管理员组 ● 属于kafkaadmin组 ● kafkasuperuser组用户	security.protocol=SASL_SSLsasl.kerberos.service.name = kafka	ssl-enable配置为“true”	sasl-ssl.port（默认21009）
	用户属于kafka组		allow.everyone.if.no.acl.found配置为“true” ssl-enable配置为“true”	sasl-ssl.port（默认21009）
	-	security.protocol=PLAINTEXT	allow.everyone.if.no.acl.found配置为“true”	port（默认21005）
	-	security.protocol=SSL	allow.everyone.if.no.acl.found配置为“true” ssl-enable配置为“true”	ssl.port（默认21008）
旧Producer	-	-	allow.everyone.if.no.acl.found配置为“true”	port（默认21005）
旧Consumer	-	-	allow.everyone.if.no.acl.found配置为“true”	ZooKeeper服务端口：clientPort（默认24002）

16.25 如何获取 Kafka Consumer Offset 信息

问题背景与现象

使用Kafka Consumer消费数据时，如何获取Kafka Consumer Offset相关信息?

Kafka API 简单说明

- 新Producer API

  指org.apache.kafka.clients.producer.KafkaProducer中定义的接口，在使用“kafka-console-producer.sh”时，默认使用此API。

- 旧Producer API

  指kafka.producer.Producer中定义的接口，在使用“kafka-console-producer.sh”时，加“--old-producer”参数会调用此API。

- 新Consumer API

  指org.apache.kafka.clients.consumer.KafkaConsumer中定义的接口，在使用“kafka-console-consumer.sh”时，加“--new-consumer”参数会调用此API。

- 旧Consumer API

  指kafka.consumer.ConsumerConnector中定义的接口，在使用“kafka-console-consumer.sh”时，默认使用此API。

说明

新Producer API和新Consumer API，在下文中统称为新API。

处理步骤

旧Consumer API

- 前提条件
  a. 系统管理员已明确业务需求，并准备一个Kafka管理员用户（属于kafkaadmin组）。
  b. 已安装Kafka客户端。
- 操作步骤
  a. 以客户端安装用户，登录安装Kafka客户端的节点。
  b. 切换到Kafka客户端安装目录，例如“/opt/kafkaclient”。

     cd /opt/kafkaclient
  c. 执行以下命令，配置环境变量。

     source bigdata_env
  d. 执行以下命令，进行用户认证。（普通模式跳过此步骤）

     kinit *组件业务用户*
  e. 执行以下命令，切换到Kafka客户端安装目录。

     cd Kafka/kafka/bin
  f. 执行以下命令，获取consumer offset metric信息。

```
bin/kafka-consumer-groups.sh --zookeeper <zookeeper_host:port>/kafka --list
```

```
bin/kafka-consumer-groups.sh --zookeeper <zookeeper_host:port>/kafka --describe --group test-consumer-group
```

     例如：

```
kafka-consumer-groups.sh --zookeeper 192.168.100.100:2181/kafka --list
kafka-consumer-groups.sh --zookeeper 192.168.100.100:2181/kafka --describe --group test-consumer-group
```

新Consumer API

- 前提条件
  a. 系统管理员已明确业务需求，并准备一个Kafka管理员用户（属于kafkaadmin组）。
  b. 已安装Kafka客户端。

- 操作步骤
  a. 以客户端安装用户，登录安装Kafka客户端的节点。
  b. 切换到Kafka客户端安装目录，例如“/opt/client”。
     cd /opt/client
  c. 执行以下命令，配置环境变量。
     source bigdata_env
  d. 执行以下命令，进行用户认证。（普通模式跳过此步骤）
     kinit *组件业务用户*
  e. 执行以下命令，切换到Kafka客户端安装目录。
     cd Kafka/kafka/bin
  f. 执行以下命令，获取consumer offset metric信息。
     kafka-consumer-groups.sh --bootstrap-server <broker_host:port> --describe --group my-group
     例如：
     kafka-consumer-groups.sh --bootstrap-server 192.168.100.100:9092 --describe --group my-group

16.26 如何针对 Topic 进行配置增加和删除

问题背景与现象

使用Kafka过程中常常需要对特定Topic进行配置或者修改。

Topic级别可以修改参数列表：

```
cleanup.policy
compression.type
delete.retention.ms
file.delete.delay.ms
flush.messages
flush.ms
index.interval.bytes
max.message.bytes
min.cleanable.dirty.ratio
min.insync.replicas
preallocate
retention.bytes
retention.ms
segment.bytes
segment.index.bytes
segment.jitter.ms
segment.ms
unclean.leader.election.enable
```

处理步骤

- 前提条件
  已安装Kafka客户端。
- 操作步骤
  a. 以客户端安装用户，登录安装Kafka客户端的节点。
  b. 切换到Kafka客户端安装目录，例如“/opt/client”。

cd /opt/client

c. 执行以下命令，配置环境变量。

source bigdata_env

d. 执行以下命令，进行用户认证。（普通模式跳过此步骤）

kinit *组件业务用户*

e. 执行以下命令，切换到Kafka客户端安装目录。

cd Kafka/kafka/bin

f. 执行以下命令，针对Topic修改配置和删除配置。

kafka-topics.sh --alter --topic <topic_name> --zookeeper <zookeeper_host:port>/kafka --config <name=value>

kafka-topics.sh --alter --topic <topic_name> --zookeeper <zookeeper_host:port>/kafka --delete-config <name>

例如：

kafka-topics.sh --alter --topic test1 --zookeeper 192.168.100.100:2181/kafka --config retention.ms=86400000

kafka-topics.sh --alter --topic test1 --zookeeper 192.168.100.100:2181/kafka --delete-config retention.ms

g. 执行以下命令，查询topic信息。

kafka-topics.sh --describe -topic <topic_name> --zookeeper <zookeeper_host:port>/kafka

16.27 如何读取“__consumer_offsets”内部 topic 的内容

用户问题

Kafka如何将consumer消费的offset保存在内部topic“ __consumer_offsets”中?

处理步骤

步骤1 以客户端安装用户，登录安装Kafka客户端的节点。

步骤2 切换到Kafka客户端安装目录，例如“/opt/client”。

cd /opt/client

步骤3 执行以下命令，配置环境变量。

source bigdata_env

步骤4 执行以下命令，进行用户认证。（普通集群跳过此步骤）

kinit *组件业务用户*

步骤5 执行以下命令，切换到Kafka客户端安装目录。

cd Kafka/kafka/bin

步骤6 执行以下命令，获取consumer offset metric信息。

kafka-console-consumer.sh --topic __consumer_offsets --zookeeper <zk_host:port>/kafka --formatter

"kafka.coordinator.group.GroupMetadataManager\$OffsetsMessageFormatter" --consumer.config <property file> --from-beginning

其中<property file>配置文件中需要增加如下内容。

```
exclude.internal.topics = false
```

例如：

kafka-console-consumer.sh --topic __consumer_offsets --zookeeper 10.5.144.2:2181/kafka --formatter "kafka.coordinator.group.GroupMetadataManager\$OffsetsMessageFormatter" --consumer.config ../config/consumer.properties --from-beginning

```
[example-group1,test2,0]::[OffsetMetadata[0,NO_METADATA],CommitTime 1487121209218,ExpirationTime 148720760
9218]
[example-group1,test2,1]::[OffsetMetadata[0,NO_METADATA],CommitTime 1487121209218,ExpirationTime 148720760
9218]
[example-group1,test2,0]::[OffsetMetadata[2,NO_METADATA],CommitTime 1487121269208,ExpirationTime 148720760
9208]
[example-group1,test2,1]::[OffsetMetadata[1,NO_METADATA],CommitTime 1487121269208,ExpirationTime 148720760
```

----结束

16.28 如何配置 Kafka 客户端 shell 命令日志

用户问题

如何设置客户端shell命令的日志输出级别?

处理步骤

步骤1 以客户端安装用户，登录安装Kafka客户端的节点。

步骤2 切换到Kafka客户端安装目录，例如“/opt/client”。

cd /opt/client

步骤3 切换到Kafka客户端配置目录。

cd Kafka/kafka/config

步骤4 编辑tools-log4j.properties文件，将WARN修改为INFO，并保存。

```
log4j.rootLogger=WARN, stderr

log4j.appender.stderr=org.apache.log4j.ConsoleAppender
log4j.appender.stderr.layout=org.apache.log4j.PatternLayout
log4j.appender.stderr.layout.ConversionPattern=[%d] %p %m (%c)%n
log4j.appender.stderr.Target=System.err
```

```
log4j.rootLogger=INFO, stderr

log4j.appender.stderr=org.apache.log4j.ConsoleAppender
log4j.appender.stderr.layout=org.apache.log4j.PatternLayout
log4j.appender.stderr.layout.ConversionPattern=[%d] %p %m (%c)%n
log4j.appender.stderr.Target=System.err
```

步骤5 切换到Kafka客户端安装目录，例如“/opt/client”。

cd /opt/client

步骤6 执行以下命令，配置环境变量。

source bigdata_env

步骤7 执行以下命令，进行用户认证。（普通集群跳过此步骤）

kinit *组件业务用户*

步骤8 执行以下命令，切换到Kafka客户端安装目录。

cd Kafka/kafka/bin

步骤9 执行以下命令，获取topic信息，在控制台可见日志打印。

```
kafka-topics.sh --list --zookeeper 10.5.144.2:2181/kafka
[2017-02-17 14:34:27,005] INFO JAAS File name: /opt/client/Kafka/./kafka/config/jaas.conf
(org.I0Itec.zkclient.ZkClient)
[2017-02-17 14:34:27,007] INFO Starting ZkClient event thread. (org.I0Itec.zkclient.ZkEventThread)
...
Test
__consumer_offsets
counter
test
test2
test3
test4
```

----结束

16.29 如何获取 Topic 的分布信息

用户问题

如何获取Topic在Broker实例的分布信息?

前置操作

- 前提条件
  已安装Kafka、ZooKeeper客户端。
- 操作步骤
  a. 以客户端安装用户，登录安装Kafka客户端的节点。
  b. 切换到Kafka客户端安装目录，例如“/opt/client”。
     cd /opt/client
  c. 执行以下命令，配置环境变量。
     source bigdata_env
  d. 执行以下命令，进行用户认证。（普通集群跳过此步骤）
     kinit *组件业务用户*
  e. 执行以下命令，切换到Kafka客户端安装目录。
     cd Kafka/kafka/bin
  f. 执行Kafka相关命令，获取Topic分布信息和副本同步信息，观察返回结果。
     kafka-topics.sh --describe --zookeeper <zk_host:port/chroot>

例如：

```
[root@mgtdat-sh-3-01-3 client]#kafka-topics.sh --describe --zookeeper 10.149.0.90:2181/kafka
Topic:topic1    PartitionCount:2  ReplicationFactor:2     Configs:
Topic: topic1   Partition: 0 Leader: 26 Replicas: 23,25 Isr: 26
Topic: topic1   Partition: 1 Leader: 24 Replicas: 24,23 Isr: 24,23
```

其中，Replicas对应副本分布信息，Isr对应副本同步信息。

处理方法 1

1. 在ZooKeeper中查询Broker ID的对应关系。

   sh zkCli.sh -server <zk_host:port>

2. 在ZooKeeper客户端执行如下命令。

   ls /kafka/brokers/ids

   get /kafka/brokers/ids/<查询出的Broker id>

   例如：

```
[root@node-master1gAMQ kafka]# zkCli.sh -server node-master1gAMQ:2181
Connecting to node-master1gAMQ:2181
Welcome to ZooKeeper!
JLine support is enabled

WATCHER::

WatchedEvent state:SyncConnected type:None path:null
[zk: node-master1gAMQ:2181(CONNECTED) 0] ls /kafka/brokers/
ids      seqid     topics
[zk: node-master1gAMQ:2181(CONNECTED) 0] ls /kafka/brokers/ids
[1]
[zk: node-master1gAMQ:2181(CONNECTED) 1] get /kafka/brokers/ids/1
{"listener_security_protocol_map":{"PLAINTEXT":"PLAINTEXT","SSL":"SSL"},"endpoints":["PLAINTEXT://192.168.2.242:9092","SSL://192.168.2.242:9093"],"rack":"/default/rack0","jmx_port":21006,"host":"192.168.2.242","timestamp":"1580886124398","port":9092,"version":4}
[zk: node-master1gAMQ:2181(CONNECTED) 2]
```

处理方法 2

获取节点和broker ID的对应关系

kafka-broker-info.sh --zookeeper <zk_host:port/chroot>

例如：

```
[root@node-master1gAMQ kafka]# bin/kafka-broker-info.sh --zookeeper 192.168.2.70:2181/kafka
Broker_ID     IP_Address
--------------------------
 1           192.168.2.242
```

16.30 Kafka 高可靠使用说明

Kafka 高可靠、高可用说明

Kafka消息传输保障机制，可以通过配置不同的参数来保障消息传输，进而满足不同的性能和可靠性要求的应用场景。

- **Kafka高可用、高性能**

  如果业务需要保证高可用和高性能，可以采用参数：

参数	默认值	说明
unclean.leader.election.enable	true	是否允许不在ISR中的副本被选举为Leader，若设置为true，可能会造成数据丢失。
auto.leader.rebalance.enable	true	是否使用Leader自动均衡功能。 如果设为true，Controller会周期性的为所有节点的每个分区均衡Leader，将Leader分配给更优先的副本。
acks	1	需要Leader确认消息是否已经接收并认为已经处理完成。该参数会影响消息的可靠性和性能。 ● acks=0 ： 如果设置为0，Producer将不会等待服务端任何响应。消息将会被认为成功。 ● acks=1 ： 如果设置为1，当副本所在Leader确认数据已写入，但是其不会等待所有的副本完全写入即返回响应。在这种情况下，如果Leader确认后但是副本未同步完成时Leader异常，那么数据就会丢失。 ● acks=-1 ： 如果设置为-1（all），意味着等待所有的同步副本确认后才认为成功，配合min.insync.replicas可以确保多副本写入成功，只要有一个副本保持活跃状态，记录将不会丢失。 **说明** 该参数在kafka客户端配置文件中配置。
min.insync.replicas	1	当Producer设置acks为-1时，指定需要写入成功的副本的最小数目。

配置高可用、高性能的影响：

须知

配置高可用、高性能模式后，数据可靠性会降低。在磁盘故障、节点故障等场景下存在数据丢失风险。

- **Kafka高可靠性配置说明**

  如果业务需要保证数据高可靠性，可以采用相关参数：

参数	建议值	说明
unclean.leader.election.enable	false	是否允许不在ISR中的副本被选举为Leader。

参数	建议值	说明
acks	-1	Producer需要Leader确认消息是否已经接收并认为已经处理完成。 acks=-1需要表示等待在ISR列表的副本都确认接收到消息并处理完成才表示消息成功。配合min.insync.replicas可以确保多副本写入成功，只要有一个副本保持活跃状态，记录将不会丢失。 **说明** 该参数在kafka客户端配置文件中配置。
min.insync.replicas	2	当Producer设置acks为-1时，指定需要写入成功的副本的最小数目。 需要满足min.insync.replicas <= replication.factor。

配置高可靠性的影响：

- 性能降低：

  需要所有的ISR列表副本，且满足最小成功的副本数确认写入成功。这样会导致单条消息时延增加，客户端处理能力下降，具体性能以现场实际测试数据为准。

- 可用性降低：

  不允许不在ISR中的副本被选举为Leader。如果Leader下线时，其他副本均不在ISR列表中，那么该分区将保持不可用，直到Leader节点恢复。

  需要所有的ISR列表副本，且满足最小成功的副本数确认写入成功。当分区的一个副本所在节点故障时，无法满足最小成功的副本数，那么将会导致业务写入失败。

配置影响

请根据业务场景对可靠性和性能要求进行评估，采用合理参数配置。

说明

- 对于价值数据，两种场景下建议Kafka数据目录磁盘配置raid1或者raid5，从而提高单个磁盘故障情况下数据可靠性。
- 不同Producer API对应的acks参数名称不同
  - 新Producer API

    指**org.apache.kafka.clients.producer.KafkaProducer**中定义的接口，acks配置为acks。
  - 旧Producer API

    指**kafka.producer.Producer**中定义的接口，acks配置名称为request.required.acks。
- 参数配置项均为Topic级别可修改的参数，默认采用服务级配置。可针对不同Topic可靠性要求对Topic进行单独配置。

  例如，配置Topic名称为test的可靠性参数：

  kafka-topics.sh --zookeeper 192.168.1.205:2181/kafka --alter --topic test --config unclean.leader.election.enable=false --config min.insync.replicas=2

  其中192.168.1.205为ZooKeeper业务IP地址。
- 如果修改服务级配置需要重启Kafka，建议在变更窗口做服务级配置修改。

16.31 使用 Kafka Shell 命令无法操作 Kafka 集群

问题背景与现象

用户执行Kafka命令查看当前topic信息，但发现业务执行失败。

具体信息如下：

```
root@Slave2bin]#./kafka-topics.sh --describe --topic example-metric1 --zookeeper
192.119.147.231:2181,192.119.147.228:2181,192.119.147.227:2181
Error while executing topic command:org.apache.zookeeper.KeeperException
$AuthFailedException:KeeperErrorCode=AuthFailedfor/brokers/topics
[2016-09-2616:58:59,873]ERRORorg.I0Itec.zkclient.exception.ZkException:org.apache.zookeeper.KeeperExcept
ion$AuthFailedException:KeeperErrorCode=AuthFailedfor/brokers/topics
atorg.I0Itec.zkclient.exception.ZkException.create(ZkException.java:68)
atorg.I0Itec.zkclient.ZkClient.retryUntilConnected(ZkClient.java:995)
atorg.I0Itec.zkclient.ZkClient.getChildren(ZkClient.java:675)
atorg.I0Itec.zkclient.ZkClient.getChildren(ZkClient.java:671)
atkafka.utils.ZkUtils.getChildrenParentMayNotExist(ZkUtils.scala:548)
atkafka.utils.ZkUtils.getAllTopics(ZkUtils.scala:798)
atkafka.admin.TopicCommand$.getTopics(TopicCommand.scala:82)
atkafka.admin.TopicCommand$.describeTopic(TopicCommand.scala:183)
atkafka.admin.TopicCommand$.main(TopicCommand.scala:66)
atkafka.admin.TopicCommand.main(TopicCommand.scala)
Causedby:org.apache.zookeeper.KeeperException$AuthFailedException:KeeperErrorCode=AuthFailedfor/
brokers/topics
atorg.apache.zookeeper.KeeperException.create(KeeperException.java:127)
atorg.apache.zookeeper.KeeperException.create(KeeperException.java:51)
atorg.apache.zookeeper.ZooKeeper.getChildren(ZooKeeper.java:2256)
atorg.apache.zookeeper.ZooKeeper.getChildren(ZooKeeper.java:2284)
atorg.I0Itec.zkclient.ZkConnection.getChildren(ZkConnection.java:114)
atorg.I0Itec.zkclient.ZkClient$4.call(ZkClient.java:678)
atorg.I0Itec.zkclient.ZkClient$4.call(ZkClient.java:675)
atorg.I0Itec.zkclient.ZkClient.retryUntilConnected(ZkClient.java:985)
...8more
(kafka.admin.TopicCommand$)
```

原因分析

用户反馈已经排查了执行此命令的账号权限，此账号具有操作Kafka组件的最高权限，不应该仍然会有权限不足的问题。

经确认执行命令有问题，访问ZooKeeper上所存放的Kafka信息，其路径（Znode）应该加上/kafka，完整的查询命令应该是：

```
root@Slave2bin]#./kafka-topics.sh --describe --topic example-metric1 --zookeeper
192.168.147.231:2181,192.168.147.228:2181,192.168.147.227:2181/kafka
```

解决办法

属于命令操作有误，具体细节请参考**Kafka Shell命令**章节。

16.32 Kafka 生产者写入单条记录过长问题

问题背景与现象

用户在开发一个Kafka应用，作为一个生产者调用新接口（org.apache.kafka.clients.producer.*）往Kafka写数据，单条记录大小为1100055，超过了kafka配置文件server.properties中message.max.bytes=1000012。用户修改了Kafka服务配置中message.max.bytes大小为5242880，同时也将replica.fetch.max.bytes大小修改为5242880后，仍然无法成功。

报异常如下：

```
..........
14749 [Thread-0] INFO  com.xxx.bigdata.kafka.example.NewProducer  - The ExecutionException occured : {}.
java.util.concurrent.ExecutionException: org.apache.kafka.common.errors.RecordTooLargeException: The
message is 1100093 bytes when serialized which is larger than the maximum request size you have
configured with the max.request.size configuration.
at org.apache.kafka.clients.producer.KafkaProducer$FutureFailure.<init>(KafkaProducer.java:739)
at org.apache.kafka.clients.producer.KafkaProducer.doSend(KafkaProducer.java:483)
at org.apache.kafka.clients.producer.KafkaProducer.send(KafkaProducer.java:430)
at org.apache.kafka.clients.producer.KafkaProducer.send(KafkaProducer.java:353)
at com.xxx.bigdata.kafka.example.NewProducer.run(NewProducer.java:150)
Caused by: org.apache.kafka.common.errors.RecordTooLargeException: The message is **** bytes when
serialized which is larger than the maximum request size you have configured with the max.request.size
configuration.
.......
```

原因分析

经分析因为在写数据到Kafka时，Kafka客户端会先比较配置项“max.request.size ”值和本次写入数据大小，若写入数据大小超过此配置项“max.request.size ”的缺省值，则抛出上述异常。

解决办法

步骤1 在初始化Kafka生产者实例时，设置此配置项“max.request.size ”的值。

例如，参考本例，可以将此配置项设置为“5252880”：

```
    // 协议类型:当前支持配置为SASL_PLAINTEXT或者PLAINTEXT
    props.put(securityProtocol, kafkaProc.getValues(securityProtocol, "SASL_PLAINTEXT"));
    // 服务名
    props.put(saslKerberosServiceName, "kafka");
```

```
    props.put("max.request.size", "5252880");
    .......
```

----结束

16.33 Kakfa 消费者读取单条记录过长问题

问题背景与现象

和“Kafka生产者写入单条记录过长问题”相对应的，在写入数据后，用户开发一个应用，以消费者调用新接口（org.apache.kafka.clients.consumer.*）到Kafka上读取数据，但读取失败，报异常大致如下：

```
...........
1687 [KafkaConsumerExample] INFO  org.apache.kafka.clients.consumer.internals.AbstractCoordinator  - Successfully joined group DemoConsumer with generation 1
1688 [KafkaConsumerExample] INFO  org.apache.kafka.clients.consumer.internals.ConsumerCoordinator  - Setting newly assigned partitions [default-0, default-1, default-2] for group DemoConsumer
2053 [KafkaConsumerExample] ERROR com.xxx.bigdata.kafka.example.NewConsumer  - [KafkaConsumerExample], Error due to
org.apache.kafka.common.errors.RecordTooLargeException: There are some messages at [Partition=Offset]: {default-0=177} whose size is larger than the fetch size 1048576 and hence cannot be ever returned. Increase the fetch size on the client (using max.partition.fetch.bytes), or decrease the maximum message size the broker will allow (using message.max.bytes).
2059 [KafkaConsumerExample] INFO  com.xxx.bigdata.kafka.example.NewConsumer  - [KafkaConsumerExample], Stopped
.......
```

原因分析

经分析因为在读取数据时Kafka客户端会比较待读取数据大小和配置项“max.partition.fetch.bytes”值，若超过此配置项值，则抛出上述异常。

解决办法

步骤1 在初始化建立Kafka消费者实例时，设置此配置项“max.partition.fetch.bytes”的值。

例如，参考本例，可以将此配置项设置为“5252880”：

```
......
// 安全协议类型
props.put(securityProtocol, kafkaProc.getValues(securityProtocol, "SASL_PLAINTEXT"));
// 服务名
props.put(saslKerberosServiceName, "kafka");

props.put("max.partition.fetch.bytes","5252880");
......
```

----结束

16.34 Kafka 集群节点内多磁盘数据量占用高

用户问题

Kafka流式集群节点内有多块磁盘的使用量很高。当达到100%时就会造成kafka不可用如何处理？

问题现象

客户购买的MRS Kafka流式集群节点内有多块磁盘，由于分区不合理及业务原因导致某几个磁盘的使用量很高。当达到100%时就会造成kafka不可用。

原因分析

需要提前干预处理磁盘数据，全局的log.retention.hours修改需要重启服务。为了不断服，可以将数据量大的单个topic老化时间根据需要改短。

处理步骤

步骤1 登录Kafka集群的流式Core节点。

步骤2 执行**df -h**命令查看磁盘使用率。

```
[root@[redacted]K kafka-logs]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/[redacted]  [redacted]  [redacted]  75G  21% /[redacted]
devtm[redacted]  9G   0% /dev[redacted]
tmpfs[redacted]  9G   0% /dev[redacted]
tmpfs[redacted]  4G   7% /run[redacted]
tmpfs[redacted]  9G   0% /sys/fs/cgroup
/dev/[redacted]  94G  1% /srv/BigData/streaming/data1
tmpfs[redacted]  6G   0% /run[redacted]
/dev/[redacted]  2G   1% /srv/BigData/streaming/data2
/dev/[redacted]  2G   1% /srv/BigData/streaming/data3
tmpfs[redacted]  6G   0% /run/[redacted]
```

步骤3 通过kafka配置文件**/opt/Bigdata/***MRS_2.1.0/1_11_Broker***/etc/server.properties**中的配置项**log.dirs**获得数据存储目录。

其中配置文件路径请根据时间环境的集群版本修改，当磁盘有多块时，该配置项有多个，逗号间隔。

```
ssl.port = 9093
log.dirs = /srv/BigData/streaming/data1/kafka-logs,/srv/BigData/streaming/data2/kafka-logs,/srv/BigData/streaming/data3/kafka-lo
gs
controlled.shutdown.enable = true
compression.type = producer
max.connections.per.ip.overrides =
log.message.timestamp.difference.max.ms = 9223372036854775807
sasl.kerberos.kinit.cmd = /opt/Bigdata/MRS_2.1.0/install/FusionInsight-kerberos-1.15.2/kerberos/bin/kinit
log.cleaner.io.max.bytes.per.second = 1.79769313486231570E308
auto.leader.rebalance.enable = true
leader.imbalance.check.interval.seconds = 300
log.cleaner.min.cleanable.ratio = 0.5
```

步骤4 使用**cd**命令进入使用率较高的磁盘对应的**步骤3**中获取的数据存储目录下。

步骤5 使用**du -sh ***命令打印出当前topic的名称及大小。

```
[root@[redacted] kafka-logs]# du -sh *
0       [redacted]offset-checkpoint
12K     [redacted]
4.0K    [redacted]t-offset-checkpoint
4.0K    [redacted]perties
4.0K    [redacted]-point-offset-checkpoint
4.0K    [redacted]ion-offset-checkpoint
20K     [redacted]
20K     [redacted]t-0
20K     [redacted]t-1
20K     [redacted]t-2
20K     [redacted]t-3
20K     [redacted]t-4
20K     [redacted]t-5
[root@node-str-coreetHK kafka-logs]# pwd
/srv/BigData/streaming/data1/kafka-logs
```

```
[root[redacted] kafka-logs]# du -sh *
0       [redacted]r-offset-checkpoint
4.0K    [redacted]art-offset-checkpoint
4.0K    [redacted]roperties
4.0K    [redacted]ry-point-offset-checkpoint
4.0K    [redacted]ation-offset-checkpoint
4.0K    [redacted]-0
4.0K    [redacted]-1
4.0K    [redacted]-2
4.0K    [redacted]-6
4.0K    [redacted]-8
[root@[redacted] kafka-logs]# pwd
/srv/BigData/streaming/data2/kafka-logs
```

```
[root@[redacted] kafka-logs]# du -sh *
0       [redacted]-offset-checkpoint
4.0K    [redacted]art-offset-checkpoint
4.0K    [redacted]operties
4.0K    [redacted]y-point-offset-checkpoint
4.0K    [redacted]tion-offset-checkpoint
4.0K    [redacted]-3
4.0K    [redacted]-4
4.0K    [redacted]-5
4.0K    [redacted]-7
4.0K    [redacted]-9
[root@node-str-coreetHK kafka-logs]# pwd
/srv/BigData/streaming/data3/kafka-logs
```

步骤6 由于Kafka的全局的数据保留时间默认是7天。部分topic由于业务写入量大，而这些topic的分区正好在上面使用量高的磁盘上，因此导致磁盘使用率较高。

- 可以通过修改全局数据的保留期为较短时间来释放磁盘空间，该方式需要重启Kafka服务才能生效，可能会影响业务运行。具体请参见步骤7。
- 可以单独将topic的数据保留期改为较短时间来释放磁盘空间，该方式无需重启Kafka服务即可生效。具体请参见步骤8。

步骤7 登录Manager页面，在Kafka的服务配置页面，切换为“全部配置”并搜索“log.retention.hours”配置项，该值默认为7天，请根据需要进行修改。

步骤8 可以单独将这些磁盘上的topic的数据老化时间修改为较短时间来解决该问题。

1. 查看topic数据过期时间。

   bin/kafka-topics.sh --describe --zookeeper *<ZooKeeper集群业务IP>*:**2181/kafka --topic kktest**

```
[root@[redacted]Nw kafka]# bin/kafka-topics.sh --describe --zookeeper 192.168.201.175:2181/kafka --topic kktest
Topic:kktest    PartitionCount:1        ReplicationFactor:1     Configs:retention.ms=1000000
        Topic: kktest   Partition: 0    Leader: 1       Replicas: 1     Isr: 1
```

2. 设置topic数据过期时间，其中**--topic**表示具体topic名称，retention.ms=具体的数据过期时间，单位是毫秒。

   kafka-topics.sh --zookeeper *<ZooKeeper集群业务IP>*:**2181/kafka --alter --topic kktest --config retention.ms=1000000**

```
[root@node-master1nTNw kafka]# kafka-topics.sh --zookeeper 192.168.201.175:2181/kafka --alter --topic kktest --config retentio
ms=1000000
WARNING: Altering topic configuration from this script has been deprecated and may be removed in future releases.
         Going forward, please use kafka-configs.sh for this functionality
Updated config for topic "kktest".
```

   设置数据过期时间之后会可能不会立刻执行，删除操作在参数**log.retention.check.interval.ms**所规定时间之后开始执行删，可以通过查看kafka的server.log检索是否有delete字段来判断删除操作是否生效，有delete字段则表示已经生效，也可以通过执行**df -h**命令查看磁盘的数据量占用情况判断设置是否生效。

```
log.retention.check.interval.ms = 300000
```

----结束

16.35 Kafka 连接 ZooKeeper 客户端时发生断链

问题现象

Kafka客户端连接ZooKeeper客户端时发生断链。

客户端日志出现报错“ Received event :disconnected::none: with disconnected Writer Zookeeper”。

