



# 数据湖探索 (DLI)

## 故障排除

天翼云科技有限公司

<b>1 Flink 作业相关问题</b> .....	<b>8</b>
1.1 使用咨询 .....	8
1.1.1 DLI Flink 作业支持哪些数据格式和数据源.....	8
1.1.2 如何给予用户授权查看 Flink 作业? .....	8
1.1.3 Flink 作业设置“异常自动重启” .....	9
1.1.4 Flink 作业如何保存作业日志 .....	9
1.1.5 Flink 作业管理界面对用户进行授权时报用户不存在怎么办 .....	9
1.1.6 手动停止了 Flink 作业，再次启动时没有提示从哪个 Checkpoint 恢复 .....	10
1.1.7 DLI 使用 SMN 主题，提示 SMN 主题不存在，怎么处理? .....	10
1.2 Flink SQL 作业相关问题 .....	10
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? .....	10
1.2.2 Flink SQL 中的 temp 流中数据是否需要定期清理，如何清理? .....	10
1.2.3 创建 Flink SQL 作业时选择 OBS 桶，提示未授权 .....	10
1.2.4 Flink SQL 作业将 OBS 表映射为 DLI 的分区表 .....	11
1.2.5 OBS 表如何映射为 DLI 的分区表? .....	12
1.2.6 在 Flink SQL 作业中创建表使用 EL 表达式，作业运行报 DLI.0005 错误.....	12
1.2.7 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.....	13
1.2.8 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.....	14
1.2.9 Flink SQL 作业读取 DIS 数据报 Not authorized 错误 .....	14
1.2.10 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 .....	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? .....	17
1.3.3 Flink Jar 包冲突，导致提交失败 .....	17
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 .....	17
1.3.5 Flink Jar 作业运行报错，报错信息为 Authentication failed.....	18
1.3.6 Flink Jar 作业设置 backend 为 OBS，报错不支持 OBS 文件系统.....	19
1.3.7 Hadoop jar 包冲突，导致 Flink 提交失败 .....	19
1.3.8 Flink jar 如何连接 SASL_SSL? .....	20
1.4 性能调优 .....	20

1.4.1 Flink 作业如何进行性能调优 .....	20
1.4.2 如何在一个 Flink 作业中将数据写入到不同的 Elasticsearch 集群中? .....	23
1.4.3 Flink 作业重启后, 如何保证不丢失数据? .....	23
1.5 运维指导 .....	24
1.5.1 Flink 作业提交错误, 如何定位 .....	24
1.5.2 Flink 作业运行异常, 如何定位 .....	24
1.5.3 Flink 作业重启后, 如何判断是否可以从 checkpoint 直接恢复 .....	24
1.5.4 作业语义检验时提示 DIS 通道不存在怎么处理? .....	25
1.5.5 如何处理作业的 OBS Bucket 没有授权? .....	25
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 .....	25
1.5.7 访问 FlinkUI/SparkUI 页面无法正常显示怎么办? .....	26
1.5.8 Jobmanager 与 Taskmanager 心跳超时, 导致 Flink 作业异常怎么办? .....	26
1.5.9 Flink jobmanager 日志一直报 Timeout expired while fetching topic metadata 怎么办? .....	27
<b>2 SQL 作业相关问题 .....</b>	<b>28</b>
2.1 使用咨询 .....	28
2.1.1 SQL 作业使用咨询 .....	28
2.2 作业开发 .....	29
2.2.1 如何合并小文件 .....	29
2.2.2 创建 OBS 表时正确指定 OBS 路径 .....	29
2.2.3 关联 OBS 桶中嵌套的 JSON 格式数据如何创建表 .....	29
2.2.4 在 SQL 语句中如何设置局部变量 .....	29
2.2.5 count 函数如何进行聚合 .....	30
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? .....	30
2.2.7 SQL 作业如何指定表的部分字段进行表数据的插入 .....	30
2.3 作业运维报错 .....	30
2.3.1 用户导表到 OBS 报 “path obs://xxx already exists” 错误 .....	30
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.; .....	31
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. ....	31
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 .....	31
2.3.5 LOAD 数据到 OBS 外表报错: IllegalArgumentException: Buffer size too small. size .....	31
2.3.6 SQL 作业运行报错: DLI.0002 FileNotFoundException .....	32
2.3.7 用户通过 CTAS 创建 hive 表报 schema 解析异常错误 .....	32
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 .....	33
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 .....	33
2.3.10 SQL 作业访问报错: File not Found .....	34
2.3.11 SQL 作业访问报错: DLI.0003: AccessControlException XXX .....	34

