MapReduce 服务

最佳实践

文档版本	01
发布日期	2021-04-13

华为云计算技术有限公司

商标声明


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 数据分析

1.1 使用 Spark2x 实现车联网车主驾驶行为分析

本手册基于华为云MapReduce服务实践所编写，用于指导您了解MRS的基本功能，利用MRS服务的Spark2x组件，对车主的驾驶行为进行分析统计，得到用户驾驶行为的分析结果。

说明

本实践仅适用于MRS 3.1.0版本，请按照指导创建集群。

本实践基本内容如下所示：

1. **场景描述**
2. **第一步：创建集群**
3. **第二步：准备Spark样例程序和样例数据**
4. **第三步：创建作业**
5. **第四步：查看作业执行结果**

场景描述

本次实践的原始数据为车主的驾驶行为信息，包括车主在日常的驾驶行为中，是否急加速、急减速、空挡滑行、超速、疲劳驾驶等信息。通过Spark2x组件的强大的分析能力，分析统计指定时间段内，车主急加速、急减速、空挡滑行、超速、疲劳驾驶等违法行为的次数。

创建集群

步骤1 进入**购买MRS集群页面**。

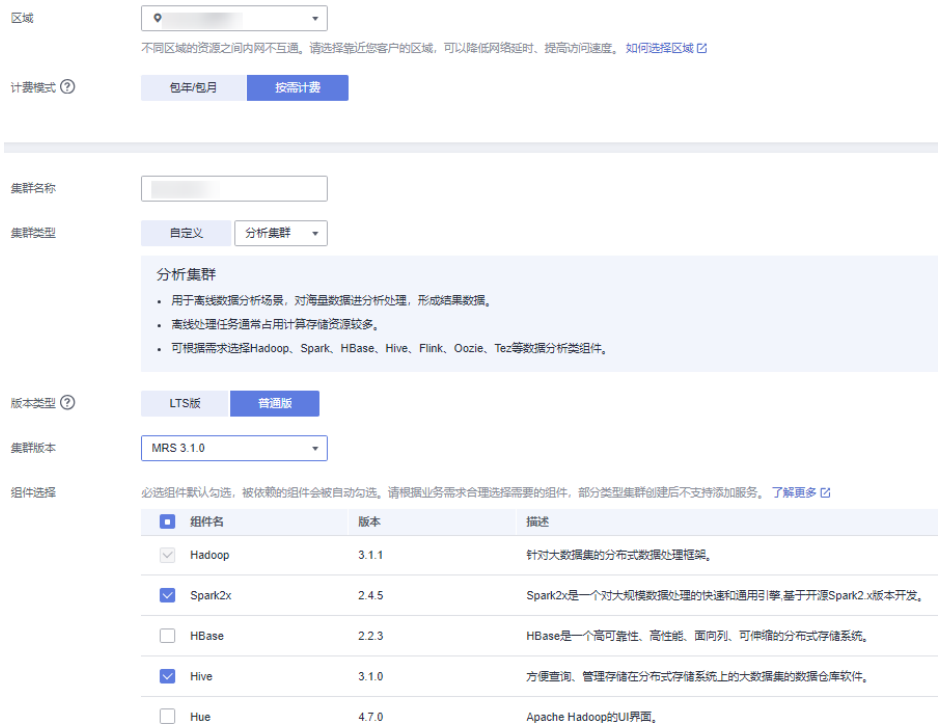
步骤2 选择“自定义购买”。

参见**表1-1**配置集群软件信息。

表 1-1 软件配置

参数名称	配置方式
区域	选择“华北-北京四” **说明** 本指导以“华北-北京四”为例进行介绍，如果您需要选择其他区域进行操作，请确保所有操作均在同一区域进行。
计费模式	按需计费
集群名称	mrs_demo
集群类型	选择“分析集群”，用来做离线数据分析
版本类型	选择“普通版”
集群版本	选择“MRS 3.1.0” **说明** 本实践仅适用于MRS 3.1.0版本。
组件选择	勾选所有组件
元数据	选择“本地元数据”

图 1-1 自定义购买-软件配置



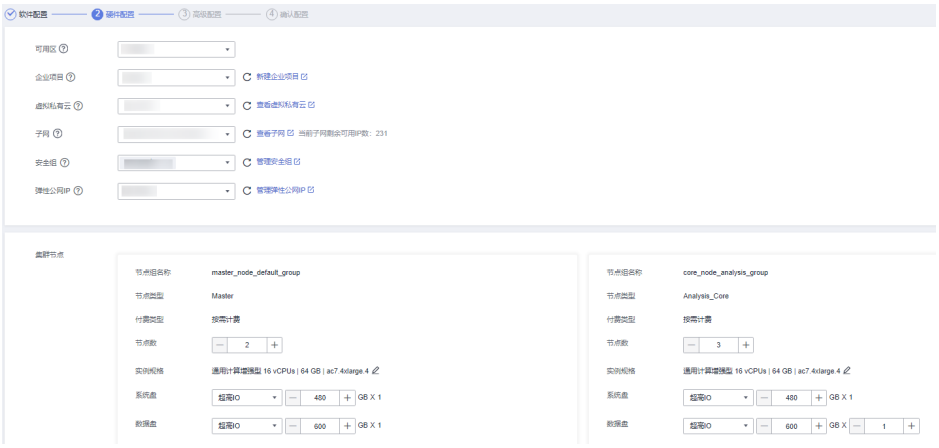
步骤3 单击“下一步”配置硬件信息。

参见**表1-2**配置集群硬件信息。

表 1-2 硬件配置

参数名称	配置方式
可用区	可用区2
企业项目	选择“default”。
虚拟私有云	选择需要创建集群的VPC，单击“查看虚拟私有云”进入VPC服务查看已创建的VPC名称和ID。如果没有VPC，需要创建一个新的VPC。
子网	选择需要创建集群的子网，可进入VPC服务查看VPC下已创建的子网名称和ID。如果VPC下未创建子网，请单击“创建子网”进行创建。
安全组	选择“自动创建”。
弹性公网IP	选择“暂不绑定”。
集群节点	保持默认值。

图 1-2 自定义购买-硬件配置



步骤4 单击“下一步”，高级配置页签参考**表1-3**配置以下信息，其他选项保持默认值。

表 1-3 高级配置

参数名称	配置方式
Kerberos认证	关闭Kerberos认证。
用户名	Manager管理员用户，目前默认为admin用户。
密码	配置Manager管理员用户的密码。
确认密码	再次输入Manager管理员用户的密码。
登录方式	选择“密码”。
用户名	用于登录弹性云服务器的用户，目前默认为root用户。

参数名称	配置方式
密码	配置登录ECS的用户密码。
确认密码	再次输入登录ECS的用户密码。

图 1-3 自定义购买-高级配置

步骤5 单击“下一步”，在“确认配置”页面检查配置集群信息，如需调整配置，可单击 ，跳转到对应页签后重新设置参数。

步骤6 勾选通信安全授权后，单击“立即购买”，进入任务提交成功页面。

步骤7 单击“返回集群列表”，可以查看到集群创建的状态。

集群创建需要时间，所创集群的初始状态为“启动中”，创建成功后状态更新为“运行中”，请您耐心等待。

----结束

准备 Spark2x 样例程序和样例数据

步骤1 创建OBS并行文件系统，用于存放Spark样例程序、样例数据、作业执行结果和日志。

1. 登录华为云管理控制台。
2. 在“服务列表”中，选择“存储 > 对象存储服务”。
3. 单击“并行文件系统 > 创建并行文件系统”，创建一个名称为“obs-demo-analysis-hwt4”的文件系统。策略等参数保持默认值。

图 1-4 创建并行文件系统

步骤2 单击文件系统名称。选择左侧导航栏“文件”，在“文件”页签下单击“新建文件夹”，分别新建program、input文件夹，如图1-5所示。

图 1-5 新建文件夹

步骤3 从 https://mrs-obs-cn-north-4.obs.cn-north-4.myhuaweicloud.com/mrs-demon-samples/demon/driver_behavior.jar路径下载样例程序driver_behavior.jar至本地。

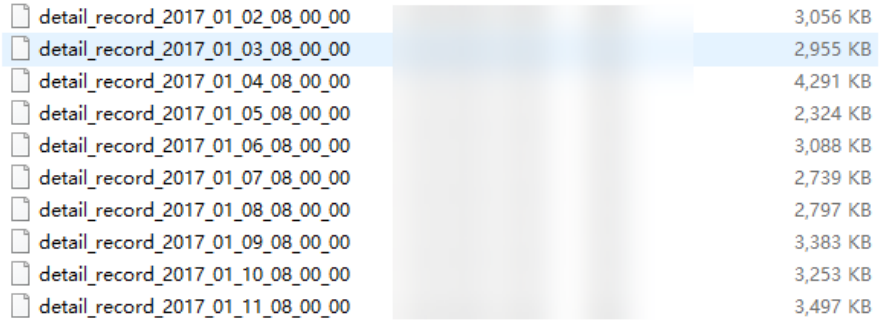
步骤4 进入“program”文件夹，单击“上传文件”，选择本地存放的driver_behavior.jar样例程序。

步骤5 单击“上传”，上传样例程序到OBS桶。

步骤6 从https://mrs-obs-cn-north-4.obs.cn-north-4.myhuaweicloud.com/mrs-demon-samples/demon/detail-records.zip获取Spark样例数据到本地。

步骤7 将下载的“detail-records.zip”解压，获取图1-6所示的样例数据。

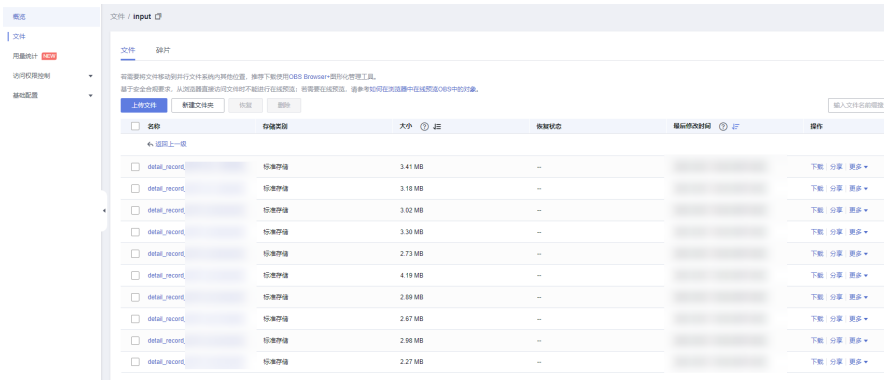
图 1-6 样例数据



步骤8 进入“input”文件夹，单击“上传文件”，选择本地存放的Spark样例数据。单击“上传”，上传样例数据到OBS文件系统。

说明

上传**步骤7**解压后的数据至“input”文件夹下，上传后如下图所示：



----结束

创建作业

步骤1 在MRS控制台左侧导航栏选择“现有集群”，单击名称为“mrs_demo”的集群。

步骤2 在集群信息页面选择“作业管理”页签，单击“添加”，进入添加作业页面。

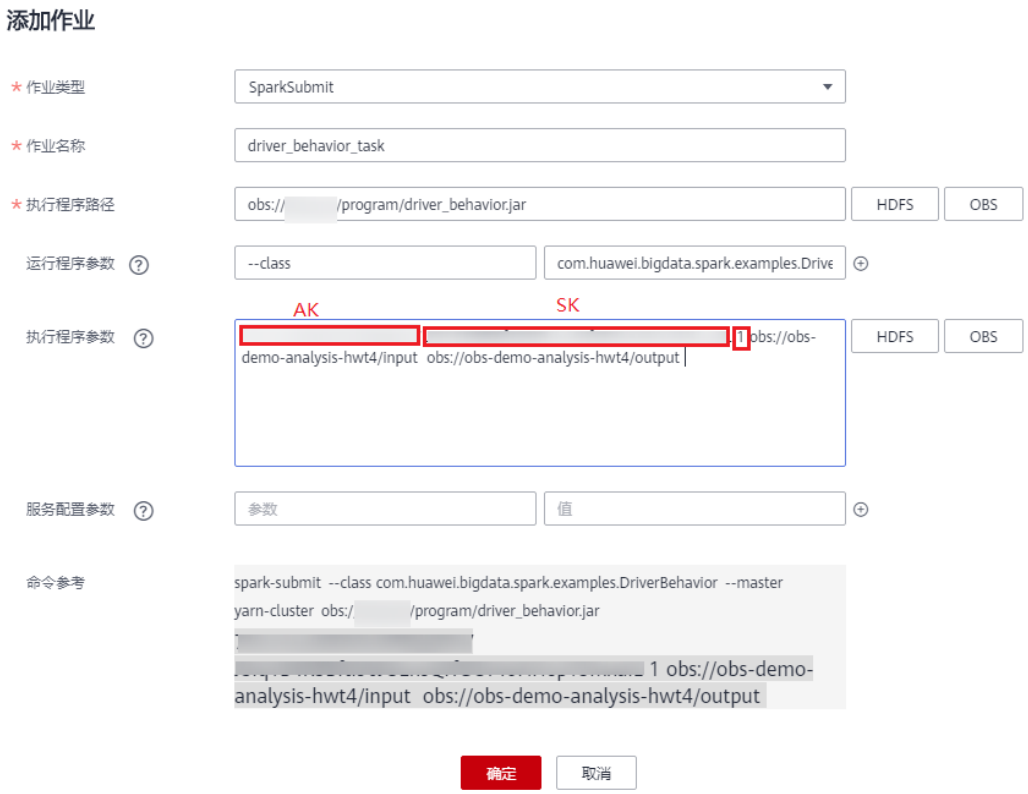
图 1-7 添加作业

步骤3 按**图1-8**完成作业参数配置。

表 1-4 配置作业信息

参数名称	配置方法
作业类型	选择“SparkSubmit”。
作业名称	输入“driver_behavior_task”。
执行程序路径	单击“OBS”，选择**准备Spark2x样例程序和样例数据**中上传的名称为driver_behavior.jar的jar包。
运行程序参数	参数选择“--class”，值输入“com.huawei.bigdata.spark.examples.DriverBehavior”。
执行程序参数	输入***“AK SK 1 输入路径 输出路径”***。 ● AK/SK请参考**说明**方式获取。 ● 1为固定输入，用于指定作业执行时调用的程序函数。 ● 输入路径可通过单击“OBS”进行选择输入路径。 ● 输出路径请手动输入一个不存在的目录，例如obs://obs-demo-analysis-hwt4/output/。 **说明** AK/SK，请通过如下方式获取。 1. 登录华为云管理控制台。 2. 单击右上角的用户名，然后选择“我的凭证”。 3. 系统跳转至“我的凭证”页面，单击“访问密钥”。 4. 单击“新增访问密钥”申请新密钥，按照提示输入密码与验证码之后，浏览器自动下载一个“credentials.csv”文件，文件为csv格式，以英文逗号分隔，中间的为AK，最后一个为SK。
服务配置参数	保持默认不配置。

图 1-8 添加作业



步骤4 单击“确定”，开始提交作业，执行程序。

----结束

查看作业执行结果

步骤1 进入“作业管理”页面，查看作业执行状态。

图 1-9 作业执行状态

步骤2 等待1~2分钟，登录OBS控制台，进入obs-demo-analysis-hwt4文件系统的output目录中，查看执行结果，在生成的csv文件所在行的“操作”列单击“下载”按钮将该文件下载到本地。

图 1-10 下载作业执行结果

步骤3 在本地将下载后的csv文件使用Excel文本打开，按照样例程序中定义的字段为每列数据进行分类，得到如下图所示作业执行结果。

图 1-11 执行结果

driverID	车牌号	急加速次数	急刹车次数	空挡滑行次数	空挡滑行时间总和	超速次数	超速时间总和	疲劳驾驶次数	停车熄油门次数	漏油次数
panxian1000005	华AX542C	395	434	330	2930	3531	33946	4307	417	441
zouan1000007	华A58M83	360	385	315	2997	3181	31248	3594	389	385
zengpeng1000000	华AZQ110	340	344	272	2894	2763	25479	3274	284	337
xiexiao1000001	华AEB132	264	261	248	2525	2324	23434	2720	314	253
duxu1000009	华AT75H8	238	284	247	2632	2301	22338	2814	264	248
hanhui1000002	华AZI419	401	444	327	2844	3349	31813	3997	433	371
shenxian1000004	华ADJ750	374	356	297	2810	3126	31494	3767	383	366
likun1000003	华AVM936	341	354	291	3043	3044	28728	3552	347	376
haowei1000008	华A709GB	321	314	255	2659	2639	25522	3204	312	318
xiezhi1000006	华A6CU11	255	310	254	2074	2535	23942	2931	312	279

----结束

1.2 使用 Hive 加载 HDFS 数据并分析图书评分情况

MRS离线处理集群，可对海量数据进行分析和处理，形成结果数据，供下一步数据应用使用。

离线处理对处理时间要求不高，但是所处理数据量较大，占用计算存储资源较多，通常通过Hive/SparkSQL引擎或者MapReduce/Spark2x实现。

本实践基于华为云MapReduce服务，用于指导您创建MRS集群后，使用Hive对原始数据进行导入、分析等操作，展示了如何构建弹性、低成本的离线大数据分析。

基本内容如下所示：

1. **创建MRS离线查询集群**。
2. **将本地数据导入到HDFS中**。
3. **创建Hive表**。
4. **将原始数据导入Hive并进行分析**。

场景描述

Hive是建立在Hadoop上的数据仓库框架，提供大数据平台批处理计算能力，能够对结构化/半结构化数据进行批量分析汇总完成数据计算。提供类似SQL的Hive Query Language语言操作结构化数据，其基本原理是将HQL语言自动转换成MapReduce任务，从而完成对Hadoop集群中存储的海量数据进行查询和分析。

Hive主要特点如下：

- 海量结构化数据分析汇总。
- 将复杂的MapReduce编写任务简化为SQL语句。
- 灵活的数据存储格式，支持JSON，CSV，TEXTFILE，RCFILE，SEQUENCEFILE，ORC（Optimized Row Columnar）这几种存储格式。

本实践以某图书网站后台用户的点评数据为原始数据，导入Hive表后通过SQL命令筛选出最受欢迎的畅销图书。

创建 MRS 离线查询集群

1. 进入**购买MRS集群页面**。
2. 选择“快速购买”，填写软件配置参数。

表 1-5 软件配置（以下参数仅供参考，可根据实际情况调整）

参数项	取值
区域	华北-北京四
计费模式	按需计费
集群名称	MRS_demo
版本类型	普通版
集群版本	MRS 3.1.0
组件选择	Hadoop分析集群
可用区	可用区1
虚拟私有云	vpc-01
子网	subnet-01
企业项目	default
Kerberos认证	不开启
用户名	admin/root
密码	设置密码登录集群管理页面及ECS节点用户的密码，例如：Test!@12345。
确认密码	再次输入设置用户密码
通信安全授权	勾选“确认授权”

图 1-12 购买 Hadoop 分析集群

3. 单击“立即购买”，等待MRS集群创建成功。

图 1-13 集群购买成功

将本地数据导入到 HDFS 中

1. 在本地已获取某图书网站后台图书点评记录的原始数据文件“book_score.txt”，例如内容如下。

   字段信息依次为：用户ID、图书ID、图书评分、备注信息

   例如部分数据节选如下：

```
202001,242,3,Good!
202002,302,3,Test.
202003,377,1,Bad!
220204,51,2,Bad!
202005,346,1,aaa
202006,474,4,None
202007,265,2,Bad!
202008,465,5,Good!
202009,451,3,Bad!
202010,86,3,Bad!
202011,257,2,Bad!
202012,465,4,Good!
202013,465,4,Good!
202014,465,4,Good!
202015,302,5,Good!
202016,302,3,Good!
...
```

2. 登录对象存储服务OBS控制台，单击“创建桶”，填写以下参数，单击“立即创建”。

表 1-6 桶参数

参数项	取值
区域	华北-北京四
数据冗余存储策略	单AZ存储
桶名称	mrs-hive
默认存储类别	标准存储
桶策略	私有
归档数据直读	关闭
企业项目	default
标签	-

图 1-14 创建 OBS 桶

等待桶创建好，单击桶名称，选择“对象 > 上传对象”，将数据文件上传至OBS桶内。

图 1-15 上传对象

3. 切换回MRS控制台，单击创建好的MRS集群名称，进入“概览”，单击“IAM用户同步”所在行的“同步”，等待约5分钟同步完成。

图 1-16 同步 IAM 用户

4. 将数据文件上传HDFS。
   a. 在“文件管理”页签，选择“HDFS文件列表”，进入数据存储目录，如“/tmp/test”。

      “/tmp/test”目录仅为示例，可以是界面上的任何目录，也可以通过“新建”创建新的文件夹。
   b. 单击“导入数据”。
      - OBS路径：选择上面创建好的OBS桶名，找到“book_score.txt”文件，勾选“我确认所选脚本安全，了解可能存在的风险，并接受对集群可能造成的异常或影响。”，单击“确定”。
      - HDFS路径：选择“/tmp/test”，单击“确定”。

图 1-17 从 OBS 导入数据到 HDFS

c. 单击“确定”，等待数据导入成功，此时数据文件已上传至MRS集群的HDFS文件系统内。

图 1-18 数据导入成功

创建 Hive 表

1. 下载并安装集群全量客户端，例如在主Master节点上安装，客户端安装目录为“/opt/client”，相关操作可参考**安装客户端**。

   也可直接使用Master节点中自带的集群客户端，安装目录为“/opt/Bigdata/client”。

2. 为主Master节点绑定一个弹性IP并在安全组中放通22端口，然后使用**root**用户登录主Master节点，进入客户端所在目录并加载变量。

   cd /opt/client

   source bigdata_env

3. 执行**beeline -n 'hdfs'**命令进入Hive Beeline命令行界面。

   执行以下命令创建一个与原始数据字段匹配的Hive表：

   create table bookscore (userid int,bookid int,score int,remarks string) row format delimited fields terminated by ','stored as textfile;

4. 查看表是否创建成功：

show tables;

```
+------------+
|  tab_name  |
+------------+
| bookscore  |
+------------+
```

将原始数据导入 Hive 并进行分析

1. 继续在Hive Beeline命令行中执行以下命令，将已导入HDFS的原始数据导入Hive表中。

   load data inpath '/tmp/test/book_score.txt' into table bookscore;

2. 数据导入完成后，执行如下命令，查看Hive表内容。

   select * from bookscore;

```
+-------------------+-------------------+-------------------+--------------------+
| bookscore.userid  | bookscore.bookid  | bookscore.score  | bookscore.remarks  |
+-------------------+-------------------+-------------------+--------------------+
| 202001            | 242               | 3                | Good!              |
| 202002            | 302               | 3                | Test.              |
| 202003            | 377               | 1                | Bad!               |
| 220204            | 51                | 2                | Bad!               |
| 202005            | 346               | 1                | aaa                |
| 202006            | 474               | 4                | None               |
| 202007            | 265               | 2                | Bad!               |
| 202008            | 465               | 5                | Good!              |
| 202009            | 451               | 3                | Bad!               |
| 202010            | 86                | 3                | Bad!               |
| 202011            | 257               | 2                | Bad!               |
| 202012            | 465               | 4                | Good!              |
| 202013            | 465               | 4                | Good!              |
| 202014            | 465               | 4                | Good!              |
| 202015            | 302               | 5                | Good!              |
| 202016            | 302               | 3                | Good!              |
...
```

   执行以下命令统计表行数：

   select count(*) from bookscore;

```
+------+
| _c0  |
+------+
| 32   |
+------+
```

3. 执行以下命令，等待MapReduce任务完成后，筛选原始数据中累计评分最高的图书top3。

   select bookid,sum(score) as summarize from bookscore group by bookid order by summarize desc limit 3;

   例如最终显示内容如下：

```
...
INFO  : 2021-10-14 19:53:42,427 Stage-2 map = 0%,  reduce = 0%
INFO  : 2021-10-14 19:53:49,572 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 2.15 sec
INFO  : 2021-10-14 19:53:56,713 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 4.19 sec
INFO  : MapReduce Total cumulative CPU time: 4 seconds 190 msec
INFO  : Ended Job = job_1634197207682_0025
INFO  : MapReduce Jobs Launched:
INFO  : Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.24 sec   HDFS Read: 7872 HDFS Write: 322 SUCCESS
INFO  : Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 4.19 sec   HDFS Read: 5965 HDFS Write: 143 SUCCESS
INFO  : Total MapReduce CPU Time Spent: 8 seconds 430 msec
INFO  : Completed executing command(queryId=omm_20211014195310_cf669633-5b58-4bd5-9837-73286ea83409); Time taken:
```

```
47.388 seconds
INFO  : OK
INFO  : Concurrency mode is disabled, not creating a lock manager
+---------+------------+
| bookid  | summarize  |
+---------+------------+
| 465     | 170        |
| 302     | 110        |
| 474     | 88         |
+---------+------------+
3 rows selected (47.469 seconds)
```

以上内容表示，ID为456、302、474的3本书籍，为累计评分最高的Top3图书。

1.3 使用 Hive 加载 OBS 数据并分析企业雇员信息

MRS Hadoop分析集群，提供Hive、Spark离线大规模分布式数据存储和计算，进行海量数据分析与查询。

本实践基于华为云MapReduce服务，用于指导您创建MRS集群后，使用Hive对OBS中存储的原始数据进行导入、分析等操作，展示了如何构建弹性、低成本的存算分离大数据分析。

基本内容如下所示：

1. **创建MRS离线查询集群**。
2. **创建OBS委托并绑定至MRS集群**。
3. **创建Hive表并加载OBS中数据**。
4. **基于HQL对数据进行分析**。

场景描述

Hive是建立在Hadoop上的数据仓库框架，提供大数据平台批处理计算能力，能够对结构化/半结构化数据进行批量分析汇总完成数据计算。提供类似SQL的Hive Query Language语言操作结构化数据，其基本原理是将HQL语言自动转换成MapReduce任务，从而完成对Hadoop集群中存储的海量数据进行查询和分析。

Hive主要特点如下：

- 海量结构化数据分析汇总。
- 将复杂的MapReduce编写任务简化为SQL语句。
- 灵活的数据存储格式，支持JSON，CSV，TEXTFILE，RCFILE，SEQUENCEFILE，ORC（Optimized Row Columnar）这几种存储格式。

本实践以用户开发一个Hive数据分析应用为例，通过客户端连接Hive后，执行HQL语句访问OBS中的Hive数据。进行企业雇员信息的管理、查询。如果需要基于MRS服务提供的样例代码工程开发构建应用，您可以参考**Hive应用开发简介**。

本实践中，雇员信息的原始数据包含以下两张表：

表 1-7 表 1 雇员信息数据

编号	姓名	支付薪水币种	薪水金额	纳税税种	工作地	入职时间
1	Wang	R	8000.01	personal income tax&0.05	China:Shenzhen	2014
3	Tom	D	12000.02	personal income tax&0.09	America:NewYork	2014
4	Jack	D	24000.03	personal income tax&0.09	America:Manhattan	2015
6	Linda	D	36000.04	personal income tax&0.09	America:NewYork	2014
8	Zhang	R	9000.05	personal income tax&0.05	China:Shanghai	2014

表 1-8 雇员联络信息数据

编号	电话	邮箱
1	135 XXXX XXXX	xxxx@xx.com
3	159 XXXX XXXX	xxxxx@xx.com.cn
4	186 XXXX XXXX	xxxx@xx.org
6	189 XXXX XXXX	xxxx@xxx.cn
8	134 XXXX XXXX	xxxx@xxxx.cn

通过数据应用，进行以下分析：

- 查看薪水支付币种为美元的雇员联系方式。
- 查询入职时间为2014年的雇员编号、姓名等字段，并将查询结果加载到新表中。
- 统计雇员信息共有多少条记录。
- 查询使用以“cn”结尾的邮箱的员工信息。

创建 MRS 离线查询集群

1. 进入**购买MRS集群页面**。
2. 选择“快速购买”，填写软件配置参数。

表 1-9 软件配置（以下参数仅供参考，可根据实际情况调整）

参数项	取值
区域	华北-北京四
计费模式	按需计费
集群名称	MRS_demo
版本类型	普通版
集群版本	MRS 3.1.0
组件选择	Hadoop分析集群
可用区	可用区1
虚拟私有云	vpc-01
子网	subnet-01
企业项目	default
Kerberos认证	不开启
用户名	admin/root
密码	设置密码登录集群管理页面及ECS节点用户的密码，例如：Test!@12345。
确认密码	再次输入设置用户密码
通信安全授权	勾选“确认授权”

图 1-19 购买 Hadoop 分析集群

3. 单击“立即购买”，等待MRS集群创建成功。

图 1-20 集群创建成功

创建 OBS 委托并绑定至 MRS 集群

说明

- MRS在IAM的委托列表中预置了**MRS_ECS_DEFAULT_AGENCY**委托，可在创建自定义过程中可以直接选择该委托，该委托拥有对象存储服务的OBSOperateAccess权限和在集群所在区域拥有CESFullAccess（对开启细粒度策略的用户）、CES Administrator和KMS Administrator权限。
- 如需使用自定义委托，请参考如下步骤进行创建委托（创建或修改委托需要用户具有Security Administrator权限）。

1. 登录华为云管理控制台。
2. 在服务列表中选择“管理与监管 > 统一身份认证服务 IAM”。
3. 选择“委托 > 创建委托”。
4. 设置“委托名称”，“委托类型”选择“云服务”，在“云服务”中选择“弹性云服务器ECS 裸金属服务器BMS”，授权ECS或BMS调用OBS服务。
5. “持续时间”选择“永久”并单击“下一步”。

图 1-21 创建委托

6. 在弹出授权页面的搜索框内，搜索“OBS OperateAccess”策略，勾选“OBS OperateAccess”策略。

图 1-22 配置权限

7. 单击“下一步”，选择权限范围方案，默认选择“所有资源”，单击“展开其他方案”，选择“全局服务资源”。
8. 在弹出的提示框中单击“知道了”，开始授权。界面提示“授权成功。”，单击“完成”，委托成功创建。
9. 返回MRS控制台，在集群列表中，单击已创建好的MRS集群名称，在集群的“概览”页面中，单击“管理委托”，选择创建好的OBS委托后单击“确定”。

图 1-23 进入 MRS 集群的概览界面

图 1-24 为集群绑定委托

创建 Hive 表并加载 OBS 中数据

1. 在服务列表中选择“存储 > 对象存储服务 OBS”，登录OBS控制台，单击“并行文件系统 > 创建并行文件系统”，填写以下参数，单击“立即创建”。

表 1-10 并行文件系统参数

参数项	取值
区域	华北-北京四
文件系统名称	hiveobs

参数项	取值
数据冗余存储策略	单AZ存储
策略	私有
归档数据直读	关闭
企业项目	default
标签	-

2. 下载并安装MRS集群客户端，例如在主Master节点上安装，客户端安装目录为“/opt/client”，相关操作可参考**安装客户端**。

   也可直接使用Master节点中自带的集群客户端，安装目录为“/opt/Bigdata/client”。

3. 为主Master节点绑定一个弹性IP并在安全组中放通22端口，然后使用**root**用户登录主Master节点，进入客户端所在目录并加载变量。

   cd /opt/client

   source bigdata_env

4. 执行**beeline**命令进入Hive Beeline命令行界面。

   执行以下命令创建一个与原始数据字段匹配的雇员信息数据表“employees_info”：

   ```
   create external table if not exists employees_info
   (
   id INT,
   name STRING,
   usd_flag STRING,
   salary DOUBLE,
   deductions MAP<STRING, DOUBLE>,
   address STRING,
   entrytime STRING
   )
   row format delimited fields terminated by ',' map keys terminated by '&'
   stored as textfile
   location 'obs://hiveobs/employees_info';
   ```

   执行以下命令创建一个与原始数据字段匹配的雇员联系信息数据表“employees_contact”：

   ```
   create external table if not exists employees_contact
   (
   id INT,
   phone STRING,
   email STRING
   )
   row format delimited fields terminated by ','
   ```

stored as textfile

location 'obs://hiveobs/employees_contact';

5. 查看表是否创建成功：

   show tables;

```
+--------------------------+
|        tab_name          |
+--------------------------+
| employees_contact        |
| employees_info           |
+--------------------------+
```

6. 将数据导入OBS对应表目录下。

   Hive内部表会默认在指定的存储空间中建立对应文件夹，只要把文件放入，表就可以读取到数据（需要和表结构匹配）。

   登录OBS控制台，在已创建的文件系统的“文件”页面，将本地的原始数据分别上传至生成的“employees_info”、“employees_contact”文件夹下。

   图 1-25 上传数据

   例如原始数据格式如下：

   info.txt：

```
1,Wang,R,8000.01,personal income tax&0.05,China:Shenzhen,2014
3,Tom,D,12000.02,personal income tax&0.09,America:NewYork,2014
4,Jack,D,24000.03,personal income tax&0.09,America:Manhattan,2015
6,Linda,D,36000.04,personal income tax&0.09,America:NewYork,2014
8,Zhang,R,9000.05,personal income tax&0.05,China:Shanghai,2014
```

   contact.txt：

```
1,135 XXXX XXXX,xxxx@xx.com
3,159 XXXX XXXX,xxxx@xx.com.cn
4,189 XXXX XXXX,xxxx@xx.org
6,189 XXXX XXXX,xxxx@xx.cn
8,134 XXXX XXXX,xxxx@xxx.cn
```

7. 在Hive Beeline客户端中，执行以下命令，查询源数据是否被正确加载。

   select * from employees_info;

```
+---------------------+-----------------------+--------------------------+-----------------------
+-------------------------------+-----------------------+--------------------------+
| employees_info.id  | employees_info.name  | employees_info.usd_flag  | employees_info.salary  |
employees_info.deductions  | employees_info.address  | employees_info.entrytime  |
+---------------------+-----------------------+--------------------------+-----------------------
+-------------------------------+-----------------------+--------------------------+
| 1                  | Wang                 | R                        | 8000.01                | {"personal income
tax":0.05}  | China:Shenzhen         | 2014                      |
| 3                  | Tom                  | D                        | 12000.02               | {"personal income
tax":0.09}  | America:NewYork        | 2014                      |
| 4                  | Jack                 | D                        | 24000.03               | {"personal income tax":0.09}
| America:Manhattan       | 2015                      |
```

```
| 6                | Linda              | D                    | 36000.04             | {"personal income
tax":0.09}  | America:NewYork         | 2014                  |
| 8                | Zhang              | R                    | 9000.05              | {"personal income
tax":0.05}  | China:Shanghai          | 2014                  |
+------------------+--------------------+----------------------+----------------------
+------------------------------+----------------------+--------------------------+
```

select * from employees_contact;

```
+----------------------+-------------------------+-------------------------+
| employees_contact.id  | employees_contact.phone  | employees_contact.email  |
+----------------------+-------------------------+-------------------------+
| 1                    | 135 XXXX XXXX           | xxxx@xx.com             |
| 3                    | 159 XXXX XXXX           | xxxx@xx.com.cn          |
| 4                    | 186 XXXX XXXX           | xxxx@xx.org             |
| 6                    | 189 XXXX XXXX           | xxxx@xx.cn              |
| 8                    | 134 XXXX XXXX           | xxxx@xxx.cn             |
+----------------------+-------------------------+-------------------------+
```

基于 HQL 对数据进行分析

在Hive Beeline客户端中，执行HQL语句，对原始数据进行分析。

1. 查看薪水支付币种为美元的雇员联系方式。

   创建新数据表进行数据清洗。

   create table employees_info_v2 as select id, name, regexp_replace(usd_flag, '\s+','') as usd_flag, salary, deductions, address, entrytime from employees_info;

   等待Map任务完成后，执行以下命令

   select a.* from employees_info_v2 a inner join employees_contact b on a.id = b.id where a.usd_flag='D';

   ```
   INFO  : MapReduce Jobs Launched:
   INFO  : Stage-Stage-3: Map: 1   Cumulative CPU: 2.95 sec   HDFS Read: 8483 HDFS Write: 317
   SUCCESS
   INFO  : Total MapReduce CPU Time Spent: 2 seconds 950 msec
   INFO  : Completed executing command(queryId=omm_20211022162303_c26d4f1b-
   a577-4d6c-919c-6cb96095b24b); Time taken: 26.259 seconds
   INFO  : OK
   INFO  : Concurrency mode is disabled, not creating a lock manager
   +-------+---------+-------------+-----------+-------------------------------+--------------------+--------------+
   | a.id  | a.name  | a.usd_flag  | a.salary  |         a.deductions          |     a.address      | a.entrytime  |
   +-------+---------+-------------+-----------+-------------------------------+--------------------+--------------+
   | 3     | Tom     | D           | 12000.02  | {"personal income tax":0.09}  | America:NewYork    | 2014         |
   | 4     | Jack    | D           | 24000.03  | {"personal income tax":0.09}  | America:Manhattan  | 2015         |
   | 6     | Linda   | D           | 36000.04  | {"personal income tax":0.09}  | America:NewYork    | 2014         |
   +-------+---------+-------------+-----------+-------------------------------+--------------------+--------------+
   3 rows selected (26.439 seconds)
   ```

2. 查询入职时间为2014年的雇员编号、姓名等字段，并将查询结果加载进表employees_info_extended中的入职时间为2014的分区中。

   创建一个表：

   create table if not exists employees_info_extended (id int, name string, usd_flag string, salary double, deductions map<string, double>, address string) partitioned by (entrytime string) stored as textfile;

   执行以下命令，在表中写入数据：

   insert into employees_info_extended partition(entrytime='2014') select id,name,usd_flag,salary,deductions,address from employees_info_v2 where entrytime = '2014';

   数据抽取成功后，查询表数据。

   select * from employees_info_extended;

```
+-----------------------------+-----------------------------+---------------------------------
+---------------------------------+-------------------------------------+---------------------------------
+-----------------------------------+
| employees_info_extended.id  | employees_info_extended.name  | employees_info_extended.usd_flag  |
employees_info_extended.salary  | employees_info_extended.deductions  |
employees_info_extended.address  | employees_info_extended.entrytime  |
+-----------------------------+-----------------------------+---------------------------------
+---------------------------------+-------------------------------------+---------------------------------
+-----------------------------------+
| 1                           | Wang                          | R                           | 8000.01                     |
{"personal income tax":0.05}        | China:Shenzhen                | 2014                          |
| 3                           | Tom                           | D                           | 12000.02                    |
{"personal income tax":0.09}        | America:NewYork               | 2014                          |
| 6                           | Linda                         | D                           | 36000.04                    |
{"personal income tax":0.09}        | America:NewYork               | 2014                          |
| 8                           | Zhang                         | R                           | 9000.05                     |
{"personal income tax":0.05}        | China:Shanghai                | 2014                          |
+-----------------------------+-----------------------------+---------------------------------
+---------------------------------+-------------------------------------+---------------------------------
+-----------------------------------+
```

3. 统计雇员信息有多少条记录。

   select count(1) from employees_info_v2;

```
+------+
| _c0  |
+------+
| 5    |
+------+
```

4. 查询使用以“cn”结尾的邮箱的员工信息。

   select a.*, b.email from employees_info_v2 a inner join employees_contact b on a.id = b.id where b.email rlike '.*cn$';

```
+-------+---------+-------------+-----------+--------------------------------+------------------+--------------
+-----------------+
| a.id  | a.name  | a.usd_flag  | a.salary  |         a.deductions           |    a.address     | a.entrytime  |
b.email     |
+-------+---------+-------------+-----------+--------------------------------+------------------+--------------
+-----------------+
| 3     | Tom     | D           | 12000.02  | {"personal income tax":0.09}  | America:NewYork  | 2014         |
xxxx@xx.com.cn  |
| 6     | Linda   | D           | 36000.04  | {"personal income tax":0.09}  | America:NewYork  | 2014         |
xxxx@xx.cn      |
| 8     | Zhang   | R           | 9000.05   | {"personal income tax":0.05}  | China:Shanghai   | 2014         |
xxxx@xxx.cn     |
+-------+---------+-------------+-----------+--------------------------------+------------------+--------------
+-----------------+
```

1.4 通过 Flink 作业处理 OBS 数据

应用场景

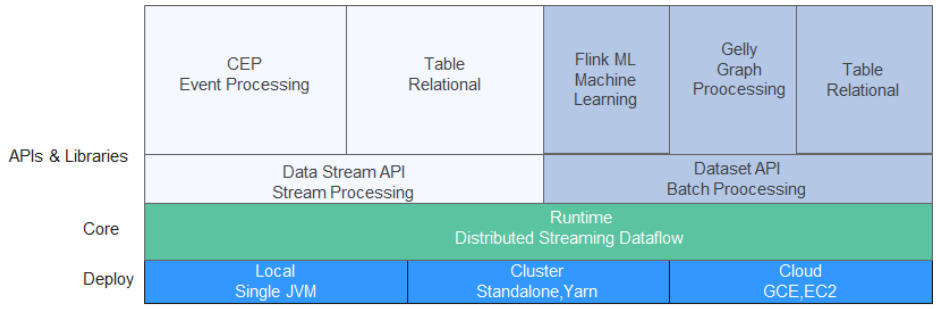
MRS支持在大数据存储容量大、计算资源需要弹性扩展的场景下，用户将数据存储在OBS服务中，使用MRS集群仅做数据计算处理的存算分离模式。

本文将向您介绍如何在MRS集群中运行Flink作业来处理OBS中存储的数据。

方案架构

Flink是一个批处理和流处理结合的统一计算框架，其核心是一个提供了数据分发以及并行化计算的流数据处理引擎。它的最大亮点是流处理，是业界最顶级的开源流处理引擎。

Flink最适合的应用场景是低时延的数据处理（Data Processing）场景：高并发pipeline处理数据，时延毫秒级，且兼具可靠性。

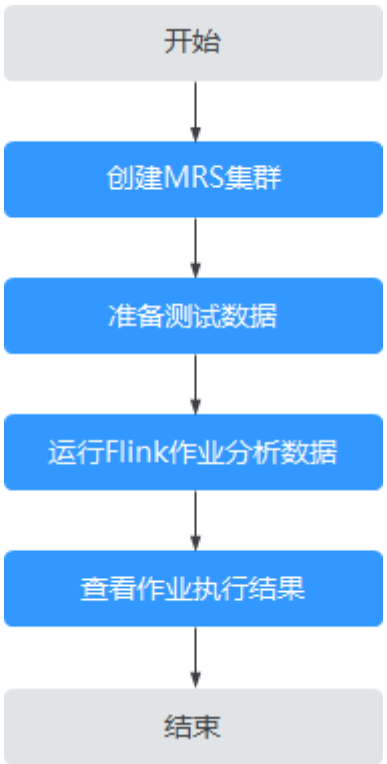


在本示例中，使用MRS集群内置的Flink WordCount作业程序，来分析OBS文件系统中保存的源数据，以统计源数据中的单词出现次数，相关程序代码信息介绍可参考**https://github.com/apache/flink/tree/master/flink-examples/flink-examples-batch/src/main/java/org/apache/flink/examples/java/wordcount**。

当然您也可以获取**MRS服务样例代码工程**，参考**Flink开发指南**开发其他Flink流作业程序。

操作流程

本示例操作流程如下：



步骤 1：创建 MRS 集群

创建并购买一个包含有Flink组件的MRS集群，详情请参见**购买自定义集群**。

说明

本文以购买的MRS 3.1.0版本的集群为例，集群未开启Kerberos认证。

在本示例中，由于要分析处理OBS文件系统中的数据，因此在集群的高级配置参数中要为MRS集群绑定IAM权限委托，使得集群内组件能够对接OBS并具有对应文件系统目录的操作权限。

您可以直接选择系统默认的“MRS_ECS_DEFAULT_AGENCY”，也可以自行创建其他具有OBS文件系统操作权限的自定义委托。

集群购买成功后，在MRS集群的任一节点内，使用**omm**用户安装集群客户端，具体操作可参考**安装并使用集群客户端**。

例如客户端安装目录为“/opt/client”。

步骤 2：准备测试数据

在创建Flink作业进行数据分析前，需要在提前准备待分析的测试数据，并将该数据上传至OBS文件系统中。

步骤1 本地创建一个“mrs_flink_test.txt”文件，例如文件内容如下：

```
This is a test demo for MRS Flink. Flink is a unified computing framework that supports both batch processing and stream processing. It provides a stream data processing engine that supports data distribution and parallel computing.
```

步骤2 在云服务列表中选择“存储 > 对象存储服务”，登录OBS管理控制台。

步骤3 单击“并行文件系统”，创建一个并行文件系统，并上传测试数据文件。

例如创建的文件系统名称为“mrs-demo-data”，单击系统名称，在“文件”页面中，新建一个文件夹“flink”，上传测试数据至该目录中。

则本示例的测试数据完整路径为“obs://mrs-demo-data/flink/mrs_flink_test.txt”。

图 1-26 上传测试数据

步骤4 （可选）上传数据分析应用程序。

使用管理台界面直接提交作业时，将已开发好的Flink应用程序jar文件也可以上传至OBS文件系统中，或者MRS集群内的HDFS文件系统中。

本示例中使用MRS集群内置的Flink WordCount样例程序，可从MRS集群的客户端安装目录中获取，即“/opt/client/Flink/flink/examples/batch/WordCount.jar”。

将“WordCount.jar”上传至“mrs-demo-data/program”目录下。

----结束

步骤 3：创建并运行 Flink 作业

方式1：在控制台界面在线提交作业。

步骤1 登录MRS管理控制台，单击MRS集群名称，进入集群详情页面。

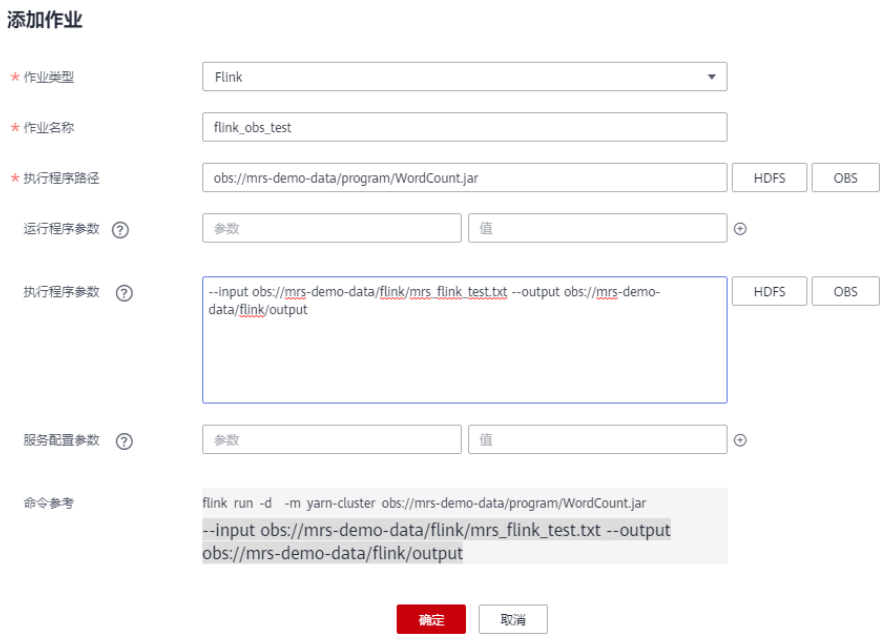
步骤2 在集群详情页的“概览”页签，单击“IAM用户同步”右侧的“单击同步”进行IAM用户同步。

步骤3 单击“作业管理”，进入“作业管理”页签。

步骤4 单击“添加”，添加一个Flink作业。

- 作业类型：Flink
- 作业名称：自定义，例如flink_obs_test。
- 执行程序路径：本示例使用Flink客户端的WordCount程序为例。
- 运行程序参数：使用默认值。
- 执行程序参数：设置应用程序的输入参数，“input”为待分析的测试数据，“output”为结果输出文件。

  例如本示例中，设置为“--input obs://mrs-demo-data/flink/mrs_flink_test.txt --output obs://mrs-demo-data/flink/output”。
- 服务配置参数：使用默认值即可，如需手动配置作业相关参数，可参考**运行Flink作业**。



步骤5 确认作业配置信息后，单击“确定”，完成作业的新增，并等待运行完成。

----结束

方式2：通过集群客户端提交作业。

步骤1 使用**root**用户登录集群客户端节点，进入客户端安装目录。

su - omm

cd /opt/client

source bigdata_env

步骤2 执行以下命令验证集群是否可以访问OBS。

hdfs dfs -ls obs://mrs-demo-data/flink

步骤3 提交Flink作业，指定源文件数据进行消费。

flink run -m yarn-cluster /opt/client/Flink/flink/examples/batch/WordCount.jar --input obs://mrs-demo-data/flink/mrs_flink_test.txt --output obs://mrs-demo/data/flink/output2

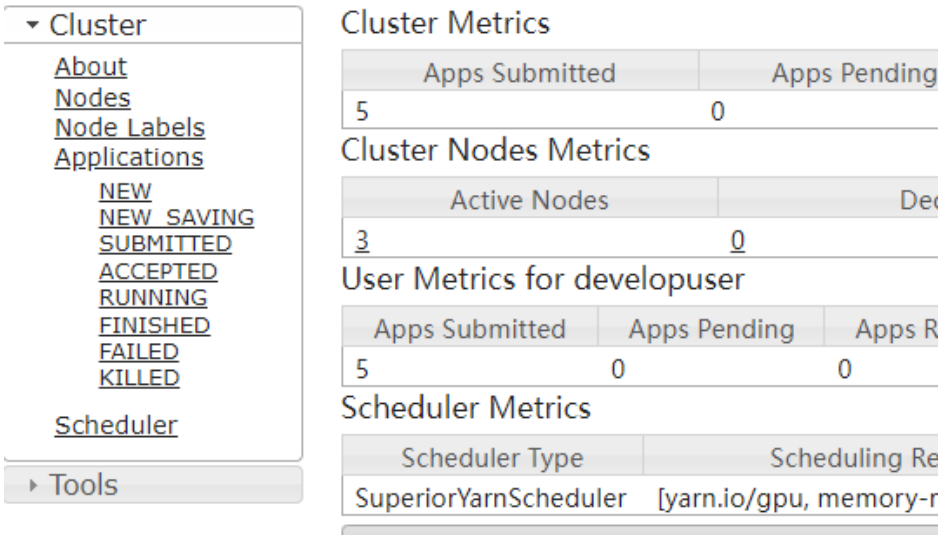
```
...
Cluster started: Yarn cluster with application id application_1654672374562_0011
Job has been submitted with JobID a89b561de5d0298cb2ba01fbc30338bc
Program execution finished
Job with JobID a89b561de5d0298cb2ba01fbc30338bc has finished.
Job Runtime: 1200 ms
```

----结束

步骤 4：查看作业执行结果

步骤1 作业提交成功后，登录MRS集群的FusionInsight Manager界面，单击“集群 > 服务 > Yarn”。

步骤2 单击“ResourceManager WebUI”后的链接进入Yarn Web UI界面，在Applications页面查看当前Yarn作业的详细运行情况及运行日志。



步骤3 等待作业运行完成后，在OBS文件系统中指定的结果输出文件中可查看数据分析输出的结果。

下载“output”文件到本地并打开，可查看输出的分析结果。

```
a 3
and 2
batch 1
both 1
computing 2
data 2
demo 1
distribution 1
engine 1
flink 2
for 1
framework 1
is 2
it 1
mrs 1
parallel 1
processing 3
provides 1
stream 2
supports 2
test 1
that 2
this 1
unified 1
```

使用集群客户端命令行提交作业时，若不指定输出目录，在作业运行界面也可直接查看数据分析结果。

```
Job with JobID xxx has finished.
Job Runtime: xxx ms
Accumulator Results:
- e6209f96ffa423974f8c7043821814e9 (java.util.ArrayList) [31 elements]

(a,3)
(and,2)
(batch,1)
(both,1)
(computing,2)
(data,2)
(demo,1)
(distribution,1)
(engine,1)
(flink,2)
(for,1)
(framework,1)
(is,2)
(it,1)
(mrs,1)
(parallel,1)
(processing,3)
(provides,1)
(stream,2)
(supports,2)
(test,1)
(that,2)
```



```
(this,1)
(unified,1)
```

----结束

1.5 通过 Spark Streaming 作业消费 Kafka 数据

应用场景

本文介绍如何使用MRS集群运行Spark Streaming作业以消费Kafka数据。

假定某个业务Kafka每1秒就会收到1个单词记录。基于业务需要，开发的Spark应用程序实现实时累加计算每个单词的记录总数的功能。

Spark Streaming样例工程的数据存储在Kafka组件中，向Kafka组件发送数据。

方案架构

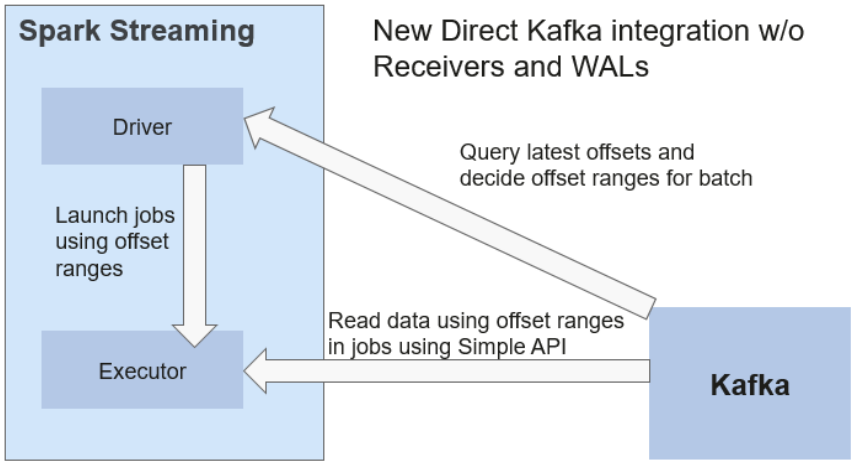
Spark是分布式批处理框架，提供分析挖掘与迭代式内存计算能力，支持多种语言（Scala/Java/Python）的应用开发。 适用以下场景：