原因分析

在Kafka客户端连接Zookeeper客户端过程中，fsync操作时间过长，导致超时。

处理步骤

步骤1 登录Zookeeper客户端所在节点。

cd *客户端安装目录*

source bigdata_env

kinit *组件业务用户*（未开启Kerberos认证集群跳过此步骤）

步骤2 执行以下命令修改文件。

vim *客户端安装目录*/**zookeeper/conf/zoo.cfg**

调大文件中参数“tickTime”，“syncLimit”的值。

例如调整“tickTime”为“3000”，“syncLimit”为“7”。

步骤3 保存文件。

----结束

17 使用 Oozie

17.1 并发提交大量 oozie 任务时，任务一直没有运行

问题现象

并发提交大量oozie任务的时候，任务一直没有运行。

原因分析

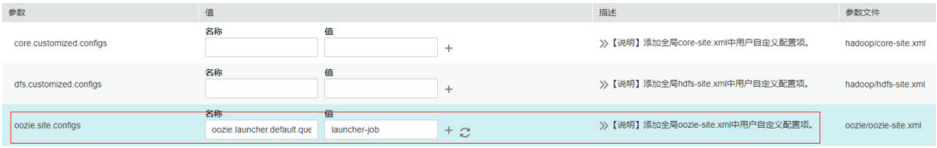
Oozie提交任务会先启动一个oozie-launcher，然后由oozie-launcher提交真正的作业运行。默认情况下launcher和真实作业会在同一个队列中。

当并发提交大量oozie任务的时候就有可能出现启动了一堆oozie-launcher，将队列的资源耗完，而没有更多资源启动真实作业，最终导致任务一直没有运行。

处理步骤

步骤1 参考“添加租户”章节新建一个队列给oozie使用，也可以直接使用创建MRS集群时生成的launcher-job队列。

步骤2 在Manager页面选择“集群 > 服务 > Oozie > 配置”，搜索参数“oozie.site.configs”，在值列添加名称“oozie.launcher.default.queue”，值为队列名称，例如“launcher-job”。



----结束

17.2 Oozie 调度 HiveSQL 作业报错处理

问题现象

MRS 3.x集群Oozie调度Hive作业任务失败，查看日志HiveSQL，实际上是运行成功的，但Yarn任务总体失败，报错如下：

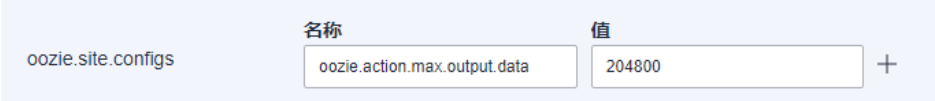
```
java.io.ioException:output.properties data exceeds its limit [2048]
```

原因分析

由于HiveSQL脚本中一次提交的作业量太大，其中包含的信息超过Oozie Launcher一次容许的最大值2KB（2048 Bytes），需要调大默认的输出参数值，调整该参数不会对集群性能造成影响。

处理步骤

步骤1 登录Manager页面，选择“集群 > 服务 > Oozie > 配置 > 全部配置 > 自定义”，在“oozie-site.xml”的配置项中添加参数“oozie.action.max.output.data”，值为“204800”，如下所示：



步骤2 添加完后，保存并重启Oozie服务。

步骤3 重新执行Oozie调度作业。

----结束

17.3 在 MRS 集群外客户端提交不了 Oozie 任务或两个小时才提交成功

问题现象

在MRS集群外客户端提交不了Oozie任务，或者两个小时才提交成功。

原因分析

通过后台日志看到一些Java安全随机数的日志，在JDK中，SecureRandom算法底层依赖操作系统提供的随机数据；在Linux中，与之相关的是“/dev/random”和“/dev/urandom”。当熵池为空时，来自“/dev/random”的读操作将被阻塞，直到熵池收集到足够的环境噪声数据，导致提交任务慢或者失败。

处理步骤

步骤1 在集群外客户端所在节点执行以下命令安装rng-tools工具：

yum install rng-tools

步骤2 执行以下命令启动rng-tools：

systemctl start rngd

若还需实现当所在节点重启后，rng-tools工具自动运行，则执行以下命令：

systemctl enable rngd

步骤3 重新运行Oozie任务。

----结束

18 使用 Presto

18.1 配置 sql-standard-with-group 创建 schema 失败报 Access Denied

用户问题

配置sql-standard-with-group创建schema失败，报Access Denied的错误。

问题现象

报错如下：

```
CREATE SCHEMA hive.sf2 WITH (location = 'obs://obs-zy1234/sf2');Query 20200224_031203_00002_g6gzy
failed: Access Denied: Cannot create schema sf2
```

原因分析

Presto中创建schema需要Hive的管理者权限，当前用户权限不足。

处理步骤

MRS Manager界面操作：

- 方法一：
  a. 登录MRS Manager页面，选择“系统设置 > 用户管理”。
  b. 在对应用户所在行的“操作”列，单击“修改”。
  c. 单击“选择并绑定角色”，为用户添加System_administrator的权限。
  d. 单击“确定”完成修改。
- 方法二：
  a. 登录MRS Manager页面，选择“系统设置 > 角色管理”。
  b. 单击“添加角色”，并配置如下参数。
     - 角色名称：配置角色名称，例如hive_admin。
     - 权限：选择“Hive”，并勾选Hive Admin Privilege。

c. 单击“确定”保存角色。
d. 选择“系统设置 > 用户管理”。
e. 在对应用户所在行的“操作”列，单击“修改”。
f. 单击“选择并绑定角色”，为用户添加新创建的hive_admin的权限。
g. 单击“确定”完成修改。

FusionInsight Manager界面操作：

- 方法一：
  a. 登录FusionInsight Manager页面，选择“系统 > 权限 > 用户”。
  b. 在对应用户所在行的“操作”列，单击“修改”。
  c. 单击角色后的“添加”，为用户添加System_administrator的权限。
  d. 单击“确定”完成修改。
- 方法二：
  a. 登录FusionInsight Manager页面，选择“系统 > 权限 > 角色”。
  b. 单击“添加角色”，并配置如下参数。
     - 角色名称：配置角色名称，例如hive_admin。
     - 配置资源权限：选择“Hive”，并勾选“Hive管理员权限”。
  c. 单击“确定”保存角色。
  d. 选择“系统 > 权限 > 用户”。
  e. 在对应用户所在行的“操作”列，单击“修改”。
  f. 单击角色后的“添加”，为用户添加新创建的hive_admin的权限。
  g. 单击“确定”完成修改。

18.2 Presto 的 coordinator 进程无法正常启动

用户问题

Presto的coordinator未知原因被终止，或者Presto的coordinator进程无法正常启动。

问题现象

Presto的coordinator无法正常启动，Manager页面上显示presto coordinator进程正常启动且状态正常，但查看后台日志coordinator进程未真正启动，只有如下日志：

```
2020-06-18T18:17:02.872+0800    INFO    main    Bootstrap       node.config-spec
 null                                                                   null
2020-06-18T18:17:02.872+0800    INFO    main    Bootstrap       node.environment
 null                                                                   mrs
2020-06-18T18:17:02.872+0800    INFO    main    Bootstrap       node.internal-address-source
 IP                                                                     IP
2020-06-18T18:17:02.872+0800    INFO    main    Bootstrap       node.location
 null                                                                   null
2020-06-18T18:17:02.872+0800    INFO    main    Bootstrap       node.bind-ip
 null                                                                   [redacted]
2020-06-18T18:17:02.872+0800    INFO    main    Bootstrap       node.external-address
 null                                                                   null
2020-06-18T18:17:02.872+0800    INFO    main    Bootstrap       node.id
 null                                                                   Coordinator-[redacted]
2020-06-18T18:17:02.872+0800    INFO    main    Bootstrap       node.internal-address
 null                                                                   [redacted]
2020-06-18T18:17:02.872+0800    INFO    main    Bootstrap       node.pool
 general                                                                general
2020-06-18T18:20:00.014+0800    INFO    main    io.airlift.log.Logging  Disabling stderr output
2020-06-18T18:20:01.777+0800    INFO    main    Bootstrap       PROPERTY
 DEFAULT                                                                RUNTIME
                 DESCRIPTION
2020-06-18T18:20:01.777+0800    INFO    main    Bootstrap       event.max-output-stage-size
 16MB                                                                   16MB
2020-06-18T18:20:01.777+0800    INFO    main    Bootstrap       query.client.timeout
 5.00m                                                                  5.00m
2020-06-18T18:20:01.777+0800    INFO    main    Bootstrap       query.initial-hash-partitions
 100                                                                    32
2020-06-18T18:20:01.777+0800    INFO    main    Bootstrap       query-manager.initialization-required-workers
 1                                                                      1
                 Minimum number of workers that must be available before the cluster will accept queries
2020-06-18T18:20:01.777+0800    INFO    main    Bootstrap       query-manager.initialization-timeout
 5.00m                                                                  5.00m
                 After this time, the cluster will accept queries even if the minimum required workers are not available
2020-06-18T18:20:01.778+0800    INFO    main    Bootstrap       query.max-concurrent-queries
 1000                                                                   1000
2020-06-18T18:20:01.778+0800    INFO    main    Bootstrap       query.max-history
 100                                                                    100
@@@
                                                                                                409945,73-83   62%
```

Presto的coordinator未真正启动即被终止了，不再打印其他日志，查看Presto的其他日志也未发现原因。

原因分析

Presto的健康检查脚本的端口检查逻辑中未做好端口的区分。

处理步骤

步骤1 使用工具分别登录集群的Master节点执行如下操作。

步骤2 执行如下命令编辑文件。

vim /opt/Bigdata/MRS_xxx/install/FusionInsight-Presto-*/ha/module/harm/plugin/script/pcd.sh

该文件中的第31行修改为“http_port_exists=$(netstat -apn | awk '{print $4, $6}' | grep :${HTTP_PORT} | grep LISTEN | wc -l)”。

```
25
26 check_status()
27 {
28         proc_exists=$(ps -ef | grep com.facebook.presto.server.PrestoServer | grep -v grep | wc -l)
29         param="-u $PRESTO_SERVER/v1/cluster"
30         if [[ ${proc_exists} == 1 ]]; then
31             http_port_exists=$(netstat -apn | awk '{print $4, $6}' | grep :${HTTP_PORT} | grep LISTEN | wc -l)
32
33             if [[ ${http_port_exists} == 1 ]] ;then
34                 log ${PCD_LOG_FILE} "INFO" "return [ normal ]"
35                 return 0
36             else
37                 log ${PCD_LOG_FILE} "ERROR" "HTTP PORT does not exist, return [ abnormal ]"
38                 return 2
39             fi
40         else
41             log ${PCD_LOG_FILE} "INFO" " coordinator process not exists, return [ abnormal ]"
42             return 2
43         fi
44 }
45
```

步骤3 保存如上修改，再在Manager页面上选择“服务管理 > Presto > 实例”，重启Coordinator进程。

----结束

18.3 Presto 查询 Kudu 表时系统报错找不到表

用户问题

使用Presto查询Kudu表报错。

问题现象

使用Presto查询Kudu表，报表找不到的错误：

```
presto:default> show tables;
           Table

 impala::default.kudu_taobao
 impala::default.kudu_tt
 impala::default.kudutest
(3 rows)

Query 20210201_030636_00026_95mzd, FINISHED, 4 nodes
Splits: 53 total, 53 done (100.00%)
0:00 [3 rows, 125B] [18 rows/s, 766B/s]

presto:default> select count(*) from kudu.default.kudu_taobao;
Query 20210201_030653_00027_95mzd failed: line 1:22: Table kudu.default.kudu_taobao does not exist
select count(*) from kudu.default.kudu_taobao

presto:default> select count(*) from kudu_taobao;
Query 20210201_030939_00028_95mzd failed: line 1:22: Table kudu.default.kudu_taobao does not exist
select count(*) from kudu_taobao

presto:default>
```

后台报错：

```
2021-02-01T15:08:13.850+0800    INFO    query-execution-10      io.prestosql.event.QueryMonitor TIMELINE: Query 20210201_070813_00007_6x
9q9 :: Transaction:[72fad2d9-8480-4435-ac8d-ac2a93bf181d] :: elapsed 71ms :: planning 15ms :: waiting 0ms :: scheduling 56ms :: running
1ms :: finishing 0ms :: begin 2021-02-01T15:08:13.730+08:00 :: end 2021-02-01T15:08:13.801+08:00
2021-02-01T15:14:17.487+0800    INFO    query-execution-19      io.prestosql.event.QueryMonitor TIMELINE: Query 20210201_071417_00008_6x
9q9 :: Transaction:[0104571a-3ec6-4013-b7c6-8219916a07ba] :: elapsed 369ms :: planning 167ms :: waiting 3ms :: scheduling 45ms :: runnin
g 85ms :: finishing 72ms :: begin 2021-02-01T15:14:17.095+08:00 :: end 2021-02-01T15:14:17.464+08:00
2021-02-01T15:15:11.127+0800    INFO    query-execution-20      io.prestosql.event.QueryMonitor TIMELINE: Query 20210201_071510_00009_6x
9q9 :: Transaction:[0dc00e86-5500-4932-a528-693cb4ad0054] :: elapsed 282ms :: planning 115ms :: waiting 0ms :: scheduling 30ms :: runnin
g 55ms :: finishing 82ms :: begin 2021-02-01T15:15:10.830+08:00 :: end 2021-02-01T15:15:11.112+08:00
2021-02-01T15:15:14.006+0800    ERROR   remote-task-callback-4 io.prestosql.execution.StageStateMachine        Stage 20210201_071513_00
010_6x9q9.1 failed
java.lang.IllegalArgumentException: No page sink provider for catalog 'kudu'
        at com.google.common.base.Preconditions.checkArgument(Preconditions.java:216)
        at io.prestosql.split.PageSinkManager.providerFor(PageSinkManager.java:67)
        at io.prestosql.split.PageSinkManager.createPageSink(PageSinkManager.java:61)
        at io.prestosql.operator.TableWriterOperator$TableWriterOperatorFactory.createPageSink(TableWriterOperator.java:114)
        at io.prestosql.operator.TableWriterOperator$TableWriterOperatorFactory.createOperator(TableWriterOperator.java:105)
        at io.prestosql.operator.DriverFactory.createDriver(DriverFactory.java:114)
        at io.prestosql.execution.SqlTaskExecution$DriverSplitRunnerFactory.createDriver(SqlTaskExecution.java:941)
        at io.prestosql.execution.SqlTaskExecution$DriverSplitRunner.processFor(SqlTaskExecution.java:1069)
        at io.prestosql.execution.executor.PrioritizedSplitRunner.process(PrioritizedSplitRunner.java:163)
        at io.prestosql.execution.executor.TaskExecutor$TaskRunner.run(TaskExecutor.java:484)
        at io.prestosql.$gen.Presto_EI_PrestoSQL_Kernel_Component_0_3_308_0100_B001_13_gbcb0afe_dirty____20210201_070255_1.run(Unknown S
ource)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        at java.lang.Thread.run(Thread.java:748)
```

原因分析

在实际的运行节点（worker实例所在节点）没有Kudu相关配置。

处理步骤

步骤1 在集群Presto所有的worker实例节点上添加配置文件kudu.properties。

配置文件保存路径：/opt/Bigdata/MRS_xxx/1_x_Worker/etc/catalog/ (请根据集群实际版本修改路径)

配置文件内容：

```
connector.name=kudu
kudu.client.master-addresses=KuduMasterIP1:port,KuduMasterIP2:port,KuduMasterIP3:port
```

说明

- KuduMaster节点IP和端口请根据实际情况填写。
- 为配置文件添加和文件保存路径下其他文件一致的文件权限、属组。

步骤2 修改完成之后，请在集群详情页面选择“组件管理 > Kudu”，单击“更多 > 重启服务”。

----结束

18.4 Presto 查询 Hive 表无数据

用户问题

使用Presto查询Hive表无数据。

问题现象

通过Tez引擎执行union相关语句写入的数据，Presto无法查询。

原因分析

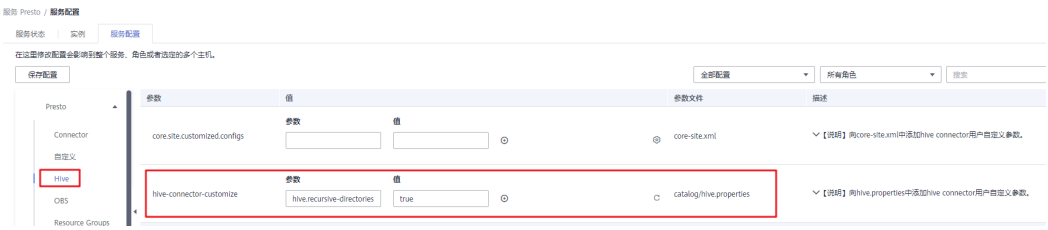
由于Hive使用Tez引擎在执行union语句时，生成的输出文件会保存在HIVE_UNION_SUBDIR目录中，而Presto默认不读取子目录下的文件，所以没有读取到HIVE_UNION_SUBDIR目录下的数据。

处理步骤

步骤1 在集群详情页面选择“组件管理 > Presto > 服务配置”。

步骤2 切换“基础配置”为全部配置“。

步骤3 在左侧导航处选择“Presto > Hive”，在“catalog/hive.properties”文件中增加“hive.recursive-directories”参数，值为“true”。



步骤4 单击“保存配置”并勾选“重新启动受影响的服务或实例”。

----结束

18.5 MRS Presto 查询报错“The node may have crashed or be under too much load”

用户问题

用户presto查询语句报错：The node may have crashed or be under too much load，具体如下图所示：

图 18-1 报错信息

问题分析

1. 登录集群Master节点，进入“/var/log/Bigdata”目录下的presto日志进行查找。
2. 根据报错信息在到对应节点去查看presto worker实例的进程日志。

   日志显示：

   ```
   java.lang.OutOfMemoryError: Java heap space
   ```

3. 根据报错预判发生了oom导致查询报错。

处理步骤

步骤1 在Manager页面查找work的JVM参数（xmx）。

图 18-2 Presto JVM 参数配置

步骤2 将xmx后面的1024改成2048，然后保存配置。

步骤3 重启Presto服务，重新尝试查询。

----结束

18.6 MRS 集群如何使用公网访问 Presto

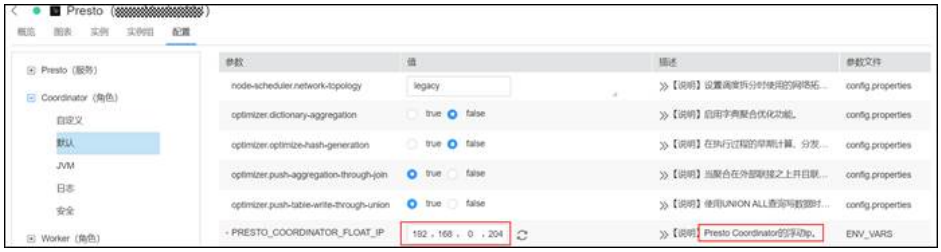
用户问题

用户调测Presto JDBC样例代码，需要使用公网访问Presto。

问题分析

登录MRS Manager上查看Presto全部配置，Coordinator角色配置：

PRESTO_COORDINATOR_FLOAT_IP = *内网IP地址*



登录Coordinator角色所在主实例节点确认该地址为网卡：eth0:PRESTO。

```
[root@node-master2nOXb ~]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.244  netmask 255.255.255.0  broadcast 192.168.0.255
        ether fa:16:3e:d5:a0:e8  txqueuelen 4000  (Ethernet)
        RX packets 30206419  bytes 11537075854 (10.7 GiB)
        RX errors 0  dropped 0  overruns 0  frame 7052
        TX packets 26042255  bytes 10821677809 (10.0 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0:DBS: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.88  netmask 255.255.255.0  broadcast 192.168.0.255
        ether fa:16:3e:d5:a0:e8  txqueuelen 4000  (Ethernet)

eth0:HUE: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.160  netmask 255.255.255.0  broadcast 192.168.0.255
        ether fa:16:3e:d5:a0:e8  txqueuelen 4000  (Ethernet)

eth0:JHS: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.38  netmask 255.255.255.0  broadcast 192.168.0.255
        ether fa:16:3e:d5:a0:e8  txqueuelen 4000  (Ethernet)

eth0:PRESTO: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.204  netmask 255.255.255.0  broadcast 192.168.0.255
        ether fa:16:3e:d5:a0:e8  txqueuelen 4000  (Ethernet)
```

Presto服务端口：非安全集群为7520；安全集群为7521；

因此公网访问需要给对应的浮动网卡绑定公网地址，然后填写正确的JDBC URL即可

URL格式：

jdbc:presto://*example_ip:7520*/**Catalog/schema**

jdbc:presto://*example_ip:7521*/**Catalog/schema**

处理步骤

步骤1 在MRS Manager管理页面找到Presto组件，记录Presto的内网浮动IP地址。

步骤2 在VPC上创建一个弹性公网IP地址，并在VPC控制台上找到MRS集群的子网，然后找到Presto的浮动IP，并给该浮动IP绑定弹性公网IP。

步骤3 在MRS服务集群的安全组中放通源地址到MRS服务Presto的端口访问，然后测试（以非安全集群为例，catalog为hive）。

```
[root@node-masterimnma ~]# presto_cli.sh --server http://119.3.164.55:7520 --catalog hive --schema bigdata
--server http://119.3.164.55:7520 --catalog hive --schema bigdata
presto:bigdata> select * from eimrs;
 id | name | worker | happies | age
----+------+--------+---------+-----
 0  |      |        |         | 30
 1  |      |        |         | 28
 2  |      |        |         | 32
(3 rows)

Query 20210731_050300_00007_p4y4b, FINISHED, 1 node
Splits: 19 total, 19 done (100.00%)
0:00 [3 rows, 96B] [10 rows/s, 326B/s]

presto:bigdata>
```

----结束

19 使用 Spark

19.1 运行 Spark 应用时修改 split 值报错

用户问题

在Spark应用下修改split值时报错。

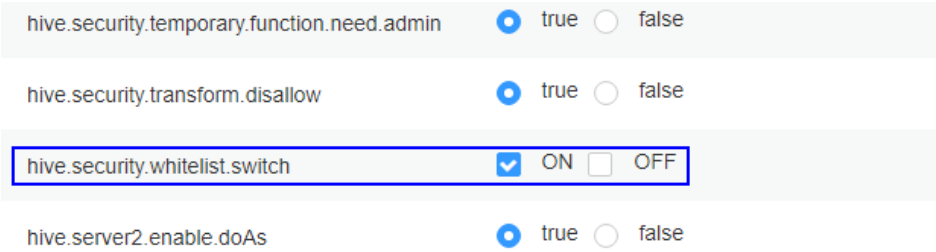
问题现象

用户需要通过修改一个split最大值来实现多个mapper，从而达到提速的目的，但是执行**set**命令修改Hive的配置时报错。

```
0: jdbc:hive2://192.168.1.18:21066/> set mapred.max.split.size=1000000;
Error: Error while processing statement: Cannot nodify mapred.max.split.size at runtime. It is not in list of
params that are allowed to be modified at runtime( state=42000,code=1)
```

原因分析

- 在安全模式下配置白名单启停参数**hive.security.whitelist.switch**时，需要运行的参数必须在**hive.security.authorization.sqlstd.confwhitelist**中配置。



- 默认白名单中没有包含**mapred.max.split.size**参数，所以运行的时候会提示不允许。

处理步骤

步骤1 进入Hive服务配置页面：

- MRS 3.*x*之前版本，单击集群名称，登录集群详情页面，选择“组件管理 > Hive > 服务配置”，单击“基础配置”下拉菜单，选择“全部配置”。

说明

如果集群详情页面没有“组件管理”页签，请先完成IAM用户同步（在集群详情页的“概览”页签，单击“IAM用户同步”右侧的“同步”进行IAM用户同步）。

- MRS 3.x及后续版本，登录FusionInsight Manager，然后选择“集群 > 服务 > Hive > 配置 > 全部配置”。

步骤2 搜索**hive.security.authorization.sqlstd.confwhitelist.append**，把**mapred.max.split.size**加进**hive.security.authorization.sqlstd.confwhitelist.append**中，详细信息可参考**从零开始使用Hive**。

步骤3 修改完成后，保存配置，重启Hive组件。

步骤4 重新在Hive Beeline命令行中执行**set mapred.max.split.size=1000000;**。

----结束

19.2 提交 Spark 任务时提示参数格式错误

问题现象

用户在使用Spark组件时，提交集群任务运行失败，提示参数格式错误。

```
[omm@node-master1-qxvMQ spark]$
[omm@node-master1-qxvMQ spark]$
[omm@node-master1-qxvMQ spark]$
[omm@node-master1-qxvMQ spark]$ ./bin/spark-submit --class cn.interf.Test  --master yarn-client /opt/client/Spark/spark1-1.0-SNAPSHOT.jar;
Error: Unrecognized option: --class cn.interf.Test  --master

Java HotSpot(TM) 64-Bit Server VM warning: Cannot open file <LOG_DIR>/gc.log due to No such file or directory

Usage: spark-submit [options] <app jar | python file> [app arguments]
Usage: spark-submit --kill [submission ID] --master [spark://...]
Usage: spark-submit --status [submission ID] --master [spark://...]
Usage: spark-submit run-example [options] example-class [example args]

Options:
  --master MASTER_URL         spark://host:port, mesos://host:port, yarn, or local.
  --deploy-mode DEPLOY_MODE   Whether to launch the driver program locally ("client") or
                              on one of the worker machines inside the cluster ("cluster")
                              (Default: client).
  --class CLASS_NAME          Your application's main class (for Java / Scala apps).
  --name NAME                 A name of your application.
  --jars JARS                 Comma-separated list of local jars to include on the driver
```

原因分析

- 执行的命令包含了非法字符。
- 上传的jar包属主属组异常。

处理步骤

步骤1 检查用户执行命令**./bin/spark-submit --class cn.interf.Test --master yarn-client** *客户端安装目录/Spark/spark1-1.0-SNAPSHOT.jar;*，排查是否引入了非法字符。

步骤2 如果是，修改非法字符，重新执行命令。

步骤3 重新执行命令后，发生其他错误，查看该jar包的属主属组信息，发现全为**root**。

步骤4 修改jar包的属主属组为**omm:wheel**，重新执行成功。

----结束

19.3 磁盘容量不足导致 Spark、Hive 和 Yarn 服务不可用

用户问题

磁盘容量不足产生致命告警，Spark、Hive和Yarn服务不可用。

问题现象

用户创建的集群显示磁盘容量不足，产生致命告警，导致Spark、Hive和Yarn服务不可用。

原因分析

由于集群磁盘容量不足，会影响到HDFS的数据写入，HDFS磁盘空间使用率超过阈值，因此导致HDFS服务异常。HDFS服务异常则会导致Spark、Hive和Yarn服务不可用。

根据该集群出现磁盘容量不足产生Spark、Hive和Yarn服务不可用的报警，扩容磁盘后不再告警，可以判断是磁盘容量不足引起HDFS功能故障所导致。

处理步骤

针对磁盘容量不足产生的告警处理步骤，请参考**ALM-12017 磁盘容量不足**。

参考信息

HDFS磁盘空间使用率超过阈值处理步骤，请参考**ALM-14001 HDFS磁盘空间使用率超过阈值**。

19.4 引入 jar 包不正确导致 Spark 任务无法运行

问题现象

执行Spark任务，任务无法运行。

原因分析

执行Spark任务时，引入的jar包不正确，导致Spark任务运行失败。

处理步骤

步骤1 登录任意Master节点。

步骤2 执行**cd /opt/Bigdata/MRS_*/install/FusionInsight-Spark-*/spark/examples/jars**命令， 查看样例程序的jar包。

说明

jar包名最多为1023字符，不能包含;|&>,<'$特殊字符，且不可为空或全空格。

步骤3 执行程序可存储于HDFS或者OBS中，不同的文件系统对应的路径存在差异。

- OBS存储路径：以“obs://”开头。示例：obs://wordcount/program/hadoop-mapreduce-examples-2.7.x.jar
- HDFS存储路径：以“/user”开头。Spark Script需要以“.sql”结尾，MapReduce和Spark需要以“.jar”结尾。sql、jar不区分大小写。

----结束

19.5 Spark 任务由于内存不够或提交作业时未添加 Jar 包，作业卡住

用户问题

Spark提交作业内存不足或提交作业时未添加Jar包导致任务长时间处于pending状态或者运行中内存溢出。

问题现象

使用Spark提交作业后，长期卡住不动。反复运行作业后报错，内容如下：

```
Exception in thread "main" org.apache.spark.SparkException: Job aborted due to stage failure:
Aborting TaskSet 3.0 because task 0 (partition 0) cannot run anywhere due to node and executor blacklist.
Blacklisting behavior can be configured via spark.blacklist.*.
```

原因分析

内存不足或提交作业时未添加Jar包，导致Spark提交的作业任务长时间处于pending状态。

处理步骤

步骤1 检查提交作业时是否添加Jar包。

- 是，执行**步骤2**。
- 否，添加Jar包，执行作业正常，**操作结束**。如果执行作业任务长时间处于pending状态，执行**步骤2**。

步骤2 登录MRS Console页面，在现有集群中，选择集群名称，在“节点信息”页面，查看当前集群的节点规格。

步骤3 提高nodemanager进程所持有的集群资源。

MRS Manager界面操作：

1. 登录MRS Manager页面，选择“服务管理 > Yarn > 服务配置”。
2. 在“参数类别”中选择“全部配置”，然后在搜索框中搜索**yarn.nodemanager.resource.memory-mb**，查看该参数值。建议配置成节点物理内存总量的75%-90%。

FusionInsight Manager界面操作：

1. 登录FusionInsight Manager。选择“集群 > 服务 > Yarn”。

2. 单击“配置”，选择“全部配置”。然后在搜索框中搜索**yarn.nodemanager.resource.memory-mb**，查看该参数值。建议配置成节点物理内存总量的75%-90%。

步骤4 修改Spark的服务配置。

MRS Manager界面操作：

1. 登录MRS Manager页面，选择“服务管理”>“Spark” >“服务配置”。
2. 在“参数类别”中选择“全部配置”，然后在搜索框中搜索**spark.driver.memory**和**spark.executor.memory**

   根据作业的需要调大或者调小该值，具体以提交的Spark作业的复杂度和内存需要为参考（一般调大）。

FusionInsight Manager界面操作：

1. 登录FusionInsight Manager。选择“集群 > 服务 > Spark”。
2. 单击“配置”，选择“全部配置”。然后在搜索框中搜索**spark.driver.memory**和**spark.executor.memory**，根据作业的需要调大或者调小该值，具体以提交的Spark作业的复杂度和内存需要为参考（一般调大）。

说明

- 如果使用到SparkJDBC作业，搜索并修改**SPARK_EXECUTOR_MEMORY**和**SPARK_DRIVER_MEMORY**两个参数取值，具体以提交的Spark作业的复杂度和内存需要为参考（一般调大）。
- 如果对核数有要求，可以搜索并修改**spark.driver.cores**和**spark.executor.cores**的核数取值。

步骤5 Spark依赖内存做计算，如果以上还是不能满足任务的提交需要，建议扩容集群。

----结束

19.6 提交 Spark 任务时报错“ClassNotFoundException”

问题现象

运行Spark任务报找不到指定的类“ClassNotFoundException”。

详细报错内容如下：

```
Exception encountered | org.apache.spark.internal.Logging$class.logError(Logging.scala:91)
org.apache.hadoop.hbase.DoNotRetryIOException: java.lang.ClassNotFoundException:
org.apache.phoenix.filter.SingleCQKeyValueComparisonFilter
```