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].....	35
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. ....	35
2.4 运维指导 .....	35
2.4.1 SQL 作业运行慢如何定位 .....	35
2.4.2 查看 DLI SQL 日志 .....	39
2.4.3 查看 DLI 的执行 SQL 记录 .....	39
2.4.4 配置 AE 参数解决数据倾斜 .....	39
2.4.5 DLI 控制台中无法查询到对应表 .....	40
2.4.6 OBS 表压缩率较高.....	41
2.4.7 如何避免字符码不一致导致的乱码.....	41
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 .....	41
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.....	42
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.....	42
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.....	43
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 .....	43
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 .....	44
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？ .....	44
2.4.15 为什么 SQL 作业一直处于“提交中”？ .....	45
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？ .....	45
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？ .....	45
2.4.18 从 DLI 导入数据到 OBS，为什么数据量出现差异？ .....	45
<b>3 Spark 作业相关问题.....</b>	<b>47</b>
3.1 使用咨询 .....	47
3.1.1 Spark 作业使用咨询 .....	47
3.2 作业开发 .....	47
3.2.1 Spark 如何将数据写入到 DLI 表中.....	47
3.2.2 通用队列操作 OBS 表如何设置 AK/SK.....	48
3.2.3 如何查看 DLI Spark 作业的实际资源使用情况.....	49
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？ .....	50
3.2.5 如何在 DLI 中运行复杂 PySpark 程序？ .....	51
3.2.6 Spark 作业访问 MySQL 数据库的方案 .....	51
3.2.7 如何通过 JDBC 设置 spark.sql.shuffle.partitions 参数提高并行度.....	52
3.2.8 Spark jar 如何读取上传文件 .....	52
3.3 作业运维报错 .....	53
3.3.1 Spark 作业访问 OBS 数据时报 ResponseCode: 403 和 ResponseStatus: Forbidden 错误.....	53
3.3.2 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] .....	53

3.3.3 Spark 作业运行大批量数据时上报作业运行超时异常错误.....	53
3.3.4 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.....	53
3.3.5 执行作业的用户数据库和表权限不足导致作业运行失败.....	53
3.4 运维指导 .....	54
3.4.1 添加 Python 包后，找不到指定的 Python 环境.....	54
3.4.2 为什么 Spark jar 作业 一直处于“提交中”？ .....	54
<b>4 产品咨询相关问题.....</b>	<b>55</b>
4.1 使用咨询 .....	55
4.1.1 什么是 DLI.....	55
4.1.2 DLI 支持哪些数据格式.....	55
4.1.3 DLI 中的 Spark 组件与 MRS 中的 Spark 组件有什么区别？ .....	55
4.1.4 DLI 的数据可存储在哪些地方.....	55
4.1.5 DLI 表与 OBS 表的区别 .....	56
4.1.6 不上传数据到 OBS，如何使用 DLI .....	56
4.1.7 DLI 是否支持导入其他租户共享 OBS 桶的数据？ .....	56
4.1.8 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？ .....	56
4.1.9 全局变量的使用中，一个子账号是否可以使用其他子账号创建的全局变量 .....	57
4.2 作业管理 .....	57
4.2.1 使用 DLI 运行几万个作业，如何管理？ .....	57
4.2.2 DLI 上已经创建好的表如何修改表字段名称？ .....	57
4.3 隐私安全 .....	57
4.3.1 DLI 是否存在 Apache Spark 命令注入漏洞（CVE-2022-33891）？ .....	57
<b>5 配额相关问题.....</b>	<b>58</b>
5.1 怎样查看我的配额 .....	58
5.2 如何申请扩大配额 .....	58
<b>6 授权相关问题.....</b>	<b>59</b>
6.1 使用咨询 .....	59
6.1.1 DLI 细粒度授权.....	59
6.1.2 如何理解 DLI 分区表的列赋权.....	60
6.2 运维指导 .....	60
6.2.1 欠费导致权限不足 .....	60
6.2.2 进行程序包更新操作时提示权限不足.....	61
6.2.3 执行 SQL 查询语句报错：DLI.0003: Permission denied for resource.....	61
6.2.4 已经给表授权，但是再次报错无法查询怎么办？ .....	61
6.2.5 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？ .....	62
6.2.6 为什么已有 View 视图的 select 权限，但是查询不了 View？ .....	62
<b>7 队列相关问题.....</b>	<b>63</b>

7.1 使用咨询 .....	63
7.1.1 DLI 队列管理是否支持修改描述内容 .....	63
7.1.2 删除队列会导致数据库中的表数据丢失吗? .....	63
7.1.3 队列异常时, DLI 怎么保证 Spark 作业的可靠性? .....	63
7.1.4 DLI 如何进行队列异常监控? .....	63
7.2 运维指导 .....	63
7.2.1 查看 DLI 队列负载 .....	63
7.2.2 如何判断当前 DLI 队列中的作业是否有积压? .....	64
7.2.3 如何将老版本的 Spark 队列切换到通用型队列 .....	64
7.2.4 为什么云监控 CES 服务看不到 DLI 队列的资源运行情况? .....	64
7.2.5 购买了 64CU 的队列资源, 运行 Spark 作业时如何分配? .....	64
7.2.6 创建扩容任务时, 提示 Queue plans create failed. The plan xxx target cu is out of quota 报错 .....	65
7.2.7 在 default 队列执行 DLI SQL 失败, 上报超时异常 .....	65
<b>8 跨源连接相关问题 .....</b>	<b>66</b>
8.1 跨源连接问题 .....	66
8.1.1 DLI 增强型跨源连接为什么要创建对等连接 .....	66
8.1.2 增强型跨源连接绑定队列失败 .....	66
8.1.3 DLI 增强型跨源连接 DWS 失败 .....	66
8.1.4 创建跨源成功但测试网络连通性失败怎么办? .....	67
8.1.5 怎样配置 DLI 队列与数据源的网络连通? .....	69
8.1.6 DLI 创建跨源连接, 进行绑定队列一直在创建中怎么办? .....	70
8.1.7 如何打通 DLI 和数据源的网络? .....	70
8.2 跨源分析问题 .....	70
8.2.1 如何实现跨数据源查询 .....	70
8.2.2 如何实现跨区域访问数据 .....	70
8.2.3 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? .....	71
8.3 跨源连接运维报错 .....	71
8.3.1 新建跨源连接, 显示已激活, 但使用时报 communication link failure 错误 .....	71
8.3.2 跨源访问 MRS HBase, 连接超时, 日志未打印错误 .....	71
8.3.3 DLI 跨源连接报错找不到子网 .....	71
8.3.4 跨源 RDS 表, 执行 insert overwrite 报 Incorrect string value 错误 .....	72
8.3.5 创建 RDS 跨源表报空指针错误 .....	72
8.3.6 对跨源 DWS 表执行 insert overwrite 操作, 报错: org.postgresql.util.PSQLException: ERROR: tuple concurrently updated .....	73
8.3.7 通过跨源表向 CloudTable Hbase 表导入数据, executor 报错: RegionTooBusyException .....	73
8.3.8 通过 DLI 跨源写 DWS 表, 非空字段出现空值异常 .....	74
8.3.9 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 .....	75
<b>9 使用 API 相关问题 .....</b>	<b>76</b>