- 数据处理（Data Processing）：可以用来快速处理数据，兼具容错性和可扩展性。
- 迭代计算（Iterative Computation）：支持迭代计算，有效应对多步的数据处理逻辑。
- 数据挖掘（Data Mining）：在海量数据基础上进行复杂的挖掘分析，可支持各种数据挖掘和机器学习算法。
- 流式处理（Streaming Processing）：支持秒级延迟的流式处理，可支持多种外部数据源。
- 查询分析（Query Analysis）：支持标准SQL查询分析，同时提供DSL（DataFrame）， 并支持多种外部输入。

Spark Streaming是一种构建在Spark上的实时计算框架，扩展了Spark处理大规模流式数据的能力。当前Spark支持两种数据处理方式：Direct Streaming和Receiver方式。

Direct Streaming方式主要通过采用Direct API对数据进行处理。以Kafka Direct接口为例，与启动一个Receiver来连续不断地从Kafka中接收数据并写入到WAL中相比，Direct API简单地给出每个batch区间需要读取的偏移量位置。然后，每个batch的Job被运行，而对应偏移量的数据在Kafka中已准备好。这些偏移量信息也被可靠地存储在checkpoint文件中，应用失败重启时可以直接读取偏移量信息。

图 1-27 Direct Kafka 接口数据传输

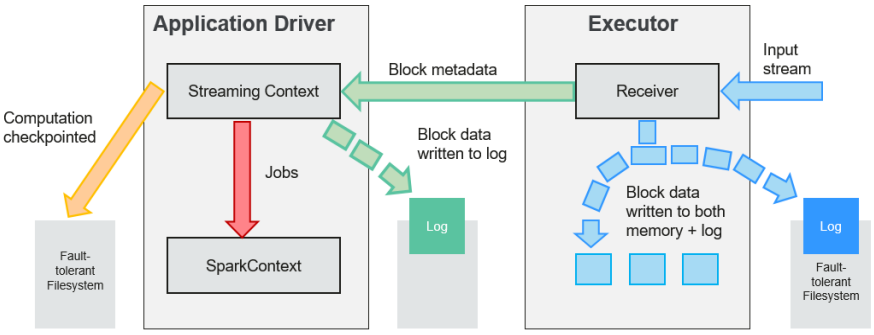


需要注意的是，Spark Streaming可以在失败后重新从Kafka中读取并处理数据段。然而，由于语义仅被处理一次，重新处理的结果和没有失败处理的结果是一致的。

因此，Direct API消除了需要使用WAL和Receivers的情况，且确保每个Kafka记录仅被接收一次，这种接收更加高效。使得Spark Streaming和Kafka可以很好地整合在一起。总体来说，这些特性使得流处理管道拥有高容错性、高效性及易用性，因此推荐使用Direct Streaming方式处理数据。

在一个Spark Streaming应用开始时（也就是Driver开始时），相关的StreamingContext（所有流功能的基础）使用SparkContext启动Receiver成为长驻运行任务。这些Receiver接收并保存流数据到Spark内存中以供处理。用户传送数据的生命周期如图1-28所示：

图 1-28 数据传输生命周期



1. 接收数据（蓝色箭头）

   Receiver将数据流分成一系列小块，存储到Executor内存中。另外，在启用预写日志（Write-ahead Log，简称WAL）以后，数据同时还写入到容错文件系统的预写日志中。

2. 通知Driver（绿色箭头）

   接收块中的元数据（Metadata）被发送到Driver的StreamingContext。这个元数据包括：

   - 定位其在Executor内存中数据位置的块Reference ID。
   - 若启用了WAL，还包括块数据在日志中的偏移信息。

3. 处理数据（红色箭头）

   对每个批次的数据，StreamingContext使用Block信息产生RDD及其Job。StreamingContext通过运行任务处理Executor内存中的Block来执行Job。

4. 周期性地设置检查点（橙色箭头）
5. 为了容错的需要，StreamingContext会周期性地设置检查点，并保存到外部文件系统中。

操作流程

华为云MapReduce服务提供了Spark服务多种场景下的样例开发工程，本章节对应示例场景的开发思路：

1. 接收Kafka中数据，生成相应DStream。
2. 对单词记录进行分类统计。
3. 计算结果，并进行打印。

步骤 1：创建 MRS 集群

步骤1 创建并购买一个包含有Spark2x、Kafka组件的MRS集群，详情请参见**购买自定义集群**。

说明

本文以购买的MRS 3.1.0版本的集群为例，集群未开启Kerberos认证。

步骤2 集群购买成功后，在MRS集群的任一节点内，安装集群客户端，具体操作可参考**安装并使用集群客户端**。

例如客户端安装目录为“/opt/client”。

----结束

步骤 2：准备应用程序

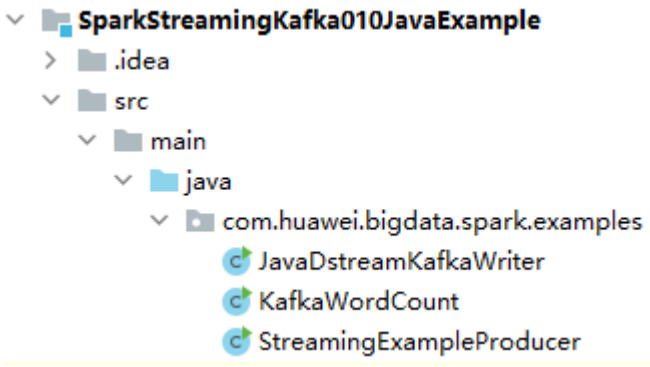
步骤1 通过开源镜像站获取样例工程。

下载样例工程的Maven工程源码和配置文件，并在本地配置好相关开发工具，可参考**通过开源镜像站获取样例工程**。

根据集群版本选择对应的分支，下载并获取MRS相关样例工程。

例如本章节场景对应示例为“SparkStreamingKafka010JavaExample”样例，获取地址：**https://github.com/huaweicloud/huaweicloud-mrs-example/tree/mrs-3.1.0/src/spark-examples/sparknormal-examples/SparkStreamingKafka010JavaExample**。

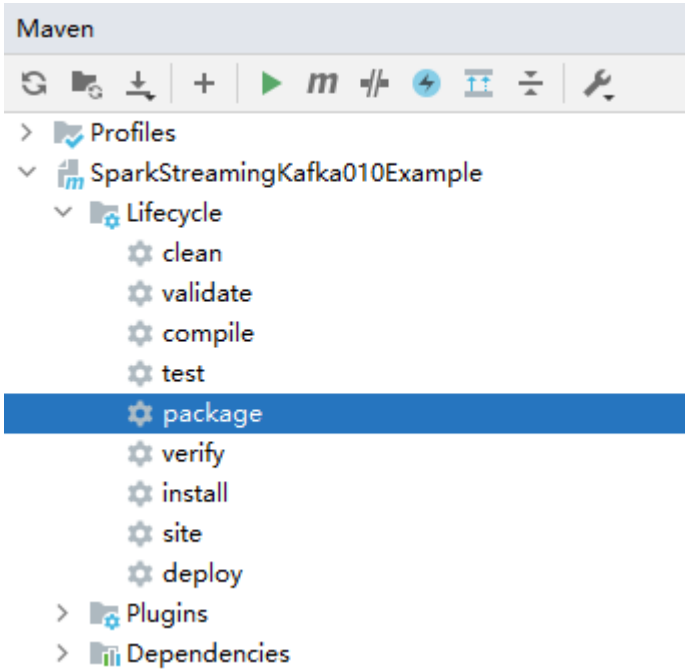
步骤2 本地使用IDEA工具导入样例工程，等待Maven工程下载相关依赖包，具体操作可参考**配置并导入样例工程**。



在本示例工程中，通过使用Streaming调用Kafka接口来获取单词记录，然后把单词记录分类统计，得到每个单词记录数，关键代码片段如下：

```
public class StreamingExampleProducer {
    public static void main(String[] args) throws IOException {
        if (args.length < 2) {
            printUsage();
        }
        String brokerList = args[0];
        String topic = args[1];
        String filePath = "/home/data/";    //源数据获取路径
        Properties props = new Properties();
        props.put(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG, brokerList);
        props.put(ProducerConfig.CLIENT_ID_CONFIG, "DemoProducer");
        props.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG, StringSerializer.class.getName());
        props.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG, StringSerializer.class.getName());
        Producer<String, String> producer = new KafkaProducer<String, String>(props);

        for (int m = 0; m < Integer.MAX_VALUE / 2; m++) {
            File dir = new File(filePath);
            File[] files = dir.listFiles();
            if (files != null) {
                for (File file : files) {
                    if (file.isDirectory()) {
                        System.out.println(file.getName() + "This is a directory!");
                    } else {
                        BufferedReader reader = null;
                        reader = new BufferedReader(new FileReader(filePath + file.getName()));
                        String tempString = null;
                        while ((tempString = reader.readLine()) != null) {
                            // Blank line judgment
                            if (!tempString.isEmpty()) {
                                producer.send(new ProducerRecord<String, String>(topic, tempString));
                            }
                        }
                        // make sure the streams are closed finally.
                        reader.close();
                    }
                }
            }
            try {
                Thread.sleep(3);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }

    private static void printUsage() {
        System.out.println("Usage: {brokerList} {topic}");
    }
}
```

步骤3 本地配置好Maven及SDK相关参数后，样例工程会自动加载相关依赖包。加载完毕后，执行package打包，获取打包后的jar文件。



例如打包后的jar文件为“SparkStreamingKafka010JavaExample-1.0.jar”。

----结束

步骤 3：上传 jar 包及源数据

步骤1 准备向Kafka发送的源数据，例如如下的“input_data.txt”文件，将该文件上传到客户端节点的“/home/data”目录下。

```
ZhangSan
LiSi
WangwWU
Tom
Jemmmy
LinDa
```

步骤2 将编译后的jar包上传到客户端节点，例如上传到“/opt”目录。

说明

如果本地网络无法直接连接客户端节点上传文件，可先将jar文件或者源数据上传至OBS文件系统中，然后通过MRS管理控制台集群内的“文件管理”页面导入HDFS中，再通过HDFS客户端使用**hdfs dfs -get**命令下载到客户端节点本地。

----结束

步骤 4：运行作业并查看结果

步骤1 使用**root**用户登录安装了集群客户端的节点。

cd /opt/client

source bigdata_env

步骤2 创建用于接收数据的Kafka topic。

kafka-topics.sh --create --zookeeper *quorumpeer实例IP地址:ZooKeeper客户端连接端口*/**kafka --replication-factor 2 --partitions 3 --topic** *topic名称*

quorumpeer实例IP地址可登录集群的FusionInsight Manager界面，在“集群 > 服务 > ZooKeeper > 实例”界面中查询，多个地址可用“,”分隔。ZooKeeper客户端连接端口可通过ZooKeeper服务配置参数“clientPort”查询，默认为2181。

例如执行以下命令：

kafka-topics.sh --create --zookeeper 192.168.0.17:2181/kafka --replication-factor 2 --partitions 2 --topic sparkkafka

```
Created topic sparkkafka.
```

步骤3 Topic创建成功后，运行程序向Kafka发送数据。

java -cp /opt/SparkStreamingKafka010JavaExample-1.0.jar:/opt/client/Spark2x/spark/jars/*:/opt/client/Spark2x/spark/jars/streamingClient010/* com.huawei.bigdata.spark.examples.StreamingExampleProducer *Broker实例IP地址:Kafka连接端口 topic名称*

Kafka Broker实例IP地址可登录集群的FusionInsight Manager界面，在“集群 > 服务 > Kafka > 实例”界面中查询，多个地址可用“,”分隔。Broker端口号可通过Kafka服务配置参数“port”查询，默认为9092。

例如执行以下命令：

java -cp /opt/SparkStreamingKafka010JavaExample-1.0.jar:/opt/client/Spark2x/spark/jars/*:/opt/client/Spark2x/spark/jars/streamingClient010/* com.huawei.bigdata.spark.examples.StreamingExampleProducer 192.168.0.131:9092 sparkkafka

```
...
    transactional.id = null
    value.serializer = class org.apache.kafka.common.serialization.StringSerializer

2022-06-08 15:43:42 INFO  AppInfoParser:117 - Kafka version: xxx
2022-06-08 15:43:42 INFO  AppInfoParser:118 - Kafka commitId: xxx
2022-06-08 15:43:42 INFO  AppInfoParser:119 - Kafka startTimeMs: xxx
2022-06-08 15:43:42 INFO  Metadata:259 - [Producer clientId=DemoProducer] Cluster ID: d54RYHthSUishVb6nTHP0A
```

步骤4 重新打开一个客户端连接窗口，执行以下命令，读取Kafka Topic中的数据。

cd /opt/client/Spark2x/spark

source bigdata_env

bin/spark-submit --master yarn --deploy-mode client --jars $(files=($SPARK_HOME/jars/streamingClient010/*.jar); IFS=,; echo "${files[*]}") --class com.huawei.bigdata.spark.examples.KafkaWordCount /opt/SparkStreamingKafka010JavaExample-1.0.jar *<checkpointDir> <brokers> <topic> <batchTime>*

- <checkPointDir>指应用程序结果备份到HDFS的路径，自行指定即可，例如“/tmp”。
- <brokers>指获取元数据的Kafka地址，格式为“*Broker实例IP地址:Kafka连接端口*”。

- <topic>指读取Kafka上的topic名称。
- <batchTime>指Streaming分批的处理间隔，例如设置为“5”。

例如执行以下命令：

cd /opt/client/Spark2x/spark

source bigdata_env

bin/spark-submit --master yarn --deploy-mode client --jars $(files=($SPARK_HOME/jars/streamingClient010/*.jar); IFS=,; echo "${files[*]}") --class com.huawei.bigdata.spark.examples.KafkaWordCount /opt/SparkStreamingKafka010JavaExample-1.0.jar /tmp 192.168.0.131:9092 sparkkafka 5

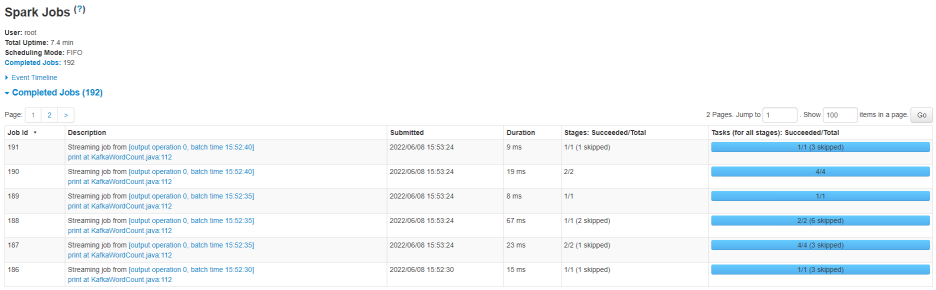
程序运行后，可查看到Kafka中数据的统计结果：

```
....
-------------------------------------------
Time: 1654674380000 ms
-------------------------------------------
(ZhangSan,6)
(Tom,6)
(LinDa,6)
(WangwWU,6)
(LiSi,6)
(Jemmmy,6)

-------------------------------------------
Time: 1654674385000 ms
-------------------------------------------
(ZhangSan,717)
(Tom,717)
(LinDa,717)
(WangwWU,717)
(LiSi,717)
(Jemmmy,717)

-------------------------------------------
Time: 1654674390000 ms
-------------------------------------------
(ZhangSan,2326)
(Tom,2326)
(LinDa,2326)
(WangwWU,2326)
(LiSi,2326)
(Jemmmy,2326)
 ...
```

步骤5 登录FusionInsight Manager界面，单击“集群 > 服务 > Spark2x”。

步骤6 在服务概览页面单击Spark WebUI后的链接地址，可进入History Server页面。

单击待查看的App ID，您可以查看Spark Streaming作业的状态。



----结束

1.6 通过 Flume 采集指定目录日志系统文件至 HDFS

应用场景

Flume是一个分布式、可靠和高可用的海量日志聚合的系统。它能够将不同数据源的海量日志数据进行高效收集、聚合、移动，最后存储到一个中心化数据存储系统中。支持在系统中定制各类数据发送方，用于收集数据。同时，提供对数据进行简单处理，并写到各种数据接受方（可定制）的能力。

Flume分为客户端和服务端，两者都是FlumeAgent。服务端对应着FlumeServer实例，直接部署在集群内部。而客户端部署更灵活，可以部署在集群内部，也可以部署在集群外。它们之间没有必然联系，都可以独立工作，并且提供的功能是一样的。

Flume客户端需要单独安装，支持将数据直接导到集群中的HDFS和Kafka等组件上。

本案例中，通过MRS自定义集群中的Flume组件，自动采集指定节点日志目录下新产生的文件并存储到HDFS文件系统中。

方案架构

Flume-NG由一个个Agent来组成，而每个Agent由Source、Channel、Sink三个模块组成，其中Source负责接收数据，Channel负责数据的传输，Sink则负责数据向下一端的发送。

图 1-29 Flume-NG 架构

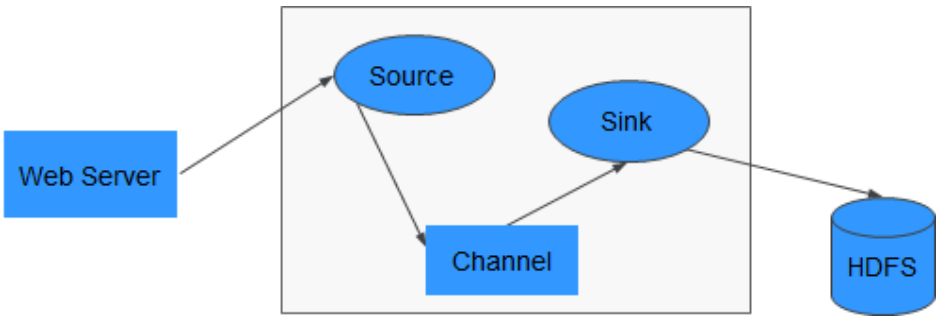
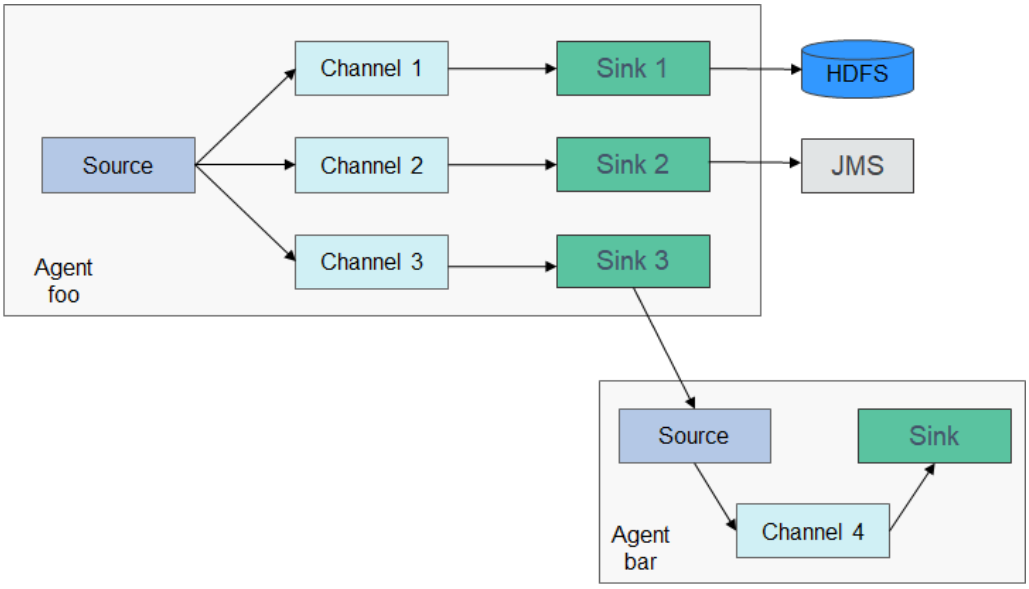


表 1-11 模块说明

名称	说明
Source	Source负责接收数据或通过特殊机制产生数据，并将数据批量放到一个或多个Channel。Source的类型有数据驱动和轮询两种。 典型的Source类型如下： ● 和系统集成的Sources：Syslog、Netcat。 ● 自动生成事件的Sources：Exec、SEQ。 ● 用于Agent和Agent之间通信的IPC Sources：Avro。 Source必须至少和一个Channel关联。
Channel	Channel位于Source和Sink之间，用于缓存来自Source的数据，当Sink成功将数据发送到下一跳的Channel或最终目的地时，数据从Channel移除。 Channel提供的持久化水平与Channel的类型相关，有以下三类： ● Memory Channel：非持久化。 ● File Channel：基于WAL（预写式日志Write-Ahead Logging）的持久化实现。 ● JDBC Channel：基于嵌入Database的持久化实现。 Channel支持事务，可提供较弱的顺序保证，可以和任何数量的Source和Sink工作。
Sink	Sink负责将数据传输到下一跳或最终目的，成功完成后将数据从Channel移除。 典型的Sink类型如下： ● 存储数据到最终目的终端Sink，比如：HDFS、HBase。 ● 自动消耗的Sink，比如：Null Sink。 ● 用于Agent间通信的IPC sink：Avro。 Sink必须作用于一个确切的Channel。

Flume也可以配置成多个Source、Channel、Sink，如**图1-30**所示：

图 1-30 Flume 结构图



步骤 1：创建 MRS 集群

步骤1 创建并购买一个包含有Flume、HDFS组件的MRS集群，详情请参见**购买自定义集群**。

说明

本文以购买的MRS 3.1.0版本的集群为例，集群未开启Kerberos认证。

步骤2 集群购买成功后，登录集群的FusionInsight Manager界面，下载集群客户端并解压。

由于Flume客户端需要单独安装，需要首先下载集群的客户端安装包到待安装Flume客户端的节点上并解压。

1. 在FusionInsight Manager“主页”页签的集群名称后单击 ··· ，单击“下载客户端”下载集群客户端。
2. 在“下载集群客户端”弹窗中填写集群客户端下载信息。

图 1-31 下载集群客户端

- “选择客户端类型”中选择“完整客户端”。
- “选择平台类型”必须与待安装节点的架构匹配，以“x86_64”为例。
- 勾选“仅保存到如下路径”，填写下载路径，本示例以“/tmp/FusionInsight-Client/”为例，需确保**omm**用户对该路径有操作权限。

3. 客户端软件包下载完成后，以**root**用户登录集群的主OMS节点，复制安装包到指定节点。

   客户端软件包默认下载至集群的主OMS节点（可通过FusionInsight Manager的“主机”界面查看带有★标识的节点），如需要在集群内其他节点上安装，执行以下命令将软件包传输至其他节点，否则本步骤可忽略。

   cd /tmp/FusionInsight-Client/

   scp -p FusionInsight_Cluster_1_Services_Client.tar *待安装Flume客户端节点的IP地址:/tmp*

4. 以**root**用户登录待安装Flume客户端的节点，进入客户端软件包所在目录后，执行以下命令解压软件包。

   tar -xvf FusionInsight_Cluster_1_Services_Client.tar

   tar -xvf FusionInsight_Cluster_1_Services_ClientConfig.tar

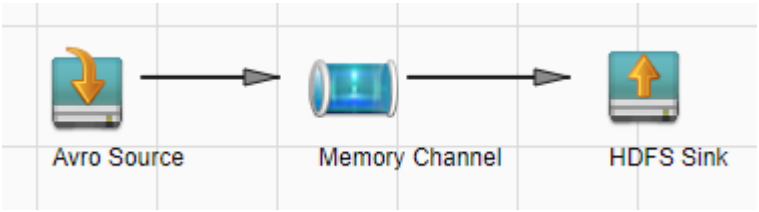
----结束

步骤 2：生成 Flume 配置文件

步骤1 登录FusionInsight Manager，选择“集群 > 服务 > Flume > 配置工具”。

步骤2 配置并导出“properties.properties”文件。

选择“Agent名”为“server”，分别选择“Avro Source”、“Memory Channel”和“HDFS Sink”模块，并连接。

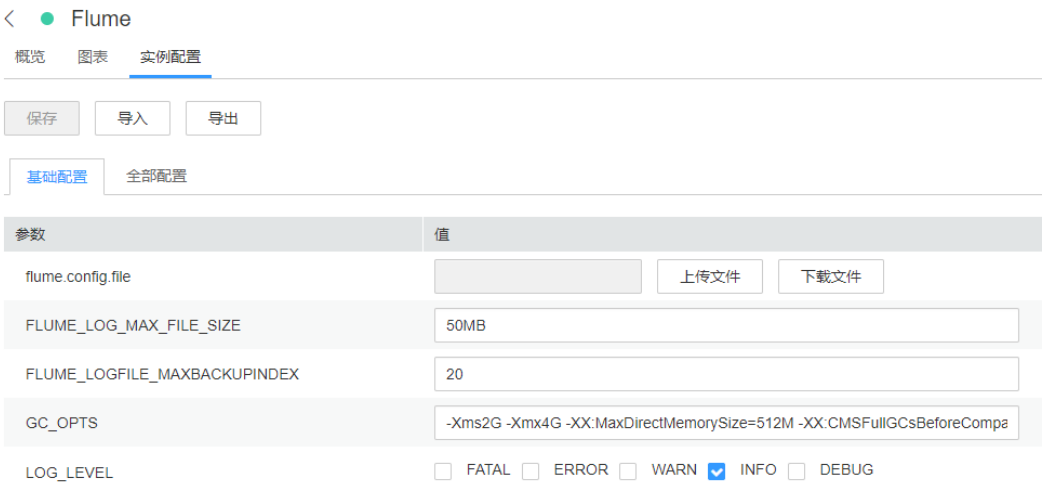


双击模块图标，配置对应参数，各模块配置参数如下（其他参数保持默认）：

类型	配置参数	描述	配置示例
Avro Source	名称	模块名称，可自定义。	test_source_1
	bind	待连接的Flume角色所在节点的IP地址，可在“集群 > 服务 > Flume > 实例”页面查看任一Flume角色实例的IP地址。	192.168.10.192
	port	连接端口，建议从21154开始配置。	21154
Memory Channel	名称	模块名称，可自定义。	test_channel_1
HDFS Sink	名称	模块名称，可自定义。	test_sink_1
	hdfs.path	日志文件写入HDFS的目录。	hdfs://hacluster/flume/test
	hdfs.filePrefix	写入HDFS后的文件名前缀。	over_%{basename}

步骤3 单击“导出”按钮，下载“properties.properties”文件到本地。

步骤4 在FusionInsight Manager界面，单击“集群 > 服务 > Flume > 实例”，单击准备上传配置文件的节点行的“Flume”角色，进入“实例配置”页面。



步骤5 选择“上传文件”，上传“properties.properties”文件。

单击“保存”，单击“确定”后等待配置完成。

步骤6 选择“集群 > 服务 > Flume > 配置工具”

选择“Agent名”为“client”，分别选择“SpoolDir Source”、“Memory Channel”和“Avro Sink”模块，并连接。

双击模块图标，配置对应参数，各模块配置参数如下（其他参数请保持默认）：

类型	配置参数	描述	配置示例
SpoolDir Source	名称	模块名称，可自定义。	test_source_1
	spoolDir	需要采集日志的目录，该目录需要对flume运行用户具有读写权限，并放入文件进行验证。	/var/log/Bigdata/audit/test
Memory Channel	名称	模块名称，可自定义。	test_channel_1
HDFS Sink	名称	模块名称，可自定义。	test_sink_1
	hostname	待连接的Flume角色所在节点的IP地址。	192.168.10.192
	port	连接端口，建议从21154开始配置。	21154

步骤7 单击“导出”按钮，下载“properties.properties”文件到本地。

步骤8 将“properties.properties”文件重命名为“client.properties.properties”，然后将该文件上传并覆盖到Flume客户端节点的“集群*客户端安装包解压路径*/Flume/FlumeClient/flume/conf”目录下。

----结束

步骤 3：安装 Flume 客户端

步骤1 以**root**用户登录待安装Flume客户端所在节点。

步骤2 进入客户端安装包解压路径，例如客户端安装包以上传至“/tmp”目录下并解压。

步骤3 执行以下命令安装Flume客户端，其中“/opt/FlumeClient”为自定义的Flume客户端安装路径。

cd /tmp/FusionInsight-Client/FusionInsight_Cluster_1_Services_ClientConfig/Flume/FlumeClient

./install.sh -d /opt/FlumeClient -c flume/conf/client.properties.properties

```
CST ... [flume-client install]: install flume client successfully.
```

----结束

步骤 4：查看日志采集结果

步骤1 Flume客户端安装成功后，向日志采集目录写入新的日志文件，验证日志是否传输成功。

例如在“/var/log/Bigdata/audit/test”目录下新建几个日志文件。

cd /var/log/Bigdata/audit/test

vi log1.txt

```
Test log file 1!!!
```

vi log2.txt

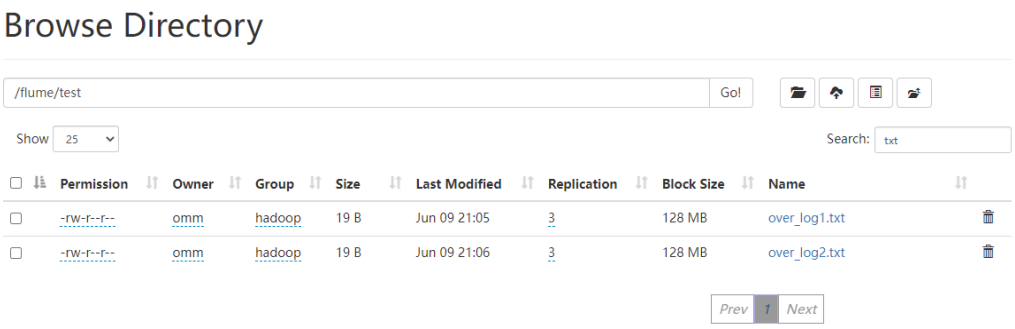
```
Test log file 2!!!
```

步骤2 写入日志文件后，执行**ll**命令查看文件列表，日志文件自动增加了后缀“.COMPLETED”，表示该文件已被采集。

```
-rw-------. 1 root root      75 Jun  9 19:59 log1.txt.COMPLETED
-rw-------. 1 root root      75 Jun  9 19:59 log2.txt.COMPLETED
```

步骤3 登录FusionInsight Manager，选择“集群 > 服务 > HDFS”，单击“NameNode(主)”对应的链接，打开HDFS WebUI界面。

步骤4 选择“Utilities > Browse the file system”，观察HDFS上“/flume/test”目录下是否有产生数据。



如上所示，文件目录下产生了收集的日志文件，文件名统一增加了前缀“over_”。

下载日志文件“over_log1.txt”并查看内容，与写入的日志文件“log1.txt”内容一致。

```
Test log file 1!!!
```

----结束

1.7 基于 Kafka 的 Word Count 数据流统计案例

应用场景

本文介绍如何使用MRS集群运行Kafka程序处理数据。

Kafka Streams是Apache Kafka提供的一个轻量级流式处理框架，其输入输出均存储在Kafka集群中。

接下来，以最常见的WordCount样例Demo进行说明。

方案架构

Kafka是一个分布式的消息发布-订阅系统。它采用独特的设计提供了类似JMS的特性，主要用于处理活跃的流式数据。

Kafka有很多适用的场景：消息队列、行为跟踪、运维数据监控、日志收集、流处理、事件溯源、持久化日志等。

Kafka有如下几个特点：

- 高吞吐量
- 消息持久化到磁盘
- 分布式系统易扩展
- 容错性好

操作流程

华为云MapReduce服务提供了Kafka服务多种场景下的样例开发工程，本章节对应示例场景的开发思路：

1. 使用Kafka客户端创建两个Topic，用于输入Topic和输出Topic。

2. 开发一个Kafka Streams完成单词统计功能，通过读取输入Topic中的消息，统计每条消息中的单词个数，从输出Topic消费数据，将统计结果以Key-Value的形式输出。

步骤 1：创建 MRS 集群

步骤1 创建并购买一个包含有Kafka组件的MRS集群，详情请参见**购买自定义集群**。

说明

本文以购买的MRS 3.1.0版本的集群为例，组件包含Hadoop、Kafka组件，集群未开启Kerberos认证。

步骤2 集群购买成功后，在MRS集群的任一节点内，安装集群客户端，具体操作可参考**安装并使用集群客户端**。

例如客户端安装在主管理节点中，安装目录为“/opt/client”。

步骤3 客户端安装完成后，在客户端内创建“lib”目录，用于放置相关jar包。

将安装客户端过程中解压的目录中Kafka相关jar包复制到“lib”目录。

例如客户端软件包的下载路径为主管理节点的“/tmp/FusionInsight-Client”目录，执行以下命令：

mkdir /opt/client/lib

cd /tmp/FusionInsight-Client/FusionInsight_Cluster_1_Services_ClientConfig

scp Kafka/install_files/kafka/libs/* /opt/client/lib

----结束

步骤 2：准备应用程序

步骤1 通过开源镜像站获取样例工程。

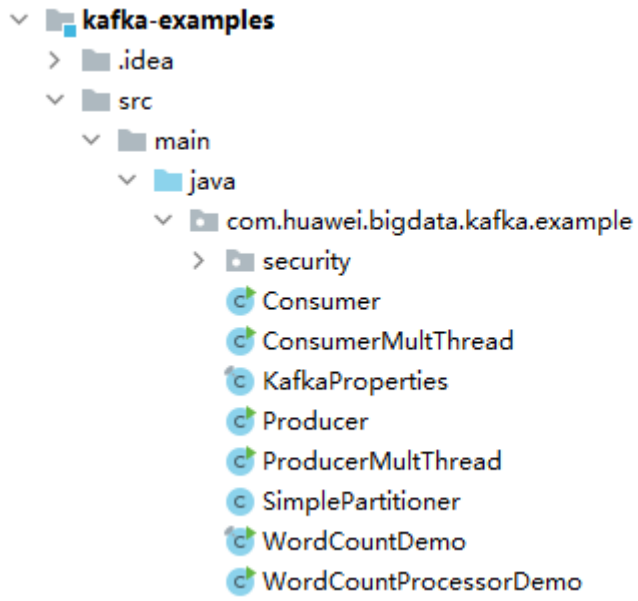
下载样例工程的Maven工程源码和配置文件，并在本地配置好相关开发工具，可参考**通过开源镜像站获取样例工程**。

根据集群版本选择对应的分支，下载并获取MRS相关样例工程。

例如本章节场景对应示例为“WordCountDemo”样例，获取地址：**https://github.com/huaweicloud/huaweicloud-mrs-example/tree/mrs-3.1.0/src/kafka-examples**。

步骤2 本地使用IDEA工具导入样例工程，等待Maven工程下载相关依赖包。

本地配置好Maven及SDK相关参数后，样例工程会自动加载相关依赖包，具体操作可参考**配置并导入样例工程**。



在示例程序“WordCountDemo”中，通过调用Kafka接口来获取单词记录，然后把单词记录分类统计，得到每个单词记录数，关键代码片段如下：

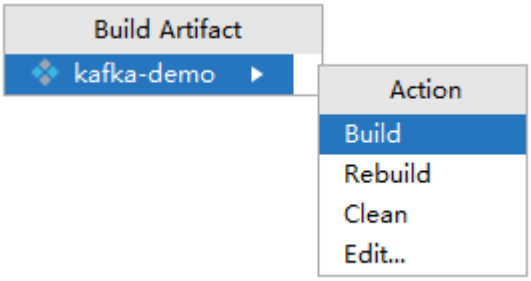
```
...
    static Properties getStreamsConfig() {
        final Properties props = new Properties();
        KafkaProperties kafkaProc = KafkaProperties.getInstance();
        // Broker地址列表，根据集群实际情况配置
        props.put(BOOTSTRAP_SERVERS, kafkaProc.getValues(BOOTSTRAP_SERVERS, "node-group-1kLFk.mrs-rbmq.com:9092"));
        props.put(SASL_KERBEROS_SERVICE_NAME, "kafka");
        props.put(KERBEROS_DOMAIN_NAME, kafkaProc.getValues(KERBEROS_DOMAIN_NAME, "hadoop.hadoop.com"));
        props.put(APPLICATION_ID, kafkaProc.getValues(APPLICATION_ID, "streams-wordcount"));
        // 协议类型：当前支持配置为SASL_PLAINTEXT或者PLAINTEXT
        props.put(SECURITY_PROTOCOL, kafkaProc.getValues(SECURITY_PROTOCOL, "PLAINTEXT"));
        props.put(CACHE_MAX_BYTES_BUFFERING, 0);
        props.put(DEFAULT_KEY_SERDE, Serdes.String().getClass().getName());
        props.put(DEFAULT_VALUE_SERDE, Serdes.String().getClass().getName());
        props.put(ConsumerConfig.AUTO_OFFSET_RESET_CONFIG, "earliest");
        return props;
    }
    static void createWordCountStream(final StreamsBuilder builder) {
        // 从 input-topic 接收输入记录
        final KStream<String, String> source = builder.stream(INPUT_TOPIC_NAME);
        // 聚合 key-value 键值对的计算结果
        final KTable<String, Long> counts = source
                .flatMapValues(value -> Arrays.asList(value.toLowerCase(Locale.getDefault()).split(REGEX_STRING)))
                .groupBy((key, value) -> value)
                .count();
        // 将计算结果的 key-value 键值对从 output topic 输出
        counts.toStream().to(OUTPUT_TOPIC_NAME, Produced.with(Serdes.String(), Serdes.Long()));
    }
...
```

说明

- BOOTSTRAP_SERVERS需根据集群实际情况，配置为Kafka Broker节点的主机名及端口，可通过集群FusionInsight Manager界面中单击“集群 > 服务 > Kafka > 实例”查看Broker**在Linux中调测程序**实例信息。
- SECURITY_PROTOCOL为连接Kafka的协议类型，在本示例中，配置为“PLAINTEXT”。

步骤3 确认“WordCountDemo.java”内的参数无误后，将工程编译后进行打包，获取打包后的jar文件。

编译jar包详细操作可参考**在Linux中调测程序**。



例如打包后的jar文件为“kafka-demo.jar”。

----结束

步骤 3：上传 jar 包及源数据

步骤1 将编译后的jar包上传到客户端节点，例如上传到“/opt/client/lib”目录下。

说明

如果本地网络无法直接连接客户端节点上传文件，可先将jar文件或者源数据上传至OBS文件系统中，然后通过MRS管理控制台集群内的“文件管理”页面导入HDFS中，再通过HDFS客户端使用**hdfs dfs -get**命令下载到客户端节点本地。

----结束

步骤 4：运行作业并查看结果

步骤1 使用**root**用户登录安装了集群客户端的节点。

cd /opt/client

source bigdata_env

步骤2 创建输入Topic和输出Topic，与样例代码中指定的Topic名称保持一致，输出Topic的清理策略设置为compact。

kafka-topics.sh --create --zookeeper *quorumpeer实例IP地址:ZooKeeper客户端连接端口*/**kafka --replication-factor 1 --partitions 1 --topic** *Topic名称*

quorumpeer实例IP地址可登录集群的FusionInsight Manager界面，在“集群 > 服务 > ZooKeeper > 实例”界面中查询，多个地址可用“,”分隔。ZooKeeper客户端连接端口可通过ZooKeeper服务配置参数“clientPort”查询，默认为2181。

例如执行以下命令：

kafka-topics.sh --create --zookeeper 192.168.0.17:2181/kafka --replication-factor 1 --partitions 1 --topic streams-wordcount-input

kafka-topics.sh --create --zookeeper 192.168.0.17:2181/kafka --replication-factor 1 --partitions 1 --topic streams-wordcount-output --config cleanup.policy=compact

步骤3 Topic创建成功后，执行以下命令运行程序。

java -cp .:/opt/client/lib/* com.huawei.bigdata.kafka.example.WordCountDemo

步骤4 重新打开一个客户端连接窗口，执行以下命令，使用“kafka-console-producer.sh”向输入Topic中写入消息：

cd /opt/client

source bigdata_env

kafka-console-producer.sh --broker-list *Broker实例IP地址:Kafka连接端口（例如192.168.0.13:9092）* **--topic streams-wordcount-input --producer.config /opt/client/Kafka/kafka/config/producer.properties**

步骤5 重新打开一个客户端连接窗口，执行以下命令，使用“kafka-console-consumer.sh”从输出Topic消费数据，查看统计结果。

cd /opt/client

source bigdata_env

kafka-console-consumer.sh --topic streams-wordcount-output --bootstrap-server *Broker实例IP地址:Kafka连接端口* **--consumer.config /opt/client/Kafka/kafka/config/consumer.properties --from-beginning --property print.key=true --property print.value=true --property key.deserializer=org.apache.kafka.common.serialization.StringDeserializer --property value.deserializer=org.apache.kafka.common.serialization.LongDeserializer --formatter kafka.tools.DefaultMessageFormatter**

向输入Topic中写入消息：

```
>This is Kafka Streams test
>test starting
>now Kafka Streams is running
>test end
```

消息输出：

```
this    1
is      1
kafka   1
streams 1
test    1
test    2
starting 1
now     1
kafka   2
streams 2
is      2
running 1
test    3
end     1
```

----结束

2 数据迁移

2.1 数据迁移方案介绍

2.1.1 准备工作

本迁移指导将指导适用于多种不同场景下的HDFS、HBase、Hive数据向MRS集群的迁移工作。由于数据迁移过程中可能存在数据覆盖、丢失、损坏等风险，因此本指导只作为参考，具体的数据迁移方案的制定及实施需要华为云支持人员协同完成。

数据迁移前源集群的准备工作，目的是防止在数据迁移过程中源集群产生新数据，导致源集群与迁移后的目标集群数据不一致。在数据迁移完成之前，目标集群应处于初始状态，期间不能运行除数据迁移作业外的其它任何业务。

云数据迁移（Cloud Data Migration, 简称CDM），是一种高效、易用的数据集成服务。CDM围绕大数据迁移上云和智能数据湖解决方案，提供了简单易用的迁移能力和多种数据源到数据湖的集成能力，降低了客户数据源迁移和集成的复杂性，有效的提高数据迁移和集成的效率，可参考**Hadoop数据迁移到华为云MRS服务**、**HBase数据迁移到华为云MRS服务**相关内容。

停止集群业务及相关服务

- 如果您的集群涉及到Kafka业务，请先停止所有向Kafka中生产数据的作业，等待Kafka的消费作业消费完Kafka中的存量数据后，再执行下一步操作。
- 停止所有与HDFS、HBase、Hive相关的业务和作业，然后停止HBase、Hive服务。

打通数据传输通道

- 当源集群与目标集群部署在同一区域的不同VPC时，请创建两个VPC之间的网络连接，打通网络层面的数据传输通道。请参见**VPC对等连接**。
- 当源集群与目标集群部署在同一VPC但属于不同安全组时，在VPC管理控制台，为每个安全组分别添加安全组规则。规则的“协议”为“ANY”，“方向”为“入方向”，“源地址”为“安全组”且是对端集群的安全组。
  - 为源集群的安全组添加入方向规则，源地址选择目标集群的安全组。
  - 为目标集群的安全组添加入方向规则，源地址选择源集群的安全组。

- 当源集群与目标集群部署在同一VPC同一安全组且两个集群都开启了Kerberos认证，需要为两个集群配置互信，具体请参考**配置跨Manager集群互信**。

2.1.2 元数据导出

为了保持迁移后数据的属性及权限等信息在目标集群上与源集群一致，需要将源集群的元数据信息导出，以便在完成数据迁移后进行必要的元数据恢复。

需要导出的元数据包括HDFS文件属主/组及权限信息、Hive表描述信息。

HDFS 元数据导出

HDFS数据需要导出的元数据信息包括文件及文件夹的权限和属主/组信息，可通过如下HDFS客户端命令导出。

```
$HADOOP_HOME/bin/hdfs dfs -ls -R <migrating_path> > /tmp/hdfs_meta.txt
```

其中，各参数的含义如下：

- *$HADOOP_HOME*：源集群Hadoop客户端安装目录。
- *<migrating_path>*：HDFS上待迁移的数据目录。
- /tmp/hdfs_meta.txt：导出的元数据信息保存在本地的路径。

说明

如果源集群与目标集群网络互通，且以超级管理员身份运行hadoop distcp命令进行数据拷贝，可以添加参数“-p”让distcp在拷贝数据的同时在目标集群上分别恢复相应文件的元数据信息，因此在这种场景下可直接跳过本步骤。

Hive 元数据导出

Hive表数据存储在HDFS上，表数据及表数据的元数据由HDFS统一按数据目录进行迁移。而Hive表的元数据根据集群的不同配置，可以存储在不同类型的关系型数据库中（如MySQL、PostgreSQL、Oracle等）。

本指导中导出的Hive表元数据即存储在关系型数据库中的Hive表的描述信息。

业界主流大数据发行版均支持Sqoop的安装，如果是自建的社区版大数据集群，可下载社区版Sqoop进行安装。借助Sqoop来解耦导出的元数据与关系型数据库的强依赖，将Hive元数据导出到HDFS上，与表数据一同迁移后进行恢复。

参考步骤如下：

步骤1 在源集群上下载并安装Sqoop工具。

请参见**http://sqoop.apache.org/**。

步骤2 下载相应关系型数据库的JDBC驱动放置到“${Sqoop_Home}/lib”目录。

步骤3 执行如下命令导出所有Hive元数据表。

例如所有导出数据保存在HDFS上的“/user/<*user_name*>/<*table_name*>”目录。

```
$Sqoop_Home/bin/sqoop import --connect jdbc:<driver_type>://<ip>:<port>/<database> --table
<table_name> --username <user> -password <passwd> -m 1
```

其中，各参数的含义如下：

- *$Sqoop_Home*：Sqoop的安装目录。

- *<driver_type>*：数据库类型。
- *<ip>*：源集群数据库的IP地址。
- *<port>*：源集群数据库的端口号。
- *<table_name>*：待导出的表名称。
- *<user>*：用户名。
- *<passwd>*：用户密码。

说明

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。

----结束

2.1.3 数据拷贝

根据源集群与目标集群分别所处的区域及网络连通性，可分为以下几种数据拷贝场景。

同 Region

当源集群与目标集群处于同一Region时，根据**打通数据传输通道**进行网络配置，打通网络传输通道。使用Distcp工具执行如下命令将源集群的HDFS、HBase、Hive数据文件以及Hive元数据备份文件拷贝至目的集群。

```
$HADOOP_HOME/bin/hadoop distcp <src> <dist> -p
```

其中，各参数的含义如下：

- *$HADOOP_HOME*：目的集群Hadoop客户端安装目录
- <src>：源集群HDFS目录
- <dist>：目的集群HDFS目录

不同 Region

当源集群与目标集群处于不同Region时，用Distcp工具将源集群数据拷贝到OBS，借助OBS跨区域复制功能（请参见**跨区域复制**）将数据复制到对应目的集群所在Region的OBS，然后通过Distcp工具将OBS数据拷贝到目的集群的HDFS上。由于执行Distcp无法为OBS上的文件设置权限、属主/组等信息，因此当前场景在进行数据导出时也需要将HDFS的元数据信息进行导出并拷贝，以防HDFS文件属性信息丢失。

线下集群向云迁移

线下集群可以通过如下两种方式将数据迁移至云：

- 云专线（DC）

  为源集群与目标集群之间建立**云专线**，打通线下集群出口网关与线上VPC之间的网络，然后参考**同Region**执行Distcp进行拷贝。

- 数据快递服务（DES）

  对于TB或PB级数据上云的场景，华为云提供**数据快递服务 DES**。将线下集群数据及已导出的元数据拷贝到DES盒子，快递服务将数据递送到华为云机房，然后通过**云数据迁移 CDM**将DES盒子数据拷贝到HDFS。

2.1.4 数据恢复

HDFS 文件属性恢复

根据导出的权限信息在目的集群的后台使用HDFS命令对文件的权限及属主/组信息进行恢复。

```
$HADOOP_HOME/bin/hdfs dfs -chmod <MODE> <path>
$HADOOP_HOME/bin/hdfs dfs -chown <OWNER> <path>
```

Hive 元数据恢复

在目的集群中安装并使用Sqoop命令将导出的Hive元数据导入MRS集群DBService。

```
$Sqoop_Home/bin/sqoop export --connect jdbc:postgresql://<ip>:20051/hivemeta --table <table_name> --username hive -password <passwd> --export-dir <export_from>
```

其中，各参数的含义如下：

- *$Sqoop_Home*：目的集群上Sqoop的安装目录。
- <*ip*>：目的集群上数据库的IP地址。
- <*table_name*>：待恢复的表名称。
- <*passwd*>：hive用户的密码。
- <*export_from*>：元数据在目的集群的HDFS地址。

说明

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。

HBase 表重建

重启目的集群的HBase服务，使数据迁移生效。在启动过程中，HBase会加载当前HDFS上的数据并重新生成元数据。启动完成后，在Master节点客户端执行如下命令加载HBase表数据。

```
$HBase_Home/bin/hbase hbck -fixMeta -fixAssignments
```

命令执行完成后，重复执行如下命令查看HBase集群健康状态直至正常。

```
hbase hbck
```

2.2 数据迁移到 MRS 前信息收集

由于离线大数据搬迁有一定的灵活性，迁移前需要掌握现有集群的详细信息，以能够更好的进行迁移决策。

业务信息调研

1. 大数据平台及业务的架构图。
2. 大数据平台和业务的数据流图（包括峰值和均值流量等）。

   识别平台数据接入源、大数据平台数据流入方式（实时数据上报、批量数据抽取）、分析平台数据流向。

   数据在平台内各个组件间的流向，比如使用什么组件采集数据，采集完数据后数据如何流向下一层组件，使用什么组件存储数据，数据处理过程中的工作流等。

3. 业务作业类型Hive SQL、Spark SQL、Spark Python等，是否需要使用MRS的第三方包，参考**MRS应用开发样例**。
4. 调度系统，需要考虑调度系统对接MRS集群。
5. 迁移后，业务割接允许中断时长，识别平台业务优先级。

   识别在迁移过程中不能中断的业务、可短时中断的业务、整体业务迁移可接受的迁移时长，梳理业务迁移顺序。
6. 客户端部署要求。
7. 业务执行时间段和高峰时间段。
8. 大数据集群的数量和大数据集群功能划分，分析平台业务模型。

   各个集群或各个组件分别负责什么业务，处理什么类型的数据。比如实时/离线数据分别使用什么组件处理、数据格式类型、压缩算法等。

集群基本信息收集

表 2-1 集群基本信息

参数	说明
集群名称	-
集群版本	MRS、CDM等集群的版本信息。
节点数及规格	调研现有集群节点数和节点规格。 如果集群硬件异构，请收集多种规格和对应节点数，参见**表2-2**。 例如： • 2台32U64G机器部署NameNode + ResourceManager • 2台32U64G机器部署HiveServer • 20台16U32G机器部署DataNode和NodeManager
是否开启Kerberos认证	是或否
权限控制及说明	调研各个开启ACL权限控制的组件和配置，通常涉及Yarn、Hive、Impala、HBase等组件。 使用Ranger、Sentry或组件开源的权限能力进行权限控制。
所在Region/AZ	云上资源填写项
虚拟私有云	云上资源填写项
子网	云上资源填写项
安全组	云上资源填写项

表 2-2 硬件信息调研表

节点组	CPU和内存信息	磁盘和网络（按节点组统计）		HDFS			Yarn	
-	-	磁盘信息（数据盘大小、磁盘IO、当前磁盘使用率和IO情况）	网络（网卡带宽大小、网络读写速度和峰值）	NameNode	DataNode	JournalNode	NodeManager	ResourceManager
master1	(16U64G)	-	-	1	-	1	-	1
master2	(16U64G)	-	-	1	-	1	-	1
master3	(16U64G)	-	-	-	-	1	-	-
Core-group1	(32U128G)*数量	-	-	-	1	-	1	-
Core-group1	(32U129G)	-	-	-	-	-	-	-
Core-group1	(32U130G)	-	-	-	-	-	-	-

大数据组件信息

使用的大数据组件信息和规划的新版本大数据集群版本信息比较，主要识别版本差异可能对迁移过程的影响，以及对迁移后业务兼容性的影响。

表 2-3 大数据组件信息

大数据组件	源端集群版本	目的端集群版本（以MRS集群版本为准）	说明
HDFS/OBS（或其他文件存储系统）	*Hadoop 2.8.3*	*Hadoop 3.3.1*	-
Hive	*1.2.1*	*2.3.3*	存储元数据的数据库：MySQL

大数据组件	源端集群版本	目的端集群版本（以MRS集群版本为准）	说明
HBase	*1.3.1*	*1.3.1*	-
Spark	*2.2.2*	*3.1.1*	-
Kafka	*1.1.0*	*2.11-2.4.0*	-
Oozie	*2.x*	*5.1.0*	-
MySQL	*5.7.1*	*RDS*	-
Flink	*1.7*	*1.15*	-
...	...	...	-

待迁移的存量数据及数据量统计

如果使用HDFS作为文件存储系统，可以通过客户端执行**hadoop fs -du -h** *HDFS文件目录*命令统计路径下的文件大小。

表 2-4 现有数据量统计

大数据组件	待迁移数据的路径	数据量大小	文件个数或表个数
HDFS/OBS（或其他文件存储系统）	*/user/helloworld*	*XXX*	总共：*XXX*个文件 小于2 MB的文件数量：*XXX*个
Hive	*/user/hive/warehouse/*	*XXX*	表个数：*XXX*
HBase	*/hbase*	*XXX*	表个数：*XXX* Region个数：*XXX*

每天新增数据量统计

每天新增数据量主要评估数据增长速度（可以按天/小时等周期维度）。在第一次全量迁移数据后，后续可以定期搬迁老集群新增数据，直到业务完成最终割接。

表 2-5 新增数据量统计

大数据组件	待迁移的数据路径	新增数据量大小
HDFS/OBS（或其他文件存储系统）	*/user/helloworld*	*XXX*
Hive	*/user/hive/warehouse/*	*XXX*

