

数据湖探索

常见问题

文档版本 01

发布日期 2024-01-30



版权所有 © 华为技术有限公司 2024。保留一切权利。

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

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为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。

安全声明

漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：

<https://www.huawei.com/cn/psirt/vul-response-process>

如企业客户须获取漏洞信息，请参见如下网址：

<https://securitybulletin.huawei.com/enterprise/cn/security-advisory>

目 录

1 Flink 作业相关问题.....	1
1.1 使用咨询.....	1
1.1.1 DLI Flink 作业支持哪些数据格式和数据源.....	1
1.1.2 如何给子用户授权查看 Flink 作业?	2
1.1.3 Flink 作业设置“异常自动重启”	3
1.1.4 Flink 作业如何保存作业日志.....	4
1.1.5 如何查看 Flink 作业输出结果?	5
1.1.6 Flink 作业管理界面对用户进行授权时报用户不存在怎么办.....	5
1.1.7 手动停止了 Flink 作业，再次启动时没有提示从哪个 Checkpoint 恢复.....	5
1.1.8 DLI Flink 现在支持到哪个版本了？是否支持 Flink 1.13？下一个版本是哪个版本？	6
1.1.9 DLI 使用 SMN 主题，提示 SMN 主题不存在，怎么处理?	6
1.2 Flink SQL 作业相关问题.....	6
1.2.1 Flink SQL 作业的消费能力如何，即一天可以处理多大的数据量?	6
1.2.2 Flink SQL 中的 temp 流中数据是否需要定期清理，如何清理?	6
1.2.3 创建 Flink SQL 作业时选择 OBS 桶，提示未授权.....	6
1.2.4 Flink SQL 作业将 OBS 表映射为 DLI 的分区表.....	7
1.2.5 Flink SQL 作业 Kafka 分区数增加或减少，不用停止 Flink 作业，实现动态感知.....	7
1.2.6 OBS 表如何映射为 DLI 的分区表?	7
1.2.7 在 Flink SQL 作业中创建表使用 EL 表达式，作业运行报 DLI.0005 错误.....	8
1.2.8 Flink 作业输出流写入数据到 OBS，通过该 OBS 文件路径创建的 DLI 表查询无数据.....	8
1.2.9 Flink SQL 作业运行失败，日志中有 connect to DIS failed java.lang.IllegalArgumentException: Access key cannot be null 错误.....	11
1.2.10 Flink SQL 作业读取 DIS 数据报 Not authorized 错误.....	11
1.2.11 Flink SQL 作业消费 Kafka 后 sink 到 es 集群，作业执行成功，但未写入数据.....	12
1.2.12 Flink Opensource SQL 如何解析复杂嵌套 JSON?	12
1.2.13 Flink Opensource SQL 从 RDS 数据库读取的时间和 RDS 数据库存储的时间为什么会不一致?	14
1.2.14 Flink SQL 和 Flink Opensource SQL 的语法有什么区别?	15
1.2.15 Flink Opensource SQL Elasticsearch 结果表 failure-handler 参数填写 retry_rejected 导致提交失败...	15
1.3 Flink Jar 作业相关问题.....	15
1.3.1 Flink Jar 作业配置 checkpoint 保存到 OBS.....	15
1.3.2 Flink Jar 作业是否支持上传配置文件，要如何操作?	18
1.3.3 Flink Jar 包冲突，导致提交失败.....	19
1.3.4 Flink Jar 作业访问 DWS 启动异常，提示客户端连接数太多错误.....	20

1.3.5 Flink Jar 作业运行报错，报错信息为 Authentication failed.....	20
1.3.6 Flink Jar 作业设置 backend 为 OBS，报错不支持 OBS 文件系统.....	21
1.3.7 Hadoop jar 包冲突，导致 Flink 提交失败.....	22
1.3.8 Flink jar 如何连接 SASL_SSL?	22
1.4 性能调优.....	22
1.4.1 Flink 作业推荐配置指导.....	22
1.4.2 Flink 作业如何进行性能调优.....	25
1.4.3 如何在一个 Flink 作业中将数据写入到不同的 Elasticsearch 集群中?	29
1.4.4 Flink 作业重启后，如何保证不丢失数据?	29
1.5 运维指导.....	31
1.5.1 Flink 作业提交错误，如何定位.....	31
1.5.2 Flink 作业运行异常，如何定位.....	31
1.5.3 Flink 作业重启后，如何判断是否可以从 checkpoint 恢复.....	32
1.5.4 作业语义检验时提示 DIS 通道不存在怎么处理?	33
1.5.5 如何处理作业的 OBS Bucket 没有授权?	33
1.5.6 DLI Flink 作业提交运行后（已选择保存作业日志到 OBS 桶），提交运行失败的情形（例如：jar 包冲突），有时日志不会写到 OBS 桶中.....	34
1.5.7 Kafka Sink 配置发送失败重试机制.....	34
1.5.8 访问 FlinkUI/SparkUI 页面无法正常显示怎么办?	35
1.5.9 Jobmanager 与 Taskmanager 心跳超时，导致 Flink 作业异常怎么办?	35
1.5.10 Flink jobmanager 日志一直报 Timeout expired while fetching topic metadata 怎么办?	36
2 SQL 作业相关问题.....	37
2.1 使用咨询.....	37
2.1.1 SQL 作业使用咨询.....	37
2.2 作业开发.....	38
2.2.1 如何合并小文件.....	38
2.2.2 DLI 如何访问 OBS 桶中的数据.....	38
2.2.3 创建 OBS 表时正确指定 OBS 路径.....	38
2.2.4 关联 OBS 桶中嵌套的 JSON 格式数据如何创建表.....	39
2.2.5 count 函数如何进行聚合.....	39
2.2.6 如何将一个区域中的 DLI 表数据同步到另一个区域中?	39
2.2.7 SQL 作业如何指定表的部分字段进行表数据的插入.....	39
2.2.8 怎样删除表数据?	39
2.3 作业运维报错.....	40
2.3.1 用户导表到 OBS 报 “path obs://xxx already exists” 错误.....	40
2.3.2 对两个表进行 join 操作时，提示：SQL_ANALYSIS_ERROR: Reference 't.id' is ambiguous, could be: t.id, t.id;.....	40
2.3.3 执行查询语句报错：The current account does not have permission to perform this operation, the current account was restricted. Restricted for no budget.....	40
2.3.4 执行查询语句报错：There should be at least one partition pruning predicate on partitioned table XX.YYY.....	40
2.3.5 LOAD 数据到 OBS 外表报错：IllegalArgumentException: Buffer size too small. size.....	41
2.3.6 SQL 作业运行报错：DLI.0002 FileNotFoundException.....	41

2.3.7 用户通过 CTAS 创建 hive 表报 schema 解析异常错误.....	42
2.3.8 在 DataArts Studio 上运行 DLI SQL 脚本，执行结果报 org.apache.hadoop.fs.obs.OBSIOException 错误.....	42
2.3.9 使用 CDM 迁移数据到 DLI，迁移作业日志上报 UQUERY_CONNECTOR_0001:Invoke DLI service api failed 错误.....	42
2.3.10 SQL 作业访问报错：File not Found.....	43
2.3.11 SQL 作业访问报错：DLI.0003: AccessControlException XXX.....	43
2.3.12 SQL 作业访问外表报错：DLI.0001: org.apache.hadoop.security.AccessControlException: verifyBucketExists on {{桶名}}: status [403].....	44
2.3.13 执行 SQL 语句报错：The current account does not have permission to perform this operation, the current account was restricted. Restricted for no budget.....	44
2.4 运维指导.....	44
2.4.1 SQL 作业运行慢如何定位.....	44
2.4.2 查看 DLI SQL 日志.....	49
2.4.3 查看 DLI 的执行 SQL 记录.....	50
2.4.4 配置 AE 参数解决数据倾斜.....	50
2.4.5 DLI 控制台中无法查询到对应表.....	51
2.4.6 OBS 表压缩率较高.....	51
2.4.7 如何避免字符码不一致导致的乱码.....	51
2.4.8 删除表后再重新创建同名的表，需要对操作该表的用户和项目重新赋权.....	51
2.4.9 DLI 分区内表导入的文件不包含分区列的数据，导致数据导入完成后查询表数据失败.....	52
2.4.10 创建 OBS 外表，因为 OBS 文件中的某字段存在回车换行符导致表字段数据错误.....	53
2.4.11 SQL 作业中存在 join 操作，因为自动广播导致内存不足，作业一直运行中.....	53
2.4.12 join 表时没有添加 on 条件，造成笛卡尔积查询，导致队列资源爆满，作业运行失败.....	54
2.4.13 手动在 OBS 表的分区目录下添加了数据，但是无法查询到该部分数据.....	54
2.4.14 为什么 insert overwrite 覆盖分区表数据的时候，覆盖了全量数据？	55
2.4.15 为什么 SQL 作业一直处于“提交中”？	55
2.4.16 跨源连接 RDS 表中 create_date 字段类型是 datetime，为什么 dli 中查出来的是时间戳呢？	55
2.4.17 SQL 作业执行完成后，修改表名导致 datasize 修改失败怎么办？	55
2.4.18 从 DLI 导入数据到 OBS，为什么数据量出现差异？	55
3 Spark 作业相关问题.....	57
3.1 使用咨询.....	57
3.1.1 Spark 作业使用咨询.....	57
3.2 作业开发.....	58
3.2.1 Spark 如何将数据写入到 DLI 表中.....	58
3.2.2 通用队列操作 OBS 表如何设置 AK/SK.....	58
3.2.3 如何查看 DLI Spark 作业的实际资源使用情况.....	59
3.2.4 将 Spark 作业结果存储在 MySQL 数据库中，缺少 pymysql 模块，如何使用 python 脚本访问 MySQL 数据库？	60
3.2.5 如何在 DLI 中运行复杂 PySpark 程序？	61
3.2.6 Spark 作业访问 MySQL 数据库的方案.....	62
3.2.7 如何通过 JDBC 设置 spark.sql.shuffle.partitions 参数提高并行度.....	63
3.2.8 Spark jar 如何读取上传文件.....	63

3.3 作业运维报错.....	64
3.3.1 运行 Spark 作业报 java.lang.AbstractMethodError.....	64
3.3.2 Spark 作业访问 OBS 数据时报 ResponseCode: 403 和ResponseStatus: Forbidden 错误.....	64
3.3.3 有访问 OBS 对应的桶的权限，但是 Spark 作业访问时报错 verifyBucketExists on XXXX: status [403]..	65
3.3.4 Spark 作业运行大批量数据时上报作业运行超时异常错误.....	65
3.3.5 使用 Spark 作业访问 sftp 中的文件，作业运行失败，日志显示访问目录异常.....	65
3.3.6 执行作业的用户数据库和表权限不足导致作业运行失败.....	65
3.3.7 为什么 Spark3.x 的作业日志中打印找不到 global_temp 数据库.....	66
3.3.8 在使用 Spark2.3.x 访问元数据时，DataSource 语法创建 avro 类型的 OBS 表创建失败.....	66
3.4 运维指导.....	66
3.4.1 添加 Python 包后，找不到指定的 Python 环境.....	66
3.4.2 为什么 Spark jar 作业一直处于“提交中”？	67
4 产品咨询相关问题.....	68
4.1 使用咨询.....	68
4.1.1 什么是 DLI.....	68
4.1.2 DLI 适用哪些场景.....	68
4.1.3 DLI 支持哪些数据格式.....	70
4.1.4 DLI Flink 与 MRS Flink 有什么区别？	70
4.1.5 DLI 中的 Spark 组件与 MRS 中的 Spark 组件有什么区别？	72
4.1.6 DLI 的数据可存储在哪些地方.....	72
4.1.7 DLI 表与 OBS 表的区别.....	72
4.1.8 不上传数据到 OBS，如何使用 DLI.....	72
4.1.9 DLI 是否支持导入其他租户共享 OBS 桶的数据？	72
4.1.10 区域和可用区.....	73
4.1.11 全局变量的使用中，一个子帐号是否可以使用其他子帐号创建的全局变量.....	74
4.2 作业管理.....	74
4.2.1 如何获取 DLI 作业样例（ Demo ）	74
4.2.2 使用 DLI 运行几万个作业，如何管理？	75
4.2.3 DLI 上已经创建好的表如何修改表字段名称？	75
4.3 隐私安全.....	75
4.3.1 DLI 是否存在 Apache Spark 命令注入漏洞（ CVE-2022-33891 ）？	75
5 计费相关问题.....	76
5.1 DLI 的计费模式是怎样的？	76
5.2 包周期队列可以转按需队列吗？	76
5.3 创建的按需队列是否支持转包周期队列？	77
5.4 数据湖队列什么情况下是空闲状态？	77
5.5 如何排查计费.....	77
5.6 购买 4000cus 规格资源，一次性购买 3 个月的和分 3 次购买，每次购买一个 1 月的，这两种付费方式的区别？	78
5.7 如何购买 30cus 规格资源？	78
5.8 在一个 16CU 的队列上运行 Flink 作业只用了 2 个 CU, 如何计费？	78
5.9 想要使用扫描套餐包计费，购买队列还要额外收费吗？	78

6 配额相关问题.....	80
6.1 什么是用户配额?	80
6.2 怎样查看我的配额.....	80
6.3 如何申请扩大配额.....	81
6.4 Failed to create the database. {"error_code":"DLI.1028","error_msg":"Already reached the maximum quota of databases:XXX". 提示配额不足, 如何处理?	81
7 授权相关问题.....	83
7.1 使用咨询.....	83
7.1.1 DLI 细粒度授权.....	83
7.1.2 如何理解 DLI 分区表的列赋权.....	85
7.2 运维指导.....	85
7.2.1 欠费导致权限不足.....	85
7.2.2 进行程序包更新操作时提示权限不足.....	86
7.2.3 执行 SQL 查询语句报错: DLI.0003: Permission denied for resource.....	86
7.2.4 已经给表授权, 但是再次报错无法查询怎么办?	86
7.2.5 表继承数据库权限后, 对表重复赋予已继承的权限会报错吗?	87
7.2.6 为什么已有 View 视图的 select 权限, 但是查询不了 View?	87
8 队列相关问题.....	88
8.1 使用咨询.....	88
8.1.1 DLI 队列管理是否支持修改描述内容.....	88
8.1.2 删除队列会导致数据库中的表数据丢失吗?	88
8.1.3 队列异常时, DLI 怎么保证 Spark 作业的可靠性?	88
8.1.4 DLI 如何进行队列异常监控?	88
8.2 运维指导.....	88
8.2.1 查看 DLI 队列负载.....	88
8.2.2 如何判断当前 DLI 队列中的作业是否有积压?	89
8.2.3 如何将老版本的 Spark 队列切换成通用型队列.....	90
8.2.4 为什么云监控 CES 服务看不到 DLI 队列的资源运行情况?	90
8.2.5 购买了 64CU 的队列资源, 运行 Spark 作业时如何分配?	90
8.2.6 创建扩容任务时, 提示 Queue plans create failed. The plan xxx target cu is out of quota 报错.....	91
8.2.7 在 default 队列执行 DLI SQL 失败, 上报超时异常.....	91
8.2.8 怎样查看弹性资源池和作业的资源使用情况?	91
9 跨源连接相关问题.....	94
9.1 跨源连接问题.....	94
9.1.1 DLI 增强型跨源连接为什么要创建对等连接.....	94
9.1.2 增强型跨源连接绑定队列失败.....	94
9.1.3 DLI 增强型跨源连接 DWS 失败.....	95
9.1.4 创建跨源成功但测试网络连通性失败怎么办?	96
9.1.5 怎样配置 DLI 队列与数据源的网络连通?	100
9.1.6 DLI 创建跨源连接, 进行绑定队列一直在创建中怎么办?	101
9.1.7 按需队列怎样绑定跨源链接?	101

9.1.8 如何打通 DLI 和数据源的网络?	101
9.2 跨源分析问题.....	101
9.2.1 如何实现跨数据源查询.....	101
9.2.2 如何实现跨区域访问数据.....	101
9.2.3 创建 DLI 关联 RDS 表时, 如果 RDS 表有自增主键或者其他自动填充的字段时, 在 DLI 中插入数据时, 对应字段如何填写?	102
9.3 跨源连接运维报错.....	102
9.3.1 新建跨源连接, 显示已激活, 但使用时报 communication link failure 错误.....	102
9.3.2 跨源访问 MRS HBase, 连接超时, 日志未打印错误.....	105
9.3.3 DLI 跨源连接报错找不到子网.....	105
9.3.4 跨源 RDS 表, 执行 insert overwrite 报 Incorrect string value 错误.....	105
9.3.5 创建 RDS 跨源表报空指针错误.....	106
9.3.6 对跨源 DWS 表执行 insert overwrite 操作, 报错: org.postgresql.util.PSQLException: ERROR: tuple concurrently updated.....	107
9.3.7 通过跨源表向 CloudTable Hbase 表导入数据, executor 报错: RegionTooBusyException.....	107
9.3.8 通过 DLI 跨源写 DWS 表, 非空字段出现空值异常.....	108
9.3.9 更新跨源目的端源表后, 未同时更新对应跨源表, 导致 insert 作业失败.....	109
10 使用 API 相关问题.....	110
10.1 如何获取 AK/SK?	110
10.2 如何获取项目 ID?	110
10.3 提交 SQL 作业时, 返回 “unsupported media Type” 信息.....	111
10.4 不同帐号的调用 API 时的项目 ID(Project ID)是固定的吗?	111
10.5 创建 SQL 作业的 API 执行超过时间限制, 运行超时报错.....	111
10.6 API 接口返回的中文字符为乱码, 如何解决?	111
11 使用 SDK 相关问题.....	112
11.1 使用 SDK 查询 SQL 作业结果, 如何设置超时时间?	112
11.2 服务器: 请求 dli.xxx,unable to resolve host address, 如何解决?	112
A 修订记录.....	113

1

Flink 作业相关问题

1.1 使用咨询

1.1.1 DLI Flink 作业支持哪些数据格式和数据源

DLI Flink作业支持如下数据格式：

Avro, Avro_merge, BLOB, CSV, EMAIL, JSON, ORC, Parquet, XML。

DLI Flink作业支持如下数据源：

CloudTable HBase, CloudTable OpenTSDB, CSS Elasticsearch, DCS, DDS, DIS, DMS, DWS, EdgeHub, MRS HBase, MRS Kafka, 开源Kafka, 文件系统, OBS, RDS, SMN

表 1-1 数据格式和支持的输入输出流

数据格式	支持的输入流	支持的输出流
Avro	-	OBS输出流
Avro_merge	-	OBS输出流
BLOB	<ul style="list-style-type: none">• DIS输入流• MRS Kafka输入流• 开源Kafka输入流	-
CSV	<ul style="list-style-type: none">• DIS输入流• OBS输入流• 开源Kafka输入流	<ul style="list-style-type: none">• DIS输出流• OBS输出流• DWS输出流（通过OBS方式转储）• 开源Kafka输出流• 文件系统输出流
EMAIL	DIS输入流	-

数据格式	支持的输入流	支持的输出流
JSON	<ul style="list-style-type: none">DIS输入流OBS输入流MRS Kafka输入流开源Kafka输入流	<ul style="list-style-type: none">DIS输出流OBS输出流MRS Kafka输出流开源Kafka输出流
ORC	-	<ul style="list-style-type: none">OBS输出流DWS输出流（通过OBS方式转储）
Parquet	-	<ul style="list-style-type: none">OBS输出流文件系统输出流
XML	DIS输入流	-

1.1.2 如何给子用户授权查看 Flink 作业？

子用户使用DLI时，可以查看队列，但是不能查看Flink作业，可以通过在DLI中对子用户授权，或在IAM中对子用户授权：

- DLI授权管理
 - 使用租户帐号，或者作业owner帐号，或有DLI Service Administrator权限的帐号，登录DLI控制台。
 - 在“作业管理”>“Flink作业”页面找到对应的作业。
 - 在对应作业的“操作”栏中选择“更多”>“权限管理”。

图 1-1 Flink 作业权限管理



- 在“授权”页面输入需要授权的用户名，勾选需要的权限。确认后，被授权用户就可以查看该作业，并且执行对应操作。

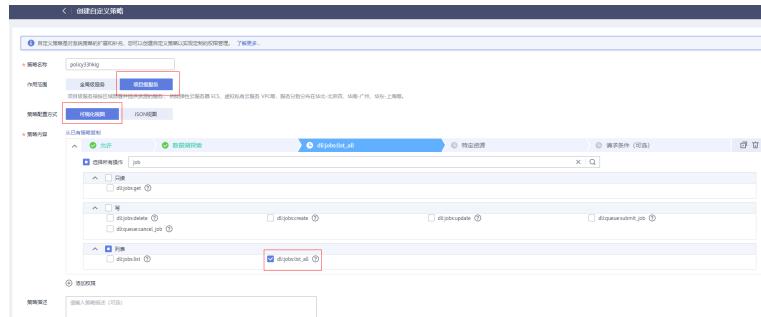
图 1-2 授权



- IAM授权管理
 - 登录统一身份认证IAM控制台，在“权限”页面，单击“创建自定义权限”。

- b. 为查看DLI Flink作业创建权限策略：
- 策略名称：可以使用默认名称，也可以自定义。
 - 作用范围：选择“项目级服务”。
 - 策略配置方式：选择“可视化视图”。
 - 策略内容：依次选择“允许”，“数据湖探索”，“dli:jobs:list_all”。单击“确认”，创建策略。

图 1-3 创建策略



- c. 在“用户组”页面，找到需要授权的用户所属的用户组，单击用户组名称，进入用户组“权限管理”页面，单击“配置权限”。
- d. 为对应的用户组授权：
- 在以下作用范围：选择“区域级项目”。
 - 拥有以下权限：勾选**b**中创建的权限策略。
或者勾选系统角色“DLI Service Admin”（权限较大，拥有DLI所有权限）也可以实现Flink作业的查看。

1.1.3 Flink 作业设置“异常自动重启”

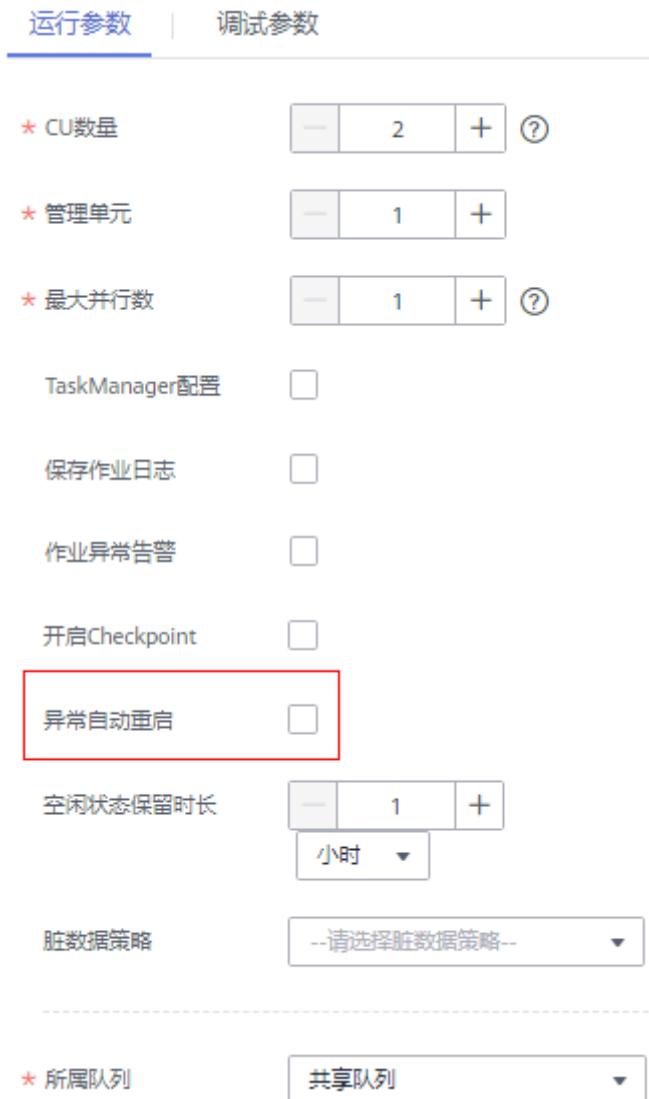
场景概述

DLI Flink作业具有高可用保障能力，通过设置“异常自动重启”功能，可在周边服务短时故障恢复后自动重启。

操作步骤

1. 登录DLI控制台，选择“作业管理”>“Flink作业”。
2. 在Flink作业编辑页面，勾选“异常自动重启”。例如，[图1-4所示](#)。

图 1-4 Flink SQL 作业编辑界面



1.1.4 Flink 作业如何保存作业日志

在创建Flink SQL作业或者Flink Jar作业时，可以在作业编辑页面，勾选“保存作业日志”参数，将作业运行时的日志信息保存到OBS。

勾选“保存作业日志”参数后，需配置“OBS桶”参数，选择OBS桶用于保存用户作业日志信息。如果选择的OBS桶是未授权状态，需要单击“OBS授权”。

日志信息的保存路径为：“桶名/jobs/logs/作业id开头的目录”。其中，“桶名”可以自定义。“/jobs/logs/作业id开头的目录”为固定格式。

在作业列表中，单击对应的作业名称，然后在“运行日志”页签，可以单击页面提供的OBS链接跳转至对应的路径下。

关于如何创建Flink SQL作业或者Flink Jar作业，请参考《[数据湖探索用户指南](#)》。

1.1.5 如何查看 Flink 作业输出结果？

- DLI支持将流作业的结果数据输出到DIS上，因此用户需要到DIS上去查看输出结果，详细操作请参见《[数据接入服务用户指南](#)》中的“从DIS获取数据”章节。
- DLI支持将Flink作业的结果数据输出到RDS上，因此用户需要到RDS上去查看输出结果，详细操作请参见《[关系型数据库快速入门](#)》。
- DLI支持将流作业的结果数据输出到SMN上，SMN服务会将输出结果反馈到用户绑定的终端设备上，详细操作请参见《[消息通知服务快速入门](#)》。
- DLI支持将Flink作业的结果数据输出到Kafka上，因此用户需要到Kafka集群上去查看输出结果，详细操作请参见[Kafka官网](#)。
- DLI支持将Flink作业的结果数据输出到CloudTable上，因此用户需要到CloudTable上去查看输出结果，详细操作请参见《[表格存储服务用户指南](#)》中的“入门”章节。
- DLI支持将Flink作业的结果数据输出到云搜索服务，因此用户需要到云搜索服务上去查看输出结果，详细操作请参见《[云搜索服务快速入门](#)》。
- DLI支持将Flink作业的结果数据输出到DCS上，因此用户需要到DCS上去查看输出结果，详细操作请参见《[分布式缓存服务快速入门](#)》。

1.1.6 Flink 作业管理界面对用户进行授权时报用户不存在怎么办

问题现象

在“作业管理 > Flink作业”，对应作业“操作”列，选择“更多 > 权限管理”，对新用户进行授权操作时报“No such user. userName:xxxx.” 错误。

解决方案

需要确认下当前用户名是否存在，并且该用户需要成功登录一次华为云系统才能进行授权操作。

1.1.7 手动停止了 Flink 作业，再次启动时没有提示从哪个 Checkpoint 恢复

问题现象

在创建Flink作业时开启了Checkpoint，指定了Checkpoint保存的OBS桶。手工停止Flink作业后，再次启动该Flink作业没有提示从哪个Checkpoint恢复。

解决方案

由于Flink Checkpoint和Savepoint生成机制及格式一致，因而，也可以通过Flink作业列表“操作”列中的“更多 > 导入保存点”，导入OBS中最新成功的Checkpoint，并从中恢复。

1. 登录DLI管理控制台，选择“作业管理 > Flink作业”。
2. 在对应Flink作业所在行，选择“操作 > 导入保存点”。
3. 在导入保存点界面选择Checkpoint的OBS桶路径，Checkpoint保存路径为：“桶名/jobs/checkpoint/作业id开头的目录”。单击“确定”完成导入操作。