9.1 提交 SQL 作业时，返回 “unsupported media Type” 信息.....	76
9.2 不同账号的调用 API 时的项目 ID(Project ID)是固定的吗? .....	76
9.3 创建 SQL 作业的 API 执行超过时间限制，运行超时报错.....	76
<b>10 使用 SDK 相关问题.....</b>	<b>78</b>
10.1 使用 SDK 查询 SQL 作业结果，如何设置超时时间? .....	78
10.2 服务器：请求 dli.xxx,unable to resolve host address, 如何解决? .....	78

# 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

### 1.1.2 如何给予用户授权查看 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
  - a. 使用租户账号，或者作业 owner 账号，或有 DLI Service Administrator 权限的账号，登录 DLI 控制台。
  - b. 在“作业管理” > “Flink 作业”页面找到对应的作业。
  - c. 在对应作业的“操作”栏中选择“更多” > “权限管理”。
  - d. 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。
- IAM 授权管理
  - a. 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。
- 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”。

### 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。

### 1.1.5 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云系统才能进行授权操作。

## 1.1.6 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。

## 1.1.7 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。

```
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 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。

```
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;
```

## 1.2.6 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}。

### 1.2.7 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。

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;
```

## 1.2.8 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。

## 1.2.9 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，没有对运行的用户账号授权获取 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。