原因分析

用户配置的默认路径不正确。

处理步骤

步骤1 登录任意Master节点。

步骤2 修改Spark客户端目录下的配置文件。

执行**vim** *客户端安装目录*/**Spark/spark/conf/spark-defaults.conf**命令，打开spark-defaults.conf文件，设置“spark.executor.extraClassPath” 取值为“${PWD}/*”。

步骤3 重新尝试提交任务。

----结束

19.7 提交 Spark 任务时 Driver 端提示运行内存超限

问题背景与现象

运行内存超限导致提交Spark任务失败。

原因分析

在Driver日志中直接打印申请的executor memory超过集群限制。

```
... INFO Client: Verifying our application has not requested more than the maximum memory capability of the cluster (6144 MB per container)
... ERROR SparkContext: Error initializing SparkContext.
java.lang.IllegalArgumentException: Required executor memory (10240+1024 MB) is above the max threshold (6144 MB) of this cluster!
```

Spark任务提交至Yarn上面，运行task的executor使用的资源受yarn的管理。从报错信息可看出，用户申请启动executor时，指定10G的内存，超出了Yarn设置的每个container的最大内存的限制，导致任务无法启动。

解决办法

修改Yarn的配置，提高对container的限制。如可通过调整“yarn.scheduler.maximum-allocation-mb”参数的大小，可控制启动的executor的资源，修改之后要重启Yarn服务。

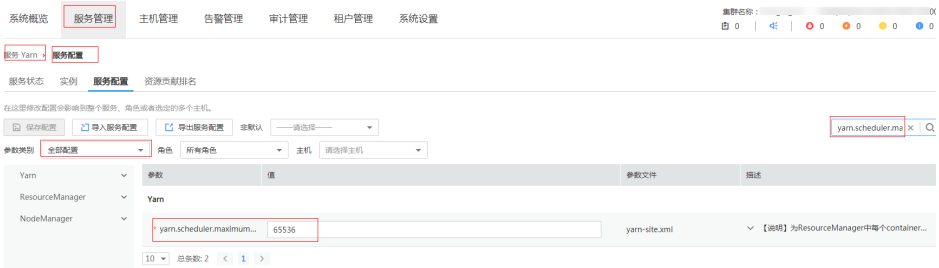
MRS Manager界面操作：

步骤1 登录MRS Manager页面。

步骤2 选择“服务管理 > Yarn > 服务配置”将“参数类别”修改为“全部配置”。

步骤3 在“搜索”栏输入“yarn.scheduler.maximum-allocation-mb”修改参数并保存重启服务。如下图所示：

图 19-1 修改 Yarn 服务参数



----结束

FusionInsight Manager界面操作：

步骤1 登录FusionInsight Manager页面。

步骤2 选择“集群 > 服务 > Yarn”，单击“配置”，选择“全部配置”。

步骤3 在“搜索”栏输入“yarn.scheduler.maximum-allocation-mb”修改参数并保存重启服务。

----结束

19.8 Yarn-cluster 模式下提交 Spark 任务报“Can't get the Kerberos realm”异常

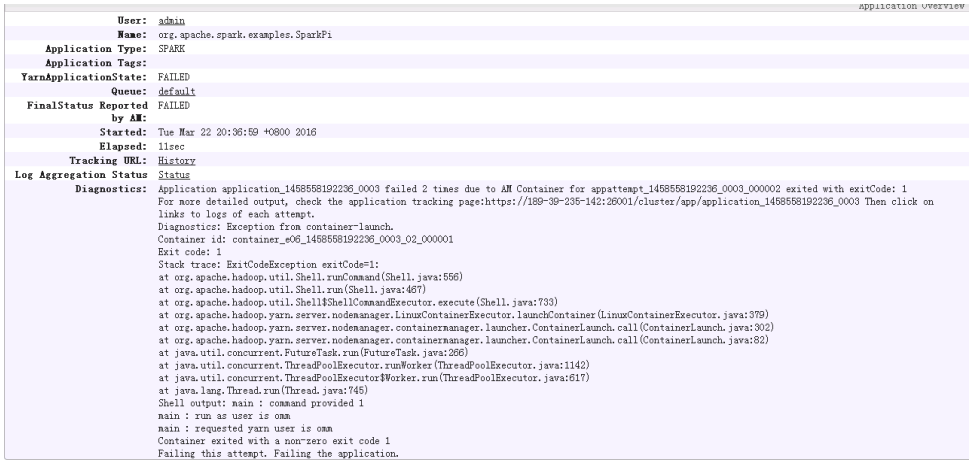
问题背景与现象

认证异常导致提交Spark任务失败，报“Can't get the Kerberos realm”异常。

原因分析

1. 在Driver端打印异常找不到连接HDFS的Token信息，报错如下：
```
16/03/22 20:37:10 WARN Client: Exception encountered while connecting to the server :
org.apache.hadoop.ipc.RemoteException(org.apache.hadoop.security.token.SecretManager
$InvalidToken): token (HDFS_DELEGATION_TOKEN token 192 for admin) can't be found in cache
16/03/22 20:37:10 WARN Client: Failed to cleanup staging dir .sparkStaging/
application_1458558192236_0003
org.apache.hadoop.ipc.RemoteException(org.apache.hadoop.security.token.SecretManager
$InvalidToken): token (HDFS_DELEGATION_TOKEN token 192 for admin) can't be found in cache
```
2. 在Yarn原生页面显示ApplicationMaster启动两次均失败，任务退出，如**图19-2**信息：

图 19-2 ApplicationMaster 启动失败



3. 查看ApplicationMaster日志看到如下异常信息：
```
Exception in thread "main" java.lang.ExceptionInInitializerError
Caused by: org.apache.spark.SparkException: Unable to load YARN support
Caused by: java.lang.IllegalArgumentException: Can't get Kerberos realm
Caused by: java.lang.reflect.InvocationTargetException
Caused by: KrbException: Cannot locate default realm
Caused by: KrbException: Generic error (description in e-text) (60) - Unable to locate Kerberos realm
org.apache.hadoop.hive.metastore.MetaStoreUtils.newInstance(MetaStoreUtils.java:1410)
... 86 more
Caused by: javax.jdo.JDOFatalInternalException: Unexpected exception caught.
NestedThrowables:java.lang.reflect.InvocationTargetException
... 110 more
```
4. 执行**./spark-submit --class** *yourclassname* **--master yarn-cluster /** *yourdependencyjars*任务以yarn-cluster模式提交任务，Driver端会在集群中启

动，由于加载的是客户端的spark.driver.extraJavaOptions，在集群节点上对应路径下找不到对应的kdc.conf文件，无法获取kerberos认证所需信息，导致ApplicationMaster启动失败。

解决办法

在客户端提交任务时，在命令行中配置自定义的spark.driver.extraJavaOptions参数这样任务运行时就不会自动加载客户端路径下“spark-defaults.conf”中的spark.driver.extraJavaOptions；或者在启动Spark任务时，通过“--conf”来指定Driver的配置，如下（此处spark.driver.extraJavaOptions“=”号后面的引号部分不能缺少）。

./spark-submit -class yourclassname --master yarn-cluster --conf spark.driver.extraJavaOptions="

-Dlog4j.configuration=file:/opt/client/Spark/spark/conf/log4j.properties -Djetty.version=x.y.z -Dzookeeper.server.principal=zookeeper/hadoop.794bbab6_9505_44cc_8515_b4eddc84e6c1.com -Djava.security.krb5.conf=/opt/client/KrbClient/kerberos/var/krb5kdc/krb5.conf -Djava.security.auth.login.config=/opt/client/Spark/spark/conf/jaas.conf -Dorg.xerial.snappy.tempdir=/opt/client/Spark/tmp -Dcarbon.properties.filepath=/opt/client/Spark/spark/conf/carbon.properties" ../yourdependencyjars

19.9 JDK 版本不匹配导致启动 spark-sql 和 spark-shell 失败

问题背景与现象

JDK版本不匹配导致客户端启动spark-sql、spark-shell失败。

原因分析

1. 在Driver端打印异常如下：
```
Exception Occurs: BadPadding 16/02/22 14:25:38 ERROR Schema: Failed initialising database. Unable to open a test connection to the given database. JDBC url = jdbc:postgresql://ip:port/sparkhivemeta, username = spark. Terminating connection pool (set lazyInit to true if you expect to start your database after your app).
```
2. SparkSQL任务使用时，需要访问DBService以获取元数据信息，在客户端需要解密密文来访问，在使用过程中，用户没有按照流程操作，没有执行配置环境变量操作，且在其客户端环境变量中存在默认的JDK版本，导致在执行解密过程中调用的解密程序执行解密异常，导致用户被锁。

解决办法

步骤1 使用**which java**命令查看默认的JAVA命令是否是客户端的JAVA。

步骤2 如果不是，请按正常的客户端执行流程。

source *${client_path}*/**bigdata_env**

kinit *用户名*，然后输入用户名对应的密码，启动任务即可。

----结束

19.10 Yarn-client 模式提交 Spark 任务时 ApplicationMaster 尝试启动两次失败

问题背景与现象

Yarn-client模式提交任务AppMaster尝试启动两次失败。

原因分析

1. Driver端异常:
```
16/05/11 18:10:56 INFO Client:
client token: N/A
diagnostics: Application application_1462441251516_0024 failed 2 times due to AM Container for
appattempt_1462441251516_0024_000002 exited with  exitCode: 10
For more detailed output, check the application tracking page:https://hdnode5:26001/cluster/app/
application_1462441251516_0024 Then click on links to logs of each attempt.
Diagnostics: Exception from container-launch.
Container id: container_1462441251516_0024_02_000001
```
2. 在ApplicationMaster日志中，异常如下:
```
2016-05-12 10:21:23,715 | ERROR | [main] | Failed to connect to driver at 192.168.30.57:23867,
retrying ... | org.apache.spark.Logging$class.logError(Logging.scala:75)
2016-05-12 10:21:24,817 | ERROR | [main] | Failed to connect to driver at 192.168.30.57:23867,
retrying ... | org.apache.spark.Logging$class.logError(Logging.scala:75)
2016-05-12 10:21:24,918 | ERROR | [main] | Uncaught exception:  | org.apache.spark.Logging
$class.logError(Logging.scala:96)
org.apache.spark.SparkException: Failed to connect to driver!
at org.apache.spark.deploy.yarn.ApplicationMaster.waitForSparkDriver(ApplicationMaster.scala:426)
at org.apache.spark.deploy.yarn.ApplicationMaster.runExecutorLauncher(ApplicationMaster.scala:292)
...
2016-05-12 10:21:24,925 | INFO  | [Thread-1] | Unregistering ApplicationMaster with FAILED (diag
message: Uncaught exception: org.apache.spark.SparkException: Failed to connect to driver!) |
org.apache.spark.Logging$class.logInfo(Logging.scala:59)
```
Spark-client模式任务Driver运行在客户端节点上(通常是集群外的某个节点)，启动时先在集群中启动AppMaster进程，进程启动后要向Driver进程注册信息，注册成功后，任务才能继续。从AppMaster日志中可以看出，无法连接至Driver，所以任务失败。

解决办法

步骤1 请检查Driver进程所在的IP是否可以**ping**通。

步骤2 启动一个Spark PI任务，会有类似如下打印信息。
```
16/05/11 18:07:20 INFO Remoting: Remoting started; listening on addresses :[akka.tcp://
sparkDriver@192.168.1.100:23662]
16/05/11 18:07:20 INFO Utils: Successfully started service 'sparkDriver' on port 23662.
```

步骤3 在该节点，也就是**步骤2**中示例的192.168.1.100上执行**netstat - anp | grep 23662**看下此端口是否打开，如下打印标明，相关端口是打开的。
```
tcp        0      0  ip:port    :::*                    LISTEN      107274/java
tcp        0      0  ip:port    ip:port                 ESTABLISHED 107274/java
```

步骤4 在AppMaster启动的节点执行**telnet 192.168.1.100 23662**看下是否可以连通该端口，请使用**root**用户和**omm**用户都执行一遍。 如果出现**Escape character is '^]'**类似打印则说明可以连通，如果出现**connection refused**则表示失败，无法连接到相关端口。

如果相关端口打开，但是从别的节点无法连通到该端口，则需要排查下相关网络配置。

说明

23662这个端口每次都是随机的，所以要根据自己启动任务打开的端口来测试。

----结束

19.11 提交 Spark 任务时连接 ResourceManager 异常

问题背景与现象

连接ResourceManager异常，导致Spark任务提交失败。

原因分析

1. 在Driver端打印异常如下，打印连接两个ResourceManager主备节点的26004端口均被拒绝：
```
15/08/19 18:36:16 INFO RetryInvocationHandler: Exception while invoking getClusterMetrics of class
ApplicationClientProtocolPBClientImpl over 33 after 1 fail over attempts. Trying to fail over after
sleeping for 17448ms.
java.net.ConnectException: Call From ip0 to ip1:26004 failed on connection exception:
java.net.ConnectException: Connection refused.
INFO RetryInvocationHandler: Exception while invoking getClusterMetrics of class
ApplicationClientProtocolPBClientImpl over 32 after 2 fail over attempts. Trying to fail over after
sleeping for 16233ms.
 java.net.ConnectException: Call From ip0 to ip2:26004 failed on connection exception:
java.net.ConnectException: Connection refused;
```
2. 在MRS Manager页面查看ResourceManager此时是否功能正常，如果Yarn服务状态故障或某个Yarn服务的实例出现未知之类的异常说明此时集群的ResourceManager可能异常。
3. 排查使用的客户端是否是集群最新的客户端。

   排查集群是否做过实例ResourceManager迁移相关操作（先卸载某个ResourceManager实例，然后在其他节点添加）。
4. 在MRS Manager页面查看审计日志，是否有相关操作的记录。

   使用**ping**命令，查看IP是否可连通。

解决办法

- 如果ResourceManager出现异常，可参考相关告警章节查看解决方法。
- 如果客户端不是最新，请重新下载安装客户端。
- 若使用**ping**命令查看IP不通，需要协调网络管理相关人员协助排查网络。
- 若集群开启高可用，尝试将Yarn参数“yarn.client.failover-sleep-base-ms”调小。

19.12 DataArts Studio 调度 Spark 作业失败

问题现象

DataArts Studio作业调度失败，显示读取“/thriftserver/active_thriftserver”路径下的数据失败。

报错信息为：

```
Can not get JDBC Connection, due to KeeperErrorCode = NoNode for /thriftserver/active_thriftserver。
```

原因分析

DataArts Studio提交Spark作业时调用Spark的JDBC方式，而Spark会启动一个名为thriftserver的进程以供客户端提供JDBC连接，JDBCServer在启动时会在ZooKeeper的“/thriftserver”目录下创建子目录“active_thriftserver”，并且注册相关连接信息。如果读不到该连接信息就会JDBC连接异常。

处理步骤

检查ZooKeeper下面是否有目标目录和注册的信息

步骤1 以**root**用户登录任意一个Master节点并初始化环境变量。

source /opt/client/bigdata_env

步骤2 执行**zkCli.sh -server '***ZooKeeper实例IP地址:ZooKeeper连接端口***'**命令登录ZooKeeper。

ZooKeeper连接端口通常为2181，可通过查看ZooKeeper服务配置参数“clientPort”获取。

步骤3 执行**ls /thriftserver**查看是否有“active_thriftserver”目录。

- 如果有active_thriftserver目录，执行**get /thriftserver/active_thriftserver**查看该目录下是否有注册的配置信息。
  - 如果有注册的配置信息，联系华为云支持人员处理。
  - 如果没有注册的配置信息，执行**步骤4**
- 如果没有active_thriftserver目录，执行**步骤4**。

步骤4 登录Manager界面，查看Spark的JDBCServer实例的主备状态是否未知。

- 是，执行**步骤5**。
- 否，联系运维人员处理。

步骤5 重启两个JDBCServer实例，查看主备实例状态恢复正常且zk下面有了目标目录和数据，作业即可恢复正常。若实例状态没有恢复请联系华为云支持人员处理。

----结束

19.13 通过 API 提交 Spark 作业后作业状态为 error

用户问题

使用API提交Spark作业后，作业状态显示为error。

问题现象

修改“/opt/client/Spark/spark/conf/log4j.properties”中的日志级别，使用API V1.1接口作业提交后，状态显示为error。

原因分析

executor会监控作业日志回显，确定作业执行结果，改为error后，检测不到输出结果，因此过期后判断作业状态为异常。

处理步骤

将“/opt/client/Spark/spark/conf/log4j.properties”中的日志级别修改为**info**。

建议与总结

建议使用V2接口提交作业接口。

19.14 MRS 集群反复上报 43006 告警

用户问题

集群反复出现“ALM-43006 JobHistory进程堆内存使用超出阈值”告警，且按照告警参考处理无效。

问题现象

集群出现告警“ALM-43006 JobHistory进程堆内存使用超出阈值”并且按照指导处理以后，运行一段时间又会出现同样的告警。

原因分析

可能存在JobHistory内存泄露问题，需要安装相应的补丁修复。

处理步骤

- 适当调大JobHistory进程堆内存。
- 如果已经调大堆内存，可以通过重启JobHistory实例规避。

19.15 在 spark-beeline 中创建或删除表失败

用户问题

客户在spark-beeline频繁创建和删除大量用户的场景下，个别用户偶现创建/删除表失败。

问题现象

创建表过程：

```
CREATE TABLE wlg_test001 (start_time STRING,value INT);
```

报错:

```
Error: org.apache.spark.sql.AnalysisException:
org.apache.hadoop.hive.ql.metadata.HiveException: MetaException(message:Failed to grant permission on
HDFSjava.lang.reflect.UndeclaredThrowableException); (state=,code=0)
```

原因分析

1. 查看MetaStore日志。

```
.hive.metastore.RetryingHMSHandler> | org.apache.hadoop.hive.ql.log.PerfLogger.PerfLogBegin(PerfLogger.java:121)
2020-08-31 14:41:38,504 | INFO  | pool-7-thread-197 | 197: create_table: Table(tableName:wlg_test001, dbName:hive_csb_csb_3f8_x48s
srbt_5lbi2edu, owner:CSB_csb_3f8_x48ssrbt, createTime:1598856098, lastAccessTime:0, retention:0, sd:StorageDescriptor(cols:[FieldS
chema(name:start_time, type:string, comment:null), FieldSchema(name:value, type:int, comment:null)], location:hdfs://hacluster/use
r/hive/warehouse/hive_csb_csb_3f8_x48ssrbt_5lbi2edu.db/wlg_test001, inputFormat:org.apache.hadoop.mapred.TextInputFormat, outputFo
rmat:org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat, compressed:false, numBuckets:-1, serdeInfo:SerDeInfo(name:null, s
erializationLib:org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, parameters:{serialization.format=1}), bucketCols:[], sortCols:
[], parameters:{}, skewedInfo:SkewedInfo(skewedColNames:[], skewedColValues:[], skewedColValueLocationMaps:{})), partitionKeys:[],
 parameters:{spark.sql.sources.schema.numParts=1, spark.sql.sources.schema.part.0={"type":"struct","fields":[{"name":"start_time",
"type":"string","nullable":true,"metadata":{}},{"name":"value","type":"integer","nullable":true,"metadata":{}}]}}, viewOriginalTex
t:null, viewExpandedText:null, tableType:MANAGED_TABLE, privileges:PrincipalPrivilegeSet(userPrivileges:{CSB_csb_3f8_x48ssrbt=[Pri
vilegeGrantInfo(privilege:INSERT, createTime:-1, grantor:spark, grantorType:USER, grantOption:true), PrivilegeGrantInfo(privilege:
SELECT, createTime:-1, grantor:spark, grantorType:USER, grantOption:true), PrivilegeGrantInfo(privilege:UPDATE, createTime:-1, gra
ntor:spark, grantorType:USER, grantOption:true), PrivilegeGrantInfo(privilege:DELETE, createTime:-1, grantor:spark, grantorType:US
ER, grantOption:true)]}, groupPrivileges:null, rolePrivileges:null)) | org.apache.hadoop.hive.metastore.HiveMetaStore$HMSHandler.l
ogInfo(HiveMetaStore.java:881)
2020-08-31 14:41:38,515 | WARN  | pool-7-thread-197 | Location: hdfs://hacluster/user/hive/warehouse/hive_csb_csb_3f8_x48ssrbt_5lb
i2edu.db/wlg_test001 specified for non-external table:wlg_test001 | org.apache.hadoop.hive.metastore.HiveMetaStore$HMSHandler.crea
te_table_core(HiveMetaStore.java:1546)
2020-08-31 14:41:38,516 | INFO  | pool-7-thread-197 | Creating directory if it doesn't exist: hdfs://hacluster/user/hive/warehouse
/hive_csb_csb_3f8_x48ssrbt_5lbi2edu.db/wlg_test001 | org.apache.hadoop.hive.common.FileUtils.mkdir(FileUtils.java:507)
2020-08-31 14:41:38,566 | INFO  | pool-7-thread-197 | 197: get_database: hive_csb_csb_3f8_x48ssrbt_5lbi2edu | org.apache.hadoop.hi
ve.metastore.HiveMetaStore$HMSHandler.logInfo(HiveMetaStore.java:881)
2020-08-31 14:41:38,578 | INFO  | pool-7-thread-197 | 197: get_table : db=hive_csb_csb_3f8_x48ssrbt_5lbi2edu tbl=wlg_test001 | org
.apache.hadoop.hive.metastore.HiveMetaStore$HMSHandler.logInfo(HiveMetaStore.java:881)
2020-08-31 14:41:38,594 | ERROR | pool-7-thread-197 | MetaException(message:Failed to grant permission on HDFSjava.lang.reflect.Un
declaredThrowableException)
        at org.apache.hadoop.hive.metastore.HiveMetaStore$HMSHandler.create_table_with_environment_context(HiveMetaStore.java:1638
)
        at sun.reflect.GeneratedMethodAccessor94.invoke(Unknown Source)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at org.apache.hadoop.hive.metastore.RetryingHMSHandler.invokeInternal(RetryingHMSHandler.java:140)
```

2. 查看HDFS日志。

```
2020-08-31 14:41:38,568 | INFO  | Socket Reader #1 for port 9820 | Authorization successful for hive/hadoop.036a3461_d09b_494f_a32
c_af273307d943.com@036A3461_D09B_494F_A32C_AF273307D943.COM (auth:KERBEROS) for protocol=interface org.apache.hadoop.hdfs.protocol
.ClientProtocol | ServiceAuthorizationManager.java:135
2020-08-31 14:41:38,586 | INFO  | IPC Server handler 7 on 9820 | IPC Server handler 7 on 9820, call Call#3822197 Retry#0 org.apach
e.hadoop.hdfs.protocol.ClientProtocol.checkAccess from 192.168.1.66:50540: org.apache.hadoop.security.AccessControlException: Perm
ission denied: user=hive, access=READ, inode="/user/hive/warehouse/hive_csb_csb_3f8_x48ssrbt_5lbi2edu.db/wlg_test001":spark:hive:d
rwx------ | Server.java:2523
2020-08-31 14:41:38,852 | INFO  | Socket Reader #1 for port 9820 | Auth successful for hwstaff_pub_0tw00ru6@036A3461_D09B_494F_A32
C_AF273307D943.COM (auth:TOKEN) | Server.java:1700
2020-08-31 14:41:38,911 | INFO  | Socket Reader #1 for port 9820 | Authorization successful for hwstaff_pub_0tw00ru6@036A3461_D09B
```

3. 权限对比（test001为异常用户创建表，test002为正常用户创建表）。

```
drwx------   - spark               hive          0 2020-08-31 14:41 /user/hive/warehouse/hive_csb_csb_3f8_x48ssrbt_5lbi2edu.db/wl
g_test001
drwxrwx---+  - spark               hive          0 2020-08-31 15:07 /user/hive/warehouse/hive_csb_csb_3f8_x48ssrbt_5lbi2edu.db/wl
g_test002
```

4. drop表时报类似下面的错。

```
dataplan_modela_csbch2;
Error: Error while compiling statement: FAILED:
SemanticException Unable to fetch table dataplan_modela_csbch2.
java.security.AccessControlException: Permission denied: user=CSB_csb_3f8_x48ssrbt,
access=READ,
inode="/user/hive/warehouse/hive_csb_csb_3f8_x48ssrbt_5lbi2edu.db/
dataplan_modela_csbch2":spark:hive:drwx------
```

5. 根因分析。

   创建集群时创建的默认用户使用了相同的uid，造成用户错乱。在大量创建用户的场景下，触发了该问题，导致在创建表时偶现Hive用户没有权限。

```
[root@node-master2iMrt ~]#
[root@node-master2iMrt ~]#
[root@node-master2iMrt ~]# id hive
uid=20013(hive/hadoop.036a3461_d09b_494f_a32c_af273307d943.com) gid=10002(hive) groups=10002(hive)
[root@node-master2iMrt ~]#
[root@node-master2iMrt ~]#
[root@node-master2iMrt ~]#
[root@node-master2iMrt ~]#
[root@node-master2iMrt ~]# id hive
uid=20013(hive) gid=10002(hive) groups=10002(hive),10001(hadoop),10000(supergroup),8003(System_administrator_186),9998(ficommon)
[root@node-master2iMrt ~]#
[root@node-master2iMrt ~]#
[root@node-master2iMrt ~]#
```

```
objectClass: krbPrincipalAux
objectClass: krbTicketPolicyAux

# hive, Peoples, hadoop.com
dn: cn=hive,ou=Peoples,dc=hadoop,dc=com
uid: hive
homeDirectory: /home/hive/hadoop.036a3461_d09b_494f_a32c_af273307d943.com
cn: hive
uidNumber: 20013
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
userPassword:: elNTSEF9cXZWS0VlMi9pYVFpdzFmUmNIUVJFUEJYZWtKLzZHMHk=
gidNumber: 10002

# hive/hadoop.036a3461_d09b_494f_a32c_af273307d943.com, Peoples, hadoop.com
dn: cn=hive/hadoop.036a3461_d09b_494f_a32c_af273307d943.com,ou=Peoples,dc=hado
 op,dc=com
uid: hive/hadoop.036a3461_d09b_494f_a32c_af273307d943.com
homeDirectory: /home/hive/hadoop.036a3461_d09b_494f_a32c_af273307d943.com
cn: hive/hadoop.036a3461_d09b_494f_a32c_af273307d943.com
uidNumber: 20013
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
gidNumber: 10002
description: [userName:"hive/hadoop.036a3461_d09b_494f_a32c_af273307d943.com"]
description: [userType:"1"]
description: [groupList:"hive,hadoop,supergroup,compcommon"]
description: [roleList:"System_administrator"]
description: [description:"aGl2ZSBkZWZhdWx0IHVzZXIjSGl2Zem7mOiupOeUqOaItw=="]
description: [createTime:"1554974652422"]
description: [defaultUser:"0"]
description: [primaryGroup:"hive"]

# hive/hadoop.036a3461_d09b_494f_a32c_af273307d943.com@036A3461_D09B_494F_A32C_
 AF273307D943.COM, 036A3461_D09B_494F_A32C_AF273307D943.COM, krbcontainer, hado
```

处理步骤

重启集群sssd进程。

以**root**用户执行**service sssd restart**命令重启sssd服务，执行**ps -ef | grep sssd**命令，查看sssd进程是否正常。

正常状态为存在/usr/sbin/sssd进程和三个子进程/usr/libexec/sssd/sssd_be、/usr/libexec/sssd/sssd_nss、/usr/libexec/sssd/sssd_pam。

19.16 集群外节点提交 Spark 作业时报错无法连接 Driver

问题现象

集群外节点和集群各个节点网络已经互通，在集群外节点使用client模式提交Spark任务到Yarn上，任务失败，报错信息为无法连接Driver。

原因分析

使用client模式提交Spark任务的时候，Spark的driver进程是在客户端这边，而后面的executor都需要和Driver进行交互来运行作业。

如果NodeManager连不上客户端所在的节点，就会报错：

```
Log Length: 174453
Showing 4096 bytes of 174453 total. Click here for the full log.
onnect to driver at ecs-d6d9-1112169:22741, retrying ... | org.apache.spark.internal.Logging$class.logError(Logging.scala:70)
2020-11-21 16:04:34,150 | ERROR | [main] | Failed to connect to driver at Driver hostnamexxxxxx:22741, retrying ... | org.apache.spark.internal.Logging$class.logError(Logging.scala:70)
2020-11-21 16:04:34,251 | ERROR | [main] | Failed to connect to driver at [redacted]:22741, retrying ... | org.apache.spark.internal.Logging$class.logError(Logging.scala:70)
2020-11-21 16:04:34,351 | ERROR | [main] | Failed to connect to driver at [redacted]:22741, retrying ... | org.apache.spark.internal.Logging$class.logError(Logging.scala:70)
2020-11-21 16:04:34,452 | ERROR | [main] | Failed to connect to driver at [redacted]:22741, retrying ... | org.apache.spark.internal.Logging$class.logError(Logging.scala:70)
2020-11-21 16:04:34,552 | ERROR | [main] | Failed to connect to driver at [redacted]:22741, retrying ... | org.apache.spark.internal.Logging$class.logError(Logging.scala:70)
2020-11-21 16:04:34,653 | ERROR | [main] | Failed to connect to driver at [redacted]:22741, retrying ... | org.apache.spark.internal.Logging$class.logError(Logging.scala:70)
2020-11-21 16:04:34,753 | ERROR | [main] | Failed to connect to driver at [redacted]:22741, retrying ... | org.apache.spark.internal.Logging$class.logError(Logging.scala:70)
2020-11-21 16:04:34,855 | ERROR | [main] | Failed to connect to driver at [redacted]:22741, retrying ... | org.apache.spark.internal.Logging$class.logError(Logging.scala:70)
2020-11-21 16:04:34,956 | ERROR | [main] | Failed to connect to driver at [redacted]:22741, retrying ... | org.apache.spark.internal.Logging$class.logError(Logging.scala:70)
2020-11-21 16:04:35,057 | ERROR | [main] | Failed to connect to driver at [redacted]:22741, retrying ... | org.apache.spark.internal.Logging$class.logError(Logging.scala:70)
2020-11-21 16:04:35,161 | ERROR | [main] | Uncaught exception: | org.apache.spark.internal.Logging$class.logError(Logging.scala:91)
org.apache.spark.SparkException: Failed to connect to driver!
        at org.apache.spark.deploy.yarn.ApplicationMaster.waitForSparkDriver(ApplicationMaster.scala:630)
```

处理步骤

在客户端的Spark配置中指定Driver的IP地址：

“<客户端安装位置>/Spark/spark/conf/spark-defaults.conf”中添加参数“spark.driver.host=driverIP”，重新运行Spark任务即可。

建议与总结

建议通过cluster模式提交作业。

19.17 运行 Spark 任务发现大量 shuffle 结果丢失

问题现象

Spark任务运行失败，查看任务日志发现大量打印shuffle文件丢失。

原因分析

Spark运行的时候会将临时产生的shuffle文件放在executor的临时目录中，方便后面获取。

而当某个executor异常退出时，NodeManager会把这个executor所在的container临时目录删除，随后其他executor再来申请这个executor的shuffle结果就会报文件找不到。

因此，遇到这样的问题需要确认是否executor异常退出，可以根据spark任务页面的executors便签页查看是否有dead状态的executor，查看各个dead状态的executor日志，确认异常退出的原因（其中可能有部分executor退出原因就是因为shuffle文件找不到，需要找到最早异常退出的executor）。