4. 再次启动Flink作业，即从对应的检查点路径恢复作业。

1.1.8 DLI Flink 现在支持到哪个版本了？是否支持 Flink 1.13？下一个版本是哪个版本？

当前Flink 最高支持Flink 1.12的版本。当前不支持Flink 1.13。DLI Flink下一个版本将支持Flink 1.15, 敬请期待。

1.1.9 DLI 使用 SMN 主题，提示 SMN 主题不存在，怎么处理？

设置DLI Flink作业的运行参数时，勾选“作业异常告警”参数，可在作业出现运行异常或者欠费情况时，将作业异常告警信息，以SMN的方式通知用户。

如何自定义SMN主题，请参见[《消息通知服务用户指南》](#)中“创建主题”章节。

如果使用已创建的SMN主题时，提示SMN主题不存在，请进入统一身份认证服务(IAM)，选择对应子账户所在的用户组，添加相应Region的SMN策略。

1.2 Flink SQL 作业相关问题

1.2.1 Flink SQL 作业的消费能力如何，即一天可以处理多大的数据量？

Flink SQL作业的消费能力与源端的数据发送、队列大小、作业参数配置均有关系，每秒10M峰值。

1.2.2 Flink SQL 中的 temp 流中数据是否需要定期清理，如何清理？

Flink SQL中的temp流类似于子查询，只是逻辑意义上的流，用于简化SQL逻辑，不会产生数据存储，因而不存在清理问题。

1.2.3 创建 Flink SQL 作业时选择 OBS 桶，提示未授权

- 问题描述

用户创建Flink SQL作业，配置参数时，选择自己创建的OBS桶，提示“该OBS桶未授权。立即授权”，单击“立即授权”后提示“服务器内部出错了，请联系华为云客服或者稍后重试”，无法授权。

- 解决方案

在报错页面，通过F12查看错误详细信息：

```
{"error_code":"DLI.10001","error_msg":"服务内部出错了。{0}请联系华为云客服或者稍后重试","error_json_opt":{"error":"Unexpected exception[NoSuchElementException: None.get]"}}
```

查看用户是否创建DLI委托，发现用户没有创建委托权限，在“全局配置”>“服务授权”页面勾选“Tenant Administrator（全局服务）”权限后，重试可以给OBS桶授权。

1.2.4 Flink SQL 作业将 OBS 表映射为 DLI 的分区表

场景概述

用户使用Flink SQL作业时，需要创建OBS分区表，用于后续进行批处理。

操作步骤

该示例将car_info数据，以day字段为分区字段，parquet为编码格式（目前仅支持parquet格式），转储数据到OBS。更多内容请参考《[数据湖探索Flink SQL语法参考](#)》。

```
create sink stream car_infos (
    carId string,
    carOwner string,
    average_speed double,
    day string
) partitioned by (day)
with (
    type = "filesystem",
    file.path = "obs://obs-sink/car_infos",
    encode = "parquet",
    ak = "{{myAk}}",
    sk = "{{mySk}}"
);
```

数据最终在OBS中的存储目录结构为：obs://obs-sink/car_infos/day=xx/part-x-x。

数据生成后，可通过如下SQL语句建立OBS分区表，用于后续批处理：

1. 创建OBS分区表。

```
create table car_infos (
    carId string,
    carOwner string,
    average_speed double
)
partitioned by (day string)
stored as parquet
location 'obs://obs-sink/car-infos';
```

2. 从关联OBS路径中恢复分区信息。

```
alter table car_infos recover partitions;
```

1.2.5 Flink SQL 作业 Kafka 分区数增加或减少，不用停止 Flink 作业，实现动态感知

- 问题描述

用户执行Flink Opensource SQL, 采用Flink 1.10版本。初期Flink作业规划的Kafka的分区数partition设置过小或过大，后期需要更改Kafka区分数。

- 解决方案

在SQL语句中添加如下参数：

```
connector.properties.flink.partition-discovery.interval-millis="3000"
```

增加或减少Kafka分区数，不用停止Flink作业，可实现动态感知。

1.2.6 OBS 表如何映射为 DLI 的分区表？

该示例将car_info数据，以day字段为分区字段，parquet为编码格式（目前仅支持parquet格式），转储数据到OBS。更多内容请参考《[数据湖探索Flink SQL语法参考](#)》。

```
create sink stream car_infos (
    carId string,
    carOwner string,
    average_speed double,
    day string
) partitioned by (day)
with (
    type = "filesystem",
    file.path = "obs://obs-sink/car_infos",
    encode = "parquet",
    ak = "{{myAk}}",
    sk = "{{mySk}}"
);
```

数据最终在OBS中的存储目录结构为：obs://obs-sink/car_infos/day=xx/part-x-x。

数据生成后，可通过如下SQL语句建立OBS分区表，用于后续批处理：

1. 创建OBS分区表。

```
create table car_infos (
    carId string,
    carOwner string,
    average_speed double
)
partitioned by (day string)
stored as parquet
location 'obs://obs-sink/car-infos';
```

2. 从关联OBS路径中恢复分区信息。

```
alter table car_infos recover partitions;
```

1.2.7 在 Flink SQL 作业中创建表使用 EL 表达式，作业运行报 DLI.0005 错误

问题现象

Flink SQL作业创建表时，表名使用EL表达式，运行作业时报如下错误：

```
DLI.0005: AnalysisException: t_user_message_input_#{{date_format(date_sub(current_date(), 1), 'yyyymmddhhmmss')}} is not a valid name for tables/databases. Valid names only contain alphabet characters, numbers and _.
```

解决方案

需要将SQL中表名的“#”字符改成“\$”即可。DLI中使用EL表达式的格式为：\${expr}。

1.2.8 Flink 作业输出流写入数据到 OBS，通过该 OBS 文件路径创建的 DLI 表查询无数据

问题现象

使用Flink作业输出流写入数据到了OBS中，通过该OBS文件路径创建的DLI表进行数据查询时，无法查询到数据。

例如，使用如下Flink结果表将数据写入到OBS的“obs://obs-sink/car_infos”路径下。

```
create sink stream car_infos_sink (
    carId string,
    carOwner string,
    average_speed double,
```

```
buyday string
) partitioned by (buyday)
with (
    type = "filesystem",
    file.path = "obs://obs-sink/car_infos",
    encode = "parquet",
    ak = "{{myAk}}",
    sk = "{{mySk}}"
);
```

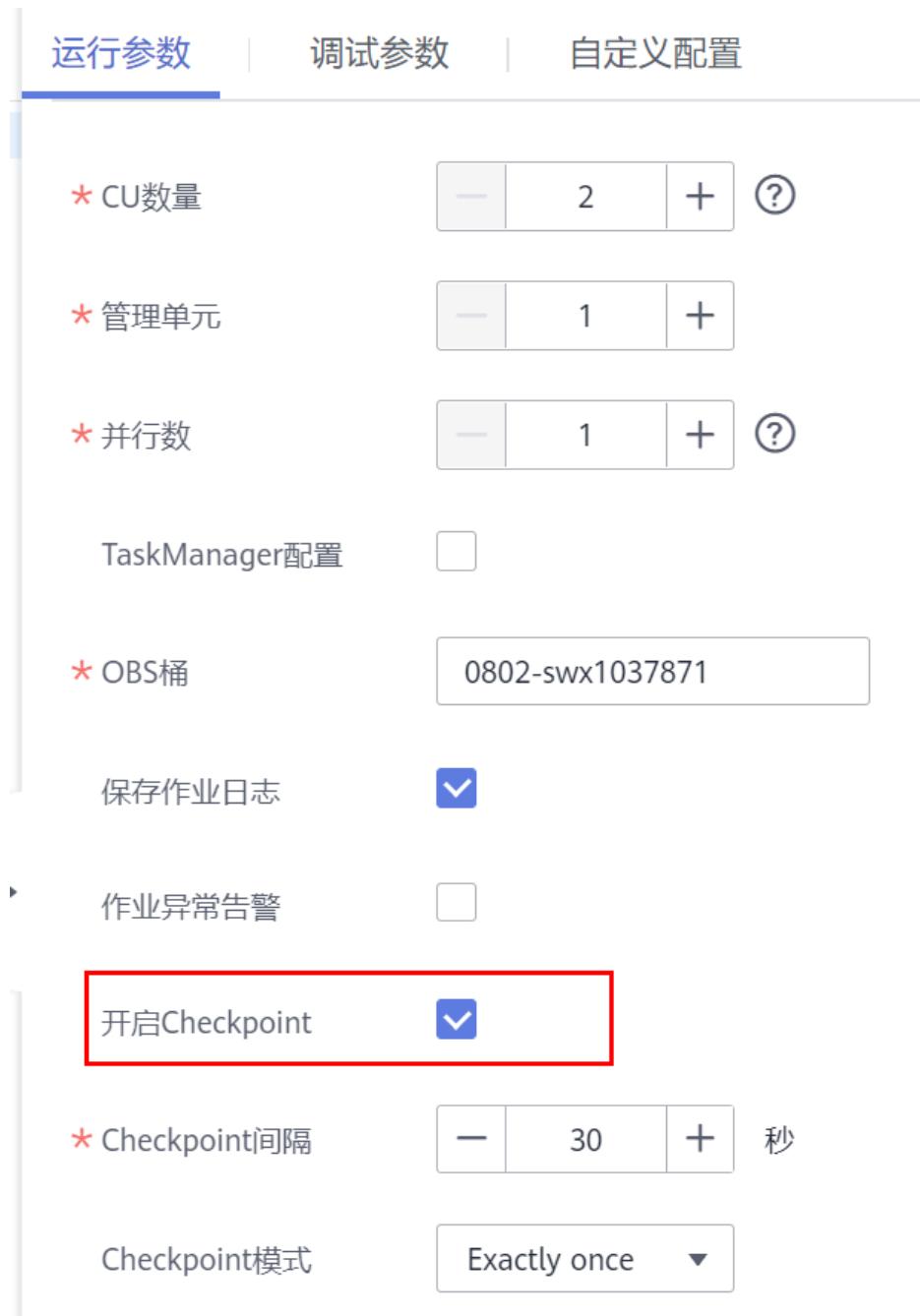
通过该OBS文件路径创建DLI分区表，在DLI查询car_infos表数据时没有查询到数据。

```
create table car_infos (
    carId string,
    carOwner string,
    average_speed double
)
partitioned by (buyday string)
stored as parquet
location 'obs://obs-sink/car_infos';
```

解决方案

1. 在DLI创建Flink结果表到OBS的作业时，如上述举例中的car_infos_sink表，是否开启了Checkpoint。如果未开启则需要开启Checkpoint参数，重新运行作业生成OBS数据文件。
开启Checkpoint步骤如下。
 - a. 到DLI管理控制台，左侧导航栏选择“作业管理 > Flink作业”，在对应的Flink作业所在行，操作列下单击“编辑”。
 - b. 在“运行参数”下，查看“开启Checkpoint”参数是否开启。

图 1-5 开启 Checkpoint



2. 确认Flink结果表的表结构和DLI分区表的表结构是否保持一致。如问题描述中 car_infos_sink 和 car_infos 表的字段是否一致。
3. 通过OBS文件创建DLI分区表后，是否执行以下命令从OBS路径中恢复分区信息。如下，在创建完DLI分区表后，需要恢复DLI分区表car_infos分区信息。

```
alter table car_infos recover partitions;
```

1.2.9 Flink SQL 作业运行失败，日志中有 connect to DIS failed java.lang.IllegalArgumentException: Access key cannot be null 错误

问题现象

在DLI上提交Flink SQL作业，作业运行失败，在作业日志中有如下报错信息：
connect to DIS failed java.lang.IllegalArgumentException: Access key cannot be null

问题根因

该Flink SQL作业在配置作业运行参数时，有选择保存作业日志或开启Checkpoint，配置了OBS桶保存作业日志和Checkpoint。但是运行该Flink SQL作业的IAM用户没有OBS写入权限导致该问题。

解决方案

1. 登录IAM控制台页面，单击“用户”，在搜索框中选择“用户名”，输入运行作业的IAM用户名。
2. 单击查询到用户名，查看该用户对应的用户组。
3. 单击“用户组”，输入查询到的用户组查询，单击用户组名称，在“授权记录”中查看当前用户的权限。
4. 确认当前用户所属用户组下的权限是否包含OBS写入的权限，比如“OBS OperateAccess”。如果没有OBS写入权限，则给对应的用户组进行授权。
5. 授权完成后，等待5到10分钟等待权限生效。再次运行失败的Flink SQL作业，查看作业运行状态。

1.2.10 Flink SQL 作业读取 DIS 数据报 Not authorized 错误

问题现象

Flink SQL作业读取DIS数据，运行该作业时，语义校验失败。具体作业失败提示信息如下：

Get dis channel xxx info failed. error info: Not authorized, please click the overview page to do the authorize action

问题原因

运行Flink作业前，没有对运行的用户帐号授权获取DIS数据的权限。

解决方案

1. 登录到DLI管理控制台，左侧导航栏选择“全局配置 > 服务授权”。
2. 在服务授权管理界面，勾选“DIS Administrator”权限，单击“更新委托授权”完成对当前用户的DIS权限授权。
3. 在“作业管理 > Flink作业”，单击对应的Flink SQL作业，重新启动和运行该作业。

1.2.11 Flink SQL 作业消费 Kafka 后 sink 到 es 集群，作业执行成功，但未写入数据

问题现象

客户创建Flink SQL作业，消费Kafka后sink到es集群，作业执行成功，但无数据。

原因分析

查看客户作业脚本内容，排查无问题，作业执行成功，出现该问题可能的原因如下：

- 数据不准确。
- 数据处理有问题。

处理步骤

步骤1 在Flink UI查看task日志，发现报错中提到json体，基本确定原因为数据格式问题。

步骤2 排查客户实际数据，发现客户Kafka数据存在多层嵌套的复杂json体。不支持解析。

步骤3 有两种方式解决此问题：

- 通过udf成jar包的形式
- 修改配置

步骤4 修改源数据格式，再次执行作业，无问题。

----结束

1.2.12 Flink Opensource SQL 如何解析复杂嵌套 JSON?

- **kafka message**

```
{  
    "id": 1234567890,  
    "name": "swq",  
    "date": "1997-04-25",  
    "obj": {  
        "time1": "12:12:12",  
        "str": "test",  
        "lg": 1122334455  
    },  
    "arr": [  
        "ly",  
        "zpk",  
        "swq",  
        "zjy"  
    ],  
    "rowinarr": [  
        {  
            "f1": "f11",  
            "f2": 111  
        },  
        {  
            "f1": "f12",  
            "f2": 222  
        }  
    ],  
    "time": "13:13:13",  
    "timestamp": "1997-04-25 14:14:14",  
    "map": {  
    }  
}
```

```
        "flink": 123
    },
    "mapinmap": {
        "inner_map": {
            "key": 234
        }
    }
}
```

- **flink opensource sql**

```
create table kafkaSource(
    id      BIGINT,
    name    STRING,
    `date`   DATE,
    obj     ROW<time1 TIME,str STRING,lg BIGINT>,
    arr     ARRAY<STRING>,
    rowinarr ARRAY<ROW<f1 STRING,f2 INT>>,
    `time`   TIME,
    `timestamp` TIMESTAMP(3),
    `map`    MAP<STRING,BIGINT>,
    mapinmap MAP<STRING,MAP<STRING,INT>>
) with (
    'connector' = 'kafka',
    'topic' = 'topic-swq-3',
    'properties.bootstrap.servers' = '10.128.0.138:9092,10.128.0.119:9092,10.128.0.212:9092',
    'properties.group.id' = 'swq-test',
    'scan.startup.mode' = 'latest-offset',
    'format' = 'json'
);
create table printSink (
    id      BIGINT,
    name    STRING,
    `date`   DATE,
    str     STRING,
    arr     ARRAY<STRING>,
    nameinarray STRING,
    rowinarr ARRAY<ROW<f1 STRING,f2 INT>>,
    f2      INT,
    `time`   TIME,
    `timestamp` TIMESTAMP(3),
    `map`    MAP<STRING,BIGINT>,
    flink   BIGINT,
    mapinmap MAP<STRING,MAP<STRING,INT>>,
    `key`   INT
) with ('connector' = 'print');

insert into
    printSink
select
    id,
    name,
    `date`,
    obj.str,
    arr,
    arr[4],
    rowinarr,
    rowinarr[1].f2,
    `time`,
    `timestamp`,
    `map`,
    `map`['flink'],
    mapinmap,
    mapinmap['inner_map']['key']
from kafkaSource;
```

- **result**

```
+I(1234567890,swq,1997-04-25,test,[ly, zpk, swq, zjy],zjy,[f11,111,
f12,222],111,13:13:13,1997-04-25T14:14:14,{flink=123},123,{inner_map={key=234}},234)
```

⚠ 注意

1. 各数据类型获取元素的方法：
 - map: map['key']
 - array: array[index]
 - row: row.key
2. array 的起始下标从 1 开始，即 array[1] 是 array 的第一个元素。
3. array 的元素必须同类型，row 的元素可以不同类型。

1.2.13 Flink Opensource SQL 从 RDS 数据库读取的时间和 RDS 数据库存储的时间为什么会不一致？

该问题的根因是数据库设置的时区不合理，一般来说，该问题出现时Flink读取的时间和RDS数据库的时间会相差13小时。

请在RDS数据库内执行如下语句

```
show variables like '%time_zone%'
```

通常结果如下：

Variable_name	Value
system_time_zone	CST
time_zone	SYSTEM

- **time_zone才是数据库时区**。这里它指向 'SYSTEM'，也就是数据库服务器的系统时间（'system_time_zone'）。而这个系统时间在这里指向 CST，所以，最终数据时区才是 CST。
- **system_time_zone是数据库所在服务器的时区，服务器是台主机**。
如本地数据库所在计算机的默认时区是中国标准时间，则查出来 'system_time_zone' 是 CST。

这不仅是重名的问题，而且在 mysql 的 time_zone 是 SYSTEM，system_time_zone 是 CST 的情况下会造成 bug。因为 CST 在 mysql 里被理解为 China Standard Time (UTC+8)，但在 Java 里被理解为 Central Standard Time (USA) (UTC-5)。Flink taskmanager 本质是一个 java 进程，在 mysql 的 jdbc 驱动的代码里会设置时区，这个时区是通过 TimeZone.getTimeZone(canonicalTimezone) 读取的。也就是说，读取的是 CST (UTC+8)，但真正设置的时区却是 CST (UTC-5)。

解决方案（二选一即可）

1. 数据库设置 time_zone 的值为非 SYSTEM，比如 +08:00。
2. jdbcUrl 带上时区。
例如 ‘jdbc:mysql://localhost:3306/test?serverTimezone=Asia/Shanghai’ 。

1.2.14 Flink SQL 和 Flink Opensource SQL 的语法有什么区别？

Flink SQL是DLI早期的自研语法，不兼容开源语法。

Flink Opensource SQL完全兼容Flink开源语法，随开源Flink 更新不断迭代。

因此推荐您使用Flink Opensource SQL。

语法参考：

- [Flink Opensource SQL1.12](#)（主力版本，推荐使用）。
- [Flink Opensource SQL1.10](#)。
- [Flink SQL](#)（已EOS，即将下线）。

1.2.15 Flink Opensource SQL Elasticsearch 结果表 failure-handler 参数填写 retry_rejected 导致提交失败

问题说明

Flink Opensource SQL Elasticsearch结果表failure-handler参数填写retry_rejected导致提交失败

问题根因

该问题属于开源设计缺陷。

解决措施

您可以尝试将retry_rejected修改为retry-rejected。

1.3 Flink Jar 作业相关问题

1.3.1 Flink Jar 作业配置 checkpoint 保存到 OBS

Flink Jar作业配置checkpoint保存到OBS步骤如下：

1. 在Flink Jar作业的Jar包代码中加入如下代码：

```
//StreamExecutionEnvironment 依赖的pom文件配置参考后续说明
StreamExecutionEnvironment env = StreamExecutionEnvironment.getExecutionEnvironment();

env.getCheckpointConfig().setCheckpointingMode(CheckpointingMode.EXACTLY_ONCE);
env.getCheckpointConfig().setCheckpointInterval(40000);

env.getCheckpointConfig().enableExternalizedCheckpoints(CheckpointConfig.ExternalizedCheckpointCl
eanup.RETAIN_ON_CANCELLATION);
RocksDBStateBackend rocksDbBackend = new RocksDBStateBackend(new
FsStateBackend("obs://${bucket}/jobs/checkpoint/my_jar"), false);
rocksDbBackend.setOptions(new OptionsFactory() {
    @Override
    public DBOptions createDBOptions(DBOptions currentOptions) {
        return currentOptions
            .setMaxLogFileSize(64 * 1024 * 1024)
            .setKeepLogFileNum(3);
    }
}

@Override
```

```
public ColumnFamilyOptions createColumnOptions(ColumnFamilyOptions currentOptions) {
    return currentOptions;
}
});
env.setStateBackend(rocksDbBackend);
```

说明

上述代码含义是以EXACTLY_ONCE模式，每隔40s保存checkpoint到OBS的\${bucket}桶中的jobs/checkpoint/my_jar路径。

其中，最重要的是保存checkpoint路径。一般是将checkpoint存入OBS桶中，路径格式如下：

- 路径格式：obs://\${bucket}/xxx/xxx/xxx
- 示例：
obs://\${mybucket}/jobs/checkpoint/jar-3
- StreamExecutionEnvironment依赖的包需要在pom文件中添加如下配置。完整pom文件可以参考[使用Flink Jar写入数据到OBS](#)。

```
<dependency>
    <groupId>org.apache.flink</groupId>
    <artifactId>flink-streaming-java_${scala.binary.version}</artifactId>
    <version>${flink.version}</version>
    <scope>provided</scope>
</dependency>
```

2. 在DLI Flink Jar作业中配置“优化参数”和“从checkpoint恢复”功能。

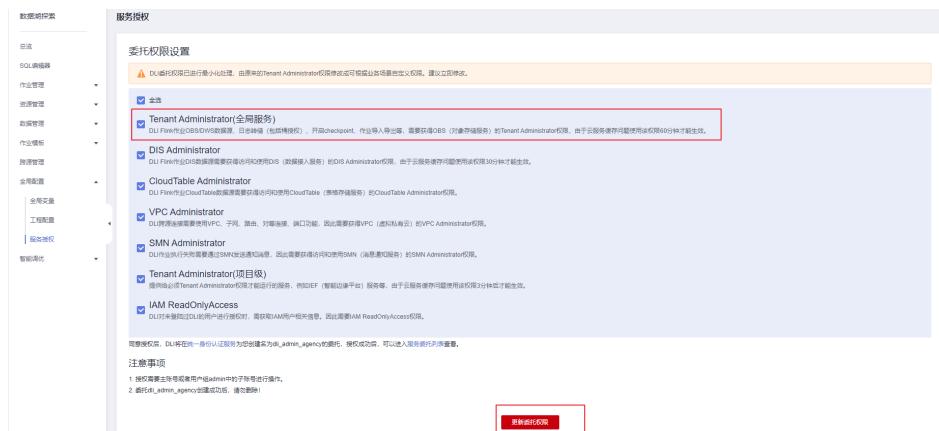
图 1-6 配置“优化参数”和“从 checkpoint 恢复”



- 参数优化的约束与限制

- 需要在DLI控制台下“全局配置 > 服务授权”开启Tenant Administrator（全局服务）。

图 1-7 开启开启 Tenant Administrator (全局服务)



ii. 写入数据到OBS的桶必须为主帐号下所创建的OBS桶。

- 配置从checkpoint恢复

i. 勾选“异常自动重启”。

ii. 勾选“从checkpoint恢复”，填写“Checkpoint路径”。

Checkpoint路径与用户在Jar包中设置的checkpoint路径相对应，格式如下：

- o “Checkpoint路径”格式为：\${bucket}/xxx/xxx/xxx

- o 示例：

如果Jar包中代码配置为：obs://mybucket/jobs/checkpoint/jar-3

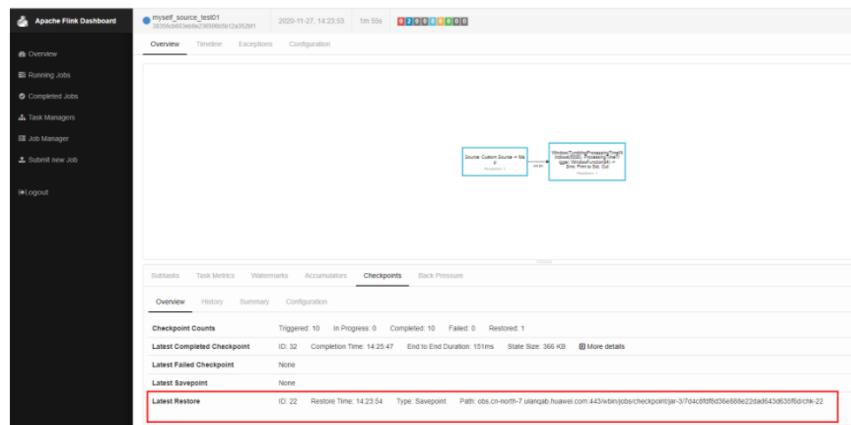
那么“Checkpoint路径”填写为：mybucket/jobs/checkpoint/jar-3

说明

- 每个Flink Jar作业配置的Checkpoint路径要保持不同，否则无法从准确的 checkpoint路径恢复。
- checkpoint路径中的OBS桶需要给DUI授权，DUI服务才能访问此桶下的文件。

3. 查看作业是否从checkpoint恢复。

图 1-8 查看是否恢复



1.3.2 Flink Jar 作业是否支持上传配置文件，要如何操作？

Flink Jar 作业上传配置文件操作流程

自定义(JAR)作业支持上传配置文件。

1. 将配置文件通过程序包管理上传到DLI；
2. 在Flink jar作业的其他依赖文件参数中，选择创建的DLI程序包；
3. 在代码中通过ClassName.class.getClassLoader().getResource("userData/fileNamed")加载该文件(其中，“fileName”为需要访问的文件名，“ClassName”为需要访问该文件的类名)。

配置文件使用方法

- 方案一：直接在main函数里面加载文件内容到内存，然后广播到各个taskmanager，这种方式适合那种需要提前加载的少量变量。
- 方案二：在open里面初始化算子的时候加载文件，可以使用相对路径/绝对路径的方式
以kafka sink为例：需要加载两个文件（userData/kafka-sink.conf, userData/client.truststore.jks）

- 使用相对路径的配置示例：

```
使用相对路径: confPath = userData/kafka-sink.conf
@Override
public void open(Configuration parameters) throws Exception {
    super.open(parameters);
    initConf();
    producer = new KafkaProducer<>(props);
}
private void initConf() {
    try {
        URL url = DliFlinkDemoDis2Kafka.class.getClassLoader().getResource(confPath);
        if (url != null) {
            LOGGER.info("kafka main-url: " + url.getFile());
        } else {
            LOGGER.info("kafka url error.....");
        }
        InputStream inputStream = new BufferedInputStream(new FileInputStream(new
File(url.getFile()).getAbsolutePath()));
        props.load(new InputStreamReader(inputStream, "UTF-8"));
        topic = props.getProperty("topic");
        partition = Integer.parseInt(props.getProperty("partition"));
        validProps();
    } catch (Exception e) {
        LOGGER.info("load kafka conf failed");
        e.printStackTrace();
    }
}
```

图 1-9 相对路径配置示例

```
ssl.secure.random.implementation = null
ssl.trustmanager.algorithm = PKIX
ssl.truststore.location = userData/client.truststore.jks
ssl.truststore.password = [hidden]
ssl.truststore.type = JKS
transaction.timeout.ms = 60000
transactional.id = null
value.serializer = class org.apache.kafka.common.serialization.StringSerializer
```