## 1.2.10 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

## 1.3 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.ExternalizedCheckpointCleanup.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
- StreamExecutionEnvironment 依赖的包需要在 pom 文件中添加如下配置。

```
<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。

- 参数优化的约束与限制
  - i. 需要在 DLI 控制台“全局配置 > 服务授权”开启 Tenant Administrator（全局服务）。
  - ii. 写入数据到 OBS 的桶必须为主账号下所创建的 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：

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



- 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 DLI 授权，DLI 服务才能访问此桶下的文件。
3. 查看作业是否从 checkpoint 恢复。

## 1.3.2 Flink Jar 作业是否支持上传配置文件，要如何操作？

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/fileName")` 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)。

## 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。

## 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”。

### 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。
- 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。

### 原因分析

1. 确认 OBS 桶名是否正确。
2. 确认所用 AKSK 是否有权限。
3. 设置依赖关系 provided 防止 Jar 包冲突。
4. 确认客户 esdk-obs-java-3.1.3.jar 的版本。
5. 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。

## 1.4 性能调优

### 1.4.1 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。

- 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。

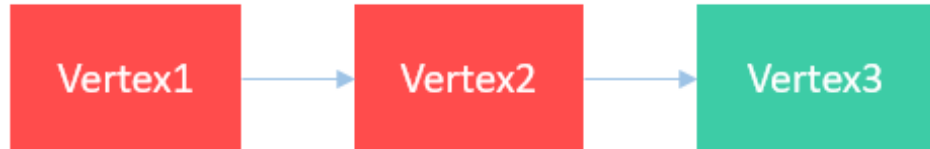
- 时延

`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：

## 性能分析

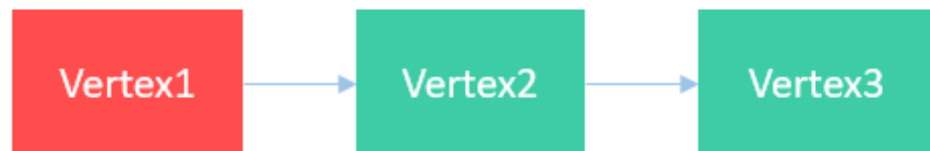
由于 Flink 的反压机制，流作业在存在性能问题的情况下，会导致数据源消费速率跟不上生产速率，从而引起 Kafka 消费组的积压。在这种情况下，可以通过算子的反压和时延，确定算子的性能瓶颈点。

- 作业最后一个算子(Sink)反压正常（绿色），前面算子反压高（红色）



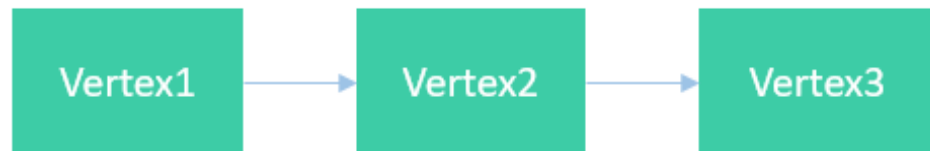
该场景说明性能瓶颈点在 sink，此时需要根据具体数据源具体优化，比如对于 JDBC 数据源，可以通过调整写出批次(`connector.write.flush.max-rows`)、JDBC 参数重写(`rewriteBatchedStatements=true`)等进行优化。

- 作业非倒数第二个算子反压高（红色）



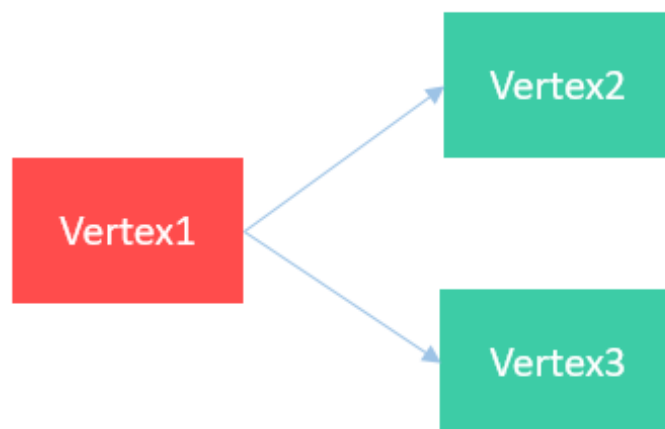
该场景说明性能瓶颈点在 Vertex2 算子，可以通过查看该算子描述，确认该算子具体功能，以进行下一步优化。

- 所有算子反压都正常（绿色），但存在数据堆积



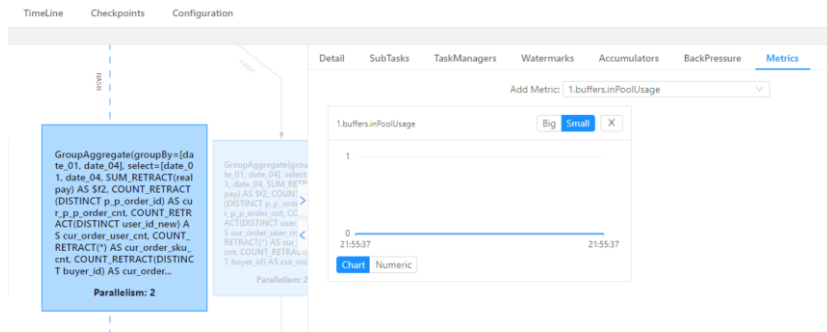
该场景说明性能瓶颈点在 Source，主要是受数据读取速度影响，此时可以通过增加 Kafka 分区数并增加 source 并发解决。

- 作业一个算子反压高（红色），而其后续的两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1-1 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

## 1.4.2 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;
```

## 1.4.3 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)。
  - 对于 Flink Jar 作业, 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。

### 📖 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。

## 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。
  - 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再执行 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。
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 直接恢复

Flink 作业重启后，可以从两个角度判断是否可以从 checkpoint 直接恢复：



- 调整或者增加优化参数、并行度，修改 Flink SQL 语句、Flink Jar 作业等操作，无法从 checkpoint 直接恢复。
- 仅修改“单 TM 所占 CU 数”参数时，可以从 checkpoint 直接恢复。

### 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 1: “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 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

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。

## 1.5.8 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1-2 异常信息

```
Jobmanager.log
2021-05-17 22:44:37.312 INFO [70] org.apache.flink.runtime.checkpoint.CheckpointCoordinator - Triggering checkpoint 223 @ 1621262677310 for job 00d04eb6e7147e59f2d2877bdb48ce4d.
2021-05-17 22:46:01.729 INFO [3659] org.apache.flink.runtime.executiongraph.ExecutionGraph - Map -> Map (4/4) (9e6b13c7ab5d3637062f1697014b4eff) switched from RUNNING to FAILED.
2021-05-17 22:46:01.729 INFO [3659] org.apache.flink.runtime.executiongraph.ExecutionGraph - java.util.concurrent.TimeoutException: Heartbeat of TaskManager with id container_1619690508608_1067_01_000003 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_000003
2021-05-17 22:45:44.078 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.client.FourLetterWordMain - connecting to node-master1fmxz.b1039268-c9c0-4c82-b21f-b1252b0f186f.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 0x2005c1aeb4c0021, 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: java.security.auth.login.LoginException: No JAAS configuration section named 'Client' was found in specified JAAS configuration file: /tmp/jaas-2036661b96e92290518.com. 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1-3 Full GC