常见的异常退出：

- executor发生OOM
- executor运行时出现多个task任务失败
- executor所在节点被清理

处理步骤

根据executor异常退出的实际原因调整或者修改任务参数或代码，重新运行Spark任务即可。

19.18 JDBCServer 长时间运行导致磁盘空间不足

用户问题

连接Spark的JDBCServer服务提交spark-sql任务到yarn集群上，在运行一段时间以后会出现Core节点的数据盘被占满的情况。

问题现象

客户连接Spark的JDBCServer服务提交spark-sql任务到yarn集群上，在运行一段时间以后会出现Core节点的数据盘被占满的情况。

后台查看磁盘使用情况，主要是JDBCServer服务的APP临时文件（shuffle生成的文件）太多，并且没有进行清理占用了大量内存。

原因分析

查询Core节点有大量文件的目录，发现大部分都是类似“blockmgr-033707b6-fbbb-45b4-8e3a-128c9bcfa4bf”的目录，里面存放了计算过程中产生的shuffle临时文件。

因为JDBCServer启动了Spark的动态资源分配功能，已经将shuffle托管给NodeManager，NodeManager只会按照APP的运行周期来管理这些文件，并不会关注单个executor所在的container是否存在。因此，只有在APP结束的时候才会清理这些临时文件。任务运行时间较长时导致临时文件过多占用了大量磁盘空间。

处理步骤

启动一个定时任务来清理超过一定时间的shuffle文件，例如每个整点清理超过6个小时的文件：

步骤1 创建脚本“clean_appcache.sh”，若存在多个数据盘，请根据实际情况修改BASE_LOC中data1的值。

- 安全集群
```
#!/bin/bash
BASE_LOC=/srv/BigData/hadoop/data1/nm/localdir/usercache/spark/appcache/application_*/
blockmgr*
find $BASE_LOC/ -mmin +360 -exec rmdir {} \;
find $BASE_LOC/ -mmin +360 -exec rm {} \;
```
- 普通集群
```
#!/bin/bash
BASE_LOC=/srv/BigData/hadoop/data1/nm/localdir/usercache/omm/appcache/application_*/
blockmgr*
find $BASE_LOC/ -mmin +360 -exec rmdir {} \;
find $BASE_LOC/ -mmin +360 -exec rm {} \;
```

步骤2 修改脚本权限。

chmod 755 clean_appcache.sh

步骤3 增加一个定时任务来启动清理脚本，脚本路径请根据实际脚本存放位置修改。

查看定时任务：**crontab -l**

编辑定时任务：**crontab -e**

```
0 * * * * sh /root/clean_appcache.sh > /dev/null 2>&1
```

----结束

19.19 spark-shell 执行 SQL 跨文件系统 load 数据到 Hive 表失败

用户问题

使用spark-shell命令执行SQL或者spark-submit提交的Spark任务里面有SQL的load命令，并且原数据和目标表存储位置不是同一套文件系统，上述两种方式MapReduce任务启动时会报错。

原因分析

当使用load导入数据到Hive表的时候，属于需要跨文件系统的情况（例如原数据在HDFS上，而Hive表数据存放在OBS上），并且文件长度大于阈值（默认32 MB），则会触发使用distcp的MapReduce任务来执行数据迁移操作。这个MapReduce任务配置直接从Spark任务配置里面提取，但是Spark任务的net.topology.node.switch.mapping.impl配置项不是hadoop的默认值，需要使用Spark的jar包，因此MapReduce会报类找不到。

处理步骤

方案一：

如果文件较小，则可以将默认长度设置得大于文件最大长度，例如最大的文件是95 MB，则设置：

```
hive.exec.copyfile.maxsize=104857600
```

方案二：

如果确实文件较大，需要使用distcp任务来提高数据迁移效率，则可以在Spark任务启动的时候增加设置参数：

```
--conf spark.hadoop.net.topology.node.switch.mapping.impl=org.apache.hadoop.net.ScriptBasedMapping
```

19.20 Spark 任务提交失败

问题现象

- Spark提交任务直接提示无法提交任务。
- Spark提示无法获取到yarn的相关jar包。
- 提示多次提交一个文件。

原因分析

- 问题1：

  最常见的无法提交任务原因是认证失败，

```
2021-04-28 17:20:03,680 | ERROR | main | java.lang.UnsatisfiedLinkError: /tmp/opencv_openpnp6650342257652861374/nu/pattern/opencv/linux/x86_64/libopencv_java430.so: /lib64/libm.so.6: version `GLIBC_2.27' not
 found (required by /tmp/opencv_openpnp6650342257652861374/nu/pattern/opencv/linux/x86_64/libopencv_java430.so) | org.apache.spark.sql.vision.VisionSparkUdfRegisters.register(VisionSparkUdfRegister.scala:34)
2021-04-28 17:23:07,612 | WARN  | main | No Partition Defined for Window operation! Moving all data to a single partition, this can cause serious performance degradation. | org.apache.spark.internal.Loggings$
class.logWarning(Logging.scala:66)
2021-04-28 17:24:08,655 | WARN  | main | No Partition Defined for Window operation! Moving all data to a single partition, this can cause serious performance degradation. | org.apache.spark.internal.Loggings$
class.logWarning(Logging.scala:66)
```

  还有可能是参数设置不正确。

- 问题2：

  集群默认会把分析节点的hadoop相关jar包添加到任务的classpath中，如果提示yarn的包找不到，一般都是因为hadoop的相关配置没有设置。

- 问题3：

  常见的场景是使用--files上传了user.keytab，然后使用--keytab又指定了同一个文件，导致一个文件多次被上传。

```
2021-04-29 10:08:56,973 | WARN  | main | Stopping a MetricsSystem that is not running | org.apache.spark.metrics.MetricsSystem.logWarning(Logging.scala:66)
Exception in thread "main" java.lang.IllegalArgumentException: Attempt to add (file:///opt/user.keytab) multiple times to the distributed cache.
        at org.apache.spark.deploy.yarn.Client$$anonfun$prepareLocalResources$10$$anonfun$apply$6.apply(Client.scala:646)
        at org.apache.spark.deploy.yarn.Client$$anonfun$prepareLocalResources$10$$anonfun$apply$6.apply(Client.scala:637)
        at scala.collection.mutable.ResizableArray$class.foreach(ResizableArray.scala:59)
        at scala.collection.mutable.ArrayBuffer.foreach(ArrayBuffer.scala:48)
        at org.apache.spark.deploy.yarn.Client$$anonfun$prepareLocalResources$10.apply(Client.scala:637)
        at org.apache.spark.deploy.yarn.Client$$anonfun$prepareLocalResources$10.apply(Client.scala:636)
        at scala.collection.immutable.List.foreach(List.scala:392)
        at org.apache.spark.deploy.yarn.Client.prepareLocalResources(Client.scala:636)
        at org.apache.spark.deploy.yarn.Client.createContainerLaunchContext(Client.scala:913)
        at org.apache.spark.deploy.yarn.Client.submitApplication(Client.scala:205)
        at org.apache.spark.scheduler.cluster.YarnClientSchedulerBackend.start(YarnClientSchedulerBackend.scala:57)
        at org.apache.spark.scheduler.TaskSchedulerImpl.start(TaskSchedulerImpl.scala:188)
        at org.apache.spark.SparkContext.<init>(SparkContext.scala:524)
        at org.apache.spark.SparkContext$.getOrCreate(SparkContext.scala:2695)
        at org.apache.spark.sql.SparkSession$Builder$$anonfun$7.apply(SparkSession.scala:956)
```

处理步骤

- 问题1：

  重新kinit一个用户并修改相应的配置参数。

- 问题2：

  查看hadoop相关的配置项是否正确，查看spark的conf目录下的core-site.xml，hdfs-site.xml，yarn-site.xml，mapred-site.xml等配置文件是否存在问题。

- 问题3：

  重新复制一个user.keytab，例如：

  cp user.keytab user2.keytab

  spark-submit --master yarn --files user.keytab --keytab user2.keytab .....

19.21 Spark 任务运行失败

问题现象

- 报错显示executor出现OOM。
- 失败的task信息显示失败原因是lost task xxx。

原因分析

- 问题1：一般出现executor OOM，都是因为数据量过大，也有可能是因为同一个executor上面同时运行的task太多。
- 问题2：有些task运行失败会报上述错误。当看到这个报错的时候，需要确认的是丢失的这个task在哪个节点上面运行，一般的情况是这个丢失的task异常退出导致的。

处理步骤

- 问题1：
  - 对于数据量过大，需要调整executor的内存大小的，使用--executor-memory指定内存大小；
  - 对于同时运行的task太多，主要看--executor-cores设置的vcore数量。
- 问题2：

需要在相应的task的日志里面查找异常原因。如果有OOM的情况，请参照问题1。

19.22 JDBCServer 连接失败

问题现象

- 提示ha-cluster不识别（unknowHost或者必须加上端口）。
- 提示连接JDBCServer失败。

原因分析

- 问题1：使用**spark-beeline**命令连接JDBCServer，因为MRS_3.0以前的JDBCServer是HA模式，因此需要使用特定的URL和MRS Spark的自带的jar包来连接JDBCServer。
- 问题2：确认JDBCServer服务是否正常，查看对应的端口是否正常监测。

处理步骤

- 问题1：需要使用特定的URL和MRS Spark的自带的jar包来连接JDBCServer。
- 问题2：确认JDBCServer服务是否正常，查看对应的端口是否正常监测。

19.23 查看 Spark 任务日志失败

问题现象

- 任务运行中查看日志失败。
- 任务运行完成，但是查看不到日志。

原因分析

- 问题1：可能原因是MapReduce服务异常。
- 问题2：可能原因如下：
  - Spark的JobHistory服务异常。
  - 日志太大，NodeManager在做日志汇聚的时候出现超时。
  - HDFS存放日志目录权限异常（默认/tmp/logs/用户名/logs）。
  - 日志已被清理（spark的JobHistory默认存放7天的eventLog，配置项为spark.history.fs.cleaner.maxAge；MapReduce默认存放15天的任务日志，配置项为mapreduce.jobhistory.max-age-ms）。
  - 如果Yarn页面上也找不到，可能是被Yarn清理了（默认存放10000个历史任务，配置项为yarn.resourcemanager.max-completed-applications）。

处理步骤

- 问题1：确认MapReduce服务是否正常，如果异常，尝试重启服务。如果还是不能恢复，需要查看后台JobhistoryServer日志。
- 问题2：依次排查可能的情况：

a. 查看Spark的JobHistory是否运行正常；
b. 通过查看yarn的app详情页面，确认日志文件是否过大，如果日志汇聚失败，页面的“Log Aggregation Status:”应该显示为失败或者超时；
c. 查看对应目录权限是否异常；
d. 查看目录下是否有对应的appid文件（Spark的eventlog存放目录：MRS 3.x及以后版本的目录是hdfs://hacluster/spark2xJobHistory2x，MRS 3.x以前版本的目录是hdfs://hacluster/sparkJobHistory，任务运行日志存放目录是hdfs://hacluster/tmp/logs/*用户名*/logs）；
e. 查看appid和当前作业的id是否超过历史记录最大值。

19.24 Spark Streaming 任务提交问题

问题现象

- 连接Kafka报类找不到。
- 连接带Kerberos的Kafka，报认证错误。
- SparkStreaming任务运行一段时间后开始报TOKEN过期问题。

原因分析

- 问题1：Spark提交任务默认不会加载kafka的相关包，所以需要在启动命令中增加--jars来指定对应kafka版本的jar包
- 问题2：连接Kafka无法使用Spark的认证信息，需要将相关的认证使用JVM的参数设置进去。
- 问题3：Spark默认使用当前客户端的认证信息提交任务，也可以使用代码login的方式。但是这两种认证方式都无法更新任务使用的Token，当提交的时候生成的Token信息过期以后就无法再使用，因此报错。解决办法是使用--keytab和--principal将keytab文件和对应用户带入任务中。

处理步骤

- 问题1：启动命令中增加--jars来指定对应kafka版本的jar包，一般是在Spark客户端目录/jars/streamingClient（0.8版本Kafka）和Spark客户端目录/jars/streamingClient010（0.10版本Kafka）。
- 问题2：参考指导文档**编辑并运行程序**。
- 问题3：使用--keytab和--principal将keytab文件和对应用户带入任务中。如果此处的keytab文件和之前Kafka的jaas.conf中配置的是同一个，则Spark会报一个文件多次上传的问题。解决办法是复制一份keytab文件，使得--files和--keytab上传不同的文件。

19.25 Spark 连接其他服务认证问题

问题现象

- Spark连接HBase，报认证失败或者连接不到HBase表。
- Spark连接HBase报找不到jar包。

原因分析

- 问题1：HBase没有获取到当前任务的认证信息，导致连接HBase的时候认证失败，无法读取到相应数据。
- 问题2：Spark默认没有加载HBase相关的jar包，需要使用--jars添加到任务中。

处理步骤

- 问题1：可以尝试开启HBase认证开关：spark.yarn.security.credentials.hbase.enabled=true。但不建议直接用HBase客户端的hbase-site.xml替换Spark客户端下的hbase-site.xml，两者并不是完全相同。
- 问题2：需要将HBase相关的包使用--jars上传。

19.26 Spark 连接 Kafka 认证错误

问题现象

MRS 2.1.0版本集群中Spark消费kafka数据，过一天以后就会出现认证失败的报错：

```
2021-06-23 10:36:00,218 | INFO  | Driver | Final app status: FAILED, exitCode: 15, (reason: User class threw exception: org.apache.spark.SparkException: Job aborted due to
stage failure: Task 1 in stage 1515.0 failed 4 times, most recent failure: Lost task 1.3 in stage 1515.0 (TID 3036, node-ana-coreiUMb, executor 1):
org.apache.kafka.common.errors.SaslAuthenticationException: An error: (java.security.PrivilegedActionException: javax.security.sasl.SaslException: GSS initiate failed [Caused
by GSSException: No valid credentials provided (Mechanism level: Failed to find any Kerberos tgt)]) occurred when evaluating SASL token received from the Kafka Broker. Kafka
Client will go to AUTHENTICATION_FAILED state.
Caused by: javax.security.sasl.SaslException: GSS initiate failed [Caused by GSSException: No valid credentials provided (Mechanism level: Failed to find any Kerberos tgt)]
    at com.sun.security.sasl.gsskerb.GssKrb5Client.evaluateChallenge(GssKrb5Client.java:211)
    at org.apache.kafka.common.security.authenticator.SaslClientAuthenticator$2.run(SaslClientAuthenticator.java:362)
    at org.apache.kafka.common.security.authenticator.SaslClientAuthenticator$2.run(SaslClientAuthenticator.java:360)
    at java.security.AccessController.doPrivileged(Native Method)
    at javax.security.auth.Subject.doAs(Subject.java:422)
    at org.apache.kafka.common.security.authenticator.SaslClientAuthenticator.createSaslToken(SaslClientAuthenticator.java:360)
    at org.apache.kafka.common.security.authenticator.SaslClientAuthenticator.sendSaslClientToken(SaslClientAuthenticator.java:270)
    at org.apache.kafka.common.security.authenticator.SaslClientAuthenticator.authenticate(SaslClientAuthenticator.java:207)
    at org.apache.kafka.common.network.KafkaChannel.prepare(KafkaChannel.java:81)
    at org.apache.kafka.common.network.Selector.poll(Selector.java:486)
    at org.apache.kafka.common.network.Selector.poll(Selector.java:424)
    at org.apache.kafka.clients.NetworkClient.poll(NetworkClient.java:460)
    at org.apache.kafka.clients.consumer.internals.ConsumerNetworkClient.poll(ConsumerNetworkClient.java:261)
    at org.apache.kafka.clients.consumer.internals.ConsumerNetworkClient.poll(ConsumerNetworkClient.java:233)
    at org.apache.kafka.clients.consumer.internals.ConsumerNetworkClient.poll(ConsumerNetworkClient.java:224)
    at org.apache.kafka.clients.consumer.internals.ConsumerNetworkClient.awaitMetadataUpdate(ConsumerNetworkClient.java:156)
    at org.apache.kafka.clients.consumer.internals.ConsumerCoordinator.poll(ConsumerCoordinator.java:304)
    at org.apache.kafka.clients.consumer.KafkaConsumer.pollOnce(KafkaConsumer.java:1149)
    at org.apache.kafka.clients.consumer.KafkaConsumer.poll(KafkaConsumer.java:1115)
    at org.apache.spark.streaming.kafka010.CachedKafkaConsumer.poll(CachedKafkaConsumer.scala:154)
    at org.apache.spark.streaming.kafka010.CachedKafkaConsumer.get(CachedKafkaConsumer.scala:86)
    at org.apache.spark.streaming.kafka010.KafkaRDDIterator.next(KafkaRDD.scala:279)
    at org.apache.spark.streaming.kafka010.KafkaRDDIterator.next(KafkaRDD.scala:231)
```

原因分析

集群版本和运行的程序包版本不匹配。

处理步骤

1. 参考**运行Spark Streaming 对接Kafka0-10样例程序**进行相关配置。
2. 当前集群为MRS 2.1.0版本，请如需使用Maven库中的提供的样例程序，请获取spark-streaming-kafka-0-10_2.11-2.3.2-mrs-2.1.jar包。

19.27 SparkSQL 读取 ORC 表报错

问题现象

Hive创建ORC存储格式的表，用SparkSQL读取该表时报错：

```
Caused by: java.util.concurrent.ExecutionException: java.lang.IndexOutOfBoundsException: Index: 0
    at java.util.concurrent.FutureTask.report(FutureTask.java:122)
    at java.util.concurrent.FutureTask.get(FutureTask.java:192)
    at org.apache.hadoop.hive.ql.io.orc.OrcInputFormat.generateSplitsInfo(OrcInputFormat.java:1016)
    ... 51 more
Caused by: java.lang.IndexOutOfBoundsException: Index: 0
    at java.util.Collections$EmptyList.get(Collections.java:4456)
    at org.apache.hadoop.hive.ql.io.orc.OrcProto$Type.getSubtypes(OrcProto.java:12240)
    at org.apache.hadoop.hive.ql.io.orc.ReaderImpl.getColumnIndicesFromNames(ReaderImpl.java:651)
    at org.apache.hadoop.hive.ql.io.orc.ReaderImpl.getRawDataSizeOfColumns(ReaderImpl.java:634)
    at org.apache.hadoop.hive.ql.io.orc.OrcInputFormat$SplitGenerator.populateAndCacheStripeDetails(OrcInputFormat.java:927)
    at org.apache.hadoop.hive.ql.io.orc.OrcInputFormat$SplitGenerator.call(OrcInputFormat.java:836)
    at org.apache.hadoop.hive.ql.io.orc.OrcInputFormat$SplitGenerator.call(OrcInputFormat.java:702)
    at java.util.concurrent.FutureTask.run(FutureTask.java:266)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at java.lang.Thread.run(Thread.java:748)
```

原因分析

该问题为一个开源社区问题：https://issues.apache.org/jira/browse/HIVE-11102，使用开源的hive 1.2.1版本包就有可能触发此问题。

MRS的Hive for Spark相关模块已修复该问题。

处理步骤

使用MRS集群自带的Hive for Spark包：

```
hive-beeline-1.2.1.spark_2.2.1-mrs-x.x.x.jar
hive-cli-1.2.1.spark_2.2.1-mrs-x.x.x.jar
hive-common-1.2.1.spark_2.2.1-mrs-x.x.x.jar
hive-exec-1.2.1.spark_2.2.1-mrs-x.x.x.jar
hive-jdbc-1.2.1.spark_2.2.1-mrs-x.x.x.jar
hive-metastore-1.2.1.spark_2.2.1-mrs-x.x.x.jar
```

华为云Maven库请参考指导**通过开源镜像站获取样例工程**。

19.28 Spark WebUI 页面上 stderr 和 stdout 无法跳转到日志页面

问题现象

使用MRS 3.x版本集群，运行Spark任务，在Spark2x服务的WebUI页面上无法直接单击stderr和stdout跳转到日志页面。

原因分析

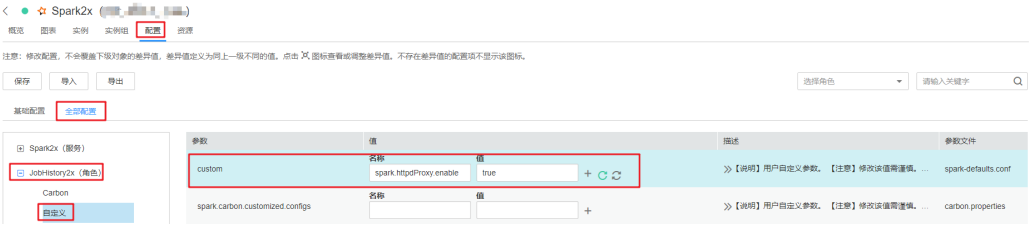
配置项“spark.httpdProxy.enable”未开启。

修改配置项查看日志处理步骤

步骤1 在客户端修改spark-defaults.conf的配置项：spark.httpdProxy.enable = true，该配置项取值默认为false。

步骤2 登录Manager界面，选择“集群 > 服务 > Spark2x > 配置 > 全部配置 > JobHistory2x > 自定义 > custom”，增加自定义配置项：spark.httpdProxy.enable=true。

图 19-3 自定义配置项



步骤3 参考上一步，分别选择“集群 > 服务 > Spark2x > 配置 > 全部配置 > JDBCServer2x > 自定义 > custom”和“集群 > 服务 > Spark2x > 配置 > 全部配置 > SparkResource2x > 自定义 > custom”添加自定义配置：spark.httpdProxy.enable=true。

步骤4 单击“保存”保存配置，并在“集群 > 服务 > Spark2x > 概览”页签右上角，选择“更多> 滚动重启服务”。

步骤5 登录Spark2x原生页面即可单击stdout或者stderr访问到executor日志页面。

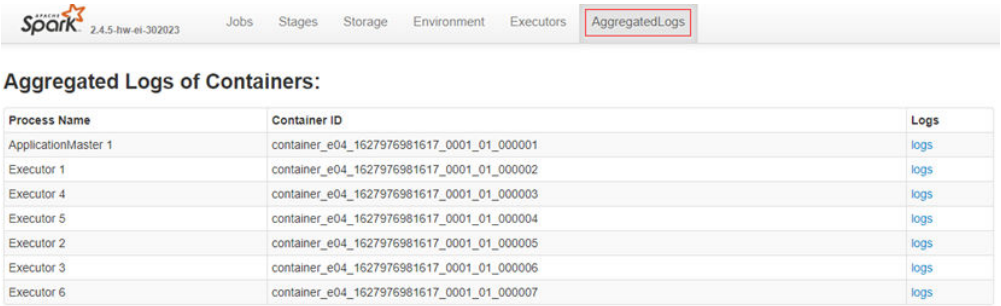
----结束

不修改配置项查看日志处理步骤

- 查看运行完成任务日志

  在Spark2x原生页面最上方有一个按钮，可以查看汇聚后任务日志：**AggregatedLogs**

图 19-4 AggregatedLogs



- 查看运行中的任务日志

  在Spark2x原生页面，通过“Jobs > Stage > Task”找到对应的executor，单击后面的stdout或者stderr查看整个executor日志。

图 19-5 Jobs

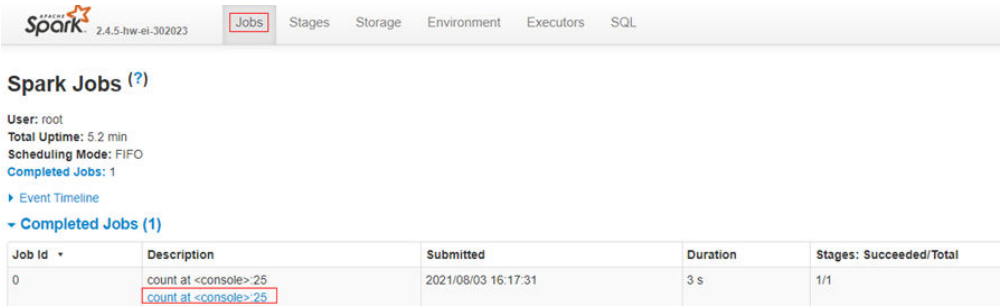


图 19-6 Stage

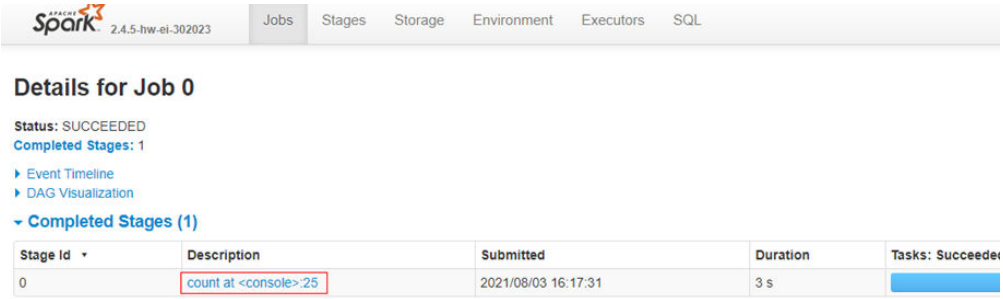
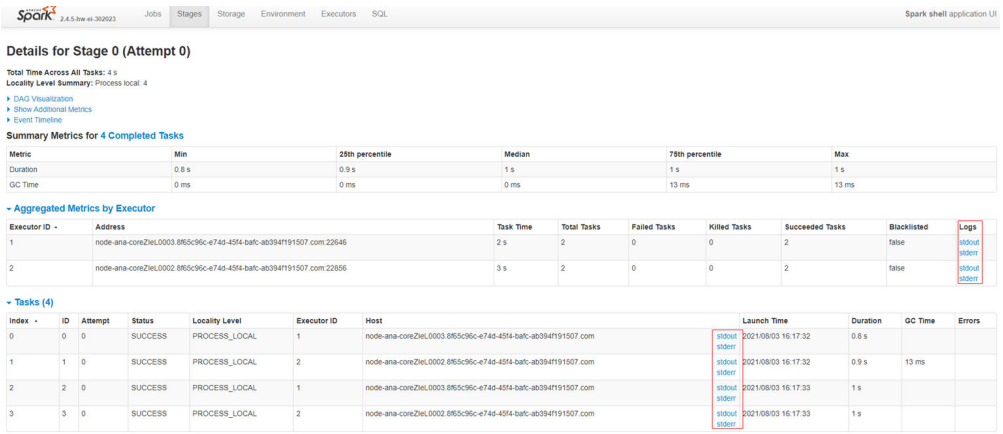


图 19-7 Task



参考信息

通过单击stdout或者stderr进入日志页面以后只会展示stdout或者stderr的日志，如果需要查看完整日志，可以通过将url里面的stdout或者stderr去掉，然后就可以直接访问这个executor的所有日志。

例如：

https://<EIP>:9022/component/Yarn/NodeManager/15/node/containerlogs/container_e04_1627976981617_0002_01_000002/root/**stderr**?start=-4096

修改为：

https://<EIP>:9022/component/Yarn/NodeManager/15/node/containerlogs/container_e04_1627976981617_0002_01_000002/root/?start=-4096

即可看到所有日志文件。

图 19-8 所有日志文件

19.29 spark-beeline 查询 Hive 视图报错

用户问题

MRS 3.1.2版本，spark-beeline查询Hive视图报错，报错如下：

```
0: jdbc:hive2://sysxc-mrs-test-node-master3ROw.mrs-qixy.com:22550/> select * from dim_mcht_organization limit 1
. . . .
```

```
0: jdbc:hive2://sysxc-mrs-test-node-master3ROw.mrs-qiyy.com:22500/> select * from dim_wcht_organization limit 1;
INFO  : Execution ID: 47
+-----------------+-----------------+-----------------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
| organization_id | instance_id     | tenant_id       | user_id | person_id | parent_id | organization_code | organization_name | organization_type | sort_no | description | extension | dr | create_time | create_person | update_time | update_person | organization_node_code | lv | rv | dt |
+-----------------+-----------------+-----------------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
| [redacted]      | [redacted]      | [redacted]      | [redacted] | [redacted] | [redacted] | [redacted] | [redacted] | NULL | 4 | NULL | NULL | 0 | 2020-11-28 22:37:00.0 | [redacted] | [redacted] | [redacted] | [redacted] | [redacted] | [redacted] | [redacted] |
+-----------------+-----------------+-----------------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
```

----结束

19.30 Spark Streaming 任务失败但是 Yarn WebUI 中显示状态为成功

问题现象

MRS 3.3.0-LTS版本集群，提交Spark Streaming任务后，由于NodeAgent重启导致对应节点上任务失败，但是在Yarn WebUI页面中查看整个任务状态为“succeeded”。

										+0800 2024	+0800 2024	+0800 2024				
application_1717547847346_0057	[redacted]	[redacted]	[redacted]	SPARK	default	0	[redacted]	[redacted]	[redacted]	+0800 2024	+0800 2024	+0800 2024	FINISHED	SUCCEEDED		

原因分析

Spark AM退出时，因为服务配置参数“spark.streaming.stopGracefullyOnShutdown”默认为“false”，系统不会完全等待AM正常退出，所以概率性的会出现有时能正常退出，任务状态显示失败，有时异常退出，状态未来得及上报的情况。

处理步骤

执行Spark Streaming任务时，添加Spark任务参数“--conf spark.streaming.stopGracefullyOnShutdown=true”，指定“spark.streaming.stopGracefullyOnShutdown”参数为“true”。

20 使用 Sqoop

20.1 Sqoop 如何连接 MySQL

用户问题

Sqoop如何连接MySQL数据库。

处理步骤

步骤1 在集群上安装客户端，查看客户端“sqoop/lib”目录下是否有MySQL驱动包。

```
[root@node-master1IoKo lib]# ls
ant-contrib-1.0b3.jar              commons-digester-1.8.jar           ivy-2.3.0.jar                          paranamer-2.7.jar
ant-eclipse-1.0-jvm1.2.jar         commons-el-1.0.jar                 jackson-annotations-2.6.3.jar          parquet-avro-1.6.0.jar
avro-1.8.2.jar                     commons-httpclient-3.0.1.jar       jackson-core-2.6.5.jar                 parquet-column-1.6.0.jar
avro-mapred-1.8.2-hadoop2.jar      commons-io-2.4.jar                 jackson-core-asl-1.9.13.jar            parquet-common-1.6.0.jar
calcite-linq4j-1.16.0.jar          commons-jexl-2.1.1.jar             jackson-databind-2.6.5.jar             parquet-encoding-1.6.0.jar
commons-beanutils-1.9.4.jar        commons-lang-2.6.jar               jackson-jaxrs-1.9.13.jar               parquet-format-2.2.0-rc1.jar
commons-beanutils-core-1.8.0.jar   commons-lang3-3.4.jar              jackson-mapper-asl-1.9.13.jar          parquet-generator-1.6.0.jar
commons-cli-1.2.jar                commons-logging-1.2.jar            jackson-xc-1.9.13.jar                  parquet-hadoop-1.6.0.jar
commons-codec-1.9.jar              commons-math-2.2.jar               jline-2.14.6.jar                       parquet-hadoop-bundle-1.8.1.jar
commons-collections-3.2.2.jar      commons-math3-3.1.1.jar            kite-data-core-1.1.0.jar               parquet-jackson-1.6.0.jar
commons-compiler-2.7.6.jar         commons-net-3.1.jar                kite-data-hive-1.1.0.jar               slf4j-api-1.7.10.jar
commons-compress-1.9.jar           commons-pool-1.5.4.jar             kite-data-mapreduce-1.1.0.jar          snappy-java-1.1.1.6.jar
commons-configuration-1.6.jar      commons-vfs2-2.0.jar               kite-hadoop-compatibility-1.1.0.jar    xz-1.5.jar
commons-configuration2-2.1.jar     hadoop-                    .jar    mysql-connector-java-5.1.47.jar
commons-dbcp-1.4.jar               hsqldb-1.8.0.10.jar                opencsv-2.3.jar
[root@node-master1IoKo lib]# pwd
/opt/allclient/Sqoop/sqoop/lib
```