大数据组件	待迁移的数据路径	新增数据量大小
HBase	*/hbase*	*XXX*

网络出口带宽能力

- 迁移数据可以使用的最大网络带宽和专线带宽（是否可调）。
- 迁移数据作业每天可以运行的时间段。

流式 Kafka 集群信息收集

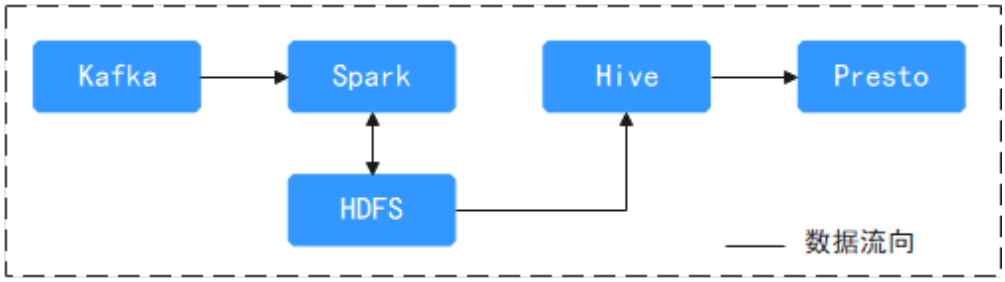
表 2-6 流式 Kafka 集群信息

收集信息项	描述
Kafka的Topic数量和名称	-
Kafka的本地数据暂存时间	如果每个Topic配置不一样，按Topic粒度收集。
每个Topic的副本数和Partition数量	默认为2，副本数越多数据越可靠，也会消耗磁盘空间。 如果每个Topic配置不一样，按Topic粒度收集。
Kafka生产和消费的流量大小	细化到Topic级别。
Kafka客户端ACK配置Acks	-

数据迁移模型样例

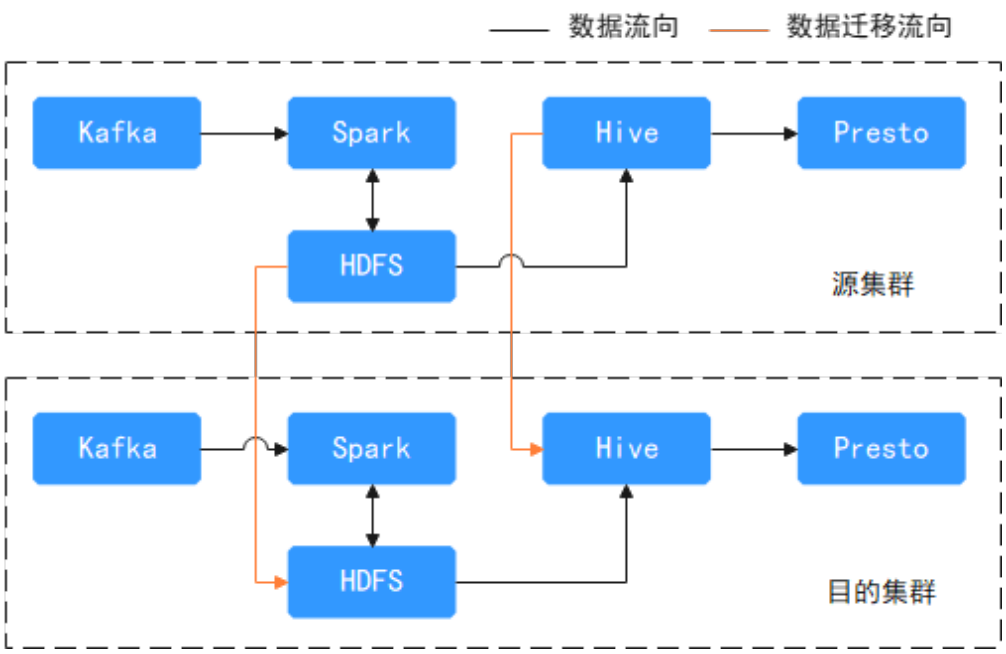
- 一个离线分析平台的客户业务系统，由Spark Streaming消费Kafka数据存入HDFS，HDFS上进行小文件合并后加载到Hive表中，运营人员可以通过Presto进行Hive数据查询。

  图 2-1 源集群业务图

- 针对大数据离线平台包括HDFS和Hive数据需要迁移，Kafka、Spark Streaming、HDFS、Hive、Presto的业务程序要在目的端集群上部署。

图 2-2 迁移示意图

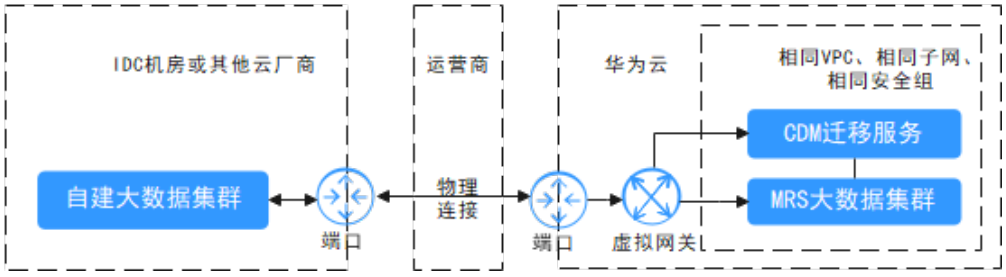


2.3 数据迁移到 MRS 前网络准备

进行大数据迁移时，需要保证源端集群和目的端集群之间的网络互通，例如使用**hadoop distcp**命令跨集群复制数据时需要所有DataNode节点网络互通。根据不同的迁移场景需要使用不通的方式先打通两套集群之间网络连接。

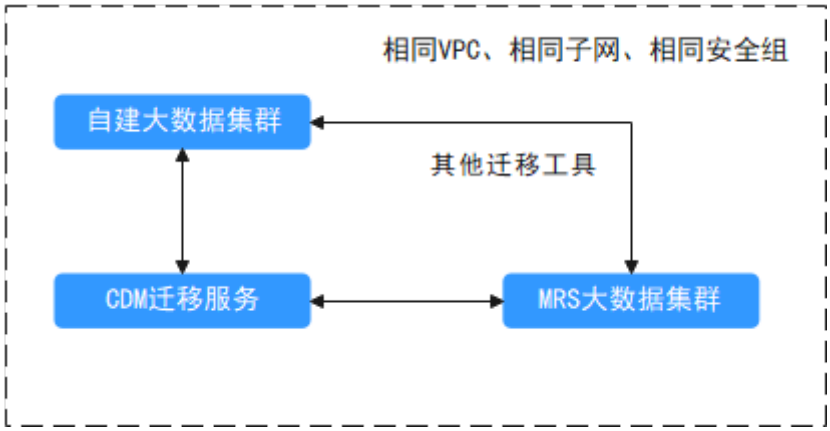
- 客户线下数据中心迁移数据到华为云MRS集群，通过云专线服务为用户搭建本地数据中心与云上VPC之间的专属连接通道。可以使用华为云的**云专线**服务或使用第三方的云专线服务来连通华为云网络。

图 2-3 线下数据中心迁移



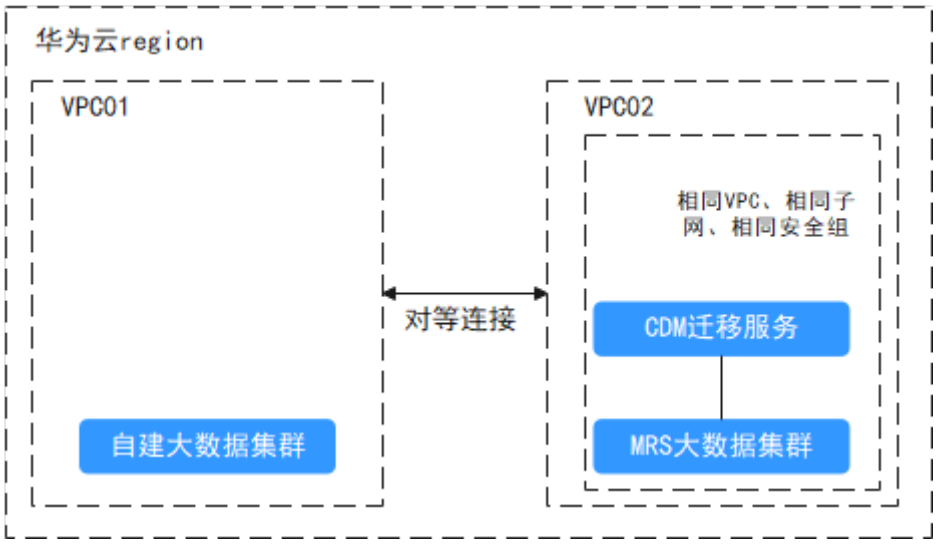
- 客户在华为云上自建大数据集群（或老版本的MRS集群）需要迁移到华为云MRS集群，且在同一个Region区域和VPC子网，可以使自建集群和MRS集群使用相同安全组、VPC、子网网络，从而保证网络连通。

图 2-4 线上同 Region 同 VPC 迁移



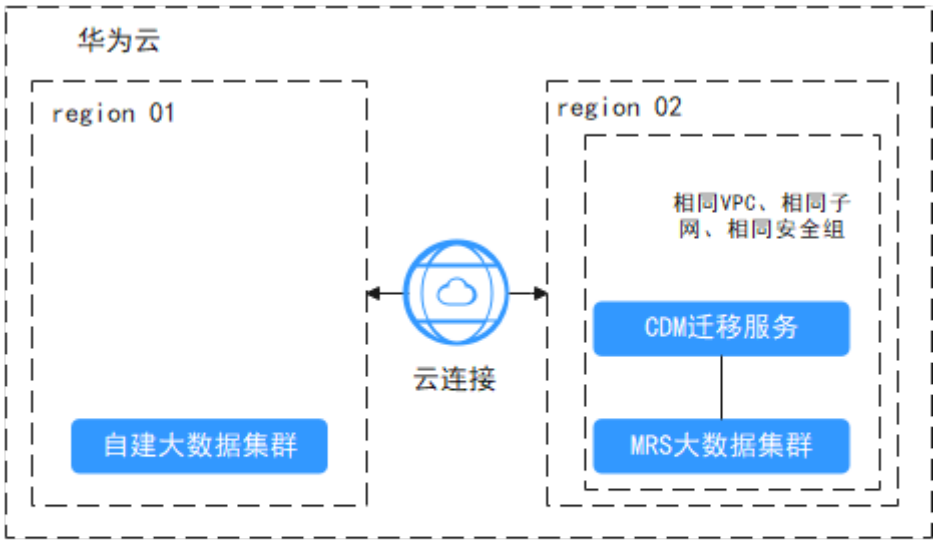
- 客户在华为云上自建大数据集群（或老版本的MRS集群）需要迁移到华为云MRS集群，且在同一个Region区域，但是使用不同VPC子网。需要使用**VPC对等连接**方式配置网络连通。

图 2-5 线上同 Region 不同 VPC 迁移



- 客户在华为云上自建大数据集群（或老版本的MRS集群）需要迁移到华为云MRS集权，但在不同Region区域，可以通过使用**云连接**构建跨区域VPC的网络连接。

图 2-6 线上不同 Region 迁移



2.4 数据迁移常见端口要求

HDFS 组件端口

表 2-7 HDFS 组件端口

配置参数	默认端口（Hadoop 2.x和Hadoop 3.x版本）	端口说明
dfs.namenode.rpc.port	9820	迁移过程中，需要访问NameNode获取文件列表。
dfs.datanode.port	25009	迁移过程中，需要访问DataNode读取具体文件数据。

ZooKeeper 组件端口

表 2-8 ZooKeeper 组件端口

配置参数	默认端口	端口说明
clientPort	2181	ZooKeeper客户端连接ZooKeeper服务器。

Kerberos 组件端口

表 2-9 Kerberos 组件端口

配置参数	默认端口	端口说明
kdc_ports	21732	Kerberos服务认证，非Kerberos集群不涉及。

Hive 组件端口

表 2-10 Hive 组件端口

配置参数	默认端口（Hive 2.x和Hive3.x版本）	端口说明
hive.metastore.port	9083	MetaStore提供Thrift服务的端口。迁移过程中，需要访问该端口查询表元数据信息。

HBase 组件端口

表 2-11 HBase 组件端口

配置参数	默认端口（HBase1.x和HBase 2.x版本）	端口说明
hbase.master.port	16000	HMaster RPC端口。该端口用于HBase客户端连接到HMaster。
hbase.regionserver.port	16020	RS (RegoinServer) RPC端口。该端口用于HBase客户端连接到RegionServer。

Manager 组件端口

表 2-12 Manager 组件端口

配置参数	默认端口	端口说明
N/A	28443	FusionInsight/MRS Manager页面端口。 CDM迁移时候访问该地址获取集群配置。
N/A	20009	FusionInsight/MRS Manager CAS协议端口，用于登录认证。

2.5 Hadoop 数据迁移到华为云 MRS 服务

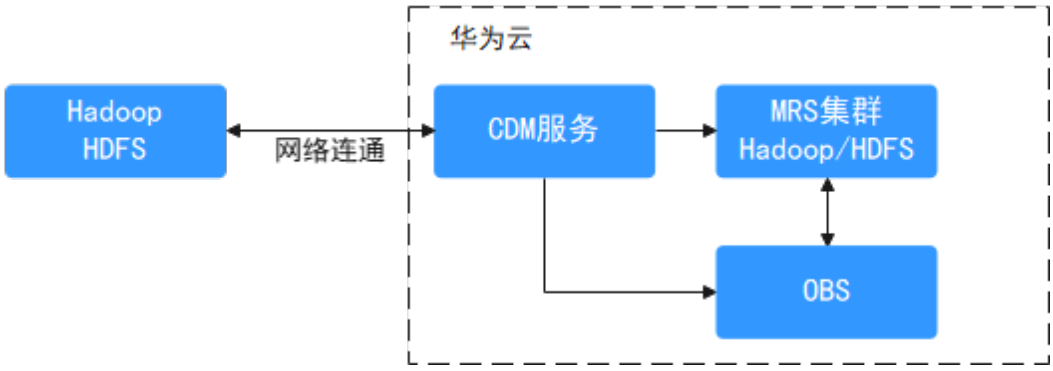
场景介绍

本章节适用于将线下IDC机房或者公有云Hadoop集群中的数据（支持数据量在几十TB级别或以下的数据量级）迁移到华为云MRS服务。

本章节以通过**华为云CDM服务** 2.9.1.200版本进行数据迁移为例介绍。不同版本操作可能有差异，具体操作详情以实际版本对应的操作指导为准。

CDM服务支持迁移的数据源可参考支持的数据源，数据源为Apache HDFS时，建议使用的版本为2.8.X、3.1.X，请执行搬迁前务必确认是否支持搬迁。

图 2-7 Hadoop 数据迁移示意



方案优势

- 简单易用：免编程，向导式任务开发界面，通过简单配置几分钟即可完成迁移任务开发。
- 迁移效率高： 基于分布式计算框架进行数据任务执行和数据传输优化，并针对特定数据源写入做了专项优化，迁移效率高。
- 实时监控：迁移过程中可以执行自动实时监控、告警和通知操作。

对系统影响

搬迁数据量较大时，对网络通信要求较高，执行搬迁任务时，可能会影响其他业务，建议在业务空闲期进行数据迁移任务。

操作步骤

步骤1 登录CDM管理控制台。

步骤2 **创建CDM集群**，该CDM集群的安全组、虚拟私有云、子网需要和迁移目的端集群保持一致，保证CDM集群和MRS集群之间网络互通。

步骤3 在“集群管理”页面单击待操作集群对应“操作”列的“作业管理”。

步骤4 在“连接管理”页签，单击“新建连接”。

步骤5 参考**CDM服务的新建连接**页面，分别添加到迁移源端集群和迁移目的端集群的两个HDFS连接。

连接类型根据实际集群来选择，如果是MRS集群，连接器类型可以选择“MRS HDFS”，如果是自建集群可以选择“Apache HDFS”。

图 2-8 HDFS 连接

步骤6 在“表/文件迁移” 页签，单击“新建作业”。

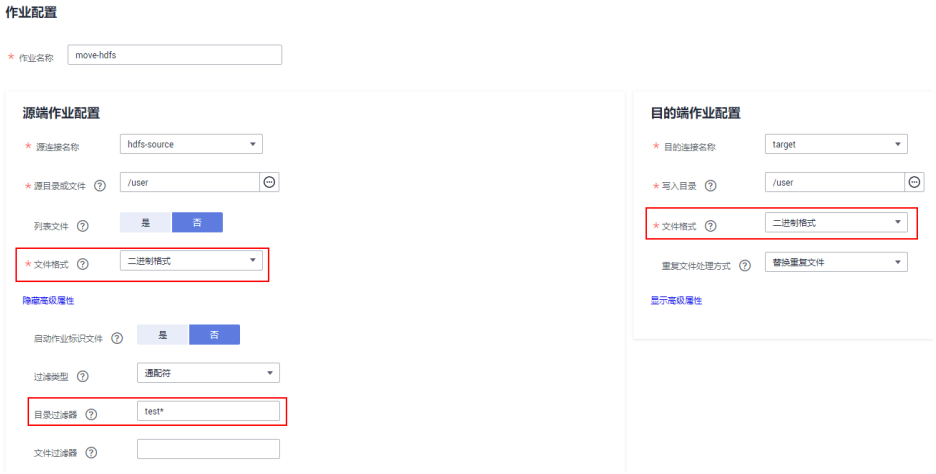
步骤7 选择源连接、目的连接：

- 作业名称：用户自定义任务名称，名称由英文字母、下划线或者数字组成，长度必须在1到256个字符之间。
- 源连接名称：选择迁移源端集群的HDFS连接，作业运行时将从此端复制导出数据。
- 目的连接名称：选择迁移目的端集群的HDFS连接，作业运行时会将数据导入此端。

步骤8 请参见**配置HDFS源端参数**配置源端连接的作业参数，需要迁移的文件夹可通过“目录过滤器”和“文件过滤器”参数设置符合规则的目录和文件进行迁移。

例如迁移匹配“/user/test*”文件夹下文件，该场景下“文件格式”固定为“二进制格式”。

图 2-9 配置作业参数



步骤9 请参见**配置HDFS目的端参数**配置目的端连接的作业参数。

步骤10 单击“下一步”进入任务配置页面。

- 如需定期将新增数据迁移至目的端集群，可在该页面进行配置，也可在任务执行后再参考**步骤14**配置定时任务。
- 如无新增数据需要后续定期迁移，则跳过该页面配置直接单击“保存”回到作业管理界面。

图 2-10 任务配置

步骤11 选择“作业管理”的“表/文件迁移”页签，在待运行作业的“操作”列单击“运行”，即可开始HDFS文件数据迁移，并等待作业运行完成。

步骤12 登录迁移目的端集群主管理节点。

步骤13 在集群客户端内执行**hdfs dfs -ls -h /user/**命令查看迁移目的端集群中已迁移的文件。

步骤14 （可选）如果源端集群中有新增数据需要定期将新增数据迁移至目的端集群，则配置定期任务增量迁移数据，直到所有业务迁移至目的端集群。

1. 在CDM集群中选择“作业管理”的“表/文件迁移”页签。
2. 在迁移作业的“操作”列选择“更多 > 配置定时任务”。
3. 开启定时执行功能，根据具体业务需求设置重复周期，并设置有效期的结束时间为所有业务割接到新集群之后的时间。

图 2-11 配置定时任务

----结束

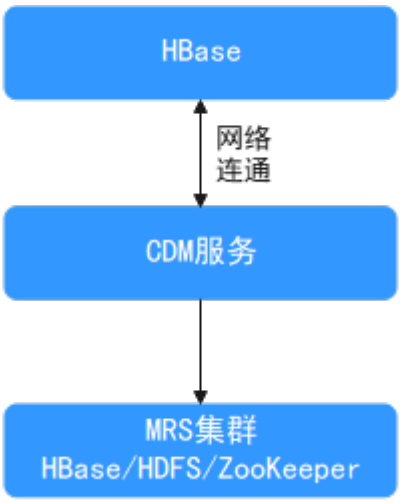
2.6 HBase 数据迁移到华为云 MRS 服务

场景介绍

本章节适用于将线下IDC机房或者公有云HBase集群中的数据（支持数据量在几十TB级别或以下的数据量级）迁移到华为云MRS服务。

本章节以通过**华为云CDM服务** 2.9.1.200版本进行数据迁移为例介绍。不同版本操作可能有差异，具体操作详情以实际版本对应的操作指导为准。

图 2-12 HBase 数据迁移示意



HBase会把数据存储在HDFS上，主要包括HFile文件和WAL文件，由配置项“hbase.rootdir”指定在HDFS上的路径，华为云MRS集群的默认存储位置是“/hbase”文件夹下。

HBase自带的一些机制和工具命令也可以实现数据搬迁，例如通过导出Snapshots快照、Export/Import、CopyTable方式等，可以参考Apache官网相关内容。

CDM服务支持迁移的数据源可参考支持的数据源，数据源为Apache HBase时，建议使用的版本为2.1.X、1.3.X，请执行搬迁前务必确认是否支持搬迁。

方案优势

场景化迁移通过迁移快照数据然后再恢复表数据的方法，能大大提升迁移效率。

对系统影响

搬迁数据量较大时，对网络通信要求较高，执行搬迁任务时，可能会影响其他业务，建议在业务空闲期进行数据迁移任务。

全量数据迁移

步骤1 登录CDM管理控制台。

步骤2 创建CDM集群，该CDM集群的安全组、虚拟私有云、子网需要和迁移目的端集群保持一致，保证CDM集群和待迁移的MRS集群之间网络互通。

步骤3 在“集群管理”页面单击待操作集群对应“操作”列的“作业管理”。

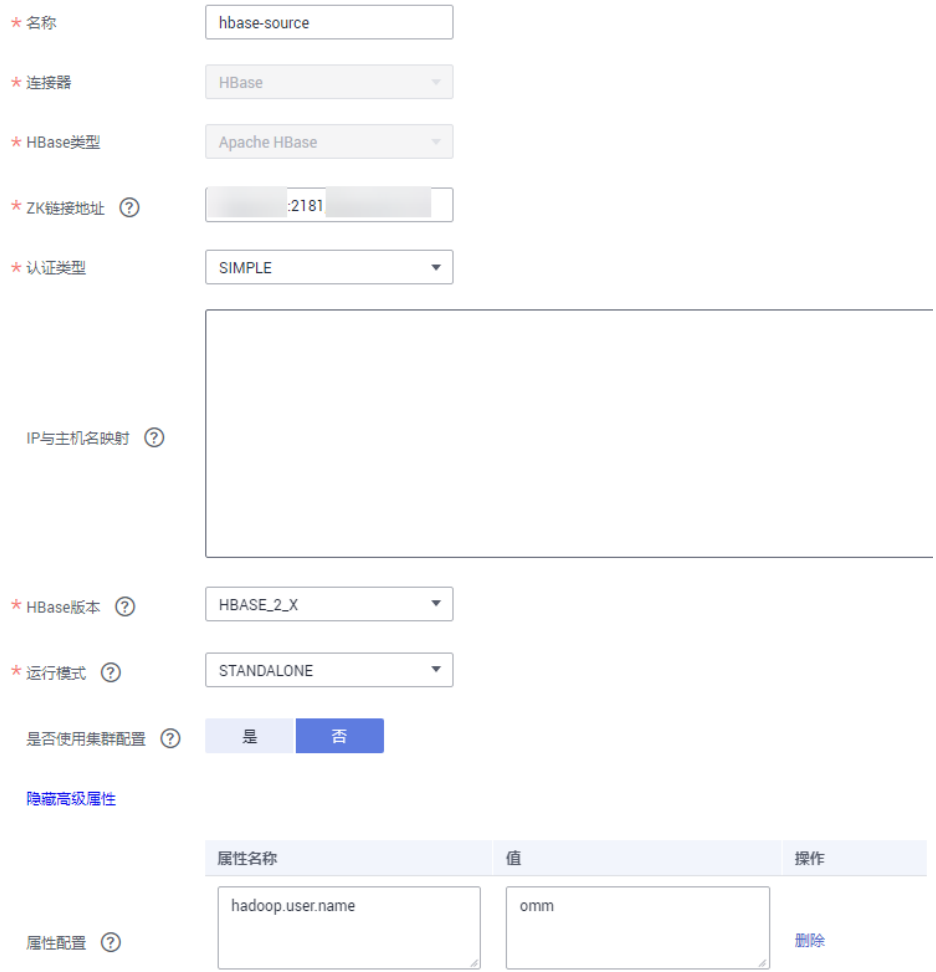
步骤4 在“连接管理”页签，单击“新建连接”。

步骤5 参考CDM服务的新建连接页面，添加到迁移源端集群的连接，连接器类型根据实际集群来选择，例如连接器类型选择“Apache HBase”。

说明

（可选）HBase迁移建议使用高权限用户，单击“显示高级属性”，新增迁移所需用户“hadoop.user.name = *用户名*（如**omm**用户）”。

图 2-13 到迁移源端集群的连接



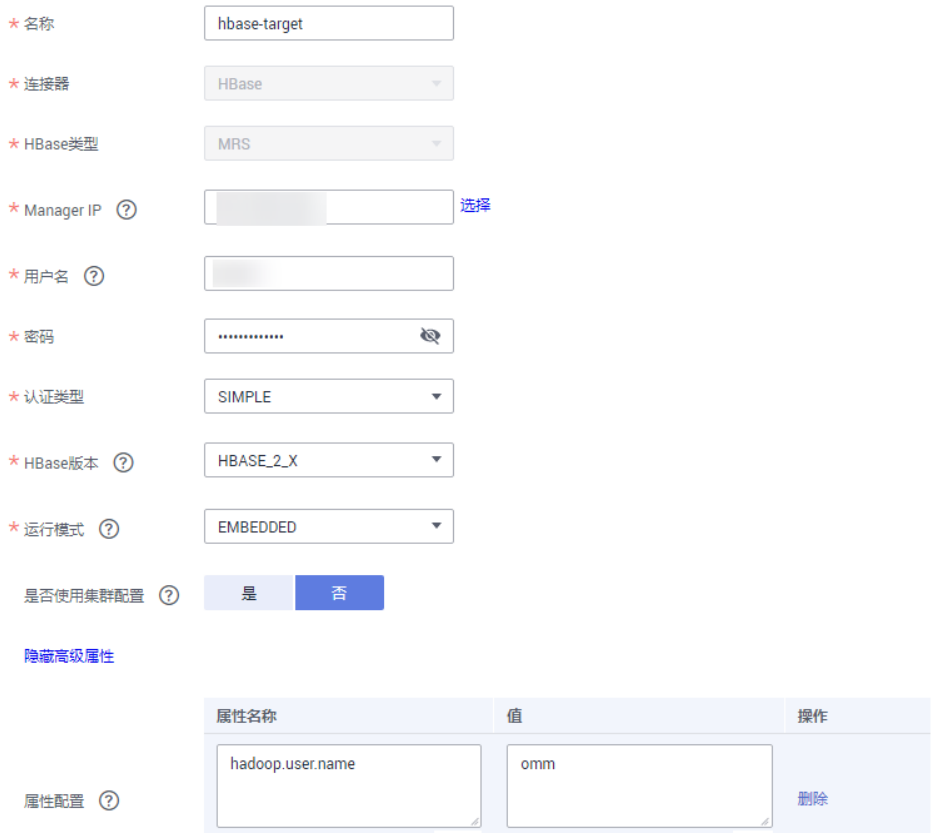
步骤6 在“连接管理”页签，单击“新建连接”。

步骤7 参考**CDM服务的新建连接**页面，添加到迁移目的端集群的连接，连接器类型根据实际集群来选择，例如连接器类型选择“MRS HBase”。

说明

（可选）HBase迁移建议使用高权限用户，单击“显示高级属性”，新增迁移所需用户“hadoop.user.name = *用户名*（如**omm**用户）”。

图 2-14 到迁移目的端集群的连接



步骤8 选择“作业管理”的“表/文件迁移”页签，单击“新建作业”。

步骤9 进入作业参数配置界面，配置作业名称、源端作业和目的端作业参数，并选择要迁移的数据表，单击“下一步”。

图 2-15 HBase 作业配置

步骤10 配置源字段和目的字段的映射关系， 并单击“下一步”。

步骤11 进入任务配置页面，不做修改，直接单击“保存”。

步骤12 选择“作业管理”的“表/文件迁移”页签，在待运行作业的“操作”列单击“运行”，即可开始HBase数据迁移。

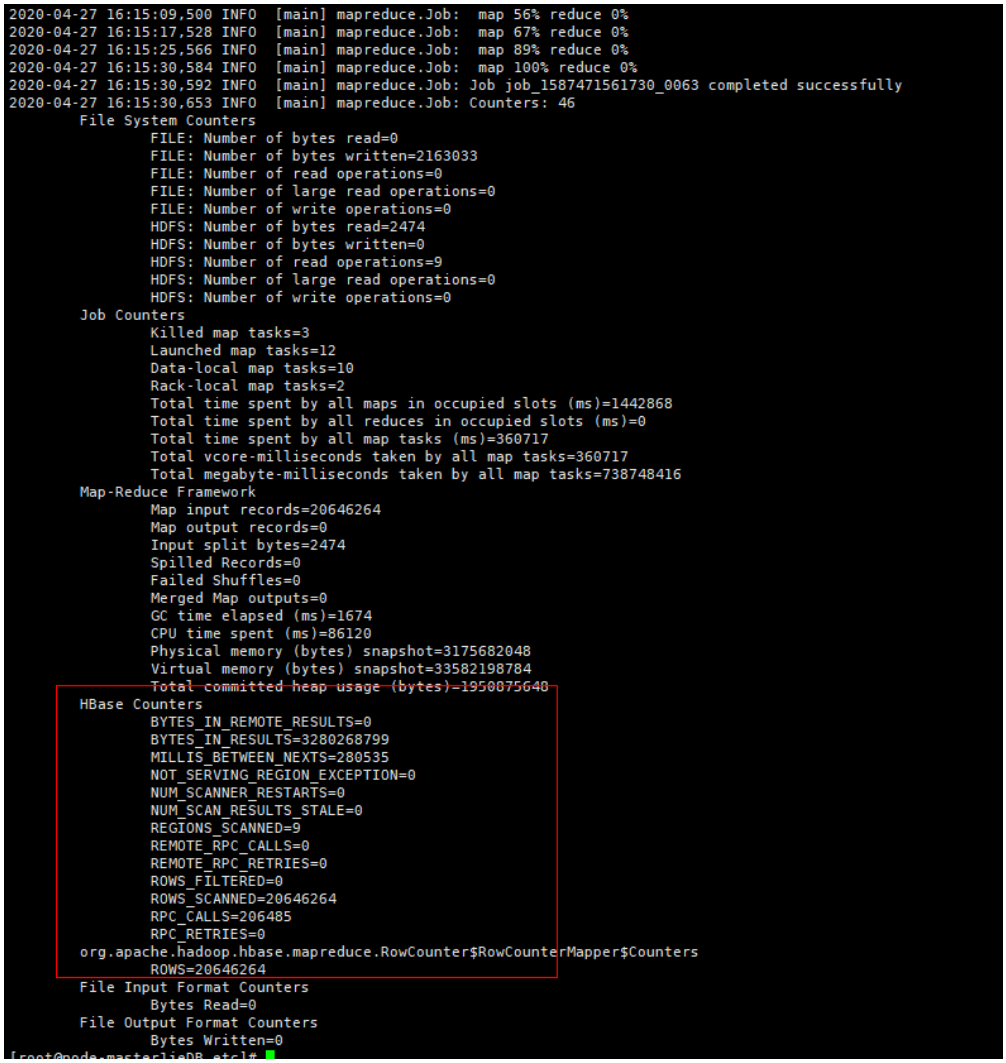
步骤13 迁移完成后，可以在目的端集群和源端集群的HBase Shell命令行中，通过同样的查询语句，对比查询结果进行验证。

例如：

- 在目的端集群和源端集群上通过查询BTable表的记录数来确认数据条数是否一致，可添加“--endtime”参数排除迁移期间源端集群上有数据更新的影响。

  hbase org.apache.hadoop.hbase.mapreduce.RowCounter *BTable* **--endtime=1587973835000**

  图 2-16 查询 BTable 表的记录数



- 可执行以下命令查询指定时间段内的数据进行对比。

  scan '*BTable* **', {TIMERANGE=>[1587973235000, 1587973835000]}**

----结束

增量数据迁移

在业务割接前，如果源端集群上有新增数据，需要定期将新增数据搬迁到目的端集群。一般每天更新的数据量在GB级别可以使用CDM的“整库迁移”指定时间段的方式进行HBase新增数据迁移。

当前使用CDM的“整库迁移”功能时的限制：如果源HBase集群中被删除操作的数据无法同步到目的端集群上。

场景迁移的HBase连接器不能与“整库迁移”共用，因此需要单独配置“HBase”连接器。

步骤1 参考**全量数据迁移**的**步骤1**~**步骤7**步骤新增两个“HBase”连接器，连接器类型根据实际集群来选择。

例如选择连接器类型时分别为源端集群和目的端集群选择“MRS HBase”和“Apache HBase”。

图 2-17 HBase 增量迁移连接

步骤2 选择“作业管理”的“整库迁移”页签，单击“新建作业”。

步骤3 进入作业参数配置界面，作业相关信息配置完成后单击“下一步”。

- 作业名称：用户自定义作业名称，例如hbase-increase。
- 源端作业配置：源连接名称请选择新创建的到源端集群的连接名称，并展开高级属性配置迁移数据的时间段。
- 目的端作业配置：目的连接名称请选择新创建的到目的端集群的连接名称，其他不填写。

图 2-18 HBase 增量迁移作业配置

步骤4 选择要迁移的数据表， 单击“下一步”，单击“保存”。

步骤5 选择“作业管理”的“整库迁移”页签，在待运行作业的“操作”列单击“运行”，即可开始HBase数据增量迁移。

----结束

2.7 Hive 数据迁移到华为云 MRS 服务

场景介绍

本章节适用于将线下IDC机房或者公有云Hive集群中的数据（支持数据量在几十TB级别或以下的数据量级）迁移到华为云MRS服务。

Hive数据迁移分两部分内容：

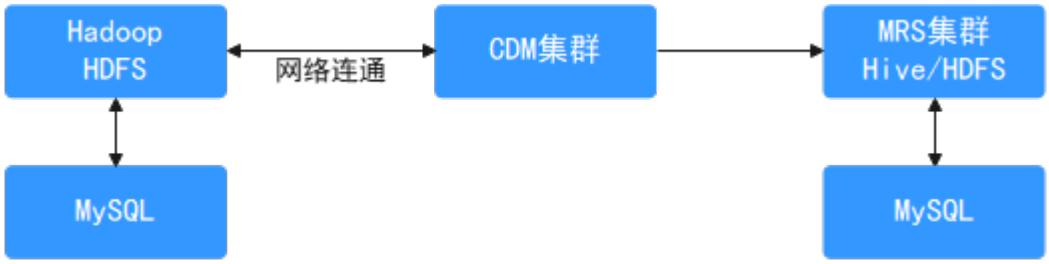
- Hive的元数据信息，存储在MySQL等数据库中。MRS Hive集群的元数据会默认存储到MRS DBService组件，也可以选择RDS（MySQL）作为外置元数据库。
- Hive的业务数据，存储在HDFS文件系统或OBS对象存储中。

使用华为云CDM服务“场景迁移功能”可以一键式便捷地完成Hive数据的迁移。

本章节以通过**华为云CDM服务** 2.9.1.200版本进行数据迁移为例介绍。不同版本操作可能有差异，具体操作详情以实际版本对应的操作指导为准。

CDM服务支持迁移的数据源可参考支持的数据源，数据源为Apache Hive时，不支持2.x版本，建议使用的版本为1.2.X、3.1.X，请执行搬迁前务必确认是否支持搬迁。

图 2-19 Hive 数据迁移示意



方案优势

场景化迁移通过迁移快照数据然后再恢复表数据的方法，能大大提升迁移效率。

对系统影响

搬迁数据量较大时，对网络通信要求较高，执行搬迁任务时，可能会影响其他业务，建议在业务空闲期进行数据迁移任务。

操作步骤

步骤1 登录CDM管理控制台。

步骤2 **创建CDM集群**，该CDM集群的安全组、虚拟私有云、子网需要和迁移目的端集群保持一致，保证CDM集群和MRS集群之间网络互通。

步骤3 在“集群管理”页面单击待操作集群对应“操作”列的“作业管理”。

步骤4 在“连接管理”页签，单击“新建连接”。

步骤5 参考**CDM服务的新建连接**页面，分别添加到迁移源端集群和迁移目的端集群的连接。

连接类型根据实际集群来选择，如果是MRS集群，连接器类型可以选择“MRS Hive”，如果是自建集群可以选择“Apache Hive”。

图 2-20 创建 Hive 连接

步骤6 在迁移目的端集群中创建数据迁移后的存储数据库。

步骤7 选择“作业管理”的“表/文件迁移”页签，单击“新建作业”。

步骤8 进入作业参数配置界面，配置作业名称，并分别为源连接和目的连接选择**步骤5**中创建的对应数据连接并选择要迁移的数据库和表名，单击“下一步”。

图 2-21 Hive 作业配置

步骤9 配置源字段和目的字段的映射关系， 并单击“下一步”。

步骤10 进入任务配置页面，不做修改，直接单击“保存”。

步骤11 选择“作业管理”的“表/文件迁移”页签，在待运行作业的“操作”列单击“运行”，即可开始Hive数据迁移。

步骤12 迁移完成后，可以在目的端集群和源端集群的Hive Beeline命令行中，通过同样的查询语句，对比查询结果进行验证。

例如在目的端集群和源端集群上通过查询**catalog_sales**表的记录数来确认数据条数是否一致。

select count(*) from *catalog_sales*;

图 2-22 源端集群数据记录

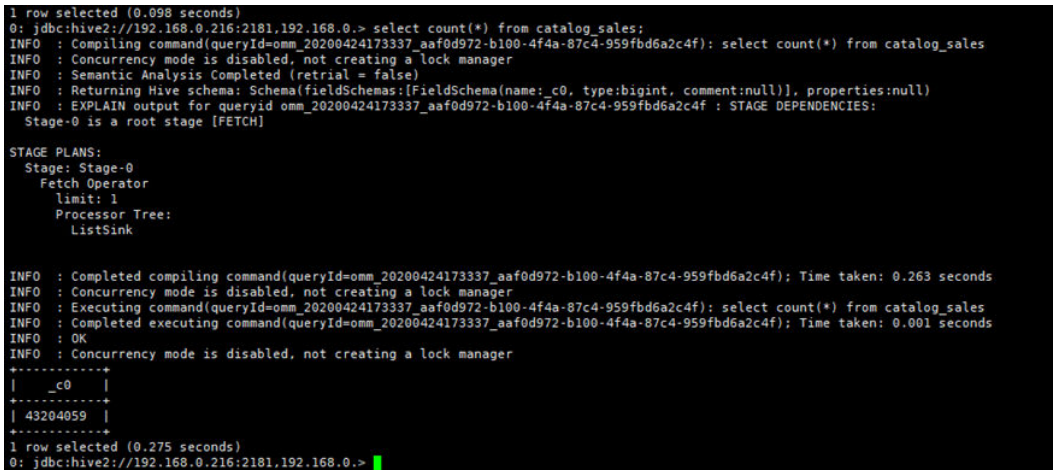
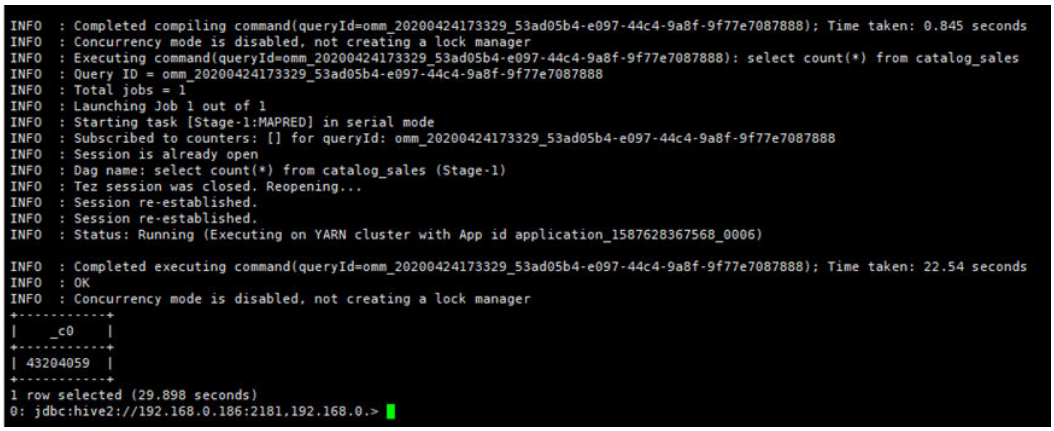


图 2-23 目的端集群数据记录



步骤13 （可选）如果源端集群中有新增数据需要定期将新增数据迁移至目的端集群，则根据数据新增方式进行不同方式的迁移。配置定期任务增量迁移数据，直到所有业务迁移至目的端集群。

- Hive表数据修改、未新增删除表、未修改已有表的数据结构：此时Hive表已经创建好，仅需迁移Hive存储在HDFS或OBS上的文件即可，请参考**Hadoop数据迁移到华为云MRS服务**页面新增数据迁移方式进行数据迁移。
- Hive表有新增：请选择“作业管理”的“表/文件迁移”页签，在Hive迁移作业的“操作”列单击“编辑”，选择新增的数据表进行数据迁移。
- Hive表有删除或已有表的数据结构有修改：请在目的端集群中手动删除对应表或手动更新变更的表结构。

----结束

2.8 使用 BulkLoad 向 HBase 中批量导入数据

经常面临向HBase中导入大量数据的情景，向HBase中批量加载数据的方式有很多种，最直接方式是调用HBase的API使用put方法插入数据；另外一种是用MapReduce的方式从HDFS上加载数据。但是这两种方式效率都不是很高，因为HBase频繁进行flush、compact、split操作需要消耗较大的CPU和网络资源，并且RegionServer压力也比较大。

本实践基于华为云MapReduce服务，用于指导您创建MRS集群后，使用BulkLoad方式向HBase中批量导入本地数据，在首次数据加载时，能极大的提高写入效率，并降低对Region Server节点的写入压力。

基本内容如下所示：

1. **创建MRS离线查询集群**。
2. **将本地数据导入到HDFS中**。
3. **创建HBase表**。
4. **生成HFile文件并导入HBase**。

场景描述

BulkLoad方式调用MapReduce的job直接将数据输出成HBase table内部的存储格式的文件HFile，然后将生成的StoreFiles加载到集群的相应节点。这种方式无需进行flush、compact、split等过程，不占用Region资源，不会产生巨量的写入I/O，所以需要较少的CPU和网络资源。

BulkLoad适合的场景：

- 大量数据一次性加载到HBase。
- 对数据加载到HBase可靠性要求不高，不需要生成WAL文件。
- 使用put加载大量数据到HBase速度变慢，且查询速度变慢时。
- 加载到HBase新生成的单个HFile文件大小接近HDFS block大小。

创建 MRS 离线查询集群

1. 进入**购买MRS集群页面**。
2. 选择“快速购买”，填写配置参数。

表 2-13 表 1 软件配置

参数项	取值
区域	华北-北京四
计费模式	按需计费
集群名称	MRS_hbase
版本类型	普通版
集群版本	MRS 3.1.0
组件选择	HBase查询集群
可用区	可用区1
企业项目	default
虚拟私有云	vpc-01
子网	subnet-01
Kerberos认证	开启

参数项	取值
用户名	root/admin
密码	设置密码登录集群管理页面及ECS节点用户的密码，例如：Test!@12345。
确认密码	再次输入设置用户密码
通信安全授权	勾选“确认授权”

图 2-24 创建 HBase 查询集群

3. 单击“立即购买”，等待MRS集群创建成功。

将本地数据导入到 HDFS 中

1. 在本地准备一个学生信息文件“info.txt”，例如内容如下：

   字段信息依次为：学号、姓名、生日、性别、住址

```
20200101245,张xx,20150324,男,City1
20200101246,李xx,20150202,男,City2
20200101247,杨xx,20151101,女,City3
20200101248,陈xx,20150218,男,City4
20200101249,李xx,20150801,女,City5
20200101250,王xx,20150315,男,City6
20200101251,李xx,20151201,男,City7
20200101252,孙xx,20150916,女,City8
20200101253,林xx,20150303,男,City9
```

2. 登录对象存储服务OBS控制台，单击“创建桶”，填写以下参数，单击“立即创建”。

表 2-14 桶参数

参数项	取值
区域	华北-北京四
桶名称	mrs-hbase
数据冗余存储策略	单AZ存储
默认存储类别	标准存储
桶策略	私有
默认加密	关闭
归档数据直读	关闭
企业项目	default
标签	-

等待桶创建好，单击桶名称，选择“对象 > 上传对象”，将数据文件上传至OBS桶内。

图 2-25 上传数据文件

3. 切换回MRS控制台，单击创建好的MRS集群名称，进入“概览”，单击“IAM用户同步”所在行的“单击同步”，等待约5分钟同步完成。
4. 将数据文件上传HDFS。
   a. 在“文件管理”页签，选择“HDFS文件列表”，进入数据存储目录，如“/tmp/test”。

      “/tmp/test”目录仅为示例，可以是界面上的任何目录，也可以通过“新建”创建新的文件夹。
   b. 单击“导入数据”。
      - OBS路径：选择上面创建好的OBS桶名，找到info.txt文件，单击“是”。
      - HDFS路径：选择HDFS路径，例如“/tmp/test”，单击“是”。
   c. 单击“确定”，等待导入成功，此时数据文件已上传至HDFS。

图 2-26 导入数据

创建 HBase 表

1. 登录集群的FusionInsight Manager页面（如果没有弹性IP，需提前购买弹性IP），新建一个用户hbasetest，绑定用户组supergroup，绑定角色System_administrator。

2. 下载并安装集群全量客户端，例如在主Master节点上安装，客户端安装目录为“/opt/client”，相关操作可参考**安装客户端**。

   也可直接使用Master节点中自带的集群客户端，安装目录为“/opt/Bigdata/client”。

3. 为主Master节点绑定一个弹性IP，然后使用**root**用户登录主Master节点，并进入客户端所在目录并认证用户。

   cd /opt/client

   source bigdata_env

   kinit hbasetest

4. 执行**hbase shell**进入HBase Shell命令行界面。

   需要根据导入数据，规划HBase数据表的表名、rowkey、列族、列，考虑好row key分配在创建表时进行预分割。

   执行以下命令创建表“student_info”。

   create 'student_info', {NAME => 'base',COMPRESSION => 'SNAPPY', DATA_BLOCK_ENCODING => 'FAST_DIFF'},SPLITS => ['1','2','3','4','5','6','7','8']

   - NAME => 'base'：HBase表列族名称。
   - COMPRESSION：压缩方式
   - DATA_BLOCK_ENCODING：编码算法
   - SPLITS：预分region

5. 查看表是否创建成功，然后退出HBase Shell命令行界面。

list

生成 HFile 文件并导入 HBase

1. 创建自定义导入的模板文件，例如模板文件为“/opt/configuration_index.xml”（模板文件样例可从“客户端安装目录/HBase/hbase/conf/index_import.xml.template”获取）。

   vi /opt/configuration_index.xml

   例如本案例中，模板文件如下：

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <configuration>
   <!--column_num要和数据文件中的列的数量对应：5列 -->
    <import column_num="5" id="first">
     <columns>
      <column type="string" index="1">P_ID</column>
      <column type="string" index="2">P_NAME</column>
      <column type="string" index="3">P_BIRTH</column>
      <column type="string" index="4">P_GENDER</column>
      <column type="string" index="5">P_DISTRICT</column>
     </columns>
   <!--reverse(P_BIRTH)：反转出生年月避免热点 -->
   <!--substring(P_NAME,0,1)：截取姓 -->
   <!--substring(P_ID,0,6)：截身学号前六位 -->
      <rowkey>
       reverse(P_BIRTH)+'_'+substring(P_NAME,0,1)+'_'+substring(P_ID,0,6)
      </rowkey>
     <qualifiers>
     <!--family的指定要和表的列族名称对应。 -->
      <normal family="base">
       <qualifier column="P_ID">H_ID</qualifier>
       <qualifier column="P_NAME">H_NAME</qualifier>
       <qualifier column="P_BIRTH">H_BIRTH</qualifier>
       <qualifier column="P_GENDER">H_GENDER</qualifier>
       <qualifier column="P_DISTRICT">H_DISTRICT</qualifier>
      </normal>
     </qualifiers>
    </import>
   </configuration>
   ```

2. 执行如下命令，生成HFile文件。

   hbase com.huawei.hadoop.hbase.tools.bulkload.ImportData -Dimport.separator=',' -Dimport.hfile.output=/tmp/test/hfile /opt/configuration_index.xml student_info /tmp/test/info.txt

   - -Dimport.separator：分隔符。
   - -Dimport.hfile.output：执行结果输出路径。
   - /opt/configuration_index.xml：指向自定义的模板文件。
   - student_info：要操作的HBase表名。
   - /tmp/test/info.txt：指的是要批量上传的HDFS数据目录。
   - com.huawei.hadoop.hbase.tools.bulkload.IndexImportData：导入时创建二级索引使用IndexImportData；如果不创建二级索引，使用ImportData

   等待MapReduce任务执行成功，输出路径下生成HFile文件。

   hdfs dfs -ls /tmp/test/hfile

   ```
   Found 2 items
   -rw-r--r--   3 hbasetest hadoop          0 2021-05-14 11:39 /tmp/test/hfile/_SUCCESS
   drwxr-xr-x   - hbasetest hadoop          0 2021-05-14 11:39 /tmp/test/hfile/base
   ```

3. 执行如下命令将HFile导入HBase表。

hbase org.apache.hadoop.hbase.mapreduce.LoadIncrementalHFiles /tmp/test/hfile student_info

4. 进入HBase Shell命令行界面，查看表内容。

   hbase shell

   scan 'student_info', {FORMATTER => 'toString'}

```
ROW                                COLUMN+CELL
 10115102_杨_202001                column=base:H_BIRTH, timestamp=2021-05-14T15:28:56.755,
value=20151101
 10115102_杨_202001                column=base:H_DISTRICT, timestamp=2021-05-14T15:28:56.755,
value=City3
 10115102_杨_202001                column=base:H_GENDER, timestamp=2021-05-14T15:28:56.755,
value=女
 10115102_杨_202001                column=base:H_ID, timestamp=2021-05-14T15:28:56.755,
value=20200101247
 10115102_杨_202001                column=base:H_NAME, timestamp=2021-05-14T15:28:56.755,
value=杨xx
 10215102_李_202001                column=base:H_BIRTH, timestamp=2021-05-14T15:28:56.755,
value=20151201
 10215102_李_202001                column=base:H_DISTRICT, timestamp=2021-05-14T15:28:56.755,
value=City7
...
```

5. 数据导入集群后，就可以继续基于大数据平台上层应用对数据进行分析处理了。

2.9 MySQL 数据迁移到 MRS 集群 Hive 分区表

MapReduce服务（MapReduce Service，简称MRS）提供企业级大数据集群云服务，里面包含HDFS、Hive、Spark等组件，适用于企业海量数据分析。

其中Hive提供类SQL查询语言，帮助用户对大规模的数据进行提取、转换和加载，即通常所称的ETL（Extraction，Transformation，and Loading）操作。对庞大的数据集查询需要耗费大量的时间去处理，在许多场景下，可以通过建立Hive分区方法减少每一次扫描的总数据量，这种做法可以显著地改善性能。

Hive的分区使用HDFS的子目录功能实现，每一个子目录包含了分区对应的列名和每一列的值。当分区很多时，会有很多HDFS子目录，如果不依赖工具，将外部数据加载到Hive表各分区不是一件容易的事情。云数据迁移服务（CDM）可以请轻松将外部数据源（关系数据库、对象存储服务、文件系统服务等）加载到Hive分区表。

本实践为您演示使用CDM云服务将MySQL数据导入到MRS集群内的Hive分区表中。

操作场景

假设MySQL数据库中有一张表“trip_data”，保存了自行车骑行记录，里面有起始时间、结束时间，起始站点、结束站点、骑手ID等信息。

“trip_data”表字段定义如图2-27所示。

图 2-27 MySQL 表字段

Column Name	#	Data Type
TripID	1	int(11)
Duration	2	int(11)
StartDate	3	timestamp
StartStation	4	varchar(64)
StartTerminal	5	int(11)
EndDate	6	timestamp
EndStation	7	varchar(64)
EndTerminal	8	int(11)
Bike	9	int(11)
SubscriberType	10	varchar(32)
ZipCodev	11	varchar(10)

使用CDM将MySQL中的导入到MRS Hive分区表，流程如下：

1. **在MRS Hive上创建Hive分区表**
2. **创建CDM集群并绑定EIP**
3. **创建MySQL连接**
4. **创建Hive连接**
5. **创建迁移作业**

前提条件

- 已经购买包含有Hive服务的MRS集群。
- 已获取连接MySQL数据库的IP地址、端口、数据库名称、用户名、密码，且该用户拥有MySQL数据库的读写权限。
- 已参考**管理驱动**，上传了MySQL数据库驱动。

在 MRS Hive 上创建 Hive 分区表

在MRS的Hive客户端中，执行以下SQL语句创建一张Hive分区表，表名与MySQL上的表trip_data一致，且Hive表比MySQL表多建三个字段y、ym、ymd，作为Hive的分区字段。

SQL语句如下：