```
24964 [concurrent mode failure]: 2149623E->2149623E(2149623E), 0.0265116 secs) 2456319E->2456233E(2456230E), [Metaspace: 75823E->75823E(113200E)], 0.0272735 secs) [Times: user=0.03 sys=0.00, real=0.03 secs]
24965 2021-05-17T22:47:03.104+0800: 26995.1451 [Full GC] (Allocation Failure) 2021-05-17T22:47:03.104+0800: 26995.1501 [CMS: 2149623E->2149623E(2149623E), 0.5646924 secs) 2456230E->2456230E(2456230E), [Metaspace:
75823E->75823E(113200E)], 0.5646924 secs) [Times: user=0.07 sys=0.00, real=0.07 secs]
24966 2021-05-17T22:47:03.474+0800: 26995.7181 [GC (CMS Initial Mark) (i CMS-initial-mark: 2149623E(2149623E))] 2456233E(2456230E), 0.0284012 secs) [Times: user=0.06 sys=0.00, real=0.03 secs]
24967 2021-05-17T22:47:03.703+0800: 26995.7471 [CMS-concurrent-mark-start]
24968 2021-05-17T22:47:03.707+0800: 26995.7501 [Full GC] (Allocation Failure) 2021-05-17T22:47:03.707+0800: 26995.7611 [CMS2021-05-17T22:47:03.947+0800: 26995.9911 [CMS-concurrent-mark: 0.236/0.244 secs) [Times: user=0.24 sys=0.00,
real=0.25 secs]
24969 [concurrent mode failure]: 2149623E->2149623E(2149623E), 0.0396635 secs) 2456319E->2456233E(2456230E), [Metaspace: 75823E->75823E(113200E)], 0.0403567 secs) [Times: user=0.04 sys=0.00, real=0.04 secs]
24970 2021-05-17T22:47:04.161+0800: 26996.5951 [Full GC] (Allocation Failure) 2021-05-17T22:47:04.162+0800: 26996.5991 [CMS: 2149623E->2149623E(2149623E), 0.5649924 secs) 2456230E->2456230E(2456230E), [Metaspace:
75823E->75823E(113200E)], 0.5649924 secs) [Times: user=0.08 sys=0.00, real=0.08 secs]
24971 2021-05-17T22:47:05.120+0800: 26997.1464 [GC (CMS Initial Mark) (i CMS-initial-mark: 2149623E(2149623E))] 2456233E(2456230E), 0.0284710 secs) [Times: user=0.05 sys=0.00, real=0.03 secs]
24972 2021-05-17T22:47:05.149+0800: 26997.1591 [CMS-concurrent-mark-start]
24973 2021-05-17T22:47:05.184+0800: 26997.2001 [Full GC] (Allocation Failure) 2021-05-17T22:47:05.187+0800: 26997.2001 [CMS2021-05-17T22:47:05.393+0800: 26997.6361 [CMS-concurrent-mark: 0.235/0.243 secs) [Times: user=0.25 sys=0.00,
real=0.26 secs]
```

## 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。

## 1.5.9 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。
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。

# 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。

#### 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。

#### 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。

#### 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 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`。

#### 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。

### 2.2.3 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 tbl using json options(path 'obs://....')
```

### 2.2.4 在 SQL 语句中如何设置局部变量

可以在 `select` 语句中通过 `where` 条件语句进行过滤，例如：

```
select * from table where part = '202012'
```

## 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。
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。
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。

## 2.3 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，请退出账号后重新登录。

### 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入数据报错。

## 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。
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，确认当前账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

当前账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，请退出账号后重新登录。

## 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”。
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。

- Spark Jar 作业:

Spark Jar 作业的归档日志文件夹以 “batch” 开头。

4. 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。
5. 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。

**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。


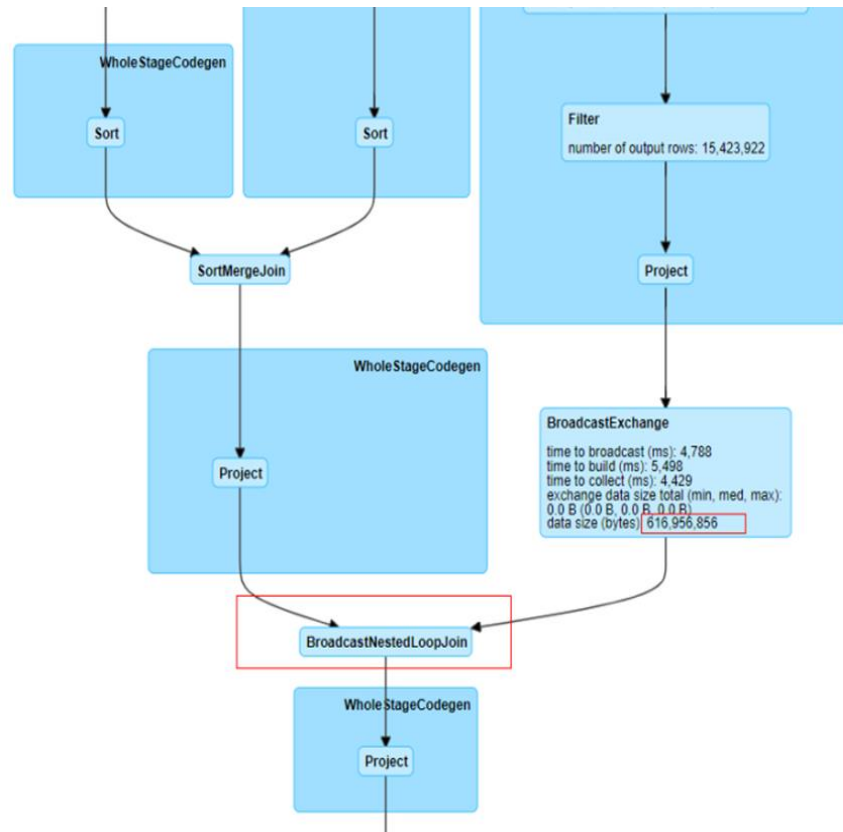
1. 登录 DLI 控制台，单击 “作业管理 > SQL 作业”。
2. 在 SQL 作业页面，在对应作业所在行，单击  按钮，查看作业详情，获取作业 ID。
3. 在对应作业的 “操作” 列，单击 “Spark UI”，进入 “Spark UI” 页面。
4. 在 “Spark UI” 页面，在上方菜单栏选择 “SQL”。参考下图，根据作业 ID，单击 Description 中的超链接。
5. 查看对应作业的 DAG 图，判断是否有 BroadcastNestedLoopJoin 节点。