- 使用绝对路径的配置示例：

```
使用绝对路径: confPath = userData/kafka-sink.conf / path = /opt/data1/hadoop/tmp/  
usercache/omm/appcache/application_xxx_0015/container_xxx_0015_01_000002/userData/  
client.truststore.jks  
@Override  
public void open(Configuration parameters) throws Exception {  
    super.open(parameters);  
    initConf();  
    String path = DliFlinkDemoDis2Kafka.class.getClassLoader().getResource("userData/  
client.truststore.jks").getPath();  
    LOGGER.info("kafka abs path " + path);  
    props.setProperty("ssl.truststore.location", path);  
    producer = new KafkaProducer<>(props);  
}  
private void initConf() {  
    try {  
        URL url = DliFlinkDemoDis2Kafka.class.getClassLoader().getResource(confPath);  
        if (url != null) {  
            LOGGER.info("kafka main-url: " + url.getFile());  
        } else {  
            LOGGER.info("kafka url error.....");  
        }  
        InputStream inputStream = new BufferedInputStream(new FileInputStream(new  
File(url.getFile()).getAbsolutePath()));  
        props.load(new InputStreamReader(inputStream, "UTF-8"));  
        topic = props.getProperty("topic");  
        partition = Integer.parseInt(props.getProperty("partition"));  
        validProps();  
    } catch (Exception e) {  
        LOGGER.info("load kafka conf failed");  
        e.printStackTrace();  
    }  
}
```

图 1-10 绝对路径配置示例

```
ssl.secure.random.implementation = null  
ssl.trustmanager.algorithm = PKIX  
ssl.truststore.location = /opt/data1/hadoop/tmp/usercache/omm/appcache/application  
ssl.truststore.password = [hidden]  
ssl.truststore.type = JKS  
transaction.timeout.ms = 60000  
transactional.id = null  
value.serializer = class org.apache.kafka.common.serialization.StringSerializer
```

1.3.3 Flink Jar 包冲突，导致提交失败

问题描述

用户Flink程序的依赖包与DLI Flink平台的内置依赖包冲突，导致提交失败。

解决方案

查看是否已包含DLI Flink运行平台中已经存在的包，如果存在，则需要将自己的Jar包删除。

DLI内置依赖包请参考《[数据湖探索用户指南](#)》。

1.3.4 Flink Jar 作业访问 DWS 启动异常，提示客户端连接数太多错误

问题描述

提交Flink Jar作业访问DWS数据仓库服务时，提示启动失败，作业日志报如下错误信息。

```
FATAL: Already too many clients, active/non-active/reserved: 5/508/3
```

原因分析

当前访问的DWS数据库连接已经超过了最大连接数。错误信息中，non-active的个数表示空闲连接数，例如，non-active为508，说明当前有大量的空闲连接。

解决方案

出现该问题时建议通过以下操作步骤解决。

1. 登录DWS命令执行窗口，执行以下SQL命令，临时将所有non-active的连接释放掉。

```
SELECT PG_TERMINATE_BACKEND(pid) from pg_stat_activity WHERE state='idle';
```
2. 检查应用程序是否未主动释放连接，导致连接残留。建议优化代码，合理释放连接。
3. 在GaussDB(DWS) 控制台设置会话闲置超时时长session_timeout，在闲置会话超过所设定的时间后服务端将主动关闭连接。
session_timeout默认值为600秒，设置为0表示关闭超时限制，一般不建议设置为0。
session_timeout设置方法如下：
 - a. 登录GaussDB(DWS) 管理控制台。
 - b. 在左侧导航栏中，单击“集群管理”。
 - c. 在集群列表中找到所需要的集群，单击集群名称，进入集群“基本信息”页面。
 - d. 单击“参数修改”页签，修改参数“session_timeout”，然后单击“保存”。
 - e. 在“修改预览”窗口，确认修改无误后，单击“保存”。

更多问题处理步骤，请参考[DWS数据库连接问题](#)。

1.3.5 Flink Jar 作业运行报错，报错信息为 Authentication failed

问题现象

Flink Jar作业运行异常，作业日志中有如下报错信息：

```
org.apache.flink.shaded.curator.org.apache.curator.ConnectionState - Authentication failed
```

问题原因

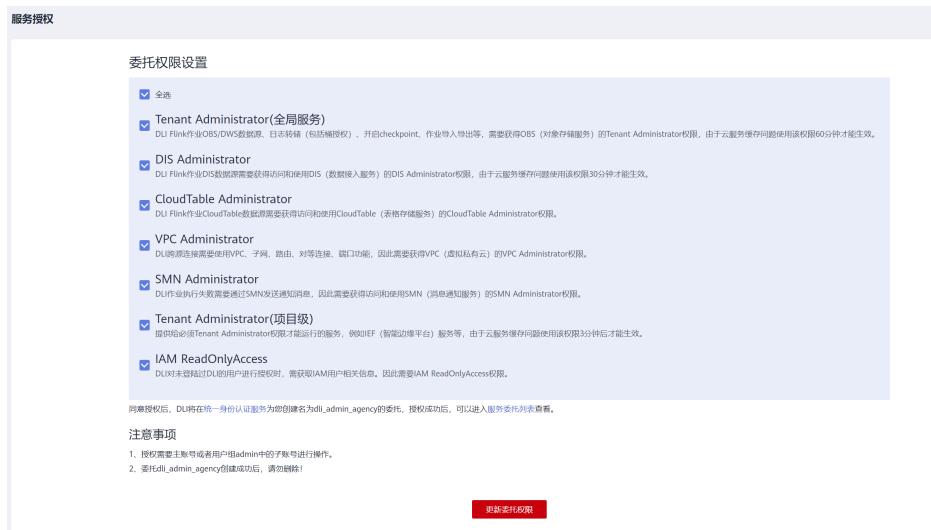
因为帐号没有在全局配置中配置服务授权，导致该帐号在创建跨源连接访问外部数据时因为权限不足而导致跨源访问失败。

解决方案

步骤1 登录DLI管理控制台，选择“全局配置 > 服务授权”。

步骤2 在服务授权界面，全选委托权限。

图 1-11 委托权限设置



步骤3 单击“更新委托授权”。界面会提示“委托权限更新成功”，表示修改成功。

步骤4 委托授权完成后，重新创建跨源连接和运行作业。

----结束

1.3.6 Flink Jar 作业设置 backend 为 OBS，报错不支持 OBS 文件系统

问题现象

客户执行Flink Jar作业，通过设置checkpoint存储在OBS桶中，作业一直提交失败，并伴有报错提交日志，提示OBS桶名不合法。

关于如何配置checkpoint保存到OBS请参考《[数据湖探索常见问题](#)》。

原因分析

- 确认OBS桶名是否正确。
- 确认所用AKSK是否有权限。
- 设置依赖关系provided防止Jar包冲突。
- 确认客户esdk-obs-java-3.1.3.jar的版本。
- 确认是集群存在问题。

处理步骤

步骤1 设置依赖关系provided。

步骤2 重启clusteragent应用集群升级后的配置。

步骤3 去掉OBS依赖，否则checkpoint会写不进OBS。

----结束

1.3.7 Hadoop jar 包冲突，导致 Flink 提交失败

问题现象

Flink 提交失败，异常为：

```
Caused by: java.lang.RuntimeException: java.lang.ClassNotFoundException: Class
org.apache.hadoop.fs.obs.metrics.OBSAMetricsProvider not found
at org.apache.hadoop.conf.Configuration.getClass(Configuration.java:2664)
at org.apache.hadoop.conf.Configuration.getClass(Configuration.java:2688)
... 31 common frames omitted
Caused by: java.lang.ClassNotFoundException: Class org.apache.hadoop.fs.obs.metrics.OBSAMetricsProvider
not found
at org.apache.hadoop.conf.Configuration.getClassByName(Configuration.java:2568)
at org.apache.hadoop.conf.Configuration.getClass(Configuration.java:2662)
... 32 common frames omitted
```

原因分析

Flink jar包冲突。用户提交的flink jar 与 DLI 集群中的hdfs jar包存在冲突。

处理步骤

步骤1 1. 将用户pom文件中的的hadoop-hdfs设置为：

```
<dependency>
<groupId>org.apache.hadoop</groupId>
<artifactId>hadoop-hdfs</artifactId>
<version>${hadoop.version}</version>
<scope> provided </scope>
</dependency>
```

或使用exclusions标签将其排除关联。

步骤2 若使用到hdfs的配置文件，则需要将core-site.xml、hdfs-site.xml、yarn-site.xml 修改为mrs-core-site.xml、mrs-hdfs-site.xml、mrs-hbase-site.xml

```
conf.addResource(HBaseUtil.class.getClassLoader().getResourceAsStream("mrs-core-site.xml"), false);
conf.addResource(HBaseUtil.class.getClassLoader().getResourceAsStream("mrs-hdfs-site.xml"), false);
conf.addResource(HBaseUtil.class.getClassLoader().getResourceAsStream("mrs-hbase-site.xml"), false);
```

----结束

1.3.8 Flink jar 如何连接 SASL_SSL?

使用Flink Jar连接开启SASL_SSL认证的Kafka。

具体操作请参考[使用Flink Jar连接开启SASL_SSL认证的Kafka](#)

1.4 性能调优

1.4.1 Flink 作业推荐配置指导

用户在创建Flink作业时，可以通过如下配置实现流应用的高可靠性能。

1. 用户在消息通知服务（SMN）中提前创建一个“主题”，并将其指定的邮箱或者手机号添加至主题订阅中。此时指定的邮箱或者手机会收到请求订阅的通知，单击链接确认订阅即可。

图 1-12 创建主题

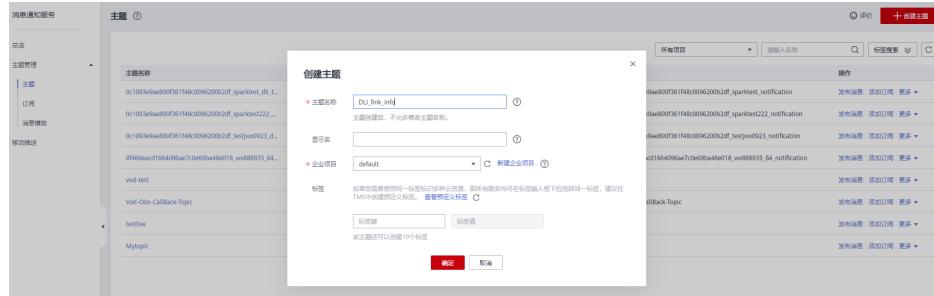


图 1-13 添加订阅



2. 登录DLI控制台，创建Flink SQL作业，编写作业SQL后，配置“运行参数”。

说明

Flink Jar作业可靠性配置与SQL作业相同，不再另行说明。

- a. 根据如下公式，配置作业的“CU数量”、“管理单元”与“最大并行数”：
$$\text{CU数量} = \text{管理单元} + (\text{算子总并行数} / \text{单TM Slot数}) * \text{单TM所占CU数}$$
例如：CU数量为9CU，管理单元为1CU，最大并行数为16，则计算单元为8CU。
如果不手动配置TaskManager资源，则单TM所占CU数默认为1，单TM slot数显示值为0，但实际上，单TM slot数值依据上述公式计算结果为2。
如果手动配置TaskManager资源，请依据上述公式计算配置，建议作业最大并行数为计算单元2倍为宜。
- b. 勾选“保存作业日志”，选择一个OBS桶。如果该桶未授权，需要单击“立即授权”进行授权。配置该参数，可以在作业异常失败后，将作业日志保存到用户的OBS桶下，方便用户定位故障原因。

图 1-14 保存作业日志



- c. 勾选“作业异常告警”，选择1中创建的“SMN主题”。配置该参数，可以在作业异常情况下，向用户指定邮箱或者手机发送消息通知，方便客户及时感知异常。

图 1-15 作业异常告警



- d. 勾选“开启Checkpoint”，依据自身业务情况调整Checkpoint间隔和模式。Flink Checkpoint机制可以保证Flink任务突然失败时，能够从最近的Checkpoint进行状态恢复重启。

图 1-16 checkpoint 参数



说明

- “Checkpoint间隔”为两次触发Checkpoint的间隔，执行Checkpoint机制会影响实时计算性能，配置间隔时间需权衡对业务的性能影响及恢复时长，**最好大于Checkpoint的完成时间**，建议设置为5分钟。
 - Exactly Once模式保证每条数据只被消费一次，At Least Once模式每条数据至少被消费一次，请依据业务情况选择。
- e. 勾选“异常自动恢复”和“从Checkpoint恢复”，根据自身业务情况选择重试次数。
- f. 配置“脏数据策略”，依据自身的业务逻辑和数据特征选择忽略、抛出异常或者保存脏数据。
- g. 选择“运行队列”。提交并运行作业。
3. 登录云监控服务CES控制台，在“云服务监控”列表中找到“数据湖探索”服务。在Flink作业中找到目标作业，单击“创建告警规则”。

图 1-17 云服务监控

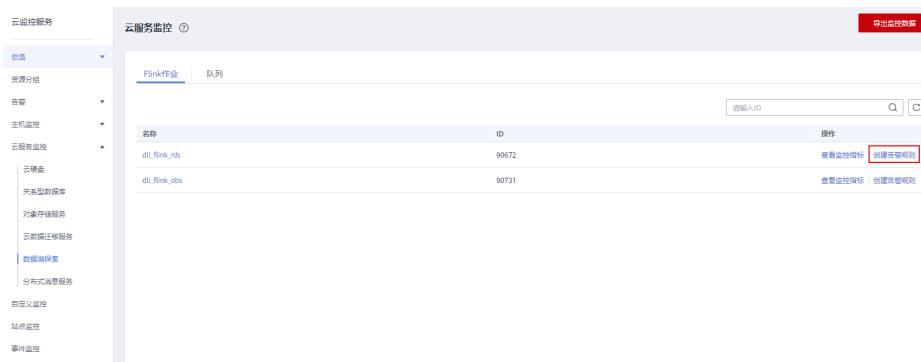


图 1-18 创建告警规则



DLI 为 Flink 作业提供了丰富的监控指标，用户可以依据自身需求使用不同的监控指标定义告警规则，实现更细粒度的作业监控。

监控指标说明请参考《数据湖探索用户指南》>《[数据湖探索监控指标说明](#)》。

1.4.2 Flink 作业如何进行性能调优

概念说明及监控查看

- 消费组积压

消费组积压可通过topic最新数据offset减去该消费组已提交最大offset计算得出，说明的是该消费组当前待消费的数据总量。

如果Flink作业对接的是kafka专享版，则可通过云监控服务(CES)进行查看。具体可选择“云服务监控 > 分布式消息服务 > kafka专享版”，单击“kafka实例名称 > 消费组”，选择具体的消费组名称，查看消费组的指标信息。

图 1-19 消费组



- 反压状态

反压状态是通过周期性对taskManager线程的栈信息采样，计算被阻塞在请求输出Buffer的线程比率来确定，默认情况下，比率在0.1以下为OK，0.1到0.5为LOW，超过0.5则为HIGH。

- 时延

Source端会周期性地发送带当前时间戳的LatencyMarker，下游算子接收到该标记后，通过当前时间减去标记中带的时间戳的方式，计算时延指标。算子的反压状态和时延可以通过Flink UI或者作业任务列表查看，一般情况下反压和高时延成对出现：

图 1-20 反压状态和时延

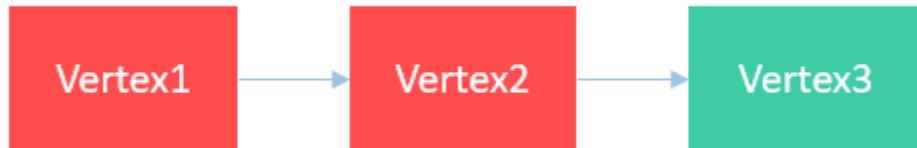
The figure shows a screenshot of the Flink task list interface. At the top, there are tabs for '任务列表' (Task List), '执行计划' (Execution Plan), '提交日志' (Submission Log), '运行日志' (Runtime Log), and '标签' (Tags). The '任务列表' tab is selected. The table below lists various tasks with their details:

名称	持续时间	最大并行度	任务	状态	反压状态	时延	发送...
GroupAggregate(groupBy=[pay_date...]	4d 16h ...	10	0 0 1 0 0 0 0	运行中	OK	156	1577976
GroupAggregate(groupBy=[data_dat...]	4d 16h ...	10	0 0 1 0 0 0 0	运行中	OK	210	2461839
Rank(strategy=[RetractStrategy], ran...	4d 16h ...	1	0 0 1 0 0 0 0	运行中	OK	767526	688989
Sink: JDBCUpsertTableSink(data_title...	4d 16h ...	1	0 0 1 0 0 0 0	运行中	OK	767578	0
SourceConversion(table=[default_cat...]	4d 16h ...	10	0 0 1 0 0 0 0	运行中	HIGH	218844	2503366
Rank(strategy=[AppendFastStrategy]...)	4d 16h ...	10	0 0 1 0 0 0 0	运行中	HIGH	3801276	1005370
GroupAggregate(groupBy=[pay_date...]	4d 16h ...	10	0 0 1 0 0 0 0	运行中	HIGH	7801878	1612469
GroupAggregate(groupBy=[data_dat...]	4d 16h ...	10	0 0 1 0 0 0 0	运行中	HIGH	12432845	2547727
Rank(strategy=[RetractStrategy], ran...	4d 16h ...	1	0 0 1 0 0 0 0	运行中	OK	22022942	630857
Sink: JDBCUpsertTableSink(data_title...	4d 16h ...	1	0 0 1 0 0 0 0	运行中	OK	22022996	0

性能分析

由于Flink的反压机制，流作业在存在性能问题的情况下，会导致数据源消费速率跟不上生产速率，从而引起Kafka消费组的积压。在这种情况下，可以通过算子的反压和时延，确定算子的性能瓶颈点。

- 作业最后一个算子(Sink)反压正常（绿色），前面算子反压高（红色）



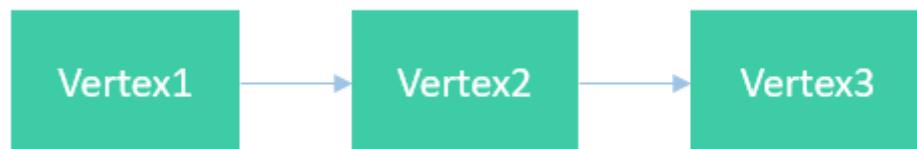
该场景说明性能瓶颈点在sink，此时需要根据具体数据源具体优化，比如对于JDBC数据源，可以通过调整写出批次(connector.write.flush.max-rows)、JDBC参数重写(rewriteBatchedStatements=true)等进行优化。

- 作业非倒数第二个算子反压高（红色）



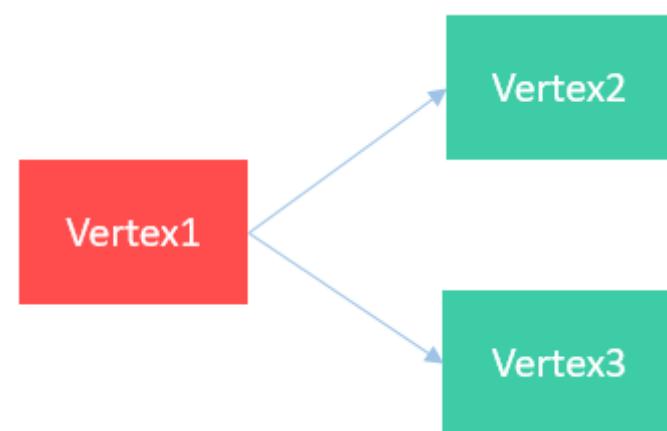
该场景说明性能瓶颈点在Vertex2算子，可以通过查看该算子描述，确认该算子具体功能，以进行下一步优化。

- 所有算子反压都正常（绿色），但存在数据堆积



该场景说明性能瓶颈点在Source，主要是受数据读取速度影响，此时可以通过增加Kafka分区数并增加source并发解决。

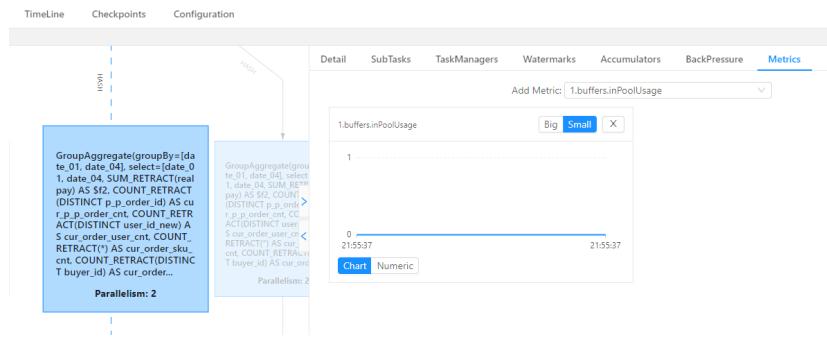
- 作业一个算子反压高（红色），而其后续的多个并行算子都不存在反压（绿色）



该场景说明性能瓶颈在Vertex2或者Vertex3，为了进一步确定具体瓶颈点算子，可以在FlinkUI页面开启inPoolUsage监控。如果某个算子并发对应的inPoolUsage

长时间为100%，则该算子大概率为性能瓶颈点，需分析该算子以进行下一步优化。

图 1-21 inPoolUsage 监控



性能调优

- rocksdb状态调优
 - topN排序、窗口聚合计算以及流流join等都涉及大量的状态操作，因而如果发现这类算子存在性能瓶颈，可以尝试优化状态操作的性能。主要可以尝试通过如下方式优化：
 - 增加状态操作内存，降低磁盘IO
 - 增加单slot cu资源数
 - 配置优化参数：
 - taskmanager.memory.managed.fraction=xx
 - state.backend.rocksdb.block.cache-size=xx
 - state.backend.rocksdb.writebuffer.size=xx
 - 开启微批模式，避免状态频繁操作
 - 配置参数：
 - table.exec.mini-batch.enabled=true
 - table.exec.mini-batch.allow-latency=xx
 - table.exec.mini-batch.size=xx
 - 使用超高IO本地盘规格机型，加速磁盘操作
 - group agg单点及数据倾斜调优

按天聚合计算或者group by key不均衡场景下，group聚合计算存在单点或者数据倾斜问题，此时，可以通过将聚合计算拆分成Local-Global进行优化。配置方式为设置调优参数: table.optimizer.aggphase-strategy=TWO_PHASE
 - count distinct优化
 - 在count distinct关联key比较稀疏场景下，即使使用Local-Global，单点问题依然非常严重，此时可以通过配置以下调优参数进行分桶拆分优化：
 - table.optimizer.distinct-agg.split.enabled=true
 - table.optimizer.distinct-agg.split.bucket-num=xx

- 使用filter替换case when:

例如：

```
COUNT(DISTINCT CASE WHEN flag IN ('android', 'iphone') THEN user_id ELSE NULL END) AS app_uv
```

可调整为

```
COUNT(DISTINCT user_id) FILTER(WHERE flag IN ('android', 'iphone')) AS app_uv
```

- 维表join优化

维表join根据左表进入的每条记录join关联键，先在缓存中匹配，如果匹配不到，则从远程拉取。因而，可以通过如下方式优化：

- 增加JVM内存并增加缓存记录条数
- 维表设置索引，加快查询速度

1.4.3 如何在一个 Flink 作业中将数据写入到不同的 Elasticsearch 集群中？

在对应的Flink作业中添加如下SQL语句。

```
create source stream ssource(xx);
create sink stream es1(xx) with (xx);
create sink stream es2(xx) with (xx);
insert into es1 select * from ssource;
insert into es2 select * from ssource;
```

1.4.4 Flink 作业重启后，如何保证不丢失数据？

DLI Flink提供了完整可靠的Checkpoint/Savepoint机制，您可以利用该机制，保证在手动重启或者作业异常重启场景下，不丢失数据。

- 为了避免系统故障导致作业异常自动重启后，丢失数据：

- 对于Flink SQL作业，您可以勾选“开启Checkpoint”，并合理配置Checkpoint间隔（权衡执行Checkpoint对业务性能的影响以及异常恢复的时长），同时勾选“异常自动重启”，并勾选“从Checkpoint恢复”。配置后，作业异常重启，会从最新成功的Checkpoint文件恢复内部状态和消费位点，保证数据不丢失及聚合算子等内部状态的精确一致语义。同时，为了保证数据不重复，建议使用带主键数据库或者文件系统作为目标数据源，否则下游处理业务需要加上去重逻辑（最新成功Checkpoint记录位点到异常时间段内的数据会重复消费）。

图 1-22 Flink 作业配置参数



- 对于Flink Jar作业，您需要参考《[Flink 自定义作业如何配置checkpoint保存到OBS](#)》，在代码中开启Checkpoint，同时如果有自定义的状态需要保存，您还需要实现ListCheckpointed接口，并为每个算子设置唯一ID。然后在作业配置中，勾选“从Checkpoint恢复”，并准确配置Checkpoint路径。

图 1-23 开启 Checkpoint



说明

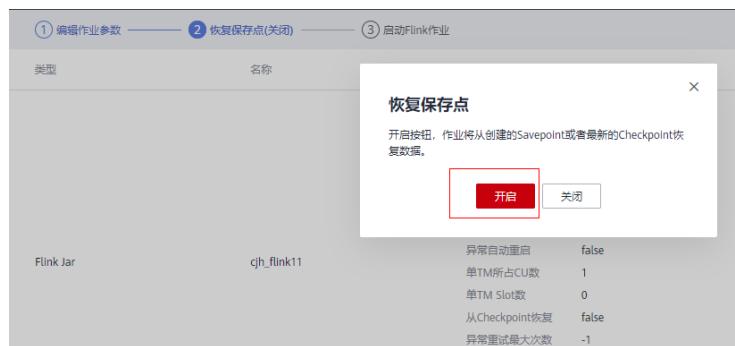
Flink Checkpoint机制可以保证Flink平台可感知内部状态的精确一致，但对于自定义Source/Sink或者有状态算子，需要合理实现ListCheckpointed接口，来保证业务数据需要的可靠性。

- 为了避免因业务修改等需要，手动重启作业后，不丢失数据：
 - 对于无内部状态的作业，您可以配置kafka数据源的启动时间或者消费位点到作业停止之前。
 - 对于有内部状态的作业，您可以在停止作业时，勾选“触发保存点”。成功后，再次启动作业时，开启“恢复保存点”，作业将从选择的savepoint文件中恢复消费位点及状态。同时，由于Flink Checkpoint和Savepoint生成机制及格式一致，因而，也可以通过Flink作业列表“操作”列中的“更多”>“导入保存点”，导入OBS中最新成功的Checkpoint，并从中恢复。

图 1-24 停止作业



图 1-25 恢复保存点



1.5 运维指导

1.5.1 Flink 作业提交错误，如何定位

1. 在Flink作业管理页面，将鼠标悬停到提交失败的作业状态上，查看失败的简要信息。
常见的失败原因可能包括：
 - CU资源不足：需扩容队列。
 - 生成jar包失败：检查SQL语法及UDF等。
2. 如果信息不足以定位或者是调用栈错误，可以进一步单击作业名称，进入到作业详情页面，选择“提交日志”页签，查看作业提交日志。

1.5.2 Flink 作业运行异常，如何定位

1. 在“Flink作业”管理页面，对应作业“操作”列单击“编辑”按钮，在作业运行界面确认作业是否勾选“保存作业日志”参数。

图 1-26 保存作业日志



- 是，则执行③。
 - 否，则运行日志不会转储OBS桶，需要先执行②保存作业运行日志。
2. 在作业运行界面勾选“保存作业日志”，在“OBS桶”参数选择存储运行日志的OBS桶。单击“启动”重新运行作业。作业重新运行完成后执行③及后续步骤。
 3. 在Flink作业列表单击对应作业名称，进入作业详情页面，选择“运行日志”页签。
 4. 单击OBS桶，获取对应作业的完整运行日志。