```
create table trip_data(TripID int,Duration int,StartDate timestamp,StartStation varchar(64),StartTerminal int,EndDate timestamp,EndStation varchar(64),EndTerminal int,Bike int,SubscriberType varchar(32),ZipCodev varchar(10))partitioned by (y int,ym int,ymd int);
```

说明

Hive表trip_data有三个分区字段：骑行起始时间的年、骑行起始时间的年月、骑行起始时间的年月日。

例如一条骑行记录的起始时间为2018/5/11 9:40，那么这条记录会保存在分区trip_data/2018/201805/20180511下面。

对trip_data进行按时间维度统计汇总时，只需要对局部数据扫描，大大提升性能。

创建 CDM 集群并绑定 EIP

步骤1 如果是独立CDM服务，参考**创建集群**创建CDM集群；如果是作为DataArts Studio服务CDM组件使用，参考**创建集群**创建CDM集群。

关键配置如下：

- CDM集群的规格，按待迁移的数据量选择，一般选择cdm.medium即可，满足大部分迁移场景。
- CDM集群所在VPC、子网、安全组，选择与MRS集群所在的网络一致。

步骤2 CDM集群创建完成后，选择集群操作列的“绑定弹性IP”，CDM通过EIP访问MySQL。

图 2-28 集群列表

说明

如果用户对本地数据源的访问通道做了SSL加密，则CDM无法通过弹性IP连接数据源。

----结束

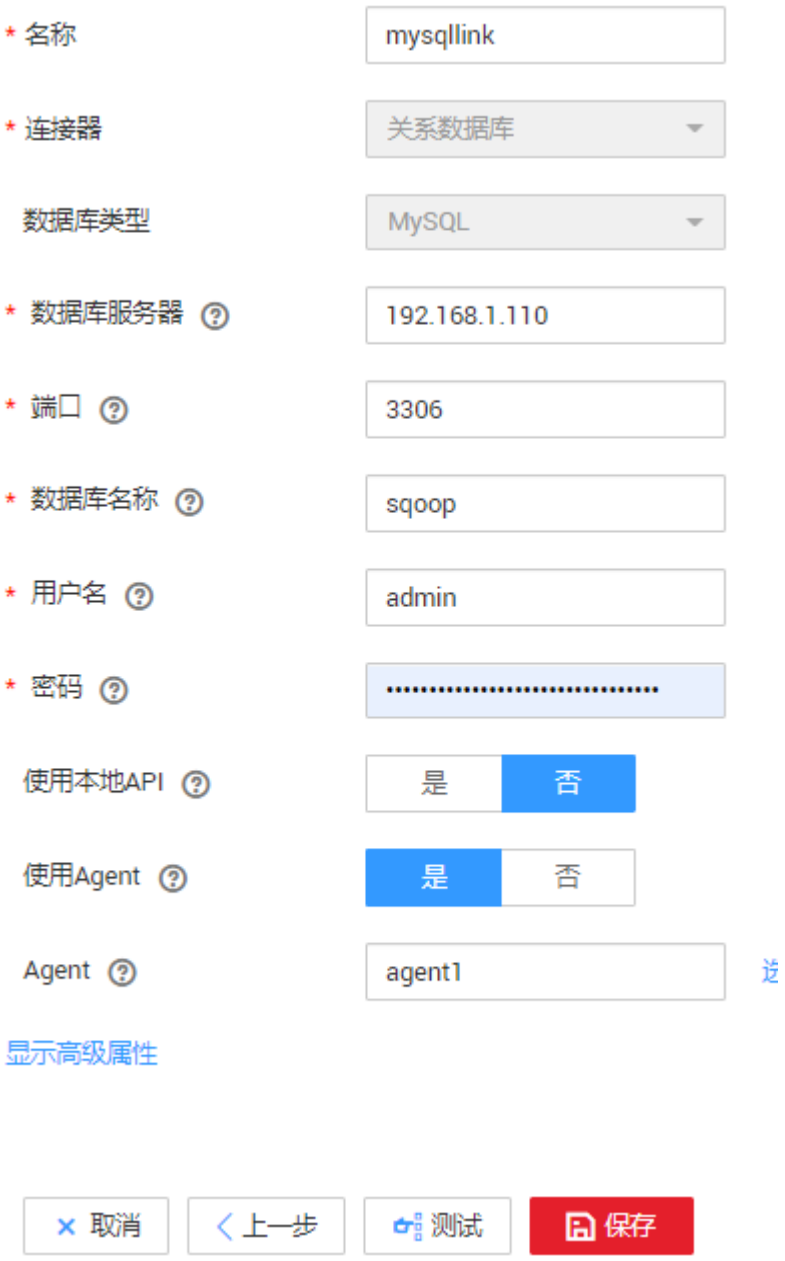
创建 MySQL 连接

步骤1 在CDM集群管理界面，单击集群后的“作业管理”，选择“连接管理 > 新建连接”，进入连接器类型的选择界面。

图 2-29 选择连接器类型

步骤2 选择“MySQL”后单击“下一步”，配置MySQL连接的参数。

图 2-30 创建 MySQL 连接



单击“显示高级属性”可查看更多可选参数，具体请参见**配置关系数据库连接**。此处保持默认，必填参数如**表2-15**所示。

表 2-15 MySQL 连接参数

参数名	说明	取值样例
名称	输入便于记忆和区分的连接名称。	mysqllink
数据库服务器	MySQL数据库的IP地址或域名。	192.168.1.110
端口	MySQL数据库的端口。	3306

参数名	说明	取值样例
数据库名称	MySQL数据库的名称。	sqoop
用户名	拥有MySQL数据库的读、写和删除权限的用户。	admin
密码	用户的密码。	-
使用Agent	是否选择通过Agent从源端提取数据。	是
Agent	单击“选择”，选择**连接Agent**中已创建的Agent。	-

步骤3 单击“保存”回到连接管理界面。

说明

如果保存时出错，一般是由于MySQL数据库的安全设置问题，需要设置允许CDM集群的EIP访问MySQL数据库。

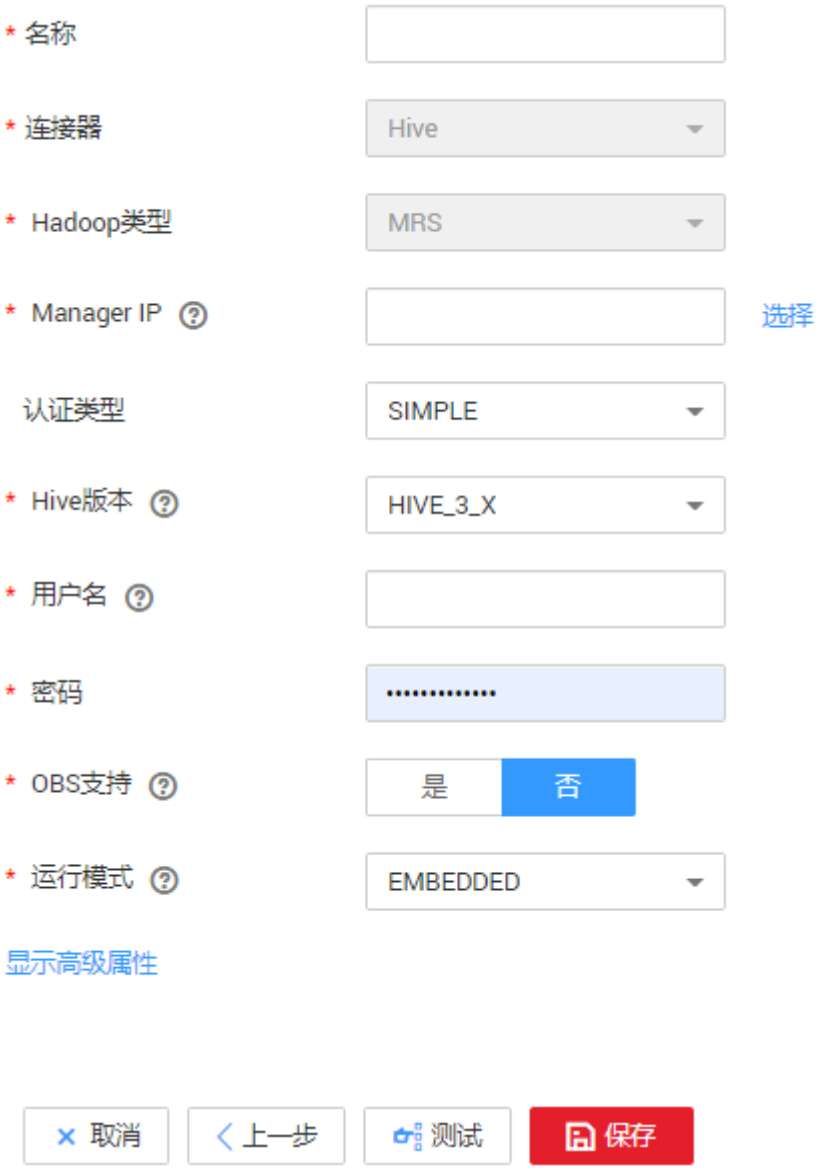
----结束

创建 Hive 连接

步骤1 在连接管理界面，单击“新建连接”，连接器类型选择“MRS Hive”。

步骤2 单击“下一步”配置Hive连接参数，如**图2-31**所示。

图 2-31 创建 Hive 连接



各参数说明如表2-16所示，需要您根据实际情况配置。

表 2-16 MRS Hive 连接参数

参数名	说明	取值样例
名称	连接的名称，根据连接的数据源类型，用户可自定义便于记忆、区分的连接名。	hivelink
Manager IP	MRS Manager的浮动IP地址，可以单击输入框后的“选择”来选定已创建的MRS集群，CDM会自动填充下面的鉴权参数。	127.0.0.1

参数名	说明	取值样例
认证类型	访问MRS的认证类型： ● SIMPLE：非安全模式选择Simple鉴权。 ● KERBEROS：安全模式选择Kerberos鉴权。	SIMPLE
Hive版本	Hive的版本。根据服务端Hive版本设置。	HIVE_3_X
用户名	选择KERBEROS鉴权时，需要配置MRS Manager的用户名和密码。从HDFS导出目录时，如果需要创建快照，这里配置的用户需要HDFS系统的管理员权限。 如果要创建MRS安全集群的数据连接，不能使用admin用户。因为admin用户是默认的管理页面用户，这个用户无法作为安全集群的认证用户来使用。您可以创建一个新的MRS用户，然后在创建MRS数据连接时，“用户名”和“密码”填写为新建的MRS用户及其密码。 **说明** ● 如果CDM集群为2.9.0版本及之后版本，且MRS集群为3.1.0及之后版本，则所创建的用户至少需具备Manager_viewer的角色权限才能在CDM创建连接；如果需要对应组件的进行库、表、数据的操作，还需要添加对应组件的用户组权限。 ● 如果CDM集群为2.9.0之前的版本，或MRS集群为3.1.0之前的版本，则所创建的用户需要具备Manager_administrator或System_administrator权限，才能在CDM创建连接。 ● 仅具备Manager_tenant或Manager_auditor权限，无法创建连接。	cdm
密码	访问MRS Manager的用户密码。	-
OBS支持	需服务端支持OBS存储。在创建Hive表时，您可以指定将表存储在OBS中。	否
运行模式	“HIVE_3_X”版本支持该参数。支持以下模式： ● EMBEDDED：连接实例与CDM运行在一起，该模式性能较好。 ● STANDALONE：连接实例运行在独立进程。如果CDM需要对接多个Hadoop数据源（MRS、Hadoop或CloudTable），并且既有KERBEROS认证模式又有SIMPLE认证模式，只能使用STANDALONE模式或者配置不同的Agent。 **说明**：STANDALONE模式主要是用来解决版本冲突问题的运行模式。当同一种数据连接的源端或者目的端连接器的版本不一致时，存在jar包冲突的情况，这时需要将源端或目的端放在STANDALONE进程里，防止冲突导致迁移失败。	EMBEDDED
是否使用集群配置	您可以通过使用集群配置，简化Hadoop连接参数配置。	否

参数名	说明	取值样例
集群配置名	仅当“是否使用集群配置”为“是”时，此参数有效。此参数用于选择用户已经创建好的集群配置。 集群配置的创建方法请参见**管理集群配置**。	hive_01

步骤3 单击“保存”回到连接管理界面。

----结束

创建迁移作业

步骤1 在CDM集群管理界面，单击集群后的“作业管理”，选择“表/文件迁移 > 新建作业”，开始创建数据迁移任务，如图2-32所示。

图 2-32 创建 MySQL 到 Hive 的迁移任务

说明

“导入前清空数据”选“是”，这样每次导入前，会将之前已经导入到Hive表的数据清空。

步骤2 作业参数配置完成后，单击“下一步”，进入字段映射界面，如图2-33所示。

映射MySQL表和Hive表字段，Hive表比MySQL表多三个字段y、ym、ymd，即是Hive的分区字段。由于没有源表字段直接对应，需要配置表达式从源表的StartDate字段抽取。

图 2-33 Hive 字段映射

步骤3 单击 进入转换器列表界面，再选择“新建转换器 > 表达式转换”。

y、ym、ymd字段的表达式分别配置如下：

DateUtils.format(DateUtils.parseDate(row[2],"yyyy-MM-dd HH:mm:ss.SSS"),"yyyy")

DateUtils.format(DateUtils.parseDate(row[2],"yyyy-MM-dd HH:mm:ss.SSS"),"yyyyMM")

DateUtils.format(DateUtils.parseDate(row[2],"yyyy-MM-dd HH:mm:ss.SSS"),"yyyyMMdd")

说明

CDM的表达式已经预置常用字符串、日期、数值等类型的字段内容转换，详细请参见**字段转换**。

步骤4 单击“下一步”配置任务参数，一般情况下全部保持默认即可。

该步骤用户可以配置如下可选功能：

- 作业失败重试：如果作业执行失败，可选择是否自动重试，这里保持默认值“不重试”。
- 作业分组：选择作业所属的分组，默认分组为“DEFAULT”。在CDM“作业管理”界面，支持作业分组显示、按组批量启动作业、按分组导出作业等操作。
- 是否定时执行：如果需要配置作业定时自动执行，请参见**配置定时任务**。这里保持默认值“否”。
- 抽取并发数：设置同时执行的抽取任务数。这里保持默认值“1”。
- 是否写入脏数据：如果需要将作业执行过程中处理失败的数据、或者被清洗过滤掉的数据写入OBS中，以便后面查看，可通过该参数配置，写入脏数据前需要先配置好OBS连接。这里保持默认值“否”即可，不记录脏数据。
- 作业运行完是否删除：这里保持默认值“不删除”。

步骤5 单击“保存并运行”，回到作业管理界面，在作业管理界面可查看作业执行进度和结果。

步骤6 作业执行成功后，单击作业操作列的“历史记录”，可查看该作业的历史执行记录、读取和写入的统计数据。

在历史记录界面单击“日志”，可查看作业的日志信息。

----结束

2.10 MRS HDFS 数据迁移到 OBS

操作场景

CDM支持文件到文件类数据的迁移，本章节以MRS HDFS至OBS为例，介绍如何通过CDM将文件类数据迁移到OBS文件系统中。

流程如下：

1. **创建CDM集群并绑定EIP**
2. **创建MRS HDFS连接**
3. **创建OBS连接**
4. **创建迁移作业**

前提条件

- 已获取OBS的访问域名、端口，以及AK、SK信息。
- 已经创建包含有Hadoop服务的MRS集群。
- 拥有EIP配额并创建EIP。

创建 CDM 集群并绑定 EIP

步骤1 如果是独立CDM服务，参考**创建集群**创建CDM集群；如果是作为DataArts Studio服务CDM组件使用，参考**创建集群**创建CDM集群。

关键配置如下：

- CDM集群的规格，按待迁移的数据量选择，一般选择“cdm.medium”即可，满足大部分迁移场景。
- CDM集群所在VPC、子网、安全组，选择与MRS集群所在的网络一致。

步骤2 CDM集群创建完成后，选择集群操作列的“绑定弹性IP”，CDM通过EIP访问MRS HDFS。

图 2-34 集群列表

说明

如果用户对本地数据源的访问通道做了SSL加密，则CDM无法通过弹性IP连接数据源。

----结束

创建 MRS HDFS 连接

步骤1 在CDM集群管理界面，单击集群后的“作业管理”，选择“连接管理 > 新建连接”，连接器类型选择“MRS HDFS”后，单击“下一步”，配置MRS HDFS链接参数。

- 名称：用户自定义连接名称，例如“mrs_hdfs_link”。
- Manage IP：MRS Manager的IP地址，可以单击输入框后的“选择”来选定已创建的MRS集群，CDM会自动填充下面的鉴权参数。
- 用户名：选择KERBEROS鉴权时，需要配置MRS Manager中创建的用户。
  从HDFS导出目录时，如果需要创建快照，这里配置的用户需要具有HDFS系统的管理员权限。
- 密码：访问MRS Manager的用户密码。
- 认证类型：访问MRS的认证类型。
- 运行模式：选择HDFS连接的运行模式。

----结束

创建 OBS 连接

步骤1 在CDM集群管理界面，单击集群后的“作业管理”，选择“连接管理 > 新建连接”，连接器类型选择“对象存储服务（OBS）”后，单击“下一步”配置OBS连接参数，如图2-35所示。

- 名称：用户自定义连接名称，例如“obslink”。
- OBS终端节点、端口：配置为OBS实际的地址信息。
- OBS桶类型：保持默认。
- 访问标识（AK）、密钥（SK）：登录OBS的AK、SK信息。

图 2-35 创建 OBS 连接

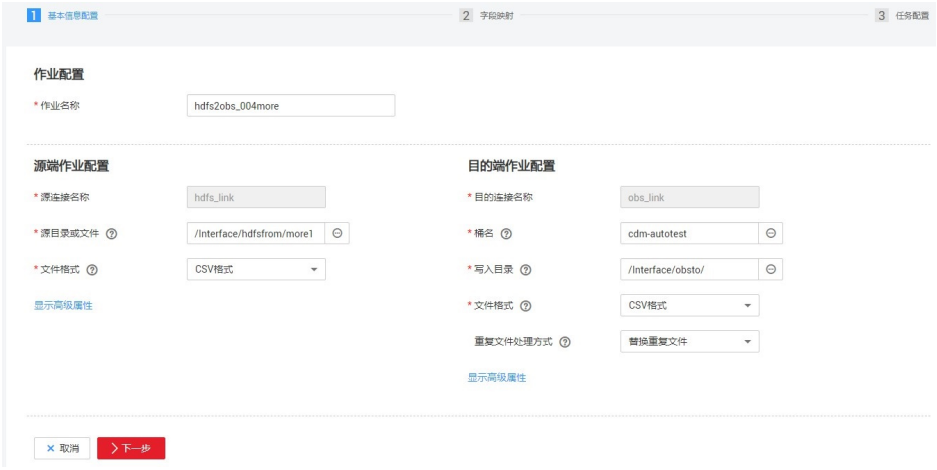
步骤2 单击“保存”回到连接管理界面。

----结束

创建迁移作业

步骤1 在CDM集群管理界面，单击集群后的“作业管理”，选择“表/文件迁移 > 新建作业”，开始创建从MRS HDFS导出数据到OBS的任务。

图 2-36 创建 MRS HDFS 到 OBS 的迁移任务



- 作业名称：用户自定义便于记忆、区分的任务名称。
- 源端作业配置
  - 源连接名称：选择**创建MRS HDFS连接**中的“hdfs_link”。
  - 源目录或文件：待迁移数据的目录或单个文件路径。
  - 文件格式：传输数据时所用的文件格式，这里选择“二进制格式”。不解析文件内容直接传输，不要求文件格式必须为二进制。适用于文件到文件的原样复制。
  - 其他可选参数一般情况下保持默认即可，详细说明请参见**配置HDFS源端参数**。
- 目的端作业配置
  - 目的连接名称：选择**创建OBS连接**中的“obs_link”。
  - 桶名：待迁移数据的桶。
  - 写入目录：写入数据到OBS服务器的目录。
  - 文件格式：迁移文件类数据到文件时，文件格式选择“二进制格式”。
  - 高级属性里的可选参数一般情况下保持默认既可，详细说明请参见**配置OBS目的端参数**。

步骤2 单击“下一步”进入字段映射界面，CDM会自动匹配源和目的字段。

- 如果字段映射顺序不匹配，可通过拖拽字段调整。
- CDM的表达式已经预置常用字符串、日期、数值等类型的字段内容转换，详细请参见**字段转换**。

步骤3 单击“下一步”配置任务参数，一般情况下全部保持默认即可。

该步骤用户可以配置如下可选功能：

- 作业失败重试：如果作业执行失败，可选择是否自动重试，这里保持默认值“不重试”。
- 作业分组：选择作业所属的分组，默认分组为“DEFAULT”。在CDM“作业管理”界面，支持作业分组显示、按组批量启动作业、按分组导出作业等操作。
- 是否定时执行：如果需要配置作业定时自动执行，请参见**配置定时任务**。这里保持默认值“否”。
- 抽取并发数：设置同时执行的抽取任务数。CDM支持多个文件的并发抽取，调大参数有利于提高迁移效率
- 是否写入脏数据：否，文件到文件属于二进制迁移，不存在脏数据。
- 作业运行完是否删除：这里保持默认值“不删除”。根据使用场景，也可配置为“删除”，防止迁移作业堆积。

步骤4 单击“保存并运行”，回到作业管理界面，在作业管理界面可查看作业执行进度和结果。

步骤5 作业执行成功后，单击作业操作列的“历史记录”，可查看该作业的历史执行记录、读取和写入的统计数据。

在历史记录界面单击“日志”，可查看作业的日志信息。

----结束

3 数据备份与恢复

3.1 HDFS 数据

打通数据传输通道

- 当源集群与目标集群部署在同一区域的不同VPC时，请创建两个VPC之间的网络连接，打通网络层面的数据传输通道。请参见**VPC对等连接**。
- 当源集群与目标集群部署在同一VPC但属于不同安全组时，在VPC管理控制台，为每个安全组分别添加安全组规则。规则的“协议”为“ANY”，“方向”为“入方向”，“源地址”为“安全组”且是对端集群的安全组。
  - 为源集群的安全组添加入方向规则，源地址选择目标集群的安全组。
  - 为目标集群的安全组添加入方向规则，源地址选择源集群的安全组。
- 当源集群与目标集群部署在同一VPC同一安全组且两个集群都开启了Kerberos认证，需为两个集群配置互信。

HDFS 数据备份

根据源集群与目标集群分别所处的区域及网络连通性，可分为以下几种数据备份场景：

- 同Region

  当源集群与目标集群处于同一Region时，根据**打通数据传输通道**进行网络配置，打通网络传输通道。使用Distcp工具执行如下命令将源集群的HDFS、HBase、Hive数据文件以及Hive元数据备份文件拷贝至目的集群。

  ```
  $HADOOP_HOME/bin/hadoop distcp <src> <dist> -p
  ```

  其中，各参数的含义如下。
  - *$HADOOP_HOME*：目的集群Hadoop客户端安装目录
  - <src>：源集群HDFS目录
  - <dist>：目的集群HDFS目录
- 不同Region

  当源集群与目标集群处于不同Region时，用Distcp工具将源集群数据拷贝到OBS，借助OBS跨区域复制功能（请参见**跨区域复制**）将数据复制到对应目的集群所在Region的OBS，然后通过Distcp工具将OBS数据拷贝到目的集群的HDFS

上。由于执行Distcp无法为OBS上的文件设置权限、属主/组等信息，因此当前场景在进行数据导出时也需要将HDFS的元数据信息进行导出并拷贝，以防HDFS文件属性信息丢失。

- 线下集群向云迁移

  线下集群可以通过如下两种方式将数据迁移至云：

  - 云专线（DC）

    为源集群与目标集群之间建立**云专线**，打通线下集群出口网关与线上VPC之间的网络，然后参考**同Region**执行Distcp进行拷贝。

  - 数据快递服务（DES）

    对于TB或PB级数据上云的场景，华为云提供**数据快递服务 DES**。将线下集群数据及已导出的元数据拷贝到DES盒子，快递服务将数据递送到华为云机房，然后通过**云数据迁移 CDM**将DES盒子数据拷贝到HDFS。

HDFS 元数据备份

HDFS数据需要导出的元数据信息包括文件及文件夹的权限和属主/组信息。可通过如下HDFS客户端命令导出。

```
$HADOOP_HOME/bin/hdfs dfs -ls -R <migrating_path> > /tmp/hdfs_meta.txt
```

其中，各参数的含义如下。

- *$HADOOP_HOME*：源集群Hadoop客户端安装目录
- *<migrating_path>*：HDFS上待迁移的数据目录
- /tmp/hdfs_meta.txt：导出的元数据信息保存在本地的路径。

说明

如果源集群与目标集群网络互通，且以超级管理员身份运行hadoop distcp命令进行数据拷贝，可以添加参数“-p”让distcp在拷贝数据的同时在目标集群上分别恢复相应文件的元数据信息。因此在这种场景下可直接跳过本步骤。

HDFS 文件属性恢复

根据导出的权限信息在目的集群的后台使用HDFS命令对文件的权限及属主/组信息进行恢复。

```
$HADOOP_HOME/bin/hdfs dfs -chmod <MODE> <path>
$HADOOP_HOME/bin/hdfs dfs -chown <OWNER> <path>
```

3.2 Hive 元数据

Hive 元数据备份

Hive表数据存储在HDFS上，表数据及表数据的元数据由HDFS统一按数据目录进行迁移。而Hive表的元数据根据集群的不同配置，可以存储在不同类型的关系型数据库中（如MySQL，PostgreSQL，Oracle等）。本指导导出的Hive表元数据即存储在关系型数据库中的Hive表的描述信息。

业界主流大数据发行版均支持Sqoop的安装，如果是自建的社区版大数据集群，可下载社区版Sqoop进行安装。借助Sqoop来解耦导出的元数据与关系型数据库的强依赖，将Hive元数据导出到HDFS上，与表数据一同迁移后进行恢复。步骤如下：

步骤1 在源集群上下载并安装Sqoop工具。请参见**http://sqoop.apache.org/**。

步骤2 下载相应关系型数据库的jdbc驱动放置到$Sqoop_Home/lib目录。

步骤3 执行如下命令导出所有Hive元数据表。所有导出数据保存在HDFS上的/user/<*user_name*>/<*table_name*>目录。

```
$Sqoop_Home/bin/sqoop import --connect jdbc:<driver_type>://<ip>:<port>/<database> --table <table_name> --username <user> -password <passwd> -m 1
```

其中，各参数的含义如下。

- *$Sqoop_Home*：Sqoop的安装目录
- <*driver_type*>：数据库类型
- <*ip*>：源集群数据库的IP地址
- <*port*>：源集群数据库的端口号
- <*table_name*>：待导出的表名称
- <*user*>：用户名
- <*passwd*>：用户密码

说明

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。

----结束

Hive 元数据恢复

在目的集群中安装并使用Sqoop命令将导出的Hive元数据导入MRS集群DBService。

```
$Sqoop_Home/bin/sqoop export --connect jdbc:postgresql://<ip>:20051/hivemeta --table <table_name> --username hive -password <passwd> --export-dir <export_from>
```

其中，各参数的含义如下。

- *$Sqoop_Home*：目的集群上Sqoop的安装目录
- <*ip*>：目的集群上数据库的IP地址
- <*table_name*>：待恢复的表名称
- <*passwd*>：hive用户的密码
- <*export_from*>：元数据在目的集群的HDFS地址。

说明

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。

3.3 Hive 数据

当前没有单独的Hive数据备份，请参考**HDFS数据**进行操作。

3.4 HBase 数据

当前HBase上可以使用的数据备份方式主要有以下几种方式，本指导为您介绍通过以下几种方式进行HBase数据导出、HBase数据导入的操作步骤。

- Snapshots
- Replication
- Export
- CopyTable
- HTable API
- Offline backup of HDFS data

表3-1分别从操作对集群的性能影响、数据空间消耗、业务中断影响、增量备份、易用性、可恢复性几个维度进行对比。

表 3-1 HBase 上数据备份对比

备份方式	性能影响	数据空间消耗	业务中断影响	增量备份	易用性	可恢复性
Snapshots	低	小	短暂中断（仅恢复时）	不支持	易	秒
Replication	低	大	无	固有	中	秒
Export	高	大	无	支持	易	高
CopyTable	高	大	无	支持	易	高
HTable API	中	大	无	支持	难	取决于用户
Offline backup of HDFS data	-	大	长时间中断	不支持	中	高

Snapshots

对表执行snapshot操作生成快照，既可以作为原表的备份，当原表出现问题的时候可以回滚恢复，也可以作为跨集群的数据备份工具。执行快照会在当前HBase上的根目录（默认为/hbase）生成“.hbase-snapshot”目录，里面有每个快照的详细信息。当执行ExportSnapshot导出快照时，会在本地提交MR任务，将快照信息以及表的HFile分别拷贝到备集群的/hbase/.hbase-snapshot和/hbase/archive中。详情请参考**http://hbase.apache.org/2.2/book.html#ops.snapshots**。

- 该方式数据备份的优点：

单表备份效率高，在线数据本地/远程备份，不中断主集群和备集群业务，可以灵活配置map的个数和限制流量，MapReduce的执行节点可不在主备集群（不占资源）。

- 该方式数据备份的缺点和限制：

  只能单表操作，备份的表名在snapshot中已经指定无法更改，且无法增量备份，运行MR需要占用本地集群资源。

在主集群执行如下操作：

步骤1 对表创建快照。例如对表member创建快照member_snapshot。

snapshot 'member','member_snapshot'

步骤2 将快照拷贝到备集群上。

hbase org.apache.hadoop.hbase.snapshot.ExportSnapshot -snapshot *member_snapshot* -copy-to hdfs://*备集群HDFS服务主NameNode节点IP:端口号*/hbase -mappers 3

- 备集群的数据目录必须为HBASE根目录（/hbase）
- mappers表示MR任务需要提交的map个数

----结束

在备集群执行如下操作：

使用restore命令在备集群自动新建表，以及与archive里的HFile建立link。

restore_snapshot 'member_snapshot'

说明

如果只是备份表数据的话，建议使用此种方式备份，SnapshotExport会在本地提交MR任务，将Snapshot和HFile拷贝到备集群，之后可以在备集群直接加载数据，效率比其他方式高很多。

Replication

Replication备份是在HBase上建立主备集群的容灾关系，当数据写入主集群，主集群通过WAL来主动push数据到备集群上，从而达到主备集群的实时同步。详情请参考**http://hbase.apache.org/2.2/book.html#_cluster_replication**。

- 该方式数据备份的优点：
  - 使用replication有别于其他几种数据备份导入方式，当配置了集群间的主备关系后，数据可以实时同步（无需人为操作）。
  - 相对而言，“备份”的动作占用集群的资源较少，对集群的性能影响小。
  - 数据同步可靠性较高，如果备集群停止一段时间后再恢复，这中间主机群的数据依然会同步到备集群。
- 该方式数据备份的缺点和限制：
  - 如果客户端写入的数据设置不写WAL，则数据无法备份到备集群。
  - 由于占用的资源少，后台是通过异步的方式同步数据，实际数据没有实时同步。
  - 对于开启表replication同步之前，主集群就已经存在的数据无法同步，需要借助其他方式导入的备集群。

- bulkload方式写入到主集群的数据无法同步（MRS上的HBase对replication做了增强，支持bulkload on replication）。

具体的使用和配置方法请参考**配置HBase备份**和**使用ReplicationSyncUp工具**来进行备份数据。

Export/Import

Export/Import主要是启动MapReduce任务对数据的表进行扫描（scan），往远端HDFS写入SequenceFile，之后Import再把SequenceFile读出来写入HBase（put）中。

- 该方式数据备份的优点：

  在线拷贝不中断业务，由于是scan->put的方式写入新表，所以比CopyTable更加灵活，可灵活配置需要获取的数据，数据可增量写入。

- 该方式数据备份的缺点和限制：

  由于Export是通过MapReduce任务往远端HDFS写入SequenceFile，之后Import再把SequenceFile读出来写入HBase，需要执行两次MapReduce任务，实际效率不高。

在主集群执行如下操作：

执行Export命令导出表。

hbase org.apache.hadoop.hbase.mapreduce.Export <tablename> <outputdir>

例如：**hbase org.apache.hadoop.hbase.mapreduce.Export member hdfs://*备集群HDFS服务主NameNode节点IP:端口号*/user/table/member**

其中，member为待导出表的名称。

在备集群执行如下操作：

步骤1 主集群执行完之后可以在备集群上查看生成的目录数据如**图3-1**。

图 3-1 目录数据

```
Cv1:~ # hdfs dfs -ls -R /user/table/member
-rw-r--r--   3 root hadoop          0 2018-06-28 14:18 /user/table/member/_SUCCESS
-rw-r--r--   3 root hadoop       2937 2018-06-28 14:18 /user/table/member/part-m-00000
```

步骤2 执行**create**命令在备集群上新建与主集群相同结构的表，例如member_import。

步骤3 执行Import命令生成HFile数据在HDFS上。

hbase org.apache.hadoop.hbase.mapreduce.Import <tablename> <inputdir>

例如：**hbase org.apache.hadoop.hbase.mapreduce.Import member_import /user/table/member -Dimport.bulk.output=/tmp/member**

- member_import为备集群上与主集群相同表结构的表
- Dimport.bulk.output为输出的HFile数据目录
- /user/table/member为从主集群上导出的数据目录

步骤4 执行Load操作将HFile数据写入HBase。

hbase org.apache.hadoop.hbase.mapreduce.LoadIncrementalHFiles /tmp/member member

- /tmp/member为**步骤3**中HFile数据的输出目录
- member为备集群上要导入数据的表名

----结束

CopyTable

拷贝表功能与导出功能类似，拷贝表也使用HBase API创建了一个MapReduce任务，以便从源表读取数据。不同的地方是拷贝表的输出是hbase中的另一张表，这张表可以在本地集群，也可以在远程集群。详情请参考**http://hbase.apache.org/2.2/book.html#copy.table**。

- 该方式数据备份的优点：

  操作简单，在线拷贝不中断业务，可以指定备份数据的startrow/endrow/timestamp。
- 该方式数据备份的缺点和限制：

  只能单表操作，远程拷贝数据量大时效率较低，MapReduce需要占用本地资源，MapReduce的map个数以表region的个数划分。

在备集群执行如下操作：

执行**create**命令在备集群上新建与主集群相同结构的表，例如member_copy。

在主集群执行如下操作：

执行CopyTable的命令拷贝表。

hbase org.apache.hadoop.hbase.mapreduce.CopyTable [--starttime=xxxxxx] [--endtime=xxxxxx] --new.name=member_copy --peer.adr=server1,server2,server3:2181:/hbase [--families=myOldCf:myNewCf,cf2,cf3] TestTable

- starttime/endtime为待拷贝数据的时间戳。
- new.name为备集群中目的表的表名，默认为和原来表名相同。
- peer.adr为备集群zookeeper节点的信息，格式为quorumer:port:/hbase。
- families为待拷贝的表的family列。

说明

如果是拷贝数据到远端集群，此种方式导入数据会在主机群上提交MapReduce任务，读取原始表的全量/部分数据之后采用put的方式写入远端集群，所以如果表的数据量很大（远程拷贝不支持bulkload），则效率会比较低。

HTable API

API的方式主要也是在代码中对原始HBase表的数据导入导出，你可以使用公用的API写自己定制的客户端应用程序来直接查询表格，或通过MapReduce任务的批量处理优势自己设计其他方法。该方式需要对Hadoop开发以及因此对生产集群带来的影响有深入的理解。应用程序开发详情请参考**MRS开发指南**。

Offline backup of HDFS data

离线备份HDFS数据，即关闭HBase服务并手工在HDFS上拷贝数据。

- 该方式数据备份的优点：
  - 可以把主集群上所有数据（包含元数据）整个复制到备集群。
  - 由于是通过Distcp直接拷贝的，所以数据备份的效率相对较高。
  - 实际操作时可以根据具体的需求灵活拷贝，可以只拷贝其中一个表的数据，也可以拷贝region中的其中一个HFile等。
- 该方式数据备份的缺点和限制：
  - 此操作会覆盖备集群上的HDFS的数据目录。
  - 如果主备集群间的HBase版本不同，HDFS目录直接拷贝可能会出现问题，例如MRS上的hbase1.3版本新增了系统表index，如果使用老版本的HDFS目录直接覆盖，会找不到该数据表。所以此种方案在执行前需要慎重考虑。
  - 此操作对用户使用HBase的能力有一定的要求，如出现异常情况需要根据实际情况执行恢复。

在主集群执行如下操作：

步骤1 执行如下命令将当前集群内存中的数据持久化到HDFS中。

flush 'tableName'

步骤2 停止HBase服务。

步骤3 使用distcp命令拷贝当前集群HDFS上的数据到备集群上。

hadoop distcp -i /hbase/data hdfs://*备集群HDFS服务主NameNode节点IP:端口号*/hbase

hadoop distcp -update -append -delete /hbase/ hdfs://*备集群HDFS服务主NameNode节点IP:端口号*/hbase/

第二条命令为增量拷贝除了data目录以外的文件，例如archive里面的数据可能还有被数据目录所引用。

----结束

在备集群执行如下操作：

步骤1 重启HBase服务，使数据迁移生效。在启动过程中，HBase会加载当前HDFS上的数据并重新生成元数据。

步骤2 启动完成后，在Master节点客户端执行如下命令加载HBase表数据。

```
$HBase_Home/bin/hbase hbck -fixMeta -fixAssignments
```

步骤3 命令执行完成后，重复执行如下命令查看HBase集群健康状态直至正常。

```
hbase hbck
```

说明

当用户使用了HBase协处理器，自定义jar包放在主集群的regionserver/hmaster上时，在备集群重启HBase之前，需要把这些自定义jar包也拷贝过来。

----结束

3.5 Kafka 数据

当需要在两个Kafka集群间作数据同步，或者将原有Kafka集群上的数据搬迁到新的Kafka集群上时就需要用到Kafka数据同步的利器——MirrorMaker。MirrorMaker是

Kafka内嵌的一个工具，其内部实际上是集成了Kafka的Consumer和Producer，它可以从一个Kafka集群消费数据然后写入另一个Kafka集群，从而实现Kafka集群间的数据同步。

本章节介绍利用MRS服务提供的MirroMaker工具实现Kafka集群数据同步、迁移的方法，请先参考**打通数据传输通道**完成两个集群的网络互通后再参考本章节操作Kafka数据迁移。

操作步骤

3.x之前版本：

步骤1 配置集群Kerberos互信。

步骤2 若计划在源集群使用MirrorMaker工具，请登录目的集群的集群详情页面，选择“组件管理”。若计划在目的集群使用MirrorMaker工具，请登录源集群的集群详情页面，选择“组件管理”。

步骤3 选择“Kafka > 服务配置”，将页面右侧的“基础配置”切换为“全部配置”。

步骤4 选择“Broker > 自定义”配置项增加如下规则。

sasl.kerberos.principal.to.local.rules = RULE:[1:$1@$0](.*@XXXYYYZZZ.COM)s/@.*//,RULE:[2:$1@$0](.*@XXXYYYZZZ.COM)s/@.*//,DEFAULT

其中：XXXYYYZZZ.COM为数据发送端集群（源集群）的域名(字母需大写)。

图 3-2 增加规则

步骤5 单击“保存配置”，按照提示选择“重新启动受影响的服务或实例。”并单击“确定”，重启Kafka服务。

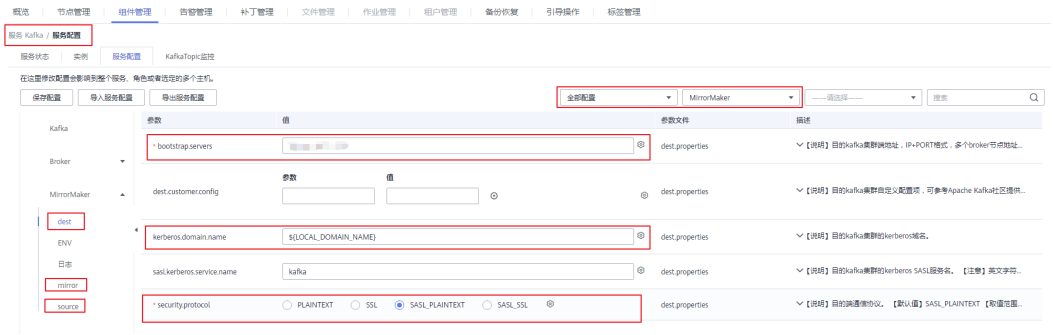
说明

开启Kerberos认证的安全集群需要执行步骤**步骤1**-**步骤5**，未开启Kerberos认证的普通集群请跳过步骤**步骤1**-**步骤5**，直接从**步骤6**开始执行。

步骤6 在计划使用MirrorMaker工具的集群，登录集群详情页面，选择“组件管理”。

步骤7 选择“Kafka > 服务配置”，切换“基础配置”为“全部配置”，“角色”为“MirrorMaker”。

图 3-3 配置 Kafka 服务的 MirrorMaker 参数



配置参数说明：

- source和dest标签中的bootstrap.servers参数分别是源Kafka集群和目的Kafka集群的broker节点列表及端口信息
- source和dest标签中的security.protocol参数请根据源Kafka集群和目的Kafka集群的服务端的实际配置情况自行选择
- 如果源Kafka集群（source）或者目的Kafka集群（dest）为安全集群则需要配置source和dest标签中的kerberos.domain.name（如果是本机可不配置，非本机根据实际值进行配置）和sasl.kerberos.service.name（默认：kafka）参数
- 在mirror标签中设置whitelist，即需要同步的topic名称。

步骤8 单击“保存配置”，按照提示选择“重新启动受影响的服务或实例。”并单击“是”，重启MirrorMaker实例。

待MirrorMaker重启完成后，数据迁移任务即已启动，可通过**KafkaManager**或者命令行工具监控MirrorMaker数据同步进度。

----结束

4 系统对接

4.1 MRS 对接 LakeFormation

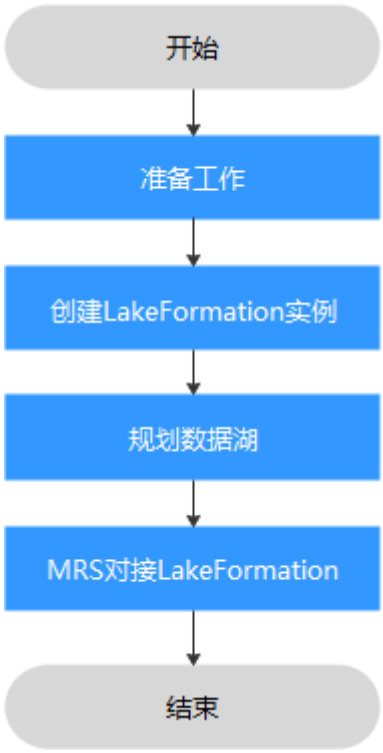
4.1.1 概述

该LakeFormation入门教程介绍了如何创建一个LakeFormation实例并与MRS集群对接，实现统一的数据湖元数据及权限管理。

使用流程简介

MRS与LakeFormation对接的使用流程如下图所示：

图 4-1 LakeFormation 使用流程



约束说明

- **MRS对接LakeFormation前，需要注意以下约束限制：**
  - MRS集群和LakeFormation实例必须同在一个云账户下且属于同一个Region。
  - LakeFormation侧创建的接入客户端所在虚拟私有云，必须与MRS集群在同一虚拟私有云下。
  - MRS集群仅支持对接LakeFormation实例中名称为hive的Catalog。
  - MRS存量集群需要先完成元数据库和权限策略向LakeFormation实例上迁移，再配置对接。
  - 如果需要迁移多个MRS集群中的元数据到同一个LakeFormation实例，MRS集群之间的Database名称不能重复。
- **MRS对接LakeFormation后，MRS组件功能约束限制：**
  - Hive暂不支持临时表功能。
  - Hive暂不支持跨集群的列加密表功能。
  - Hive WebHCat暂不支持对接LakeFormation。
  - Hive创建内表时如果表目录不为空，则禁止创建表。
  - Hudi表创建前，需要先在LakeFormation上添加Hudi表目录的路径授权，赋予OBS读写权限。
  - Hudi表不支持在LakeFormation管理面编辑表的字段，只能通过Hudi客户端增删改表的字段。
  - Flink读写Hudi场景下同步Hive表，仅支持使用hive_sync.mode=jdbc，不支持hms方式。
  - Spark使用小权限用户登录客户端创建数据库时，如果用户没有default库的OBS路径权限，将提示缺少权限，实际创建数据库成功。
- **MRS对接LakeFormation后，权限策略约束限制：**
  - 通过LakeFormation授权仅支持将LakeFormation角色作为授权主体，不支持IAM用户或IAM用户组作为授权主体。
  - PolicySync进程不会修改集群内RangerAdmin Hive模块的默认策略，默认策略仍然生效。
  - PolicySync进程启动后，会与LakeFormation实例的权限进行比对，删除LakeFormation上不存在的非默认策略，请先完成权限策略迁移到LakeFormation实例上。
  - RangerAdmin WebUI界面的Hive模块，禁止执行添加、删除权限非默认策略的操作，统一在LakeFormation实例的数据权限界面进行授权操作。
  - MRS集群取消对接LakeFormation后，RangerAdmin的非默认策略不会清理，需要人工进行清理。
  - Hive暂不支持Grant授权的SQL语句，需统一在LakeFormation实例的数据权限界面进行授权操作。
  - MRS暂不支持LakeFormation行过滤权限能力。

4.1.2 准备工作

配置 LakeFormation 实例

步骤1 登录华为云管理控制台，在左上角单击“  ”，选择“大数据 > 湖仓构建 LakeFormation”进入LakeFormation控制台。

步骤2 单击页面右上角“购买实例”，参考**创建LakeFormation实例**创建LakeFormation实例。

步骤3 创建名称为“hive”的Catalog、名称为“default”的数据库，如果实例中已存在则请跳过该步骤。详细操作可参考**管理元数据**。

MRS对接LakeFormation仅支持对接LakeFormation实例的数据目录名称为hive的Catalog。

1. 确认左上角实例是新创建的LakeFormation实例名称后，进入“元数据 > Catalog”页面。
2. 单击“创建Catalog”，名称填“hive”（**固定名称，不可自定义**），不选择存储路径单击“提交”。
   - Catalog名称：hive（**固定名称，不可自定义**）
   - 选择位置：选择Catalog对应的OBS存储目录，需提前创建，例如选择“obs://lakeformation-test/hive”，单击“确定”。
3. 单击已创建的Catalog名称“hive”进入数据库页面，单击“创建数据库”，配置以下信息并单击“提交”：
   - 库名称：default（**固定名称，不可自定义**）
   - 选择位置：需要选择hive Catalog存储路径下的位置，例如“obs://lakeformation-test/catalog1/database1”。

步骤4 在“数据权限 > 数据授权”页面，可根据业务需求对hive数据目录进行基于用户、用户组的授权。详细操作请参考**新增授权**章节。

步骤5 单击“接入管理 > 创建客户端”，创建LakeFormation实例接入管理客户端。其中“虚拟私有云”和“所属子网”需要与待对接的MRS集群保持一致。详细操作请参考**管理接入客户端**。

MRS集群的VPC子网信息可通过MRS管理控制台中，MRS集群的概览页面中获取。

客户端创建完成后，在客户端详情信息中获取对应客户端的“接入IP”信息并记录。

----结束

创建对接 LakeFormation 权限的委托

步骤1 登录华为云管理控制台，选择“统一身份认证服务”。

步骤2 在左侧导航栏选择“委托”，单击右上角的“创建委托”，配置相关参数，单击“下一步”。

参数配置如下：

- 委托名称：例如“visit_lakeformation_agency”
- 委托类型：选择“普通账号”

- 委托的账号：输入被委托的华为云账号名称
- 持续时间：根据实际情况自定义

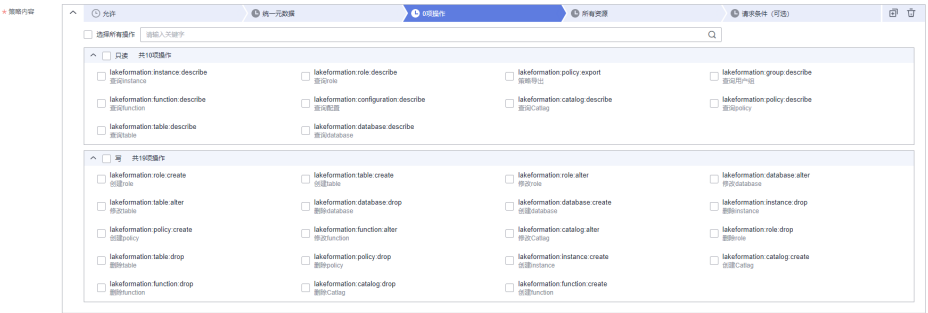
图 4-2 创建委托

步骤3 在选择策略界面右上角单击“新建策略”，配置如下信息，单击“下一步”。

- 策略名称：例如“dev_visit_lakeformation”
- 策略配置方式：“可视化视图”或“JSON视图”
- 策略内容：

  策略中必须包含“lakeformation:policy:export”和“lakeformation:role:describe”。其他参数按照实际需求进行配置。

  - 可视化视图：“云服务”选择“湖仓构建”；“操作”中选择所需操作权限。其他参数按照实际需求进行配置。

  - JSON视图，例如配置策略内容如下：

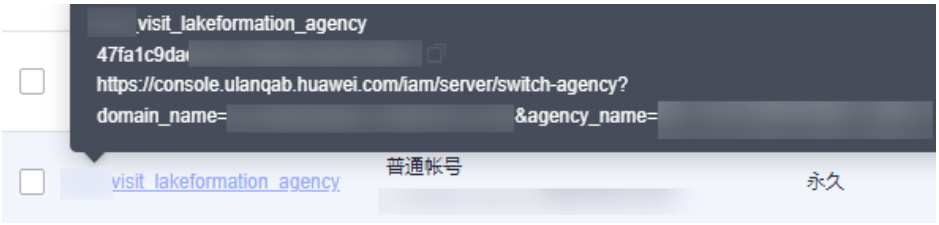
```
{
    "Version": "1.1",
    "Statement": [
        {
```

```
            "Effect": "Allow",
            "Action": [
                "lakeformation:table:create",
                "lakeformation:database:alter",
                "lakeformation:table:alter",
                "lakeformation:database:drop",
                "lakeformation:database:create",
                "lakeformation:role:describe",
                "lakeformation:policy:create",
                "lakeformation:policy:export",
                "lakeformation:function:alter",
                "lakeformation:function:describe",
                "lakeformation:table:drop",
                "lakeformation:catalog:describe",
                "lakeformation:table:describe",
                "lakeformation:function:drop",
                "lakeformation:database:describe",
                "lakeformation:function:create",
                "lakeformation:transaction:operate"
            ]
        }
    ]
}
```

步骤4 勾选新建的策略名称例如“dev_visit_lakeformation”，单击“下一步”。

步骤5 “设置最小授权范围”根据实际情况选择授权的资源范围，单击“确定”，创建委托。

步骤6 在“委托”页面，将鼠标放到新创建的委托名称上，获取具备访问LakeFormation权限的委托ID。



----结束

创建对接 OBS 权限的委托

步骤1 登录华为云管理控制台，选择“统一身份认证服务”。

步骤2 在左侧导航栏选择“委托”，单击右上角的“创建委托”，选择相关参数，单击“下一步”。

参数选择如下：

- 委托名称：例如“visit_obs_agency”
- 委托类型：选择“普通账号”
- 委托的账号：输入被委托的华为云账号名称
- 持续时间：根据实际情况自定义

步骤3 在选择策略界面右上角单击“新建策略”，配置如下信息，单击“下一步”。

- 策略名称：例如“dev_visit_obs”
- 策略配置方式：JSON视图

- 策略内容：填入如下信息。

```
{
   "Version": "1.1",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "obs:bucket:GetBucketLocation",
            "obs:bucket:ListBucketMultipartUploads",
            "obs:object:GetObject",
            "obs:object:ModifyObjectMetaData",
            "obs:object:DeleteObject",
            "obs:object:ListMultipartUploadParts",
            "obs:bucket:HeadBucket",
            "obs:object:AbortMultipartUpload",
            "obs:bucket:ListBucket",
            "obs:object:PutObject"
         ],
         "Resource": [
            "OBS:*:*:bucket:*",
            "OBS:*:*:object:*"
         ]
      }
   ]
}
```

说明

Resource参数中“bucket”的参数值表示OBS桶名称，“object”的参数值表示OBS对象名称，可根据需要指定名称。配置为“*”表示对所有OBS桶或OBS对象适用此策略。

- 其他参数按照实际需求进行配置。

步骤4 勾选新建的策略名称例如“dev_visit_obs”，单击“下一步”。

步骤5 “设置最小授权范围”根据实际情况选择授权的资源范围，单击“确定”，创建委托。

步骤6 在“委托”页面，将鼠标放到新创建的委托名称上，获取具备访问OBS权限的委托ID。

----结束

创建对接 ECS/BMS 云服务委托

步骤1 登录华为云管理控制台，选择“统一身份认证服务”。

步骤2 在左侧导航栏选择“委托”，单击右上角的“创建委托”，设置相关参数，单击“下一步”。

参数选择如下：

- 委托名称：例如“lakeformation_test”
- 委托类型：选择“云服务”
- 云服务：选择“ECS BMS”
- 持续时间：根据实际情况自定义

步骤3 在选择策略界面右上角单击“新建策略”，配置如下信息，单击“下一步”。

- 策略名称：自定义
- 策略配置方式：选择JSON视图

- 策略内容：配置如下信息

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Action": [
                "iam:agencies:assume"
            ],
            "Resource": {
                "uri": [
                    "/iam/agencies/授予给自身账号具备访问LakeFormation权限的委托ID",
                    "/iam/agencies/授予给自身账号具备访问OBS权限的委托ID"
                ]
            },
            "Effect": "Allow"
        }
    ]
}
```