图2-1 作业的 DAG 图。

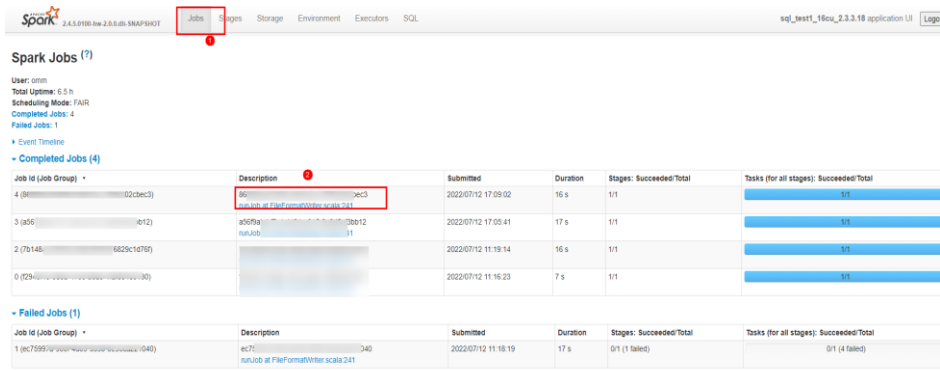


6. 如果存在广播，则参考 [SQL 作业中存在 join 操作](#)，因为自动广播导致内存不足，作业一直运行中处理。

## 可能原因 2：数据倾斜

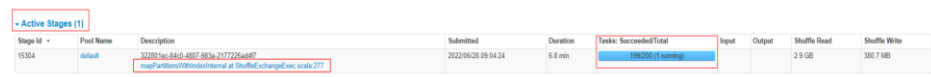
判断当前作业运行慢是否是数据倾斜导致：

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。
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。



Job ID (Job Group)	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
4 (8e0c02dec3)	sun.job at FileFormatWrite.scale.241	2022/07/12 17:09:02	16 s	1/1	1/1
3 (a56)	sun.job at FileFormatWrite.scale.241	2022/07/12 17:00:41	17 s	1/1	1/1
2 (79148)	sun.job at FileFormatWrite.scale.241	2022/07/12 11:19:14	16 s	1/1	1/1
0 (29)	sun.job at FileFormatWrite.scale.241	2022/07/12 11:16:23	7 s	1/1	1/1

5. 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。



Stage ID	Pool Name	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
15304	default	sun.job at FileFormatWrite.scale.241	2022/06/28 09:04:24	6.6 min	198/200 (1 min)			2.9 GB	382.7 MB

6. 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”。

7. 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问题。

参考图 2-2 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2-2 数据倾斜示例图



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)	Shuffle Size (Disk)
42	743708	0	RUNNING	PROCESS_LOCAL	54			2021/06/03 13:34:08	0 ms	21 s	52.9 GB / 47670629		0.0 B / 0	106.9 GB	52.1 GB
118	743784	0	SUCCESS	PROCESS_LOCAL	8			2021/06/03 13:34:08	27 ms		203.0 B / 1	4 ms	610.0 B / 1	0.0 B	0.0 B
36	743702	0	SUCCESS	PROCESS_LOCAL	104			2021/06/03 13:34:08	25 ms		370.0 B / 1	4 ms	369.0 B / 1	0.0 B	0.0 B
168	743834	0	SUCCESS	PROCESS_LOCAL	13			2021/06/03 13:34:08	24 ms		337.0 B / 1	4 ms	355.0 B / 1	0.0 B	0.0 B
120	743786	0	SUCCESS	PROCESS_LOCAL	80			2021/06/03 13:34:08	0.1 s		337.0 B / 1	4 ms	355.0 B / 1	0.0 B	0.0 B

### 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。

1. 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;
```

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2-3 查找 job id

```
)t;
[2021/03/19 10:02:59 GMT+0800] [INFO] dli.sql.badRecordsPath=obs://dli-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 [ht
[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”。
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2-4 数据倾斜样例

Description	Submitted	Duration	Tasks: Succeeded/Total
a48e2dfa-bf14-461e-8863-be29f578e3b6 mapPartitionsWithIndexInternal at ShuffleExchangeExec.scala:296 (kill)	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。

3. 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：

1. 使用创建表的用户账号登录到 DLI 管理控制台，选择“数据管理 > 库表管理”。
2. 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。
3. 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。
4. 单击“确定”完成权限授权。