图 1-27 查看运行日志



5. 下载最新“jobmanager.log”文件，搜索“RUNNING to FAILED”关键字，通过上下文的错误栈，确认失败原因。
6. 如果“jobmanager.log”文件中的信息不足以定位，可以在运行日志中找到对应的“taskmanager.log”日志，搜索“RUNNING to FAILED”关键字，确认失败原因。

1.5.3 Flink 作业重启后，如何判断是否可以从 checkpoint 恢复

什么是从 checkpoint 恢复？

Flink Checkpoint 是一种容错恢复机制。这种机制保证了实时程序运行时，遇到异常或者机器问题时能够进行自我恢复。

从 checkpoint 恢复的原则

- 通常当作业执行失败、资源异常重启等非人为触发的异常场景时，支持从 checkpoint 恢复。
- 但是如果修改了作业的运算逻辑，作业的计算逻辑已发生更改，不支持从 checkpoint 恢复。

应用场景

我们列举了一些常见的从checkpoint恢复的场景供您参考，如表1-2所示。

更多场景你可以使用[从checkpoint恢复的原则](#)结合实际情况进行判断。

表 1-2 从 checkpoint 恢复的常见场景

场景	是否支持恢复	说明
调整或者增加并行数	不支持	该操作修改了作业的并行数，即修改了作业的运行逻辑。
修改Flink SQL语句、Flink Jar作业等操作	不支持	该操作修改了作业对资源的算法逻辑。 例如原有的算法的语句是执行加减运算，当前需要恢复的状态将算法的语句修改成为乘除取余的运算，是无法从checkpoint直接恢复的。
修改“静态流图”	不支持	该操作修改了作业对资源的算法逻辑。
修改“单TM所占CU数”参数	支持	对计算资源的修改并没有影响到作业算法或算子的运行逻辑。
作业运行异常或物理停电	支持	当对作业参数未作出修改。

1.5.4 作业语义检验时提示 DIS 通道不存在怎么处理？

处理方法如下：

1. 登录到DIS管理控制台，在左侧菜单栏选择“通道管理”。检查Flink作业SQL语句中的DIS通道是否存在。
2. 如果Flink作业中的DIS通道还未创建，请参见[《数据接入服务用户指南》](#)中“开通DIS通道”章节。
确保创建的DIS通道和Flink作业处于统一区域。
3. 如果DIS通道已创建，则检查确保DIS通道和Flink流作业是否处与同一区域。

1.5.5 如何处理作业的 OBS Bucket 没有授权？

用户在执行一个作业，提示OBS Bucket没有授权时，用户需要进行如下操作：

- 步骤1** 在“全局配置”>“服务授权”页面配置“Tenant Administrator（全局服务）”权限。
- 步骤2** 单击“作业管理”，找到操作的作业名称。
- 步骤3** 单击“操作”列表中的“编辑”，进入“作业编辑”页面。
- 步骤4** 用户需要在该作业的“作业编辑”页面进行“运行参数配置”。
1. 选择“开启Checkpoint”或“保存作业日志”。

2. 选择“OBS桶”。
3. 单击“OBS授权”。

----结束

1.5.6 DLI Flink 作业提交运行后（已选择保存作业日志到 OBS 桶），提交运行失败的情形（例如：jar 包冲突），有时日志不会写到 OBS 桶中

DLI Flink作业提交或运行失败时，对应生成的作业日志保存方式，包含以下三种情况：

- 提交失败，只会在submit-client下生成提交日志。
- 运行失败且在1分钟内的日志，可以直接在管理控制台页面查看，具体如下：
在“作业管理”>“Flink作业”页面，单击对应的作业名称，进入作业详情页面，单击“运行日志”可以查看实时日志。
- 运行失败且超过1分钟(日志转储周期1分钟)，会在application_xx下生成运行日志。

另外，由于DLI服务端已经内置了Flink的依赖包，并且基于开源社区版本做了安全加固。为了避免依赖包兼容性问题或日志输出及转储问题，打包时请注意排除以下文件：

- 系统内置的依赖包，或者在Maven或者Sbt构建工具中将scope设为provided
- 日志配置文件（例如：“log4j.properties”或者“logback.xml”等）
- 日志输出实现类JAR包（例如：log4j等）

在此基础上，taskmanager.log会随日志文件大小和时间滚动。

1.5.7 Kafka Sink 配置发送失败重试机制

• 问题描述

用户执行Flink Opensource SQL, 采用Flink 1.10版本。Flink Sink写Kafka报错后作业失败：

```
Caused by: org.apache.kafka.common.errors.NetworkException: The server disconnected before a response was received.
```

• 问题原因

由于CPU使用率过高，导致网络闪断。

• 解决方案

在SQL语句中配置发送失败重试：connector.properties.retries=5

```
create table kafka_sink(
    car_type string
    , car_name string
    , primary key (union_id) not enforced
) with (
    "connector.type" = "upsert-kafka",
    "connector.version" = "0.11",
    "connector.properties.bootstrap.servers" = "xxxx:9092",
    "connector.topic" = "kafka_car_topic",
    "connector.sink.ignore-retraction" = "true",
    "connector.properties.retries" = "5",
    "format.type" = "json"
);
```

1.5.8 访问 FlinkUI/SparkUI 页面无法正常显示怎么办？

问题现象

访问FlinkUI/SparkUI页面无法正常显示，打开页面返回信息不全。

根因分析

根据分析发现队列为按需队列，队列运行作业的时候会重新创建集群，大概需要10分钟左右才能完成集群创建。在集群创建完成之前，访问FlinkUI会导致缓存空的projectID，从而导致无法访问。出现该问题可能原因是由于集群未创建。

解决方案

建议将队列更换为专属队列，空闲状态下集群不会被释放。或者提交作业等一段时间之后在查看FlinkUI。

1.5.9 Jobmanager 与 Taskmanager 心跳超时，导致 Flink 作业异常怎么办？

问题现象

Jobmanager与Taskmanager心跳超时，导致Flink作业异常。

图 1-28 异常信息

```
Jobmanager.log
2021-05-17 22:24:37.312 INFO [70] org.apache.flink.runtime.checkpoint.CheckpointCoordinator - Triggering checkpoint 223 @ 1621262677310 for job 00d04eb6e7147e59f2d2877bdb48ce4d.
[70] 2021-05-17 22:46:01.726 INFO [70] org.apache.flink.runtime.executiongraph.ExecutionGraph - Map->Map (4/4) {9e6b13c7ab5d3637062f1697014b4eff} switched from RUNNING to FAILED.
java.util.concurrent.TimeoutException: Heartbeat of TaskManager with id container_1619690508608_1067_01_00003 timed out.
at org.apache.flink.runtime.jobmaster.JobMaster$TaskManagerHeartbeatListener.notifyHeartbeatTimeout(JobMaster.java:1642)
at org.apache.flink.runtime.heartbeat.HeartbeatManagerImpl$HeartbeatMonitor.run(HeartbeatManagerImpl.java:335)

container_1619690508608_1067_01_00003
// 可以发现zookeeper的client 也超时了
2021-05-17 22:46:44.078 INFO [85] org.apache.zookeeper.ClientCnxn - Client session timed out, have not heard from server in 64331ms for sessionid 0x18000013858becca, closing socket connection and attempting reconnect.
2021-05-17 22:46:39.547 INFO [85] org.apache.zookeeper.ClientCnxn - Connecting to node-master1.flnx.b1039268-c9c0-4c82-b21f-b1252bf186f.com:2181.
2021-05-17 22:47:00.358 INFO [22] org.apache.flink.shaded.zookeeper.org.apache.zookeeper.ClientCnxn - Unable to read additional data from server sessionid 0x2005c1ae84c0021, likely server has closed socket, closing socket connection and attempting reconnect.
2021-05-17 22:47:11.223 WARN [85] org.apache.zookeeper.ClientCnxn - SASL configuration failed: javax.security.auth.login.LoginException: No JAAS configuration section named 'Client' was found in specified JAAS configuration file: /tmp/jaas-2036661896892296518.conf. Will continue connection to Zookeeper server without SASL authentication, if Zookeeper server allows it.
```

根因分析

1. 检查网络是否发生闪断，分析集群负载是否很高。
2. 如果频繁出现Full GC, 建议排查代码，确认是否有内存泄漏。

图 1-29 Full GC

```
24461 [MemoryRegionAllocatorFailure] 2149423M<->1118208M[114832M], 0.24514K mem= 2149423M->2454253M[2454320M], [Metaspaces: 75823M->75823M[1118208M]], 0.0272735 sec[s] [Times: user=0.03 sys=0.00, real=0.03 sec[s]]
24462 2021-05-17T22:47:05.105+0800: 2454253M[2454320M], 0.24514K mem= 2149423M->2454253M[2454320M], [Metaspaces: 75823M->75823M[1118208M]], 0.0272735 sec[s] [Times: user=0.03 sys=0.00, real=0.03 sec[s]]
24463 2021-05-17T22:47:05.112+0800: 0.0452451 sec[s] [Times: user=0.17 sys=0.00, real=0.51 sec[s]] [GC (CMS Initial Mark) { CMS-Initial-Mark: 2149423M[2149432M] } 2454253M[2454320M], 0.0452451 sec[s] [Times: user=0.04 sys=0.00, real=0.03 sec[s]]
24464 2021-05-17T22:47:05.113+0800: 2454253M[2454320M], 0.0452451 sec[s] [Times: user=0.04 sys=0.00, real=0.03 sec[s]]
24465 2021-05-17T22:47:05.103-0800: 246957.751 [CMS0201-05-17T22:47:05.94+0800: 246957.951] [CMS-concurrent-mark: 0.236/0.244 sec[s]] [Times: user=0.24 sys=0.00, real=0.25 sec[s]] [GC (CMS Allocation Failure) { CMS-Allocation-Failure: 2149423M[2149432M], 0.8596435 sec[s] } 2454253M[2454320M], 0.0403087 sec[s] [Times: user=0.84 sys=0.00, real=0.04 sec[s]]]
24466 2021-05-17T22:47:05.104-0800: 246957.958 [CMS0201-05-17T22:47:05.95+0800: 246957.958] [CMS-concurrent-mark: 0.236/0.244 sec[s]] [GC (CMS Allocation Failure) { CMS-Allocation-Failure: 2149423M[2149432M], 0.8596435 sec[s] } 2454253M[2454320M], 0.0403087 sec[s] [Times: user=0.84 sys=0.00, real=0.04 sec[s]]]
24467 2021-05-17T22:47:05.114-0800: 246957.958 [CMS0201-05-17T22:47:05.95+0800: 246957.958] [CMS-concurrent-mark: 0.236/0.244 sec[s]] [GC (CMS Allocation Failure) { CMS-Allocation-Failure: 2149423M[2149432M], 0.8596435 sec[s] } 2454253M[2454320M], 0.0403087 sec[s] [Times: user=0.84 sys=0.00, real=0.04 sec[s]]]
24468 2021-05-17T22:47:05.115-0800: 246957.958 [CMS0201-05-17T22:47:05.95+0800: 246957.958] [CMS-concurrent-mark: 0.236/0.244 sec[s]] [GC (CMS Allocation Failure) { CMS-Allocation-Failure: 2149423M[2149432M], 0.8596435 sec[s] } 2454253M[2454320M], 0.0403087 sec[s] [Times: user=0.84 sys=0.00, real=0.04 sec[s]]]
24469 2021-05-17T22:47:05.116-0800: 246957.958 [CMS0201-05-17T22:47:05.95+0800: 246957.958] [CMS-concurrent-mark: 0.236/0.244 sec[s]] [GC (CMS Allocation Failure) { CMS-Allocation-Failure: 2149423M[2149432M], 0.8596435 sec[s] } 2454253M[2454320M], 0.0403087 sec[s] [Times: user=0.84 sys=0.00, real=0.04 sec[s]]]
24470 2021-05-17T22:47:05.156-0800: 246957.200 [CMS0201-05-17T22:47:05.157+0800: 246957.436] [CMS-concurrent-mark-start]
24471 2021-05-17T22:47:05.157-0800: 246957.200 [CMS0201-05-17T22:47:05.157+0800: 246957.200] [CMS-concurrent-mark: 0.235/0.243 sec[s]] [Times: user=0.25 sys=0.00, real=0.25 sec[s]]
```

处理步骤

- 如果频繁Full GC, 建议排查代码，是否有内存泄漏。
- 增加单TM所占的资源。

- 联系技术支持，修改集群心跳配置参数。

1.5.10 Flink jobmanager 日志一直报 Timeout expired while fetching topic metadata 怎么办？

1. 首先测试地址连通性。

操作方法请参考[测试地址连通性](#)。

2. 若果网络不可达，需要先配置网络连通。确保DLI 队列与外部数据源的网络连接性正常。

配置详情请参考[配置网络连通](#)。

2 SQL 作业相关问题

2.1 使用咨询

2.1.1 SQL 作业使用咨询

DLI 是否支持创建临时表？

问题描述：临时表主要用于存储临时中间结果，当事务结束或者会话结束的时候，临时表的数据可以自动删除。例如MySQL中可以通过：“create temporary table ...”语法来创建临时表，通过该表存储临时数据，结束事务或者会话后该表数据自动清除。当前DLI是否支持该功能？

解决措施：当前DLI不支持创建临时表功能，只能根据当前业务逻辑控制来实现相同功能。DLI支持的SQL语法可以参考[创建DLI表](#)。

可以本地连接 DLI 吗？支持远程工具连接吗？

暂不支持。请在控制台提交作业。

详细操作请参考[数据湖探索快速入门](#)。

DLI SQL 作业超过 12h 会被 kill 掉吗？

默认情况下，为了保障队列的稳定运行，超过12h的SQL作业会被系统按超时取消处理。

用户可以通过dli.sql.job.timeout（单位是秒）参数配置超时时间。

DLI 支撑本地测试 Spark 作业吗？

DLI暂不支持本地测试Spark作业，您可以安装DLI Livy工具，通过Livy工具提供的交互式会话能力调测Spark作业。

推荐使用[使用Livy提交Spark Jar作业](#)。

DLI 表(OBS 表 / DLI 表)数据支持删除某行数据吗?

DLI 表(OBS表 / DLI 表)数据暂不支持删除某行数据。

2.2 作业开发

2.2.1 如何合并小文件

使用SQL过程中，生成的小文件过多时，会导致作业执行时间过长，且查询对应表时耗时增大，建议对小文件进行合并。

1. 设置配置项。

```
spark.sql.shuffle.partitions = 分区数量 (即此场景下最终生成的文件数量)
```

2. 执行SQL。

```
INSERT OVERWRITE TABLE tablename  
select * FROM tablename distribute by rand()
```

2.2.2 DLI 如何访问 OBS 桶中的数据

1. 创建OBS表。

具体语法请参考《[数据湖探索SQL语法参考](#)》。

2. 添加分区。

具体语法请参考《[数据湖探索SQL语法参考](#)》。

3. 往分区导入OBS桶中的数据。

具体语法请参考《[数据湖探索SQL语法参考](#)》。

4. 查询数据。

具体语法请参考《[数据湖探索SQL语法参考](#)》。

2.2.3 创建 OBS 表时正确指定 OBS 路径

场景概述

创建OBS表时，OBS路径须指定到数据库下的具体表层路径。路径格式为：obs://xxx/
数据库名/表名。

创建OBS表更多语法介绍请参考《[数据湖探索Spark SQL语法参考](#)》。

正确示例

```
CREATE TABLE `di_seller_task_activity_30d`(`user_id` STRING COMMENT '用户ID') SORTED as parquet  
LOCATION 'obs://akc-bigdata/akdc.db/di_seller_task_activity_30d'
```

错误示例

```
CREATE TABLE `di_seller_task_activity_30d`(`user_id` STRING COMMENT '用户ID') SORTED as parquet  
LOCATION 'obs://akc-bigdata/akdc.db'
```

说明

如果指定路径为akdc.db时，进行insert overwrite操作时，会将akdc.db下的数据都清空，造成重大事故。

2.2.4 关联 OBS 桶中嵌套的 JSON 格式数据如何创建表

DLI支持关联OBS桶中嵌套的JSON格式数据，使用异步模式建表即可。

建表语句如下：

```
create table tb1 using json options(path 'obs://....')
```

2.2.5 count 函数如何进行聚合

使用count函数进行聚合的正确用法如下：

```
SELECT
    http_method,
    count(http_method)
FROM
    apigateway
WHERE
    service_id = 'ecs' Group BY http_method
```

或者

```
SELECT
    http_method
FROM
    apigateway
WHERE
    service_id = 'ecs' DISTRIBUTE BY http_method
```

错误用法：将会报错。

```
SELECT
    http_method,
    count(http_method)
FROM
    apigateway
WHERE
    service_id = 'ecs' DISTRIBUTE BY http_method
```

2.2.6 如何将一个区域中的 DLI 表数据同步到另一个区域中？

可以使用OBS跨区域复制功能实现，步骤如下：

1. 将区域一中的DLI表数据导出到自定义的OBS桶中。
具体请参考《[数据湖探索用户指南](#)》。
2. 通过OBS跨区域复制功能将数据复制至区域二的OBS桶中。
具体请参考《[对象存储服务用户指南](#)》。
3. 根据需要导入或使用对应的数据。

2.2.7 SQL 作业如何指定表的部分字段进行表数据的插入

当前DLI不支持指定部分列字段进行表数据插入，执行插入操作需要所有表字段数据一起插入。

2.2.8 怎样删除表数据？

可通过使用TRUNCATE清除DLI表或者OBS表的数据。

详情请参考[清空数据](#)。

2.3 作业运维报错

2.3.1 用户导表到 OBS 报 “path obs://xxx already exists” 错误

用户可新建一个不存在的OBS目录或手动删除已存在的OBS目录，再重新提交作业。删除已存在的OBS目录后，目录下的所有数据将会被删除。请谨慎执行此删除操作。

2.3.2 对两个表进行 join 操作时，提示：SQL_ANALYSIS_ERROR: Reference 't.id' is ambiguous, could be: t.id, t.id.;

出现这个提示，表示进行join操作的两个表中包含相同的字段，但是在执行命令时，没有指定该字段的归属。

例如：在表tb1和tb2中都包含字段“id”。

错误的命令：

```
select id from tb1 join tb2;
```

正确的命令：

```
select tb1.id from tb1 join tb2;
```

2.3.3 执行查询语句报错：The current account does not have permission to perform this operation, the current account was restricted. Restricted for no budget.

请先确认是否欠费，如有欠费请充值。

如果充值后仍然提示相同的错误，请退出帐号后重新登录。

2.3.4 执行查询语句报错：There should be at least one partition pruning predicate on partitioned table XX.YYY

- 原因分析：

上述报错信息说明：partitioned table XX.YYY执行查询时，其查询条件中未使用其表分区列。

查询分区表时，查询条件中每个分区表必须包含至少一个分区列才允许执行，否则不允许执行。

- 解决方法：

建议用户参考如下例子查询分区表：

其中partitionedTable为分区表，partitionedColumn为分区列，查询语句为：

```
SELECT * FROM partitionedTable WHERE partitionedColumn = XXX
```

查询每个分区表时必须包含至少一个分区条件。

2.3.5 LOAD 数据到 OBS 外表报错：IllegalArgumentException: Buffer size too small. size

问题描述

在Spark SQL作业中，使用LOAD DATA命令导入数据到DLI表中时报如下错误：

```
error.DLI.0001: IllegalArgumentException: Buffer size too small. size = 262144 needed = 2272881
```

或者如下错误

```
error.DLI.0999: InvalidProtocolBufferException: EOF in compressed stream footer position: 3 length: 479 range: 0 offset: 3 limit: 479 range 0 = 0 to 479 while trying to read 143805 bytes
```

问题原因

上述报错可能原因是当前导入的文件数据量较大，同时因为spark.sql.shuffle.partitions参数设置的并行度过大，导致缓存区大小不够而导致数据报错。

解决方案

建议可以尝试调小spark.sql.shuffle.partitions参数值来解决缓冲区不足问题。具体该参数设置步骤如下：

1. 登录DLI管理控制台，单击“作业管理 > SQL作业”，对应SQL作业行的操作列，单击“编辑”按钮，跳转到“SQL编辑器”。
2. 在“SQL编辑器”中，单击“设置”，参考如下图配置添加参数。

图 2-1 设置参数



3. 重新执行作业。

2.3.6 SQL 作业运行报错：DLI.0002 FileNotFoundException

问题现象

SQL作业执行报错，报错信息大致如下：

```
Please contact DLI service. DLI.0002: FileNotFoundException: getFileStatus on obs://xxx: status [404]
```

解决方案

请排查在同一时间点是否还有另外作业对当前报错作业操作的表信息有删除操作。

DLI不允许同时有多个作业在同一时间点对相同表进行读写操作，否则会造成作业冲突，导致作业运行失败。

2.3.7 用户通过 CTAS 创建 hive 表报 schema 解析异常错误

目前DLI支持hive语法创建TEXTFILE、SEQUENCEFILE、RCFILE、ORC、AVRO、PARQUET文件类型的表，如果用户CTAS建表指定的文件格式为AVRO类型，而且直接使用数字作为查询语句（SELECT）的输入，如“CREATE TABLE tb_avro STORED AS AVRO AS SELECT 1”则会报schema解析异常。

此问题的原因是如果不指定列名，则会把SELECT后的内容同时作为列名和插入值，而AVRO格式的表不支持列名为数字，所以会报解析schema异常错误。

用户可以通过“CREATE TABLE tb_avro STORED AS AVRO AS SELECT 1 AS colName”指定列名的方式解决该问题，或者将存储格式指定为除AVRO以外的其它格式。

2.3.8 在 DataArts Studio 上运行 DLI SQL 脚本，执行结果报 org.apache.hadoop.fs.obs.OBSIOException 错误

问题现象

在DataArts Studio上运行DLI SQL脚本，执行结果的运行日志显示语句执行失败，错误信息为：

```
DLI.0999: RuntimeException: org.apache.hadoop.fs.obs.OBSIOException: initializing on obs://xxx.csv: status [-1] - request id [null] - error code [null] - error message [null] - trace :com.obs.services.exception.ObsException: OBS servcie Error Message. Request Error:  
...  
Cause by: ObsException: com.obs.services.exception.ObsException: OBSs servcie Error Message. Request Error: java.net.UnknownHostException: xxx: Name or service not known
```

问题根因

第一次执行DLI SQL脚本，用户没有在DLI控制台上同意隐私协议导致在DataArts Studio运行SQL脚本报错。

解决方案

1. 登录DLI控制台，选择“SQL编辑器”，输入任意执行一个SQL语句，比如“select 1”。
2. 弹出隐私协议后，勾选“同意以上隐私协议”，单击“确定”。

说明

该隐私协议只需要在第一次执行时同意即可，后续再次运行不再会弹出和确认。

3. 重新在DataArts Studio上运行DLI SQL脚本，脚本运行正常。

2.3.9 使用 CDM 迁移数据到 DLI，迁移作业日志上报 UQUERY_CONNECTOR_0001:Invoke DLI service api failed 错误

问题现象

在CDM迁移数据到DLI，迁移作业提交后，在CDM作业迁移日志中查看作业执行失败，具体日志有如下报错信息：

```
org.apache.sqoop.common.SqoopException: UQUERY_CONNECTOR_0001:Invoke DLI service api failed,  
failed reason is %s.
```

```
at org.apache.sqoop.connector.uquery.intf.impl.UQueryWriter.close(UQueryWriter.java:42)
at org.apache.sqoop.connector.uquery.processor.Dataconsumer.run(Dataconsumer.java:217)
at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
at java.util.concurrent.FutureTask.run(FutureTask.java:266)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
at java.lang.Thread.run(Thread.java:748)
```

问题原因

在CDM界面创建迁移作业，配置DLI目的连接参数时，“资源队列”参数错误选成了DLI的“通用队列”，应该选择DLI的“SQL队列”。

解决方案

1. 登录DLI管理控制台，选择“队列管理”，在队列管理界面查看是否有“SQL队列”类型的队列。
 - 是，执行[3](#)。
 - 否，执行[2](#)购买“SQL队列”类型的队列。
2. 单击“购买队列”创建队列，其中队列类型选择“SQL队列”，选择其他参数后提交创建。
3. 在CDM侧重新配置迁移作业的DLI目的连接参数，其中资源队列”参数选择已创建的DLI“SQL队列”。
4. CDM重新提交迁移作业，查看作业执行日志。

2.3.10 SQL 作业访问报错：File not Found

问题现象

执行SQL作业访问报错：File not Found。

解决措施

文件报错找不到，一般是读写冲突产生的，建议查询一下SQL查询报错表的时候，是否有作业正在覆盖写对应数据。

2.3.11 SQL 作业访问报错：DLI.0003: AccessControlException XXX

问题现象

SQL作业访问报错：DLI.0003: AccessControlException XXX。

解决措施

请查看下AccessControlException写的OBS桶，确认当前帐号是否有访问桶的权限。

2.3.12 SQL 作业访问外表报错：DLI.0001: org.apache.hadoop.security.AccessControlException: verifyBucketExists on {{桶名}}: status [403]

问题现象

SQL作业访问外表报错：DLI.0001:
org.apache.hadoop.security.AccessControlException: verifyBucketExists on {{桶名}}:
status [403]。

解决措施

当前帐号没有访问该外表所在OBS桶的权限，请添加相应OBS权限再执行该查询。

2.3.13 执行 SQL 语句报错：The current account does not have permission to perform this operation, the current account was restricted. Restricted for no budget.

问题现象

执行SQL语句报错：The current account does not have permission to perform this operation, the current account was restricted. Restricted for no budget。

解决措施

请先确认帐号否欠费，如帐号欠费请续费后重试。

如果续费后重试仍然提示相同的错误，请退出帐号后重新登录。

2.4 运维指导

2.4.1 SQL 作业运行慢如何定位

作业运行慢可以通过以下步骤进行排查处理。

可能原因 1：FullGC 原因导致作业运行慢

判断当前作业运行慢是否是FullGC导致：

1. 登录DLI控制台，单击“作业管理 > SQL作业”。
2. 在SQL作业页面，在对应作业的“操作”列，单击“更多 > 归档日志”。

图 2-2 归档日志



3. 在OBS目录下，获取归档日志文件夹，详细如下。

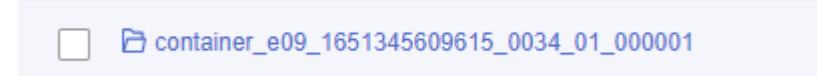
- Spark SQL作业：

查看带有“driver”或者为“container_xxx_000001”的日志文件夹则为需要查看的Driver日志目录。

图 2-3 带有 driver 的归档日志文件夹名示例



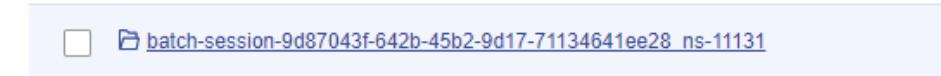
图 2-4 container_xxx_000001 归档日志文件夹示例



- Spark Jar作业：

Spark Jar作业的归档日志文件夹以“batch”开头。

图 2-5 Spark Jar 作业归档日志文件夹名示例



- 进入归档日志文件目录，在归档日志文件目录下，下载“gc.log.*”日志。
 - 打开已下载的“gc.log.*”日志，搜索“Full GC”关键字，查看日志中是否有时间连续，并且频繁出现“Full GC”的日志信息。

图 2-6 Full GC 日志

FullGC问题原因定位和解决

原因1 小文件过多：当一个表中的小文件过多时，可能会造成Driver内存FullGC。

1. 登录DLI控制台，选择SQL编辑器，在SQL编辑器页面选择问题作业的队列和数据库。
2. 执行以下语句，查看作业中表的文件数量。“表名”替换为具体问题作业中的表名称。

```
select count(distinct fn) FROM (select input_file_name() as fn from 表名) a
```
3. 如果小文件过多，则可以参考[如何合并小文件](#)来进行处理。

原因2 广播表：广播也可能会造成Driver内存的FullGC。

1. 登录DLI控制台，单击“作业管理 > SQL作业”。
2. 在SQL作业页面，在对应作业所在行，单击▼按钮，查看作业详情，获取作业ID。

图 2-7 获取作业 ID

The screenshot shows the 'SQL' tab of the 'Job Details' section. It displays a table with columns: 队列 (Queue), 执行引擎 (Execution Engine), 用户名 (User Name), 类型 (Type), 状态 (Status), and 执行语句 (Execution Statement). One row is selected, showing 'testdb' as the queue, 'spark' as the execution engine, and 'QUERY' as the type. The status is '已成功' (Success). The execution statement is a complex query involving COUNT(DISTINCT fn) and subqueries. Below the table, there's a detailed view of the job: 作业ID (Job ID) is e802011-0000-4219-00a0-0e709fe48f1, 作业位置 (Job Location) is QUERY, 执行时长 (Execution Duration) is 31.02s, 执行时间 (Execution Time) is 2022/07/12 19:22:04 GMT+00:00, 标签 (Tags) is empty, 已扫描的数据量 (Scanned Data Volume) is 70.07 MB, and 跟踪 (Tracking) is testdb.