说明

- 授予给自身账号具备访问LakeFormation权限的委托ID：可参考**步骤6**获取。
- 授予给自身账号具备访问OBS权限的委托ID：可参考**步骤6**获取。

步骤4 选择新创建的自定义委托名称，单击“下一步”。

步骤5 “设置最小授权范围”根据实际情况选择授权的资源范围，单击“确定”，创建委托完成。

----结束

创建 LakeFormation 数据连接

步骤1 登录MRS控制台，在导航栏选择“数据连接”。

步骤2 单击“新建数据连接”。

步骤3 参考**表4-1**配置相关参数，单击“确定”完成创建。

表 4-1 配置 LakeFormation 数据连接

参数	说明
类型	选择“LakeFormation”，仅MRS 3.3.0-LTS版本支持连接该类型。
名称	数据连接的名称。
LakeFormation实例	选择LakeFormation实例名称。 该实例需要先在LakeFormation实例创建后在此处引用，具体请参考**创建LakeFormation实例**。单击“查看LakeFormation实例”查看已创建的实例。
虚拟私有云	需要与待对接的MRS集群在同一虚拟私有云。
子网	选择子网名称。
VPC终端节点	选择VPC终端节点，或单击“创建对应LakeFormation实例的VPC终端节点”进行创建。 选择VPC终端节点后，产生的费用将由VPCEP服务收取。

参数	说明
LakeFormation委托	选择“现有委托”，并选择**创建对接LakeFormation权限的委托**创建的委托，例如“visit_lakeformation_agency”。

图 4-3 新建 LakeFormation 数据连接

步骤4 创建完成后，在“数据连接”页面记录已创建数据连接的ID。

----结束

获取账号 ID 信息

步骤1 登录管理控制台。

步骤2 单击用户名，在下拉列表中单击“我的凭证”。

步骤3 在“API凭证”页面获取“账号ID”、项目列表中查看项目ID。

----结束

4.1.3 创建集群时配置 LakeFormation 数据连接

该章节指导用户在创建MRS 3.3.0-LTS集群时配置LakeFormation数据连接，并在创建完成后配置MRS集群相关参数完成与LakeFormation的对接。

创建集群时配置 LakeFormation 数据连接

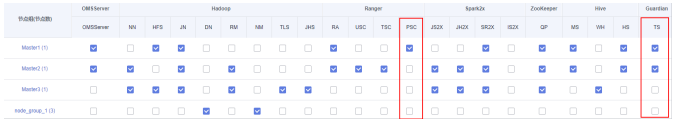
步骤1 进入**购买MRS集群页面**。

步骤2 单击“购买集群”，进入“购买集群”页面。

步骤3 在购买集群页面，选择“自定义购买”。

步骤4 参考**购买自定义拓扑集群**进行配置并创建集群，且集群需满足**表4-2**中要求。

表 4-2 LakeFormation 数据连接参数说明

参数	参数说明
版本类型	LTS版
集群版本	MRS 3.3.0-LTS 当前仅MRS 3.3.0-LTS版本支持在创建集群时配置LakeFormation数据连接。
组件选择	必须包含Hadoop、Ranger、Hive、Guardian、Spark（可选）、Flink（可选）等组件。
元数据	选择“外置数据连接”，并配置以下参数： 1. LakeFormation元数据：单击按钮开启。 2. LakeFormation连接实例：选择**创建LakeFormation数据连接**已创建的LakeFormation数据连接名称。 3. 数据连接类型：保持默认。
虚拟私有云	与LakeFormation数据连接所在的虚拟私有云保持一致。
子网	选择子网名称。
拓扑调整	选择“开启”，并确认Ranger组件至少添加1个PolicySync（PSC）实例（该实例部署节点需要同时包含RangerAdmin实例）、Guardian组件至少添加2个TokenSever（TS）实例。 
Kerberos认证	开启
委托	选择“现有委托”，并选择**创建对接ECS/BMS云服务委托**创建的委托。

步骤5 参考**配置MRS 3.3.0-LTS版本集群**配置组件存算分离、下载客户端等操作。

----结束

配置 MRS 3.3.0-LTS 版本集群

步骤1 登录MRS集群的FusionInsight Manager页面，具体操作请参考**访问FusionInsight Manager（MRS 3.x及之后版本）**。

步骤2 配置Guardian。

1. 在FusionInsight Manager界面，选择“集群 > 服务 > Guardian > 配置 > 全部配置”，搜索并修改以下参数后，单击“保存”。

表 4-3 配置 Guardian 参数

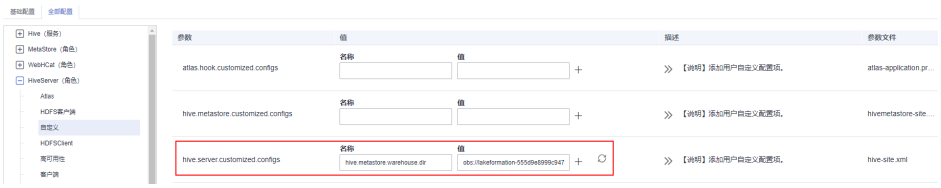
参数	含义	取值
token.server.access.iam.domain.id	访问IAM的用户对应的账号ID。 参考**获取账号ID信息**获取账号ID信息。	xxx
token.server.access.iam.project.id	访问IAM的用户对应的项目ID。 参考**获取账号ID信息**获取项目ID信息。	xxx
token.server.access.label.agency.name	指定IAM委托的名字，需要具有访问OBS的权限。 即**创建对接OBS权限的委托**创建的委托名称。	visit_obs_agency
fs.obs.delegation.token.providers	delegation.token的产生类名，默认为空。	com.huawei.mrs.dt.MRSDelegationTokenProvider,com.huawei.mrs.dt.GuardianDTProvider
fs.obs.guardian.accesslabel.enabled	是否开启使用Guardian对接OBS的access label。	true
fs.obs.guardian.enabled	是否开启使用Guardian。	true

2. 进入Guardian服务“概览”页面，选择“更多 > 重启服务”。

步骤3 配置Hive对接OBS文件系统。

1. 在FusionInsight Manager界面，选择“集群 > 服务 > Hive > 配置 > 全部配置”。
2. 在左侧的导航列表中选择“HiveServer > 自定义”。在自定义配置项中，给参数“hive.server.customized.configs”添加配置项“hive.metastore.warehouse.dir”，设置值为**配置LakeFormation实例**章节获取的hive数据目录在OBS中的存储路径。

图 4-4 hive.metastore.warehouse.dir 配置



3. 单击“保存”，保存配置。

步骤4 配置Spark对接OBS文件系统。如果集群不存在Spark组件请跳过该步骤。

1. 在FusionInsight Manager界面，选择“集群 > 服务 > Spark > 配置 > 全部配置”。
2. 在左侧的导航列表中选择“JDBCServer > 自定义”，参考下表增加自定义参数及值。

表 4-4 Spark 参数配置

自定义参数	参数值
custom	– 参数：spark.sql.warehouse.location.first – 值：true
spark.hive-site.customized.configs	– 参数：hive.metastore.warehouse.dir – 值：设置为**配置LakeFormation实例**章节获取的hive数据目录在OBS中的存储路径。

3. 在左侧的导航列表中选择“SparkResource > 自定义”，参考**表4-4**配置参数。
4. 单击“保存”，保存配置。

步骤5 在MRS集群“组件管理”页签，查看是否存在“配置超期”的组件，如果存在请单击“操作”列的“重启”，重启相关组件。

步骤6 重新下载并安装MRS集群完整客户端。具体操作请参考**安装客户端**。

步骤7 如果需要在管理控制台执行作业提交操作，需要更新集群内置客户端配置文件。

在MRS集群概览页面，获取弹性IP，使用该IP登录Master节点，执行如下命令刷新集群内置客户端。

su - omm

sh /opt/executor/bin/refresh-client-config.sh

步骤8 登录客户端安装节点，通过Hive客户端查看数据库，确认对接成功。

source *客户端安装路径*/**bigdata_env**

kinit *组件业务用户*

beeline

show databases;desc database default;

!q

步骤9 通过Spark客户端，查看数据库，确认对接成功。如果集群不存在Spark组件请跳过该步骤。

source *客户端安装路径*/**Spark/component_env**

spark-sql

show databases;desc database default;

----结束

4.1.4 通过 Ranger 为 MRS 集群内用户绑定 LakeFormation 角色

LakeFormation与MRS集群对接后，可以在Ranger WebUI界面为MRS集群内的用户或用户组绑定LakeFormation角色，则绑定的用户或用户组拥有该角色的权限。

前提条件

- 已创建角色，并为该角色添加权限，具体操作请参考**角色授权**。
- 已完成MRS与LakeFormation实例的对接。

操作步骤

步骤1 登录MRS管理控制台，选择“现有集群”，单击待操作的集群名称。

步骤2 单击“IAM用户同步”后的“同步”，等待同步成功。

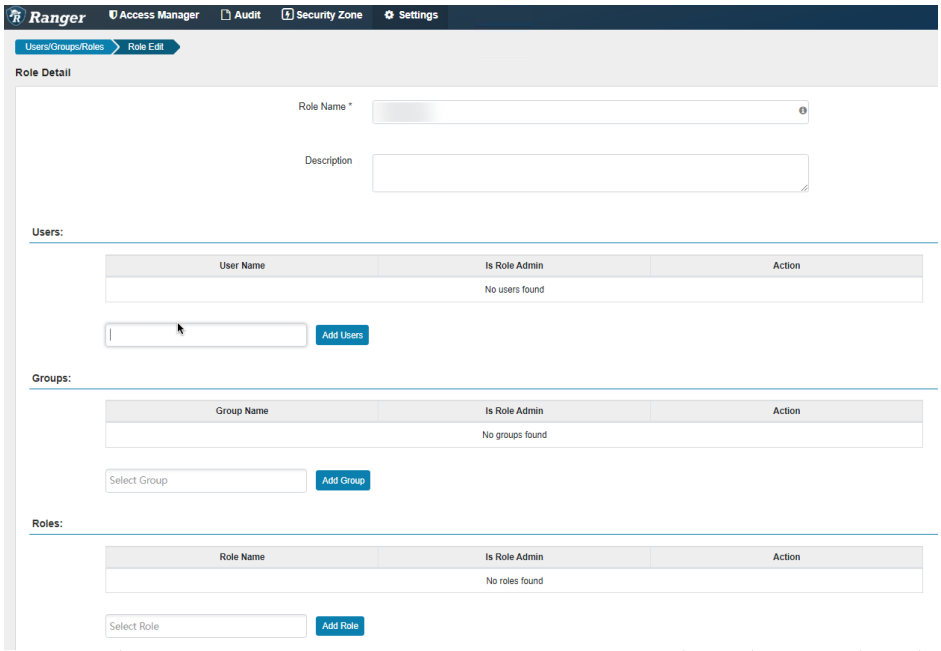
步骤3 以**admin**用户登录MRS集群的FusionInsight Manager页面，具体操作请参考**访问FusionInsight Manager（MRS 3.x及之后版本）**。

步骤4 以**rangeradmin**用户登录Ranger WebUI界面。

1. 选择“集群 > 服务 > Ranger”，进入Ranger服务概览页面。
2. 单击“基本信息”区域中的“RangerAdmin”，进入Ranger WebUI界面。
3. 在Ranger WebUI界面，单击右上角用户名，选择“Log Out”，退出当前用户。
4. 使用**rangeradmin**用户或者其他具有Ranger管理员权限用户重新登录。**rangeradmin**用户默认密码请参考**用户账号一览表**获取。

步骤5 在Ranger WebUI界面选择“Settings > Roles”。

步骤6 单击已创建的LakeFormation角色名称，进入“Role Edit”界面。



步骤7 分别在“Users”区域单击“Select User”、“Groups”区域单击“Select Group”配置拥有该角色权限的用户或者用户组。

- Users：MRS集群系统中已创建的用户。

- Groups：MRS集群系统中已创建的用户组。

步骤8 单击“Save”保存配置。

配置完成后，所选择的用户和用户组即拥有该角色的权限。

----结束

4.2 使用 DBeaver 访问 Phoenix

本章节以DBeaver 6.3.5版本为例，讲解如何访问MRS 3.1.0未开启Kerberos认证的集群，且该集群的HBase服务未开启Ranger鉴权。

前提条件

- 已安装DBeaver 6.3.5，DBeaver软件下载链接为：**https://dbeaver.io/files/6.3.5/dbeaver-ce-6.3.5-x86_64-setup.exe**。
- 已创建包含HBase组件的MRS 3.1.0未开启Kerberos认证的集群。
- 已安装HBase客户端。
- 已安装JDK 1.8.0_*x*。

操作步骤

步骤1 向DBeaver安装目录下的“dbeaver.ini”文件中增加JDK 1.8.0_*x*的bin目录，例如：C:\Program Files\Java\jdk1.8.0_121\bin，则新增如下内容：

图 4-5 新增 JDK 的 bin 目录

```
-vm
C:\Program Files\Java\jdk1.8.0_121\bin
```

步骤2 在**https://archive.apache.org/dist/phoenix/apache-phoenix-5.0.0-HBase-2.0/bin/apache-phoenix-5.0.0-HBase-2.0-bin.tar.gz**中下载Phoenix软件包并解压，获取“phoenix-5.0.0-HBase-2.0-client.jar”。

步骤3 从客户端安装节点上的“*客户端安装目录*/HBase/hbase/conf”目录下下载“hbase-site.xml”文件。使用压缩软件打开**步骤2**获取到的“phoenix-5.0.0-HBase-2.0-client.jar”，将“hbase-site.xml”拖拽到该Jar文件中。

步骤4 打开DBeaver，选择“文件 > 新建 > DBeaver > 数据库连接”。

图 4-6 创建数据库连接

步骤5 单击“下一步”，在选择新连接类型界面选中“Apache Phoenix”并单击“下一步”。

图 4-7 选择数据连接类型

步骤6 单击“编辑驱动设置”。

图 4-8 编辑驱动设置

步骤7 单击“添加文件”，选择准备好的“phoenix-5.0.0-HBase-2.0-client.jar”，如果有多个驱动包，需先删除，只保留手动添加的“phoenix-5.0.0-HBase-2.0-client.jar”。

图 4-9 删除原有的驱动包

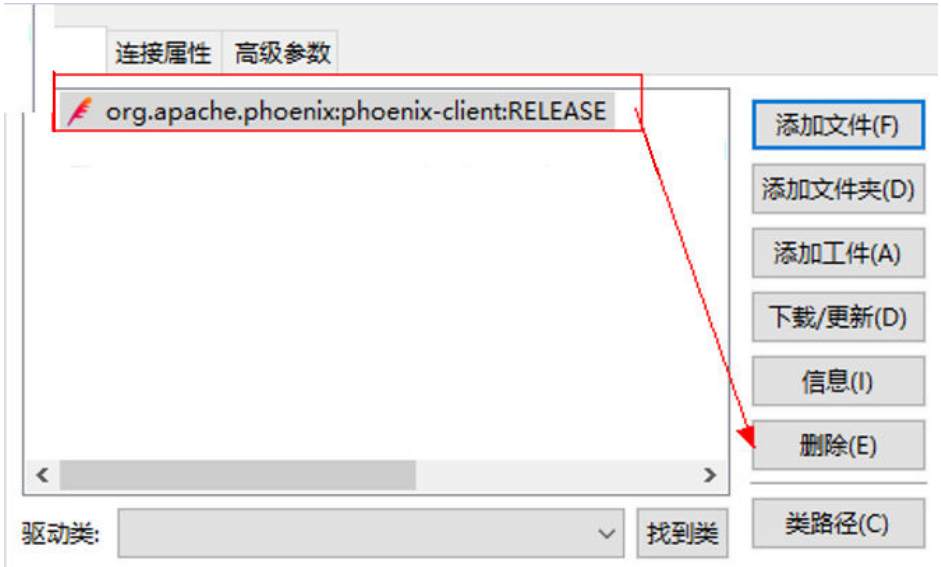
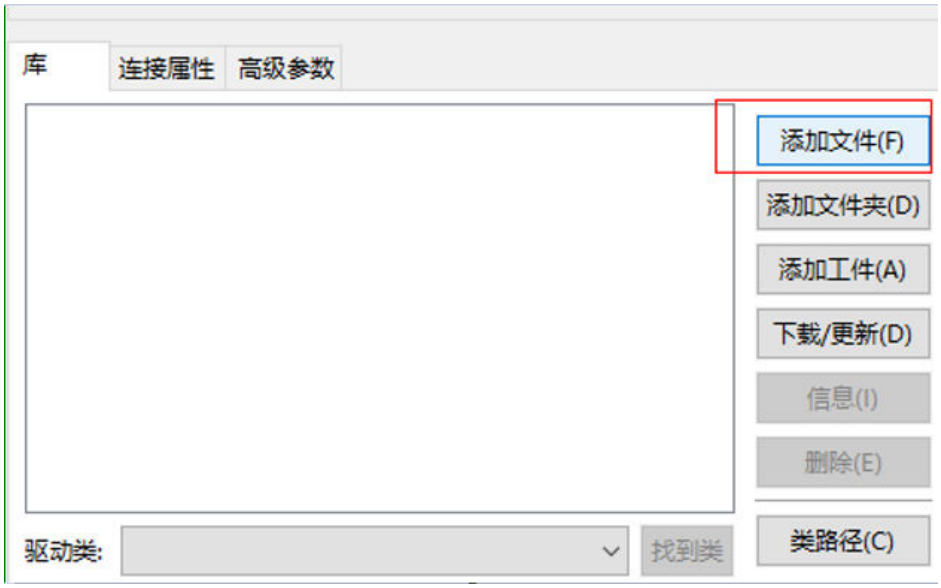
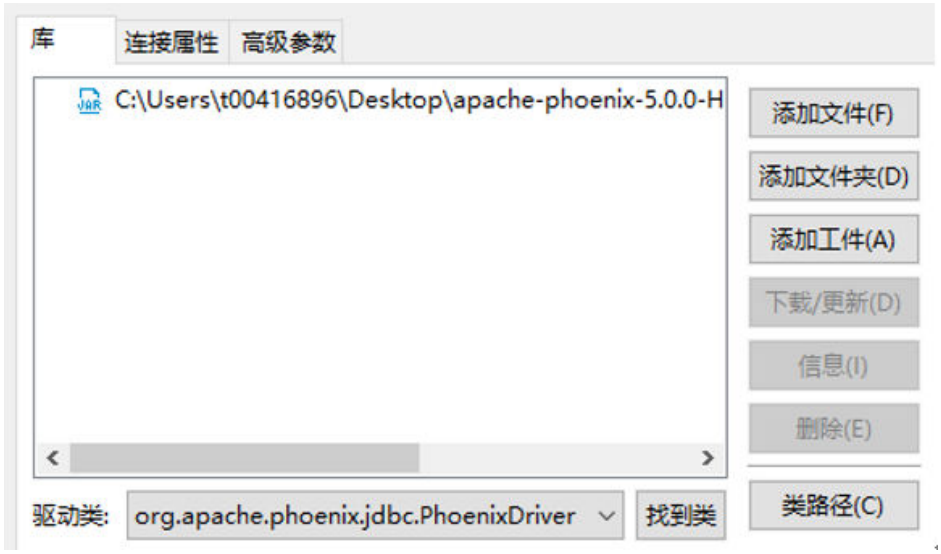


图 4-10 添加 Phoenix Jar 包



步骤8 单击“找到类”，加载完成后在左侧框中选择“org.apache.phoenix.jdbc.PhoenixDriver”。

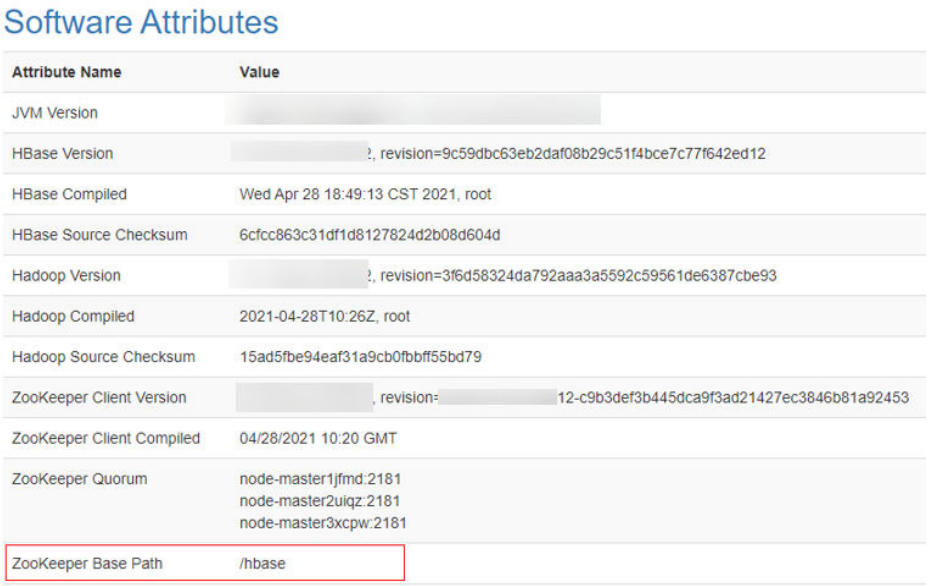
图 4-11 加载驱动类



步骤9 添加Zookeeper Base Path。

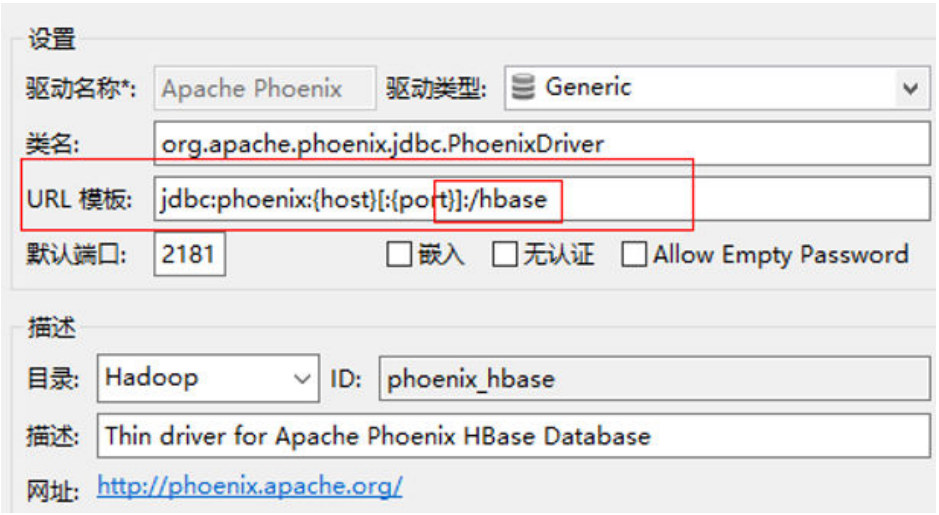
1. 登录FusionInsight Manager，选择“集群 > 服务 > HBase”，单击“HMaster WebUI”右侧的超链接，登录HBase WebUI界面，搜索“Zookeeper Base Path”，并获取该值，如下图所示“Zookeeper Base Path”的值为“/hbase”：

图 4-12 查看 Zookeeper Base Path



2. 将“ZooKeeper Base Path”值配置到URL模板中，即在原有的URL地址后面增加“:/hbase”即可，并单击确定：

图 4-13 配置 URL 模板



步骤10 配置EIP。如果本地Windows与集群之间网络不通，需要为每个HBase节点以及ZooKeeper节点配置EIP，并且在本地Windows的hosts文件添加所有节点的公网IP对应主机域名的映射关系，例如：

```
100.        node-master3xCPw node-master3xCPw.
100.        node-group-1ZqBd0001 node-group-1ZqBd0001.
100.        node-master2uIQz node-master2uIQz.
100.        0 node-group-1ZqBd0002 node-group-1ZqBd0002.
```

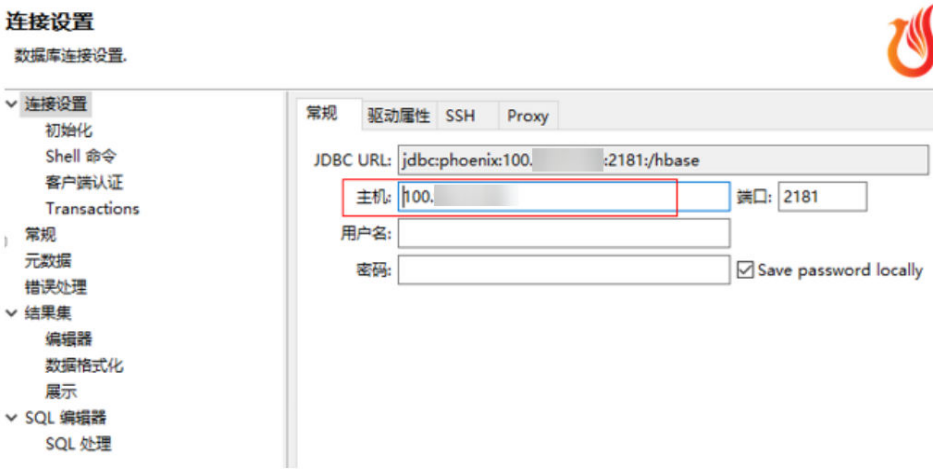
说明

如果使用的是Windows云服务器，并且与集群的网络是通的，则无需配置EIP。

步骤11 登录FusionInsight Manager，选择“集群 > 服务 > ZooKeeper > 实例”。

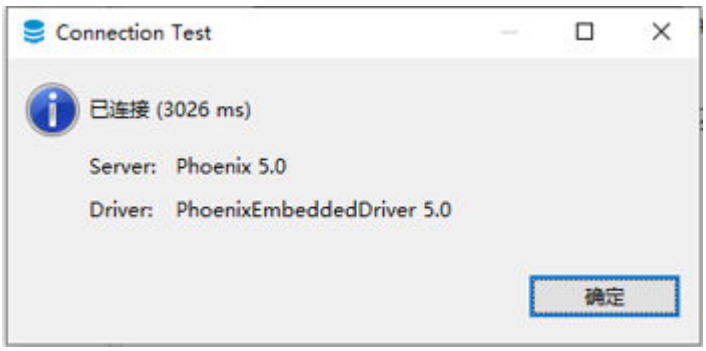
选择任一节点，将该节点对应的EIP填入主机框中（如果使用的是Windows云服务器，并且与集群的网络是通的，直接填写节点的业务IP地址即可）。

图 4-14 配置主机



步骤12 单击“测试连接”，如图10 测试连接所示表示对接成功，单击“确定”。

图 4-15 测试连接



步骤13 以客户端安装用户登录安装HBase客户端的节点，执行以下命令创建新的命名空间*MY_NS*:

cd *客户端安装目录*

source bigdata_env

hbase shell

create_namespace "*MY_NS***"**

步骤14 打开DBeaver，选择“SQL编辑器 > 新建SQL编辑器”，即可执行相关SQL语句，例如：

1. 在编辑器中输入以下命令并选择“SQL 编辑器 > 执行 SQL 语句”，即可在**default**中创建表*TEST*:

   CREATE TABLE IF NOT EXISTS *TEST* **(id VARCHAR PRIMARY KEY, name VARCHAR);**

   UPSERT INTO TEST(id,name) VALUES ('1','jamee');

2. 在编辑器中输入以下命令并选择“SQL 编辑器 > 执行 SQL 语句”，即可在*MY_NS*中创建表*TEST*并插入数据：

   CREATE TABLE IF NOT EXISTS *MY_NS.TEST* **(id integer not null primary key, name varchar);**

   UPSERT INTO *MY_NS.TEST* **VALUES(1,'John');**

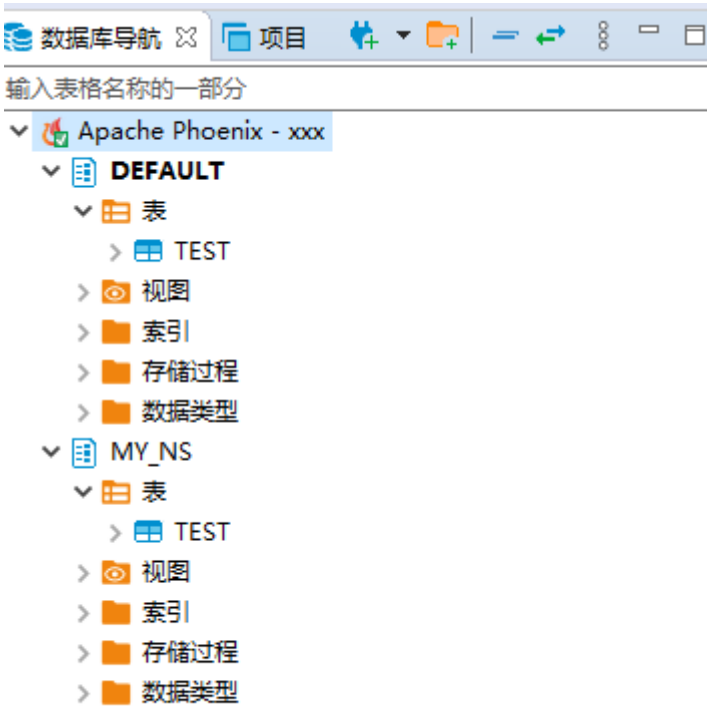
   UPSERT INTO *MY_NS.TEST* **VALUES(2,'Tom');**

   UPSERT INTO *MY_NS.TEST* **VALUES(3,'Manson');**

   UPSERT INTO *MY_NS.TEST* **VALUES(4,'Aurora');**

步骤15 右键单击连接名称，单击“刷新”，再单击连接名称左侧的 ，即可查看在**default**和*MY_NS*中新建的表：

图 4-16 查看表



----结束

4.3 使用 DBeaver 访问 HetuEngine

本章节以DBeaver 7.2.0版本为例，讲解如何访问HetuEngine。

前提条件

- DBeaver已正常安装。DBeaver软件下载链接：**https://dbeaver.io/files/7.2.0/** 。
- 已在集群中创建“人机”用户，如**hetu_user**，可参考**创建HetuEngine用户**。启用Ranger鉴权的集群需根据业务需求为该**hetu_user**添加Ranger权限，可参考**添加HetuEngine的Ranger访问权限策略**。

- 已创建计算实例并运行正常，可参考**创建HetuEngine计算实例**。

操作步骤

步骤1 下载HetuEngine客户端获取JDBC jar包。

1. 登录FusionInsight Manager。
2. 选择“集群 > 服务 > HetuEngine > 概览”。
3. 在页面右上角，选择“更多 > 下载客户端”，根据界面提示下载“完整客户端”文件到本地。
4. 解压HetuEngine客户端压缩包文件“FusionInsight_Cluster_*集群ID*_HetuEngine_Client.tar”获取jdbc文件，并存放在本地，例如“D:\test”。

   说明

   jdbc文件获取方法：

   在“FusionInsight_Cluster_*集群ID*_HetuEngine_ClientConfig\HetuEngine\xxx\”路径下解压获取“hetu-jdbc-*.jar”文件。

   备注：xxx为“arm”或“x86”。

步骤2 在本地hosts文件添加主机映射。

根据使用HSFabric方式或HSBroker方式添加对应实例所在主机映射，格式为：主机IP 主机名

例如：192.168.42.90 server-2110081635-0001

说明

Windows本地hosts文件存放路径举例：“C:\Windows\System32\drivers\etc”。

步骤3 打开DBeaver，选择“数据库 > 新建连接”，在“ALL”中搜索“PrestoSQL”并打开PrestoSQL。

步骤4 单击“编辑驱动设置”，参考下表信息设置相关参数。

表 4-5 驱动设置信息

参数名	参数值
类名	io.prestosql.jdbc.PrestoDriver

参数名	参数值
URL模板	• 通过HSFabric方式访问HetuEngine jdbc:presto://<*HSFabricIP1:port1*>,<*HSFabricIP2:port2*>,<*HSFabricIP3:port3*>/hive/default?serviceDiscoveryMode=hsfabric 示例： jdbc:presto://192.168.42.90:29902,192.168.42.91:29902,192.168.42.92:29902/hive/default?serviceDiscoveryMode=hsfabric • 通过HSBroker方式访问HetuEngine jdbc:presto://<*HSBrokerIP1:port1*>,<*HSBrokerIP2:port2*>,<*HSBrokerIP3:port3*>/hive/default?serviceDiscoveryMode=hsbroker 示例： jdbc:presto://192.168.42.90:29860,192.168.42.91:29860,192.168.42.92:29860/hive/default?serviceDiscoveryMode=hsbroker

说明

- 获取HSFabric、HSBroker节点IP及端口号：
  1. 登录FusionInsight Manager。
  2. 选择“集群 > 服务 > HetuEngine > 实例”，获取HSFabric或HSBroker所有实例的业务IP，可选择一个或多个正常状态的进行连接。
  3. 获取端口号，选择“集群 > 服务 > HetuEngine > 配置 > 全部配置”：

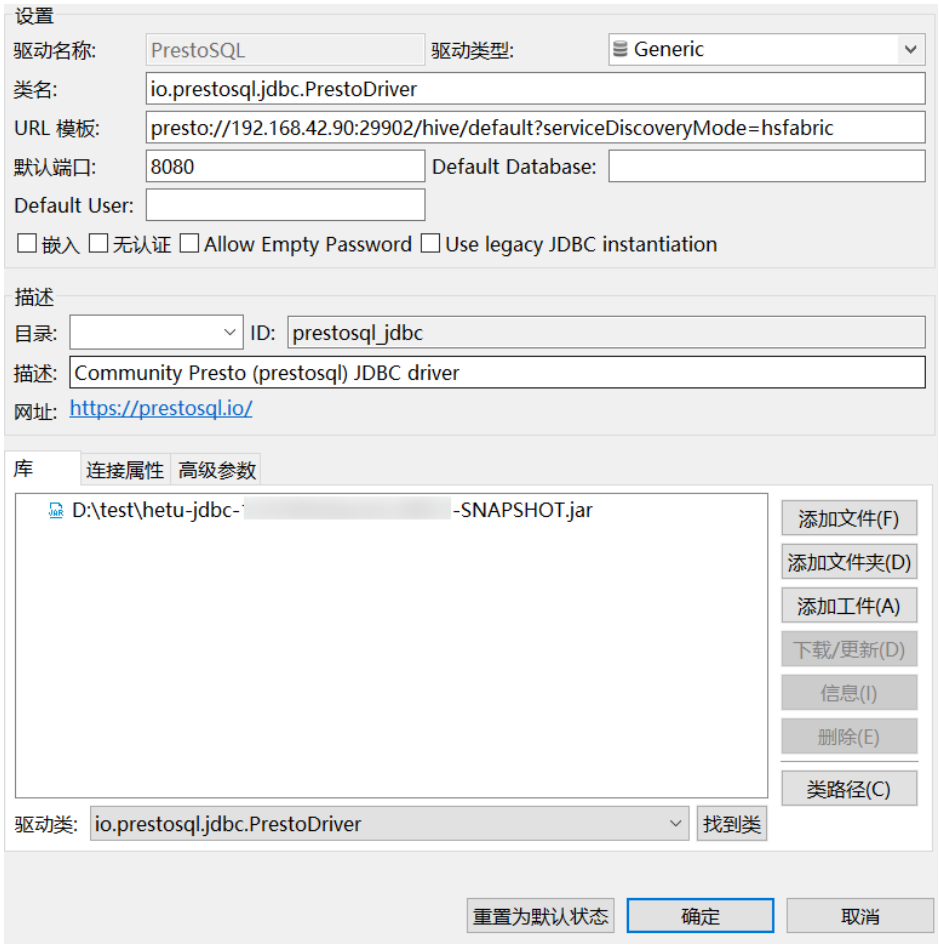
     搜索“gateway.port”，获取HSFabric的端口号，安全模式默认为29902，普通模式默认29903；

     搜索“server.port”，获取HSBroker的端口号，安全模式默认为29860，普通模式默认29861；
- 如果连接不成功，请关闭代理重试。

步骤5 单击“添加文件”，上传**步骤1**中获取的JDBC驱动包。

步骤6 单击“找到类”，自动获取驱动类，单击“确定”完成驱动设置，如下图所示。如果“库”中存在“io.prestosql:presto-jdbc:RELEASE”，单击“找到类”前需将其删掉。

图 4-17 安全模式驱动设置

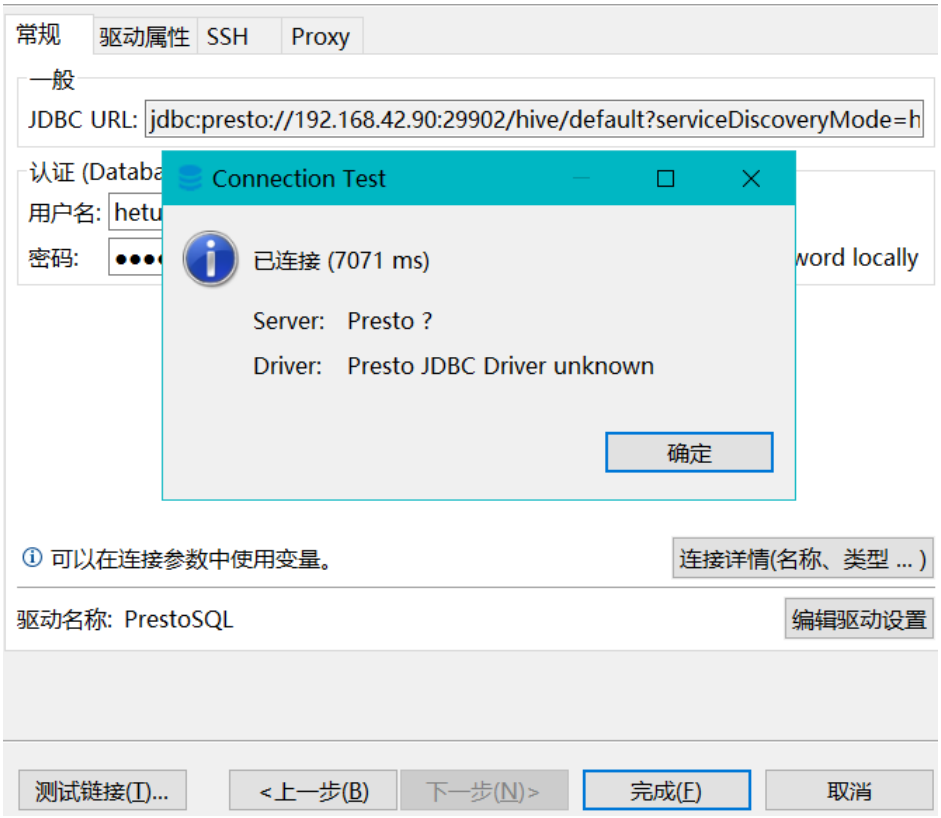


步骤7 连接设置。

- 安全模式（开启Kerberos认证的集群）：

  在创建新连接页面的“常规”页签，输入**前提条件**中创建的用户名和密码，单击“测试链接”，连接成功后，单击“确定”，再单击“完成”。可单击“连接详情（名称、类型...）”修改连接名称。

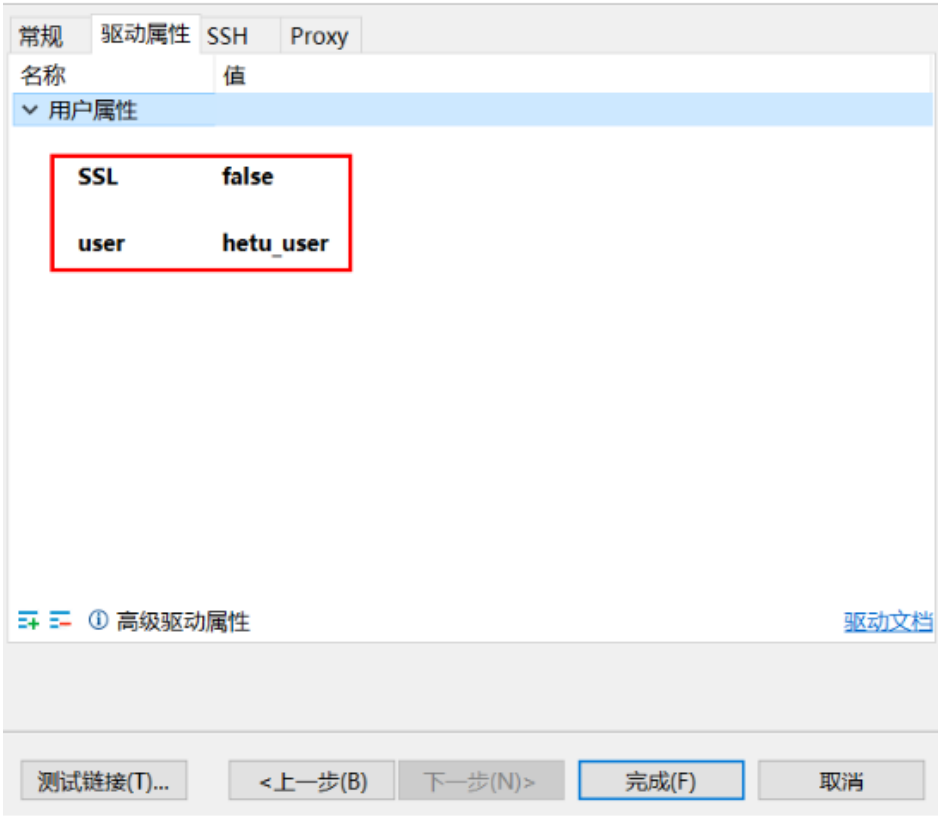
图 4-18 安全模式“常规”参数设置



- 普通模式（未开启Kerberos认证的集群）：

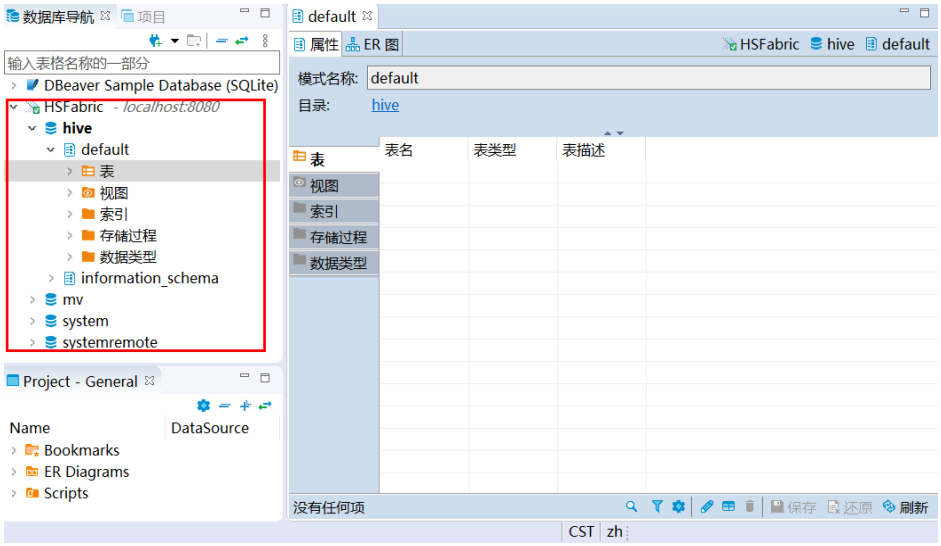
  在创建新连接页面的“驱动属性”配置如下参数，“user”为**前提条件**中创建的用户。单击“测试链接”，连接成功后，单击“确定”，再单击“完成”。可单击“连接详情（名称、类型...）”修改连接名称。

图 4-19 普通模式“驱动属性”参数设置



步骤8 连接成功后进入如图4-20所示页面。

图 4-20 连接成功



----结束

4.4 使用 FineBI 访问 HetuEngine

本章节以FineBI 5.1.9版本为例，讲解如何访问安全模式集群的HetuEngine。

前提条件

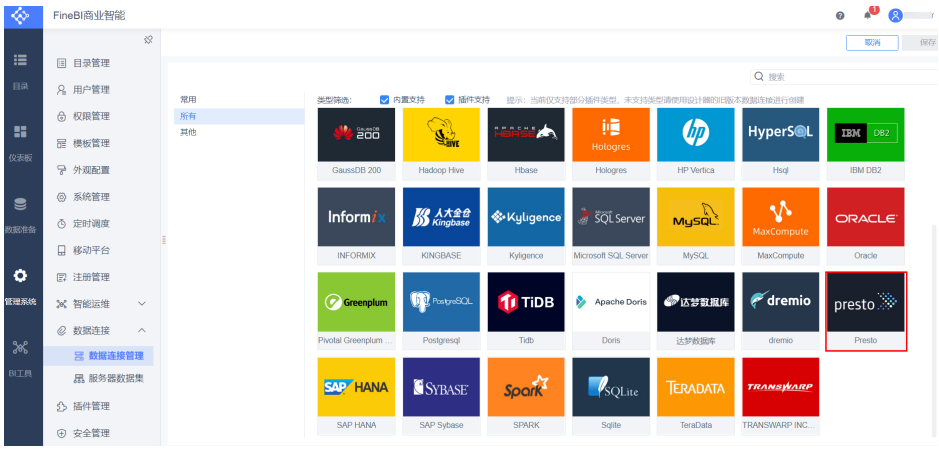
- 已安装FineBI。
- 已获取JDBC jar包文件，获取方法请参考**步骤1**。
- 已在集群中创建“人机”用户，如**hetu_user**，可参考**创建HetuEngine用户**。启用Ranger鉴权的集群需根据业务需求为该**hetu_user**添加Ranger权限，可参考**添加HetuEngine的Ranger访问权限策略**。
- 已创建计算实例并运行正常，可参考**创建HetuEngine计算实例**。

操作步骤

步骤1 将获取的jar包放在目录“%FineBI%\webapps\webroot\WEB-INF\lib”，重启FineBI。

步骤2 打开FineBI，选择“管理系统 > 数据连接 > 数据连接管理 > 新建数据连接 > 所有”，选择“Presto”并打开。参考**图4-21**新建连接。

图 4-21 新建连接



步骤3 参考下表配置连接参数。配置完成后单击“测试连接”，或在“模式”单击“点击连接数据库”测试数据连接，测试成功后单击“保存”。

表 4-6 HSFabric 连接参数

参数名	参数值
数据连接名称	自定义
驱动	io.prestosql.jdbc.PrestoDriver
数据库名称	hive/default?serviceDiscoveryMode=hsfabric
主机	hsfabric实例所在节点IP
端口	HSFabric服务gateway.port端口
用户名	已创建的“人机”用户的用户名，如：admintest

参数名	参数值
密码	已创建的“人机”用户的用户密码 **说明** • 使用用户名密码方式登录时需要配置该参数。 • 未启用Kerberos认证（普通模式）的集群不填写该参数。
编码	自动
数据库连接URL	• 启用Kerberos认证（安全模式）的集群 jdbc:presto://<*HSFabricIP1:port1*>,<*HSFabricIP2:port2*>,<*HSFabricIP3:port3*>/hive/default?serviceDiscoveryMode=hsfabric，详情请参考**表4-5**。 • 未启用Kerberos认证（普通模式）的集群 jdbc:presto://<*HSFabricIP1:port1*>,<*HSFabricIP2:port2*>,<*HSFabricIP3:port3*>/hive/default?serviceDiscoveryMode=hsfabric&SSL=false。

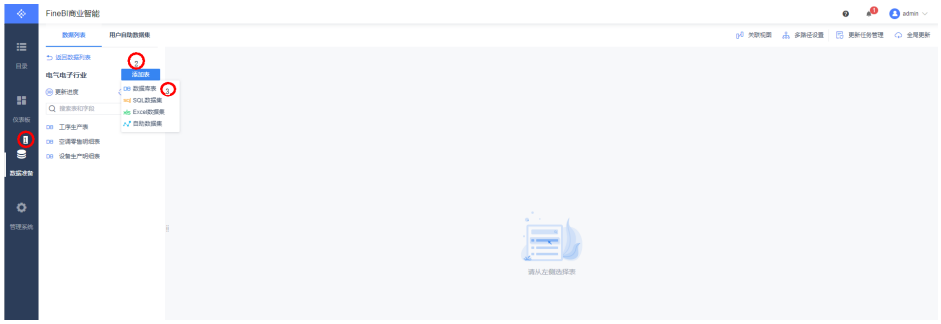
表 4-7 HSbroker 连接参数

参数名	参数值
数据连接名称	自定义
驱动	io.prestosql.jdbc.PrestoDriver
数据库名称	hive/default?serviceDiscoveryMode=hsbroker
主机	hsbroker实例所在节点ip
端口	Hsbroker服务端口
用户名	已创建的“人机”用户的用户名，如：admintest
密码	已创建的“人机”用户的用户密码 **说明** • 使用用户名密码方式登录时需要配置该参数。 • 未启用Kerberos认证（普通模式）的集群不填写该参数。
编码	自动

参数名	参数值
数据库连接URL	• 启用Kerberos认证（安全模式）的集群 jdbc:presto://<*HSBrokerIP1:port1*>,<*HSBrokerIP2:port2*>,<*HSBrokerIP3:port3*>/hive/default?serviceDiscoveryMode=hsbroker，详情请参考表4-5。 • 未启用Kerberos认证（普通模式）的集群 jdbc:presto://<*HSBrokerIP1:port1*>,<*HSBrokerIP2:port2*>,<*HSBrokerIP3:port3*>/hive/default?serviceDiscoveryMode=hsbroker&SSL=false

步骤4 参考图4-22所示配置数据库表，选择“数据准备 > 数据列表”，单击“添加分组”，选择“添加表 > 数据库表 ”。

图 4-22 配置数据



步骤5 设置需要用于做分析的表，如图4-23~图4-25所示。

图 4-23 单击“数据连接”

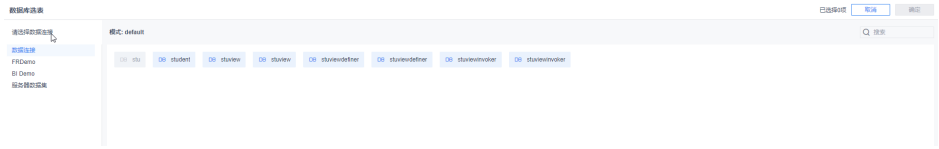


图 4-24 选择数据库

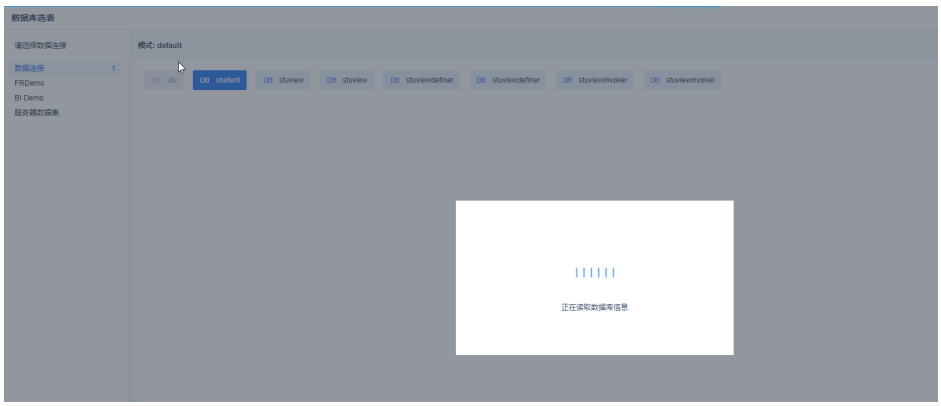
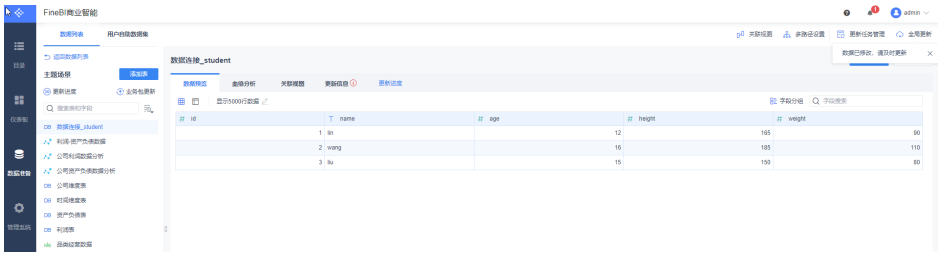