5. 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”。
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：

```
nameid  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”。

## 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。
- 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>'2221-12-01 00:00:00';
```

## 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。

**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

# 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。

#### 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。

#### 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。

#### 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。

## 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

- 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()
```

#### 📖 说明

出于安全考虑，不建议在 obs 路径上带 AK/SK 信息。而且，如果是在 OBS 目录上建表，建表语句 path 字段给定的 obs 路径不能包含 AK/SK 信息。

## 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

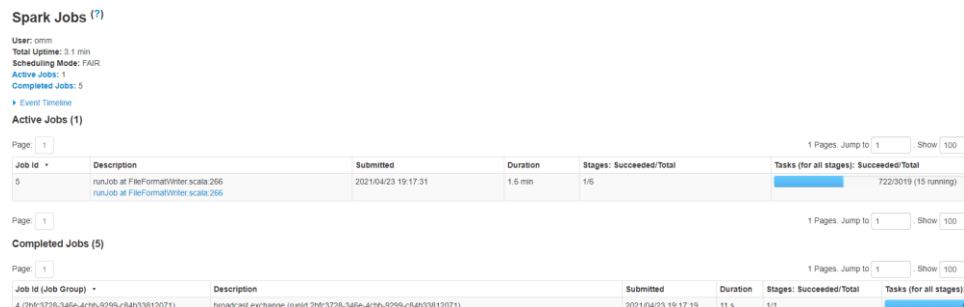
在创建 Spark 作业时，配置了“高级配置”中的参数，此处才会显示对应的内容。

### 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3-1 SparkUI



The screenshot shows the SparkUI interface with the following data:

Spark Jobs (7)							
User: omni							
Total Uptime: 3.1 min							
Scheduling Mode: FAIR							
Active Jobs: 1							
Completed Jobs: 5							
Event Timeline							
Active Jobs (1)							
Page: 1						1 Pages: Jump to: 1	Show: 100
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	1/5	722/3019 (15 running)		
Page: 1						1 Pages: Jump to: 1	Show: 100
Completed Jobs (5)							
Page: 1						1 Pages: Jump to: 1	Show: 100
Job Id (Job Group)	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages):		
4 (2bc3728-346e-4c0b-9299-c84033812071)	broadcast exchange (numid 2bc3728-346e-4c0b-9299-c84033812071)	2021/04/23 19:17:19	11 s	1/1			

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3-2 Driver 信息

spark.driver.cores	2
spark.driver.extraJavaOptions	-XX:CIcC XX:+Exit XX:+Use Dlog4j.co Dcarbon. Dsccon
spark.driver.extraLibraryPath	Bigdata/c
spark.driver.host	spark-4cc
spark.driver.memory	7G
spark.driver.port	7078
spark.driver.userClassPathFirst	false
spark.dynamicAllocation.cachedExecutorIdleTimeout	600s

图3-3 Executor 信息

spark.executor.cores	2
spark.executor.extraClassPath	hadoop/c
spark.executor.extraJavaOptions	-XX:CIcC XX:+Exit XX:+User Dlog4j.co Dcarbon.l Dsccon
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？

- 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：
  - 将 egg 包上传到指定的 OBS 桶路径下。
  - 登录 DLI 管理控制台，单击“数据管理 > 程序包管理”。
  - 在“程序包管理”页面，单击右上角“创建”可创建程序包。
  - 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.a.egg 包所在的 OBS 路径。
    - 分组设置和分组名称根据情况选择。
  - 单击“确定”完成程序包上传。
  - 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。
- 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。

通过 API 创建跨源连接请参考。

### 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完美运行他的程序呢？

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，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 第三方案程序库（支持 zip、egg 等）。

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3-4 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，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.stamte()
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。

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 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  
servicie 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](#)。

### 3.3.2 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。

### 3.3.3 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。

### 3.3.4 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。

### 3.3.5 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。

## 3.4 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、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 支持哪些数据格式

DLI 支持如下数据格式：

- Parquet
- CSV
- ORC
- Json
- Avro

### 4.1.3 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。

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。

### 4.1.4 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中，目前支持 CloudTable, CSS, DCS, DDS, DWS, MRS, RDS 等。

#### 4.1.5 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。

#### 4.1.6 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。


#### 4.1.7 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。

#### 4.1.8 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. 单击页面右上角的“My Quota”图标。系统进入“服务配额”页面。
3. 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？

目前系统暂不支持在线调整配额大小。如您需要调整配额，请拨打热线或发送邮件至客服，客服会及时为您处理配额调整的需求，并以电话或邮件的形式告知您实时进展。

在拨打热线或发送邮件之前，请您准备好以下信息：

- 帐号名，获取方式如下：



登录云帐户管理控制台，在右上角单击帐户名，选择“我的凭证”，在“我的凭证”页面获取“帐号名”。

- 配额信息，包括：服务名、配额类别、需要的配额值。

## 4.1.9 全局变量的使用中，一个子账号是否可以使用其他子账号创建的全局变量

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账号同项目下的用户共用。
- 如果同账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，保证同账号同项目下唯一，此时具备该全局变量修改权限的用户均可修改对应的变量值。
- 如果同账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 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。
- 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。

### 4.2.2 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 配额相关问题

## 5.1 怎样查看我的配额

1. 登录管理控制台。
2. 单击页面右上角的“My Quota”图标。  
系统进入“服务配额”页面。
3. 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。

## 5.2 如何申请扩大配额

### 如何申请扩大配额？

目前系统暂不支持在线调整配额大小。如您需要调整配额，请拨打热线或发送邮件至客服，客服会及时为您处理配额调整的需求，并以电话或邮件的形式告知您实时进展。

在拨打热线或发送邮件之前，请您准备好以下信息：

- 帐号名，获取方式如下：  
登录云帐户管理控制台，在右上角单击帐户名，选择“我的凭证”，在“我的凭证”页面获取“帐号名”。
- 配额信息，包括：服务名、配额类别、需要的配额值。

# 6 授权相关问题

## 6.1 使用咨询

### 6.1.1 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，您可以在云账号中给员工创建 IAM 用户，并使用策略来控制他们对云资源的访问范围。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，您只需要为您账号中的资源进行付费。

如果云账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 6-1 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6-1 DLI 系统权限

系统角色/策略名称	描述	类别
DLI FullAccess	数据湖探索所有权限。	系统策略
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
Tenant Administrator	租户管理员。 <ul style="list-style-type: none"> <li>操作权限：具有所有云服务的管理和使用权限。创建后，可通过 ACL 赋权给其他子用户使用。</li> <li>作用范围：项目级服务。</li> </ul>	系统角色
DLI Service Admin	DLI 服务管理员。 <ul style="list-style-type: none"> <li>操作权限：具有数据湖探索服务队列、数据的管理和使用权限。创建后，可通过 ACL 赋权给其他子用户使用。</li> <li>作用范围：项目级服务。</li> </ul>	系统角色

## 6.1.2 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。

## 6.2 运维指导

### 6.2.1 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。当前账号余额不足后，该账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。当前账号欠费后，该账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。

## 6.2.2 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。

## 6.2.3 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，是否配置该账号的查询权限。

## 6.2.4 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。

## 6.2.5 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。

## 6.2.6 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。

# 7 队列相关问题

## 7.1 使用咨询

### 7.1.1 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。

### 7.1.2 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。

### 7.1.3 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。

### 7.1.4 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。

## 7.2 运维指导

### 7.2.1 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

1. 在控制台搜索“云监控服务 CES”。
2. 进入 CES 后，在页面左侧“云服务监控”列表中，单击“数据湖探索”。
3. 选择队列进行查看。

## 7.2.2 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。

## 7.2.3 如何将老版本的 Spark 队列切换到通用型队列

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。

## 7.2.4 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。

## 7.2.5 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。

## 7.2.6 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。

## 7.2.7 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 跨源连接相关问题

## 8.1 跨源连接问题

### 8.1.1 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

### 8.1.2 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。

### 8.1.3 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: PSQLError: 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

## 8.1.4 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，建议您从高频原因往低频原因排查，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。

## 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。

## 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。

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。

## 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 8-1、图 8-2 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8-1 查看连接日志-2

对等连接ID	弹性资源池	连接状态
^ -	test	● 已失败

日志详情