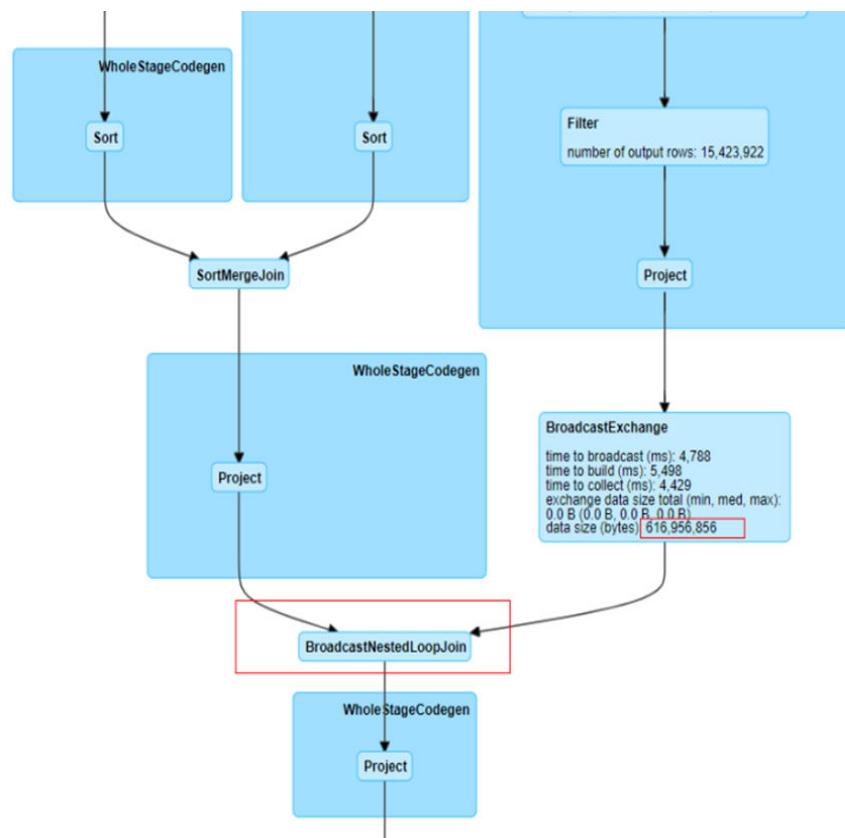
3. 在对应作业的“操作”列，单击“Spark UI”，进入“Spark UI”页面。
4. 在“Spark UI”页面，在上方菜单栏选择“SQL”。参考下图，根据作业ID，单击Description中的超链接。

图 2-8 单击作业链接

The screenshot shows the Spark UI interface with the 'SQL' tab selected. At the top, it says 'Completed Queries: 4' and 'Failed Queries: 1'. The 'Completed Queries (4)' section has a table with columns: ID, Description, Submitted, Duration, and Job IDs. The first row (ID 4) has a red box around its 'Description' link. The 'Failed Queries (1)' section has a table with columns: ID, Description, Submitted, Duration, Succeeded Job IDs, and Failed Job IDs. The single entry has a red box around its 'Description' link. The URL in the browser address bar is 'sql_test1_16cu_2.3.3.18/application UI'.

5. 查看对应作业的DAG图，判断是否有BroadcastNestedLoopJoin节点。

图 2-9 作业的 DAG 图。



6. 如果存在广播，则参考[SQL作业中存在join操作，因为自动广播导致内存不足，作业一直运行中](#)处理。

可能原因 2：数据倾斜

判断当前作业运行慢是否是数据倾斜导致：

1. 登录DLI控制台，单击“作业管理 > SQL作业”。
2. 在SQL作业页面，在对应作业所在行，单击▼按钮，查看作业详情信息，获取作业ID。

图 2-10 获得作业 ID

队列	执行引擎	用户名	类型	状态	执行语句	日期范围	输出结果时间	默认使用内联重试策略	操作
testdb	spark		QUERY	已成功		2022/07/11 19:10:29	31.82s	2022/07/12 19:21:32 GMT+08:00	▼ 调阅 Spark UI 更多

作业ID: es02c11-09d4-4219-88a0-0e709de4671
作业类型: QUERY
运行时长: 31.82s
结束时间: 2022/07/12 19:22:04 GMT+08:00
耗磁盘: -
已扫描的数据量: 70.87 MB
数据库: testdb

3. 在对应作业的“操作”列，单击“Spark UI”，进入到Spark UI页面。
4. 在“Spark UI”页面，在上方菜单栏选择“Jobs”。参考下图，根据作业ID，单击链接。

The screenshot shows the Apache Spark UI interface. The top navigation bar includes 'Jobs' (highlighted with a red box), 'Stages', 'Storage', 'Environment', 'Executors', and 'SQL'. The main area is titled 'Spark Jobs (7)'. It displays two sections: 'Completed Jobs (4)' and 'Failed Jobs (1)'. The 'Completed Jobs' section lists four entries with their descriptions, submission times, durations, and stage success rates. The 'Failed Jobs' section lists one entry with its description, submission time, duration, and failure count. A red box highlights the 'Description' column for the first completed job.

- 根据Active Stage可以看到当前正在运行的Stage运行情况，单击Description中的超链接。

The screenshot shows the 'Active Stages (1)' tab in the Apache Spark UI. It displays a single stage entry with its details: Stage Id, Pool Name, Description (highlighted with a red box), Submitted, Duration, and Tasks (Success/Total). A red box also highlights the 'Tasks: Success/Total' column.

- 在Stage中，可以看到每一个Task开始运行时间“Launch Time”，以及Task运行耗时时间“Duration”。
- 单击“Duration”，可以根据耗时进行排序，排查是否存在单个Task耗时过长导致整体作业时间变长。

参考图2-11可以看到数据倾斜时，单个任务的shuffle数据远大于其他Task的数据，导致该任务耗时时间变长。

图 2-11 数据倾斜示例图

The screenshot shows the 'Aggregated Metrics by Executor' tab in the Apache Spark UI. It displays a table of tasks with columns for Index, ID, Attempt, Status, Locality Level, Executor ID, Host, Launch Time, Duration, GC Time, Shuffle Read Size / Records, Write Time, Shuffle Write Size / Records, Shuffle Spill (Memory), and Shuffle Sp (Disk). Red boxes highlight the 'Launch Time', 'Duration', and 'Shuffle Read Size / Records' columns for several tasks, illustrating data skew.

数据倾斜原因和解决：

Shuffle的数据倾斜基本是由于join中的key值数量不均衡导致。

- 对join连接条件进行group by 和count，统计每个连接条件的key值的数量。示例如下：

lefttbl表和righttbl表进行join关联，其中lefttbl表的num为连接条件的key值。则可以对lefttbl.num进行group by和count统计。

```
SELECT * FROM lefttbl a LEFT join righttbl b on a.num = b.int2;
SELECT count(1) as count,num from lefttbl group by lefttbl.num ORDER BY count desc;
```

从图2-12可以看出，num为1的数量远大于其他值的数量。

图 2-12 示例数据统计演示

count ↓	num ↓
737	1
1	81
1	385

2. 考虑在对应key值上添加concat(cast(round(rand() * 999999999) as string)随机数进行打散。
3. 如果确实因为单个key值倾斜严重且不可对key值拼接随机值打散，则参考[配置AE参数解决数据倾斜](#)处理。

2.4.2 查看 DLI SQL 日志

场景概述

日常运维时需要查看DLI SQL日志。

操作步骤

1. 在DataArts Studio控制台获取DataArts Studio执行DLI作业的job id。

图 2-13 查看日志

作业名	状态	运行方式	计划开始时间	开始时间	运行时间 (min)	创建人	操作
dm_trfc_prd_high_commission_info	运行成功	正常调度	2021/03/19 10:01:00 GMT+08:00	2021/03/19 10:01:02 GMT+08:00	3.0	zhouxuelin	停止 复制 等待... 更多
dm_trfc_prd_high_commission_info	运行成功	正常调度	2021/03/19 10:01:00 GMT+08:00	2021/03/19 10:01:04 GMT+08:00	0	-	查看日志 更多

图 2-14 查找 job id

```
) t;
[2021/03/19 10:02:59 GMT+0800] [INFO] dli.sql.badRecordsPath=obs://dlf-log-060f26783a00109c2f7ac00d662b
[2021/03/19 10:03:00 GMT+0800] [INFO] DLI job id is:3d249de0-96c3-46d7-a788-adde1207376a
[2021/03/19 10:03:00 GMT+0800] [INFO] Spark UI path of DLI SQL Job [dm_trfc_prd_high_commission_info] is [http://10.0.0.19:4040]
[2021/03/19 10:03:00 GMT+0800] [INFO] submit job success. Waiting for job execution to complete...
[2021/03/19 10:04:00 GMT+0800] [INFO] DLI sql execute success
```

2. 在DLI控制台，选择“作业管理”>“SQL作业”。
3. 在SQL作业管理页面，输入对应的job id，找到对应的作业。
4. 在“操作”列中，单击“更多”>“归档日志”>“下载日志到本地”。

图 2-15 归档日志

作业名	执行引擎	启动时间	作业类型	状态	操作
sql1_1024	spark	2021/03/19 10:03:00 GMT+08:00	INSERT	已成功	停止 复制 等待... 更多

5. 在所下载的日志中搜索对应jobId，即可查看具体的执行日志。

2.4.3 查看 DLI 的执行 SQL 记录

场景概述

执行SQL作业过程中需要查看对应的记录。

操作步骤

1. 登录DLI管理控制台。
2. 在左侧导航栏单击“作业管理”>“SQL作业”进入SQL作业管理页面。
3. 输入作业ID或者执行的语句可以筛选所要查看的作业。

2.4.4 配置 AE 参数解决数据倾斜

场景概述

如果观察到SQL执行时间较长，可进入SparkUI查看对应SQL的执行状态。

如果观察到一个stage运行时间超过20分钟且只剩余一个task在运行，即为数据倾斜的情况。

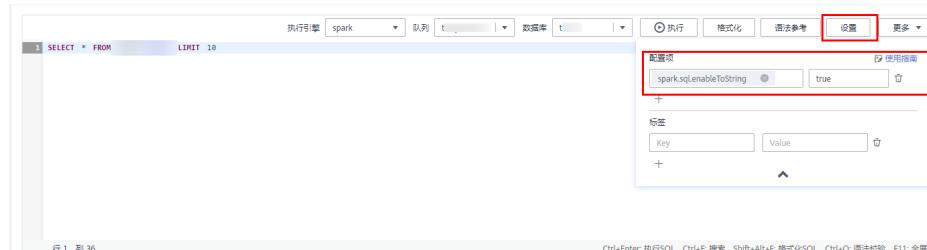
图 2-16 数据倾斜样例



Description	Submitted	Duration	Tasks: Succeeded/Total
a48e2dfa-bf14-461e-8863-be29f578e3b6 mapPartitionsWithIndexInternal at ShuffleExchangeExec.scala:296	2021/03/17 20:15:52	9.1 min	24/25 (1 running)

操作步骤

1. 登录数据湖探索管理控制台，选择“SQL作业”，在要修改的作业所在行的“操作”列，单击“编辑”进入SQL编辑器界面。
2. 在SQL编辑器界面，单击“设置”，在“配置项”尝试添加以下几个Spark参数进行解决。



参数项如下，冒号前是配置项，冒号后是配置项的值。

```
spark.sql.enableToString:false  
spark.sql.adaptive.join.enabled:true  
spark.sql.adaptive.enabled:true  
spark.sql.adaptive.skewedJoin.enabled:true  
spark.sql.adaptive.enableToString:false  
spark.sql.adaptive.skewedPartitionMaxSplits:10
```

说明

spark.sql.adaptive.skewedPartitionMaxSplits表示倾斜拆分力度，可不加，默认为5，最大为10。

- 单击“执行”重新运行作业，查看优化效果。

2.4.5 DLI 控制台中无法查询到对应表

问题现象

已知存在某DLI表，但在DLI页面查询不到该表。

问题根因

已有表但是查询不到时，大概率是因为当前登录的用户没有对该表的查询和操作权限。

解决措施

联系创建该表的用户，让该用户给需要操作该表的其他用户赋予查询和操作的权限。
赋权操作如下：

- 使用创建表的用户帐号登录到DLI管理控制台，选择“数据管理 > 库表管理”。
- 单击对应的数据库名称，进入到表管理界面。在对应表的“操作”列，单击“权限管理”，进入到表权限管理界面。
- 单击“授权”，授权对象选择“用户授权”，用户名选择需要授权的用户名，勾选对应需要操作的权限。如“查询表”、“插入”等根据需要勾选。
- 单击“确定”完成权限授权。
- 授权完成后，再使用已授权的用户登录DLI控制台，查看是否能正常查询到对应表。

2.4.6 OBS 表压缩率较高

当Parquet/Orc格式的OBS表对应的文件压缩率较高时（跟文本相比，超过5倍压缩率），建议在提交导入数据到DLI表作业时，在submit-job请求体conf字段中配置“dli.sql.files.maxPartitionBytes=33554432”，该配置项默认值为128MB，将其配置成32MB，可以减少单个任务读取的数据量，避免因过高的压缩比，导致解压后单个任务处理的数据量过大。

2.4.7 如何避免字符码不一致导致的乱码

DLI只支持UTF-8文本格式，执行创建表和导入操作时，用户的数据需要是以UTF-8编码。

2.4.8 删除表后再重新创建同名的表，需要对操作该表的用户和项目重新赋权

问题场景

A用户通过SQL作业在某数据库下创建了表testTable，并且授权testTable给B用户插入和删除表数据的权限。后续A用户删除了表testTable，并重新创建了同名的表testTable，如果希望B用户继续保留插入和删除表testTable数据的权限，则需要重新对该表进行权限赋予。

问题根因

删除表后再重建同名的表，该场景下表权限不会自动继承，需要重新对需要操作该表的用户或项目进行赋权操作。

解决方案

表删除再创建后，需要重新对需要操作该表的用户或项目进行赋权操作。具体操作如下：

1. 在管理控制台左侧，单击“数据管理”>“库表管理”。
2. 单击需要设置权限的表所在的数据库名，进入该数据库的“表管理”页面。
3. 单击所选表“操作”栏中的“权限管理”，将显示该表对应的权限信息。
4. 单击表权限管理页面右上角的“授权”按钮。
5. 在弹出的“授权”对话框中选择相应的权限。
6. 单击“确定”，完成表权限设置。

2.4.9 DLI 分区内表导入的文件不包含分区列的数据，导致数据导入完成后查询表数据失败

问题现象

DLI分区内表导入了CSV文件数据，导入的文件数据没有包含对应分区列的字段数据。分区表查询时需要指定分区字段，导致查询不到表数据。

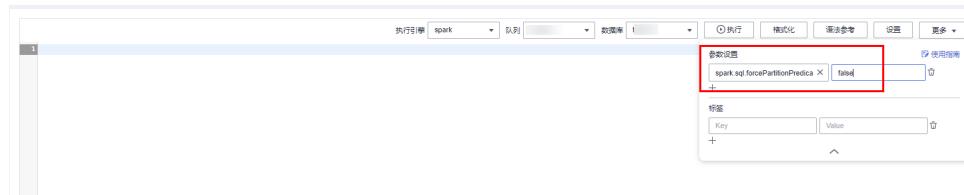
问题根因

DLI分区内表在导入数据时，如果文件数据没有包含分区字段，则系统会默认指定分区值“`__HIVE_DEFAULT_PARTITION__`”，当前Spark判断分区为空时，则会直接返回null，不返回具体的数据。

解决方案

1. 登录DLI管理控制台，在“SQL编辑器”中，单击“设置”。
2. 在参数设置中，添加参数
“`spark.sql.forcePartitionPredicatesOnPartitionedTable.enabled`”，值设置为“`false`”。

图 2-17 参数设置



3. 上述步骤参数设置完成后，则可以进行全表查询，不用查询表的时候要包含分区字段。

2.4.10 创建 OBS 外表，因为 OBS 文件中的某字段存在回车换行符导致表字段数据错误

问题现象

创建OBS外表，因为指定的OBS文件内容中某字段包含回车换行符导致表字段数据错误。

例如，当前创建的OBS外表语句为：

```
CREATE TABLE test06 (name string, id int, no string) USING csv OPTIONS (path "obs://dli-test-001/test.csv");
```

test.csv文件内容如下：

```
Jordon,88,"aa  
bb"
```

因为最后一个字段的aa和bb之间存在回车换行。创建OBS外表后，查询test06表数据内容显示如下：

```
name  id  classno  
Jordon  88  aa  
bb"  null  null
```

解决方案

创建OBS外表时，通过multiLine=true来指定列数据包含回车换行符。针对举例的建表语句，可以通过如下示例解决：

```
CREATE TABLE test06 (name string, id int, no string) USING csv OPTIONS (path "obs://dli-test-001/test.csv",multiLine=true);
```

2.4.11 SQL 作业中存在 join 操作，因为自动广播导致内存不足，作业一直运行中

问题现象

SQL作业中存在join操作，作业提交后状态一直是运行中，没有结果返回。

问题根因

Spark SQL作业存在join小表操作时，会触发自动广播所有executor，使得join快速完成。但同时该操作会增加executor的内存消耗，如果executor内存不够时，导致作业运行失败。

解决措施

1. 排查执行的SQL中是否有使用“`/*+ BROADCAST(u) */`”强制做broadcastjoin。如果有，则需要去掉该标识。
2. 设置`spark.sql.autoBroadcastJoinThreshold=-1`，具体操作如下：
 - a. 登录DLI管理控制台，单击“作业管理 > SQL作业”，在对应报错作业的“操作”列，单击“编辑”进入到SQL编辑器页面。
 - b. 单击“设置”，在参数设置中选择“`spark.sql.autoBroadcastJoinThreshold`”参数，其值设置为“`-1`”。
 - c. 重新单击“执行”，运行该作业，观察作业运行结果。

2.4.12 join 表时没有添加 on 条件，造成笛卡尔积查询，导致队列资源爆满，作业运行失败

问题现象

运行的SQL语句中存在join表，但是join没有添加on条件，多表关联造成笛卡尔积查询，最终导致队列资源占满，该队列上的作业运行失败。

例如，如下问题SQL语句，存在三个表的left join，并且没有指定on条件，造成笛卡尔积查询。

```
select
    case
        when to_char(from_unixtime(fs.special_start_time), 'yyyy-mm-dd') < '2018-10-12' and row_number()
over(partition by fg.goods_no order by fs.special_start_time asc) = 1 then 1
            when to_char(from_unixtime(fs.special_start_time), 'yyyy-mm-dd') >= '2018-10-12' and fge.is_new = 1
then 1
        else 0 end as is_new
from testdb.table1 fg
left join testdb.table2 fs
left join testdb.table3 fge
where to_char(from_unixtime(fs.special_start_time), 'yyyymmdd') = substr('20220601',1,8)
```

解决措施

在使用join进行多表关联查询时，不管表数据量大小，join时都需要指定on条件来减少多表关联的数据量，从而减轻队列的负荷，提升查询效率。

例如，问题现象中的问题语句可以根据业务场景，在join时通过指定on条件来进行优化，这样会极大减少关联查询的结果集，提升查询效率。

```
select
    case
        when to_char(from_unixtime(fs.special_start_time), 'yyyy-mm-dd') < '2018-10-12' and row_number()
over(partition by fg.goods_no order by fs.special_start_time asc) = 1 then 1
            when to_char(from_unixtime(fs.special_start_time), 'yyyy-mm-dd') >= '2018-10-12' and fge.is_new = 1
then 1
        else 0 end as is_new
from testdb.table1 fg
left join testdb.table2 fs on fg.col1 = fs.col2
left join testdb.table3 fge on fg.col3 = fge.col4
where to_char(from_unixtime(fs.special_start_time), 'yyyymmdd') = substr('20220601',1,8)
```

2.4.13 手动在 OBS 表的分区目录下添加了数据，但是无法查询到该部分数据

问题现象

手动在OBS表的分区目录下上传了分区数据，但是在SQL编辑器中查询该表新增的分区数据时却查询不到。

解决方案

手动添加分区数据后，需要刷新OBS表的元数据信息。具体操作如下：
MSCK REPAIR TABLE *table_name*;

执行完上述命令后，再执行对应OBS分区表的数据查询即可。

2.4.14 为什么 insert overwrite 覆盖分区表数据的时候，覆盖了全量数据？

如果需要动态覆盖DataSource表指定分区数据，需要先配置参数：
`dli.sql.dynamicPartitionOverwrite.enabled=true`，再通过“`insert overwrite`”语句实现，“`dli.sql.dynamicPartitionOverwrite.enabled`”默认值为“`false`”。
详细说明请参考[插入数据](#)。

2.4.15 为什么 SQL 作业一直处于“提交中”？

SQL作业一直在提交中，有以下几种可能：

- 刚购买DLI队列后，第一次进行SQL作业的提交。需要等待5~10分钟，待后台拉起集群后，即可提交成功。
- 若刚刚对队列进行网段修改，立即进行SQL作业的提交。需要等待5~10分钟，待后台重建集群后，即可提交成功。
- 按需队列，队列已空闲状态（超过1个小时），则后台资源已经释放。此时进行SQL作业的提交。需要等待5~10分钟，待后台重新拉起集群后，即可提交成功。

2.4.16 跨源连接 RDS 表中 `create_date` 字段类型是 `datetime`，为什么 dli 中查出来的是时间戳呢？

Spark中没有`datetime`数据类型，其使用的是`TIMESTAMP`类型。

您可以通过函数进行转换。

例如：

```
select cast(create_date as string), * from table where create_date>'2021-12-01 00:00:00';
```

`TIMESTAMP`类型详细可参考[TIMESTAMP数据类型](#)。

2.4.17 SQL 作业执行完成后，修改表名导致 `datasize` 修改失败怎么办？

如果执行SQL后立即修改了表名，可能会导致表的数据大小结果不正确。

如需修改表名，建议在SQL作业执行完成后，间隔5分钟再修改表名。

2.4.18 从 DLI 导入数据到 OBS，为什么数据量出现差异？

问题现象

使用DLI插入数据到OBS临时表文件，数据量有差异。

根因分析

出现该问题可能原因如下：

- 作业执行过程中，读取数据量错误。
- 验证数据量的方式不正确。

通常在执行插入数据操作后，如需确认插入数据量是否正确，建议通过查询语句进行查询。

如果OBS对存入的文件数量有要求，可以在插入语句后加入“DISTRIBUTE BY number”。例如，在插入语句后添加“DISTRIBUTE BY 1”，可以将多个task生成的多个文件汇总为一个文件。

操作步骤

步骤1 在管理控制台检查对应SQL作业详情中的“结果条数”是否正确。检查发现读取的数据量是正确的。

图 2-18 检查读取的数据量

作业ID	作业类型	执行语句	配置项
850f462d-a398-43fd-9a17-f0a180f974e5	QUERY	select ename from emp order by salary desc	spark.sql.shuffle.partitions:1
1.95 KB	已扫描数据		
数据已缓存	结果状态	(查看结果)	

步骤2 确认客户验证数据量的方式是否正确。客户验证的方式如下：

1. 通过OBS下载数据文件。
2. 通过文本编辑器打开数据文件，发现数据量缺失。

根据该验证方式，初步定位是因为文件数据量较大，文本编辑器无法全部读取。

通过执行查询语句，查询OBS数据进一步进行确认，查询结果确认数据量正确。

因此，该问题为验证方式不正确造成。

----结束

参考信息

插入数据的SQL语法，请参考《[数据湖探索Spark SQL语法参考](#)》。

3 Spark 作业相关问题

3.1 使用咨询

3.1.1 Spark 作业使用咨询

DLI Spark 作业是否支持定时周期任务作业

DLI Spark不支持作业调度，用户可以通过其他服务，例如数据湖管理治理中心DataArts Studio服务进行调度，或者通过API/SDK等方式对作业进行自定义调度。

使用DataArts Studio服务进行作业开发请参考《[数据治理中心用户指南](#)》。

Spark SQL 语法创建表时是否支持定义主键

Spark SQL语法不支持定义主键。

DLI Spark jar 作业是否能访问 DWS 跨源表？

可以访问。

详细操作请参考[访问DWS](#)和[访问SQL库表](#)。

如何查看 Spark 内置依赖包的版本？

DLI内置依赖包是平台默认提供的依赖包，用户打包Spark或Flink jar作业jar包时，不需要额外上传这些依赖包，以免与平台内置依赖包冲突。

查看Spark内置依赖包的版本请参考[内置依赖包](#)。

资源包管理中的包是否能够下载？

资源包仅提供托管服务，不提供下载功能。

如何使用 API 通过公网访问 DLI？

DLI域名公网访问请使用域名访问：dli.{regionid}.myhuaweicloud.com

- 数据湖探索的终端节点请参考[终端节点](#)。
- 数据湖探索API请参考：[数据湖探索API](#)。

DLI 自定义的 Spark3.1.1 镜像，需要把第三方依赖 jar 放到哪个路径下呢？

DLI自定义的Spark3.1.1镜像，建议将第三方依赖jar存放/opt/spark/jars目录。

3.2 作业开发

3.2.1 Spark 如何将数据写入到 DLI 表中

使用Spark将数据写入到DLI表中，主要设置如下参数：

- fs.obs.access.key
- fs.obs.secret.key
- fs.obs.impl
- fs.obs.endpoint

示例如下：

```
import logging
from operator import add
from pyspark import SparkContext

logging.basicConfig(format='%(message)s', level=logging.INFO)

#import local file
test_file_name = "D://test-data_1.txt"
out_file_name = "D://test-data_result_1"

sc = SparkContext("local","wordcount app")
sc._jsc.hadoopConfiguration().set("fs.obs.access.key", "myak")
sc._jsc.hadoopConfiguration().set("fs.obs.secret.key", "mysk")
sc._jsc.hadoopConfiguration().set("fs.obs.impl", "org.apache.hadoop.fs.obs.OBSFileSystem")
sc._jsc.hadoopConfiguration().set("fs.obs.endpoint", "myendpoint")

# red: text_file rdd object
text_file = sc.textFile(test_file_name)

# counts
counts = text_file.flatMap(lambda line: line.split(" ")).map(lambda word: (word, 1)).reduceByKey(lambda a,
b: a + b)
# write
counts.saveAsTextFile(out_file_name)
```

3.2.2 通用队列操作 OBS 表如何设置 AK/SK

建议使用临时AK/SK，获取方式可参见[统一身份认证服务_获取临时AK/SK](#)。

说明

认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全。

- 获取结果为AK和SK时，设置如下：
 - 代码创建SparkContext

```
val sc: SparkContext = new SparkContext()
sc.hadoopConfiguration.set("fs.obs.access.key", ak)
sc.hadoopConfiguration.set("fs.obs.secret.key", sk)
```

- 代码创建SparkSession

```
val sparkSession: SparkSession = SparkSession
  .builder()
  .config("spark.hadoop.fs.obs.access.key", ak)
  .config("spark.hadoop.fs.obs.secret.key", sk)
  .enableHiveSupport()
  .getOrCreate()
```