图 4-25 数据预览

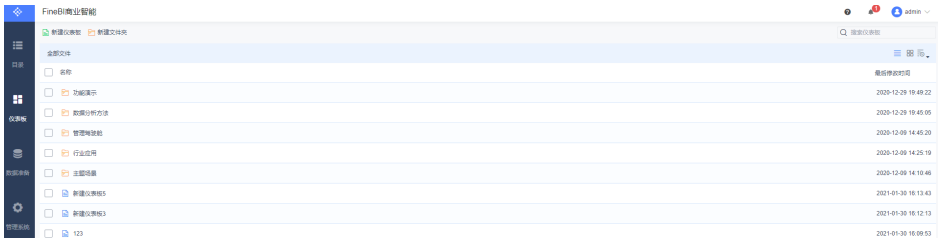


步骤6 单击“更新信息”中的“单表更新”，进行数据同步。

图 4-26 数据同步

步骤7 单击“仪表板”，单击“新建仪表板”，输入相关名称单击“确定”。

图 4-27 新建仪表板

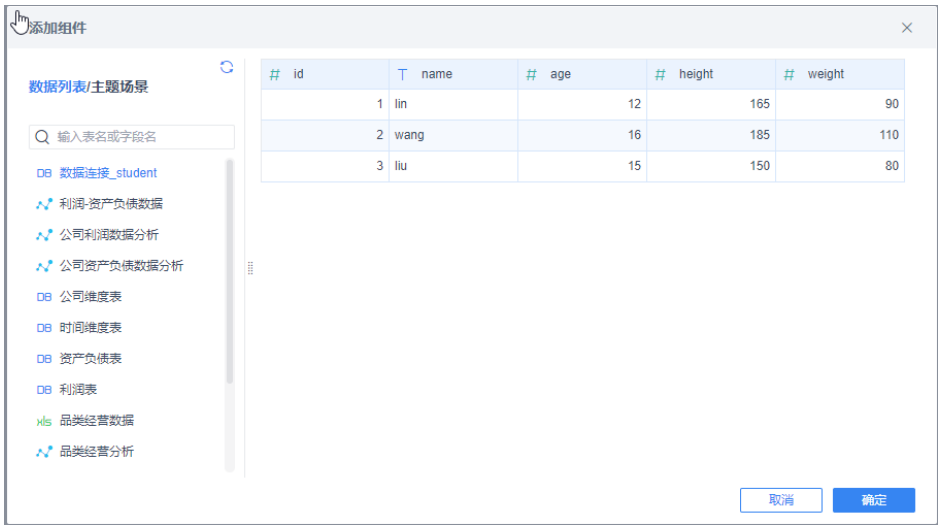


步骤8 单击“添加组件”。

图 4-28 添加组件

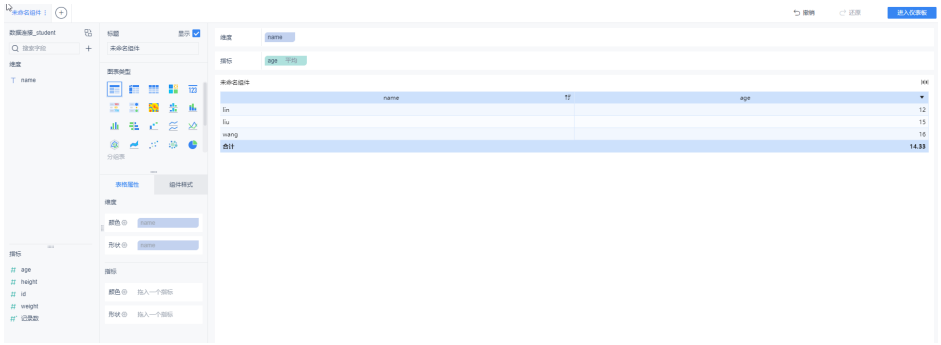
步骤9 添加步骤5配置的需要用于分析的数据表。

图 4-29 添加数据表



步骤10 将“name”拖入“维度”，将“age”拖入“指标”，即可分析年龄的平均值。如**图 4-30**所示。

图 4-30 分析表



步骤11 如果需要用图显示，则可在“图表类型”中选择相对应的图。样例中是选择“柱状图”。

图 4-31 选择图表类型

----结束

4.5 使用 Tableau 访问 HetuEngine

本章节以Tableau Desktop 2022.2版本为例，讲解如何访问安全模式集群的HetuEngine。

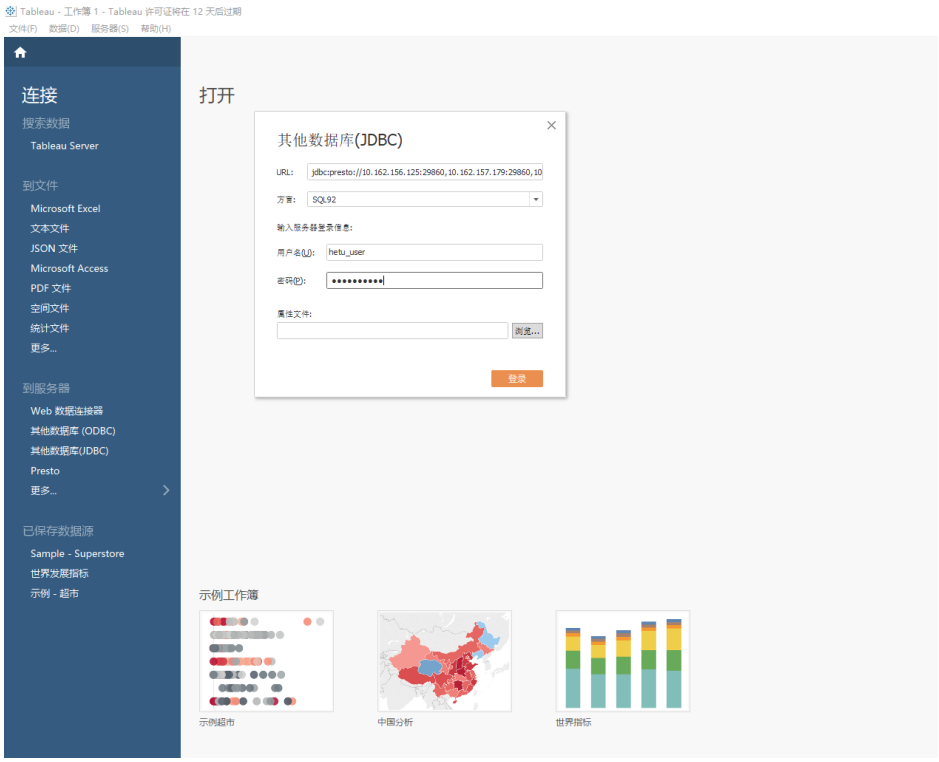
前提条件

- 已安装Tableau Desktop。
- 已获取JDBC jar包文件，获取方法请参考**步骤1**。
- 已在集群中创建“人机”用户，如**hetu_user**，可参考**创建HetuEngine用户**。启用Ranger鉴权的集群需根据业务需求为该**hetu_user**添加Ranger权限，可参考**添加HetuEngine的Ranger访问权限策略**。
- 已创建计算实例并运行正常，可参考**创建HetuEngine计算实例**。

操作步骤

步骤1 将获取的Jar包放在Tableau安装目录，如“C:\Program Files\Tableau\Drivers”。

步骤2 打开Tableau，选择“到服务器 > 其他数据库(JDBC)”，输入URL和已创建的“人机”用户的用户名及密码，单击“登录”。支持HSFabric方式和HSBroker方式连接，URL格式详情可参考**表4-5**。



步骤3 登录成功后，将要操作的数据表拖到右边操作窗口，刷新数据。

----结束

4.6 使用永洪 BI 访问 HetuEngine

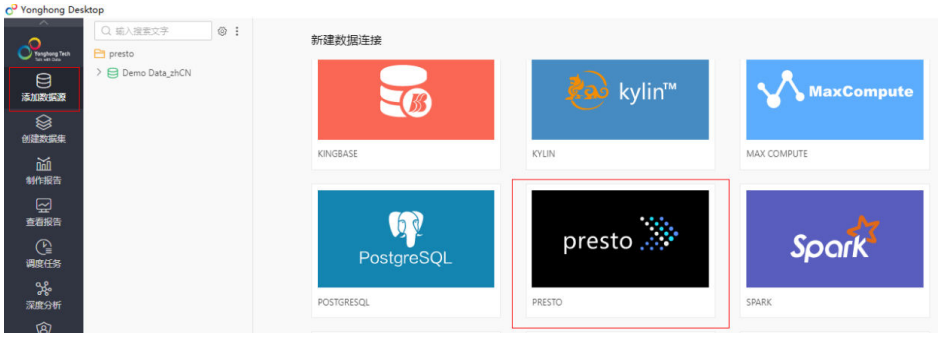
本章节以Yonghong Desktop 9.1版本为例，讲解如何访问安全模式集群的HetuEngine。

前提条件

- 已安装Yonghong Desktop。
- 已获取JDBC jar包文件，获取方法请参考**步骤1**。
- 已在集群中创建“人机”用户，如**hetu_user**，可参考**创建HetuEngine用户**。启用Ranger鉴权的集群需根据业务需求为该**hetu_user**添加Ranger权限，可参考**添加HetuEngine的Ranger访问权限策略**。
- 已创建计算实例并运行正常，可参考**创建HetuEngine计算实例**。

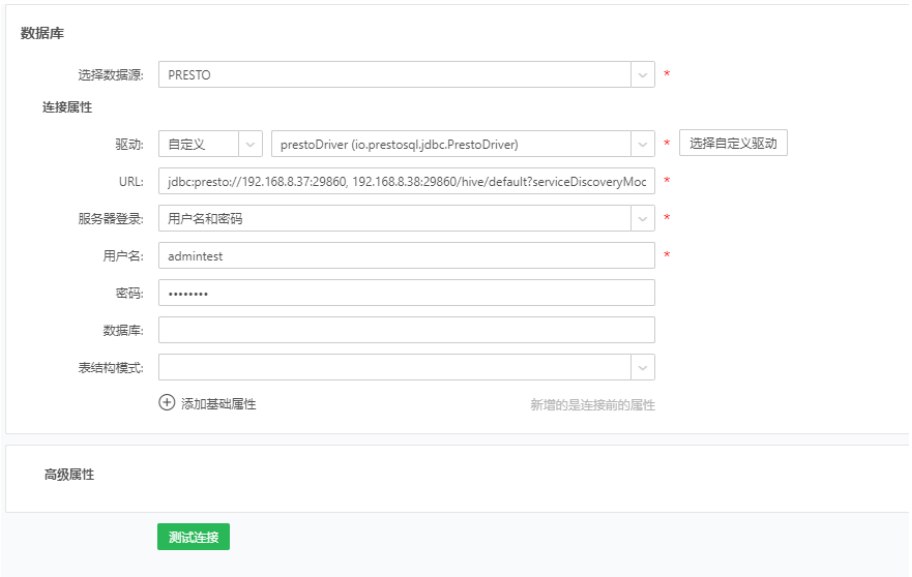
操作步骤

步骤1 打开Yonghong Desktop，选择“添加数据源 > presto”。

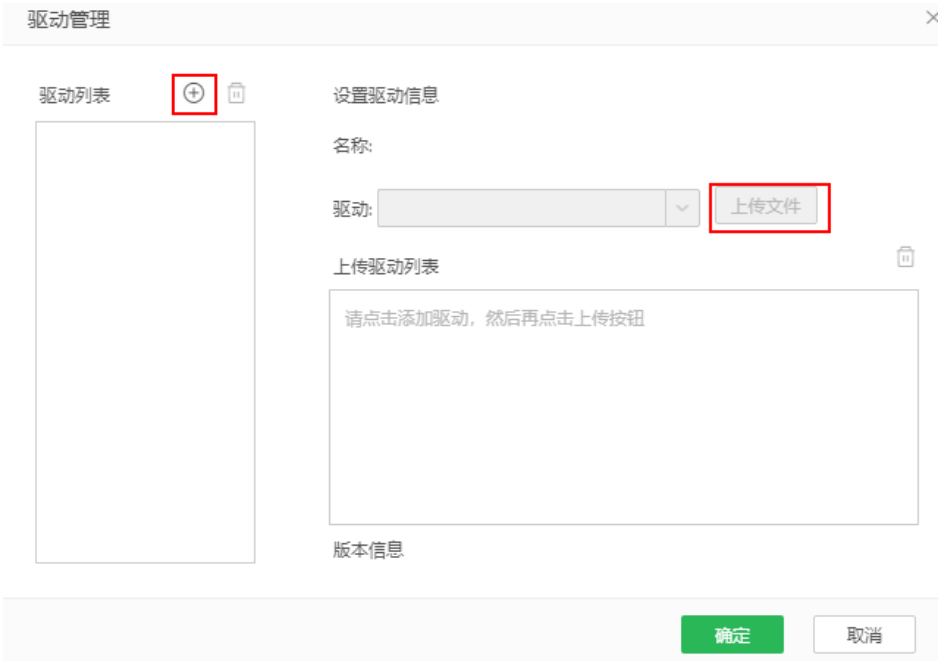


步骤2 在数据源配置页面参考**图4-32**完成参数配置，“用户名”和“密码”为已创建的“人机”用户的用户名和用户密码。配置完成后可以单击“测试连接”测试。

图 4-32 数据源配置



- 驱动：选择“自定义 > 选择自定义驱动”，单击 ，编辑驱动名称，单击“上传文件”上传已获取的JDBC jar包，单击“确定”。

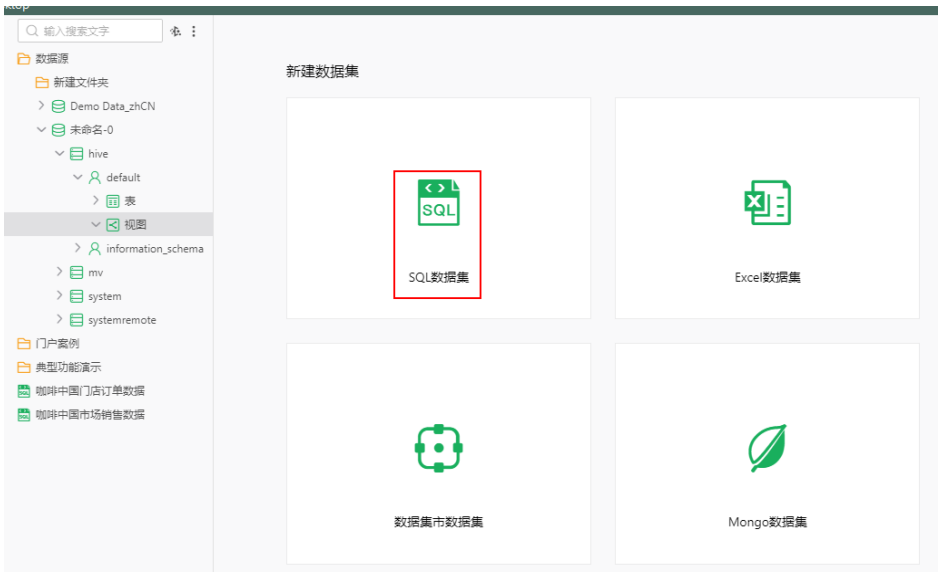


- URL：支持HSFabric方式和HSBroker方式，详情请参考表4-5。
- 服务器登录：选择“用户名和密码”并填写相应的用户名及密码。

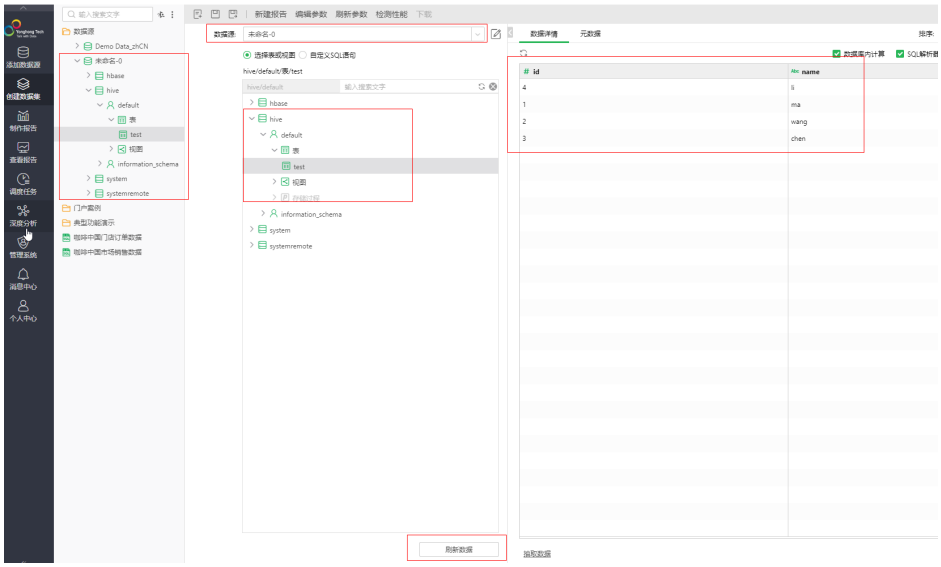
步骤3 单击“新建数据集”，在弹出的页面参考图4-33修改保存路径及文件名称，单击“确定”保存修改路径及文件名称。

图 4-33 修改路径及名称

步骤4 在数据源选择**步骤3**新建的数据集的文件名称，此处以默认文件名称“未命名-0”为例，选择“未命名-0 > hive > default > 视图”，在右侧“新建数据集”选择“SQL数据集”。



步骤5 在“数据源”处选择**步骤3**新建的数据集，显示所有表信息，选中其中一个表，如“test”表，单击“刷新数据”，可在右侧“数据详情”中显示表的所有信息。



----结束

4.7 Hive 对接外置自建关系型数据库

说明

- 本章节适用于MRS 3.x及后续版本。
- 自建关系型数据库：本章节以对接开源MySQL和Postgres数据库进行说明。
- 在已有Hive数据的集群上外置元数据库后，之前的元数据表不会自动同步。因此在安装Hive之初就要确认好元数据是外置数据库还是内置到DBService，如果是外置自建数据库，则需在安装Hive时或者暂无Hive数据时将元数据外置，安装后不允许修改，否则将会造成原有元数据丢失。
- 当外置元数据到MySQL后，Hive仅表名、字段名、表描述支持中文，其余暂不支持。

Hive支持开源MySQL和Postgres元数据库。

步骤1 安装开源MySQL或Postgres数据库。

说明

数据库安装节点需与集群处于同一网段，能互相访问。

步骤2 上传驱动包。

- Postgres：

  使用开源驱动包替换集群已有的驱动包。将Postgres驱动包“postgresql-42.2.5.jar”上传至所有MetaStore实例节点“${BIGDATA_HOME}/third_lib/Hive”目录下（开源驱动包下载地址：**https://repo1.maven.org/maven2/org/postgresql/postgresql/42.2.5/**）。

  在上传驱动包的所有MetaStore实例节点上执行以下命令修改驱动包权限：

  cd ${BIGDATA_HOME}/third_lib/Hive

  chown omm:wheel postgresql-42.2.5.jar

  chmod 600 postgresql-42.2.5.jar

- MySQL：

  进入MySQL官网（**https://www.mysql.com/**），选择“DOWNLOADS > MySQL Community(GPL) DownLoads > Connector/J”下载对应版本的驱动包。

  - MRS 8.2.0之前版本，将MySQL对应版本的驱动包上传至所有Metastore实例节点“/opt/Bigdata/FusionInsight_HD_*/install/FusionInsight-Hive-*/hive-*/lib/”目录下。
  - MRS 8.2.0及之后版本，将MySQL对应版本的驱动包上传至所有Metastore实例节点“${BIGDATA_HOME}/third_lib/Hive”目录下。

  在上传驱动包的所有MetaStore实例节点上执行以下命令修改驱动包权限：

  cd /opt/Bigdata/FusionInsight_HD_*/install/FusionInsight-Hive-*/hive-*/lib/

  chown omm:wheel mysql-connector-java-*.jar

  chmod 600 mysql-connector-java-*.jar

步骤3 在自建数据库中创建用户、元数据库，并为用户赋予该库的所有权限。例如：

- 以数据库管理员用户在Postgres中执行以下命令创建数据库“test”和用户“testuser”，并授予“test”数据库的所有权限给“testuser”用户。

  create user testuser with password '*password*';

create database test owner testuser;

grant all privileges on database test to testuser;

- 以数据库管理员用户在MySQL中执行以下命令创建数据库“test”和用户“testuser”，并授予“test”数据库的所有权限给“testuser”用户。

  create database test;

  create user 'testuser'@'%' identified by '*password*';

  grant all privileges on test.* to 'testuser';

  flush privileges;

步骤4 导入元数据建表SQL。

- Postgres的SQL文件路径：${BIGDATA_HOME}/FusionInsight_HD_*/install/FusionInsight-Hive-*/hive-*/scripts/metastore/upgrade/postgres/hive-schema-3.1.0.postgres.sql

  Postgres导入sql文件的命令：

  ./bin/psql -U *username* **-d** *databasename* **-f hive-schema-3.1.0.postgres.sql**

  其中：

  ./bin/psql：在Postgres安装目录下。

  username：登录Postgres的用户名。

  databasename：数据库库名。

- MySQL的SQL文件路径：${BIGDATA_HOME}/FusionInsight_HD_*/install/FusionInsight-Hive-*/hive-*/scripts/metastore/upgrade/mysql/hive-schema-3.1.0.mysql.sql

  MySQL导入sql文件的命令：

  ./bin/mysql -u *username* **-p -D** *databasename***<hive-schema-3.1.0.mysql.sql**

  其中：

  ./bin/mysql：在MySQL安装目录下。

  username：登录MySQL的用户名。

  databasename：数据库库名。

步骤5 登录FusionInsight Manager，选择“集群 > 服务 > Hive > 配置 > 全部配置 > Hive（服务） > MetaDB”，修改以下参数并保存，使Hive的配置对接到开源数据库。

表 4-8 参数说明

参数名	默认值	描述
javax.jdo.option.ConnectionDriverName	org.postgresql.Driver	Metastore上连接元数据的驱动类。 ● 外置MySQL，则值为：com.mysql.jdbc.Driver ● 外置Postgres，则值为：org.postgresql.Driver

参数名	默认值	描述
javax.jdo.option.ConnectionURL	jdbc:postgresql://%{DBSERVICE_FLOAT_IP}%{DBServer}:%{DBSERVICE_CPORT}/hivemeta?socketTimeout=60	Metastore元数据JDBC链接的URL。 • 外置MySQL，则值为：jdbc:mysql://*MySQL的IP.MySQL的端口*/*test*?characterEncoding=utf-8 • 外置Postgres，则值为：jdbc:postgresql://*Postgres的IP.Postgres的端口号*/*test* **说明** “test”为步骤3中在MySQL或PgSQL中创建的数据库名称。
javax.jdo.option.ConnectionUserName	hive${SERVICE_INDEX}${SERVICE_INDEX}	Metastore上连接外置元数据数据库的用户名。

步骤6 在MetaStore中修改Postgres数据库密码，选择“集群 > *待操作集群名* > 服务 > Hive > 配置 > 全部配置 > MetaStore（角色） > MetaDB”，修改以下参数并保存。

表 4-9 参数说明

参数名	默认值	描述
javax.jdo.option.extend.ConnectionPassword	******	Metastore上连接外置元数据数据库的用户密码。密码后台会加密。

步骤7 登录所有MetaStore服务的后台节点，检查本地目录“/opt/Bigdata/tmp”是否存在。

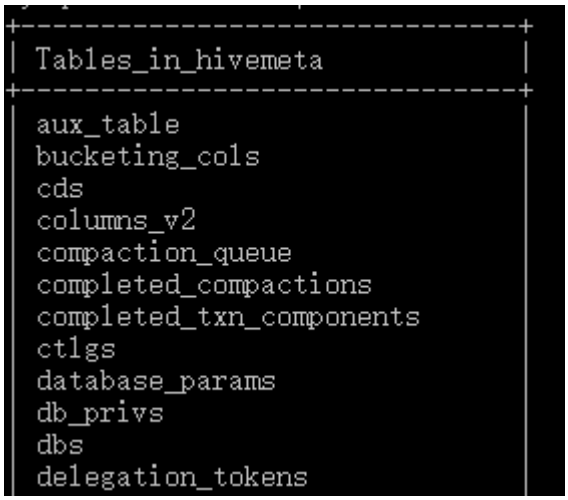
- 存在，直接执行步骤步骤8。
- 不存在，则先执行以下命令，创建目录。

  mkdir -p /opt/Bigdata/tmp

  chmod 755 /opt/Bigdata/tmp

步骤8 保存配置，选择“概览 > 更多 > 重启服务”，输入密码开始重启Hive服务。

步骤9 Hive重启完成后，登录MySQL或Postgres数据库，可以查看到步骤3创建的元数据库中有元数据表生成：



步骤10 验证Hive元数据库是否外置成功：

1. 以客户端安装用户登录安装Hive客户端的节点：
   cd *客户端安装目录*
   source bigdata_env
   kinit *组件业务用户*（未开启Kerberos认证的集群请跳过该操作）
2. 执行以下命令登录Hive客户端命令行：
   beeline
3. 执行以下命令创建表**test**：
   create table *test*(**id int,str1 string,str2 string);**
4. 在MySQL或Postgres的**test**数据库中执行以下命令查看是否有表**test**相关信息：
   select * from TBLS;
   能查看到表**test**相关信息则说明外置Hive数据库成功，例如：
   - 在MySQL中查看的结果为：

     ```
     mysql> mysql> select * from TBLS;
     +--------+-------------+-------+------------------+-------+------------+-----------+--------+----------+---------------+--------------------+--------------------+--------------------+
     | TBL_ID | CREATE_TIME | DB_ID | LAST_ACCESS_TIME | OWNER | OWNER_TYPE | RETENTION | SD_ID  | TBL_NAME | TBL_TYPE      | VIEW_EXPANDED_TEXT | VIEW_ORIGINAL_TEXT | IS_REWRITE_ENABLED |
     +--------+-------------+-------+------------------+-------+------------+-----------+--------+----------+---------------+--------------------+--------------------+--------------------+
     |      6 |  1673413291 |     1 |                0 | root  | USER       |         0 |      6 | test1    | MANAGED_TABLE | NULL               | NULL               |                    |
     +--------+-------------+-------+------------------+-------+------------+-----------+--------+----------+---------------+--------------------+--------------------+--------------------+
     1 row in set (0.00 sec)
     ```

   - 在Postgres中查看的结果为：

     ```
     hivemeta=> select * from "TBLS";
      TBL_ID | CREATE_TIME | DB_ID | LAST_ACCESS_TIME | OWNER | OWNER_TYPE | RETENTION | SD_ID | TBL_NAME |    TBL_TYPE    | VIEW_EXPANDED_TEXT | VIEW_ORIGINAL_TEXT | IS_REWRITE_ENABLED
     --------+-------------+-------+------------------+-------+------------+-----------+-------+----------+----------------+--------------------+--------------------+--------------------
           2 |  1673425195 |     1 |                0 | root  | USER       |         0 |     2 | test1    | MANAGED_TABLE  |                    |                    | f
     (1 row)
     ```

----结束

4.8 Hive 对接 CSS 服务

操作场景

利用Elasticsearch-Hadoop插件，完成Hive和CSS服务的Elasticsearch直接的数据交互，通过Hive外部表的方式，可以快速将Elasticsearch索引数据映射到Hive表中。

说明

该章节适用于MRS 3.x及之后版本。

前提条件

已安装MRS的Hive服务和CSS的Elasticsearch服务，并且两个集群之间网络互通。

操作步骤

步骤1 在云搜索服务的“集群管理”页面上，单击集群“操作”列的“Kibana”访问集群，在Kibana的左侧导航中选择“Dev Tools”，进入Console界面，输入以下命令创建索引“ddj_study_card_ratio_v12”：

```
PUT ddj_study_card_ratio_v12
{
   "mappings" : {
    "properties" : {
      "uniq_id" : {
        "type" : "text",
        "fields" : {
          "keyword" : {
            "type" : "keyword",
            "ignore_above" : 256
          }
        }
      }
    }
   }
}
```

返回如下结果表示索引创建成功：

```
{
  "acknowledged" : true,
  "shards_acknowledged" : true,
  "index" : "ddj_study_card_ratio_v12"
}
```

步骤2 执行以下命令向“ddj_study_card_ratio_v12”索引中插入数据：

```
POST /ddj_study_card_ratio_v12/_doc/_bulk
{"index":{}}
{"id":"1", "uniq_id":"23323"}
```

当返回结果信息中“errors”字段的值为“false”时，表示导入数据成功。

步骤3 根据当前CSS服务中的Elasticsearch版本在**Past Releases**下载对应的Jar包。

例如：Elasticsearch 7.6.2对应的Jar包为“elasticsearch-hadoop-7.6.2.jar”。

说明

- Jar包版本需与CSS服务的Elasticsearch版本保持一致，本章节以开启安全模式的“Elasticsearch 7.6.2”集群为例进行相关操作，具体请以实际情况为准。
- 若有额外的自定义模块，也单独打包成一份Jar包。

步骤4 将**步骤3**的Jar包分别上传到所有的HiveServer所在节点的“/opt/Bigdata/third_lib/Hive”目录下，并执行以下命令修改权限。

chown omm:wheel -R /opt/Bigdata/third_lib/Hive

步骤5 登录FusionInsight Manager，选择“集群 > 服务 > Hive > 实例”，勾选所有的HiveServer实例，选择“更多 > 重启实例”，重启HiveServer实例。

步骤6 在**Maven中心仓**下载“commons-httpclient-3.1.jar”，并将该Jar包及**步骤3**的Jar上传至集群中安装了HDFS和Hive客户端的任一的节点上。

步骤7 以客户端安装用户，登录**步骤6**上传了Jar包的节点。

步骤8 执行以下命令认证用户。

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9 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10 执行以下命令将**步骤6**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11 执行以下命令让Hive在执行命令行任务时加载指定Jar包。

beeline

add jar *Jar包存放在hdfs中的路径*;（每个Jar包分别执行一次该命令）

步骤12 执行以下命令创建Elasticsearch外部表。

```
CREATE EXTERNAL TABLE `ddj_study_card_ratio_v12_test`(
  `uniq_id` string)
ROW FORMAT SERDE
  'org.elasticsearch.hadoop.hive.EsSerDe'
STORED BY
  'org.elasticsearch.hadoop.hive.EsStorageHandler'
WITH SERDEPROPERTIES (
  'field.delim'='',
  'serialization.format'='')
TBLPROPERTIES (
  'bucketing_version'='2',
  'es.index.auto.create'='false',
  'es.mapping.date.rich'='false',
  'es.net.http.auth.pass'='Pzh6537projectx',
  'es.net.http.auth.user'='elastic',
  'es.nodes'='vpcep-e0b33065-75b7-4193-8395-dbd00d10bc39.cn-east-3.huaweicloud.com',
  'es.nodes.wan.only'='true',
  'es.port'='9200',
  'es.read.metadata'='true',
  'es.resource'='ddj_study_card_ratio_v12',
  'es.set.netty.runtime.available.processors'='false',
  'es.write.operation'='index',
  'last_modified_by'='root',
  'last_modified_time'='1655264909',
  'transient_lastDdlTime'='1655264909');
```

说明

关键参数说明：

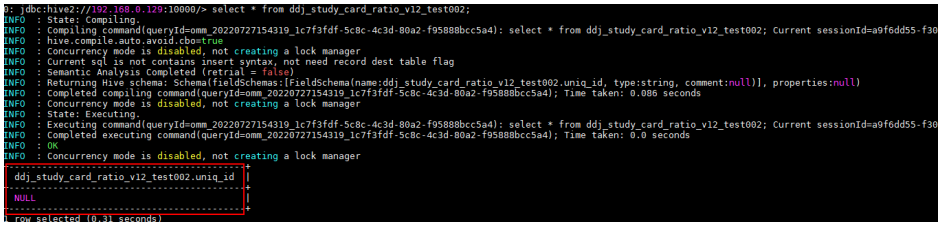
- es.net.http.auth.pass、es.net.http.auth.user：在Kibana中创建的具有**步骤1**创建的索引的操作权限的用户密码及用户名，详细内容请参见**使用Kibana创建用户并授权**。
- es.nodes：需要连接的IP，可登录CSS管理控制台，在集群列表的“内网访问地址”列即可查看对应集群的IP地址。
- es.port：外部访问Elasticsearch集群端口，默认为9200。
- es.resource：**步骤1**创建的索引名称。

更多参数配置可参考开源社区文档**https://www.elastic.co/guide/en/elasticsearch/hadoop/6.1/hive.html**。

步骤13 执行以下命令查看**步骤12**创建的Elasticsearch外部表：

select * from *ddj_study_card_ratio_v12_test*;

当返回结果信息中无报错信息，并且查询成功时，表示Hive成功对接CSS服务。查询结果如下所示：



----结束

4.9 Hive 对接外部 LDAP

本操作适用于MRS 3.1.0及之后版本。

步骤1 登录Manager。

步骤2 在Manager界面，选择“集群 > *待操作集群的名称* > 服务 > Hive > 配置 > 全部配置 > HiveServer（角色） > 安全”。

步骤3 配置如下参数的值。

表 4-10 参数配置

参数名称	参数描述	备注
hive.server2.authentication	hive server认证方式	【取值范围】KERBEROS或LDAP 【默认值】KERBEROS
hive.server2.authentication.ldap.baseDN	LDAP basedn	-
hive.server2.authentication.ldap.password	LDAP密码	健康检查所使用的LDAP密码
hive.server2.authentication.ldap.url.ip	LDAP IP	-
hive.server2.authentication.ldap.url.port	LDAP端口	【默认值】389
hive.server2.authentication.ldap.userDNPattern	LDAP userDNPattern	若该参数值有多个，使用“:”分隔。例如：cn=%s,ou=People1,dc=huawei,dc=com: cn=%s,ou=People2,dc=huawei,dc=com
hive.server2.authentication.ldap.username	LDAP用户名	健康检查所使用的LDAP用户名

步骤4 修改完成后，单击左上方“保存”，在弹出的对话框中单击“确定”保存配置。

步骤5 选择“集群 > *待操作集群的名称* > 服务 > Hive > 实例”，勾选配置状态为“配置过期”的实例，选择“更多 > 重启实例”重启受影响的Hive实例。

----结束

4.10 使用 MRS Spark SQL 访问 DWS

华为云提供MapReduce服务（MRS），可在云上快速构建和运营全栈云原生大数据平台。它包含HDFS、Hive、HBase、Spark等大数据组件，专为分析海量企业数据而量身定制。

Spark提供了类似SQL的Spark SQL语言，用于对结构化数据进行操作。使用Spark SQL，可以访问不同的数据库，用户可以从这些数据库中提取数据，处理并加载到不同的数据存储中。

本实践演示如何使用MRS Spark SQL访问GaussDB (DWS)数据。

说明

本章节仅适用于MRS 3.x及之后版本。

前提条件

- **创建MRS集群**，包含Spark组件。
- 如果MRS集群**开启了Kerberos认证**，登录FusionInsight Manager页面，选择“系统 > 权限 > 用户”，添加一个人机用户sparkuser，用户组（hadoop、hive），主组（hadoop）。并参考**添加Spark2x的Ranger访问权限策略**章节，添加“ADD JAR操作”权限。如果MRS集群未开启Kerberos认证，则无需创建用户。
- 安装MRS集群客户端，具体请参考**安装客户端**。
- **创建GaussDB(DWS)集群**，为确保网络连通，GaussDB (DWS)集群需与MRS集群“可用区”、“虚拟私有云”、“安全组”配置相同。
- 已获取连接GaussDB (DWS)数据库的IP地址、端口、数据库名称、用户名和密码。此外，操作用户必须具有GaussDB (DWS)表的读写权限。

操作步骤

步骤1 准备数据，在GaussDB(DWS)集群中创建数据库和表：

1. 登录GaussDB(DWS)管理控制台，单击DWS集群“操作”列的“登录”。
2. 登录现有GaussDB(DWS)集群的默认数据库gaussdb，执行以下命令，创建数据库“dws_test”。

   CREATE DATABASE dws_test;
3. 连接到创建的新数据库，执行以下命令，创建表“dws_order”。

   CREATE SCHEMA dws_data;

   CREATE TABLE dws_data.dws_order

   (order_id VARCHAR,

   order_channel VARCHAR,

   order_time VARCHAR,

   cust_code VARCHAR,

   pay_amount DOUBLE PRECISION,

   real_pay DOUBLE PRECISION);
4. 执行以下命令插入数据到表“dws_order”中。

INSERT INTO dws_data.dws_order VALUES ('202306270001', 'webShop', '2023-06-27 10:00:00', 'CUST1', 1000, 1000);

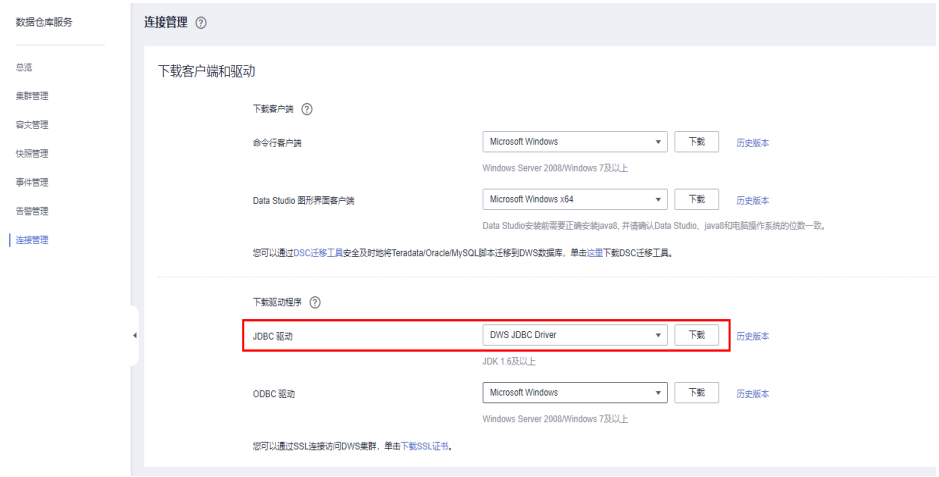
INSERT INTO dws_data.dws_order VALUES ('202306270002', 'webShop', '2023-06-27 11:00:00', 'CUST2', 5000, 5000);

5. 查询表数据，验证数据是否插入。

   SELECT * FROM dws_data.dws_order;

步骤2 下载GaussDB (DWS)数据库JDBC驱动并上传到MRS集群。

1. 登录GaussDB (DWS)管理控制台，单击左侧的“连接管理”，下载JDBC驱动，如下图所示：



2. 解压，获取“gsjdbc200.jar”文件，并上传到MRS集群主Master节点，例如上传到“/tmp”目录下。
3. 使用**root**用户登录MRS集群主Master节点，执行如下命令：

   cd *{客户端安装目录}*

   source bigdata_env

   kinit sparkuser（首次认证需要修改密码，未开启Kerberos认证，则无需执行kinit命令。）

   hdfs dfs -put /tmp/gsjdbc200.jar /tmp

```
[root@node-master2lsnt tmp]# hdfs dfs -put ./gsjdbc200.jar /tmp
SLF4J: Class path contains multiple SLF4J bindings.
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/client/HDFS/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.30.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]
put: `/tmp/gsjdbc200.jar': File exists
[root@node-master2lsnt tmp]#
```

步骤3 在MRS Spark中创建数据源表，并访问DWS表：

1. 登录Spark客户端节点，执行如下命令：

   cd *客户端安装目录*

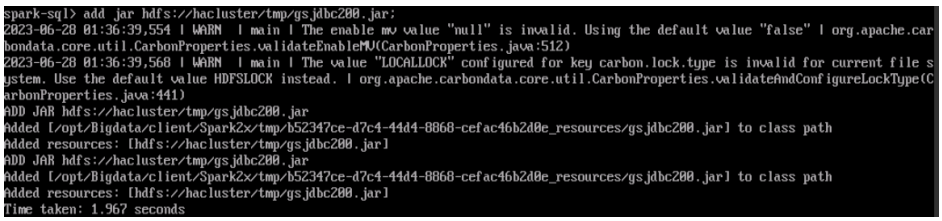
source ./bigdata_env

kinit sparkuser

spark-sql --master yarn

2. 执行如下命令，添加驱动程序jar：

   add jar hdfs://hacluster/tmp/gsjdbc200.jar;



3. 执行如下命令，在Spark中创建数据源表，访问DWS数据：

   CREATE TABLE IF NOT EXISTS spk_dws_order

   USING JDBC OPTIONS (

   'url'='jdbc:gaussdb://192.168.0.228:8000/dws_test',

   'driver'='com.huawei.gauss200.jdbc.Driver',

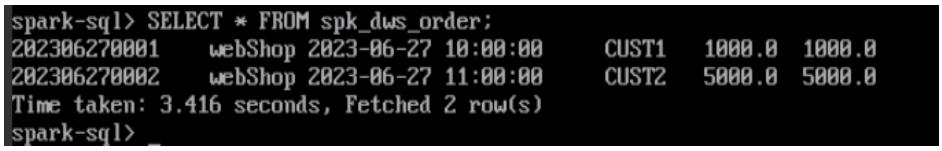
   'dbtable'='dws_data.dws_order',

   'user'='dbadmin',

   'password'='*xxx*');

4. 查询Spark表，验证显示的数据是否与DWS数据相同：

   SELECT * FROM spk_dws_order;



   可以验证返回的数据与**步骤1**中所示的数据相同。

----结束

4.11 MRS Kafka 对接 Kafka Eagle

前提条件

- 创建并购买一个包含Kafka组件的MRS 3.1.0版本集群，集群未开启Kerberos认证，详情可参考**创建MRS集群**。
- 安装MRS集群客户端，具体请参考**安装客户端**。

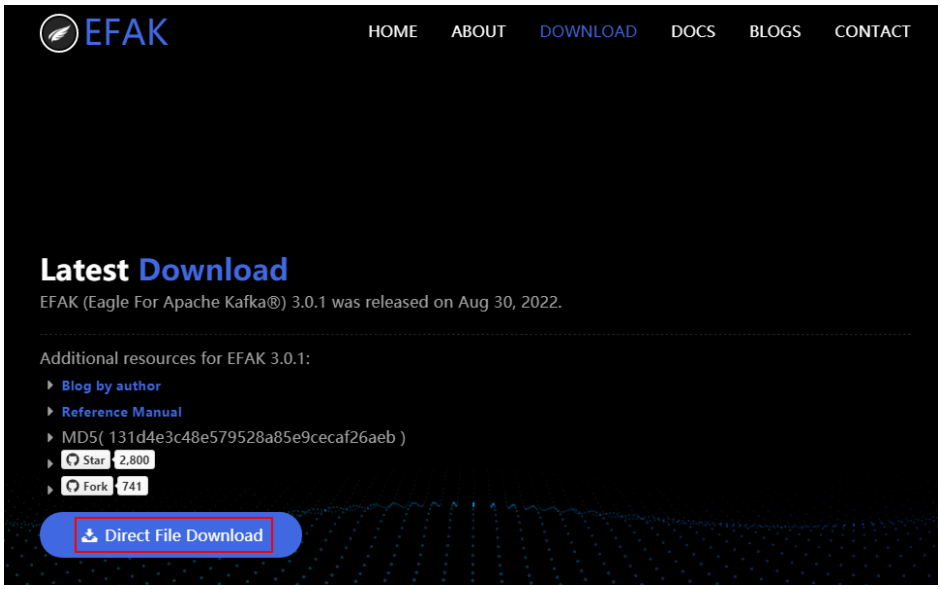
Kafka Eagle 简介

Kafka Eagle是一款分布式、高可用的Kafka监控软件，提供丰富的Kafka监控指标，例如：Kafka集群的Broker数、Topic数、Consumer数、Topic LogSize Top10、Topic Capacity Top10、Lag挤压、CPU/Memory监控等。

Eagle新版本中改名为EFAK。

Kafka Eagle 安装步骤

步骤1 下载**Kafka Eagle**，此处以EFAK3.0.1版本为例，具体以实际为准。



例如获取到kafka-eagle-bin-3.0.1.tar.gz软件包。

步骤2 登录FusionInsight Manager，选择“集群 > 服务 > Kafka > 配置 > 全部配置”搜索并修改参数“KAFKA_JMX_IP”的值为“${BROKER_IP}”。

图 4-34 修改 Kafka 参数

步骤3 修改完成后，单击左上方“保存”，在弹出的对话框中单击“确定”保存配置。

步骤4 单击“概览”页签，选择右上方“更多 > 重启服务”重启Kafka服务。

步骤5 以**root**用户登录集群主节点，将获取到的EFAK安装包kafka-eagle-bin-3.0.1.tar.gz放到集群目录下，例如“/opt”，执行以下命令解压。

cd /opt

tar -xvf kafka-eagle-bin-3.0.1.tar.gz

cd kafka-eagle-bin-3.0.1

tar -xvf efak-web-3.0.1-bin.tar.gz

步骤6 在“opt”目录下新建目录，例如“efak”，并将“efak-web-3.0.1”复制到“/opt/efak”目录下。

mkdir /opt/efak

cp -r /opt/kafka-eagle-bin-3.0.1/efak-web-3.0.1 /opt/efak/

步骤7 添加环境变量。

vi /etc/profile

新增“export KE_HOME”参数，参数值为efak-web-3.0.1文件所在路径（例如“/opt/efak/efak-web-3.0.1”）；在“export PATH”参数值后添加“$KE_HOME/bin”。例如：

```
export KE_HOME=/opt/efak/efak-web-3.0.1
export PATH=$PATH:$KE_HOME/bin
```

步骤8 修改“system-config.properties”配置文件。

cd /opt/efak/efak-web-3.0.1/conf/

vi system-config.properties

```
#设置集群
kafka.eagle.zk.cluster.alias=cluster1
cluster1.zk.list=10.20.90.24:2181
#cluster2.zk.list=xdn10:2181,xdn11:2181,xdn12:2181
#修改kafka jmx uri的配置
cluster1.efak.jmx.uri=service:jmx:rmi:///jndi/rmi://%s/kafka
#修改kafka mysql jdbc driver address数据库相关的配置
efak.driver=com.mysql.cj.jdbc.Driver
efak.url=jdbc:mysql://IP:Port/ke?
useUnicode=true&characterEncoding=UTF-8&zeroDateTimeBehavior=convertToNull
efak.username=root
efak.password=XXX
```

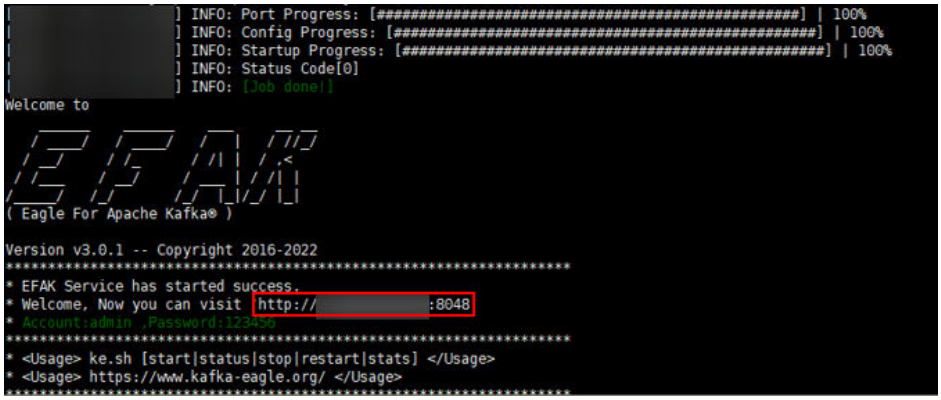
说明

- “cluster1.zk.list”的值是Kafka组件参数“metrics.reporter.zookeeper.url”的值，具体可通过登录FusionInsight Manager，选择“集群 > 服务 > Kafka > 配置 > 全部配置”搜索参数“metrics.reporter.zookeeper.url”查看。
- “efak.url”的值为MySQL JDBC连接字符串，具体以实际为准。
- “efak.username”的值为连接数据库使用的用户名称。
- “efak.password”的值为连接数据库使用的用户名称所对应的密码。

步骤9 启动EFAK服务。

sh /opt/efak/efak-web-3.0.1/bin/ke.sh start

启动成功显示如下，获取EFAK WebUI登录地址。



步骤10 使用获取到的登录地址，访问EFAK WebUI界面。

说明

访问EFKA WebUI界面默认初始账号密码admin/123456

登录后可以查看Kafka集群监控页面、Topic监控页面、Consumer监控页面，例如：

图 4-35 Kafka 集群监控

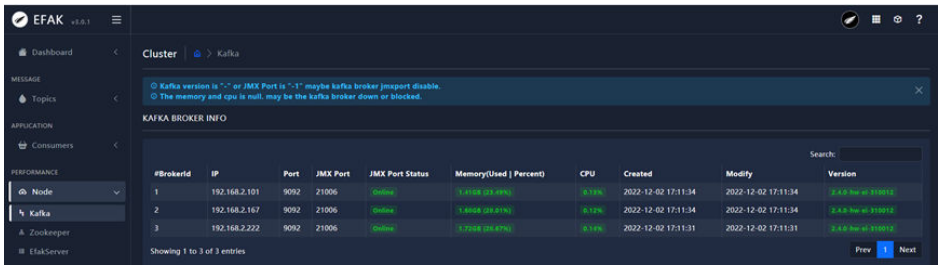


图 4-36 Topic 监控

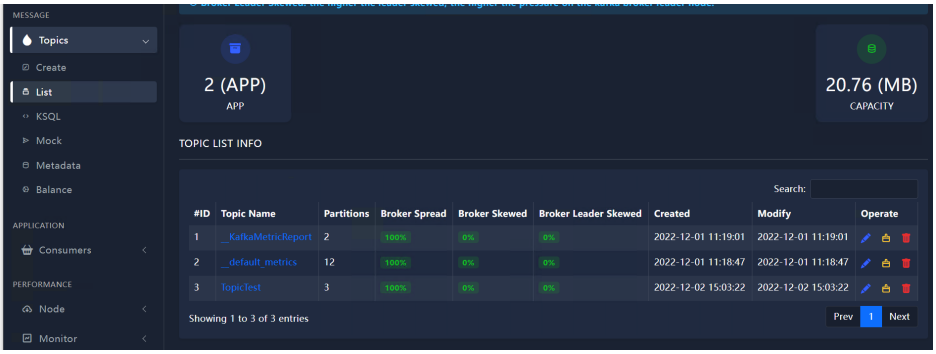
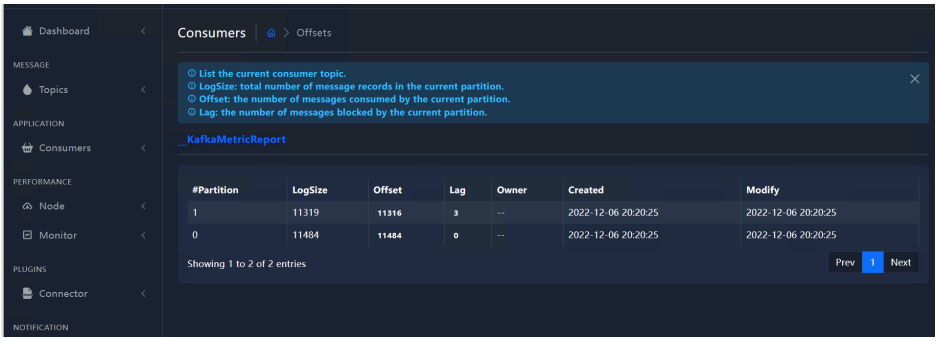


图 4-37 Consumer 监控



----结束

常见问题

问题现象:

无法获取Kafka CPU与内存监控信息日志提示。

```
java.io.IOException cannot be cast to javax.management.remote.JMXConnector
```

问题原因：

jmx地址配置错误。默认jmx地址为：

```
cluster1.efak.jmx.uri=service:jmx:rmi:///jndi/rmi://%s/jmxrmi
```

解决办法：

MRS内Kafka jmx名称为kafka，需要设置为：

cluster1.efak.jmx.uri=service:jmx:rmi:///jndi/rmi://%s/**kafka**

4.12 MRS 对接 Jupyter Notebook

4.12.1 方案概述

在MRS服务中可以配合Jupyter Notebook使用PySpark，能够提高机器学习、数据探索和ETL应用开发效率。

本实践指导用户如何在MRS集群中配置Jupyter Notebook来使用Pyspark。

具体流程如下：

1. **在集群外节点安装客户端**
2. **安装Python3**
3. **安装Jupyter Notebook**
4. **验证Jupyter Notebook访问MRS**

说明

本实践仅适用于MRS 3.x及之后版本，且在集群外客户端节点中安装Python3。

4.12.2 在集群外节点安装客户端

步骤1 准备一台不属于MRS集群的Linux弹性云服务器，绑定一个弹性IP，参考**集群外节点安装客户端**章节安装集群客户端，例如安装目录为“/opt/client”。

步骤2 确认MRS集群是否开启了Kerberos认证。

- 是，执行**步骤3**。
- 否，执行**安装Python3**。

步骤3 登录集群的Manager界面。

步骤4 单击“系统 > 权限 > 用户”，创建一个业务用户。

用户类型为人机用户，用户组选择“hadoop”，主组选择“hadoop”，角色选择“Manager_operator”。

例如创建的用户为“mrs-test”。

图 4-38 创建 MRS 业务用户

步骤5 使用**root**用户，登录到集群客户端节点，执行如下命令配置环境变量并进行认证。

source /opt/client/bigdata_env

kinit *mrs-test*

说明

第一次认证需要修改用户密码。

----结束

4.12.3 安装 Python3

说明

本章节仅适用于**集群外客户端**节点安装Python3。

步骤1 使用**root**用户，登录集群外客户端节点，执行如下命令，检查是否安装了Python3。

python3 --version

```
[root@ecs-notebook FusionInsight_Cluster_1_Services_ClientConfig]# python3 --version
-bash: python3: command not found
```

- 是，执行**步骤8**。
- 否，执行**步骤2**。

步骤2 安装Python，此处以Python 3.6.6为例。

1. 执行如下命令，安装相关依赖：
   yum install zlib zlib-devel zip -y
   yum install gcc-c++
   yum install openssl-devel
   yum install sqlite-devel -y
   如果pandas库需要额外安装如下依赖：
   yum install -y xz-devel
   yum install bzip2-devel
2. 下载对应Python版本源码。
   wget https://www.python.org/ftp/python/3.6.6/Python-3.6.6.tgz
3. 执行如下命令，解压python源码压缩包，例如下载在“opt”目录下。
   cd /opt
   tar -xvf Python-3.6.6.tgz
4. 创建Python的安装目录，此处以“/opt/python36”为例。
   mkdir /opt/python36
5. 编译Python。
   cd /opt/python-3.6.6
   ./configure --prefix=/opt/python36
   执行成功，显示结果如下：

```
configure: creating ./config.status
config.status: creating Makefile.pre
config.status: creating Modules/Setup.config
config.status: creating Misc/python.pc
config.status: creating Misc/python-config.sh
config.status: creating Modules/ld_so_aix
config.status: creating pyconfig.h
creating Modules/Setup
creating Modules/Setup.local
creating Makefile


If you want a release build with all stable optimizations active (PGO, etc),
please run ./configure --enable-optimizations
```