步骤2 在客户端目录下加载环境变量。

source bigdata_env

步骤3 执行Kerberos用户认证。

如果集群已启用Kerberos认证，执行以下命令认证具有操作权限的用户，否则跳过此步骤。

kinit *MRS集群用户*

步骤4 连接数据库。命令中如果携带认证密码信息可能存在安全风险，在执行命令前建议关闭系统的history命令记录功能，避免信息泄露。

sqoop list-databases --connect jdbc:mysql://*数据库IP地址:3306/* **--username** *数据库登录用户名* **--password** *密码*

```
[root@node-master2oDll opt]# sqoop list-databases --connect jdbc:mysql://[redacted]3306/ --username root --password [redacted]
Warning: /opt/hadoopclient/Sqoop/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hadoopclient/HDFS/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/Bigdata/client/HDFS/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoopclient/Hive/HCatalog/lib/slf4j-log4j12-1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoopclient/HBase/hbase/geomesa/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoopclient/HBase/hbase/ranger-2.0.0-hbase-plugin/install/lib/slf4j-log4j12-1.7.30.jar!/org/slf4j/impl/StaticLoggerB
SLF4J: Found binding in [jar:file:/opt/hadoopclient/HBase/hbase/lib/client-facing-thirdparty/slf4j-log4j12-1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.cla
SLF4J: Found binding in [jar:file:/opt/hadoopclient/HBase/hbase/lib/jdbc/slf4j-log4j12-1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoopclient/HBase/hbase/tools/hbase-hbck2-[redacted].jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoopclient/HBase/hbase/tools/hbase-tools-[redacted].jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
2022-01-29 10:56:53,092 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7
2022-01-29 10:56:53,936 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
2022-01-29 10:56:54,132 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
Sat Jan 29 10:56:54 CST 2022 WARN: Establishing SSL connection without server's identity verification is not recommended. According to MySQL 5.5.45+, 5.6.26+
ments SSL connection must be established by default if explicit option isn't set. For compliance with existing applications not using SSL the verifyServerCe
 set to 'false'. You need either to explicitly disable SSL by setting useSSL=false, or set useSSL=true and provide truststore for server certificate verifi
information_schema
bx
mysql
performance_schema
sys
test
vstest
```

上图所示则代表sqoop连接MySQL成功。

----结束

20.2 Sqoop 读取 MySQL 数据到 HBase 时报 HBaseAdmin.<init>方法找不到异常

问题

使用MRS的Sqoop客户端（1.4.7版本），从MySQL数据库中指定表抽取数据，存放到HBase（2.2.3版本）指定的表中，报出异常：

```
Trying to load data into HBASE through Sqoop getting below error.
Exception in thread "main" java.lang.NoSuchMethodError:
org.apache.hadoop.hbase.client.HBaseAdmin.<init>(Lorg/apache/hadoop/conf/Configuration;)V
```

完整异常信息如图所示：

```
and provide truststore for server certificate verification.
2022-01-28 14:37:35,764 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `t_o_eso_users` AS t LIM
IT 1
2022-01-28 14:37:35,786 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `t_o_eso_users` AS t LIM
IT 1
2022-01-28 14:37:35,797 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /opt/Bigdata/client/HDFS/hadoop
Note: /tmp/sqoop-root/compile/792dbda207bec0305d1989403855dfa2/t_o_eso_users.java uses or overrides a deprecated A
PI.
Note: Recompile with -Xlint:deprecation for details.
2022-01-28 14:37:36,678 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-root/compile/792dbda207bec0305d1
989403855dfa2/t_o_eso_users.jar
2022-01-28 14:37:36,691 WARN manager.MySQLManager: It looks like you are importing from mysql.
2022-01-28 14:37:36,691 WARN manager.MySQLManager: This transfer can be faster! Use the --direct
2022-01-28 14:37:36,691 WARN manager.MySQLManager: option to exercise a MySQL-specific fast path.
2022-01-28 14:37:36,691 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)
2022-01-28 14:37:36,716 INFO mapreduce.ImportJobBase: Beginning import of t_o_eso_users
2022-01-28 14:37:36,717 INFO Configuration.deprecation: mapred.job.tracker is deprecated. Instead, use mapreduce.j
obtracker.address
2022-01-28 14:37:36,815 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
2022-01-28 14:37:36,833 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job
.maps
Exception in thread "main" java.lang.NoSuchMethodError: org.apache.hadoop.hbase.client.HBaseAdmin.<init>(Lorg/apac
he/hadoop/conf/Configuration;)V
    at org.apache.sqoop.mapreduce.HBaseImportJob.jobSetup(HBaseImportJob.java:163)
    at org.apache.sqoop.mapreduce.ImportJobBase.runImport(ImportJobBase.java:268)
    at org.apache.sqoop.manager.SqlManager.importTable(SqlManager.java:692)
    at org.apache.sqoop.manager.MySQLManager.importTable(MySQLManager.java:127)
    at org.apache.sqoop.tool.ImportTool.importTable(ImportTool.java:520)
    at org.apache.sqoop.tool.ImportTool.run(ImportTool.java:628)
    at org.apache.sqoop.Sqoop.run(Sqoop.java:147)
    at org.apache.hadoop.util.ToolRunner.run(ToolRunner.java:76)
    at org.apache.sqoop.Sqoop.runSqoop(Sqoop.java:183)
    at org.apache.sqoop.Sqoop.runTool(Sqoop.java:234)
    at org.apache.sqoop.Sqoop.runTool(Sqoop.java:243)
    at org.apache.sqoop.Sqoop.main(Sqoop.java:252)
```

执行Sqoop抽取数据命令样例，命令中如果携带认证密码信息可能存在安全风险，在执行命令前建议关闭系统的history命令记录功能，避免信息泄露。：

```
sqoop import \
--connect jdbc:mysql://MySQL服务器地址:端口号/database1 \
--username admin \
--password xxx \
--table table1 \
--hbase-table table2 \
--column-family info \
```

```
--hbase-row-key id \
--hbase-create-table --m 1
```

处理步骤

Sqoop客户端安装完成之后，没有直接引入HBase相关的依赖jar包，需要通过手动导入指定低版本的HBase相关依赖jar包。解决方法步骤如下：

步骤1 确认Sqoop客户端和HBase客户端是否在同一个路径下。

- 是，执行**步骤2**。
- 否，删除原有的Sqoop和HBase客户端文件，从FusionInsight Manager上下载完整的客户端安装在同一路径下，执行**步骤2**。

步骤2 以**root**用户登录Sqoop客户端安装节点。

步骤3 下载以下HBase 1.6.0版本的jar包上传到Sqoop客户端的“lib”目录下。

- **hbase-client-1.6.0.jar**
- **hbase-common-1.6.0.jar**
- **hbase-protocol-1.6.0.jar**
- **hbase-server-1.6.0.jar**

步骤4 上传包之后，修改包的权限，可以设置为755，具体执行命令为：

chmod 755 *包名称*

步骤5 在客户端目录下执行以下命令刷新Sqoop客户端：

source bigdata_env

重新执行sqoop命令。

----结束

20.3 通过 Hue 创建 Sqoop 任务将数据从 HBase 导入 HDFS 时报错

用户问题

利用Hue的Sqoop操作把HBase中的数据导入HDFS时报错：

```
Caused by: java.lang.ClassNotFoundException: org.apache.htrace.Trace
```

注意

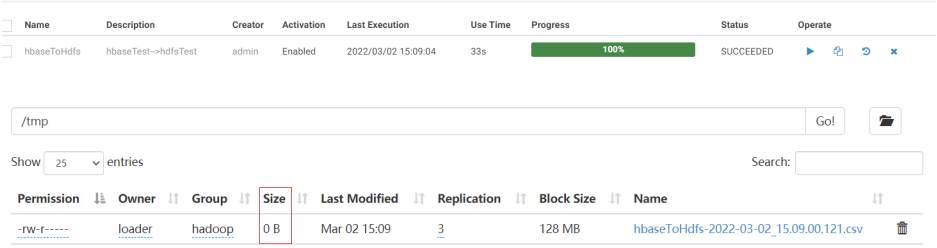
本章节仅适用于MRS 1.9.2版本集群。

```
2022-03-02 15:09:00,264 [main] ERROR org.apache.sqoop.connector.hbase.HBaseExtractor  - An exceptional condition has occurred.
org.apache.sqoop.common.SqoopException: HBASE_CONNECTOR_0011:Failed to open table.
        at org.apache.sqoop.connector.hbase.HBaseExtractor.openDB(HBaseExtractor.java:239)
        at org.apache.sqoop.connector.hbase.HBaseExtractor.access$100(HBaseExtractor.java:34)
        at org.apache.sqoop.connector.hbase.HBaseExtractor$1.run(HBaseExtractor.java:86)
        at org.apache.sqoop.connector.hbase.HBaseExtractor$1.run(HBaseExtractor.java:76)
        at org.apache.sqoop.connector.hbase.HBaseExtractor.extract(HBaseExtractor.java:114)
        at org.apache.sqoop.connector.hbase.HBaseExtractor.extract(HBaseExtractor.java:34)
        at org.apache.sqoop.job.mr.SqoopMapper.runInternal(SqoopMapper.java:156)
        at org.apache.sqoop.job.mr.SqoopMapper.run(SqoopMapper.java:79)
        at org.apache.hadoop.mapred.MapTask.runNewMapper(MapTask.java:787)
        at org.apache.hadoop.mapred.MapTask.run(MapTask.java:341)
        at org.apache.hadoop.mapred.YarnChild$2.run(YarnChild.java:188)
        at java.security.AccessController.doPrivileged(Native Method)
        at javax.security.auth.Subject.doAs(Subject.java:422)
        at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1840)
        at org.apache.hadoop.mapred.YarnChild.main(YarnChild.java:182)
Caused by: java.lang.reflect.InvocationTargetException
```

```
Caused by: java.lang.reflect.InvocationTargetException
        at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)
        at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:62)
        at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:45)
        at java.lang.reflect.Constructor.newInstance(Constructor.java:423)
        at org.apache.hadoop.hbase.client.ConnectionFactory.createConnection(ConnectionFactory.java:238)
        at org.apache.hadoop.hbase.client.ConnectionManager.createConnection(ConnectionManager.java:454)
        at org.apache.hadoop.hbase.client.ConnectionManager.createConnection(ConnectionManager.java:447)
        at org.apache.hadoop.hbase.client.ConnectionManager.getConnectionInternal(ConnectionManager.java:325)
        at org.apache.hadoop.hbase.client.HTable.<init>(HTable.java:184)
        at org.apache.hadoop.hbase.client.HTable.<init>(HTable.java:150)
        at org.apache.sqoop.connector.hbase.HBaseExtractor.openDB(HBaseExtractor.java:236)
        ... 14 more
Caused by: java.lang.NoClassDefFoundError: org/apache/htrace/Trace
        at org.apache.hadoop.hbase.zookeeper.RecoverableZooKeeper.exists(RecoverableZooKeeper.java:245)
        at org.apache.hadoop.hbase.zookeeper.ZKUtil.checkExists(ZKUtil.java:436)
        at org.apache.hadoop.hbase.zookeeper.ZKClusterId.readClusterIdZNode(ZKClusterId.java:65)
        at org.apache.hadoop.hbase.client.ZooKeeperRegistry.getClusterId(ZooKeeperRegistry.java:105)
        at org.apache.hadoop.hbase.client.ConnectionManager$HConnectionImplementation.retrieveClusterId(ConnectionManager.java:944)
        at org.apache.hadoop.hbase.client.ConnectionManager$HConnectionImplementation.<init>(ConnectionManager.java:720)
        ... 25 more
Caused by: java.lang.ClassNotFoundException: org.apache.htrace.Trace
        at java.net.URLClassLoader.findClass(URLClassLoader.java:382)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:419)
```

问题现象

Sqoop任务运行成功，但HDFS中的CSV文件无内容。



原因分析

由于jar包冲突或者缺少相关jar包导致。

处理步骤

步骤1 进入Sqoop的安装目录下查找文件。

1. 进入Sqoop节点的“/opt/Bigdata/MRS_1.9.2/install/FusionInsight-Sqoop-1.99.7/FusionInsight-Sqoop-1.99.7/server/lib”目录下，进行grep查找。

```
[root@node-master1PMPi lib]# pwd
/opt/Bigdata/MRS_1.9.2/install/FusionInsight-Sqoop-1.99.7/FusionInsight-Sqoop-1.99.7/server/lib
[root@node-master1PMPi lib]# grep org.apache.htrace.Trace *
Binary file htrace-core-3.1.0-incubating.jar matches
[root@node-master1PMPi lib]#
```

2. 进入Yarn WebUI界面，查看运行的任务的报错具体信息。

```
Log Type: syslog
Log Upload Time: Thu Mar 03 15:19:29 +0800 2022
Log Length: 74284
2022-03-03 15:19:08,177 INFO [main] org.apache.hadoop.mapreduce.v2.app.MRAppMaster: Created MRAppMaster for application appattempt_1646291845172_
2022-03-03 15:19:08,357 INFO [main] org.apache.hadoop.mapreduce.v2.app.MRAppMaster:
/************************************************************
[system properties]
os.name: Linux
os.version: 3.10.0-327.62.59.83.h162.x86_64
java.home: /opt/Bigdata/jdk1.8.0_232/jre
java.runtime.version: 1.8.0_232-[redacted]TDK_V100R001C00SPC173B001-b09
java.vendor: [redacted] Technologies Co., Ltd
java.version: 1.8.0_232
java.vm.name: OpenJDK 64-Bit Server VM
java.class.path: /srv/BigData/hadoop/data1/nm/localdir/usercache/loader/appcache/application_1646291845172_0001/container_e01_1646291845172_0001
java.io.tmpdir: /srv/BigData/hadoop/data1/nm/localdir/usercache/loader/appcache/application_1646291845172_0001/container_e01_1646291845172_0001_0
user.dir: /srv/BigData/hadoop/data1/nm/localdir/usercache/loader/appcache/application_1646291845172_0001/container_e01_1646291845172_0001_01_0000
user.name:
************************************************************/
```

3. 将java.class.path复制出来，搜索htrace-core。

```
Bigdata/MRS_1.9.2/install/FusionInsight-Hadoop-2.8.3/hadoop/share/hadoop/tools/lib/activation-1.1.jar:/opt/Bigdata/MRS_1.9.2/install/FusionInsight-Hadoop-2.8.3/
hadoop/share/hadoop/tools/lib/hadoop-aws-2.8.3-mrs-1.9.0.jar:/opt/Bigdata/MRS_1.9.2/install/FusionInsight-Hadoop-2.8.3/hadoop/share/hadoop/tools/lib/
joda-time-2.9.4.jar:/opt/Bigdata/MRS_1.9.2/install/FusionInsight-Hadoop-2.8.3/hadoop/share/hadoop/tools/lib/hadoop-yarn-server-common-2.8.3-mrs-1.9.0.jar:/opt/
Bigdata/MRS_1.9.2/install/FusionInsight-Hadoop-2.8.3/hadoop/share/hadoop/tools/lib/curator-framework-2.7.1.jar:/opt/Bigdata/MRS_1.9.2/install/
FusionInsight-Hadoop-2.8.3/hadoop/share/hadoop/tools/lib/jettison-1.1.jar:/opt/Bigdata/MRS_1.9.2/install/FusionInsight-Hadoop-2.8.3/hadoop/share/hadoop/tools/lib/
servlet-api-2.5.jar:/opt/Bigdata/MRS_1.9.2/install/FusionInsight-Hadoop-2.8.3/hadoop/share/hadoop/tools/lib/guava-11.0.2.jar:/opt/Bigdata/MRS_1.9.2/install/
FusionInsight-Hadoop-2.8.3/hadoop/share/hadoop/tools/lib/apache-log4j-extras-1.1.jar:/opt/Bigdata/MRS_1.9.2/install/FusionInsight-Hadoop-2.8.3/hadoop/share/hadoop
/tools/lib/htrace-core4-4.0.1-incubating.jar:/opt/Bigdata/MRS_1.9.2/install/FusionInsight-Hadoop-2.8.3/hadoop/share/hadoop/tools/lib/jetty-util-6.1.26.jar:/opt/
Bigdata/MRS_1.9.2/install/FusionInsight-Hadoop-2.8.3/hadoop/share/hadoop/tools/lib/hadoop-gridmix-2.8.3-mrs-1.9.0.jar:/opt/Bigdata/MRS_1.9.2/install/
FusionInsight-Hadoop-2.8.3/hadoop/share/hadoop/tools/lib/jsp-api-2.1.jar:/opt/Bigdata/MRS_1.9.2/install/FusionInsight-Hadoop-2.8.3/hadoop/share/hadoop/tools/lib/
hadoop-sls-2.8.3-mrs-1.9.0.jar:/opt/Bigdata/MRS_1.9.2/install/FusionInsight-Hadoop-2.8.3/hadoop/share/hadoop/tools/lib/commons-lang-2.6.jar:/opt/Bigdata/MRS_1.9.2
/install/FusionInsight-Hadoop-2.8.3/hadoop/share/hadoop/tools/lib/netty-all-4.0.23.Final.jar:/opt/Bigdata/MRS_1.9.2/install/FusionInsight-Hadoop-2.8.3/hadoop/
share/hadoop/tools/lib/zookeeper-3.5.1-mrs-1.9.0.jar:/opt/Bigdata/MRS_1.9.2/install/FusionInsight-Hadoop-2.8.3/hadoop/share/hadoop/tools/lib/
Aa "" htrace Find Find Prev
```

4. 复制jar包到如下位置。

   cp /opt/Bigdata/MRS_1.9.2/install/FusionInsight-Sqoop-1.99.7/FusionInsight-Sqoop-1.99.7/server/lib/*htrace-core-3.1.0-incubating.jar* **/opt/Bigdata/MRS_1.9.2/install/FusionInsight-Hadoop-2.8.3/hadoop/share/hadoop/common/lib/**

5. 修改权限。

   chmod 777 *htrace-core-3.1.0-incubating.jar*（真实复制的jar包）

   chown omm:ficommon *htrace-core-3.1.0-incubating.jar*（真实复制的jar包）

6. 查看hosts文件，对其他所有节点进行同样的复制jar包操作。

```
[root@node-master1PMPi lib]# cat /etc/hosts
::1         localhost       localhost.localdomain   localhost6
127.0.0.1       localhost       localhost.localdomain   localh
127.0.0.1       image-pipeline-1004600  image-pipeline-100460
127.0.0.1       ecs-73f1-191-base       ecs-73f1-191-base
1.1.1.1 hadoop.d0edba23_74ce_4527_9e04_22bc21853bb9.com
1.1.1.1     hadoop.hadoop.com
1.1.1.1     hacluster
1.1.1.1     haclusterX
1.1.1.1     haclusterX1
1.1.1.1     haclusterX2
1.1.1.1     haclusterX3
1.1.1.1     haclusterX4
1.1.1.1     ClusterX
1.1.1.1     manager
192.168.9.152 node-master1PMPi.mrs-5s0w.com
192.168.9.200 node-ana-coretmgV.mrs-5s0w.com
[root@node-master1PMPi lib]#
```

7. 重新运行sqoop任务，产生报错如下：

```
        at java.lang.Thread.run(Thread.java:745)
Caused by: com.google.protobuf.ServiceException: java.lang.NoClassDefFoundError: com/yammer/metrics/core/Gauge
        at org.apache.hadoop.hbase.ipc.AbstractRpcClient.callBlockingMethod(AbstractRpcClient.java:240)
        at org.apache.hadoop.hbase.ipc.AbstractRpcClient$BlockingRpcChannelImplementation.callBlockingMethod(AbstractRp
        at org.apache.hadoop.hbase.protobuf.generated.ClientProtos$ClientService$BlockingStub.scan(ClientProtos.java:35
        at org.apache.hadoop.hbase.client.ClientSmallReversedScanner$SmallReversedScannerCallable.call(ClientSmallRever
        ... 9 more
Caused by: java.lang.NoClassDefFoundError: com/yammer/metrics/core/Gauge
        at org.apache.hadoop.hbase.ipc.AbstractRpcClient.callBlockingMethod(AbstractRpcClient.java:225)
        ... 12 more
Caused by: java.lang.ClassNotFoundException: com.yammer.metrics.core.Gauge
        at java.net.URLClassLoader.findClass(URLClassLoader.java:382)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:419)
        at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:352)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:352)
        ... 13 more
2022-03-03 15:45:01,714 [main] INFO  org.apache.sqoop.job.mr.SqoopMapper  - Extractor has finished
2022-03-03 15:45:01,715 [main] INFO  org.apache.sqoop.job.mr.SqoopMapper  - Stopping progress service
2022-03-03 15:45:01,727 [main] INFO  org.apache.sqoop.job.mr.SqoopOutputFormatLoadExecutor  - SqoopOutputFormatLoadExec
2022-03-03 15:45:01,776 [OutputFormatLoader-consumer] INFO  org.apache.sqoop.job.mr.SqoopOutputFormatLoadExecutor  - Lo
2022-03-03 15:45:01,777 [main] INFO  org.apache.sqoop.job.mr.SqoopOutputFormatLoadExecutor  - SqoopOutputFormatLoadExec

Log Type: stdout
Log Upload Time: Thu Mar 03 15:45:15 +0800 2022
Log Length: 0

Log Type: syslog
```

步骤2 去HBase的安装目录下查找文件。

1. 进入HBase的lib目录下，进行grep查找。

```
[root@node-master1PMPi lib]# pwd
/opt/Bigdata/MRS_1.9.2/install/FusionInsight-HBase-1.3.1/hbase/lib
[root@node-master1PMPi lib]# grep com.yammer.metrics.core.Gauge *
grep: jline: Is a directory
Binary file metrics-core-2.2.0.jar matches
grep: native: Is a directory
grep: ruby: Is a directory
grep: ruby_luna: Is a directory
[root@node-master1PMPi lib]#
```

2. 继续复制jar包。

   cp /opt/Bigdata/MRS_1.9.2/install/FusionInsight-HBase-1.3.1/hbase/lib/*metrics-core-2.2.0.jar* /opt/Bigdata/MRS_1.9.2/install/FusionInsight-Hadoop-2.8.3/hadoop/share/hadoop/common/lib/

3. 修改文件权限。

   chmod 777 *metrics-core-2.2.0.jar*（真实复制的jar包）

   chown omm:ficommon *metrics-core-2.2.0.jar*（真实复制的jar包）

4. 查看hosts文件，对其他所有节点进行同样的复制jar包操作。
5. 继续运行sqoop任务，成功。

```
2022-03-03 15:50:16,923 INFO [main] org.apache.zookeeper.ZooKeeper: Session: 0xf0000078a3e0c58 closed
2022-03-03 15:50:16,924 INFO [main-EventThread] org.apache.zookeeper.ClientCnxn: EventThread shut down for session: 0xf0000078a3e0c58
2022-03-03 15:50:16,934 INFO [main] org.apache.sqoop.job.mr.SqoopMapper: Extractor has finished
2022-03-03 15:50:16,935 INFO [main] org.apache.sqoop.job.mr.SqoopMapper: Stopping progress service
2022-03-03 15:50:16,942 INFO [main] org.apache.sqoop.job.mr.SqoopOutputFormatLoadExecutor: SqoopOutputFormatLoadExecutor::SqoopRecordWriter is about to be closed
2022-03-03 15:50:17,397 INFO [OutputFormatLoader-consumer] org.apache.sqoop.job.mr.SqoopOutputFormatLoadExecutor: Loader has finished
2022-03-03 15:50:17,398 INFO [main] org.apache.sqoop.job.mr.SqoopOutputFormatLoadExecutor: SqoopOutputFormatLoadExecutor::SqoopRecordWriter is closed
2022-03-03 15:50:17,398 INFO [main] org.apache.hadoop.mapred.Task: Task:attempt_1646292920879_0002_m_000000_0 is done. And is in the process of committing
2022-03-03 15:50:17,435 INFO [main] org.apache.hadoop.mapred.Task: Task 'attempt_1646292920879_0002_m_000000_0' done.
2022-03-03 15:50:17,437 INFO [main] org.apache.hadoop.mapred.Task: Final Counters for attempt_1646292920879_0002_m_000000_0: Counters: 26
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=662083
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=107
                HDFS: Number of bytes written=10
                HDFS: Number of read operations=1
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=1
        Map-Reduce Framework
                Map input records=0
                Map output records=1
                Input split bytes=107
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=239
                CPU time spent (ms)=2200
                Physical memory (bytes) snapshot=669523968
                Virtual memory (bytes) snapshot=2697564160
                Total committed heap usage (bytes)=600834048
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=0
        org.apache.sqoop.submission.counter.SqoopCounters
                FILES_WRITTEN=1
                ROWS_READ=1
                ROWS_WRITTEN=1
2022-03-03 15:50:17,538 INFO [main] org.apache.hadoop.metrics2.impl.MetricsSystemImpl: Stopping MapTask metrics system...
2022-03-03 15:50:17,538 INFO [main] org.apache.hadoop.metrics2.impl.MetricsSystemImpl: MapTask metrics system stopped.
2022-03-03 15:50:17,538 INFO [main] org.apache.hadoop.metrics2.impl.MetricsSystemImpl: MapTask metrics system shutdown complete.
```

----结束

处理总结

1. 将sqoop的lib下htrace-core-3.1.0-incubating.jar和hbase的lib下的metrics-core-2.2.0.jar，复制到“/opt/Bigdata/MRS_1.9.2/install/FusionInsight-Hadoop-2.8.3/hadoop/share/hadoop/common/lib/”下。
2. 修改jar包的文件权限为777和omm:ficommon。
3. 所有节点均采取以上操作，重新运行sqoop任务即可。

20.4 通过 Sqoop 从 Hive 导出数据到 MySQL 8.0 时报数据格式错误

本章节仅适用于MRS 3.1.0版本集群。

用户问题

MRS 3.1.0集群Sqoop从Hive导出数据到MySQL 8.0时报数据格式错误。

问题现象

```
2022-03-31 19:56:44,581 ERROR mapreduce.ExportJobBase: Export job failed!
2022-03-31 19:56:44,581 ERROR tool.ExportTool: Error during export:
Export job failed!
        at org.apache.sqoop.mapreduce.ExportJobBase.runExport(ExportJobBase.java:445)
        at org.apache.sqoop.manager.SqlManager.exportTable(SqlManager.java:931)
        at org.apache.sqoop.tool.ExportTool.exportTable(ExportTool.java:80)
        at org.apache.sqoop.tool.ExportTool.run(ExportTool.java:99)
        at org.apache.sqoop.Sqoop.run(Sqoop.java:147)
        at org.apache.hadoop.util.ToolRunner.run(ToolRunner.java:76)
        at org.apache.sqoop.Sqoop.runSqoop(Sqoop.java:183)
        at org.apache.sqoop.Sqoop.runTool(Sqoop.java:234)
        at org.apache.sqoop.Sqoop.runTool(Sqoop.java:243)
        at org.apache.sqoop.Sqoop.main(Sqoop.java:252)
```

```
2022-03-31 18:58:00,000 | INFO  | main | map.input.length is deprecated. Instead, use mapreduce.map.input.length | Configuration.java:1409
2022-03-31 18:58:01,001 | ERROR | main |  | TextExportMapper.java:93
2022-03-31 18:58:01,001 | ERROR | main | Exception raised during data export | TextExportMapper.java:94
2022-03-31 18:58:01,001 | ERROR | main |  | TextExportMapper.java:95
2022-03-31 18:58:01,001 | ERROR | main | Exception:  | TextExportMapper.java:97
java.lang.RuntimeException: Can't parse input data: 'kg'
        at hkatg_agr_prod_city_summ__loadFromFields(hkatg_agr_prod_city_summ.java:911)
        at hkatg_agr_prod_city_summ.parse(hkatg_agr_prod_city_summ.java:738)
        at org.apache.sqoop.mapreduce.TextExportMapper.map(TextExportMapper.java:89)
        at org.apache.sqoop.mapreduce.TextExportMapper.map(TextExportMapper.java:39)
        at org.apache.hadoop.mapreduce.Mapper.run(Mapper.java:146)
        at org.apache.sqoop.mapreduce.AutoProgressMapper.run(AutoProgressMapper.java:64)
        at org.apache.hadoop.mapred.MapTask.runNewMapper(MapTask.java:799)
        at org.apache.hadoop.mapred.MapTask.run(MapTask.java:347)
        at org.apache.hadoop.mapred.YarnChild$1.run(YarnChild.java:183)
        at java.security.AccessController.doPrivileged(Native Method)
        at javax.security.auth.Subject.doAs(Subject.java:422)
        at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1761)
        at org.apache.hadoop.mapred.YarnChild.main(YarnChild.java:177)
Caused by: java.lang.NumberFormatException
        at java.math.BigDecimal.<init>(BigDecimal.java:497)
        at java.math.BigDecimal.<init>(BigDecimal.java:383)
        at java.math.BigDecimal.<init>(BigDecimal.java:809)
        at hkatg_agr_prod_city_summ__loadFromFields(hkatg_agr_prod_city_summ.java:872)
        ... 12 more
2022-03-31 18:58:01,001 | ERROR | main | Dumping data is not allowed by default, please run the job with -Dorg.apache.sqoop.export.text.dump_data_on_error=true to get corrupted line. | TextExportMapper.java:101
2022-03-31 18:58:01,001 | ERROR | main | On input file: hdfs://hacluster/user/hive/warehouse/dm_agr_prod_city_summ02/000000_0 | TextExportMapper.java:108
2022-03-31 18:58:01,001 | ERROR | main | At position 0 | TextExportMapper.java:111
2022-03-31 18:58:01,001 | ERROR | main |  | TextExportMapper.java:113
2022-03-31 18:58:01,001 | ERROR | main | Currently processing split: | TextExportMapper.java:114
2022-03-31 18:58:01,001 | ERROR | main | Paths:/user/hive/warehouse/dm_agr_prod_city_summ02/000000_0:0+26 | TextExportMapper.java:115
2022-03-31 18:58:01,001 | ERROR | main |  | TextExportMapper.java:117
2022-03-31 18:58:01,001 | ERROR | main | This issue might not necessarily be caused by current input | TextExportMapper.java:118
2022-03-31 18:58:01,001 | ERROR | main | due to the batching nature of export. | TextExportMapper.java:119
2022-03-31 18:58:01,001 | ERROR | main |  | TextExportMapper.java:120
2022-03-31 18:58:01,001 | INFO  | Thread-17 | Auto-progress thread is finished. keepGoing=false | ProgressThread.java:158
2022-03-31 18:58:01,001 | WARN  | main | Exception running child : java.io.IOException: Can't export data, please check failed map task logs
        at org.apache.sqoop.mapreduce.TextExportMapper.map(TextExportMapper.java:122)
        at org.apache.sqoop.mapreduce.TextExportMapper.map(TextExportMapper.java:39)
        at org.apache.hadoop.mapreduce.Mapper.run(Mapper.java:146)
        at org.apache.sqoop.mapreduce.AutoProgressMapper.run(AutoProgressMapper.java:64)
        at org.apache.hadoop.mapred.MapTask.runNewMapper(MapTask.java:799)
        at org.apache.hadoop.mapred.MapTask.run(MapTask.java:347)
        at org.apache.hadoop.mapred.YarnChild$1.run(YarnChild.java:183)
        at java.security.AccessController.doPrivileged(Native Method)
        at javax.security.auth.Subject.doAs(Subject.java:422)
        at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1761)
        at org.apache.hadoop.mapred.YarnChild.main(YarnChild.java:177)
Caused by: java.lang.RuntimeException: Can't parse input data: 'kg'
        at hkatg_agr_prod_city_summ__loadFromFields(hkatg_agr_prod_city_summ.java:911)
        at hkatg_agr_prod_city_summ.parse(hkatg_agr_prod_city_summ.java:738)
        at org.apache.sqoop.mapreduce.TextExportMapper.map(TextExportMapper.java:89)
        ... 10 more
Caused by: java.lang.NumberFormatException
        at java.math.BigDecimal.<init>(BigDecimal.java:497)
        at java.math.BigDecimal.<init>(BigDecimal.java:383)
        at java.math.BigDecimal.<init>(BigDecimal.java:809)
        at hkatg_agr_prod_city_summ__loadFromFields(hkatg_agr_prod_city_summ.java:872)
        ... 12 more
 | YarnChild.java:199
2022-03-31 18:58:01,001 | INFO  | main | Running cleanup for the task | Task.java:1484
2022-03-31 18:58:01,001 | INFO  | main | Stopping MapTask metrics system... | MetricsSystemImpl.java:210
2022-03-31 18:58:01,001 | INFO  | main | MapTask metrics system stopped. | MetricsSystemImpl.java:216
2022-03-31 18:58:01,001 | INFO  | main | MapTask metrics system shutdown complete. | MetricsSystemImpl.java:611
```