- 获取结果为ak、sk和securitytoken时，鉴权时，临时AK/SK和securitytoken必须同时使用，设置如下：

- 代码创建SparkContext

```
val sc: SparkContext = new SparkContext()
sc.hadoopConfiguration.set("fs.obs.access.key", ak)
sc.hadoopConfiguration.set("fs.obs.secret.key", sk)
sc.hadoopConfiguration.set("fs.obs.session.token", sts)
```

- 代码创建SparkSession

```
val sparkSession: SparkSession = SparkSession
  .builder()
  .config("spark.hadoop.fs.obs.access.key", ak)
  .config("spark.hadoop.fs.obs.secret.key", sk)
  .config("spark.hadoop.fs.obs.session.token", sts)
  .enableHiveSupport()
  .getOrCreate()
```

3.2.3 如何查看 DLI Spark 作业的实际资源使用情况

查看 Spark 作业原始资源配置

登录DLI 控制台，单击左侧“作业管理”>“Spark作业”，在作业列表中找到需要查看的Spark作业，单击“作业ID”前的 ，即可查看对应Spark作业的原始资源配置参数。

说明

在创建Spark作业时，配置了“高级配置”中的参数，此处才会显示对应的内容。创建Spark作业请参考《[创建Spark作业](#)》。

图 3-1 查看 Spark 作业原始资源配置

The screenshot shows the DLI Control Panel with the "Jobs" tab selected. A specific job entry is highlighted, showing its configuration details. The configuration JSON is displayed in a large text area:

```
{
  "sc_type": "A",
  "cluster_name": "citest",
  "jars": [],
  "pyFiles": [],
  "rdd": [],
  "modelFiles": [],
  "resources": [],
  "modules": [],
  "groups": [],
  "numExecutors": 3,
  "archives": [],
  "conf": {},
  "file": "spark-examples_2.11-2.1.0.luxor.jar",
  "args": [],
  "className": "org.apache.spark.examples.SparkPi",
  "autoRecovery": false,
  "minRecoveryDelayTime": 10000,
  "maxRetryTimes": 20
}
```

查看 Spark 作业实时运行资源

查看Spark作业实时运行资源，即查看有多少CU正在运行。

1. 登录DLI控制台，单击左侧“作业管理”>“Spark作业”，在作业列表中找到需要查看的Spark作业，单击“操作”列中的“SparkUI”。
2. 在SparkUI页面可查看Spark作业实时运行资源。

图 3-2 SparkUI

Active Jobs (1)					
Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
5	runJob at FileFormatWriter.scala:266 runJob at FileFormatWriter.scala:266	2021/04/23 19:17:31	1.6 min	1/6	722/3019 (15 running)
Completed Jobs (5)					
4 (2bfc3728-346e-4ccb-9299-c84b33812071)	broadcast exchange (runId 2bfc3728-346e-4ccb-9299-c84b33812071)	2021/04/23 19:17:19	11 s	1/1	1 Pages. Jump to 1 Show 100

3. 在SparkUI页面还可以查看Spark作业原始资源配置（只对新集群开放）。
在SparkUI页面，单击“Environment”，可以查看Driver信息和Executor信息。

图 3-3 Driver 信息

spark.driver.core	2
spark.driver.extraJavaOptions	-XX:CICo XX:+ExitT XX:+Use Dlog4j.co Dcarbon. Dscc.con
spark.driver.extraLibraryPath	Bigdata/c
spark.driver.host	spark-4c
spark.driver.memory	7G
spark.driver.port	7078
spark.driver.userClassPathFirst	false
spark.dynamicAllocation.cachedExecutorIdleTimeout	600s

图 3-4 Executor 信息

spark.executor.core	2
spark.executor.extraClassPath	hadoop/c
spark.executor.extraJavaOptions	-XX:CICo XX:+ExitT XX:+Use Dlog4j.co Dcarbon.j Dscc.con
spark.executor.extraLibraryPath	Bigdata/c
spark.executor.heartbeatInterval	10000ms
spark.executor.id	driver
spark.executor.instances	7
spark.executor.memory	8G
spark.executor.memoryOverhead	2G
spark.executor.periodicGC.interval	30min

3.2.4 将 Spark 作业结果存储在 MySQL 数据库中，缺少 pymysql 模块，如何使用 python 脚本访问 MySQL 数据库？

1. 缺少pymysql模块，可以查看是否有对应的egg包，如果没有，在“程序包管理”页面上传pyFile。具体步骤参考如下：
 - a. 将egg包上传到指定的OBS桶路径下。

- b. 登录DLI管理控制台，单击“数据管理 > 程序包管理”。
 - c. 在“程序包管理”页面，单击右上角“创建”可创建程序包。
 - d. 在“创建程序包”对话框，配置如下参数：
 - 包类型：PyFile。
 - OBS路径：选择[1.aegg](#)包所在的OBS路径。
 - 分组设置和分组名称根据情况选择。
 - e. 单击“确定”完成程序包上传。
 - f. 在报错的Spark作业编辑页面，“依赖python文件”处选择已上传的egg程序包，重新运行Spark作业。
2. pyspark作业对接MySQL，需要创建跨源链接，打通DLI和RDS之间的网络。
通过管理控制台创建跨源连接请参考《[数据湖探索用户指南](#)》。
通过API创建跨源连接请参考《[数据湖探索API参考](#)》。

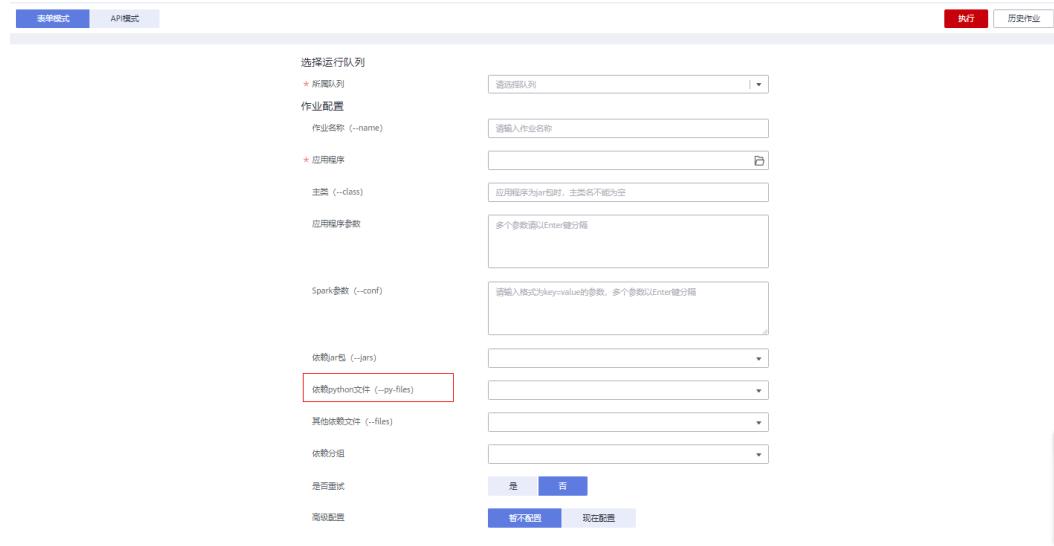
3.2.5 如何在 DLI 中运行复杂 PySpark 程序？

数据湖探索（DLI）服务对于PySpark是原生支持的。

对于数据分析来说Python是很自然的选择，而在大数据分析中PySpark无疑是不二选择。对于JVM语言系的程序，通常会把程序打成Jar包并依赖其他一些第三方的Jar，同样的Python程序也有依赖一些第三方库，尤其是基于PySpark的融合机器学习相关的大数据分析程序。传统上，通常是直接基于pip把Python库安装到执行机器上，对于DLI这样的Serverless化服务用户无需也感知不到底层的计算资源，那如何来保证用户可以更好的运行他的程序呢？

DLI服务在其计算资源中已经内置了一些常用的机器学习的算法库（具体可以参考“[数据湖探索 DLI > 用户指南> 数据管理> 程序包管理> 内置依赖包](#)”），这些常用算法库满足了大部分用户的使用场景。对于用户的PySpark程序依赖了内置算法库未提供的程序库该如何呢？其实PySpark本身就已经考虑到这一点了，那就是基于PyFiles来指定依赖，在DLI Spark作业页面中可以直接选取存放在OBS上的Python第三方程序库（支持zip、egg等）。

图 3-5 Spark 作业编辑页面



对于依赖的这个Python第三方库的压缩包有一定的结构要求，例如，PySpark程序依赖了模块moduleA（import moduleA），那么其压缩包要求满足如下结构：

图 3-6 压缩包结构要求

```
xxx.zip
  moduleA
    a.py
    b.py
  ...

```

即在压缩包内有一层以模块名命名的文件夹，然后才是对应类的Python文件，通常下载下来的Python库可能不满足这个要求，因此需要重新压缩。同时对压缩包的名称没有要求，所以建议可以把多个模块的包都压缩到一个压缩包里。至此，已经可以完整的运行起来一个大型、复杂的PySpark程序了。

3.2.6 Spark 作业访问 MySQL 数据库的方案

通过DLI Spark作业访问MySQL数据库中的数据有如下两种方案：

- 方案1：在DLI中创建队列，队列计费模式为包年/包月队列或按需专属，创建增强型跨源连接，再通过跨源表读取MySQL数据库中的数据，该方案需要用户自行编写java代码或scala代码。
- 方案2：先使用云数据迁移服务CDM将MySQL数据库中的数据导入OBS桶中，再通过Spark作业读取OBS桶中的数据，如果用户已有CDM集群，该方案比方案1简单，且不会对现有数据库造成压力。

3.2.7 如何通过 JDBC 设置 spark.sql.shuffle.partitions 参数提高并行度

操作场景

Spark作业在执行shuffle类语句，包括group by、join等场景时，常常会出现数据倾斜的问题，导致作业任务执行缓慢。

该问题可以通过设置spark.sql.shuffle.partitions提高shuffle read task的并行度来进行解决。

设置 spark.sql.shuffle.partitions 参数提高并行度

用户可在JDBC中通过set方式设置dli.sql.shuffle.partitions参数。具体方法如下：

```
Statement st = conn.createStatement()  
st.execute("set spark.sql.shuffle.partitions=20")
```

3.2.8 Spark jar 如何读取上传文件

Spark可以使用SparkFiles读取 --file中提交上来的文件的本地路径，即：
SparkFiles.get("上传的文件名")。

□ 说明

- Driver中的文件路径与Executor中获取的路径位置是不一致的，所以不能将Driver中获取到的路径作为参数传给Executor去执行。
- Executor获取文件路径的时候，仍然需要使用SparkFiles.get(“filename”)的方式获取。
- SparkFiles.get()方法需要spark初始化以后才能调用。

图 3-7 添加其他依赖文件



代码段如下所示

```
package main.java  
  
import org.apache.spark.SparkFiles  
import org.apache.spark.sql.SparkSession  
  
import scala.io.Source
```

```
object DliTest {  
    def main(args:Array[String]): Unit = {  
        val spark = SparkSession.builder  
            .appName("SparkTest")  
            .getOrCreate()  
  
        // driver 获取上传文件  
        println(SparkFiles.get("test"))  
  
        spark.sparkContext.parallelize(Array(1,2,3,4))  
            // Executor 获取上传文件  
            .map(_ => println(SparkFiles.get("test")))  
            .map(_ => println(Source.fromFile(SparkFiles.get("test")).mkString)).collect()  
    }  
}
```

3.3 作业运维报错

3.3.1 运行 Spark 作业报 java.lang.AbstractMethodError

Spark 2.3对内部接口Logging做了行为变更，如果用户代码里直接继承了该Logging，且编译时使用的是低版本的Spark，那么应用程序在Spark 2.3的环境中运行将会报java.lang.AbstractMethodError。

解决措施有如下两种方案：

- 基于Spark 2.3重新编译应用
- 使用sl4j+log4j来实现日志功能，而不是直接继承Spark内部接口Logging。具体如下：

```
<dependency>  
    <groupId>org.slf4j</groupId>  
    <artifactId>slf4j-api</artifactId>  
    <version>1.7.16</version>  
</dependency>  
<dependency>  
    <groupId>org.slf4j</groupId>  
    <artifactId>slf4j-log4j12</artifactId>  
    <version>1.7.16</version>  
</dependency>  
<dependency>  
    <groupId>log4j</groupId>  
    <artifactId>log4j</artifactId>  
    <version>1.2.17</version>  
</dependency>  
  
private val logger = LoggerFactory.getLogger(this.getClass)  
logger.info("print log with sl4j+log4j")
```

3.3.2 Spark 作业访问 OBS 数据时报 ResponseCode: 403 和 ResponseStatus: Forbidden 错误

问题现象

Spark程序访问OBS数据时上报如下错误。

```
Caused by: com.obs.services.exception.ObsException: Error message:Request Error.OBS servcie Error  
Message. -- ResponseCode: 403, ResponseStatus: Forbidden
```

解决方案

Spark程序访问OBS数据时，需要通过配置AK、SK的访问进行访问。具体访问方式可以参考：[通用队列操作OBS表如何设置AK/SK](#)。

3.3.3 有访问 OBS 对应的桶的权限，但是 Spark 作业访问时报错 verifyBucketExists on XXXX: status [403]

请排查“全局配置 > 作业配置”对应OBS桶是否被设置为了DLI日志桶，日志桶不能用于DLI的其他业务功能中。

3.3.4 Spark 作业运行大批量数据时上报作业运行超时异常错误

Spark作业程序访问大批量数据，如访问DWS大批量数据库数据时，建议设置并发数，启动多任务的方式运行，避免作业运行超时。

具体并发设置可以参考[对接DWS样例代码](#)中的partitionColumn和numPartitions相关字段和案例描述。

3.3.5 使用 Spark 作业访问 sftp 中的文件，作业运行失败，日志显示访问目录异常

Spark作业不支持访问sftp，建议将文件数据上传到OBS，再通过Spark作业进行读取和分析。

Spark读取OBS文件数据，详见[使用Spark Jar作业读取和查询OBS数据](#)。

3.3.6 执行作业的用户数据库和表权限不足导致作业运行失败

问题现象

Spark作业运行报数据库权限不足，报错信息如下：

```
org.apache.spark.sql.AnalysisException: org.apache.hadoop.hive.ql.metadata.HiveException:  
MetaException(message:Permission denied for resource: databases.xxx,action:SPARK_APP_ACCESS_META)
```

解决方案

需要给执行作业的用户赋数据库的操作权限，具体操作参考如下：

1. 在DLI管理控制台左侧，单击“数据管理”>“库表管理”。
2. 单击所选数据库“操作”栏中的“权限管理”，将显示该数据库对应的权限信息。
3. 在数据库权限管理页面右上角单击“授权”。
4. 在“授权”弹出框中，选择“用户授权”或“项目授权”，填写需要授权的用户名或选择需要授权的项目，选择相应的权限。
5. 单击“确定”，完成授权。

3.3.7 为什么 Spark3.x 的作业日志中打印找不到 global_temp 数据库

问题描述

Spark3.x的作业日志中提示找不到global_temp数据库。

根因分析

global_temp数据库是Spark3.x默认内置的数据库，是Spark的全局临时视图。

通常在Spark作业执行注册viewManager时，会校验该数据库在metastore是否存在，如果该数据库存在则会导致Spark作业执行失败。

因此当Spark3.x的作业日志中如果出现一条访问catalog查询该数据库，并提示数据库不存在是为了确保Spark作业正常运行，无需执行其他操作。

3.3.8 在使用 Spark2.3.x 访问元数据时，DataSource 语法创建 avro 类型的 OBS 表创建失败

问题描述

使用Spark访问元数据时，DataSource语法创建avro类型的OBS表创建失败。

图 3-8 avro 类型的 OBS 创建失败

```
150
151
152
153 org.apache.spark.sql.AnalysisException: Failed to find data source: avro. Please find an Avro package at
154
155
156
157
158
```

根因分析

当前Spark2.3.x不支持创建avro类型的OBS表，Spark2.4.x及以上的版本支持avro类型的OBS表。

解决方案

在使用DataSource语法创建avro类型的OBS表时，请选择Spark2.4.x及以上版本进行创建。

3.4 运维指导

3.4.1 添加 Python 包后，找不到指定的 Python 环境

添加Python3包后，找不到指定的Python环境。

可以通过在conf文件中，设置
spark.yarn.appMasterEnv.PYSPARK_PYTHON=python3，指定计算集群环境为
Python3环境。

目前，新建集群环境均已默认为Python3环境。

3.4.2 为什么 Spark jar 作业一直处于“提交中”？

Spark jar 作业一直处于“提交中”可能是队列剩余的CU量不足导致作业无法提交。

查看队列的剩余步骤如下：

1. 查看队列CU使用量。

点击“云监控服务 > 云服务监控 > 数据探索湖 > 队列监控 > 队列CU使用量”。

2. 计算剩余CU量。

队列剩余CU量=队列CU量 - 队列CU使用量。

当队列剩余CU量小于用户提交的CU量，则需要等待资源，才能提交成功。

4 产品咨询相关问题

4.1 使用咨询

4.1.1 什么是 DLI

数据湖探索（Data Lake Insight，简称DLI）是完全兼容Apache Spark、Apache Flink、trino生态，提供一站式的流处理、批处理、交互式分析的Serverless融合处理分析服务。用户不需要管理任何服务器，即开即用。支持标准SQL/Spark SQL/Flink SQL，支持多种接入方式，并兼容主流数据格式。数据无需复杂的抽取、转换、加载，使用SQL或程序就可以对云上CloudTable、RDS、DWS、CSS、OBS、ECS自建数据库以及线下数据库的异构数据进行探索。

4.1.2 DLI 适用哪些场景

DLI服务适用于海量日志分析、异构数据源联邦分析、大数据ETL处理。

海量日志分析

- 游戏运营数据分析

游戏公司不同部门日常通过游戏数据分析平台，分析每日新增日志获取所需指标，通过数据来辅助决策。例如：运营部门通过平台获取新增玩家、活跃玩家、留存率、流失率、付费率等，了解游戏当前状态及后续响应活动措施；投放部门通过平台获取新增玩家、活跃玩家的渠道来源，来决定下一周期重点投放哪些平台。

- 优势

- 高效的Spark编程模型：使用Spark Streaming直接从DIS中获取数据，进行数据清理等预处理操作。只需编写处理逻辑，无需关心多线程模型。
- 简单易用：直接使用标准SQL编写指标分析逻辑，无需关注背后复杂的分布式计算平台。
- 按需计费：日志分析按实效性要求按周期进行调度，每次调度之间存在大量空闲期。DLI按需计费只在使用期间收费，成本较独占队列降低50%以上。

- 建议搭配以下服务使用

OBS, DIS, DWS, RDS

异构数据源联邦分析

- 车企数字化服务转型

面临市场新的竞争压力及出行服务不断变革，车企通过构建车联云平台和车机OS，将互联网应用与用车场景打通，完成车企数字化服务转型，从而为车主提供更好的智联出行体验，增加车企竞争力，促进销量增长。例如：通过对车辆日常指标数据（电池、发动机，轮胎胎压、安全气囊等健康状态）的采集和分析，及时将维保建议回馈给车主。

- 优势

- 多源数据分析免搬迁：关系型数据库RDS中存放车辆和车主基本信息，表格存储CloudTable中存放实时的车辆位置和健康状态信息，数据仓库DWS中存放周期性统计的指标。通过DLI无需数据搬迁，对多数据源进行联邦分析。
- 数据分级存储：车企需要保留全量历史数据支撑审计类等业务，低频进行访问。温冷数据存放在低成本的对象存储服务OBS上，高频访问的热数据存放在数据引擎（CloudTable和DWS）中，降低整体存储成本。
- 告警快速敏捷触发服务器弹性伸缩：对CPU、内存、硬盘空间和带宽无特殊要求。

- 建议搭配以下服务使用

DIS、CDM、OBS、DWS、RDS、CloudTable

大数据 ETL 处理

- 运营商大数据分析

运营商数据体量在PB~EB级，其数据种类多，有结构化的基站信息数据，非结构化的消息通信数据，同时对数据的时效性有很高的要求，DLI服务提供批处理、流处理等多模引擎，打破数据孤岛进行统一的数据分析。

- 优势

- 大数据ETL：具备TB~EB级运营商数据治理能力，能快速将海量运营商数据做ETL处理，为分布式批处理计算提供分布式数据集。
- 高吞吐低时延：采用Apache Flink的Dataflow模型，高性能计算资源，从用户自建的Kafka、MRS-Kafka、DMS-Kafka消费数据，单CU每秒吞吐1千~2万条消息。
- 细粒度权限管理：P公司内部有N个子部门，子部门之间需要对数据进行共享和隔离。DLI支持计算资源按租户隔离，保障作业SLA；支持数据权限控制到表/列，帮助企业实现部门间数据共享和权限管理。

- 建议搭配以下服务使用

OBS、DIS、DataArts Studio

地理大数据分析

- 地理大数据分析

地理大数据具有大数据的相关特征，数据体量巨大，例如全球卫星遥感影像数据量达到PB级；数据种类多，有结构化的遥感影像栅格数据、矢量数据，非结构化的空间位置数据、三维建模数据；在大体量的地理大数据中，通过高效的挖掘工具或者挖掘方法实现价值提炼，是用户非常关注的话题。

- 优势

- 提供地理专业算子：支持全栈Spark能力，具备丰富的Spark空间数据分析算法算子，全面支持结构化的遥感影像数据、非结构化的三维建模、激光点云等巨量数据的离线批处理，支持带有位置属性的动态流数据实时计算处理。

- CEP SQL：提供地理位置分析函数对地理空间数据进行实时分析，用户仅需编写SQL便可实现例如偏航检测，电子围栏等地理分析场景。
- 大数据治理能力：能快速将海量遥感影像数据接入上云，快速完成影像数据切片处理，为分布式批处理计算提供弹性分布式数据集。
- 建议搭配以下服务使用
DIS、CDM、DES、OBS、RDS、CloudTable

4.1.3 DLI 支持哪些数据格式

DLI支持如下数据格式：

- Parquet
- CSV
- ORC
- Json
- Avro

4.1.4 DLI Flink 与 MRS Flink 有什么区别？

DLI Flink是天然的云原生基础架构。在内核引擎上DLI Flink进行了多处核心功能的优化，并且提供了企业级的一站式开发平台，自带开发和运维功能，免除自建集群运维的麻烦；在connector方面除了支持开源connector之外，还可以对接云上Mysql、GaussDB、MRS HBase、DMS、DWS、OBS等，开箱即用；在资源方面，产品可以自适应业务的流量，智能对资源进行弹性伸缩，保障业务稳定性，不需要人工进行额外调试。

DLI Flink与MRS Flink的功能对比如表1所示。

表 4-1 DLI Flink 与 MRS Flink 功能对比

类型	特点	DLI Flink	MRS Flink
特色能力	产品模式	全托管（无需人力运维集群）	半托管（需要人力运维集群）
	弹性扩缩容	<ul style="list-style-type: none">• 支持集群容器化部署。• 用户可以根据业务负载进行弹性扩缩容，能够基于作业的负载动态调整作业使用资源大小。• 支持基于作业优先级动态调整作业的使用资源。	仅支持YARN集群。

类型	特点	DLI Flink	MRS Flink
	上下游数据连接	<ul style="list-style-type: none">除了开源connector之外，还提供开箱即用的connector，包括数据库（RDS、GaussDB）、消息队列（DMS）、数据仓库（DWS）、对象存储（OBS）相比开源connector有较多易用性和稳定性提升。	仅提供开源connector。
开发与运维	监控、告警	<ul style="list-style-type: none">支持对接华为云CES监控平台，支持对接华为云SMN告警系统，用户可通过邮件、短信、电话、第三方办公工具（webhook模式）支持对接企业内部统一监控告警系统（prometheus）。支持Flink作业速率、输入输出数据量、作业算子反压值、算子延迟、作业cpu和内存使用率查看。	仅支持Flink UI
	多版本支持	支持不同作业使用不同Flink版本	单Flink集群仅支持单版本下的作业开发
	易用性	<p>即开即用，Serverless架构，跨AZ容灾能力。</p> <ul style="list-style-type: none">用户仅编写SQL代码，无需编译，只需关心业务代码。支持用户通过编写SQL连接各个数据源，如RDS、DWS、Kafka、Elasticsearch等数据源；用户无需登录维护集群，在控制台上完成一键提交，无需接触集群。支持Flink SQL作业快速开启checkpoint。支持Flink作业日志转储保留，便于作业分析。	<p>需要一定的技术能力完成代码编译、集群搭建、配置、运维。</p> <ul style="list-style-type: none">用户需要自己编写完整代码并进行编译。用户需要登录集群使用命令进行提交，且需要维护集群。用户需要在代码里写入checkpoint才能开启。
	作业模板	内置多个常见Flink SQL通用场景模板，帮助您快速了解和构建作业代码	暂无
企业安全	访问控制	与华为云IAM权限打通，支持多角色的访问控制	暂无

类型	特点	DLI Flink	MRS Flink
	空间隔离	支持租户级和项目级的资源和代码隔离，满足多团队协作需求	暂无

4.1.5 DLI 中的 Spark 组件与 MRS 中的 Spark 组件有什么区别？

DLI服务的Spark组件是全托管式服务，用户对Spark组件不感知，仅仅可以使用该服务，且接口为封装式接口。具体请参考《[数据湖探索用户指南](#)》。

MRS服务Spark组件的是建立在客户的购买MRS服务所分配的虚机上，用户可以根据实际需求调整及优化Spark服务，支持各种接口调用。具体请参考《[MapReduce服务开发指南](#)》。

4.1.6 DLI 的数据可存储在哪些地方

DLI服务的数据可存储在如下地方：

- OBS: SQL作业，Spark作业，Flink作业使用的数据均可以存储在OBS服务中，降低存储成本。
- DLI: DLI内部使用的是列存的Parquet格式，即数据以Parquet格式存储。存储成本较高。
- 跨源作业可将数据存储在对应的服务器中，目前支持CloudTable，CSS，DCS，DDS，DWS，MRS，RDS等。

4.1.7 DLI 表与 OBS 表的区别

- DLI表表示数据存储在本服务内部，用户不感知数据存储路径。
- OBS表表示数据存储在用户自己账户的OBS桶中，源数据文件由用户自己管理。
- DLI表相较于OBS表提供了更多权限控制和缓存加速的功能，性能相较于外表性能更好，但是会收取存储费用。

4.1.8 不上传数据到 OBS，如何使用 DLI

当前DLI只支持对云上数据分析，数据还是要传到云上；但是用户可以折中处理：将待分析的数据脱敏后上传到OBS临时存放，分析完之后将结果导出使用，同时将OBS临时存放数据删除；该方法主要适用于定期（如每天）对增量数据一次性分析统计，然后利用分析结果支撑业务的场景；对频繁需要分析的数据，还是建议将数据上传到云上。

4.1.9 DLI 是否支持导入其他租户共享 OBS 桶的数据？

在DLI中，同一个租户下子账户共享OBS桶中的数据是支持导入的，但是租户级别共享OBS桶中的数据无法导入。

4.1.10 区域和可用区

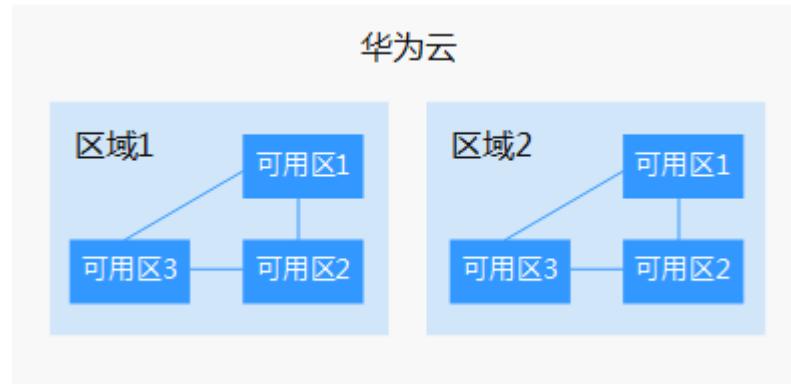
什么是区域、可用区？

我们用区域和可用区来描述数据中心的位置，您可以在特定的区域、可用区创建资源。

- **区域 (Region)**：从地理位置和网络时延维度划分，同一个Region内共享弹性计算、块存储、对象存储、VPC网络、弹性公网IP、镜像等公共服务。Region分为通用Region和专属Region，通用Region指面向公共租户提供通用云服务的Region；专属Region指只承载同一类业务或只面向特定租户提供业务服务的专用Region。
- **可用区 (AZ, Availability Zone)**：一个AZ是一个或多个物理数据中心的集合，有独立的风火水电，AZ内逻辑上再将计算、网络、存储等资源划分成多个集群。一个Region中的多个AZ间通过高速光纤相连，以满足用户跨AZ构建高可用性系统的需求。