```
Failed to get subnet for project id: [REDACTED], subnet id: [REDACTED]
[REDACTED], error code: 404, message: {"code":"VPC.0202","message":"Query resource by id [REDACTED] fail the subnet could not be found."}
```

图8-2 查看连接日志-3



## 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。

## 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。
2. 在 VPC-对等连接控制台查看对等连接信息。
3. 查看队列对应的 VPC 的路由表信息。

## 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。

## 8.1.5 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。

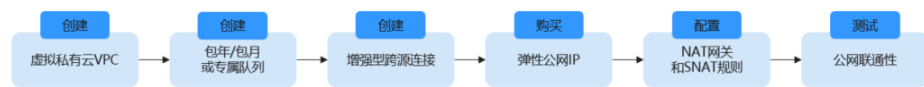
图8-3 增强型跨源连接配置流程



- **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8-4 配置 DLI 队列访问公网流程



## 8.1.6 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。

## 8.1.7 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。

## 8.2 跨源分析问题

### 8.2.1 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。

### 8.2.2 如何实现跨区域访问数据

1. 打通不同区域间的 VPC 网络。
2. 在 DLI 中创建增强型跨源链接，绑定队列。

3. 添加 DLI 的路由。

### 8.2.3 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。

## 8.3 跨源连接运维报错

### 8.3.1 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。
  - a. 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”。

- 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。

### 8.3.2 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。

详细操作请参考《数据湖探索用户指南》中的“修改主机信息”章节。

### 8.3.3 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。



## 8.3.4 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 emoji 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

## 8.3.5 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。

## 8.3.6 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。

## 8.3.7 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()。

## 8.3.8 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: PSQLErrorException: 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 cir 为 double 类型。

图8-5 创建源表



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

2. 目标表对应字段类型为 decimal(9,6)。

图8-6 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。

## 8.3.9 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。

# 9 使用 API 相关问题

## 9.1 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”。

## 9.2 不同账号的调用 API 时的项目 ID(Project ID)是固定的吗？

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

## 9.3 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。

# 10 使用 SDK 相关问题

---

## 10.1 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。

## 10.2 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。