   执行**make -j8**命令，执行成功，显示结果如下：

```
creating build/scripts-3.6
copying and adjusting /tmp/python366/Python-3.6.6/Tools/scripts/pydoc3 -> build/scripts-3.6
copying and adjusting /tmp/python366/Python-3.6.6/Tools/scripts/idle3 -> build/scripts-3.6
copying and adjusting /tmp/python366/Python-3.6.6/Tools/scripts/2to3 -> build/scripts-3.6
copying and adjusting /tmp/python366/Python-3.6.6/Tools/scripts/pyvenv -> build/scripts-3.6
changing mode of build/scripts-3.6/pydoc3 from 644 to 755
changing mode of build/scripts-3.6/idle3 from 644 to 755
changing mode of build/scripts-3.6/2to3 from 644 to 755
changing mode of build/scripts-3.6/pyvenv from 644 to 755
renaming build/scripts-3.6/pydoc3 to build/scripts-3.6/pydoc3.6
renaming build/scripts-3.6/idle3 to build/scripts-3.6/idle3.6
renaming build/scripts-3.6/2to3 to build/scripts-3.6/2to3-3.6
renaming build/scripts-3.6/pyvenv to build/scripts-3.6/pyvenv-3.6
```

   执行**make install**命令，执行成功，显示结果如下：

```
rm -f /opt/python36/share/man/man1/python3.1
(cd /opt/python36/share/man/man1; ln -s python3.6.1 python3.1)
if test "xupgrade" != "xno"  ; then \
        case upgrade in \
                upgrade) ensurepip="--upgrade" ;; \
                install|*) ensurepip="" ;; \
        esac; \
         ./python -E -m ensurepip \
                $ensurepip --root=/ ; \
fi
Looking in links: /tmp/tmp6ldv525m
Collecting setuptools
Collecting pip
Installing collected packages: setuptools, pip
Successfully installed pip-10.0.1 setuptools-39.0.1
```

6. 执行如下命令，配置Python环境变量。

   export PYTHON_HOME=/opt/python36

   export PATH=$PYTHON_HOME/bin:$PATH

7. 执行**python3 --version**命令，显示结果如下，表示Python已经安装完成。

   Python 3.6.6

步骤3 验证Python3。

pip3 install helloword

python3

import helloworld

helloworld.say_hello("test")

```
[root@ecs-notebook Python-3.6.6]# pip3 install helloword
Collecting helloword
  Downloading https://files.pythonhosted.org/packages/1b/bf/f0f69f122158e0e98b5d95987a7ef5add3f8a340c6eb78d5871f855ca04e/helloword-0.0.1-py3-none-any.whl
Installing collected packages: helloword
Successfully installed helloword-0.0.1
You are using pip version 10.0.1, however version 21.3.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
[root@ecs-notebook Python-3.6.6]# python3
Python 3.6.6 (default, Dec 15 2021, 06:12:40)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-44)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import helloworld
helloworld.say_hello("test")Hello, Sara!
>>>
'Hello, test!'
>>>
```

步骤4 测试安装第三方Python库（如pandas、sklearn）。

pip3 install pandas

```
[root@ecs-mrs-test Python-3.6.6]# pip3 install pandas
Collecting pandas
  Downloading https://files.pythonhosted.org/packages/c3/e2/00cacecafbab071c787019f00ad84ca3185952f6bb9bca9550ed83870d4d/pandas-1.1.5-cp36-cp36m-man
x86_64.whl (9.5MB)
    100% |                                | 9.5MB 6.5MB/s
Collecting pytz>=2017.2 (from pandas)
  Downloading https://files.pythonhosted.org/packages/60/2e/dec1cc18c51b8df33c7c4d0a321b084cf38e1733b98f9d15018880fb4970/pytz-2022.1-py2.py3-none-an
503kB)
    100% |                                | 512kB 47.2MB/s
Collecting python-dateutil>=2.7.3 (from pandas)
  Downloading https://files.pythonhosted.org/packages/36/7a/87837f39d0296e723bb9b62bbb257d0355c7f6128853c78955f57342a56d/python_dateutil-2.8.2-py2.p
-any.whl (247kB)
    100% |                                | 256kB 54.5MB/s
Collecting numpy>=1.15.4 (from pandas)
  Downloading https://files.pythonhosted.org/packages/45/b2/6c7545bb7a38754d63048c7696804a0d947328125d81bf12beaa692c3ae3/numpy-1.19.5-cp36-cp36m-man
_x86_64.whl (13.4MB)
    100% |                                | 13.4MB 4.2MB/s
Collecting six>=1.5 (from python-dateutil>=2.7.3->pandas)
  Downloading https://files.pythonhosted.org/packages/d9/5a/e7c31adbe875f2abbb91bd84cf2dc52d792b5a01506781dbcf25c91daf11/six-1.16.0-py2.py3-none-any
Installing collected packages: pytz, six, python-dateutil, numpy, pandas
Successfully installed numpy-1.19.5 pandas-1.1.5 python-dateutil-2.8.2 pytz-2022.1 six-1.16.0
You are using pip version 10.0.1, however version 21.3.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
```

pip3 install backports.lzma

```
[root@ecs-mrs-test Python-3.6.6]# pip3 install backports.lzma
Collecting backports.lzma
  Using cached https://files.pythonhosted.org/packages/21/0f/1a9990233076d48aa2084100ba289ca162975e73a688f3a56c0ee2bb441a/backports.lzma-0.0.14.tar.gz
Installing collected packages: backports.lzma
  Running setup.py install for backports.lzma ... done
Successfully installed backports.lzma-0.0.14
You are using pip version 10.0.1, however version 21.3.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
```

pip3 install sklearn


```
[root@ecs-mrs-test Python-3.6.6]# pip3 install sklearn
Collecting sklearn
  Downloading https://files.pythonhosted.org/packages/1e/7a/dbb3be0ce9bd5c8b7e3d87328e79063f8b263b2b1bfa4774cb1147bfcd3f/sklearn-0.0.tar.
Collecting scikit-learn (from sklearn)
  Downloading https://files.pythonhosted.org/packages/f5/ef/bcd79e8d59250d6e8478eb1290dc6e05be42b3be8a86e3954146adbc171a/scikit_learn-0.2
ylinux1_x86_64.whl (20.0MB)
    100% |                                | 20.0MB 3.4MB/s
Collecting joblib>=0.11 (from scikit-learn->sklearn)
  Downloading https://files.pythonhosted.org/packages/3e/d5/0163eb0cfa0b673aa4fe1cd3ea9d8a81ea0f32e50807b0c295871e4aab2e/joblib-1.1.0-py2
(306kB)
    100% |                                | 307kB 46.5MB/s
Requirement already satisfied: scipy>=0.19.1 in /root/.local/lib/python3.6/site-packages (from scikit-learn->sklearn) (1.5.4)
Collecting threadpoolctl>=2.0.0 (from scikit-learn->sklearn)
  Downloading https://files.pythonhosted.org/packages/61/cf/6e354304bcb9c6413c4e02a747b600061c21d38ba51e7e544ac7bc66aecc/threadpoolctl-3.
nl
Requirement already satisfied: numpy>=1.13.3 in /opt/python36/lib/python3.6/site-packages (from scikit-learn->sklearn) (1.19.5)
Installing collected packages: joblib, threadpoolctl, scikit-learn, sklearn
  Running setup.py install for sklearn ... done
Successfully installed joblib-1.1.0 scikit-learn-0.24.2 sklearn-0.0 threadpoolctl-3.1.0
You are using pip version 10.0.1, however version 21.3.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
```

步骤5 执行命令**python3 -m pip list**，查看安装结果。

```
[root@ecs-mrs-test Python-3.6.6]# python3 -m pip list
Package         Version
--------------- -------
cycler          0.11.0
joblib          1.1.0
kiwisolver      1.3.1
numpy           1.19.5
pandas          1.1.5
pip             10.0.1
pyparsing       3.0.7
python-dateutil 2.8.2
pytz            2022.1
scikit-learn    0.24.2
scipy           1.5.4
setuptools      39.0.1
six             1.16.0
sklearn         0.0
threadpoolctl   3.1.0
```

步骤6 打包Python.zip

cd /opt/python36/

zip -r python36.zip ./*

步骤7 上传到HDFS指定目录。

hdfs dfs -mkdir /user/python

hdfs dfs -put python36.zip /user/python

步骤8 配置MRS客户端。

进入Spark客户端安装目录“/opt/client/*Spark2x*/spark/conf”，在“spark-defaults.conf”配置文件如下参数。

```
spark.pyspark.driver.python=/usr/bin/python3
spark.yarn.dist.archives=hdfs://hacluster/user/python/python36.zip#Python
```

----结束

4.12.4 安装 Jupyter Notebook

步骤1 使用**root**用户登录客户端节点，执行如下命令安装Jupyter Notebook。

pip3 install jupyter notebook

显示结果如下，表示安装成功：

```
Successfully installed MarkupSafe-2.0.1 Send2Trash-1.8.0 argon2-cffi-21.3.0 argon2-cffi-bindings-21.2.0 async-generator-1.10 attrs-21.2.0 backcall-0.2.0 bleach-4.1.0 cffi-1.15.0 dataclasses-0.8 decorator-5.1.0 defusedxml-0.7.1 entrypoints-0.3 importlib-metadata-4.8.2 ipykernel-5.5.6 ipython-7.16.2 ipython-genutils-0.2.0 ipywidgets-7.6.5 jedi-0.17.2 jinja2-3.0.3 jsonschema-4.0.0 jupyter-1.0.0 jupyter-client-7.1.0 jupyter-console-6.4.0 jupyter-core-4.9.1 jupyterlab-pygments-0.1.2 jupyterlab-widgets-1.0.2 mistune-0.8.4 nbclient-0.5.9 nbconvert-6.0.7 nbformat-5.1.3 nest-asyncio-1.5.4 notebook-6.4.6 packaging-21.3 pandocfilters-1.5.0 parso-0.7.1 pexpect-4.8.0 pickleshare-0.7.5 prometheus-client-0.12.0 prompt-toolkit-3.0.24 ptyprocess-0.7.0 pycparser-2.21 pygments-2.10.0 pyparsing-3.0.6 pyrsistent-0.18.0 python-dateutil-2.8.2 pyzmq-22.3.0 qtconsole-5.2.2 qtpy-1.11.3 six-1.16.0 terminado-0.12.1 testpath-0.5.0 tornado-6.1 traitlets-4.3.3 typing-extensions-4.0.1 wcwidth-0.2.5 webencodings-0.5.1 widgetsnbextension-3.5.2 zipp-3.6.0
You are using pip version 10.0.1, however version 21.3.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
```

步骤2 为保障系统安全，需要生成一个密文密码用于登录Jupyter，放到Jupyter Notebook的配置文件中。

执行如下命令，需要输入两次密码：（进行到Out[3]退出）

ipython

```
[root@ecs-notebook python36]# ipython
Python 3.6.6 (default, Dec 20 2021, 09:32:25)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.16.2 -- An enhanced Interactive Python. Type '?' for help.
In [1]: from notebook.auth import passwd
In [2]: passwd()
Enter password:
Verify password:
Out[2]: 'argon2:$argon2id$v=19$m=10240,t=10,p=8$g14BqLddl927n/unsyPlLQ
$YmoKJzbUfNG7LcxylJzm90bgbKWUIiHy6ZV+ObTzdcA
```

步骤3 执行如下命令生成Jupyter配置文件。

jupyter notebook --generate-config

步骤4 修改配置文件。

vi ~/.jupyter/jupyter_notebook_config.py

添加如下配置：

```
# -*- coding: utf-8 -*-
c.NotebookApp.ip='*' #此处填写ecs对应的内网IP
c.NotebookApp.password = u'argon2:$argon2id$v=19$m=10240,t=10,p=8$NmoAVwd8F6vFP2rX5ZbV7w
$SyueJoC0a5TbCuHYzqfSx1vQcFvOTTryR+0uk2MNNZA' # 填写步骤2，Out[2]密码生成的密文
c.NotebookApp.open_browser = False # 禁止自动打开浏览器
c.NotebookApp.port = 9999 # 指定端口号
c.NotebookApp.allow_remote_access = True
```

----结束

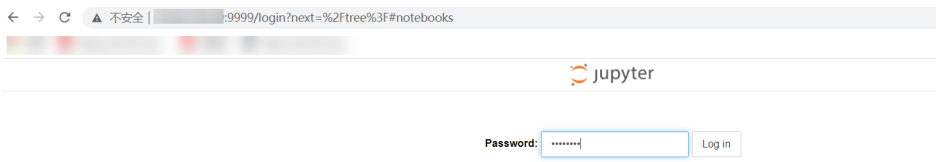
4.12.5 验证 Jupyter Notebook 访问 MRS

步骤1 在客户端节点执行如下命令，启动Jupyter Notebook。

PYSPARK_PYTHON=./Python/bin/python3 PYSPARK_DRIVER_PYTHON=jupyter-notebook PYSPARK_DRIVER_PYTHON_OPTS="--allow-root" pyspark --master yarn --executor-memory 2G --driver-memory 1G

步骤2 在浏览器中输入“*弹性IP地址*:9999”地址，登录到Jupyter WebUI（保证ECS的安全组对外放通本地公网IP和9999端口），登录密码为步骤2设置的密码。

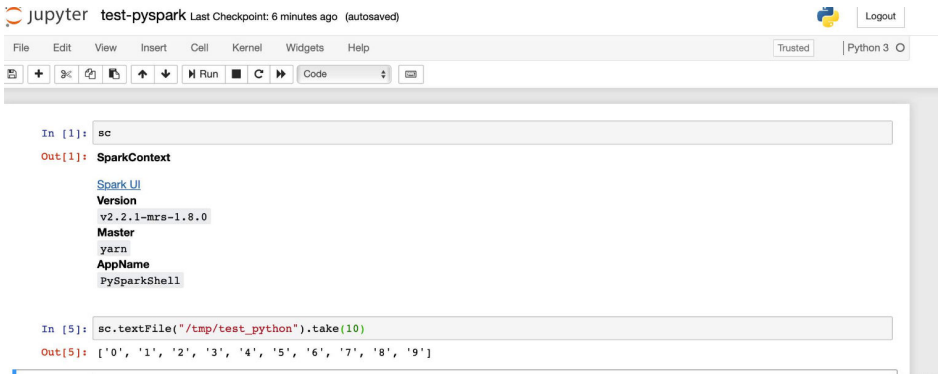
图 4-39 登录 Jupyter WebUI



步骤3 创建代码。

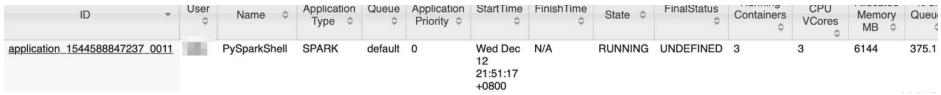
创建一个新的python3任务，使用Spark读取文件。

图 4-40 创建 Python 任务



登录到集群Manager界面，在Yarn的WebUI页面上查看提交的pyspark应用。

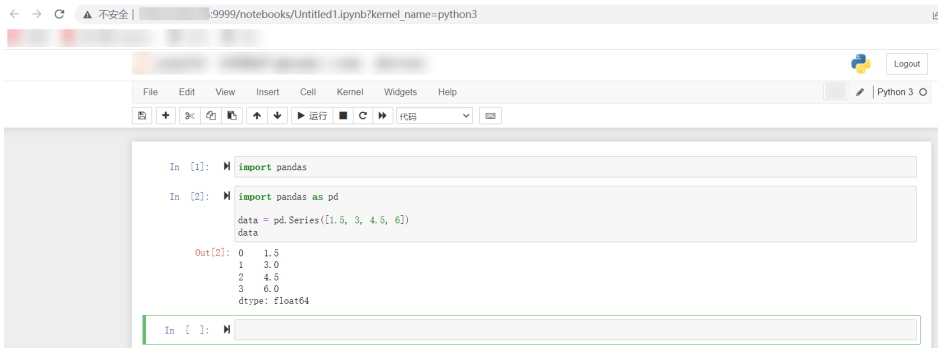
图 4-41 查看任务运行情况



ID	User	Name	Application Type	Queue	Application Priority	StartTime	FinishTime	State	FinalStatus	Running Containers	CPU VCores	Allocated Memory MB	Queue
application_1544588847237_0011	[redacted]	PySparkShell	SPARK	default	0	Wed Dec 12 21:51:17 +0800	N/A	RUNNING	UNDEFINED	3	3	6144	375.1

步骤4 验证pandas库调用。

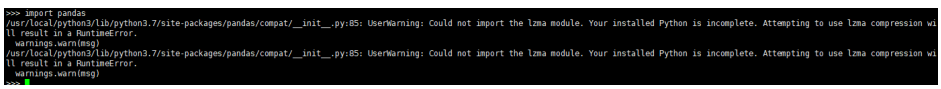
图 4-42 验证 pandas



----结束

常见问题

pandas本地import使用时，报错如下：



参考以下步骤进行处理：

步骤1 执行命令**python -m pip install backports.lzma**安装lzma模块，如下图所示：

```
[root@master ~]# python -m pip install backports.lzma
Looking in indexes: http://mirrors.aliyun.com/pypi/simple/
Requirement already satisfied: backports.lzma in /usr/local/python3/lib/python3.7/site-packages (0.0.14)
You are using pip version 10.0.1, however version 19.3.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
```


步骤2 进入“/usr/local/python3/lib/python3.6”目录（机器不同，目录也有所不同，可以通过**which**命令来查找当前运行python是使用的那个目录的），然后编辑lzma.py文件。

将：

```
from _lzma import *
from _lzma import _encode_filter_properties, _decode_filter_properties
```

更改为：

```
try:
    from _lzma import *
    from _lzma import _encode_filter_properties, _decode_filter_properties
except ImportError:
    from backports.lzma import *
    from backports.lzma import _encode_filter_properties, _decode_filter_properties
```

修改前：

```
"""Interface to the liblzma compression library.

This module provides a class for reading and writing compressed files,
classes for incremental (de)compression, and convenience functions for
one-shot (de)compression.

These classes and functions support both the XZ and legacy LZMA
container formats, as well as raw compressed data streams.
"""

__all__ = [
    "CHECK_NONE", "CHECK_CRC32", "CHECK_CRC64", "CHECK_SHA256",
    "CHECK_ID_MAX", "CHECK_UNKNOWN",
    "FILTER_LZMA1", "FILTER_LZMA2", "FILTER_DELTA", "FILTER_X86", "FILTER_IA64",
    "FILTER_ARM", "FILTER_ARMTHUMB", "FILTER_POWERPC", "FILTER_SPARC",
    "FORMAT_AUTO", "FORMAT_XZ", "FORMAT_ALONE", "FORMAT_RAW",
    "MF_HC3", "MF_HC4", "MF_BT2", "MF_BT3", "MF_BT4",
    "MODE_FAST", "MODE_NORMAL", "PRESET_DEFAULT", "PRESET_EXTREME",

    "LZMACompressor", "LZMADecompressor", "LZMAFile", "LZMAError",
    "open", "compress", "decompress", "is_check_supported",
]

import builtins
import io
import os
from _lzma import *
from _lzma import _encode_filter_properties, _decode_filter_properties
import _compression
```

修改后：


```
These classes and functions support both the XZ and legacy LZMA
container formats, as well as raw compressed data streams.
"""

__all__ = [
    "CHECK_NONE", "CHECK_CRC32", "CHECK_CRC64", "CHECK_SHA256",
    "CHECK_ID_MAX", "CHECK_UNKNOWN",
    "FILTER_LZMA1", "FILTER_LZMA2", "FILTER_DELTA", "FILTER_X86", "FILTER_IA64",
    "FILTER_ARM", "FILTER_ARMTHUMB", "FILTER_POWERPC", "FILTER_SPARC",
    "FORMAT_AUTO", "FORMAT_XZ", "FORMAT_ALONE", "FORMAT_RAW",
    "MF_HC3", "MF_HC4", "MF_BT2", "MF_BT3", "MF_BT4",
    "MODE_FAST", "MODE_NORMAL", "PRESET_DEFAULT", "PRESET_EXTREME",

    "LZMACompressor", "LZMADecompressor", "LZMAFile", "LZMAError",
    "open", "compress", "decompress", "is_check_supported",
]

import builtins
import io
import os
#from _lzma import *
#from _lzma import _encode_filter_properties, _decode_filter_properties
try:
    from _lzma import *
    from _lzma import _encode_filter_properties, _decode_filter_properties
except ImportError:
    from backports.lzma import *
    from backports.lzma import _encode_filter_properties, _decode_filter_properties
import _compression
```

步骤3 保存退出，然后再次import。

```
[root@master python3.7]# python
Python 3.7.0 (default, Oct 26 2019, 01:19:22)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-36)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import pandas
>>>
```

----结束

5 ClickHouse 设计开发规范

5.1 规范概述

内容介绍

本文主要描述ClickHouse数据管理全生命周期过程中，数据库规划、建模设计、开发、调优、运维的规则建议和指导。

通过这些约束和建议，指导开发者在ClickHouse数据库开发使用过程中能够最大化发挥数据库的优势，保障ClickHouse数据库高性能、稳定可靠运行。用户可更专注于上层业务，释放数据更大的价值。

本文档主要包含以下内容：

项目	描述
数据库规划	集群业务规划、容量规划、数据分布。
数据库设计	Database设计、宽表设计、分布式表设计、本地表设计、分区设计、索引设计、物化视图设计。
数据库开发	简单查询、聚合查询、join查询、数据增/删/改等SQL开发。
数据库调优	调优思路、参数调优、系统调优、SQL改写调优。
数据库运维	监控、告警、日志、系统表/视图。

适用范围

本文档适用于ClickHouse数据库设计、数据库开发、数据库测试、数据库运维以及DBA和业务使用人员。

5.2 集群规划

5.2.1 ClickHouse 集群业务规划

- 集群规模

  建议单集群不超过256节点规模。

- 集群负载

  对于不同业务负载的业务，需要分开集群部署，便于不同负载的业务进行资源隔离。

- 集群并发

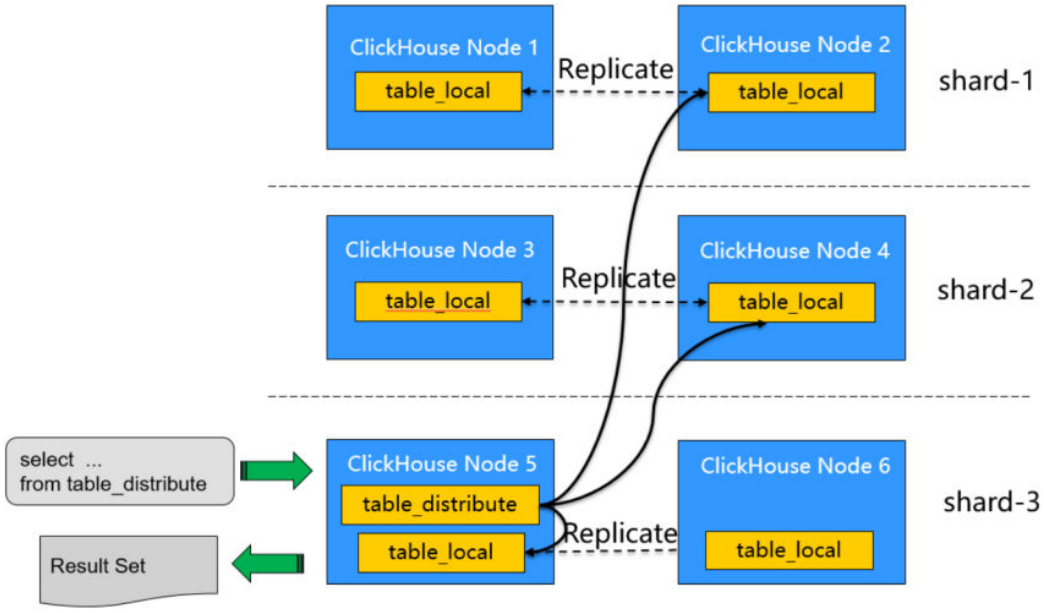
  由于ClickHouse单个SQL会最大化使用每个主机上的CPU/内存/IO资源，对于复杂SQL查询（复杂聚合、复杂join计算）能够支持50~100并发，对于简单的SQL查询，支持100~200左右查询。

  如果集群有混合负载（要求极致性能的点查/范围查询和有大数据量聚合及join查询），建议将不同类型的负载拆分到不同集群；对于集群规划有远远超过100个并发业务系统，也需要设计将业务分摊到不同的集群。

5.2.2 ClickHouse 数据分布设计

Shard 和副本概念介绍

图 5-1 ClickHouse 集群架构图



从横向来看ClickHouse数据库集群，所有数据都会平均分布到多个shard分片中进行保存，数据平均分布后，保证了查询的高度并行性，以提升数据的查询性能。

从纵向来看，每个shard内部有多个副本组成，保证分片数据的高可靠性，以及计算的高可靠性。

数据分布设计

- Shard数据分片均匀分布

  建议用户的数据均匀分布到集群中的多个shard分片，如图5-1所示有3个分片。

假如有30 GB数据需要写入到集群中，需要将30 GB数据均匀切分后分别放到shard-1、shard-2和shard-3的3个分片节点中，以充分发挥MPP查询时并行计算能力，避免数据在shard间倾斜计算出现木桶效应，导致SQL查询性能较差。

可通过弹性负载均衡（Elastic Load Balance，简称ELB）访问ClickHouse，来实现数据均匀。

- Shard内数据副本高可靠存储

  数据写入单shard中的一个副本后，ClickHouse会自动异步将数据同步到其他副本，如图5-1中的shard-3。

  如果将10GB数据导入ClickHouse Node 5节点副本，ClickHouse会自动异步将数据同步到ClickHouse Node 6节点副本，保证shard-3分片数据的高可靠性存储。

5.2.3 ClickHouse 容量规划设计

为了能够更好的发挥ClickHouse分布式查询能力，在集群规划阶段需要合理设计集群数据分布存储。

当前ClickHouse能力为单机磁盘容量达到80%后会上报告警信息，磁盘容量达90%后集群会处于只读状态。

出现磁盘告警信息后需要考虑是否是容量不足问题，如果是容量不足问题需要尽快考虑集群扩容，提升集群整体容量存储。

ClickHouse节点及容量规划如下：

- 磁盘规划

  由于ClickHouseServer业务数据主要存储在本地磁盘上，数据量可能会随着集群使用时间增长而增长，通常建议ClickHouse数据盘单独挂载，元数据盘共享第一个数据盘目录。

- 磁盘实际容量

  由于磁盘存在1MB = 1024KB或者1000KB的不同算法，一般来说，磁盘实际可用容量 = 磁盘标注容量 * 0.9。

  例如磁盘标注容量为1.2 TB，实际容量为1200 * 0.9 = 1080 GB。

- 计算公式

  假设历史数据量为H，每日增量为A，单节点磁盘容量为C，数据保留M天，集群副本数为R，则ClickHouseServer物理节点数计算公式如下：

  ClickHouseServer物理节点数N = [R * (H + A * M)] / C

5.2.4 ClickHouse 依赖服务设计

为了保证ClickHouse服务的稳定，需要提早规划好对于底层依赖服务的设计，主要是ZooKeeper，尤其是在使用replicated*系列表引擎的场景下。

1. ZooKeeper默认部署在MRS集群的Master节点，根据节点CPU和内存规格，调整ZooKeeper实例的最大可用内存。

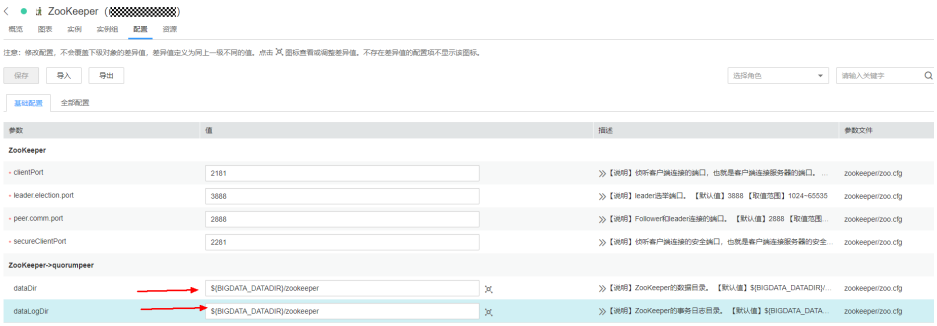
   登录MRS集群的FusionInsight Manager界面，单击“集群 > 服务 > ZooKeeper > 配置 > 全部配置 > quorumpeer > 系统”，调整“GC_OPTS”参数：-Xmx*最大内存数*GB。

   最大内存数参考值：master节点内存-16GB * 0.65（保守估计值）

   说明

   修改完成后需要重启ZooKeeper服务。

2. 修改ZooKeeper的数据盘和日志盘默认配置，改为不同磁盘。



3. 完成后同步修改ClickHouse服务的ZooKeeper相关配置。

   登录MRS集群的FusionInsight Manager界面，单击“集群 > 服务 > ClickHouse > 配置 > 全部配置 > ClickHouse > Zookeeper”。配置调整后通常不需要重启Clickhouse服务。

5.3 数据库设计

5.3.1 DataBase 设计

业务隔离设计-各业务分库设计

在业务规划时，不同业务归属于不同数据库，便于后续对应用户关联的数据库下表、视图等数据库对象权限的分离管理和维护。

业务隔离设计-不要在 system 库中创建业务表

system数据库是ClickHouse默认的系统数据库，默认数据库中的系统表记录的是系统的配置、元数据等的信息数据。

业务在使用ClickHouse的时候，需要指定自己业务的数据库进行连接和使用，业务相关的表创建在自己业务库中，不要将业务的表创建在系统数据库中，避免对系统数据库造成不必要的影响。

命名规范设计规则

- 所有命名采用26个英文字母和0~9这10个自然数，加上下划线_组成，一般不要出现其他符号。
- 对象名尽量的短，能表达业务所使用数据库含义即可，以英文单词、单词组合或英文单词缩写组成，不以数字或下划线_开头。
- 命名尽量不要使用SQL保留字，请注意大小写敏感。如果必须要使用一些保留关键字，请使用双引号（""）或者反引号（`）进行转义。

5.3.2 表引擎使用场景选择

ClickHouse中最强大的表引擎当属MergeTree（合并树）引擎及该系列其他引擎，根据业务场景选择合适的引擎。

表引擎选择建议

- 自助报表分析、行为数据分析，在不涉及重复数据聚合的情况下，建议使用ReplicatedMergeTree表引擎。
- 涉及到物化视图等聚合函数的场景，建议使用ReplicatedAggregatingMergeTree表引擎。
- 经常有数据去重或有update修改数据的场景下，建议使用ReplacingMergeTree表引擎，配合使用argMax函数获取最新数据。

表 5-1 应用场景列表

引擎名称	应用场景
MergeTree	ClickHouse中最重要的引擎，基于分区键（partitioning key）的数据分区分块存储、前缀稀疏索引（order by和primary key）。
ReplacingMergeTree	相对于MergeTree，它会用最新的数据覆盖具有相同主键的重复项。 删除老数据的操作是在分区异步merge的时候进行处理，只有同一个分区的数据才会被去重，分区间及shard间重复数据不会被去重，所以应用侧想要获取到最新数据，需要配合argMax函数一起使用。
SummingMergeTree	当合并SummingMergeTree表的数据片段时，ClickHouse会把所有具有相同主键的行进行汇总，将同一主键的行替换为包含sum后的一行记录。 如果主键的组合方式使得单个键值对应于大量的行，则可以显著的减少存储空间并加快数据查询的速度。
AggregatingMergeTree	该引擎继承自MergeTree，并改变了数据片段的合并逻辑。 ClickHouse会将一个数据片段内所有具有相同主键（准确的说是排序键）的行替换成一行，这一行会存储一系列聚合函数的状态。可以使用AggregatingMergeTree表引擎来做增量数据的聚合统计，包括物化视图的数据聚合。
CollapsingMergeTree	在创建时与MergeTree基本一样，除了最后多了一个参数，需要指定Sign位（必须是Int8类型）。 CollapsingMergeTree会异步地删除（折叠）除了特定列Sign1和-1值以外的所有字段的值重复的行。
VersionedCollapsingMergeTree	是CollapsingMergeTree的升级，使用不同的collapsing算法，该算法允许使用多个线程以任何顺序插入数据。

引擎名称	应用场景
Replicated* MergeTree	只有Replicated*MergeTree系列引擎是上面介绍的引擎的多副本版本，为了提升数据和服务的可靠性，建议使用副本引擎： • ReplicatedMergeTree • ReplicatedSummingMergeTree • ReplicatedReplacingMergeTree • ReplicatedAggregatingMergeTree • ReplicatedCollapsingMergeTree • ReplicatedVersionedCollapsingMergeTree • ReplicatedGraphiteMergeTree

5.3.3 宽表设计

5.3.3.1 宽表设计原则

宽表设计原则

由于ClickHouse的宽表查询性能较优，且当前ClickHouse可支持上万列的宽表横向扩展。

在大部分场景下，有大表两表join以及多表join的场景，且多个join的表数据变化更新频率较低，这种情况，建议对多个表join查询逻辑提前进行加工处理，将处理后的数据写入到一个宽表中，宽表中包含所有要查询的数据字段，以供后续应用完全自助OLAP的高性能查询。

表命名规范

数据库表名称命名规则：

- 在数据库中，表名命名要求在当前数据库内唯一。
- 表名要求以字符开始，可以包含字符（a～z，A～Z）、数字（0～9）及下划线（_）。

5.3.3.2 表字段设计

规则

- 不允许用字符类型存放时间或日期类数据，尤其是需要对该日期字段进行运算或者比较的时候。
- 不允许用字符类型存放数值类型的数据，尤其是需要对该数值字段进行运算或者比较的时候。字符串的过滤效率相对于整型或者特定时间类型有下降。

建议

- 不建议表中存储过多的Nullable列，可以考虑字符串使用“NA”，数值型用0作为缺省值。过多使用Nullable将消耗更多内存。

- 建议规划好业务所需的列，必要时可提前预置一些属性列，避免频繁的增删列。
- 数值类型：UInt8/UInt16/UInt32/UInt64、Int8/Int16/Int32/Int64, Float32/Float64等，选择不同长度，性能差别较大。

  建议根据业务场景所需选择最小满足的类型使用。

  - 示例

    ```
    CREATE TABLE counter ON CLUSTER default_cluster
    (
    `when`     DateTime DEFAULT now(),
    `device`    UInt32,
    `value`     Float32,
    `value64`   Float64
    )
    ENGINE = MergeTree
    PARTITION BY toYYYYMM(when)
    ORDER BY (device, when)
    ```

    表中有Float32类型的字段value和Float64的字段value64插入数据的查询表现如下：

    ```
    INSERT INTO counter
    SELECT
    toDateTime('2019-01-01 00:00:00') + toInt64(number / 10) AS when,
    (number % 10) + 1 AS device,
    (device * 3) + (number / 10000) AS value,
    value
    FROM system.numbers
    LIMIT 100000000;
    ```

    往value和value64插入相同的数据，总数据量1亿条。

  - 查询Float32字段

    ```
    SELECT countDistinct(value)
    FROM counter
    WHERE device = 1

    ┌─uniqExact(value)─┐
    │         10000000 │
    └──────────────────┘

    1 rows in set. Elapsed: 0.750 sec. Processed 10.04 million rows, 80.35 MB (13.39 million rows/s., 107.14 MB/s.)
    ```

    耗时：0.750秒。

  - 查询Float64字段

    ```
    SELECT countDistinct(value64)
    FROM counter
    WHERE device = 1

    ┌─uniqExact(value64)─┐
    │           10000000 │
    └────────────────────┘

    1 rows in set. Elapsed: 0.929 sec. Processed 10.04 million rows, 120.52 MB (10.81 million rows/s., 129.76 MB/s.)
    ```

    耗时：0.929秒。

    结果：Float32类型的查询时间比Float64更快。

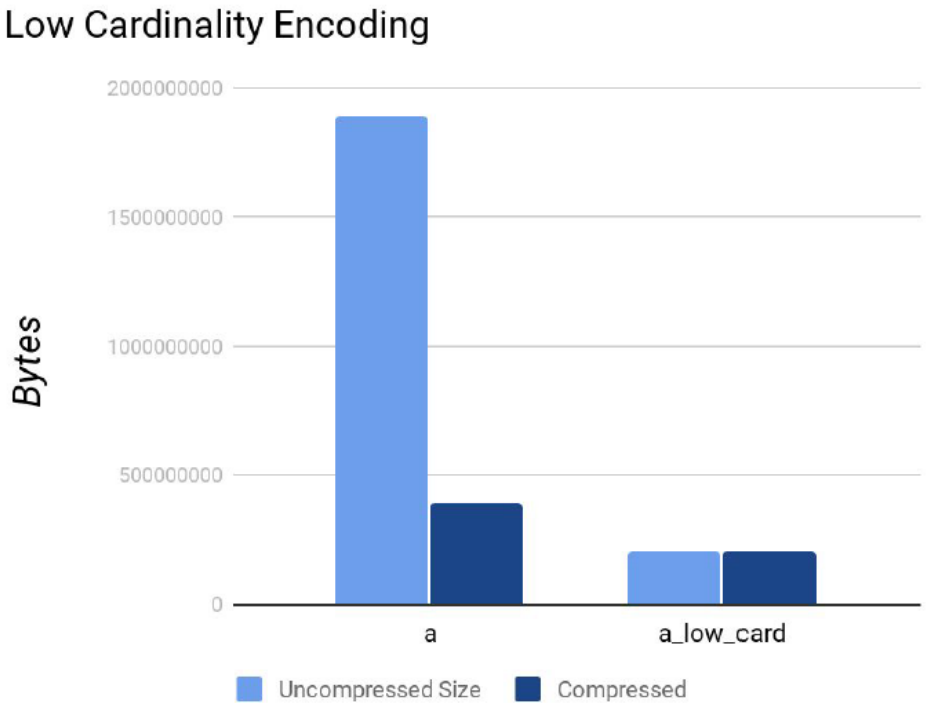
- 低基数维度（基数1万内），建议使用LowCardinality修饰符，提升查询性能。
  - 维度的基数（Cardinality）：指的是该维度在数据集中出现的不同值的个数。例如“国家”是一个维度，如果有200个不同的值，那么此维度的基数就是200。
  - 根据官方建议和实践经验，在维度基数小于1万的时候，对维度字段做LowCardinality编码，导入性能会有略微下降，查询性能提升明显，数据存储空间下降明显。
  - 在默认的情况下，声明了LowCardinality的字段会基于数据生成一个全局字典，并利用倒排索引建立Key和位置的对应关系。如果数据的基数大于8192，也就是说不同的值多于8192个，则会将一个全局字典拆分成多个局部字典（low_cardinality_max_dictionary_size参数控制，默认8192）。

- 示例

```
CREATE TABLE test_codecs ON CLUSTER default_cluster
(
`a` String,
`a_low_card` LowCardinality(String) DEFAULT a
)
ENGINE = MergeTree
PARTITION BY tuple()
ORDER BY tuple();
```

其中，字段a是原生字符串，字段a_low_card基于a做了低基维编码。

- 数据存储的对比



- 查询性能对比

```
SELECT a AS a, count(*) AS c FROM test_codecs
GROUP BY a ORDER BY c ASC LIMIT 10
. . .
10 rows in set. Elapsed: 0.681 sec. Processed 100.00 million
rows, 2.69 GB (146.81 million rows/s., 3.95 GB/s.)
```

Faster

```
SELECT a_lc AS a, count(*) AS c FROM test_codecs
GROUP BY a ORDER BY c ASC LIMIT 10
. . .
10 rows in set. Elapsed: 0.148 sec. Processed 100.00 million
rows, 241.16 MB (675.55 million rows/s., 1.63 GB/s.)
```

查询性能有5倍的提升。

5.3.3.3 本地表设计

规则

- 单表（分布式表）的记录数不要超过万亿，对于万亿以上表的查询，性能较差，且集群维护难度变大。单表（本地表）不超过百亿。
- 表的设计都要考虑到数据的生命周期管理，需要进行TTL表属性设置或定期老化清理表分区数据。
- 单表的字段建议不要超过5000列。

  因为当一次插入的数据大小超过“min_bytes_for_wide_part”（默认值:10485760），ClickHouse写入会按每列1 MB（Nullable类型2MB）来预申请内存，容易出现内存超限的错误：

  ```
  Received exception from server (version 22.3.4):
  Code:241. DB::Exception: Received from localhost:9000. DB::Exception: Memory limit (for query) exceeded: would use 9.31 Gib (attempt to allocate chunk of 1048591 bytes), maximum: 9.31 GiB
  ```

  可以通过调大“min_bytes_for_wide_part”来规避。

参考案例

- MergeTree引擎在建表的时候支持列字段和表级的TTL。

  当列字段中的值过期时，ClickHouse会将其替换成数据类型的默认值。如果分区内，某一列的所有值均已过期，则ClickHouse会从文件系统中删除这个分区目录下的列文件。当表内的数据过期时，ClickHouse会删除所有对应的行。

  在列上配置TTL：

  ```
  CREATE TABLE default.t_column_ttl ON CLUSTER default_cluster
  (
  `did` Int32,
  `app_id` Int32,
  `region` Int32,
  `pt_d` Date,
  `create_time` Datetime,
  `product_desc1` String TTL create_time + toIntervalSecond(10),
  `product_desc2` String TTL create_time + toIntervalMonth(10),
  `product_desc3` String TTL create_time + toIntervalHour(10)
  )
  ENGINE = MergeTree()
  PARTITION BY toYYYYMMDD(pt_d)
  ORDER BY (app_id, region);
  ```

  在表上配置TTL：

  ```
  CREATE TABLE default.t_table_ttl ON CLUSTER default_cluster
  (
  `did` Int32,
  `app_id` Int32,
  `region` Int32,
  `pt_d` Date,
  `create_time` Datetime
  )
  ENGINE = MergeTree()
  PARTITION BY toYYYYMMDD(pt_d)
  ORDER BY (app_id, region)
  TTL create_time + toIntervalMonth(12);
  ```

  TTL详细使用见官网链接：

  https://clickhouse.tech/docs/en/engines/table-engines/mergetree-family/mergetree/#table_engine-mergetree-ttl

- 通过外部系统管理数据的生命周期，定时清理过期数据。

  清理数据SQL命令示例：

DROP TABLE default.table_with_non_default_policy ON CLUSTER default_cluster NO delay; #删除表

ALTER TABLE default.table_with_non_default_policy ON CLUSTER default_cluster drop partition 201901; #删除分区

本地表建表参考：

```
CREATE TABLE default.my_table_local ON CLUSTER default_cluster
(
`did`      Int32,
`app_id`   Int32,
`region`   Int32,
`pt_d`     Date
)
ENGINE = ReplicatedMergeTree('/clickhouse/tables/{shard}/default/my_table_local', '{replica}')
PARTITION BY toYYYYMMDD(pt_d)
PRIMARY KEY(app_id)
ORDER BY (app_id, region)
SETTINGS index_granularity = 8192;
```

- 表引擎选择：

  ReplicatedMergeTree：支持副本特性的MergeTree引擎，也是最常用的表引擎，其他表引擎参考使用场景介绍进行选择。

- ZooKeeper上的表元数据信息存储路径“/clickhouse/tables/{shard}/default/my_table_local”：

  {cluster}表示集群名称，{shard}是分片名称，{replica}是分片中的副本编号，这几个宏变量直接写即可，建表时不需要替换为常量值。

  default：表示创建的表名放到哪个数据库下面，在创建表时需要根据实际情况进行替换。

- on cluster：创建的集群

  建表会创建到集群中所有节点上，否则需要自己手动一个个节点去创建，一个个节点创建过程比较繁琐，创建比较慢；如果在集群中部分节点未创建表，在查询时会遇到无表信息的错误提示。

- no delay：立刻生效

  在删除表或修改表语法中加上no delay，表示立即删除，否则会等8分钟以后进行删除，如果未加no delay语法，删除表后需要立即创建同名的表名可能会遇到错误，创建不成功。

- order by：排序字段

  查询时最常使用且过滤性最高的字段作为排序字段。依次按照访问频度从高到低、维度基数从小到大来排。排序字段不宜太多，建议不超过4个，否则merge的压力会较大。排序字段不允许为null，如果存在null值，需要做数据转换。

- primary key：主键字段

  创建主键索引，值为排序字段的前导列，否则不允许创建表，为访问频率最高的字段创建索引，提升查询性能，查询时会通过索引数据快速的找到数据文件中的数据块所在位置信息。

- partition by：分区字段

  分区键不允许为null，如果字段中有null值，需要做数据转换处理。

- 表级别的参数配置：

  index_granularity：稀疏索引粒度配置，默认是8192，一般不需要修改。

建表定义，参考链接：

https://clickhouse.tech/docs/en/engines/table-engines/mergetree-family/mergetree/

5.3.3.4 分布式表设计

建议

分布式表建表参考：

```
CREATE TABLE default.my_table_dis ON CLUSTER default_cluster
AS mybase.my_table_local
ENGINE = Distributed(default_cluster, default, my_table_local, rand());
```

使用说明

- 分布式表名称：default.my_table_dis。
- 本地表名称：default.my_table_local。
- 通过“AS”关联分布式表和本地表，保证分布式表的字段定义跟本地表一致。
- 分布式表引擎的参数说明：
  - default_cluster：集群名称。
  - default：本地表所在库名。
  - my_table_local：本地表名。
  - rand()：可选参数，分片键（sharding key），可以是表中一列的原始数据（如did），也可以是函数调用的结果。

    如轮训方式：rand()，表示在写入数据时直接将数据插入到分布式表，分布式表引擎会按轮训算法将数据发送到各个分片。

    注意

    该键是写分布式表保证数据均匀分布在各分片的唯一方式。

规则

不建议写分布式表。

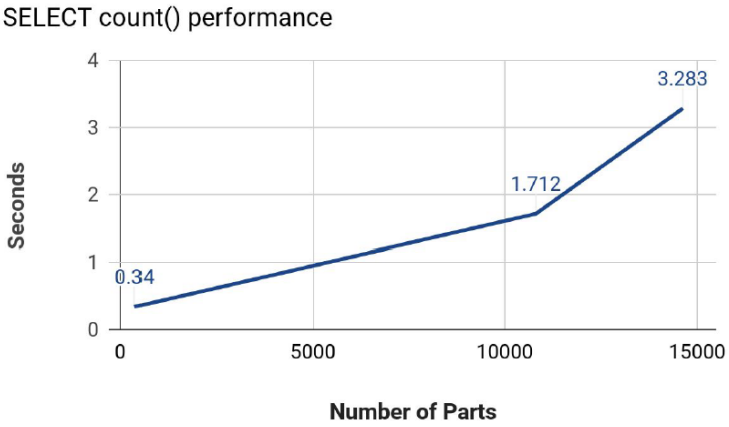
由于分布式表写数据是异步方式，客户端SQL由Balancer路由到一个节点之后，一批写入数据会先落入写入的节点，随后根据分布式表schema定义数据分布规则，将数据异步发送到各个shard的各个副本。整个过程数据异步发送，且数据会在一个节点临时存储，会导致网络、磁盘都会成为瓶颈，且写入成功后不一定能查询到最新一致性数据等问题。

5.3.3.5 分区设计

合理设置分区键，控制分区数在一千以内，分区字段使用整型。

分区 part 数与查询性能关系

图 5-2 分区 part 数与查询性能关系图



分区建议

- 建议使用toYYYYMMDD（pt_d）作为分区键，pt_d是date类型。
- 如果业务场景需要做小时分区，使用pt_d、pt_h做联合分区键，其中pt_h是整型小时数。
- 如果保存多年数据，建议考虑使用月做分区，toYYYYMM（pt_d）。
- 综合考虑数据分区粒度、每个批次提交的数据量、数据的保存周期等因素，合理控制part数量。

5.3.3.6 索引设计

一级索引设计

- 在建表设计时指定主键字段的建议：按查询时最常使用且过滤性最高的字段作为主键。依次按照访问频度从高到低、维度基数从小到大来排列。数据是按照主键排序存储的，查询的时候，通过主键可以快速筛选数据，合理的主键设计，能够大大减少读取的数据量，提升查询性能。例如所有的分析，都需要指定业务的id，则可以将业务id字段作为主键的第一个字段顺序。
- 根据业务场景合理设计稀疏索引粒度

  ClickHouse的主键索引采用的是稀疏索引存储，稀疏索引的默认采样粒度是8192行，即每8192行取一条记录在索引文件中，实践建议：
  - 索引粒度越小，对于小范围的查询更有效，避免查询资源的浪费；
  - 索引粒度越大，则索引文件越小，索引文件的处理会更快；
  - 超过10亿的表索引粒度可设为16384，其他设为8192或者更小值。

二级跳数索引设计

跳数索引使用参考：

- 使用说明

  对于*MergeTree引擎，支持配置跳数索引，即一种数据局部聚合的粗糙索引，对数据块创建索引，选择性的保留一部分原始数据（minmax、set）， 或者是保留

计算后的中间数据（bloomfilter）。在查询时，选择忽略加载不会包含结果的数据块，从而达到加速查询的效果。

- 索引定义

  INDEX *index_name expr* **TYPE** *type（...）* **GRANULARITY** *granularity_value*

  - Expr：属性表达式，基于字段或者字段的表达式来创建索引；
  - type（...）：支持的索引类型，minmax、set等；
  - Granularity：创建索引的记录粒度。比如index_granularity = 8192，granularity配置为3，则使用8192*3条记录创建一条索引数据。

- 创建索引样例

```
CREATE TABLE skip_index_test ON CLUSTER default_cluster
(
ID     String,
URL    String,
Code   String,
EventTime Date,
INDEX a ID TYPE minmax GRANULARITY 5,
INDEX b (length(ID) * 8) TYPE set(100) GRANULARITY 5,
INDEX c (ID, Code) TYPE ngrambf_v1(3, 256, 2, 0) GRANULARITY 5,
INDEX d ID TYPE tokenbf_v1(256, 2, 0) GRANULARITY 5,
INDEX e ID TYPE bloom_filter(0.025) GRANULARITY 5
) ENGINE = MergeTree()
ORDER BY ID ;
```

  - **minmax索引**

    记录了一段数据范围内的最小和最大极值，其索引的作用类似分区目录的minmax索引，能够快速跳过无用的数据区间。

    ```
    INDEX a ID TYPE minmax GRANULARITY 5
    ```

    上述示例中minmax索引会记录这段数据区间内ID字段的极值。极值的计算涉及每5个index_granularity区间中的数据。

  - **set索引**

    直接记录了声明字段或表达式的取值（唯一值，无重复），其完整形式为set（max_rows），其中max_rows是一个阈值，表示在一个index_granularity内，索引最多记录的数据行数。如果max_rows=0，则表示无限制。

    ```
    INDEX b (length(ID) * 8) TYPE set(100) GRANULARITY 5
    ```

    上述示例中set索引会记录数据中ID的长度*8后的取值。其中，每个index_granularity内最多记录100条。

  - **布隆过滤器**

    - **bloom_filter索引**

      为指定的列存储布隆过滤器。

      可选的参数false_positive用来指定从布隆过滤器收到错误响应的几率。取值范围是 (0,1)，默认值：0.025。

      支持的数据类型：Int*，UInt*，Float*，Enum，Date，DateTime，String，FixedString，Array，LowCardinality，Nullable。

    - **ngrambf_v1索引**

      记录的是数据短语的布隆表过滤器，只支持String和FixedString数据类型。只能够提升in、notIn、like、equals和notEquals查询的性能，其完整形式为：

      ngrambf_v1(n, size_of_bloom_filter_in_bytes, number_of_hash_functions, random_seed)

      这些参数是一个布隆过滤器的标准输入，如果接触过布隆过滤器，应该会对此十分熟悉。

具体的含义如下：

- n：token长度，依据n的长度将数据切割为token短语。
- size_of_bloom_filter_in_bytes：布隆过滤器的大小。
- number_of_hash_functions：布隆过滤器中使用Hash函数的个数。
- random_seed：Hash函数的随机种子。

- **tokenbf_v1索引**

  是ngrambf_v1的变种，同样也是一种布隆过滤器索引。tokenbf_v1除了短语token的处理方法外，其他与ngrambf_v1是完全一样的。tokenbf_v1会自动按照非字符的、数字的字符串分割token。

  INDEX d ID TYPE tokenbf_v1(256,2,0) GRANULARITY 5

- 索引创建详见官方文档

  https://clickhouse.tech/docs/en/engines/table-engines/mergetree-family/mergetree/#table_engine-mergetree-data_skipping-indexes

- 建表后再创建索引

  ALTER TABLE table_name add INDEX min_max_index (etl_time) TYPE minmax GRANULARITY 3;

- 删除索引

  ALTER TABLE table_name DROP INDEX min_max_index;

- 单表跳数索引数量

  由于索引的创建对数据导入性能有影响，建议单表跳数索引的总数量控制在5个以内。

5.3.4 物化视图设计

5.3.4.1 物化视图概述

由于TTL规则不会从原始表中同步到物化视图表，因此源表中带有TTL规则时，物化视图表同样需要配置TTL规则，并且建议与源表保持一致。

表 5-2 普通物化视图与 projection 对比

物化视图类型	原表数据与物化视图一致性	灵活性	物化视图开发及维护复杂度
普通物化视图	数据从原表同步到物化视图需要时间窗。	• 灵活性较高，有新的业务可开发新的物化视图。 • 可开发复杂逻辑SQL语句的物化视图。	复杂度较高，需要开发很多物化视图，每个物化视图都需要单独去管理和维护。
projection	数据实时同步，数据写入即可查询到物化视图最新数据。	创建表时指定的物化视图语法，新的SQL业务需要修改表结构。	不需要开发很多物化视图，任意查询SQL会自动重写命中物化视图。