原因分析

通过日志判断得知是格式异常。

处理步骤

确认分隔符、表字段的格式无问题，在sqoop语句中添加--columns绑定对应字段。

sqoop export --connect jdbc:mysql://*数据库IP地址:端口号/数据库名* **--username** *数据库用户名* **--password** *密码* **--table** *表名* **--columns** *列字段（多个列用英文逗号分开）* **-export-dir** *导出地址* **--fields-terminated-by** *分隔符* **--input-null-string '\\N' --input-null-non-string '\\N' -m 1**

命令中如果携带认证密码信息可能存在安全风险，在执行命令前建议关闭系统的history命令记录功能，避免信息泄露。

样例：

sqoop export --connect jdbc:mysql://192.168.0.6:3306/lidengpeng --username root --password *用户密码* **--table hkatg_agr_prod_city_summ --columns year,city_name,city_code,prod_code,prod_name,prod_type,sown_area,area_unit,yield_wegt,yield_unit,total_wegt,total_wegt_unit,data_sorc_code,etl_time -export-dir hdfs://hacluster/user/hive/warehouse/dm_agr_prod_city_summ02 --fields-terminated-by ',' --input-null-string '\\N' --input-null-non-string '\\N' -m 1**

20.5 通过 sqoop import 命令从 PgSQL 导出数据到 Hive 时报错

背景

使用sqoop import命令抽取开源PgSQL数据库内容到MRS HDFS或Hive中。

用户问题

使用sqoop命令可以查询PgSQL表，但是执行sqoop import命令导入导出时报错：

```
The authentication type 5 is not supported. Check that you have configured the pg_hba.conf file to include
the client's IP address or subnet。
```

原因分析

1. 连接PgSQL MD5认证不通过，需要在“pg_hba.conf”中配置白名单。
2. 在执行sqoop import命令时，会启动MapReduce任务，由于MRS Hadoop安装目下（例如“/opt/Bigdata/FusionInsight_HD_*/1_*_NodeManager/install/hadoop/share/hadoop/common/lib”）自带了PgSQL驱动包“gsjdbc4-*.jar”，与开源PgSQL服务不兼容导致报错。

处理步骤

步骤1 用户在“pg_hba.conf”文件配置白名单。

步骤2 驱动重复，集群自带，将其余驱动清除，所有Core节点上的gsjdbc4 jar包去掉，在sqoop/lib目录下添加PgSQL Jar包即可。

mv /opt/Bigdata/FusionInsight_HD_*/1_*_NodeManager/install/hadoop/share/hadoop/common/lib/gsjdbc4-*.jar /tmp

```
]$ mv /opt/Bigdata/FusionInsight_HD_8.1.0.1/1_2_NodeManager/install/hadoop/share/hadoop/common/lib/gsjdbc4-V100R003C10SPC125.jar /tmp
]$ exit
```

----结束

20.6 通过 Sqoop 读取 MySQL 数据并写 parquet 文件到 OBS 时失败

用户问题

通过Sqoop读取MySQL数据，然后直接写到OBS，指定parquet格式时写入报错，不指定parquet时不报错。

问题现象

```
2022-02-09 16:36:53,393 ERROR sqoop.Sqoop: Got exception running Sqoop: org.kitesdk.data.DatasetNotFoundException: Unknown dataset URI pattern: dataset:obs://for
mrs/user/hive/warehouse/dws.db/dws_ks_vip_user_valid_member_i_d/pts=2022-01-09/part-00000-e6a4dd58-f01b-4d0d-906d-3b515815811e.c000
Check that JARs for obs datasets are on the classpath
org.kitesdk.data.DatasetNotFoundException: Unknown dataset URI pattern: dataset:obs://formrs/user/hive/warehouse/dws.db/dws_ks_vip_user_valid_member_i_d/pts=2022
-01-09/part-00000-e6a4dd58-f01b-4d0d-906d-3b515815811e.c000
Check that JARs for obs datasets are on the classpath
        at org.kitesdk.data.spi.Registration.lookupDatasetUri(Registration.java:128)
        at org.kitesdk.data.Datasets.load(Datasets.java:103)
        at org.kitesdk.data.Datasets.load(Datasets.java:140)
        at org.kitesdk.data.mapreduce.DatasetKeyInputFormat$ConfigBuilder.readFrom(DatasetKeyInputFormat.java:92)
        at org.kitesdk.data.mapreduce.DatasetKeyInputFormat$ConfigBuilder.readFrom(DatasetKeyInputFormat.java:139)
        at org.apache.sqoop.mapreduce.JdbcExportJob.configureInputFormat(JdbcExportJob.java:83)
        at org.apache.sqoop.mapreduce.ExportJobBase.runExport(ExportJobBase.java:434)
        at org.apache.sqoop.manager.SqlManager.exportTable(SqlManager.java:931)
        at org.apache.sqoop.tool.ExportTool.exportTable(ExportTool.java:80)
        at org.apache.sqoop.tool.ExportTool.run(ExportTool.java:99)
        at org.apache.sqoop.Sqoop.run(Sqoop.java:147)
        at org.apache.hadoop.util.ToolRunner.run(ToolRunner.java:76)
        at org.apache.sqoop.Sqoop.runSqoop(Sqoop.java:183)
        at org.apache.sqoop.Sqoop.runTool(Sqoop.java:234)
        at org.apache.sqoop.Sqoop.runTool(Sqoop.java:243)
        at org.apache.sqoop.Sqoop.main(Sqoop.java:252)
2022-02-09 16:36:53,398 WARN metrics.OBSAMetricsProvider: Fetch slotId failed.
[root@ecs-gateway mrsclient]#
[root@ecs-gateway mrsclient]# sqoop export --connect jdbc:mysql://10.50.160.241:3306/data_market --username root --password ******** --table dws_ks_vip_user_vali
d_member_test_export -export-dir obs://formrs/user/hive/warehouse/dws.db/dws_ks_vip_user_valid_member_i_d/pts=2022-01-09/part-00000-e6a4dd58-f01b-4d0d-906d-3b515
815811e.c000 --fields-terminated-by ',' -m 1
```

原因分析

parquet格式不支持hive3，用Hcatalog方式写入。

处理步骤

采用Hcatalog的方式，参数指定对应的Hive库和表，需要修改SQL语句指定到具体字段（需要修改脚本）。命令中如果携带认证密码信息可能存在安全风险，在执行命令前建议关闭系统的history命令记录功能，避免信息泄露。

原来的脚本：

```
sqoop import --connect 'jdbc:mysql://10.160.5.65/huawei_pos_online_00?
zeroDateTimeBehavior=convertToNull' --username root --password xxx
--split-by id
--num-mappers 2
--query 'select * from pos_remark where 1=1 and $CONDITIONS'
--target-dir obs://za-test/dev/huawei_pos_online_00/pos_remark
--delete-target-dir
--null-string '\\N‘
--null-non-string '\\N‘
--as-parquetfile
```

修改后的脚本（可以执行成功）：

```
sqoop import --connect 'jdbc:mysql://10.160.5.65/huawei_pos_online_00?
zeroDateTimeBehavior=convertToNull' --username root --password xxx
--split-by id
--num-mappers 2
--query 'select
id,pos_case_id,pos_transaction_id,remark,update_time,update_user,is_deleted,creator,modifier,gmt_created,g
mt_modified,update_user_id,tenant_code from pos_remark where 1=1 and $CONDITIONS‘
--hcatalog-database huawei_dev
--hcatalog-table ods_pos_remark
```

20.7 通过 Sqoop 迁移数据库数据时报错

用户问题

- MRS Sqoop迁移数据库数据报错如下：
  ```
  Communications link failure;
  The driver has not received any packets from the server;
  ```
- 源端为DWS数据库时报错如下：
  ```
  got exception running sqoop .java.lang.Runtime.Exception, could not load db driver class。
  ```

图 20-1 Sqoop 迁移数据库数据报错

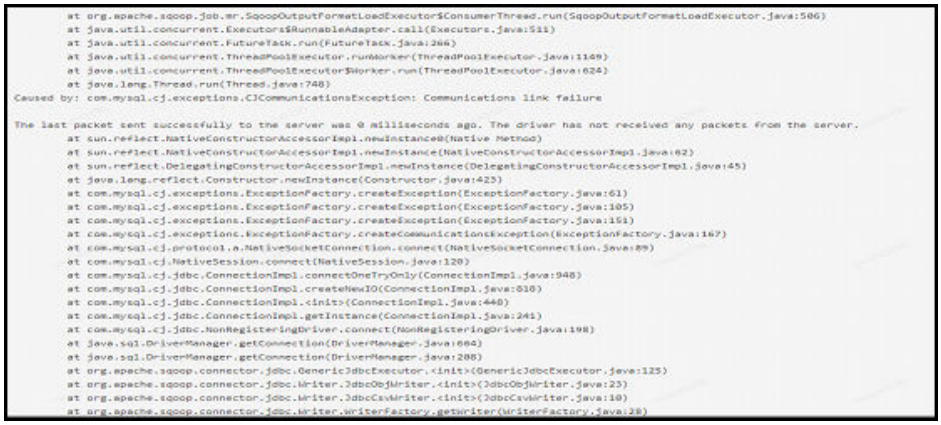


图 20-2 源端为 DWS 时报错

```
 (via --connection-manager). Sqoop is going to fall back to org.apache.sqoop.manager.GenericJdbcManager. Please specify explicitly which connection man
ager should be used next time.
2021-12-23 16:22:53,122 INFO manager.SqlManager: Using default fetchSize of 1000
2021-12-23 16:22:53,122 INFO tool.CodeGenTool: Beginning code generation
2021-12-23 16:22:53,143 ERROR sqoop.Sqoop: Got exception running Sqoop: java.lang.RuntimeException: Could not load db driver class: com.[redacted].gauss200
.jdbc
java.lang.RuntimeException: Could not load db driver class: com.[redacted].gauss200.jdbc
    at org.apache.sqoop.manager.SqlManager.makeConnection(SqlManager.java:875)
    at org.apache.sqoop.manager.GenericJdbcManager.getConnection(GenericJdbcManager.java:59)
    at org.apache.sqoop.manager.SqlManager.execute(SqlManager.java:763)
    at org.apache.sqoop.manager.SqlManager.execute(SqlManager.java:786)
    at org.apache.sqoop.manager.SqlManager.getColumnInfoForRawQuery(SqlManager.java:289)
    at org.apache.sqoop.manager.SqlManager.getColumnTypesForRawQuery(SqlManager.java:260)
    at org.apache.sqoop.manager.SqlManager.getColumnTypesForQuery(SqlManager.java:253)
    at org.apache.sqoop.manager.ConnManager.getColumnTypes(ConnManager.java:336)
    at org.apache.sqoop.orm.ClassWriter.getColumnTypes(ClassWriter.java:1872)
    at org.apache.sqoop.orm.ClassWriter.generate(ClassWriter.java:1671)
    at org.apache.sqoop.tool.CodeGenTool.generateORM(CodeGenTool.java:106)
    at org.apache.sqoop.tool.ImportTool.importTable(ImportTool.java:501)
    at org.apache.sqoop.tool.ImportTool.run(ImportTool.java:628)
    at org.apache.sqoop.Sqoop.run(Sqoop.java:147)
    at org.apache.hadoop.util.ToolRunner.run(ToolRunner.java:76)
    at org.apache.sqoop.Sqoop.runSqoop(Sqoop.java:183)
    at org.apache.sqoop.Sqoop.runTool(Sqoop.java:234)
    at org.apache.sqoop.Sqoop.runTool(Sqoop.java:243)
    at org.apache.sqoop.Sqoop.main(Sqoop.java:252)
```

问题分析

从上述两个报错分析，可能是驱动问题，服务端未正确加载JDBC驱动，或者是网络原因，由于Sqoop迁移数据过程，MapReduce任务是运行在分析core节点，因此需要保证正常加载驱动类的情况下，保证源端数据库和集群节点的网络正常通信。

处理步骤

1. 检查源端数据库端口对集群节点均放通。

   若使用EIP公网通信，那么集群节点均需要绑定公网，尤其是分析节点，必须绑定公网才能保证作业正常运行。

   说明

   sqoop实例一般分布在Master节点，但作业真正运行在分析节点，因此创建数据库连接，是和Master节点通信，作业是通过Yarn任务分发到分析节点执行。

2. 检查Sqoop实例安装目录下是否放置数据库驱动jar包，并保证jar包路径正确，文件权限及属主属组正确；配置文件正确配置；保证这些后需要重启Sqoop实例后才可以正常加载驱动。
   - MRS 3.X集群驱动路径为：/opt/Bigdata/FusionInsight_Current/1_xx_LoaderServer/install/FusionInsight-Sqoop-1.99.3/server/webapps/loader/WEB-INF/ext-lib/
   - MRS3.X之前版本集群驱动路径为：/opt/Bigdata/MRS_XXX/install/FusionInsight-Sqoop-1.99.7/FusionInsight-Sqoop-1.99.7/server/jdbc/

     说明

     3.x之前版本需要修改配置文件。

3. 如果是命令行提交作业，建议指定--driver参数。

   如源端是DWS时命令行加：--driver com.xxx.gauss200.jdbc.Driver ，如下图。

   图 20-3 源端是 dws 时 driver 参数

```
[root@bigdata-hadoop-node-master20V01_lib]# sqoop import -Dorg.apache.sqoop.splitter.allow_text_splitter=true \
>  --connect jdbc:gaussdb://[redacted]:8000/postgres \        --driver com.huawei.gauss200.jdbc.Driver
>  --username [redacted]
>  --password [redacted]
>  --query "SELECT id,created_date,city_name,lat,lon,replace(replace(raw_text,chr(10),''),chr(13),'') as raw_text,service,service_id,replace(replace(sl
ot_json,chr(10),''),chr(13),'') as slot_json,slot_type,time,wstext,app_version,vin,app_platform_entity_id,model_series,is_test from voiceda.ods_talk_im
port_transfer where \$CONDITIONS" \
>  --delete-target-dir \
>  --target-dir obs://bigdata-sc/databases_dev/ghy_ods_dev.db/ods_talk_import_transfer \
>  --fields-terminated-by "\001" \
>  -m 1
```

   其他数据库根据实际情况填写--driver参数。

21 使用 Storm

21.1 Storm WebUI 页面中 events 超链接地址无效

用户问题

Storm组件的Storm UI页面中events超链接地址无效。

问题现象

用户提交拓扑后无法查看拓扑数据处理日志，按钮events地址无效。

原因分析

MRS集群提交拓扑时默认不开启拓扑数据处理日志查看功能。

处理步骤

步骤1 进入服务页面：

- MRS 3.x之前版本，单击集群名称，登录集群详情页面，选择“组件管理”。

  说明

  如果集群详情页面没有“组件管理”页签，请先完成IAM用户同步（在集群详情页的“概览”页签，单击“IAM用户同步”右侧的“同步”进行IAM用户同步）。

- MRS 3.x及后续版本，登录FusionInsight Manager，然后选择“集群 > *待操作的集群名*称 > 服务 ”。

步骤2 登录Storm WebUI：

- MRS 2.x及之前版本：选择“Storm”，在“Storm 概述”的“Storm WebUI”，单击任意一个UI链接，打开Storm的WebUI。

  说明

  第一次访问Storm WebUI，需要在浏览器中添加站点信任以继续打开页面。

- MRS 3.x及后续版本：选择“Storm > 概览”，在“基本信息”的“Storm WebUI”，单击任意一个UI链接，打开Storm的WebUI。

步骤3 单击“Topology Summary”区域的指定拓扑名称，打开拓扑的详细信息。

步骤4 在“Topology actions”区域单击“Kill”删除已经提交的Storm拓扑。

步骤5 重新提交Storm拓扑，并开启查看拓扑数据处理日志功能，在提交Storm拓扑时增加参数“topology.eventlogger.executors”，该参数设置为一个不为0的正整数。例如：

storm jar *拓扑包路径 拓扑Main方法的类名称 拓扑名称* **-c topology.eventlogger.executors=X**

步骤6 在Storm UI界面，单击“Topology Summary”区域的指定拓扑名称，打开拓扑的详细信息。

步骤7 在“Topology actions”区域单击“Debug”，输入采样数据的百分比数值，并单击“OK”开始采样。

步骤8 单击拓扑的“Spouts”或“Bolts”任务名称，在“Component summary”单击“events”即可打开处理数据日志。

说明

如需开启特定“Spouts”或“Bolts”任务的拓扑数据处理日志查看功能，请单击拓扑的“Spouts”或“Bolts”任务名称后，“Topology actions”区域单击“Debug”按钮，输入采样数据的百分比数值。

----结束

21.2 提交 Storm 拓扑失败排查思路

问题背景与现象

使用MRS流式集群，主要安装ZooKeeper、Storm、Kafka。

使用客户端命令，提交Topology失败。

可能原因

- Storm服务异常。
- 客户端用户没有进行安全认证或者认证过期。
- 提交拓扑中包含storm.yaml文件和服务端冲突。

原因分析

用户提交拓扑失败，可能原因客户端侧问题或者Storm侧问题。

1. 查看Storm状态。

   对于MRS 3.x之前的版本集群：登录MRS Manager，在MRS Manager页面，选择“服务管理 > Storm”，查看Storm服务当前状态，发现状态为“良好”，且监控指标内容显示正确。

   对于MRS 3.x及后续版本集群：登录FusionInsight Manager。选择“集群 > 服务 > Storm”，查看Storm服务当前状态，发现状态为“良好”，且监控指标内容显示正确。

2. 查看客户端提交日志，发现打印KeeperExceptionSessionExpireException异常信息，如下所示：

```
org.apache.zookeeper.KeeperException$SessionExpiredException: KeeperErrorCode = Session expired
        at org.apache.zookeeper.KeeperException.create(KeeperException.java:131) ~[zookeeper-3.5.0.jar:3.5.0-V100R002C00B109]
        at org.apache.curator.framework.imps.CuratorFrameworkImpl.checkBackgroundRetry(CuratorFrameworkImpl.java:710) [curator-framework-2.5.0.jar:na]
        at org.apache.curator.framework.imps.CuratorFrameworkImpl.processBackgroundOperation(CuratorFrameworkImpl.java:510) [curator-framework-2.5.0.jar:na]
        at org.apache.curator.framework.imps.BackgroundSyncImpl41.processResult(BackgroundSyncImpl.java:50) [curator-framework-2.5.0.jar:na]
        at org.apache.zookeeper.ClientCnxn$EventThread.processEvent(ClientCnxn.java:484) [zookeeper-3.5.0.jar:3.5.0-V100R002C00B109]
        at org.apache.zookeeper.ClientCnxn$EventThread.queuePacket(ClientCnxn.java:498) [zookeeper-3.5.0.jar:3.5.0-V100R002C00B109]
        at org.apache.zookeeper.ClientCnxn.finishPacket(ClientCnxn.java:731) [zookeeper-3.5.0.jar:3.5.0-V100R002C00B109]
        at org.apache.zookeeper.ClientCnxn.conLossPacket(ClientCnxn.java:765) [zookeeper-3.5.0.jar:3.5.0-V100R002C00B109]
        at org.apache.zookeeper.ClientCnxn.access$2700(ClientCnxn.java:97) [zookeeper-3.5.0.jar:3.5.0-V100R002C00B109]
        at org.apache.zookeeper.ClientCnxn$SendThread.cleanup(ClientCnxn.java:1391) [zookeeper-3.5.0.jar:3.5.0-V100R002C00B109]
        at org.apache.zookeeper.ClientCnxn$SendThread.run(ClientCnxn.java:1314) [zookeeper-3.5.0.jar:3.5.0-V100R002C00B109]
2016-05-31 09:25:24 | INFO  | [main] | Session: 0x100273947600ab4b closed | org.apache.zookeeper.ZooKeeper (ZooKeeper.java:968)
Exception in thread "main" java.lang.RuntimeException: Exception while initializing NimbusLeaderElections
        at backtype.storm.nimbus.NimbusLeaderElections.init(NimbusLeaderElections.java:86)
        at backtype.storm.utils.NimbusClient.getConfiguredClient(NimbusClient.java:39)
        at backtype.storm.StormSubmitter.submitTopology(StormSubmitter.java:159)
        at backtype.storm.StormSubmitter.submitTopologyWithProgressBar(StormSubmitter.java:256)
        at backtype.storm.StormSubmitter.submitTopologyWithProgressBar(StormSubmitter.java:236)
        at storm.starter.WordCountTopology.main(WordCountTopology.java:94)
Caused by: org.apache.zookeeper.KeeperException$ConnectionLossException: KeeperErrorCode = ConnectionLoss for /storm/nimbus-leader
        at org.apache.zookeeper.KeeperException.create(KeeperException.java:99)
        at org.apache.zookeeper.KeeperException.create(KeeperException.java:51)
        at org.apache.zookeeper.ZooKeeper.exists(ZooKeeper.java:1501)
        at org.apache.curator.framework.imps.ExistsBuilderImpl$2.call(ExistsBuilderImpl.java:172)
        at org.apache.curator.framework.imps.ExistsBuilderImpl$2.call(ExistsBuilderImpl.java:161)
        at org.apache.curator.RetryLoop.callWithRetry(RetryLoop.java:107)
        at org.apache.curator.framework.imps.ExistsBuilderImpl.pathInForeground(ExistsBuilderImpl.java:157)
        at org.apache.curator.framework.imps.ExistsBuilderImpl.forPath(ExistsBuilderImpl.java:148)
        at org.apache.curator.framework.imps.ExistsBuilderImpl.forPath(ExistsBuilderImpl.java:36)
        at backtype.storm.nimbus.NimbusLeaderElections.init(NimbusLeaderElections.java:66)
        ... 5 more
```

上述错误是由于在提交拓扑之前没有进行安全认证或者认证后TGT过期导致。

解决方法参考**步骤1**。

3. 查看客户端提交日志，发现打印ExceptionInIntializerError异常信息，提示Found multiple storm.yaml resources。如下所示：

```
Exception in thread "main" java.lang.ExceptionInInitializerError
        at backtype.storm.topology.TopologyBuilder.createTopology(TopologyBuilder.java:106)
        at com.huawei.streaming.storm.example.wordcount.WordCountTopology.cmdSubmit(WordCountTopology.java:117)
        at com.huawei.streaming.storm.example.wordcount.WordCountTopology.submitTopology(WordCountTopology.java:80)
        at com.huawei.streaming.storm.example.wordcount.WordCountTopology.main(WordCountTopology.java:71)
Caused by: java.lang.RuntimeException: Found multiple storm.yaml resources. You're probably bundling the Storm jars with your topology jar.
        at backtype.storm.utils.Utils.findAndReadConfigFile(Utils.java:151)
        at backtype.storm.utils.Utils.readStormConfig(Utils.java:206)
        at backtype.storm.utils.Utils.<clinit>(Utils.java:70)
        ... 4 more
```

该错误是由于业务jar包中存在storm.yaml文件，和服务端的storm.yaml文件冲突导致的。

解决方法参考**步骤2**。

4. 如果不是上述原因，则请参考**提交Storm拓扑失败，提示Failed to check principle for keytab**。

解决办法

步骤1 认证异常。

1. 登录客户端节点，进入客户端目录。
2. 执行以下命令重新提交任务。（业务jar包和Topology根据实际情况替换）

   source bigdata_env

   kinit *用户名*

   storm jar storm-starter-topologies-0.10.0.jar storm.starter.WordCountTopology test

步骤2 拓扑包异常。

排查业务jar，将业务jar中storm.yaml文件删除，重新提交任务。

----结束

21.3 提交 Storm 拓扑失败，提示 Failed to check principle for keytab

问题背景与现象

使用MRS流式安全集群，主要安装ZooKeeper、Storm、Kafka等。

定义拓扑访问HDFS、HBase等组件，使用客户端命令，提交Topology失败。

可能原因

- 提交拓扑中没有包含用户的keytab文件。
- 提交拓扑中包含的keytab和提交用户不一致。
- 客户端/tmp目录下已存在user.keytab，且宿主非运行用户。

原因分析

1. 查看日志发现异常信息Can not found user.keytab in storm.jar。具体信息如下：
```
[main] INFO  b.s.StormSubmitter - Get principle for stream@HADOOP.COM success
[main] ERROR b.s.StormSubmitter - Can not found user.keytab in storm.jar.
Exception in thread "main" java.lang.RuntimeException: Failed to check principle for keytab
at backtype.storm.StormSubmitter.submitTopologyAs(StormSubmitter.java:219)
```
   查看提交的拓扑运行Jar，发现没有包含keytab文件。
2. 查看日志发现异常信息The submit user is invalid,the principle is 。具体信息如下：
```
[main] INFO  b.s.StormSubmitter - Get principle for stream@HADOOP.COM success
[main] WARN  b.s.s.a.k.ClientCallbackHandler - Could not login: the client is being asked for a
password, but the  client code does not currently support obtaining a password from the user. Make
sure that the client is configured to use a ticket cache (using the JAAS configuration setting
'useTicketCache=true)' and restart the client. If you still get this message after that, the TGT in the
ticket cache has expired and must be manually refreshed. To do so, first determine if you are using a
password or a keytab. If the former, run kinit in a Unix shell in the environment of the user who is
running this client using the command 'kinit <princ>' (where <princ> is the name of the client's
Kerberos principal). If the latter, do 'kinit -k -t <keytab> <princ>' (where <princ> is the name of the
Kerberos principal, and <keytab> is the location of the keytab file). After manually refreshing your
cache, restart this client. If you continue to see this message after manually refreshing your cache,
ensure that your KDC host's clock is in sync with this host's clock.
[main] ERROR b.s.StormSubmitter - The submit user is invalid,the principle is : stream@HADOOP.COM
Exception in thread "main" java.lang.RuntimeException: Failed to check principle for keytab
at backtype.storm.StormSubmitter.submitTopologyAs(StormSubmitter.java:219)
```
   业务提交拓扑时使用的认证用户为stream，但是在拓扑提交过程中提示submit user是无效用户，表明内部校验失败。
3. 查看提交的拓扑运行Jar，发现包含keytab文件。

   查看user.keytab文件，发现principal为zmk_kafka。
```
[root@8-5-148-6 client]# klist -kt user.keytab
Keytab name: FILE:user.keytab
KVNO Timestamp         Principal
---- ----------------- --------------------------------------------------------
   1 12/19/16 16:28:17 zmk_kafka@HADOOP.COM
   1 12/19/16 16:28:17 zmk_kafka@HADOOP.COM
```
   发现认证用户和user.keytab文件中principal不对应。
4. 查看日志发现异常信息Delete the tmp keytab file failed，the keytab file is ：/tmp/user.keytab，具体信息如下：

```
[main] WARN b.s.StormSubmitter - Delete the tmp keytab file failed, the keytab file is : /tmp/
user.keytab
[main] ERROR b.s.StormSubmitter - The submit user is invalid,the principle is : hbase1@HADOOP.COM
Exception in thread "main" java.lang.RuntimeException: Failed to check principle for keytab
at backtype.storm.StormSubmitter.submitTopologyAs(StormSubmitter.java:213)
```

查看系统/tmp目录，发现存在user.keytab文件，且文件宿主非运行用户。

解决办法

- 提交拓扑时携带用户user.keytab文件。
- 提交拓扑时的用户需要和user.keytab文件用户一致。
- 删除/tmp目录下不对应的user.keytab文件。

21.4 提交 Storm 拓扑后 Worker 日志为空

现象描述

在Eclipse中远程提交拓扑成功之后，无法在Storm WebUI查看拓扑的详细信息，并且每个拓扑的Bolt和Spout所在Worker节点在一直变化。查看Worker日志，日志内容为空。

可能原因

Worker进程启动失败，触发Nimbus重新分配任务，在其他Supervisor上启动Worker。由于Worker启动失败后会继续重启，导致Worker节点在一直变化，且Worker日志内容为空。Worker进程启动失败的可能原因有两个：

- 提交的Jar包中包含“storm.yaml”文件。

  Storm规定，每个“classpath”中只能包含一个“storm.yaml”文件，如果多于一个那么就会产生异常。使用Storm客户端提交拓扑，由于客户端“classpath”配置和Eclipse远程提交方式“classpath”不一样，客户端会自动加载用户的Jar包到“classpath”，从而使“classpath”中存在两个“storm.yaml”文件。
- Worker进程初始化时间较长，超过Storm集群设置Worker启动超时时间，导致Worker被Kill从而一直进行重分配。

定位思路

1. 使用Storm客户端提交拓扑，检查出重复“storm.yaml”问题。
2. 重新打包Jar包，然后再提交拓扑。
3. 修改Storm集群关于Worker启动超时参数。

处理步骤

步骤1 使用Eclipse远程提交拓扑后Worker日志为空，则使用Storm客户端，提交拓扑对应的Jar包，查看提示信息。

例如，Jar包中包含两个不同路径下的“storm.yaml”文件，系统显示以下信息：

```
Exception in thread "main" java.lang.ExceptionInInitializerError
  at com.XXX.example.WordCountTopology.createConf(WordCountTopology.java:132)
 Caused by: java.lang.RuntimeException: Found multiple storm.yaml resources. You're probably bundling the
Storm jars with your topology jar. [jar:file:/XXX/streaming-0.9.2/bin/stormDemo.jar!/storm.yaml, file:/XXX/
Streaming/streaming-0.9.2/conf/storm.yaml]
  at backtype.storm.utils.Utils.findAndReadConfigFile(Utils.java:151)
```

