数据湖探索

常见问题

文档版本 01

发布日期 2025-09-08

华为技术有限公司

商标声明


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

注意

您购买的产品、服务或特性等应受华为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 DLI 产品咨询类

1.1 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 1-1 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	• 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。
	上下游数据连接	• 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。

类型	特点	DLI Flink	MRS Flink
开发与运维	监控、告警	• 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	即开即用，Serverless架构，跨AZ容灾能力。 • 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。	需要一定的技术能力完成代码编译、集群搭建、配置、运维。 • 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
	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

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

DLI和MRS都支持Spark组件，但在服务模式、接口方式、应用场景和性能特性上存在一些差异。

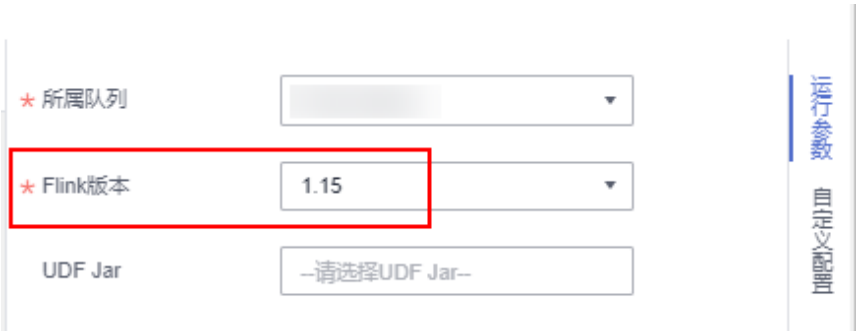
- DLI服务的Spark组件是全托管式服务，用户对Spark组件不感知，仅仅可以使用该服务，且接口为封装式接口。
  DLI的这种模式减轻了运维负担，可以更专注于数据处理和分析任务本身。
  具体请参考《**数据湖探索用户指南**》。
- MRS服务Spark组件的是建立在客户的购买MRS服务所分配的虚机上，用户可以根据实际需求调整及优化Spark服务，支持各种接口调用。
  MRS的这种模式提供了更高的自由度和定制性，适合有大数据处理经验的用户使用。
  具体请参考《**MapReduce服务开发指南**》。

1.3 怎样升级 DLI 作业的引擎版本

DLI提供了Spark和Flink计算引擎，为用户提供了一站式的流处理、批处理、交互式分析的Serverless融合处理分析服务，当前，Flink计算引擎推荐版本：Flink 1.15，Spark计算引擎推荐版本： Spark 3.3.1。

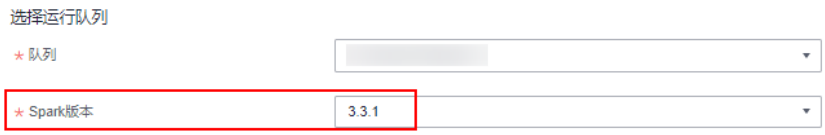
本节操作介绍如何升级作业的引擎版本。

- **SQL作业**:
  SQL作业不支持配置引擎版本，需要您重新新建队列执行SQL作业，新创建的队列会默认使用新版本的Spark引擎。
- **Flink OpenSource SQL作业:**
  a. 登录DLI管理控制台。
  b. 选择“作业管理 > Flink作业”，在作业列表中选择待操作的Flink OpenSource SQL作业。
  c. 单击操作列的“编辑”，进入作业编辑页面。
  d. 在右侧的“运行参数”配置区域，选择新的Flink版本。
  使用Flink 1.15以上版本的引擎执行作业时，需要在自定义配置中配置委托信息，其中key为"flink.dli.job.agency.name"，value为委托名，否则可能会影响作业运行。**了解自定义DLI委托**
  Flink 1.15语法参考请查看**Flink 1.15语法概览**。