图4-1阐明了区域和可用区之间的关系。

图 4-1 区域和可用区



目前，华为云已在全球多个地域开放云服务，您可以根据需求选择适合自己的区域和可用区。更多信息请参见[华为云全球站点](#)。

如何选择区域？

选择区域时，您需要考虑以下几个因素：

- **地理位置**

一般情况下，建议就近选择靠近您或者您的目标用户的区域，这样可以减少网络时延，提高访问速度。不过，在基础设施、BGP网络品质、资源的操作与配置等方面，中国大陆各个区域间区别不大，如果您或者您的目标用户在中国大陆，可以不用考虑不同区域造成的网络时延问题。

香港、曼谷等其他地区和国家提供国际带宽，主要面向非中国大陆地区的用户。如果您或者您的目标用户在中国大陆，使用这些区域会有较长的访问时延，不建议使用。

- 在除中国大陆以外的亚太地区有业务的用户，可以选择“中国-香港”、“亚太-曼谷”或“亚太-新加坡”区域。
- 在非洲地区有业务的用户，可以选择“南非-约翰内斯堡”区域。

- 在欧洲地区有业务的用户，可以选择“欧洲-巴黎”区域。
- 资源的价格
不同区域的资源价格可能有差异，请参见[华为云服务价格详情](#)。

如何选择可用区？

是否将资源放在同一可用区内，主要取决于您对容灾能力和网络时延的要求。

- 如果您的应用需要较高的容灾能力，建议您将资源部署在同一区域的不同可用区内。
- 如果您的应用要求实例之间的网络延时较低，则建议您将资源创建在同一可用区内。

区域和终端节点

当您通过API使用资源时，您必须指定其区域终端节点。有关区域和终端节点的更多信息，请参阅[地区和终端节点](#)。

4.1.11 全局变量的使用中，一个子帐号是否可以使用其他子帐号创建的全局变量

全局变量可用于简化复杂参数。例如，可替换长难复杂变量，提升SQL语句可读性。

全局变量的使用具有以下约束限制：

- 存量敏感变量只有创建用户才能使用，其余普通全局变量同帐号同项目下的用户共用。
- 如果同帐号同项目下存在多个相同名称的全局变量时，需要将多余相同名称的全局变量删除，保证同帐号同项目下唯一，此时具备该全局变量修改权限的用户均可以修改对应的变量值。
- 如果同帐号同项目下存在多个相同名称的全局变量，优先删除用户自建的。如果仅存在唯一名称的全局变量，则具备删除权限即的用户均可删除该全局变量。

4.2 作业管理

4.2.1 如何获取 DLI 作业样例（Demo）

为了方便用户更好地使用DLI，DLI服务提供了供作业开发的Demo样例，您可以通过[DLI样例代码](#)获取。

该样例代码的目录内容介绍如下：

- dli-flink-demo：开发Flink作业时的样例代码参考。例如，样例代码实现读取Kafka源表数据写入到HDFS、DWS、Hive等结果表中的功能。
- dli-spark-demo：开发Spark作业时的样例代码参考。具体如下：
 - “dli-spark-demo-obs” 读取和写入OBS数据。
 - “dli-spark-demo-redis” 读取和写入Redis数据。
- dli-pyspark-demo：使用Python语言开发Spark作业。例如，样例包中的样例代码实现创建Redis表的功能。

4.2.2 使用 DLI 运行几万个作业，如何管理？

用户需要使用DLI运行几万个作业，建议如下：

- 将几万个作业根据不同的类型分组，不同类型的作业通过不同的队列运行。关于如何购买队列，请参考《[创建队列](#)》。
- 或者创建IAM子用户，将不同类型的作业通过不同的用户执行。具体请参考《[数据湖探索用户指南](#)》。

4.2.3 DLI 上已经创建好的表如何修改表字段名称？

当前DLI上已经创建好的表不支持修改表字段名称，即不支持修改表的列名。

可以通过重新创建表，定义新的表字段名称，再把旧表数据迁移到新表进行解决。

4.3 隐私安全

4.3.1 DLI 是否存在 Apache Spark 命令注入漏洞 (CVE-2022-33891) ?

不存在。因为DLI没有启动spark.acls.enable配置项，所以不涉及Apache Spark 命令注入漏洞 (CVE-2022-33891) 。

5 计费相关问题

5.1 DLI 的计费模式是怎样的？

DLI的队列分为以下几种计费模式，您可以根据实际业务情况选择创建不同计费模式的队列。

- **按需计费的队列：**
如果您使用频率较低，购买队列后，使用时才会计费（按自然小时计费）。
队列在空闲（无作业运行）1h后计算资源会被释放；再次使用时，需要重新分配计算资源，大约会耗费5~10min时间。
- **包年/包月队列：**
如果您使用频率较高，建议购买包年/包月队列，计算资源专属，空闲（无作业运行）时不会释放，使用体验更佳，且相较于按需队列费用更低。
- **按需 + 专属资源模式的队列：**
如果您有短时的专属资源需求，也可以在购买按需队列时勾选专属资源模式，空闲（无作业运行）时不会释放，按需使用（计费周期等于队列的生命周期）。
- **按扫描量计费的队列：**
仅对default队列生效。

创建队列的操作指导请参考：[创建队列](#)

具体的计费模式请参考：《[计费说明](#)》。

具体价格请参考：《[数据湖探索价格详情](#)》。

5.2 包周期队列可以转按需队列吗？

包周期队列与按需队列间不支持相互转换，如需更换队列，请重新购买新的队列。

- **自建队列购买方式如下：**
购买队列请参考[创建队列](#)。
- **弹性资源池购买方式如下：**
具体弹性资源池的购买和添加队列操作可以参考《数据湖探索用户指南》中的“[创建弹性资源池](#)”和“[弹性资源池添加队列](#)”相关章节。

5.3 创建的按需队列是否支持转包周期队列？

目前创建的按需队列不支持转包周期队列。

5.4 数据湖队列什么情况下是空闲状态？

队列空闲状态是指在DLI作业管理中，该队列下均无SQL作业运行，或者Flink作业运行、Spark作业运行。

即一个自然小时内无作业运行，该自然小时为空闲状态。不会进行计费。

通常按需计费的队列，在空闲1h后计算资源会被释放，再次使用时，需要重新分配计算资源，可能会耗费5~10min时间。

按需计费以小时为单位进行结算。不足一小时按一小时计费，小时数按整点计算。

例如 12:10 ~12:30、12:10 ~12:55 期间使用，则收取1个小时的费用。

具体的计费模式请参考：[计费说明](#)。

5.5 如何排查计费

如果您在使用DLI服务的过程中，感觉计费有异常，可按照以下步骤进行排查：

□ 说明

以下操作涉及DLI管理控制台界面，详细介绍请参考《数据湖探索用户指南》中的《[作业管理](#)》。

- SQL作业
 - a. 登录DLI管理控制台。
 - b. 进入“作业管理”>“SQL作业”页面。
 - c. 查看需要确认的作业详情，确认在扣费时间段内是否有以下操作：
 - i. 是否使用自建的队列。
 - ii. 是否执行SQL作业。
 - 如果使用了自建的队列执行作业，即为正常计费，默认的计费模式是按CU时计费。
计费详情可参考《[数据湖探索价格详情](#)》。
 - 如果使用的是default队列执行作业，则按照扫描量计费。
计费详情可参考《[数据湖探索价格详情](#)》。
 - 如果没有执行SQL作业，则可继续排查Spark作业。
- Spark作业
 - a. 进入“作业管理”>“Spark作业”页面。
 - b. 查看作业是否已经完成，如果未完成，执行作业的计费模式是按CU时计费。
计费详情可参考《[数据湖探索价格详情](#)》。
如果没有执行Spark作业，则可继续排查Flink作业。

- Flink作业
 - a. 进入“作业管理”>“Flink作业”页面。
 - b. 查看作业是否已经完成，如果未完成，执行作业的计费模式是按CU时计费。
计费详情可参考《[数据湖探索价格详情](#)》。
如果在扣费时间段内没有执行过作业，则请继续排查是否是因为存储了大量数据而产生的计费。
- 数据存储
 - a. 进入“数据管理”>“库表管理”页面。
 - b. 查看所建的数据库中是否有存储在DLI中的数据：
 - i. 单击所要查看的数据库名，进入对应的“表管理”页面；
 - ii. 查看对应表的“数据存储位置”是否为DLI。
如果有存储在DLI中的数据，则按照存储计费的标准计费。
计费详情可参考[计费详情](#)可参考《[数据湖探索价格详情](#)》。

如果上述内容排查后均无问题，可提交[工单](#)，将会有服务支撑人员在后台为您排查。

5.6 购买 4000cus 规格资源，一次性购买 3 个月的和分 3 次购买，每次购买一个 1 月的，这两种付费方式的区别？

一次性购买3个月的，购买的是1个规格为4000CUs的队列，使用期限为3个月。

分3次购买，每次购买1个月的，购买的是3个规格为4000CUs的队列，1个月到期时，3个队列都会到期。

5.7 如何购买 30cus 规格资源？

这种情况，请购买2个16cus规格的队列，不会对业务造成影响。

关于如何购买队列，请参考《数据湖探索用户指南》中的“[创建队列](#)”。

5.8 在一个 16CU 的队列上运行 Flink 作业只用了 2 个 CU，如何计费？

运行Flink作业，按需计算的计费标准是0.35元/CU时，但是这个作业需要在一个队列上面运行，这个队列规格是16CU，运行这个Flink作业只用了2个CU，实际的费用是按照以下公式计算：

CU时资费=使用的队列CU数*使用时长*单价。

和任务所需CU无关，只和使用的队列CU大小有关。即“ $16 * 0.35 \text{元}/\text{CU时}$ ”。

使用时长按自然小时计费，不足一个小时按一个小时计费。

5.9 想要使用扫描套餐包计费，购买队列还要额外收费吗？

按数据扫描量计费是针对默认default队列，即使用default队列时，是按扫描量计费。

您单独购买的队列，可选择按需计费（按CU时计费）或包年包月计费。

具体的计费模式请参考：[计费说明](#)。

6 配额相关问题

6.1 什么是用户配额？

华为云对用户的资源数量和容量做了限制。如果资源配额限制满足不了用户的使用需求，可以通过[工单](#)系统来提交您的申请，并告知您申请提高配额的理由。在通过我们的审理之后，我们会更新您的配额并进行通知。关于配额的具体操作说明，请参见[关于配额](#)。

6.2 怎样查看我的配额

1. 登录管理控制台。
2. 单击管理控制台左上角的 ，选择区域和项目。
3. 在页面右上角，选择“资源 > 我的配额”。
系统进入“服务配额”页面。

图 6-1 我的配额



4. 您可以在“服务配额”页面，查看各项资源的总配额及使用情况。
如果当前配额不能满足业务要求，请参考后续操作，申请扩大配额。

6.3 如何申请扩大配额

如何申请扩大配额？

1. 登录管理控制台。
2. 在页面右上角，选择“资源 > 我的配额”。
系统进入“服务配额”页面。

图 6-2 我的配额



3. 单击“申请扩大配额”。
4. 在“新建工单”页面，根据您的需求，填写相关参数。
其中，“问题描述”项请填写需要调整的内容和申请原因。
5. 填写完毕后，勾选协议并单击“提交”。

6.4 Failed to create the database.

{"error_code":"DLI.1028"; "error_msg": "Already reached the maximum quota of databases:XXX". 提示配额不足，如何处理？}

怎样查看我的配额

1. 登录管理控制台。
2. 单击管理控制台左上角的 ，选择区域和项目。
3. 在页面右上角，选择“资源 > 我的配额”。
系统进入“服务配额”页面。

图 6-3 我的配额



4. 您可以在“服务配额”页面，查看各项资源的总配额及使用情况。
如果当前配额不能满足业务要求，请参考后续操作，申请扩大配额。

如何申请扩大配额？

1. 登录管理控制台。
2. 在页面右上角，选择“资源 > 我的配额”。
系统进入“服务配额”页面。

图 6-4 我的配额



3. 单击“申请扩大配额”。
4. 在“新建工单”页面，根据您的需求，填写相关参数。
其中，“问题描述”项请填写需要调整的内容和申请原因。
5. 填写完毕后，勾选协议并单击“提交”。

7 授权相关问题

7.1 使用咨询

7.1.1 DLI 细粒度授权

DLI服务不仅在服务本身有一套完善的权限控制机制，同时还支持通过统一身份认证服务（Identity and Access Management，简称IAM）细粒度鉴权，可以通过在IAM创建策略来管理DLI的权限控制。

通过IAM，您可以在华为云帐号中给员工创建IAM用户，并使用策略来控制他们对华为云资源的访问范围。例如您的员工中有负责软件开发的人员，您希望他们拥有DLI的使用权限，但是不希望他们拥有删除DLI等高危操作的权限，那么您可以使用IAM为开发人员创建用户，通过授予仅能使用DLI，但是不允许删除DLI的权限策略，控制他们对DLI资源的使用范围。

□ 说明

对于新建的用户，需要先登录一次DLI，记录元数据，后续才可正常使用。

IAM是华为云提供权限管理的基础服务，无需付费即可使用，您只需要为您帐号中的资源进行付费。关于IAM的详细介绍，请参见[《IAM产品介绍》](#)。

如果华为云帐号已经能满足您的需求，不需要创建独立的IAM用户进行权限管理，您可以跳过本章节，不影响您使用DLI服务的其他功能。

DLI 系统权限

如[表7-1](#)所示，包括了DLI的所有系统权限。

权限类别：根据授权精程度分为角色和策略。

- 角色：IAM最初提供的一种根据用户的工作职能定义权限的粗粒度授权机制。该机制以服务为粒度，提供有限的服务相关角色用于授权。由于华为云各服务之间存在业务依赖关系，因此给用户授予角色时，可能需要一并授予依赖的其他角色，才能正确完成业务。角色并不能满足用户对精细化授权的要求，无法完全达到企业对权限最小化的安全管控要求。
- 策略：IAM最新提供的一种细粒度授权的能力，可以精确到具体服务的操作、资源以及请求条件等。基于策略的授权是一种更加灵活的授权方式，能够满足企业

对权限最小化的安全管控要求。例如：针对DLI服务，管理员能够控制IAM用户仅能对某一类云服务器资源进行指定的管理操作。

表 7-1 DLI 系统权限

系统角色/策略名称	描述	类别	依赖关系
DLI FullAccess	数据湖探索所有权限。	系统策略	该角色有依赖，需要在同项目中勾选依赖的角色： <ul style="list-style-type: none">创建跨源连接：VPC ReadOnlyAccess创建包年/包月资源：BSS Administrator创建标签：TMS FullAccess、EPS EPS FullAccess使用OBS存储：OBS OperateAccess创建委托：Security Administrator
DLI ReadOnlyAccess	数据湖探索只读权限。 只读权限可控制部分开放的、未鉴权的DLI资源和操作。例如创建全局变量、创建程序包以及程序包组、default队列提交作业、default数据库下建表、创建跨源连接、删除跨源连接等操作。	系统策略	无
Tenant Administrator	租户管理员。 <ul style="list-style-type: none">操作权限：具有数据湖探索服务资源的所有执行权限。创建后，可通过ACL赋权给其他子用户使用。作用范围：项目级服务。	系统角色	无

系统角色/策略名称	描述	类别	依赖关系
DLI Service Admin	<p>数据湖探索管理员。</p> <ul style="list-style-type: none">操作权限：具有数据湖探索服务资源的所有执行权限。创建后，可通过ACL赋权给其他子用户使用。作用范围：项目级服务。	系统角色	无
DLI Service User	<p>数据湖探索普通用户。</p> <ul style="list-style-type: none">操作权限：具有数据湖探索服务资源的使用权限。无创建资源权限（不可创建数据库（Database）和队列（queue）），其他所有操作均需管理员赋权之后才能使用。作用范围：项目级服务。	系统角色	无

如果系统策略不满足授权要求，管理员可以创建自定义策略，并通过给用户组授予自定义策略来进行精细的访问控制，自定义策略是对系统策略的扩展和补充。详细操作请参考[创建自定义策略](#)。

7.1.2 如何理解 DLI 分区表的列赋权

用户无法对分区表的分区列进行权限操作。

当用户对分区表的任意一列非分区列有权限，则默认对分区列有权限。

当查看用户在分区表上的权限的时候，不会显示对分区列有权限。

7.2 运维指导

7.2.1 欠费导致权限不足

使用DLI提交作业时，提示欠费导致权限不足而提交失败，可以通过查看用户token中的roles信息进行确认：

- op_restrict：表示租户（因余额不足）而受限制。当前帐号余额不足后，该帐号下的所有在线用户token被吊销，如果再次登录，获取的token会被加入op_restrict权限，用户的操作会受到限制。

- op_suspended：表示租户（因欠费等原因）被暂停。当前帐号欠费后，该帐号下的所有在线用户token被吊销，如果再次登录，获取的token会被加入op_suspended的权限，用户操作会受到限制（不包含云服务用户）。

如果提示有这两个role，操作就会受限。

7.2.2 进行程序包更新操作时提示权限不足

问题现象

在程序包管理下，对已经存在的程序包进行更新操作时，提示如下报错信息：

```
"error_code": "DLI.0003", "error_msg": "Permission denied for resource 'resources. xxx', User = 'xxx', Action = \"UPDATE_RESOURCE\"."
```

解决方案

需要给执行作业的用户赋程序包的操作权限，具体操作参考如下：

1. 在DLI管理控制台左侧，单击“数据管理 > 程序包管理”。
2. 在“程序包管理”页面，单击程序包“操作”列中的“权限管理”，进入“用户权限信息”页面。
3. 在单击页面右上角“授权”可对用户进行程序包组/程序包授权，勾选“更新组”权限。
4. 单击“确定”，完成授权。

7.2.3 执行 SQL 查询语句报错：DLI.0003: Permission denied for resource....

问题现象

执行SQL查询语句，提示没有对应资源查询权限。

报错信息：DLI.0003: Permission denied for resource
'databases.dli_test.tables.test.columns.col1', User = '{UserName}', Action =
'SELECT'.

解决措施

出现该问题的原因是由于当前用户没有该表的查询权限。

您可以进入“数据管理 > 库表管理”查找对应库表，查看权限管理，是否配置该帐号的查询权限。

授权方式请参考资料[表权限管理](#)。

7.2.4 已经给表授权，但是再次报错无法查询怎么办？

已经给表授权，且测试查询成功，但一段时间后重试报错无法查询。

这种情况请先检查当前表的权限是否还存在，一般存在两种情况：

- 用户权限被取消。
- 查看表的创建时间，看表是否被他人删除重建，删除表后重建的相同表名并不视作同一张表，不会继承删除表的权限。

7.2.5 表继承数据库权限后，对表重复赋予已继承的权限会报错吗？

当表继承了数据库的权限时，无需重复对表赋予已继承的权限。

在控制台操作表权限时：

- 如果“用户授权”赋予表的权限与继承权限相同，系统会提示已有该权限无需重复操作。
- 通过“项目授权”赋予的权限与继承权限相同时，系统不再向您提醒重复的权限信息。

7.2.6 为什么已有 View 视图的 select 权限，但是查询不了 View？

问题描述

用户A创建了表Table1。

用户B基于Table1创建了视图View1。

赋予用户C Table1的查询表权限后，用户C查询View失败。

根因分析

用户C查询View失败的原因是由于用户B不具备Table1的查询表权限。

解决措施

请在DLI赋予用户B Table1的查询表权限后，用户C重试查询View1。

8 队列相关问题

8.1 使用咨询

8.1.1 DLI 队列管理是否支持修改描述内容

目前已经创建完成的队列不支持修改描述内容，只能在购买队列时进行添加。

8.1.2 删除队列会导致数据库中的表数据丢失吗？

删除队列不会导致您数据库中的表数据丢失。

8.1.3 队列异常时，DLI 怎么保证 Spark 作业的可靠性？

应用侧调用DLI提交作业，需要有重试机制来保证Spark作业的可靠性。如果队列发生异常时，在后续队列恢复后，通过应用侧重试来保证作业的正常提交。

8.1.4 DLI 如何进行队列异常监控？

DLI为用户提供了作业失败的topic订阅功能。

1. 登录DLI控制台。
2. 单击左侧“队列管理”，进入队列管理页面。
3. 在队列管理页面，单击左上角“创建消息通知主题”进行配置。详细操作请参考《[数据湖探索用户指南](#)》。

8.2 运维指导

8.2.1 查看 DLI 队列负载

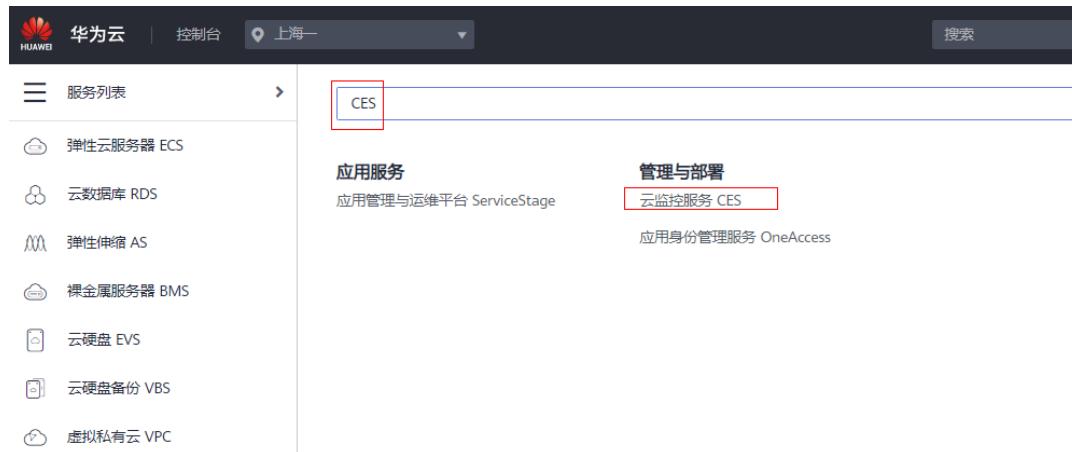
场景概述

如果需要确认DLI队列的运行状态，决定是否运行更多的作业时需要查看队列负载。

操作步骤

- 在控制台搜索“云监控服务 CES”。

图 8-1 搜索 CES



- 进入CES后，在页面左侧“云服务监控”列表中，单击“数据湖探索”。

图 8-2 云服务监控



- 选择队列进行查看。

图 8-3 查看队列负载



8.2.2 如何判断当前 DLI 队列中的作业是否有积压？

问题描述

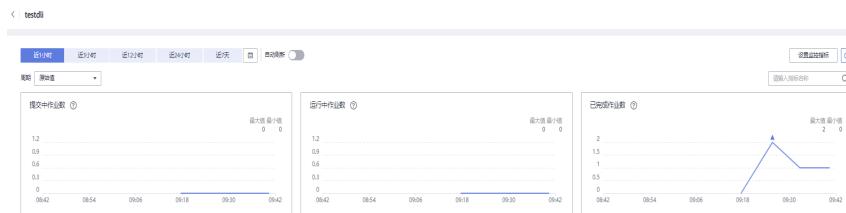
需要查看DLI的队列中作业状态为“提交中”和“运行中”的作业数，判断当前队列中的作业是否有积压。

解决方案

可以通过“云监控服务 CES”来查看DLI队列中不同状态的作业情况，具体操作步骤如下：

1. 在控制台搜索“云监控服务 CES”，进入云监控服务控制台。
2. 在左侧导航栏选择“云服务监控 > 数据湖探索”，进入到云服务监控页面。
3. 在云服务监控页面，“名称”列对应队列名称，单击对应队列名称，进入到队列监控页面。
4. 在队列监控页面，分别查看以下指标查看当前队列的作业运行情况。
 - a. “提交中作业数”：展示当前队列中状态为“提交中”的作业数量。
 - b. “运行中作业数”：展示当前队列中状态为“运行中”的作业数量。
 - c. “已完成作业数”：展示当前队列中状态为“已成功”的作业数量。

图 8-4 查看队列监控指标



8.2.3 如何将老版本的 Spark 队列切换成通用型队列

当前DLI服务包括“SQL队列”和“通用队列”两种队列类型。其中，“SQL队列”用于运行SQL作业，“通用队列”兼容老版本的Spark队列，用于运行Spark作业和Flink作业。

通过以下步骤，可以将老版本的“Spark队列”转换为新的“通用队列”。

1. 重新购买“通用队列”。
2. 将在旧的“Spark队列”中的作业迁移到新的“通用型队列”中，即在提交Spark作业时指定新的队列。
3. 释放旧的“Spark队列”，即删除或退订队列。

8.2.4 为什么云监控 CES 服务看不到 DLI 队列的资源运行情况？

DLI队列在没有作业运行时没有资源和流量的使用，该场景下在CES则不会显示该队列的运行情况。

8.2.5 购买了 64CU 的队列资源，运行 Spark 作业时如何分配？

在DLI中，64CU=64core256G。

在对应的Spark作业中，如果Driver占用了4core16G，那么Executor能占用的就是60core 240G。

8.2.6 创建扩容任务时，提示 Queue plans create failed. The plan xxx target cu is out of quota 报错

场景概述

创建弹性扩缩容定时任务时界面报错，提示信息为：Queue plans create failed. The plan xxx target cu is out of quota。

解决方法

该问题是当前帐号的CU配额不够导致，需要申请扩大配额。关于如何查看配额，如何申请扩大配额，请参考[《关于配额》](#)。

8.2.7 在 default 队列执行 DLI SQL 失败，上报超时异常

问题现象

使用default队列提交SQL作业，作业运行异常，排查作业日志显示Execution Timeout 异常。异常日志参考如下：

[ERROR] Execute DLI SQL failed. Please contact DLI service.
[ERROR] Error message:Execution Timeout

问题原因

default队列是系统预置的默认公共队列，主要用来体验产品功能。当多个用户通过该队列提交作业时，容易发生流控，从而导致作业提交失败。

解决方案

建议不要使用default队列提交作业，可以在DLI控制台新购买SQL队列来提交作业。具体步骤如下：

1. 在DLI管理控制台的左侧导航栏中，选择“队列管理”。
2. 单击“队列管理”页面右上角“购买队列”进行创建队列。
3. 在“购买队列”页面，选择对应参数。注意“队列类型”选择“SQL队列”。
4. 使用新创建的队列重新提交SQL作业。

8.2.8 怎样查看弹性资源池和作业的资源使用情况？

在大数据分析的日常工作中，合理分配和管理计算资源，可以提供良好的作业执行环境。

您可以根据作业的计算需求和数据规模分配资源、调整任务执行顺序，调度不同的弹性资源池或队列资源以适应不同的工作负载。待提交作业所需的CUs需小于等于弹性资源池的剩余可用CUs，才可以确保作业任务的正常执行。

本节操作介绍查看弹性资源池计算资源使用情况、作业所需CU数的查看方法。

怎样查看弹性资源池的资源使用情况？

1. 登录DLI管理管理控制台。
2. 选择“资源管理 > 弹性资源池”。

在弹性资源池的列表页查看资源池的“实际CUs”和“已使用CUs”。

- 实际CUs：弹性资源池当前分配的可用CUs。
- 已使用CUs：当前弹性资源池已经被分配使用的CUs

待提交作业所需的CUs需小于等于弹性资源池的剩余可用CUs，才可以确保作业任务的正常执行。

作业资源的占用情况请参考[怎样查看作业所需的资源CUs数？](#)。

怎样查看作业所需的资源 CUs 数？

- **SQL作业：**

请通过云监控服务提供的监控面板查看运行中的作业数和提交中的作业数，并根据作业数量判断SQL作业整体的资源占用情况。

- **Flink 作业：**

- 登录DLI管理管理控制台。
- 选择“作业管理 > Flink作业”。
- 单击作业名称进入作业详情页面。
- 选择“作业配置信息 > 资源配置”
- 查看作业的CU数量，即作业占用资源总CUs数。

该CUs数可以编辑作业页面进行配置，CUs数量 = 管理单元 + (算子总并行数 / 单TM Slot数) * 单TM所占CUs数。

图 8-5 查看 Flink 作业所需 CUs 数



- **Spark作业：**

- 登录DLI管理管理控制台。
- 选择“作业管理 > Spark作业”。
- 选择要查看的作业，单击操作列的“编辑”进入作业配置页面。
即可查看作业配置的计算资源规格。

计算公式如下：

Spark作业CUs数=Executor所占CU数+driver所占CUs数

Executor所占CU数=max { [(Executor个数 x Executor内存) ÷4],
(Executor个数 x Executor CPU核数) } x1

driver所占CUs数=max [(driver内存÷4), driver CPU核数] x1

说明

- Spark作业未开启高级配置时默认按A类型资源规格配置。
- Spark作业中显示计算资源规格的单位为CPU单位，1CU包含1CPU和4GB内存。上述公式中x1代表CPU单位转换为CU单位。
- 请分别使用内存和CPU核数计算所需的CUs，取两者中的最大值作为Executor 或 driver所需的CU数。

图 8-6 查看 Spark 作业所需 CUs 数

The screenshot shows the 'Compute Resource Specification' section of a Spark job configuration. It includes fields for selecting a resource package (选择依赖资源) and specifying resource configurations (资源规格). A dropdown menu shows a selected configuration: 'B(16核64G内存; Executor内存: 8G, Executors个数: 7个, Exec...)'.