步骤2 重新打包Jar包，且不能包含“storm.yaml”文件、“log4j”和“slf4j-log4j”相关的Jar包。

步骤3 使用IntelliJ IDEA远程提交新打包的Jar包。

步骤4 查看是否可以在WebUI查看拓扑的详细信息和Worker日志内容。

步骤5 在Manager页面修改Storm集群关于Worker启动超时参数（参数说明请参考**参考信息**），保存并重启Storm服务。

- MRS Manager界面操作入口：登录MRS Manager，依次选择 “服务管理 > Storm> 配置”。
- FusionInsight Manager界面操作入口：登录FusionInsight Manager，选择“集群 > 服务 > Storm > 配置”

步骤6 重新提交待运行的Jar包。

----结束

参考信息

1. nimbus.task.launch.secs和supervisor.worker.start.timeout.secs这两个参数分别代表nimbus端和supervisor端对于拓扑启动的超时容忍时间，一般nimbus.task.launch.secs的值要大于等于supervisor.worker.start.timeout.secs的值（建议相等或略大，如果超出太多会影响任务重分配的效率）。
   - nimbus.task.launch.secs：nimbus在超过该参数配置的时间内没有收到拓扑的task发的心跳时，会将该拓扑重新分配（分配给别的supervisor），同时会刷新zk中的任务信息，supervisor读到zk中的任务信息并且与自己当前所启动的拓扑进行比较，如果存在拓扑已经不属于自己，那么则会删除该拓扑的元数据，也就是/srv/Bigdata/streaming_data/stormdir/supervisor/stormdist/{worker-id}目录。
   - supervisor.worker.start.timeout.secs：supervisor启动worker后，在该参数配置的时间内没有收到worker的心跳时，supervisor会主动停掉worker，等待worker的重新调度，一般在业务启动时间较长时适当增加该参数的值，保证worker能启动成功。

     如果supervisor.worker.start.timeout.secs配置的值比nimbus.task.launch.secs的值大，那么则会出现supervisor的容忍时间没到，仍然继续让worker启动，而nimbus却认定该业务启动超时，将该业务分配给了其他主机，这时supervisor的后台线程发现任务不一致，删除了拓扑的元数据，导致接下来worker在启动过程中要读取stormconf.ser时，发现该文件已经不存在了，就会抛出FileNotFoundException。
2. nimbus.task.timeout.secs和supervisor.worker.timeout.secs这两个参数则分别代表nimbus端和supervisor端对于拓扑运行过程中心跳上报的超时容忍时间，一般nimbus.task.timeout.secs的值要大于等于supervisor.worker.timeout.secs的值（建议相等或略大），原理同上。

21.5 提交 Storm 拓扑后 Worker 运行异常，日志提示 Failed to bind to XXX

现象描述

提交业务拓扑后，发现Worker无法正常启动。查看Worker日志，日志提示Failed to bind to: *Host_ip:Port*。

```
"2017-12-28 04:24:40,153" | INFO  | [main] | Create Netty Server Netty-server-localhost-29101, buffer_size: 5242880, maxWorkers: 1 | backtype.storm.messaging.netty.Server (Server.java:110)
"2017-12-28 04:24:40,170" | ERROR | [main] | Error on initialization of server mk-worker | backtype.storm.daemon.worker (NO_SOURCE_FILE:0)
org.apache.storm.shade.org.jboss.netty.channel.ChannelException: Failed to bind to: dggcbgfi056-stm/10.3.47.75:29101
        at org.apache.storm.shade.org.jboss.netty.bootstrap.ServerBootstrap.bind(ServerBootstrap.java:272) ~[storm-core-0.10.0.jar:0.10.0]
        at backtype.storm.messaging.netty.Server.<init>(Server.java:132) ~[storm-core-0.10.0.jar:0.10.0]
        at backtype.storm.messaging.netty.Context.bind(Context.java:74) ~[storm-core-0.10.0.jar:0.10.0]
        at backtype.storm.daemon.worker$worker_data$fn__3842.invoke(worker.clj:214) ~[storm-core-0.10.0.jar:0.10.0]
        at backtype.storm.util$assoc_apply_self.invoke(util.clj:921) ~[storm-core-0.10.0.jar:0.10.0]
        at backtype.storm.daemon.worker$worker_data.invoke(worker.clj:211) ~[storm-core-0.10.0.jar:0.10.0]
        at backtype.storm.daemon.worker$fn__4006$exec_fn__1339__auto__$reify__4008.run(worker.clj:430) ~[storm-core-0.10.0.jar:0.10.0]
        at java.security.AccessController.doPrivileged(Native Method) ~[?:1.8.0_72]
        at javax.security.auth.Subject.doAs(Subject.java:422) ~[?:1.8.0_72]
        at backtype.storm.daemon.worker$fn__4006$exec_fn__1339__auto____4007.invoke(worker.clj:428) ~[storm-core-0.10.0.jar:0.10.0]
        at clojure.lang.AFn.applyToHelper(AFn.java:186) ~[clojure-1.6.0.jar:?]
        at clojure.lang.AFn.applyTo(AFn.java:144) ~[clojure-1.6.0.jar:?]
        at clojure.core$apply.invoke(core.clj:624) ~[clojure-1.6.0.jar:?]
        at backtype.storm.daemon.worker$fn__4006$mk_worker__4083.doInvoke(worker.clj:409) [storm-core-0.10.0.jar:0.10.0]
        at clojure.lang.RestFn.invoke(RestFn.java:551) [clojure-1.6.0.jar:?]
        at backtype.storm.daemon.worker$_main.invoke(worker.clj:544) [storm-core-0.10.0.jar:0.10.0]
        at clojure.lang.AFn.applyToHelper(AFn.java:171) [clojure-1.6.0.jar:?]
        at clojure.lang.AFn.applyTo(AFn.java:144) [clojure-1.6.0.jar:?]
        at backtype.storm.daemon.worker.main(Unknown Source) [storm-core-0.10.0.jar:0.10.0]
Caused by: java.net.BindException: Address already in use
        at sun.nio.ch.Net.bind0(Native Method) ~[?:1.8.0_72]
        at sun.nio.ch.Net.bind(Net.java:433) ~[?:1.8.0_72]
        at sun.nio.ch.Net.bind(Net.java:425) ~[?:1.8.0_72]
        at sun.nio.ch.ServerSocketChannelImpl.bind(ServerSocketChannelImpl.java:222) ~[?:1.8.0_72]
```

可能原因

随机端口范围配置错误。

定位思路

1. 检查worker相关信息日志。
2. 检查绑定端口的进程信息。
3. 检查随机端口范围配置。

原因分析

1. 通过SSH登录Worker启动失败主机，通过**netstat -anp | grep <port>**命令，查看占用端口的进程ID信息。其中port修改为实际端口号。
2. 通过**ps -ef | grep <pid>**命令查看进程的详细信息，其中pid为查询出的实际进程ID。

```
[root@dgg-dggcbgfi056-stm ~]# netstat -anp|grep 29101
tcp        0      0 10.3.47.75:29101          10.3.47.76:21005            ESTABLISHED 40601/java
[root@dgg-dggcbgfi056-stm ~]#
[root@dgg-dggcbgfi056-stm ~]#
[root@dgg-dggcbgfi056-stm ~]# ps -efl|grep 40601
vocadmin  40601  40524 53 02:38 ?        03:39:50 [redacted]/Bigdata/jdk1.8.0_72//bin/java -server -DignoreReplayReqDetect -Dzookeeper.server.principal=zookeeper/hadoop.hadoop.com -Djava.security.auth.login.config=/[redacted]/Bigdata/FusionInsight_V100R002C60U10/etc/2_34_Kerberos client/kdc.conf -Dzookeeper.request.timeout=120000 -Xms10G -Xmx20G -XX:+UseG1GC -XX:+PrintGCDetails -XX:+PrintGCDateStamps -XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=10 -XX:GCLogFileSize=1M -Xloggc:/var/log/Bigdata/streaming/supervisor/A_VOC_3102_ECC_NOISE_HANDLE-1048-1514399876/resources/A_VOC_3102_ECC_NOISE_HANDLE-1048-1514399876-worker-29122-gc.log -Djava.library.path=/srv/BigData/streaming_data/stormdir/supervisor/stormdist/A_VOC_3102_ECC_NOISE_HANDLE-1048-1514399876/resources:/usr/local/lib:/opt/local/lib:/usr/lib -Dlogfile.name=A_VOC_3102_ECC_NOISE_HANDLE-1048-1514399876-worker-29122.log -Dstorm.home=/opt/[redacted]Bigdata/FusionInsight_V100R002C60U10/FusionInsight-Streaming-0.10.0/streaming -Dstorm.conf.file= -Dstorm.options= -Dstorm.log.dir=/var/log/Bigdata/streaming/supervisor -Dlogging.sensitivity=S1 -Dlog4j.configurationFile=/opt/huawei/Bigdata/FusionInsight_V100R002C60U10/etc/2_37_Supervisor/worker.xml -Dstorm.id=A_VOC_3102_ECC_NOISE_HANDLE-1048-1514399876 -Dworker.id=8dc21b2b-cbad-4d35-979b-c19b9b4a123 -Dworker.port=29122 -Dproc_backtype.storm.daemon.worker -cp [redacted]/FusionInsight_V100R002C60U10/FusionInsight-Streaming-0.10.0/streaming/lib/clojure-1.6.0.jar:[redacted]/Bigdata/FusionInsight_V100R002C60U10/FusionInsight-Streaming-0.10.0/streaming/lib/mlog-1.2.jar:[redacted]/Bigdata/FusionInsight_V100R002C60U10/FusionInsight-Streaming-0.10.0/streaming/lib/clf4j-api-1.7.7.jar:[redacted]Bigdata/FusionInsight_V100R002C60U10/FusionInsight-Streaming
```

发现占用端口的进程为worker进程，该进程为另一个拓扑业务进程。同时根据进程详细信息发现，分配给该进程的端口为29122。

3. 通过**lsof -i:<port>**命令，查看连接详细信息。其中port为实际端口号。

```
root@dgg-dggcbgfi056-stm supervisor]# lsof -i:29101
COMMAND   PID     USER   FD   TYPE   DEVICE SIZE/OFF NODE NAME
java    40601 vocadmin  185u  IPv4 306565038      0t0  TCP dggcbgfi056-stm:29101->dggcbgfi058-kfk:21005 (ESTABLISHED)
root@dgg-dggcbgfi056-stm supervisor]#
root@dgg-dggcbgfi056-stm supervisor]#
```

发现29101端口连接对端端口为21005，而21005为Kafka服务端端口。

说明业务层作为客户端连接Kafka获取消息，业务端口分配通过OS的随机端口分配范围来确定。

4. 通过**cat /proc/sys/net/ipv4/ip_local_port_range**命令查看随机端口范围。

```
[root@dgg-dggcbgfi056-stm supervisor]#
[root@dgg-dggcbgfi056-stm supervisor]#
[root@dgg-dggcbgfi056-stm supervisor]# cat /proc/sys/net/ipv4/ip_local_port_range
10240   65000
[root@dgg-dggcbgfi056-stm supervisor]#
```

5. 发现随机端口范围过大，和MRS的服务端口范围存在冲突。

说明

MRS的服务端口范围：20000~30000。

处理步骤

步骤1 修改随机端口范围。

vi /proc/sys/net/ipv4/ip_local_port_range

```
32768 61000
```

步骤2 停止占用服务端口的业务进程，释放端口。（停止业务拓扑）

----结束

21.6 使用 jstack 命令查看进程堆栈提示“well-known file is not secure”

问题背景与现象

使用jstack命令查看进程堆栈信息时报以下错误：

```
omm@hadoop02:~> jstack 62517
62517: well-known file is not secure
```

原因分析

1. 由于执行命令的用户与当前查看pid信息的进程提交用户不一致导致。
2. Storm引入区分用户执行任务特性，在启动worker进程时将给进程的uid和gid改为提交用户和ficommon，目的是为了logviewer可以访问到worker进程的日志同时日志文件只开放权限到640。这样会导致切换到提交用户后对Worker进程执行jstack和jmap等命令执行失败，原因是提交用户的默认gid并不是ficommon，需要通过ldap命令修改提交用户的gid为9998（ficommon）才可执行。

解决办法

共有两种方式解决该问题。

方式一：通过storm原生页面查看进程堆栈

步骤1 登录Storm原生界面。

MRS Manager界面操作：

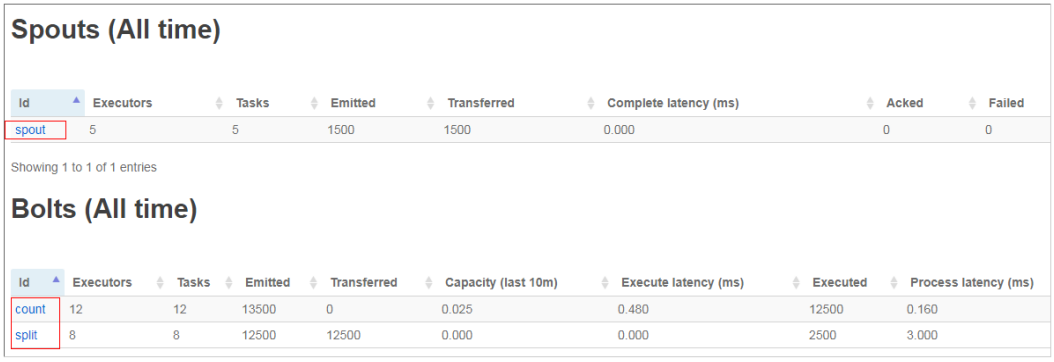
1. 访问MRS Manager。
2. 在Manager选择“服务管理 > Storm”，在“Storm 概述”的“Storm WebUI”，单击任意一个UI链接，打开Storm的WebUI。

FusionInsight Manager界面操作：

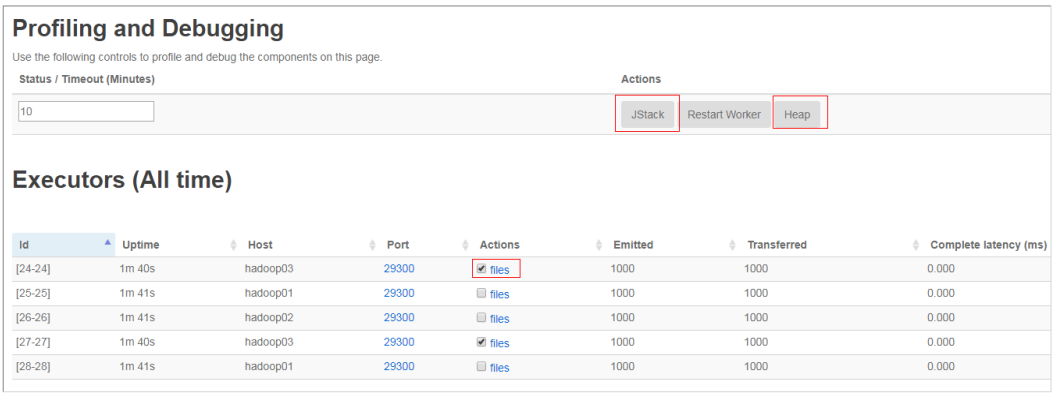
1. 访问FusionInsight Manager。
2. 在Manager选择“集群 > 服务 > Storm”，在“概览”的“Storm WebUI”，单击任意一个UI链接，打开Storm的WebUI。

步骤2 选择要查看的拓扑。

步骤3 选择要查看的spout或者bolt。



步骤4 选择要查看的节点日志文件，再选择JStack或者Heap按钮，其中JStack对应的是堆栈信息，Heap对应的是堆信息：



----结束

方式二：通过修改自定义参数查看进程堆栈

步骤1 进入Storm服务参数配置界面。

MRS Manager界面操作：登录MRS Manager页面，选择“服务管理 > Storm > 服务配置”，“参数类别”选择“全部配置”。

FusionInsight Manager界面操作：登录FusionInsight Manager，选择“集群 > 服务 > Yarn”，单击“配置”，选择“全部配置”。

步骤2 在左侧导航栏选择“supervisor > 自定义”，添加一个变量supervisor.run.worker.as.user=false。

步骤3 保存配置，勾选“重新启动受影响的服务或实例。”并单击“确定”重启服务。

步骤4 重新提交拓扑。

步骤5 后台节点切为omm用户执行jps命令即可查看worker的pid。

```
omm@hadoop02:~> jps | grep worker
22485 worker
111402 worker
```

步骤6 执行jstack pid，即可查看jstack信息。

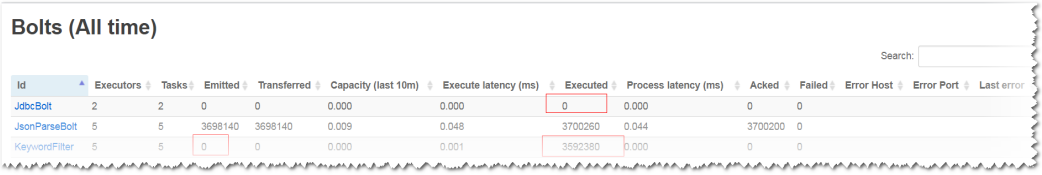
```
omm@hadoop02:~> jstack 22485
2018-05-26 08:46:24
Full thread dump Java HotSpot(TM) 64-Bit Server VM (25.144-b01 mixed mode):

"Attach Listener" #82 daemon prio=9 os_prio=0 tid=0x0000000001c95000 nid=0xb840 waiting on condition [0x0000000000000000]
   java.lang.Thread.State: RUNNABLE

"pool-14-thread-1" #81 daemon prio=5 os_prio=0 tid=0x00007f7ebc931000 nid=0x6113 waiting on condition [0x00007f7eb5ddf000]
   java.lang.Thread.State: TIMED_WAITING (parking)
        at sun.misc.Unsafe.park(Native Method)
        - parking to wait for  <0x00000000dfe820a0> (a java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject)
        at java.util.concurrent.locks.LockSupport.parkNanos(LockSupport.java:215)
        at java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.awaitNanos(AbstractQueuedSynchronizer.java:2078)
        at java.util.concurrent.ScheduledThreadPoolExecutor$DelayedWorkQueue.take(ScheduledThreadPoolExecutor.java:1093)
        at java.util.concurrent.ScheduledThreadPoolExecutor$DelayedWorkQueue.take(ScheduledThreadPoolExecutor.java:809)
        at java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1074)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1134)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        at java.lang.Thread.run(Thread.java:748)
```

----结束

21.7 使用 Storm-JDBC 插件开发 Oracle 写入 Bolt 时发现数据无法写入

现象描述

使用Storm-JDBC插件开发Oracle写入Bolt，发现能连上Oracle数据库，但是无法向Oracle数据库里面写数据。



Bolts (All time)

Search:

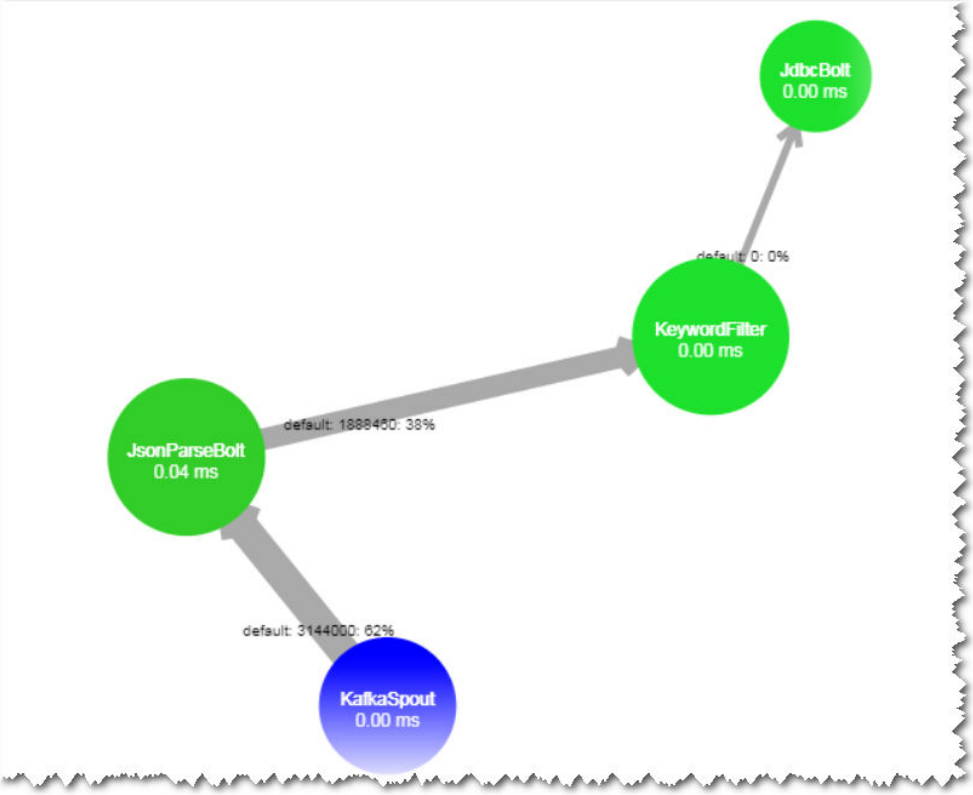
Id	Executors	Tasks	Emitted	Transferred	Capacity (last 10m)	Execute latency (ms)	Executed	Process latency (ms)	Acked	Failed	Error Host	Error Port	Last error
JdbcBolt	2	2	0	0	0.000	0.000	0	0.000	0	0			
JsonParseBolt	5	5	3698140	3698140	0.009	0.048	3700260	0.044	3700200	0			
KeywordFilter	5	5	0	0	0.000	0.001	3592380	0.000	0	0			

可能原因

- 拓扑定义异常。
- 数据库表结果定义异常。

原因分析

1. 通过Storm WebUI查看拓扑DAG图，发现DAG图与拓扑定义一致。

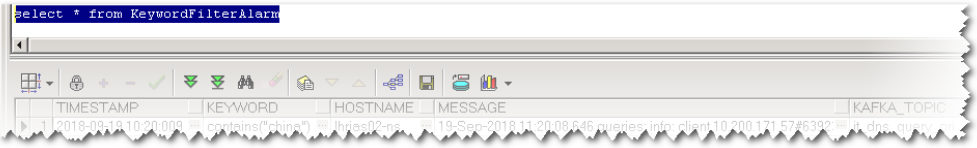


2. 查看KeyWordFilter Bolt输出流字段定义和发送消息字段发现一致。

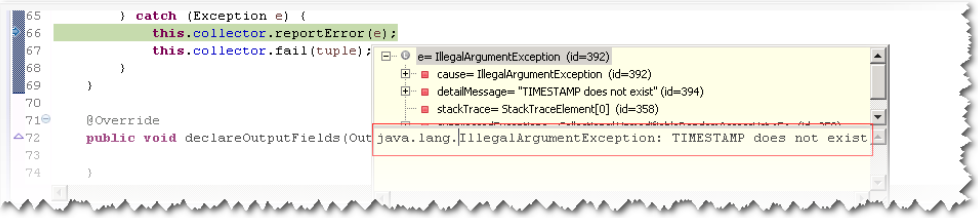
```
@Override
public void declareOutputFields(OutputFieldsDeclarer declarer)
{
    declarer.declare(new Fields("timestamp","keyword","hostname","message","kafka_topic") );
```

```
    if( flag )
    {
        String keyword = expParser.getKeyword();
        System.out.println( message );
        collector.emit(new Values( timestamp, keyword , hostname , message, kafka_topic ));
```

3. 查看Oracle数据库中表定义，发现字段名为大写，与流定义字段名称不一致。



4. 单独调试execute方法，发现抛出字段不存在。



处理步骤

修改流定义字段名称为大写，与数据库表定义字段一致。

21.8 Storm UI 查看信息时显示 Internal Server Error

问题背景与现象

使用MRS版本安装集群，主要安装ZooKeeper、Strom。

通过MRS Manager中的Storm Status页面UI连接访问信息时显示如下信息：

```
Internal Server Error
org.apache.thrift7.transport.TTransportException: Frame size (306030) larger than max length (1048576)!
```

可能原因

- Storm服务中Nimbus异常。
- Storm集群信息较多超过系统默认Thrift传输大小的设置。

原因分析

1. 查看Storm服务状态及监控指标：
   - MRS Manager界面操作：登录MRS Manager，依次选择“服务管理 > Storm”，查看当前Storm状态，发现状态为良好，且监控指标内容显示正确。
   - FusionInsight Manager界面操作：登录FusionInsight Manager，选择“集群 > 服务 > Storm”，查看当前Storm状态，发现状态为良好，且监控指标内容显示正确。
2. 选择“实例”页签，查看Nimbus实例状态，显示正常。
3. 查看当前Storm集群thrift相关配置，发现nimbus.thrift.max_buffer_size参数配置为1048576（1M）。
4. 上述配置和异常信息中信息一致，说明当前配置的Thrift的buffer size小于集群信息所需的buffer size。

解决方法

调整Storm集群中Thrift的Buffer Size大小，具体大小根据错误信息进行实际调整。

步骤1 进入Storm服务参数配置界面。

- MRS Manager界面操作：登录MRS Manager页面，选择“服务管理 > Storm > 服务配置”，“参数类别”选择“全部配置”。
- FusionInsight Manager界面操作：登录FusionInsight Manager，选择“集群 > 服务 > Yarn”，单击“配置”，选择“全部配置”。

步骤2 修改nimbus.thrift.max_buffer_size参数为10485760（10M）。

步骤3 保存配置，勾选“重新启动受影响的服务或实例”并单击“确定”重启服务。

----结束

22 使用 Ranger

22.1 Hive 启用 Ranger 鉴权后，在 Hue 页面能查看到没有权限的表和库

问题现象

普通集群（未开启Kerberos认证）中，Hive启用Ranger鉴权后，集群用户在Hue页面能查看到没有被授权的数据表和数据库内容。

原因分析

Hive启用Ranger鉴权后，默认的Hive策略中有2个关于database的public组策略， 所有用户都属于public组，默认给public组配有default数据库的创表和所有其他数据库的**create**权限，因此默认所有的用户都有**show databases**和**show tables**的权限，如果不想让某些用户有**show databases**和**show tables**权限，可在Ranger WEBUI中删除该默认public组策略，并赋予需要查看的用户权限。

处理步骤

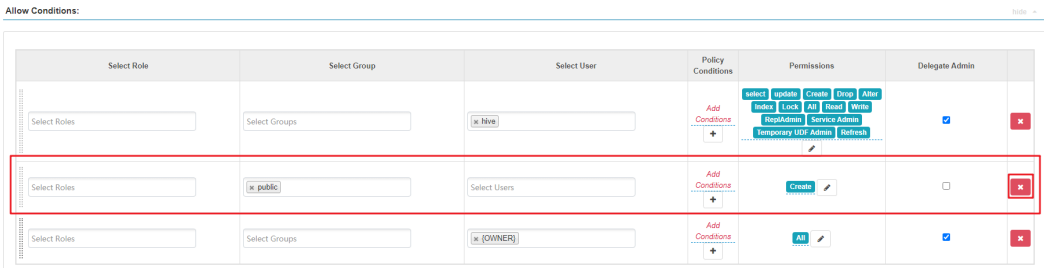
步骤1 登录Ranger WebUI界面。

步骤2 在“Service Manager”区域内，单击Hive组件名称，进入Hive组件安全访问策略列表页面。

步骤3 分别单击“all - database”和“default database tables columns”策略所在行的按钮。

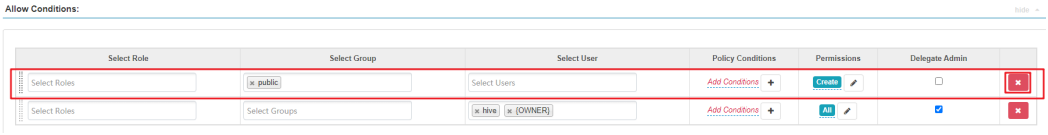
步骤4 删除“public”组策略。

图 22-1 all - database 策略



版权所有 © 华为云计算技术有限公司

图 22-2 default database tables columns 策略



步骤5 在Hive组件安全访问策略列表页面，单击“Add New Policy”为相关用户或者用户组添加资源访问策略，具体请参考**配置组件权限策略**。

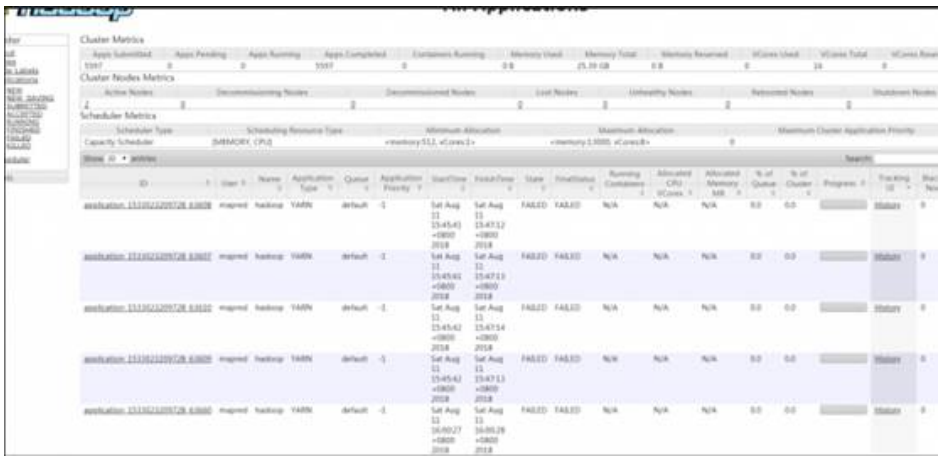
----结束

23 使用 Yarn

23.1 集群启动 Yarn 后产生大量作业占用资源

问题现象

MRS 2.x及之前版本集群，用户的MRS集群启动Yarn后产生大量作业，占用集群计算资源。



原因分析

集群安全组入口方向的Any协议源地址配置为0.0.0.0/0，导致集群可能遭受了外部网络攻击。

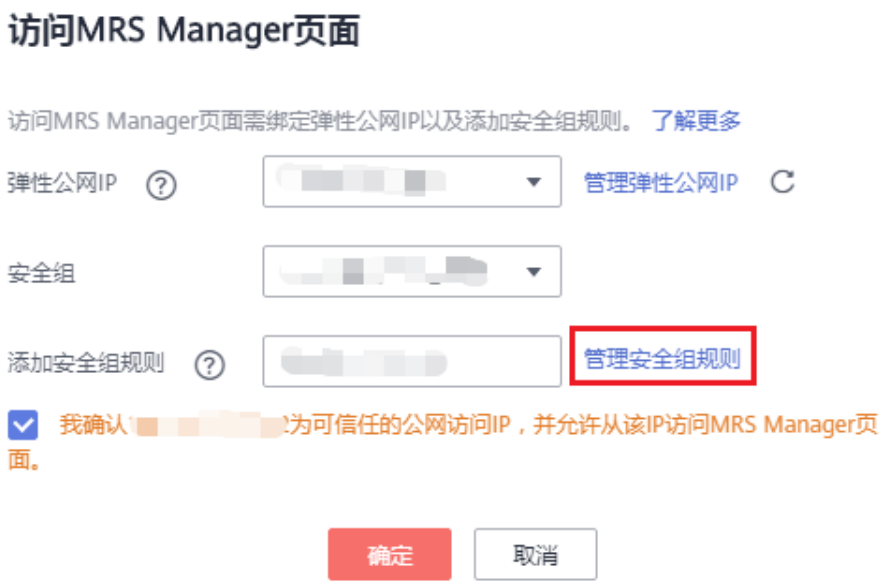
IPv4	Any	Any	0.0.0.0/0
IPv4	Any	Any	0.0.0.0/0
IPv4	Any	Any	0.0.0.0/0