说明

Projection仅在MRS 3.2.0及以上的版本集群中支持。

5.3.4.2 普通物化视图设计

建议

- 在查询方式固定的场景，建议使用物化视图加速。

  物化视图创建参考如下：

  a. 明细表创建

  ```
  CREATE TABLE counter ON CLUSTER default_cluster
  (
  when DateTime DEFAULT now(),
  device UInt32,
  value Float32
  ) ENGINE=MergeTree
  PARTITION BY toYYYYMM(when)
  ORDER BY (device, when);
  ```

  b. 聚合表创建

  ```
  CREATE TABLE counter_daily_agg ON CLUSTER default_cluster
  (
  day DateTime,
  device UInt32,
  count UInt64,
  max_value_state AggregateFunction(max, Float32),
  min_value_state AggregateFunction(min, Float32),
  avg_value_state AggregateFunction(avg, Float32)
  )
  ENGINE = SummingMergeTree()
  PARTITION BY tuple()
  ORDER BY (device, day);
  ```

  说明

  AggregateFunction类型的字段使用二进制存储，在写入数据时，需要调用*State函数；而在查询数据时，则需要调用相应的*Merge函数。其中，*表示定义时使用的聚合函数。

  c. 物化视图创建

  ```
  CREATE MATERIALIZED VIEW counter_daily_mv ON CLUSTER default_cluster
  TO counter_daily_agg
  AS
  SELECT
  toStartOfDay(when) as day,
  device,
  count(*) as count,
  maxState(value) AS max_value_state,
  minState(value) AS min_value_state,
  avgState(value) AS avg_value_state
  FROM counter
  WHERE when >= toDate('2019-01-01 00:00:00')
  GROUP BY device, day
  ORDER BY device, day;
  ```

  说明

  创建物化视图counter_daily_mv，数据存储到表counter_daily_agg中，数据源来自counter。

- 聚合表在明细表名后加上_{type}_agg后缀；物化视图添加 _{type}_mv后缀。
- 物化视图、聚合表保持与明细表同样的分区类型及ttl时间。

- 物化视图中的group by字段名称与明细表对应字段名称一致；select子句返回列名称与聚合表中列的名称保持一致。
- 物化视图创建时不会进行语法校验，只有发生实际数据插入与查询时才会出错。
- 物化视图上线前，需做好充分验证。

规则

- 物化视图（Materialized View）显式指定聚合表。

  在创建物化视图时，使用TO关键字为物化视图指定数据存储表。

  如果不显示指定聚合表，则会创建隐式表.inner.mv1，与物化视图绑定。
- 用于数据预聚合的物化视图，聚合表使用聚合引擎。

  如果不用聚合引擎，则每次数据插入，会对明细表的全量数据重新计算，而不是只处理增量数据。
- 聚合表中，聚合指标定义成聚合类型（AggregateFunction）。

  物化视图的指标列与聚合表中对应字段名称一致，命名规范如下：

  {aggrateFunction}_{columnName}_state

  聚合表创建样例：

```
CREATE TABLE counter_daily_agg ON CLUSTER default_cluster
(
day DateTime,
device UInt32,
count UInt64,
max_value_state AggregateFunction(max, Float32),
min_value_state AggregateFunction(min, Float32),
avg_value_state AggregateFunction(avg, Float32)
)
ENGINE = SummingMergeTree()
PARTITION BY tuple()
ORDER BY (device, day);
```

- 在创建物化视图时，如果用到了多表联查，只有左表发生数据插入时才会触发物化视图数据修改。
- 禁止在创建物化视图时使用POPULATE关键字。

  使用POPULATE方式创建物化视图期间，如果有数据插入，则可能丢失。
- 推荐的历史数据同步方式：

```
-- create MV c where date >= in_the_future
CREATE MATERIALIZED VIEW mv1 ON CLUSTER default_cluster
TO dest
AS
SELECT  a,  d,  count()  AS cnt
FROM source
WHERE d >= '2020-11-01'
GROUP BY a, d;
-- arrives 2020-11-01
INSERT INTO dest -- insert all for before in_the_future
SELECT  a,  d,  count()  AS cnt
FROM source
WHERE d < '2020-11-01' -- piece by piece by 1 month (or .. day)
GROUP BY a, d;
```

- 修改明细表、聚合表结构，严格按照以下步骤实施：

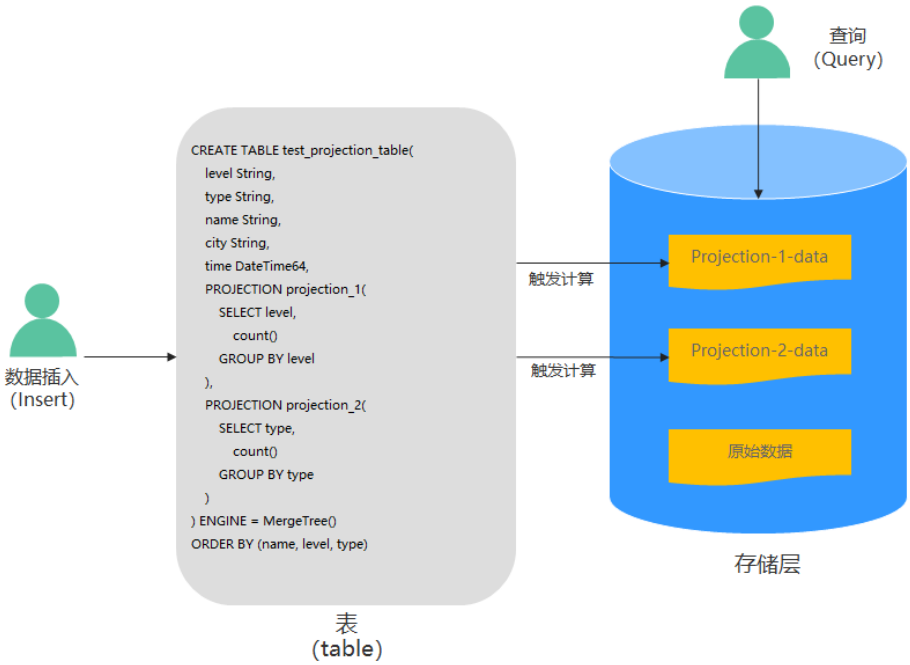
  a. 停止明细表数据插入。

  b. 修改聚合表结构设计。

  c. 删除物化视图表。

d. 重新创建新转化关系的物化视图。

5.3.4.3 Projection 设计



说明

Projection仅在MRS 3.2.0及以上的版本集群中支持。

projection 定义

```
CREATE TABLE test_projection_table(
    level String,
    type String,
    name String,
    city String,
    time DateTime64,
    PROJECTION projection_1(
        SELECT level,
            count()
        GROUP BY level
    ),
    PROJECTION projection_2(
        SELECT type,
            count()
        GROUP BY type
    )
) ENGINE = MergeTree()
ORDER BY (name, level, type)
```

通过表属性修改方式创建 projection

在创建好projection后还可以对projection进行修改，具体语句如下：

```
ALTER TABLE test_projection_table
ADD PROJECTION projection_3(
        SELECT type,
            level
        GROUP BY type,
```



```
        level
)
```

Projection 的使用

- 如下SQL查询的时候会走表达式：
  SELECT type, count() FROM test_projection_table WHERE type = 'A' GROUP BY type;
- 而如下SQL不会走projection，因为city不在projection的定义中。
  SELECT city, count() FROM test_projection_table WHERE type = 'A' GROUP BY city;
- 具体可以通过explain查看执行计划，如果出现ReadFromStorage (MergeTree(with projection))，表示命中projection。

命中 projection 使用规则

- Where条件必须是Projection定义中Group By的子集。
- Group By必须是Projection定义中Group By的子集。
- Select必须是Projection定义中Select的子集。
- 多表join场景不支持Projection特性，此种场景建议用普通物化视图实现。

5.3.5 逻辑视图设计

建议如下：

- 业务逻辑上有很多比较复杂的SQL运算，可以封装为一个视图，后续查询时只查询视图，简化业务查询使用。
- 如果业务间有权限隔离诉求，可将部分数据查询封装到视图中，使用视图方只能看到视图下有限行及列的数据。

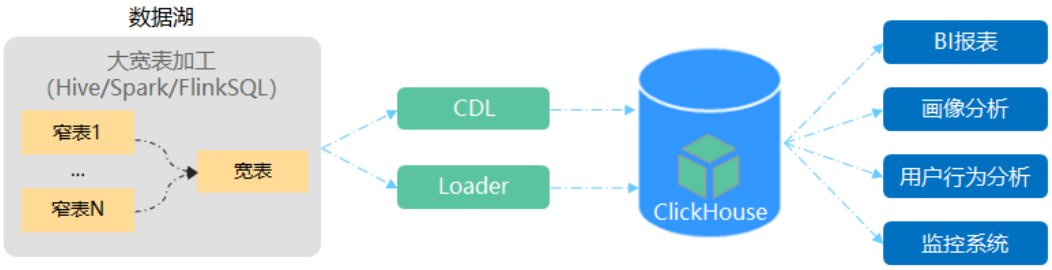
5.4 数据库开发

5.4.1 数据入库

5.4.1.1 数据入库工具

最佳实践方案

ClickHouse数据加工流程最佳实践：在数据湖中通过Hive&Spark（批量）/FlinkSQL（增量）加工成大宽表后，通过CDL/Loader工具实时同步到ClickHouse，下游BI工具和应用进行实时OLAP分析。



数据加工

建议使用Hive/Spark进行数据批量加工，FilkSQL进行数据增量加工。

数据入库

建议使用CDL（增量实时同步）和Loader（批量同步）工具进行数据同步，也可选择HDFS外表（CK集群只支持X86平台）用户自己写调度程序进行数据导入。

5.4.1.2 数据入库规范

规则

- 写本地表，查询分布式表，提升写入和查询性能，保证写入和查询的数据一致性。
- 只有在去重诉求的场景下，可以使用分布式表插入，通过sharding key将要去重的数据转发到同一个shard，便于后续去重查询。
- 外部模块保证数据导入的幂等性。

  ClickHouse不支持数据写入的事务保证。通过外部导入数据模块控制数据的幂等性，比如某个批次的数据导入异常，则drop对应的分区数据或清理掉导入的数据后，重新导入该分区或批次数据。
- 大批量少频次的写入。

  ClickHouse的每次数据插入，都会生成一到多个part文件，如果data part过多，merge压力会变大，甚至出现各种异常影响数据插入。建议每个批次5k到100k行，写入字段不能太多，太多字段情况下要减少写入行数，以降低对写入节点的内存和CPU压力，每秒不超过1次插入。
- 多副本并行导入。

  有大数据的导入场景，建议将数据提前拆分成多份，在一个shard内的多个副本同时导入，以分摊一个节点导入数据的压力，同时能提升数据入库的性能，缩短入库时间。

  常见错误：

  Too many parts(304). Merges are processing significantly slower than inserts

  原因分析：MergeTree的merge的速度跟不上目录生成的速度，数据目录越来越多就会抛出这个异常。

建议

- 一次只插入一个分区内的数据

  如果数据属于不同的分区，则每次插入，不同分区的数据会独立生成part文件，导致part总数量膨胀，建议一批插入的数据属于同一个分区。
- 写入速率

  单节点写入速度为50~200MB/S，如果写入的数据每行为1Kb，那么写入的速度为50，000到200，000行每秒，如果行数据容量更小，那么写入速度将更高，如果写入性能不够，可以使用多个副本同时写入，同一时间每个副本写入的数据保持均衡。
- 慎用分布式表批量插入
  - 写分布式表，数据会分发到集群的所有本地表，每个本地表插入的数据量是总插入量的1/N，batch size可能比较小，导致data part过多，merge压力变大，甚至出现异常影响数据插入；

  - 数据的一致性问题：数据先在分布式表写入节点的主机落盘，然后数据被异步地发送到本地表所在主机进行存储，中间没有一致性的校验，如果分布式表写入数据的主机出现宕机，会存在数据丢失风险；
  - 对于数据写分布式表和数据写本地表相比，分布式表数据写入性能也会变慢，单批次分布式表写，写入节点的磁盘和网络IO会成为性能瓶颈点。
  - 分布式表转发给各个shard成功与否，插入数据的客户端无法感知，转发失败的数据会不断重试转发，消耗CPU。

- 大批量数据导入要分时、分节点、扩容

  如果数据盘为SATA盘，当大批量数据集中插入时候，会抢占磁盘，使得磁盘长时间处于繁忙状态，影响其他alter类操作的效率。

  尽量避免批量导数据的SQL并发执行，会给磁盘和ClickHouse并发能力带来冲击。

- Kafka数据入库

  不建议建ClickHouse kafka表引擎，进行数据同步到ClickHouse中，当前CK的kafka引擎有会导致kafka引擎数据入库产生性能等诸多问题，通过用户使用经验，需要应用侧自己写kafka的数据消费，攒批写入ClickHouse，提升ClickHouse的入库性能。

- 使用分区替换或增加的方式写入数据

  为避免目标表写入脏数据导致的删改，先将数据写入临时表，再从临时表写入目标表。

  操作步骤如下：

  a. 创建一张与目标表table_dest结构、分区键、排序键、主键、存储策略、引擎都一致的临时表table_source。

  b. 先把数据写到临时表，一次只写入一个分区的数据，检查临时表的数据准确无误。

  c. 使用以下SQL查看目标表的分区：

     SELECT partition AS `partition`,sum(rows) AS `count` FROM system.parts WHERE active AND database=='*数据库名*' AND table=='*表名*' GROUP BY partition ORDER BY partition ASC;

  d. 如果目标表存在该分区，将分区替换到目标表，语法如下：

     ALTER TABLE table_dest [ON CLUSTER cluster] REPLACE PARTITION partition_expr FROM table_source;

  e. 如果目标表不存在该分区，将分区增加到目标表，语法如下：

     ALTER TABLE table_dest [ON CLUSTER cluster] REPLACE PARTITION tuple() partition_expr FROM table_source;

5.4.2 数据查询

数据查询规则

- 禁止select *查询

  只查询需要的字段可以减少磁盘io和网络io，提升查询性能。

- 使用uniqCombined替代distinct

  uniqCombined对去重逻辑进行了优化，通过近似去重提升十倍查询性能，如果对查询允许有误差，可以使用uniqCombined替代，否则还继续使用distinct语法。

- 降低对表的修改频次

  默认场景下ClickHouse执行alter语句是异步执行，对同一张表频繁执行alter操作可能导致业务失败。

- 多表复杂join拆分为两表join或子查询

  多表复杂join场景，建议拆分为两两表join，且两表join为大小表join，小小表join，尽量避免大大表join。也可以将多表复杂join拆分为子查询模式。

  SELECT name FROM tab_a WHERE id IN (SELECT id FROM tab_b WHERE name = 'xx');

  ⚠ 注意

  这里说的大表为条件过滤后的总数据量，千万级以上的数据量可定义为大表。

- 关联查询必须大表join小表

  对于ClickHouse来说，原则上需要把多表join模型提前加工为宽表模型，但是在一些情况下，多个表，甚至是维度表变化比较频繁情况下，不太适合进行宽表加工处理，不得已必须使用Join模型以实时查询到最新数据。那么join，建议2表join，大表join小表，小表在后（大表join小表），并必须有关联条件。小表的数据量控制在百万~千万行级别，且需要在join前尽量把小表数据通过条件进行有效过滤。

- join/in/not in需要添加Global关键字

  在通常的join/in/not in时候，需要在前面添加Global关键字，避免查询放大问题。

数据查询建议

- 建议查询指定分区

  通过指定分区字段会减少底层数据库扫描的文件数量，提升查询性能，实际经验：700个分区的千列大表，需要查询一个分区中有7000万数据，其他699个分区中无数据，虽然只有一个分区有数据，其他分区无数据，但是查询指定分区为百毫秒级性能，没有指定分区查询性能为1~2秒左右，性能相差20倍。

- 慎用final查询

  在查询语句的最后跟上final，通常是对于ReplacingMergeTree引擎，数据不能完全去重情况下，有些开发人员习惯写final关键字进行实时合并去重操作（merge-on-read），保证查询数据无重复数据。可以通过argMax函数或其他方式规避此问题。

数据修改

- 建议慎用delete、update的mutation操作

  标准SQL的更新、删除操作是同步的，即客户端要等服务端反回执行结果（通常是int值）；而ClickHouse的update、delete是通过异步方式实现的，当执行update语句时，服务端立即返回执行成功还是失败结果，但是实际上此时数据还没有修改完成，而是在后台排队等着进行真正的修改，可能会出现操作覆盖的情况，也无法保证操作的原子性。

  a. 业务场景要求有update、delete等操作，建议使用ReplacingMergeTree、CollapsingMergeTree、VersionedCollapsingMergeTree引擎，使用方式参见：**https://clickhouse.tech/docs/zh/engines/table-engines/mergetree-family/collapsingmergetree/**。

- 建议少或不增删数据列

  业务提前规划列个数，如果将来有更多列要使用，可以规划预留多列，避免在生产系统跑业务过程中进行大量的alter table modify列操作，导致不可以预知的性能、数据一致性问题。

- 对于批量数据清理，建议根据分区来操作：

  ALTER TABLE table_name DROP PARTITION partition_name;

- 禁止修改索引列

  对索引列的修改会导致现有索引失效，触发重建索引，期间查询数据不准确。

  如果业务场景必须修改索引列，推荐用ReplacingMergeTree引擎建表，使用数据写入+去重引擎代替数据更新场景：**https://clickhouse.tech/docs/zh/engines/table-engines/mergetree-family/collapsingmergetree/**。

数据 merge

建议谨慎执行optimize操作，Optimize一般会对表做重写操作，建议在业务压力小时候进行操作，否则对IO/MEM/CPU资源有较大消耗，导致业务查询变慢或不可用。

5.4.3 数据库应用开发

在ClickHouse的使用过程中，由于使用不规范的方式访问和查询，导致业务失败的情况时有发生。此外，偶尔也会发生因为网络闪断等导致连接和查询失败的情况。

MRS提供了ClickHouse的样例代码工程，旨在提供连接重试机制和规范化用户连接和查询的方法，从而减少业务失败的风险，提升系统的稳定性和可靠性。

本样例代码工程包含了连接、查询和插入相关规则和建议，以及相关的代码示例，可以帮助客户更好地理解和实践这些方法。通过使用本代码样例，客户可以有效地降低业务失败的概率，提升用户体验和业务质量。

操作步骤

步骤1 先获取clickhouse-example样例代码工程。

代码获取地址：**https://github.com/huaweicloud/huaweicloud-mrs-example/blob/mrs-3.1.2/src/clickhouse-examples/**。

步骤2 在样例工程“conf”目录下有一个“clickhouse-example.proerties”配置文件，其中各项的配置的作用如下所示：

```
#连接节点或Balancer的ip列表，ip之间用逗号隔开
loadBalancerIPList=
#是否需要开启ssl,如果取值为true，则loadBalancerHttpsPort必填
sslUsed=true
#端口号
loadBalancerHttpPort=
loadBalancerHttpsPort=
#ClickHouse安全模式开关，安全模式集群时该参数固定为true。
CLICKHOUSE_SECURITY_ENABLED=true
#连接的用户名
user=
#连接的用户的密码
password=
#集群名称
clusterName=
#数据库名称
databaseName=
#表名称
tableName=
```

```
#一个批次写入的条数
batchRows=10000
#写入数据的总批次
batchNum=10
#ip:port。安全模式下https端口，普通模式下http端口
clickhouse_dataSource_ip_list=
#ip:tcp port
native_dataSource_ip_list=ip:port,ip:port,ip:port
```

步骤3 在Demo.java有三种连接JDBC的样例：节点的JDBC连接、banlancer的JDBC连接和tcp端口的banlancer的JDBC连接。

步骤4 Demo提供了createDatabase、createTable、insertData和queryData的样例。

----结束

规则

- 大批量少频次的插入。

  内容要求：ClickHouse的每次数据插入都会生成一到多个part文件，如果data part过多则会导致merge压力变大，甚至出现服务异常影响数据插入。建议一次插入10万行，每秒不超过1次插入。

- 一次只插入一个分区内的数据。

  内容要求：如果数据属于不同的分区，则每次插入，不同分区的数据会独立生成part文件,导致part总数量膨胀。甚至写入报错“Merges are processing significantly slower than inserts“。一批次写入的数据，对应的分区数太多。ClickHouse建表之后insert batch时，会对不同的分区创建一个目录。如果一个batch里面的数据对应了过多的分区，那么一次insert就会生成较多的分区目录，后台merge线程处理速度跟不上分区增加的速度，社区规格是每秒不超过一个数据目录。

  具体的操作：确认一个batch的数据对应了多少个分区，insert的时候，尽量保证一个batch包含的分区数是1。

- 慎用delete、update操作。

  内容要求：建议使用CollapsingMergeTree、VersionedCollapsingMergeTree引擎或根据分区批量清理。

- ClickHouse需要写本地表。

  内容要求：连接balancer写入报错Request Entity Too Large。这是由于Nginx对http请求体大小有限制，而一次写入的数据量超过了这个限制。

  规避：修改Nginx配置项client_max_body_size为一个较大的值。

  解决：写本地表，不要通过balancer写入数据。

建议

- 查询增加重试机制

  clickhouse-example.proerties的配置文件的loadBalancerIPList可以配置多个ip，在二次样例代码中已经实现从第一个ip开始连接查询，查询失败时，继续连接下一个ip进行查询。

- 每个应用配置的loadBalancerIPList顺序不要一致，以免对balancer ip产生访问热点

  例如应用一配置loadBalancerIPList=ip1，ip2，ip3，应用二配置loadBalancerIPList=ip3，ip1，ip2。

- 根据连接方式选择端口

  普通集群默认开启8123端口，安全集群默认开启8443端口。

  端口号查看方式：在集群的Manager界面选择“集群 > 服务 > ClickHouse > 配置”。

  - 用于通过HTTP连接到ClickHouse server的端口默认为8123。
  - 用于通过HTTPS连接到ClickHouse server的端口默认为8443。
  - 用户客户端通过TCP连接到ClickHouse server的端口默认为9000。
  - 用户客户端通过TCP ssl连接到ClickHouse server的端口默认为9440。
  - ClickHouseBalancer的HTTP端口默认为21425。
  - ClickHouseBalancer的HTTPS端口默认为21426。

5.5 数据库调优

5.5.1 调优思路

ClickHouse的总体性能调优思路为性能瓶颈点分析、关键参数调整以及SQL调优。在调优过程中，需要综合系统资源、吞吐量、集群负载等各种因素来分析，定位性能问题，设定调优目标，调优达到客户所需目标即可。

ClickHouse调优人员需要系统软件架构、软硬件配置、数据库架构原理及配置参数、并发控制、查询处理和数据库应用有广泛而深刻的理解和认识，才能在调优过程中找到关键瓶颈点，解决性能问题。

图 5-3 调优流程

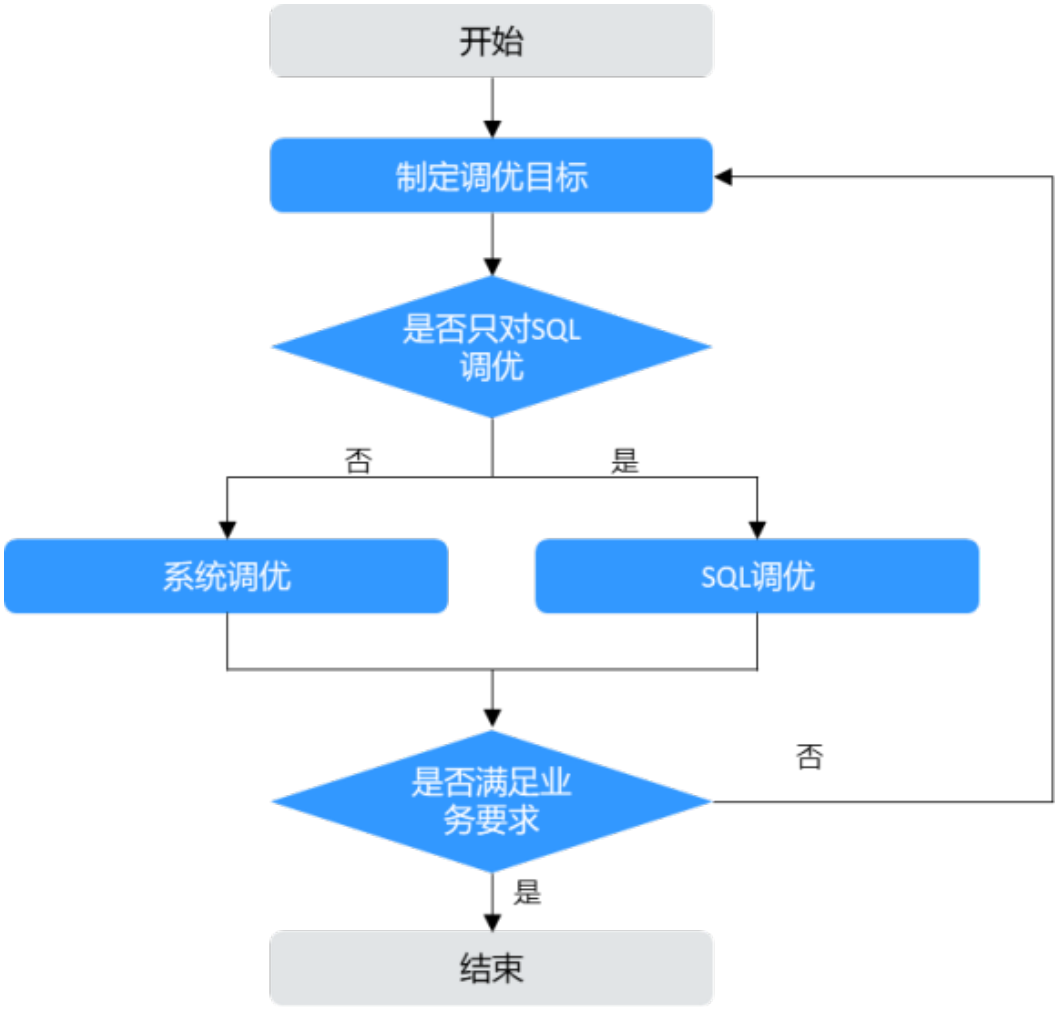


表 5-3 调优流程说明

流程	描述
系统调优	对OS操作系统级参数和数据库的调优，充分地利用主机的CPU、内存、I/O和网络资源，提升整个系统查询的吞吐量，同时数据库参数也调整到最优状态。
SQL调优	审视业务所用SQL语句是否存在可优化空间，包括： • 分析数据分布是否有倾斜，对于大表数据是否平均分布在各个shard。 • 分析建表语句，查看是否有建立分区、一级索引、二级索引、排序键是否指定等。 • 分析查询SQL是否使用了分区和索引，检查查询过滤条件比较频繁的列是否安排在建表时指定的索引及排序键的靠前位置。
数据库参数调优	通过调优数据参数，提升数据库性能，保障数据库稳定运行。

更多信息可参考ClickHouse社区文档相关调优内容**https://clickhouse.com/docs/en/intro**。

5.5.2 系统调优

通过FusionInsight Manager查看主机上的CPU、内存、I/O和网络资源使用情况，确认这些资源是否已被充分利用，分以下几种情况：

- 每个节点资源占用都比较均匀

  通过观察资源在每个节点都使用比较均匀，说明系统资源使用比较正常，可以先不关注，可以去分析SQL语句是否有进一步优化的余地。

- 有个别节点资源占用比较高

  如果观察到个别节点占用资源较高，需要针对占用资源较高的节点分析，分析当前的SQL语句是什么原因导致部分节点占用比其他节点更多资源，是计算还是数据存储倾斜导致，或者是软件bug导致。

- 每个节点资源占用都比较高

  如果集群所有节点资源占用都比较高，说明集群整体比较忙，需要单独确认需要调优的SQL语句，单独调优。如果SQL也无调优余地，集群资源达到瓶颈，需要通过扩容来提升查询性能，达到调优目标。

5.5.3 SQL 调优

规则

1. 合理使用数据表的分区字段和索引字段。

   MergeTree引擎，数据是以分区目录的形式进行组织存储的，在进行的数据查询时，使用分区可以有效跳过无用的数据文件，减少数据的读取。

   MergeTree引擎会根据索引字段进行数据排序，并且根据index_granularity的配置生成稀疏索引。根据索引字段查询，能快速过滤数据，减少数据的读取，大大提升查询性能。

2. 不要用**select ***，只查询需要的字段，减少机器负载，提升查询性能。

   OLAP分析场景，一张大宽表通常能有几百上千列，选择其中少数的几列做维度列、指标列计算。匹配这种场景下，ClickHouse的数据也是按照列存储的。如果使用**select ***，会大大加重系统的压力。

3. 通过limit限制查询返回的数据量，节省计算资源、减少网络开销。

   如果返回的数据量过大，客户端有可能出现内存溢出等服务异常。

   对于前端使用ClickHouse的场景，如果要查询的数据量比较大，建议每次可适当的进行分页查询返回数据，以减少查询数据量对网络带宽和计算资源的占用。

   【不做limit限制】

   ```
   SELECT dict_value
   FROM zeus.did_mapping
   ```

   ```
     Showed first 10000.

   10002340 rows in set. Elapsed: 1.124 sec. Processed 10.00 million rows, 190.10 MB (8.90 million rows/s., 169.10 MB/s.)
   ```

   耗时：1.124

   【做limit限制】

   ```
   SELECT dict_value
   FROM zeus.did_mapping
   LIMIT 10
   ```

   ```
   10 rows in set. Elapsed: 0.002 sec.
   ```

耗时：0.002

```
SELECT dict_value
FROM zeus.did_mapping
LIMIT 10

┌─dict_value─────────────────────────┐
│ 0012f9f3-3183-497b-839b-174adb45199f │
│ 002625ac-c1a1-47a3-9e6e-6f31f4b7a7c7 │
│ 007db765-8dc7-46ac-a7c2-067dbd5b9611 │
│ 009721be-3e9f-4137-84b5-0dbdd3f2cd52 │
│ 00f1fc5a-2194-4927-88f6-00288ca6fcf9 │
│ 010f8c3f-2049-450b-8edc-af70f5ac89b0 │
│ 0151d5a3-22b7-4bee-886b-d00129526001 │
│ 015b3c69-4fbb-4175-a313-eb56dfd4f38c │
│ 017d4a43-1957-4057-8e63-87de66d0fbb8 │
└────────────────────────────────────┘

10 rows in set. Elapsed: 0.002 sec.
```

4. join查询时小表在右。

   两表JOIN时，会将右表数据加载到内存中，再根据右表数据遍历左表做匹配，将小表放在右边，减少匹配查询的次数。根据使用的情况，大表join小表的性能比小表join大表的性能有数量级的提升。

   【大表在左小表在右】

   ```
   SELECT count(a.id)
   FROM
   (
   SELECT id
   FROM mytable
   WHERE id < 100000000
   ) AS a
   INNER JOIN
   (
   SELECT id
   FROM mytable
   WHERE id < 1000000
   ) AS b ON a.id = b.id;
   ```

   耗时：0.145 sec。

   【大表在右小表在左】

   ```
   SELECT count(a.id)
   FROM
   (
   SELECT id
   FROM mytable
   WHERE id < 1000000
   ) AS a
   INNER JOIN
   (
   SELECT id
   FROM mytable
   WHERE id < 100000000
   ) AS b ON a.id = b.id;
   ```

   耗时：0.996 sec。

5. ClickHouse不支持limit下推，SQL生成时需要优化，以免SQL性能受影响。

   【错误示例】

   ```
   select did from (select did from tableA) limit 10;
   ```

   【正确示例】

   ```
   select did from (select did from tableA limit 10);
   ```


6. 基于大宽表做数据分析，尽量不要使用大表join大表的操作。

   ClickHouse分布式join的性能较差，建议在模型侧将数据聚合成大宽表再导入ClickHouse。

   【两表join查询】

```
SELECT
col1,
col2
FROM
(
SELECT
t1.col1 AS col1,
t2.col2 AS col2
FROM
(
SELECT
did,
col1
FROM table1
WHERE cc_pt_d = '2020-03-30'
) AS t1
LEFT JOIN
(
SELECT
did AS did_v2,
col2
FROM table2
WHERE pt_d = '2020-03-30'
) AS t2 ON t2.did_v2 = t1.did
) AS t
GROUP BY
col1,
col2
LIMIT 10;
耗时: 40秒。
```

   【大宽表查询】

```
SELECT
col1,
col2
FROM
table1
GROUP BY
col1,
col2
LIMIT 10;
耗时: 8秒。
```

建议

1. 明确数据查询的范围，增加条件过滤和查询的数据周期过滤，缩小数据查询范围。

   【示例】

```
SELECT uniqCombined(did) from pp.scene_model where pt_d <‘2020-11-10’ and pt_d >
‘2020-11-03’;
```

2. 在分组、join等操前做数据过滤，减少计算的数据量。

   【效果对比】


```
SELECT
 Dest d, Name n, count(*) c, avg(ArrDelayMinutes)
  FROM ontime
   JOIN airports ON (airports.IATA = ontime.Dest)
    GROUP BY d, n HAVING c > 100000 ORDER BY d DESC
     LIMIT 10
```

3.878 seconds

Faster

```
SELECT dest, Name n, c AS flights, ad FROM (
 SELECT Dest dest, count(*) c, avg(ArrDelayMinutes) ad
  FROM ontime
   GROUP BY dest HAVING c > 100000
     ORDER BY ad DESC
) LEFT JOIN airports ON airports.IATA = dest LIMIT 10
```

1.177 seconds

3. 用PREWHERE替代WHERE，优先过滤数据，加速查询。

   PREWHERE相对于WHERE在执行时的区别：首先只读取PREWHERE表达式所指定的列，根据条件做数据过滤，再根据过滤后的数据读取其他列。这通常会减少磁盘读取数据的压力。

   PREWHERE只支持*MergeTree系列的表。系统配置**optimize_move_to_prewhere**默认开启，将WHERE转成PREWHERE，可以根据自己的业务场景调整这个配置。

   查询语句中同时有PREWHERE和WHERE,在这种情况下,PREWHERE先于WHERE执行。

4. 合理配置最大并发数。

   Clickhouse快是因为采用了并行处理机制，即使一个查询，默认也会用服务器一半的CPU去执行，所以ClickHouse对高并发查询的场景支持的不够。

   官方默认的最大并发数是100，可以根据实际场景调整并发配置，实际使用中并发数配置的是150,建议不超过200。

5. 部署负载均衡组件，查询基于负载均衡组件进行，避免单点查询压力太大影响性能。

   ClickHouse支持连接集群中的任意节点查询，如果查询集中到一台节点，可能会导致该节点的压力过大并且可靠性不高。建议使用ClickHouseBalancer或者其他负载均衡服务，均衡查询负载，提升可靠性。

6. 用近似去重（uniqCombined、uniq）替代精确去重。

   ClickHouse提供多种近似去重算法，通过count_distinct_implementation配置，支持将countDistinct语法转成所配置的近似算法。查询性能有数量级的提升。

   近似算法的误差一般在1%以内。在数据准确度要求不高，比如趋势分析等，建议使用近似去重提升用户体验。

   【使用精确去重查询】

```
SELECT countDistinct(dict_value)
FROM zeus.did_mapping

┌─uniqExact(dict_value)─┐
│              10002340 │
└───────────────────────┘

1 rows in set. Elapsed: 1.280 sec. Processed 10.00 million rows, 190.10 MB (7.81 million rows/s., 148.49 MB/s.)
```

   耗时：1.280秒。

   【使用近似查询】

```
SELECT uniq(dict_value)
FROM zeus.did_mapping

┌─uniq(dict_value)─┐
│         10046324 │
└──────────────────┘

1 rows in set. Elapsed: 0.061 sec. Processed 10.00 million rows, 190.10 MB (162.85 million rows/s., 3.10 GB/s.)
```

耗时：0.061秒。

7. 对于字符串类型的字段做复杂计算，建议先编码成整数类型，以提升计算性能。

   【 字符编码前，32字节的String类型字段did 】

```
CREATE TABLE default.Test_String ON Cluster default_cluster
(
  `EventDate` DateTime,
  `did`       String,
  `UserID`    UInt32,
  `ver`       UInt16
)
ENGINE = ReplicatedMergeTree('/clickhouse/tables/{shard}/default/Test_String', '{replica}')
PARTITION BY toYYYYMM(EventDate)
ORDER BY (EventDate, intHash32(UserID))
SETTINGS index_granularity = 8192;
select count(distinct did) from dws_wallet_xxx_mlb_ds;
执行耗时：142秒。
```

   【 字符编码后，将32位长String转码成int类型 】

```
CREATE TABLE default.Test_Int ON Cluster default_cluster
(
  `EventDate` DateTime,
  `did`       UInt32,
  `UserID`    UInt32,
  `ver`       UInt16
)
ENGINE = ReplicatedMergeTree('/clickhouse/tables/{shard}/default/Test_Int', '{replica}')
PARTITION BY toYYYYMM(EventDate)
ORDER BY (EventDate, intHash32(UserID))
SETTINGS index_granularity = 8192;
select count(distinct did_int) from dws_wallesst_xxx_mlb_ds;
执行耗时：34秒。
```

8. 高基数（大于10W）字段（int类型），使用bitmap做精确去重。

   【 countDistinct做精确去重 】

```
select count(distinct did_int) from dws_wallet_xxx_mlb_ds;
执行耗时：34秒。
```

   【 countDistinct做精确去重 】

```
select groupBitmapMergeState(arrayReduce('groupBitmapState', [toUInt64(did)])) as user1 from
t_r_309;
执行耗时：8秒。
```

9. 使用物化视图加速查询。

   对于查询方式比较固定的场景，建议使用物化视图，提前做好数据聚合，相对于查询明细表，性能有数量级的提升。

   【 物化视图创建 】

   明细表、物化视图创建参见 【 建议 】 物化视图创建参考。

   【 明细表插入数据 】

```
INSERT INTO counter SELECT
toDateTime('2019-01-01 00:00:00') + toInt64(number / 10) AS when,
(number % 10) + 1 AS device,
((device * 3) + (number / 10000)) + ((rand() % 53) * 0.1) AS value
FROM system.numbers
LIMIT 100000000;
```

   【 查询明细表 】


```
SELECT
    device,
    count(*) AS count,
    max(value) AS max,
    min(value) AS min,
    avg(value) AS avg
FROM counter
GROUP BY device
ORDER BY device ASC

┌─device─┬────count─┬──────max─┬─────min─┬───────────────avg─┐
│      1 │ 10000000 │ 10008.164 │   3.008 │ 5005.6002815747115 │
│      2 │ 10000000 │ 10011.185 │  6.0251 │  5008.600215326505 │
│      3 │ 10000000 │ 10014.112 │  9.0512 │  5011.599968641471 │
│      4 │ 10000000 │ 10017.152 │ 12.0163 │   5014.59998214844 │
│      5 │ 10000000 │ 10020.165 │ 15.1274 │  5017.600102860719 │
│      6 │ 10000000 │ 10023.101 │ 18.0385 │  5020.600059550402 │
│      7 │ 10000000 │  10026.19 │ 21.0416 │  5023.599433079225 │
│      8 │ 10000000 │ 10029.195 │ 24.0457 │  5026.600215895193 │
│      9 │ 10000000 │ 10032.197 │ 27.0668 │  5029.600555278349 │
│     10 │ 10000000 │ 10035.196 │ 30.0029 │  5032.600203040115 │
└────────┴──────────┴──────────┴─────────┴───────────────────┘

10 rows in set. Elapsed: 0.211 sec. Processed 100.00 million rows, 800.00 MB (474.79 million rows/s., 3.80 GB/s.)
```

【查询物化视图】

```
SELECT
    device,
    sum(count) AS count,
    maxMerge(max_value_state) AS max,
    minMerge(min_value_state) AS min,
    avgMerge(avg_value_state) AS avg
FROM counter_daily
GROUP BY device
ORDER BY device ASC

┌─device─┬────count─┬──────max─┬─────min─┬───────────────avg─┐
│      1 │ 10000000 │ 10008.164 │   3.008 │ 5005.6002815747115 │
│      2 │ 10000000 │ 10011.185 │  6.0251 │  5008.600215326504 │
│      3 │ 10000000 │ 10014.112 │  9.0512 │  5011.599968641471 │
│      4 │ 10000000 │ 10017.152 │ 12.0163 │   5014.59998214844 │
│      5 │ 10000000 │ 10020.165 │ 15.1274 │  5017.600102860719 │
│      6 │ 10000000 │ 10023.101 │ 18.0385 │  5020.600059550402 │
│      7 │ 10000000 │  10026.19 │ 21.0416 │  5023.599433079225 │
│      8 │ 10000000 │ 10029.195 │ 24.0457 │  5026.600215895193 │
│      9 │ 10000000 │ 10032.197 │ 27.0668 │  5029.600555278349 │
│     10 │ 10000000 │ 10035.196 │ 30.0029 │  5032.600203040115 │
└────────┴──────────┴──────────┴─────────┴───────────────────┘

10 rows in set. Elapsed: 0.002 sec. Processed 2.11 thousand rows, 194.74 KB (1.05 million rows/s., 96.96 MB/s.)
```

【效果对比】

使用物化视图后，遍历的数据量从1亿下降到2000，耗时从0.211秒下降到0.002秒，性能提升100倍。

10. 使用bitmap做跨表预估计算。

【场景】

用户画像，用户数预估：计算t_r_309和t_r_308 join后，did字段的基数。

【表join示例】

```
SELECT countDistinct(a.did)
FROM
(
SELECT DISTINCT did
FROM t_r_309
) AS a
INNER JOIN
(
SELECT DISTINCT did
FROM t_r_308
) AS b ON a.did = b.did;
```

【bitmap实现示例】

```
SELECT bitmapAndCardinality(user1, user2)
FROM
(
SELECT
1 AS join_id,
groupBitmapMergeState(arrayReduce('groupBitmapState', [toUInt32(did)])) AS user1
FROM t_r_309
) AS a
INNER JOIN
(
```

```
SELECT
1 AS join_id,
groupBitmapMergeState(arrayReduce('groupBitmapState', [toUInt32(did)])) AS user2
FROM t_r_308
) AS b ON a.join_id = b.join_id;
```

【效果对比】

多张表join后计算，随着join数越多，时延越大，基本在几十秒以上。使用bitmap计算预估，耗时在3秒以内。

11. 使用GLOBAL JOIN/IN替换普通的JOIN。

    ClickHouse基于分布式表的查询会转换成所有分片的本地表的操作，再汇总结果。实际使用中，join和global join的执行逻辑差别很大，建议使用global join做分布式表查询。

    【场景说明】

    a. 查询的集群有N个分片（shard）
    b. A_all是分布式表，对应的本地表是A_local
    c. B_all是分布式表，对应的本地表是B_local

    【分布式表直接join示例】

    ```
    SELECT * FROM A_all AS t1 JOIN B_all AS t2 ON t1.id = t2.id;
    ```

    执行逻辑如下：

    a. 在发起查询的节点，将查询分发到所有分片，转成A_all Join B_local。
    b. 在收到a中每个请求的分片，再将请求分发到所有分片，转成A_local Join B_local。
    c. 可以看到，分布式表的join操作，存在查询放大的问题。

    【分布式表global join示例】

    ```
    SELECT * FROM A_all AS t1 GLOBAL JOIN B_all AS t2 ON t1.id = t2.id;
    ```

    执行逻辑如下：

    a. 在查询发起的节点，查询B_all的所有数据到本地的缓存表T中，并将T分发到所有节点。
    b. 查询发起的节点，将本地缓存表T分发到所有分片。
    c. 每个分片执行A_local join T。
    d. 在收到a中每个请求的分片，再将请求分发到所有分片，转成A_local Join B_local。

    【效果对比】

    可以看到，使用GLOBAL关键字后，查询的放大减少了很多。不过，由于需要将右表汇总再分发到所有机器，如果右表的数据量很大，需要考虑机器的内存，避免内存溢出。

12. 数据压缩算法的选择，建议使用默认的lz4压缩算法。

    ClickHouse提供了两种数据压缩方式供选择：LZ4和ZSTD。

    默认的LZ4压缩方式，会提供更快的执行效率，但是同时，要付出较多的磁盘容量占用的代价。

13. ReplacingMergeTree表引擎数据查询，需要先做数据去重合并提升性能。

    如果使用去重引擎进行数据查询，且使用argMax函数和final关键字，会导致整个查询性能较差，需要提前对重复数据做合并去重optimize操作，查询时候直接查询不需要使用argMax函数和final关键字，提升查询性能。

5.5.4 参数调整最佳实践

参数名	参数描述	默认值	建议值	是否需要重启生效
max_memory_usage_for_all_queries	单台服务器上所有查询的内存使用量，默认没有限制。建议根据机器的总内存，预留一部分空间，防止内存不够导致服务或者机器宕机。	0	机器总内存的80%	否
max_memory_usage	单个查询在单台服务器的能使用的最大内存。	10G	50GB	否（新版本可通过多租户方式配置）
max_bytes_before_external_group_by	确定了在GROUP BY中启动将临时数据转存到磁盘上的内存阈值。默认值为0表示这项功能将被禁用。一般：设置为max_memory_usage/2。	0	25GB	否
max_execution_time	单次查询耗时的最长时间，单位为秒。默认没有限制。	0	300	否
max_threads	执行请求的最大线程数。默认情况下是按照机器CPU核数自动确定的。单并发情况下线程数越大越好（该值要小于CPU核数），多并发情况建议设置为CPU核数/2的值。	CPU核数/2	64	否
max_result_rows	限制返回结果行数，默认为0不限制。	0	100000	否
distributed_product_mode	默认SQL中的子查询不允许使用分布式表，修改为local表示将子查询中对分布式表的查询转换为对应的本地表。	deny	根据场景定： deny/local/global/allow	否
background_pool_size	后台用于merge的线程池大小。	16	64	否
log_queries	system.query_log表的开关。默认值为0，不存在该表。修改为1，系统会自动创建system.query_log表，并记录每次query的日志信息。	0	1	否

参数名	参数描述	默认值	建议值	是否需要重启生效
skip_unavailable_shards	当通过分布式表查询时，遇到无效的shard是否跳过。默认值为0表示不跳过，抛异常。设置值为1表示跳过无效shard。	0	建议使用默认值。异常时，调整为1，提供有损服务。	否
max_bytes_before_external_sort	如果没有足够的内存，可以使用该参数来设置外部排序（在磁盘中创建一些临时文件）。默认为0表示禁用外部排序功能，当内存不够时直接抛错，设置了该值order by可以正常完成，但是速度非常慢。	0	25GB	否
keep_alive_timeout	服务端与客户端保持长连接的时长，单位为秒。	10	600	否
max_concurrent_queries	最大支持的查询并发。	100	150	否
session_timeout_ms	Clickhouse服务和ZooKeeper保持的会话时长，超过该时间ZooKeeper还收不到Clickhouse的心跳信息，会将与Clickhouse的session断开。	3000	120000	否

5.6 数据库运维

5.6.1 日志运维管理

1. 日志级别、日志文件大小、日志文件数目的修改设置。
   - ClickHouse支持日志级别的动态调整。

     登录FusionInsight Manager界面，访问“集群 > 服务 > ClickHouse > 配置 > 全部配置 > ClickHouseServer > 日志 > logger.level”，可进行日志级别动态调整。日志级别优先级从低到高分别是trace、debug、information、warning、error、fatal，程序会打印高于或等于所设置级别的日志，设置的日志等级越低，打印出来的日志就越详细。
   - ClickHouse支持日志文件大小和文件数目的调整。

     登录FusionInsight Manager界面，访问“集群 > 服务 > ClickHouse > 配置 > 全部配置 > ClickHouseServer > 日志”，可修改ClickHouseServer审计日志和运行日志的文件大小和文件数目。
   - ClickHouse支持ClickHouseBalancer日志文件大小和文件数目的调整。

     登录FusionInsight Manager界面，访问“集群 > 服务 > ClickHouse > 配置 > 全部配置 > ClickHouseBalancer > 日志”，可修改ClickHouseBalancer日志文件的大小和文件数目。

2. 支持日志在线检索和日志收集。
   - 支持在线检索ClickHouse日志内容。

     登录FusionInsight Manager界面，访问“运维 > 日志 > 在线检索”，在“服务”中选择“ClickHouse”，“检索内容”填写日志检索关键字，通过“检索”在线检索ClickHouse日志内容。
   - 支持ClickHouse日志内容收集。

     登录FusionInsight Manager界面，访问“运维 > 日志 > 下载”，在“服务”中选择“ClickHouse”，“主机”中选择主机节点或默认所有主机节点，通过“下载”收集ClickHouse对应的日志文件。

5.6.2 日志管理规则

日志路径

- ClickHouse相关日志的默认存储路径为：“${BIGDATA_LOG_HOME}/clickhouse”。
- ClickHouseServer运行相关日志：“/var/log/Bigdata/clickhouse/clickhouseServer/ *.log”。
- ClickHouseBalancer运行日志：“/var/log/Bigdata/clickhouse/balance/*.log”。
- ClickHouseServer审计日志：“/var/log/Bigdata/audit/clickhouse/clickhouse-server-audit.log”。
- ClickHouse数据迁移日志：“/var/log/Bigdata/clickhouse/migration/${task_name}/clickhouse-copier_{timestamp}_{processId}/copier.log”。

日志归档规则

- ClickHouse日志启动了自动压缩归档功能，缺省情况下，当日志大小超过100MB的时（此日志文件大小可进行配置），会自动压缩。
- 压缩后的日志文件名规则为：“*<原有日志名>*.[*编号*].gz”。
- 默认最多保留最近的10个压缩文件，压缩文件保留个数可以在Manager界面中配置。

5.6.3 日志详细信息

日志类型	日志文件名	描述
ClickHouse相关日志	/var/log/Bigdata/clickhouse/clickhouseServer/clickhouse-server.err.log	ClickHouseServer服务运行错误日志文件路径。
	/var/log/Bigdata/clickhouse/clickhouseServer/checkService.log	ClickHouseServer服务运行关键日志文件路径。
	/var/log/Bigdata/clickhouse/clickhouseServer/clickhouse-server.log	
	/var/log/Bigdata/clickhouse/clickhouseServer/ugsync.log	用户角色同步工具打印日志。
	/var/log/Bigdata/clickhouse/clickhouseServer/prestart.log	ClickHouse预启动日志。

日志类型	日志文件名	描述
	/var/log/Bigdata/clickhouse/clickhouseServer/start.log	ClickHouse启动日志。
	/var/log/Bigdata/clickhouse/clickhouseServer/checkServiceHealthCheck.log	ClickHouse健康检查日志。
	/var/log/Bigdata/clickhouse/clickhouseServer/checkugsync.log	用户角色同步检查日志。
	/var/log/Bigdata/clickhouse/clickhouseServer/checkDisk.log	ClickHouse磁盘检测日志文件路径。
	/var/log/Bigdata/clickhouse/clickhouseServer/backup.log	ClickHouse在Manager上执行备份恢复操作的日志文件路径。
	/var/log/Bigdata/clickhouse/clickhouseServer/stop.log	ClickHouse停止日志。
	/var/log/Bigdata/clickhouse/clickhouseServer/postinstall.log	ClickHouse的postinstall.sh脚本调用日志。
	/var/log/Bigdata/clickhouse/balance/start.log	ClickHouseBalancer服务启动日志文件路径。
	/var/log/Bigdata/clickhouse/balance/error.log	ClickHouseBalancer服务运行错误日志文件路径。
	/var/log/Bigdata/clickhouse/balance/access_http.log	ClickHouseBalancer服务运行http日志文件路径。
	/var/log/Bigdata/clickhouse/balance/access_tcp.log	ClickHouseBalancer服务运行tcp日志文件路径。
	/var/log/Bigdata/clickhouse/balance/checkService.log	ClickHouseBalancer服务检查日志。
	/var/log/Bigdata/clickhouse/balance/postinstall.log	ClickHouseBalacer的postinstall.sh脚本调用日志。
	/var/log/Bigdata/clickhouse/balance/prestart.log	ClickHouseBalancer服务预启动日志文件路径。
	/var/log/Bigdata/clickhouse/balance/stop.log	ClickHouseBalancer服务关闭日志文件路径。
	/var/log/Bigdata/clickhouse/clickhouseServer/auth.log	ClickHouse服务认证日志。
	/var/log/Bigdata/clickhouse/clickhouseServer/cleanService.log	重装实例异常产生的记录日志。

日志类型	日志文件名	描述
	/var/log/Bigdata/clickhouse/clickhouseServer/offline_shard_table_manager.log	ClickHouse入服/退服日志。
	/var/log/Bigdata/clickhouse/clickhouseServer/traffic_control.log	ClickHouse主备容灾流量控制日志。
	/var/log/Bigdata/clickhouse/clickhouseServer/clickhouse_migrate_metadata.log	ClickHouse元数据搬迁日志。
	/var/log/Bigdata/clickhouse/clickhouseServer/clickhouse_migrate_data.log	ClickHouse业务数据搬迁日志。
	/var/log/Bigdata/clickhouse/clickhouseServer/changePassword.log	ClickHouse修改用户密码日志。
数据迁移日志	/var/log/Bigdata/clickhouse/migration/*数据迁移任务名*/clickhouse-copier_*{timestamp}_{processId}*/copier.log	参考使用ClickHouse数据迁移工具，使用迁移工具时产生的运行日志。
	/var/log/Bigdata/clickhouse/migration/*数据迁移任务名*/clickhouse-copier_*{timestamp}_{processId}*/copier.err.log	参考使用ClickHouse数据迁移工具，使用迁移工具时产生的错误日志。
	/var/log/Bigdata/tomcat/clickhouse/auto_balance/*数据迁移任务名*/balance_manager.log	参考使用ClickHouse数据迁移工具，勾选一键均衡产生的运行日志。
clickhouse-tomcat日志	/var/log/Bigdata/tomcat/clickhouse/web_clickhouse.log	ClickHouse自定义UI运行日志。
	/var/log/Bigdata/tomcat/audit/clickhouse/clickhouse_web_audit.log	clickhouse的数据迁移审计日志。
ClickHouse审计日志	/var/log/Bigdata/audit/clickhouse/clickhouse-server-audit.log	ClickHouse的审计日志文件路径。