Below this, there is a note: 'System provides 3 types of resource specifications for selection. Specific configuration items support modification: Executor memory, Executor CPU cores, Executor count, Driver CPU cores, and Driver memory. The final configuration result is based on the modified data.'.

The configuration table lists the following parameters:

参数	当前值	修改范围
Executor内存	8	GB
Executor CPU核数	2	
Executor个数	7	
driver CPU核数	2	
driver内存	7	GB

9 跨源连接相关问题

9.1 跨源连接问题

9.1.1 DLI 增强型跨源连接为什么要创建对等连接

创建DLI增强型跨源连接时，创建对等连接的目的是打通网络。以MRS为例，如果DLI和MRS集群在同一个VPC中，安全组且放通的情况下，可以不用配置对等连接。

9.1.2 增强型跨源连接绑定队列失败

问题现象

客户创建增强型跨源连接后，在队列管理测试网络连通性，网络不通，单击对应的跨源连接查看详情，发现绑定队列失败，报错信息如下：

```
Failed to get subnet 86ddcf50-233a-449d-9811-cfef2f603213. Response code : 404, message :  
{"code":"VPC.0202","message":"Query resource by id 86ddcf50-233a-449d-9811-cfef2f603213 fail.the  
subnet could not be found."}
```

原因分析

DLI跨源连接需要使用VPC、子网、路由、对等连接、端口功能，因此需要获得VPC（虚拟私有云）的VPC Administrator权限。

客户未给VPC服务授权导致绑定队列失败。

处理步骤

在DLI控制台，单击“全局配置”>“服务授权”，选择VPC服务，更新委托权限。

图 9-1 服务授权



9.1.3 DLI 增强型跨源连接 DWS 失败

问题现象

客户创建增强型跨源连接DLI和DWS，安全组已配置出方向规则到关联队列，使用的是密码形式的跨源认证，报DLI.0999: PSQLEException: The connection attempt failed。

原因分析

出现该问题可能原因如下：

- 安全组配置不正确
- 子网配置不正确

处理步骤

步骤1 检查客户安全组是否放通，安全组放通规则如下所示。

- 入方向规则：检查本安全组内的入方向网段及端口是否已开放，若没有则添加。
- 出方向规则：检查出方向规则网段及端口是否开放（建议所有网段开放）。

客户安全组入方向和出方向配置的都是DLI队列的子网。建议客户将入方向源地址配成0.0.0.0/0，端口8000，表示任意地址都可以访问DWS8000端口。

步骤2 将入方向源地址配成0.0.0.0/0，端口8000，仍然无法连接，继续排查子网配置。客户的DWS子网关联了网络ACL。网络ACL是一个子网级别的可选安全层，通过与子网关联的出方向/入方向规则控制出入子网的数据流。关联子网后，网络ACL默认拒绝所有出入子网的流量，直至添加放通规则。通过检查，发现其DWS所在子网关联的ACL是空值。

因此，问题的原因是：客户子网关联了网络ACL，但是没有配置出入规则，造成IP地址不可访问。

步骤3 客户配置子网出入规则后，测试成功。

----结束

参考信息

关于出入规则，可以参考《[新建跨源连接，显示已激活，但使用时报communication link failure错误](#)》。

9.1.4 创建跨源成功但测试网络连通性失败怎么办？

问题描述

创建跨源并绑定新创建的DLI队列，测试跨源的网络连通性时失败，有如下报错信息：
failed to connect to specified address

排查思路

以下排查思路根据原因的出现概率进行排序，建议您从高频率原因往低频率原因排查，从而帮助您快速找到问题的原因。

如果解决完某个可能原因仍未解决问题，请继续排查其他可能原因。

- [检查是否在域名或者IP后添加了端口](#)
- [检查是否连接的是对端VPC和子网](#)
- [检查队列的网段是否与数据源网段是否重合](#)
- [检查是否为DLI授权了VPC Administrator权限](#)
- [检查对端安全组是否放通队列的网段](#)
- [检查增强型跨源连接对应的对等连接的路由信息](#)
- [检查VPC网络是否设置了ACL规则限制了网络访问](#)

检查是否在域名或者 IP 后添加了端口

测试连通性时需要添加端口号。

例如，测试队列与指定RDS实例连通性，本例RDS实例使用3306端口。

测试连通性如下所示。

图 9-2 测试地址连通性



检查是否连接的是对端 VPC 和子网

创建增强型跨源连接时需要填写对端的VPC和子网。

例如，测试队列与指定RDS实例连通性，创建连接时需要填写RDS的VPC和子网信息。

图 9-3 创建连接



检查队列的网段是否与数据源网段是否重合

绑定跨源的DLI队列网段和数据源网段不能重合。

您可以从连接日志判断是否是队列与数据源网段冲突。

如图9-4所示，即当前队列A网段与其他队列B网段冲突，且队列B已经建立了与数据源C的增强型跨源连接。因此提示队列A与数据源C的网段冲突，无法建立新的增强型跨源连接。

解决措施：修改队列网段或重建队列。

建议创建队列时就规划好网段划分，否则冲突后只能修改队列网段或重建队列。

图 9-4 查看连接日志-1

对等连接ID	弹性资源池	连接状态
fb39c54c-251a-40e2-a404-5550e94b7032	demo	已激活
查看失败原因	test_0721	已失败

日志详情

```
Failed to create route, vpc id: [REDACTED], error code: 400, message: ["NeutronError": {"type": "DestinationAlreadyExist", "message": "Destination 172.16.16.0/20 already exist for router [REDACTED]", "detail": ""}] The possible reason is that queue on the same network segment have just been moved. The enhanced datasource-source binding can be retried after 10 minutes.
```

检查是否为 DLI 授权了 VPC Administrator 权限

您可以从连接日志判断是否是VPC Administrator权限不足导致的连接失败。

如图9-5、图9-6所示，无法获取对端的子网ID、路由ID，因此跨源连接失败。

解决措施：请在服务授权添加VPC Administrator授权，取消IAM ReadOnlyAccess授权。

图 9-5 查看连接日志-2



图 9-6 查看连接日志-3



图 9-7 DLI 服务授权

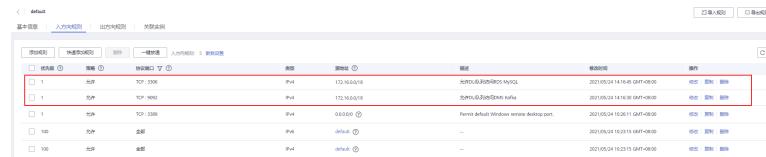


检查对端安全组是否放通队列的网段

创建完跨源连接后，连接的Kafka、DWS、RDS等实例还需要在实例的安全组下添加DLI网段的安全组规则。以对端连接RDS为例：

1. 在DLI管理控制台，单击“资源管理 > 队列管理”，选择您所绑定的队列，单击队列名称旁的 按钮，获取队列的网段信息。
2. 在RDS控制台“实例管理”页面，单击对应实例名称，查看“连接信息”>“数据库端口”，获取RDS数据库实例端口。
3. 单击“连接信息”>“安全组”对应的安全组名称，跳转到RDS实例安全组管理界面。单击“入方向规则 > 添加规则”，优先级设置为“1”，协议选择“TCP”，端口选择RDS数据库实例端口，源地址填写DLI队列的网段。单击“确定”完成配置。

图 9-8 安全组规则



检查增强型跨源连接对应的对等连接的路由信息

检查增强型跨源连接对应的对等连接的路由表，该路由表的本端路由地址网段是否和别的网段有重合，如果重合，路由可能转发错误。

1. 获取增强型跨源连接创建的对等连接ID。

图 9-9 获取对等连接 ID



2. 在VPC-对等连接控制台查看对等连接信息。

图 9-10 查看对等连接



图 9-11 查看队列网段



3. 查看队列对应的VPC的路由表信息。

图 9-12 查看路由表目的地址-1



检查 VPC 网络是否设置了 ACL 规则限制了网络访问

网络ACL对子网进行防护，检查对应子网是否配置了ACL，是否设置了ACL规则限制了网络访问。

例如当您设置了安全组放通队列的网段，同时设置的网络ACL规则包含拒绝该地址访问，那么此安全组规则不生效。

您可以参考[添加网络ACL规则](#)或[修改网络ACL规则](#)放通对应协议端口。

9.1.5 怎样配置 DLI 队列与数据源的网络连通？

• 配置DLI队列与内网数据源的网络联通

DLI在创建运行作业需要连接外部其他数据源，如：DLI连接MRS、RDS、CSS、Kafka、DWS时，需要打通DLI和外部数据源之间的网络。

DLI提供的增强型跨源连接功能，底层采用对等连接的方式打通与目的数据源的vpc网络，通过点对点的方式实现数据互通。

图 9-13 增强型跨源连接配置流程



增强型跨源连接配置示例请参考[配置DLI队列与内网数据源的网络联通](#)。

- **配置DLI队列与公网网络联通**

通过配置SNAT规则，添加到公网的路由信息，可以实现队列到和公网的网络打通。

图 9-14 配置 DLI 队列访问公网流程



详细操作请参考[配置DLI 队列与公网网络联通](#)。

9.1.6 DLI 创建跨源连接，进行绑定队列一直在创建中怎么办？

跨源连接创建慢，有以下几种可能：

- 购买DLI队列后，第一次进行绑定队列。通常需要等待5~10分钟，待后台拉起集群后，即可创建成功。
- 若刚刚对队列进行网段修改，立即进行绑定队列。通常需要等待5~10分钟，待后台重建集群后，即可创建成功。

9.1.7 按需队列怎样绑定跨源链接？

增强型跨源仅支持“包年/包月”队列和“按需”的专属队列。

因此只有在购买按需队列时勾选了“专属资源模式”，资源专属，按需使用（计费周期等于队列的生命周期），才能绑定跨源链接。

9.1.8 如何打通 DLI 和数据源的网络？

DLI 增强型跨源连接底层采用对等连接，直接打通DLI集群与目的数据源的VPC网络，通过点对点的方式实现数据互通。

创建增强型跨源连接的操作步骤请参考[创建增强型跨源连接](#)。

9.2 跨源分析问题

9.2.1 如何实现跨数据源查询

跨数据源查询步骤如下：

1. 假设数据存放在华为云上多个不同的地方，例如存放在OBS上。
2. 在DLI中创建数据表，数据表对应的位置配置为数据所在的位置，例如OBS的某个桶路径（该数据实际还是在OBS中，不用迁移）。
3. 在DLI中编写SQL直接查询分析这些数据。

9.2.2 如何实现跨区域访问数据

1. 打通不同区域间的VPC网络，具体请参考《[跨区域同账号VPC互通](#)》或《[跨区域跨账号VPC互通](#)》。
2. 在DLI中创建增强型跨源链接，绑定队列。具体请参考《[增强型跨源连接](#)》。

3. 添加DLI的路由。具体请参考《[增强型跨源连接](#)》。

9.2.3 创建 DLI 关联 RDS 表时，如果 RDS 表有自增主键或者其他自动填充的字段时，在 DLI 中插入数据时，对应字段如何填写？

在DLI中插入数据时，ID字段填写NULL。

9.3 跨源连接运维报错

9.3.1 新建跨源连接，显示已激活，但使用时报 communication link failure 错误

- 原因

网络连通性问题，建议用户检查安全组选择是否正确，检查安全组网络（vpc）配置。

- 解决方法：

示例：创建RDS跨源，使用时报“communication link failure”错误。

- 将原有跨源连接删除重新创建。再次创建时，必须确保所选“安全组”、“虚拟私有云”、“子网”和“目的地址”与RDS中的设置完全一致。

说明

请选择正确的“服务类型”，本示例中为“RDS”。

图 9-15 创建经典型跨源连接-RDS



- b. 检查安全组网络（vpc）配置。

若按照**步骤1**重建跨源连接后还是报错“communication link failure”，则检查vpc配置。

- 经典型跨源：

- 入方向规则：检查本安全组内的入方向网段及端口是否已开放，若没有则添加。

检查网段及端口是否配置。

图 9-16 检查网段及端口是否配置



如果不存在，则进行添加。

图 9-17 添加入方向规则



- 出方向规则：检查出方向规则网段及端口是否开放（建议所有网段开放）。

检查网段及端口是否配置。

图 9-18 检查网段及端口是否配置。



如果不存在，则进行添加。

图 9-19 添加出方向规则



■ 增强型跨源

检查DLI队列对应网段是否开放，若没有，则在vpc中添加出方向网段。

- 1) 在DLI服务找到跨源连接绑定队列对应的网段

图 9-20 查找跨源连接绑定队列对应的网段



- 2) 在虚拟私有云安全组中查看DLI队列对应的网段是否已配置。

图 9-21 查看 vpc 中对应安全组中 DLI 队列对应网段



如果没有配置，则进行添加。

图 9-22 在 VPC 中添加对应网段



如果按照上述步骤检查之后，还是存在问题，请联系技术人员提供帮助。

9.3.2 跨源访问 MRS HBase，连接超时，日志未打印错误

用户在跨源连接中没有添加集群主机信息，导致KRB认证失败，故连接超时，日志也未打印错误。建议配置主机信息后重试。

在“增强型跨源”页面，单击该连接“操作”列中的“修改主机信息”，在弹出的对话框中，填写主机信息。格式为：“IP 主机名/域名”，多条信息之间以换行分隔。

MRS主机信息获取，详细请参考《数据湖探索用户指南》中的“[修改主机信息](#)”章节描述。

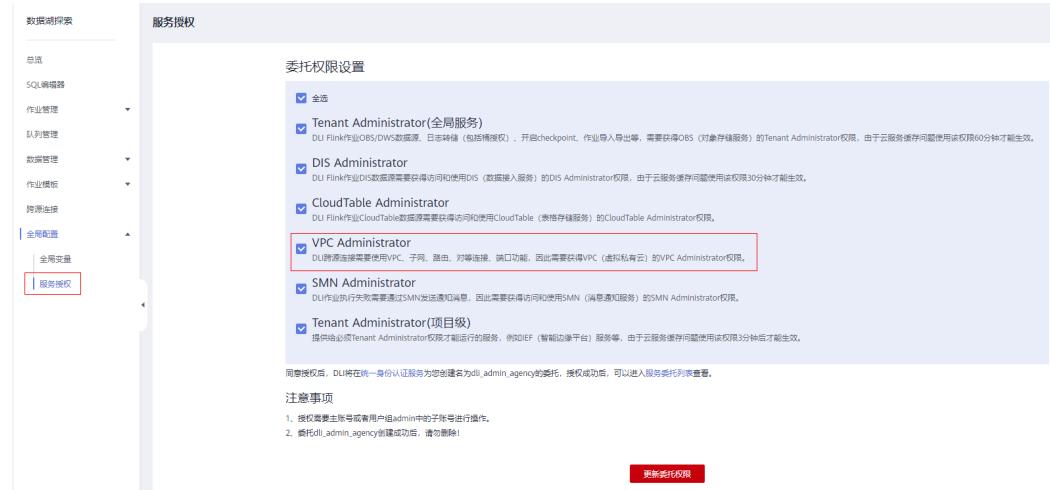
9.3.3 DLI 跨源连接报错找不到子网

跨源连接创建对等连接失败，报错信息如下：

```
Failed to get subnet 2c2bd2ed-7296-4c64-9b60-ca25b5eee8fe. Response code : 404, message :  
{ "code": "VPC.0202", "message": "Query resource by id 2c2bd2ed-7296-4c64-9b60-ca25b5eee8fe fail.the  
subnet could not be found." }
```

创建跨源连接之前，需要确认是否勾选了“VPC Administrator”，如果只是勾选了全局的tenant administrator，会提示找不到子网。

图 9-23 服务授权



9.3.4 跨源 RDS 表，执行 insert overwrite 报 Incorrect string value 错误

问题现象

客户在数据治理中心DataArts Studio创建DLI的跨源RDS表，执行insert overwrite语句向RDS写入数据报错：DLI.0999: BatchUpdateException: Incorrect string value: '\xF0\x9F\x90\xB3' for column 'robot_name' at row 1。

原因分析

客户的数据中存在emoj表情，这些表情是按照四个字节一个单位进行编码的，而通常使用的utf-8编码在mysql数据库中默认是按照三个字节一个单位进行编码的，这个原因导致将数据存入mysql数据库时出现错误。

出现该问题可能原因如下：

- 数据库编码问题。

处理步骤

修改字符集为utf8mb4。

步骤1 执行如下SQL更改数据库字符集。

```
ALTER DATABASE DATABASE_NAME DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
```

步骤2 执行如下SQL更改表字符集。

```
ALTER TABLE TABLE_NAME DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
```

步骤3 执行如下SQL更改表中所有字段的字符集。

```
ALTER TABLE TABLE_NAME CONVERT TO CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
```

----结束

参考信息

[如何确保RDS for MySQL数据库字符集正确](#)

9.3.5 创建 RDS 跨源表报空指针错误

问题现象

客户创建RDS跨源表失败，报空指针的错误。

原因分析

客户建表语句：

```
CREATE TABLE IF NOT EXISTS dli_to_rds
USING JDBC OPTIONS (
'url'='jdbc:mysql://to-rds-1174405119-oLRHAGE7.datasource.com:5432/postgreDB',
'driver'='org.postgresql.Driver',
'dbtable'='pg_schema.test1',
'passwdauth' = 'xxx',
'encryption' = 'true');
```

客户的RDS数据库为PostGre集群，url的协议头填写错误导致。

处理步骤

修改url为'url'='jdbc:postgresql://to-rds-1174405119-oLRHAGE7.datasource.com:5432/postgreDB'，重新创建跨源表成功。

创建跨源表关联RDS语法请参考《[数据湖探索SQL语法参考](#)》。

9.3.6 对跨源 DWS 表执行 insert overwrite 操作，报错： `org.postgresql.util.PSQLException: ERROR: tuple concurrently updated`

问题现象

客户对DWS执行并发insert overwrite操作，报错：org.postgresql.util.PSQLException: ERROR: tuple concurrently updated。

原因分析

客户作业存在并发操作，同时对一张表执行两个insert overwrite操作。

一个cn在执行：

```
TRUNCATE TABLE BI_MONITOR.SAA_OUTBOUND_ORDER_CUST_SUM
```

另外一个cn在执行：

```
call bi_monitor.pkg_saa_out_bound_monitor_p_saa_outbound_order_cust_sum
```

这个函数里面有delete 和 insert SAA_OUTBOUND_ORDER_CUST_SUM的操作。

处理步骤

修改作业逻辑，避免对同一张表并发执行insert overwrite操作。

9.3.7 通过跨源表向 CloudTable Hbase 表导入数据，executor 报错：`RegionTooBusyException`

问题现象

客户通过DLI跨源表向CloudTable Hbase导入数据，原始数据：HBASE表，一个列簇，一个rowkey运行一个亿的模拟数据，数据量为9.76GB。导入1000W条数据后作业失败。

原因分析

1. 查看driver错误日志。
2. 查看executor错误日志。
3. 查看task错误日志。

结论：rowkey过于集中，出现了热点region。

处理步骤

1. Hbase做预分区。
2. 把rowkey散列化。

建议与总结

建议DLI在写入数据时也将数据离散化，避免大量数据写入同一个regionServer，同时，在insert语句后增加distribute by rand()。

9.3.8 通过 DLI 跨源写 DWS 表，非空字段出现空值异常

问题现象

客户在DWS建表，然后在DLI创建跨源连接读写该表，突然出现如下异常，报错信息显示DLI向该表某非空字段写入了空值，因为非空约束存在导致作业出错。

报错信息如下：

```
DLI.0999: PSQLEException: ERROR: dn_6009_6010: null value in column "ctr" violates not-null constraint  
Detail: Failing row contains (400070309, 9.00, 25, null, 2020-09-22, 2020-09-23 04:30:01.741).
```

原因分析

1. DLI源表对应字段ctr为double类型。

图 9-24 创建源表



```
CREATE EXTERNAL TABLE ...  
    ...  
    'ctr' DOUBLE,  
    ...  
)
```

2. 目标表对应字段类型为decimal(9,6)。

图 9-25 创建目标表



```
CREATE TABLE ...  
    ...  
    'ctr' DECIMAL(9, 6),  
    ...  
) USING DWS OPTIONS (...  
    ...  
) TBLPROPERTIES (...  
)
```

3. 查询源表数据，发现导致问题产生的记录ctr值为1675，整数位（4位）超出所定义的decimal精度（9 - 6 = 3位），导致double转decimal时overflow产生null值，而对应dws表字段为非空导致插入失败。

处理步骤

修改目的表所定义的decimal精度即可解决。

9.3.9 更新跨源目的端源表后，未同时更新对应跨源表，导致 insert 作业失败

问题现象

客户在DLI中创建了DWS跨源连接和DWS跨源表，然后对DWS中的源表schema进行更新，执行DLI作业，发现DWS中源表schema被修改为更新前的形式，导致schema不匹配，作业执行失败。

原因分析

DLI跨源表执行insert操作时，会将DWS源表删除重建，客户没有对应更新DLI端跨源表建表语句，导致更新的DWS源表被替换。

处理步骤

新建DLI跨源表，并添加建表配置项 `truncate = true`（只清空表数据，不删除表）。

建议与总结

在更新跨源目的端源表后，必须同时更新对应DLI跨源表。

10 使用 API 相关问题

10.1 如何获取 AK/SK?

访问密钥即AK/SK（Access Key ID/Secret Access Key），表示一组密钥对，用于验证调用API发起请求的访问者身份，与密码的功能相似。用户通过调用API接口进行云资源管理（如创建集群）时，需要使用成对的AK/SK进行加密签名，确保请求的机密性、完整性和请求双方身份的正确性。获取AK/SK操作步骤如下：

1. 注册并登录华为云管理控制台。
2. 将鼠标移动到右上角用户名上，在下拉列表中单击“我的凭证”。
3. 在左侧导航栏单击“访问密钥”。
4. 单击“新增访问密钥”，进入“新增访问密钥”页面。
5. 根据提示输入对应信息，单击“确定”，在弹出的提示页面单击“立即下载”。
6. 下载成功后，打开凭证文件，获取AK/SK信息。

说明

为防止访问密钥泄露，建议您将其保存到安全的位置。

10.2 如何获取项目 ID?

项目ID是系统所在区域的ID。用户在调用API接口进行云资源管理（如创建集群）时，需要提供项目ID。

查看项目ID步骤如下：

1. 注册并登录华为云管理控制台。
 2. 将鼠标移动到右上角用户名上，在下拉列表中单击“我的凭证”。
- 在“我的凭证”页面的项目列表中查看项目ID。例如
project_id:"5a3314075bfa49b9ae360f4ecd333695"。

10.3 提交 SQL 作业时，返回“unsupported media Type”信息

在DLI提供的REST API中，可以在请求URI中附加请求消息头，例如：Content-Type。

“Content-Type”为消息体的类型（格式），默认取值为“application/json”。

提交SQL作业的URI为：POST /v1.0/{project_id}/jobs/submit-job

其“Content-Type”只支持“application/json”，若设置为“text”则会报错，报错信息为“unsupported media Type”。

10.4 不同帐号的调用 API 时的项目 ID(Project ID)是固定的吗？

不同IAM帐号调用相同区域（Region）同一个企业项目下的资源项目ID（Project ID）是一样的。

10.5 创建 SQL 作业的 API 执行超过时间限制，运行超时报错

问题现象

DLI上调用“提交SQL作业”API运行超时，报如下错误信息：
There are currently no resources tracked in the state, so there is nothing to refresh.

问题根因

API以同步模式调用运行时会有两分钟的超时时间限制，如果API调用超过该时间限制则会超时报错。

解决方案

调用“提交SQL作业”API时可以通过设置“dli.sql.sqlasync.enabled”参数为“true”来异步运行该作业。

具体可以参考[提交SQL作业API](#)。

10.6 API 接口返回的中文字符为乱码，如何解决？

DLI接口返回的结果编码格式为“UTF-8”，在调用接口获取返回结果时需要对返回的信息编码转换为“UTF-8”。

例如，参考如下实现对返回的response.content内容做编码格式转换，确保返回的中文格式不会乱码。

```
print(response.content.decode("utf-8"))
```

11 使用 SDK 相关问题

11.1 使用 SDK 查询 SQL 作业结果，如何设置超时时间？

查询SQL作业结果SDK，在提交时会确认作业状态，系统设置的超时时间是300s，如果作业不是“FINISHED”状态，在达到300s后会抛出异常超时的错误。

建议使用getJobId()获取jobid，然后调用queryJobResultInfo(String jobId)或者cancelJob(String jobId)获取结果或者取消作业。

11.2 服务器：请求 dli.xxx,unable to resolve host address, 如何解决？

1. 首先使用ping命令测试dli.xxx是否可以访问。
如果dli.xxx可以正常访问，建议下DNS解析是否配置正确。
2. DLI 不支持跨区域服务，需在一个区域内进行访问。

A 修订记录

发布日期	修订记录
2024-01-30	<ul style="list-style-type: none">新增怎样查看弹性资源池和作业的资源使用情况？。
2023-05-11	<ul style="list-style-type: none">新增在使用Spark2.3.x访问元数据时，DataSource语法创建avro类型的OBS表创建失败。
2023-03-31	<ul style="list-style-type: none">新增为什么Spark3.x的作业日志中打印找不到global_temp数据库。
2022-10-28	<ul style="list-style-type: none">新增创建跨源成功但测试网络连通性失败怎么办？
2022-10-10	<ul style="list-style-type: none">常见问题手册结构调整。新增DLI Flink与MRS Flink有什么区别？
2022-09-29	<ul style="list-style-type: none">新增SQL作业使用咨询。新增Spark作业使用咨询。