处理步骤

步骤1 登录MRS集群页面，在“现有集群”中，单击对应的集群名称，进入集群详情页面。

步骤2 单击“集群管理页面”后面的“前往 Manager”，弹出“访问MRS Manager页面”。

步骤3 单击“管理安全组规则”，检查安全组规则配置。



步骤4 检查入口方向Any协议的源地址是否为0.0.0.0/0。

如果是，修改入口方向Any协议的远端为指定IP地址。如果不是，则无需修改。

步骤5 修改成功后，重启集群。

----结束

建议与总结

关闭入口方向的Any协议，或者指定入口方向的Any协议远端为指定IP。

23.2 通过客户端 hadoop jar 命令提交任务后返回“GC overhead”报错

问题背景与现象

通过客户端提交任务，客户端返回内存溢出的报错结果：

```
main path:hdfs://hacluster/user/
17/09/18 08:29:57 INFO hdfs.DFSClient: Created HDFS_DELEGATION_TOKEN token 22890097 for wangyou on ha-hdfs:hacluster
17/09/18 08:29:57 INFO security.TokenCache: Got dt for hdfs://hacluster; Kind: HDFS_DELEGATION_TOKEN, Service: ha-hdfs:hacluster, Ident: (HDFS_DELEGATION_TOKEN token 2
2890097 for
17/09/18 08:29:57 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with
ToolRunner to remedy this.
17/09/18 08:32:42 INFO retry.RetryInvocationHandler: Exception while invoking getListing of class ClientNamenodeProtocolTranslatorPB over fi1-cn-003/10.113.246.10:2500
0. Trying to fail over immediately.
java.io.IOException: com.google.protobuf.ServiceException: java.lang.OutOfMemoryError: GC overhead limit exceeded
        at org.apache.hadoop.ipc.ProtobufHelper.getRemoteException(ProtobufHelper.java:47)
        at org.apache.hadoop.hdfs.protocolPB.ClientNamenodeProtocolTranslatorPB.getListing(ClientNamenodeProtocolTranslatorPB.java:578)
        at sun.reflect.GeneratedMethodAccessor2.invoke(Unknown Source)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:497)
        at org.apache.hadoop.io.retry.RetryInvocationHandler.invokeMethod(RetryInvocationHandler.java:191)
        at org.apache.hadoop.io.retry.RetryInvocationHandler.invoke(RetryInvocationHandler.java:102)
        at com.sun.proxy.$Proxy10.getListing(Unknown Source)
        at org.apache.hadoop.hdfs.DFSClient.listPaths(DFSClient.java:1757)
        at org.apache.hadoop.hdfs.DistributedFileSystem$DirListingIterator.hasNextNoFilter(DistributedFileSystem.java:1024)
        at org.apache.hadoop.hdfs.DistributedFileSystem$DirListingIterator.hasNext(DistributedFileSystem.java:999)
        at org.apache.hadoop.mapreduce.lib.input.FileInputFormat.singleThreadedListStatus(FileInputFormat.java:304)
        at org.apache.hadoop.mapreduce.lib.input.FileInputFormat.listStatus(FileInputFormat.java:265)
        at org.apache.hadoop.mapreduce.lib.input.CombineFileInputFormat.getSplits(CombineFileInputFormat.java:217)
        at org.apache.hadoop.mapreduce.lib.input.DelegatingInputFormat.getSplits(DelegatingInputFormat.java:115)
        at org.apache.hadoop.mapreduce.JobSubmitter.writeNewSplits(JobSubmitter.java:306)
        at org.apache.hadoop.mapreduce.JobSubmitter.writeSplits(JobSubmitter.java:323)
        at org.apache.hadoop.mapreduce.JobSubmitter.submitJobInternal(JobSubmitter.java:200)
        at org.apache.hadoop.mapreduce.Job$10.run(Job.java:1290)
        at org.apache.hadoop.mapreduce.Job$10.run(Job.java:1287)
        at java.security.AccessController.doPrivileged(Native Method)
        at javax.security.auth.Subject.doAs(Subject.java:422)
        at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1673)
        at org.apache.hadoop.mapreduce.Job.submit(Job.java:1287)
```

原因分析

从报错堆栈可以看出是任务在提交过程中分片时在读取HDFS文件阶段内存溢出了，一般是由于该任务要读取的小文件很多导致内存不足。

解决办法

步骤1 排查启动的MapReduce任务是否对应的HDFS文件个数很多，如果很多，减少文件数量，提前先合并小文件或者尝试使用combineInputFormat来减少任务读取的文件数量。

步骤2 增大hadoop命令执行时的内存，该内存在客户端中设置，修改“*客户端安装目录/HDFS/component_env*”文件中“CLIENT_GC_OPTS”的“-Xmx”参数，将该参数的默认值改大，比如改为512m。然后执行**source component_env**命令，使修改的参数生效。

```
export YARN_ROOT_LOGGER=INFO,console

#GC_OPTS for client operation.
CLIENT_GC_OPTS="-Xmx512m -Djava.io.tmpdir=${HADOOP_HOME}"

export HADOOP_CLIENT_OPTS="$CLIENT_GC_OPTS"
```

----结束

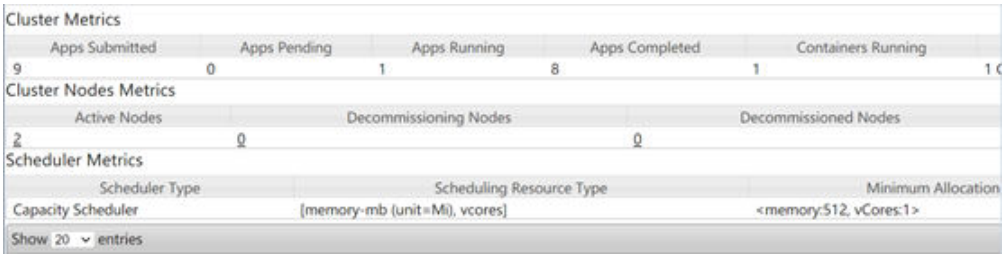
23.3 Yarn 汇聚日志过大导致节点磁盘被占满

用户问题

集群的磁盘使用率很高。

问题现象

- Manager管理页面下主机管理显示磁盘使用率过高。
- Yarn WebUI界面上显示只有少量任务在运行。



- 登录到集群的Master节点执行**hdfs dfs -du -h /** 命令发现如下文件占用大量磁盘空间。

```
22.5 G   45.0 G   /tmp/logs/root/logs/application_1589278244866_0153
18.4 M   36.8 M   /tmp/logs/root/logs/application_1589278244866_0154
23.4 G   46.8 G   /tmp/logs/root/logs/application_1589278244866_0155
23.5 G   46.9 G   /tmp/logs/root/logs/application_1589278244866_0156
23.7 G   47.4 G   /tmp/logs/root/logs/application_1589278244866_0157
23.7 G   47.4 G   /tmp/logs/root/logs/application_1589278244866_0158
22.5 G   45.0 G   /tmp/logs/root/logs/application_1589278244866_0159
18.5 M   37.0 M   /tmp/logs/root/logs/application_1589278244866_0160
22.5 G   45.0 G   /tmp/logs/root/logs/application_1589278244866_0161
18.8 M   37.6 M   /tmp/logs/root/logs/application_1589278244866_0162
24.0 G   48.0 G   /tmp/logs/root/logs/application_1589278244866_0163
121.3 K  242.7 K  /tmp/logs/root/logs/application_1589278244866_0164
1.1 M    2.1 M    /tmp/logs/root/logs/application_1589278244866_0165
1.1 M    2.1 M    /tmp/logs/root/logs/application_1589278244866_0166
1.1 M    2.1 M    /tmp/logs/root/logs/application_1589278244866_0167
1.1 M    2.1 M    /tmp/logs/root/logs/application_1589278244866_0168
```

- Mapreduce服务的汇聚日志配置参数如下：

* yarn.log-aggregation.retain-check-interval-seconds	86400
* yarn.log-aggregation.retain-seconds	1296000

原因分析

客户提交任务的操作过于频繁，且聚合后的日志文件被删除的时间配置为1296000，即聚合日志保留15天，导致汇聚的日志无法在短时间内释放，从而引起磁盘被占满。

处理步骤

步骤1 登录Manager页面，进入Mapreduce服务配置参数页面。

- MRS Manager界面操作：登录MRS Manager，选择“服务管理 > Mapreduce > 服务配置 > 全部配置”。
- FusionInsight Manager界面操作：登录FusionInsight Manager，选择“集群 > 服务 > Mapreduce > 配置 > 全部配置”。

步骤2 搜索“yarn.log-aggregation.retain-seconds”参数，并根据实际情况将yarn.log-aggregation.retain-seconds调小，例如调整为： 259200，即聚合日志保留3天，到期后自动释放磁盘空间。

步骤3 保存配置，如果有弹出窗口，不勾选“重新启动受影响的服务或实例”。

步骤4 在业务空闲时重启配置过期的服务，重启服务会导致上层服务业务中断，影响集群的管理维护和业务，建议在空闲时执行。

1. 登录Manager页面。
2. 重启Mapreduce、Yarn服务。

----结束

23.4 MapReduce 任务异常，临时文件未删除

用户问题

MapReduce任务异常临时文件为什么没有删除?

说明

MR任务即MapReduce任务，关于MapReduce介绍请参考**MapReduce**。

问题现象

HDFS临时目录文件过多，占用内存。

原因分析

MapReduce任务提交时会将相关配置文件、jar包和-files参数后添加的文件都上传至HDFS的临时目录，方便Container启动后获取相应的文件。系统通过配置项“yarn.app.mapreduce.am.staging-dir”决定具体存放位置，默认值是“/tmp/hadoop-yarn/staging”。

正常运行的MapReduce任务会在Job结束以后就清理这些临时文件，但是当Job对应的Yarn任务异常退出时，这些临时文件不会被清理，长时间积攒导致该临时目录下的文件数量越来越多，占用存储空间越来越多。

处理步骤

步骤1 登录集群客户端。

1. 以**root**用户登录任意一个Master节点，用户密码为创建集群时用户自定义的密码。
2. 如果集群开启Kerberos认证，执行如下命令进入客户端安装目录并设置环境变量，再认证用户并按照提示输入密码，该密码请向管理员获取。

   cd *客户端安装目录*

   source bigdata_env

   kinit hdfs
3. 如果集群未开启Kerberos认证，执行如下命令切换到**omm**用户，再进入客户端安装目录设置环境变量。

   su - omm

   cd *客户端安装目录*

   source bigdata_env

步骤2 获取文件列表。

hdfs dfs -ls /tmp/hadoop-yarn/staging/*/.staging/ | grep "^drwx" | awk '{print $8}' > job_file_list

job_file_list文件中就是所有任务的文件夹列表，文件内容参考：

```
/tmp/hadoop-yarn/staging/omm/.staging/job__<Timestamp>_<ID>
```

步骤3 统计当前运行中的任务。

mapred job -list 2>/dev/null | grep job_ | awk '{print $1}' > run_job_list

run_job_list文件里面就是当前正在运行的JobId列表，文件内容格式为：

```
job_<Timestamp>_<ID>
```

步骤4 删除job_file_list文件中正在运行中的任务，确保在删除过期数据时不会误删正在运行任务的数据。

cat run_job_list | while read line; do sed -i "/$line/d" job_file_list; done

步骤5 删除过期数据。

cat job_file_list | while read line; do hdfs dfs -rm -r $line; done

步骤6 清除临时文件。

rm -rf run_job_list job_file_list

----结束

23.5 Yarn 客户端的端口信息错误导致提交任务后报错connection refused

用户问题

请求提交任务的Yarn的ResourceManager报错connection refused，且配置的Yarn端口为8032。

问题现象

MRS的Yarn ResourceManager中的一个节点无法连接，且配置的Yarn端口为8032。

原因分析

该业务应用在集群外部运行，且使用的是老集群的客户端，配置的Yarn端口是8032，与MRS服务的Yarn ResourceManager实际端口不同。从而使请求提交任务的Yarn的ResourceManager报错connection refused。

处理步骤

步骤1 更新MRS服务客户端。

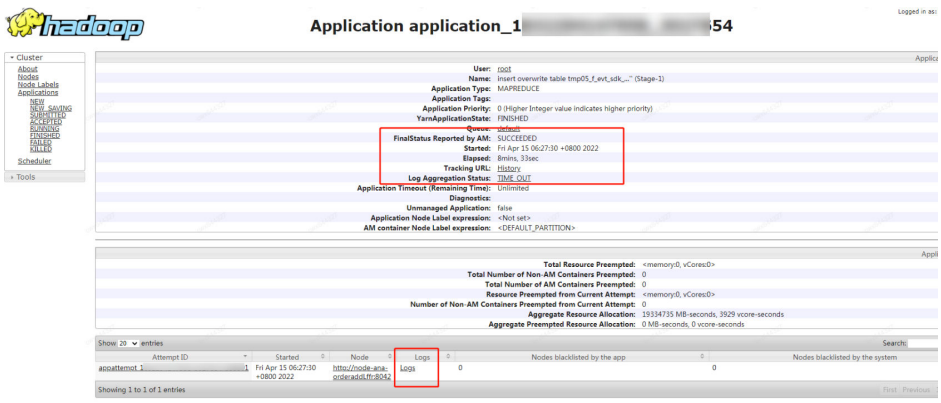
步骤2 重试提交作业。

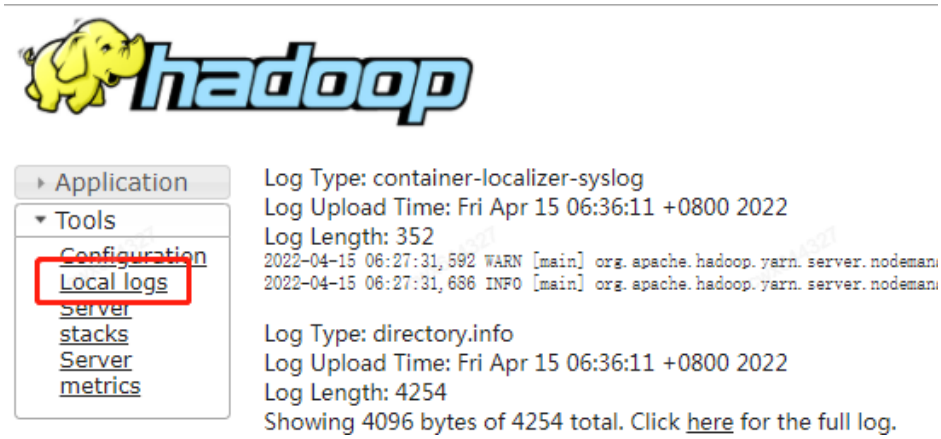
----结束

23.6 通过 Yarn WebUI 查看作业日志时提示“Could not access logs page!”

问题背景与现象

登录Yarn WebUI界面查看作业日志“Logs”，然后单击“Local logs”，界面提示“Could not access logs page!”。





原因分析

该Local logs是用来访问服务的日志，但由于安全考虑，该功能暂不支持从Yarn WebUI界面访问。用户可以登录ResourceManager主节点查看ResourceManager的日志。

处理步骤

步骤1 登录Manager界面，选择“集群 > 组件 > Yarn > 实例”，查看并获取“ResourceManager(主)”实例的业务IP地址。

步骤2 以**root**用户登录ResourceManager主节点。

步骤3 进入“/var/log/Bigdata/yarn/rm”路径查看ResourceManager的日志。

版权所有 © 华为云计算技术有限公司

cd /var/log/Bigdata/yarn/rm

----结束

23.7 通过 Yarn WebUI 查看队列信息时提示“ERROR 500”错误

问题背景与现象

在Yarn使用Capacity调度器时，单击Yarn WebUI页面的队列名称时，系统上报“ERROR 500”的错误。

```
HTTP ERROR 500 javax.servlet.ServletException: javax.servlet.ServletException: java.lang.IllegalArgu
mentException:
Illegal character in queryat index 81: https://XXX:20026/Yarn/ResourceManager/21/cluster/scheduler?
openQueues= ^default$
```

原因分析

页面链接无法识别符号“^”，导致页面访问失败。

处理步骤

步骤1 登录Manager页面，选择“集群 > 服务 > Yarn > 配置 > 全部配置”。

步骤2 在搜索框搜索“yarn.resourcemanager.webapp.pagination.enable”。

步骤3 如果该参数值为“true”（默认为“true”），请修改为“false”，并保存配置。

步骤4 在Yarn服务页面选择“实例”页签，勾选所有的ResourceManager实例，选择“更多 > 滚动重启实例”，等待实例启动完成。

----结束

23.8 通过 Yarn WebUI 页面查看作业日志提示“ERROR 500”错误

问题背景与现象

在MRS上提交的SparkStreaming流式作业执行一段时间后，查看Yarn WebUI页面全量日志报“HTTP ERROR 500 org.apache.http.ConnectionCloseException: Premature end of chunk coded message body:closing chunk expected”。

原因分析

该报错是由于作业运行时间过长，查看Yarn WebUI页面全量日志时需要展示的日志过大，导致报500内部错误。此时需要把作业生成的聚合日志调小，使其分段生成即可。

处理步骤

步骤1 以**root**用户登录安装了Spark2x/Spark客户端的节点。

步骤2 执行以下命令编辑文件。

vim $SPARK_HOME/conf/log4j-executor.properties

步骤3 调小“log4j.appender.sparklog.MaxFileSize”参数值（表示生成一份日志文件的最大存储大小，默认为50MB），例如修改为“20MB”。

步骤4 调小“log4j.appender.sparklog.MaxBackupIndex”参数值（表示滚动保存的文件数量，最大为10个文件，达到第11个覆盖第一个生成的文件），例如修改为“5”。

步骤5 保存文件。

步骤6 重新提交作业，作业运行正常。

----结束

23.9 使用 Yarn 客户端命令查询历史作业报错

问题现象

执行Yarn客户端命令查询历史作业报错，进程被终止。具体错误如下:

```
[root@node-master1mmup HDFS]#  yarn app -list -appStates All
WARNING: YARN_CONF_DIR has been replaced by HADOOP_CONF_DIR. Using value of YARN_CONF_DIR.
WARNING: YARN_ROOT_LOGGER has been replaced by HADOOP_ROOT_LOGGER. Using value of YARN_ROOT_LOGGER.
#
# java.lang.OutOfMemoryError: Java heap space
# -XX:OnOutOfMemoryError=""kill -9 %p""
#   Executing /bin/sh -c ""kill -9 8642""...
sh: kill -9 8642: command not found
Error: A JNI error has occurred, please check your installation and try again
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
        at java.util.Arrays.copyOf(Arrays.java:3236)
        at sun.misc.Resource.getBytes(Resource.java:117)
        at java.net.URLClassLoader.defineClass(URLClassLoader.java:463)
        at java.net.URLClassLoader.access$100(URLClassLoader.java:74)
        at java.net.URLClassLoader$1.run(URLClassLoader.java:369)
        at java.net.URLClassLoader$1.run(URLClassLoader.java:363)
        at java.security.AccessController.doPrivileged(Native Method)
        at java.net.URLClassLoader.findClass(URLClassLoader.java:362)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:418)
        at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:352)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:351)
        at sun.launcher.LauncherHelper.checkAndLoadMain(LauncherHelper.java:601)
[root@node-master1mmup HDFS]# hdfs dfs -ls /
#
# java.lang.OutOfMemoryError: GC overhead limit exceeded
# -XX:OnOutOfMemoryError=""kill -9 %p""
#   Executing /bin/sh -c ""kill -9 26409""...
sh: kill -9 26409: command not found
Exception in thread "main" java.lang.OutOfMemoryError: GC overhead limit exceeded
        at java.lang.String.substring(String.java:1969)
        at java.net.URLStreamHandler.parseURL(URLStreamHandler.java:254)
        at sun.net.www.protocol.file.Handler.parseURL(Handler.java:67)
        at java.net.URL.<init>(URL.java:640)
        at java.net.URL.<init>(URL.java:508)
        at sun.misc.URLClassPath$JarLoader.parseClassPath(URLClassPath.java:1232)
```

原因分析

客户端分配内存不足，导致提交命令报错。

处理步骤

步骤1 以**root**用户登录HDFS客户端安装节点。

步骤2 执行以下命令编辑文件。

vim /opt/client/HDFS/component_env

步骤3 调大参数“ export GC_OPTS_HDFS”的值，例如修改为“Xmx512m”。

步骤4 保存文件。

步骤5 执行以下命令重新加载环境变量。

source bigdata_env

----结束

23.10 TimelineServer 目录文件数量到达上限

问题现象

MRS 3.x版本集群，ResourceManager日志显示TimelineServer数据目录数量到达上限打印大量错误日志。

异常日志内容如下：

```
The directory item limit of /tmp/hadoop-omm/yarn/timeline/generic-history/ApplicationHistoryDataRoot is exceeded: limit=1048576 items=1048576
```

原因分析

TimelineServer在MRS 3.x版本会使用一个HDFS的目录（例如以上报错中的“/tmp/hadoop-omm/yarn/timeline/generic-history/ApplicationHistoryDataRoot”路径）来存放历史任务信息，导致该目录下的文件不断累积，直到到达HDFS配置的目录数量上限（“dfs.namenode.fs-limits.max-directory-items”默认为“1048576”）。

此时请将“yarn.timeline-service.generic-application-history.enabled”（客户端查询app任务数据时是否从TimelineServer服务获取）参数设置为“false”，直接从ResourceManager上面获取app任务数据。

处理步骤

步骤1 登录FusionInsight Manager，选择“集群 > 服务 > Yarn > 配置 > 全部配置”。

步骤2 在左侧导航栏选择“Yarn（服务） > 自定义”，在自定义页面的“yarn.yarn-site.customized.configs”参数后添加“yarn.timeline-service.generic-application-history.enabled”，值为“false”，单击“保存”。

步骤3 滚动重启ResourceManager和TimelineServer实例。

单击Yarn服务的“实例”页签，勾选所有ResourceManager和TimelineServer实例，选择“更多 > 滚动重启实例”。

步骤4 （可选）根据业务需要，在业务低峰时滚动重启NodeManager。

步骤5 实例重启完成后，删除HDFS上的存放历史任务信息的目录，例如“/tmp/hadoop-omm/yarn/timeline/generic-history/ApplicationHistoryDataRoot”。

1. 以客户端安装用户登录客户端安装目录，并配置环境变量。

   cd *客户端安装目录*

   source bigdata_env

2. 执行以下命令认证用户，未开启Kerberos认证的用户跳过该步骤。

   kinit *业务用户*

3. 执行以下命令删除HDFS上的相关目录。

   hdfs dfs -rm -r */tmp/hadoop-omm/yarn/timeline/generic-history/ApplicationHistoryDataRoot/*

----结束

24 使用 ZooKeeper

24.1 通过 MRS 客户端连接 ZooKeeper 报错

用户问题

MRS集群如何访问ZooKeeper?

问题现象

在MRS的Master节点使用zkcli.sh访问ZooKeeper存在报错。

原因分析

用户使用的命令有误，造成报错的发生。

处理步骤

步骤1 获取ZooKeeper的IP地址及端口信息，具体请参考**如何获取ZooKeeper地址？**。

步骤2 以**root**用户登录Master节点。

步骤3 初始化环境变量。

source /opt/client/bigdata_env

步骤4 执行以下命令连接ZooKeeper。

zkCli.sh -server *ZooKeeper所在节点的IP.端口*

ZooKeeper所在节点的IP即为**步骤1**中查到的结果，多个IP之间以逗号间隔。

步骤5 使用**ls /**等常用的命令查看ZooKeeper上的信息。

----结束

24.2 由于主备 Master 节点时间未同步导致 ZooKeeper 服务不可用

问题背景与现象

MRS集群ZooKeeper服务无法启动，出现ZooKeeper服务不可用告警。

原因分析

以**root**用户登录主、备Master节点，执行**ntpq -p**命令查看两节点时间未同步。

```
[root@node-master1uOko ~]# ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 100.125.0.251   10.230.52.36     3 u    3   16  377    0.312   43.253   0.758
*LOCAL(0)        .LOCL.           0 l    -   16  377    0.000    0.000   0.000
[root@node-master1uOko ~]# date
Tue Apr 27 09:12:36 CST 2021
```

解决办法

步骤1 以**root**用户登录主OMS节点执行**ifconfig**命令，查找当前节点的浮动IP。

例如执行后结果如下，则当前节点的浮动IP为192.168.0.3。

```
[root@node-master1uOko ~]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.176  netmask 255.255.255.0  broadcast 192.168.0.255
        ether fa:16:3e:11:d1:cf  txqueuelen 1000  (Ethernet)
        RX packets 381241289  bytes 83002817678 (77.3 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 383655523  bytes 132910473015 (123.7 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0:wsom: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.3  netmask 255.255.255.0  broadcast 192.168.0.255
        ether fa:16:3e:11:d1:cf  txqueuelen 1000  (Ethernet)

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 0  (Local Loopback)
        RX packets 1722974128  bytes 660620647435 (615.2 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1722974128  bytes 660620647435 (615.2 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

步骤2 登录NTP时间异常的Master节点，执行以下命令关闭NTP服务。

service ntpd stop

步骤3 执行以下命令手动同步主、备Master节点的时间。

/usr/sbin/ntpdate *步骤1查询到的浮动IP*

步骤4 执行以下命令启动NTP服务。

service ntpd start

步骤5 执行**ntpstat**命令查看时间同步结果。

```
[root@node-master1uOko ~]# ntpstat
synchronised to local net at stratum 1
   time correct to within 7948 ms
   polling server every 16 s
[root@node-master1uOko ~]# date
Tue Apr 27 09:14:53 CST 2021
```

----结束

25 存算分离类

25.1 没有/tmp 目录权限导致执行访问 OBS 的作业报错

问题现象

在使用MRS多用户访问OBS的场景中，执行Spark、Hive、Presto等作业时，出现“/tmp”目录没有权限的报错。

问题原因

作业执行过程中有临时目录，提交作业的用户对临时目录没有操作权限。

解决方法

步骤1 在集群“概览”页签中，查询并记录集群所绑定的委托名称。

步骤2 登录IAM服务控制台。

步骤3 选择“权限 > 创建自定义策略”。

- 策略名称：请输入策略名称。
- 作用范围：请选择“全局级服务”。
- 策略配置方式：请选择“可视化视图”。
- 策略内容：
  a. “允许”选择“允许”。
  b. “云服务”选择“对象存储服务 (OBS)”。
  c. “操作”勾选所有“写”、“列表”和“只读”权限。
  d. “特定资源”选择：
     i. “object”选择“通过资源路径指定”，并单击“添加资源路径”分别在“路径”中输入***obs_bucket_name*/tmp/**和***obs_bucket_name*/tmp/***。此处以**/tmp**目录为例，如需其他目录权限请参考该步骤添加对应目录及该目录下所有对象的资源路径。
     ii. “bucket”选择“通过资源路径指定”，并单击“添加资源路径”在“路径”中输入***obs_bucket_name***。

其中**obs_bucket-name**请使用实际的OBS桶名替换。若桶类型为“并行文件系统”需要再添加***obs_bucket_name*/tmp/**路径，桶类型为“对象存储”则不需要添加***obs_bucket_name*/tmp/**路径。

e. （可选）请求条件，暂不添加。

图 25-1 自定义策略

步骤4 单击“确定”完成策略添加。

步骤5 选择“委托”，并在**步骤1**中查询到的委托所在行的“操作”列单击“权限配置”。

步骤6 查询并勾选**步骤3**中创建的策略。

步骤7 单击“确定”完成委托权限配置。

----结束

25.2 使用 Hadoop 客户端删除 OBS 上数据时报.Trash 目录没有权限错误

问题描述

执行**hadoop fs -rm obs://<obs_path>**出现如下报错：

```
exception [java.nio.file.AccessDeniedException: user/root/.Trash/Current/: getFileStatus on user/root/.Trash/Current/: status [403]
```

问题原因

hadoop删除文件时会将文件先移动到.Trash目录，若该目录没有权限则出现403报错。

解决方法

方案一：

使用**hadoop fs -rm -skipTrash**命令来删除文件。

方案二：

在集群对应的委托中添加访问.Trash目录的权限。

步骤1 在集群“概览”页签中，查询并记录集群所绑定的委托名称。

步骤2 登录IAM服务控制台。

步骤3 选择“权限 > 创建自定义策略”。

- 策略名称：请输入策略名称。
- 作用范围：请选择“全局级服务”。
- 策略配置方式：请选择“可视化视图”。
- 策略内容：
  a. “允许”选择“允许”。
  b. “云服务”选择“对象存储服务 (OBS)”。
  c. “操作”勾选所有操作权限。
  d. “特定资源”选择：
     i. “object”选择“通过资源路径指定”，并单击“添加资源路径”分别在“路径”中输入.Trash目录，例如***obs_bucket_name*/user/root/.Trash/***。
     ii. “bucket”选择“通过资源路径指定”，并单击“添加资源路径”在“路径”中输入***obs_bucket_name***。

     其中**obs_bucket-name**请使用实际的OBS桶名替换。
  e. （可选）请求条件，暂不添加。

图 25-2 自定义策略

步骤4 单击“确定”完成策略添加。

步骤5 选择“委托”，并在**步骤1**中查询到的委托所在行的“操作”列单击“权限配置”。

步骤6 查询并勾选**步骤3**中创建的策略。

步骤7 单击“确定”完成委托权限配置。

步骤8 重新执行**hadoop fs -rm obs://<obs_path>**命令。

----结束

25.3 由于 MRS 集群节点的 NTP 时间不同步导致集群访问 OBS 时鉴权失败

问题现象

集群访问OBS上报403异常。

原因分析

集群Master节点NTP时间与集群外节点的NTP服务器时间不同步，时间相差超过15min，导致集群访问OBS时鉴权失败，上报403异常。

处理步骤

步骤1 以**root**用户登录集群主Master节点。

步骤2 修改“${BIGDATA_HOME}/om-server/om/inst/conf/oms-config.ini”和“${BIGDATA_HOME}/om-server/OMS/workspace0/conf/oms-config.ini”配置文件，在“ntp_server_ip”参数值前添加“ntp.myhuaweicloud.com,”

```
...
ntp_server_ip=ntp.myhuaweicloud.com,10.127.1.0
...
```

步骤3 以**root**用户登录集群备Master节点，执行**步骤2**。

步骤4 等待约2分钟，在主Master节点执行以下命令观察IP同步成功。

ntpq -np

例如执行后结果如下：

```
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*   .127.1.0      .LOCL.           0 l   10   16  377    0.000   +0.000   0.000
 100.79. .        10.44.  .        3 u   13   16  377    0.394  +47.857   0.280
```

----结束