- **Flink Jar作业**：
  a. 登录DLI管理控制台。
  b. 选择“作业管理 > Flink作业”，在作业列表中选择待操作的Flink Jar作业。
  c. 单击操作列的“编辑”，进入作业编辑页面。
  d. 在参数配置区域，选择新的Flink版本。
     使用Flink 1.15以上版本的引擎执行作业时，需要在优化参数中配置委托信息，其中key为"flink.dli.job.agency.name"，value为委托名，否则可能会影响作业运行。**了解自定义DLI委托**
     Flink 1.15语法参考请查看**Flink 1.15语法概览**。

- **Spark Jar作业**：
  a. 登录DLI管理控制台。
  b. 选择“作业管理 > Spark作业”，在作业列表中选择待操作的Spark Jar作业。
  c. 单击操作列的“编辑”，进入作业编辑页面。
  d. 在参数配置区域，选择新的Spark版本。
     使用Spark3.3以上版本的引擎执行作业时，需要Spark参数中配置自定义的委托名称，否则可能会影响作业运行。**了解自定义DLI委托**



- **了解更多**：
  - **FLINK 官网升级指导**
  - **FLINK 1.15 release note**

1.4 DLI 的数据可存储在哪些地方

DLI 支持存储哪些格式的数据?

DLI支持如下数据格式：

- Parquet
- CSV
- ORC
- Json

- Avro

DLI 服务的数据可以存储在哪些地方?

- 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。

DLI 表与 OBS 表有什么区别?

- 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。

1.5 DLI 是否支持导入其他租户共享 OBS 桶的数据?

DLI支持将同一个租户下子账户共享OBS桶中的数据导入，但是租户级别共享OBS桶中的数据无法导入。

DLI不支持导入其他租户共享的OBS桶中的数据，主要是为了确保数据的安全性和数据隔离。

对于需要跨租户共享和分析数据的场景，建议先将数据脱敏后上传到OBS桶中，再进行数据分析，分析完成后及时删除OBS桶中的临时数据，以确保数据安全

1.6 区域和可用区

什么是区域、可用区?

区域和可用区用于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1-1阐明了区域和可用区之间的关系。

图 1-1 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**。

1.7 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以修改对应的变量值。
- 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。

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

不存在。

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）。

该漏洞主要影响在启用了ACL（访问控制列表）时，可以通过提供任意用户名来执行命令导致数据安全受到威胁。

DLI在设计时充分考虑了数据安全和数据隔离，因此没有启用相关的配置项，所以不会受到这个漏洞的影响。

1.9 怎样管理在 DLI 上运行的作业

管理大量的DLI作业时您可以采用以下方案：

- **作业分组：**

  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。
- **创建IAM子用户**

  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**》。

此外DLI还提供了作业管理功能，包括编辑、启动、停止、删除作业，以及导出和导入作业。您可以利用这些功能来定期维护和管理作业。

1.10 怎样修改 DLI 上已经创建好的表的字段名称?

DLI本身不支持直接修改表的字段名称，但您可以通过以下步骤来解决这个问题表数据迁移的方式来解决该问题：

1. 创建新表：创建一个新表，并定义新的表字段名称。
2. 迁移数据：使用INSERT INTO ... SELECT语句将旧表的数据迁移到新表中。
3. 删除旧表：在确保新表完全替代旧表并且数据迁移完成后，您可以删除旧表以避免表数据混淆。

2 DLI 弹性资源池和队列类

2.1 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:**

  a. 登录DLI管理管理控制台。
  b. 选择“作业管理 > Flink作业”。
  c. 单击作业名称进入作业详情页面。
  d. 选择“作业配置信息 > 资源配置”

e. 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 2-1 查看 Flink 作业所需 CUs 数

- **Spark作业：**
  a. 登录DLI管理管理控制台。
  b. 选择“作业管理 > Spark作业”。
  c. 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 2-2 查看 Spark 作业所需 CUs 数

2.2 怎样判断当前 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 2-3 查看队列监控指标

2.3 怎样查看 DLI 队列负载?

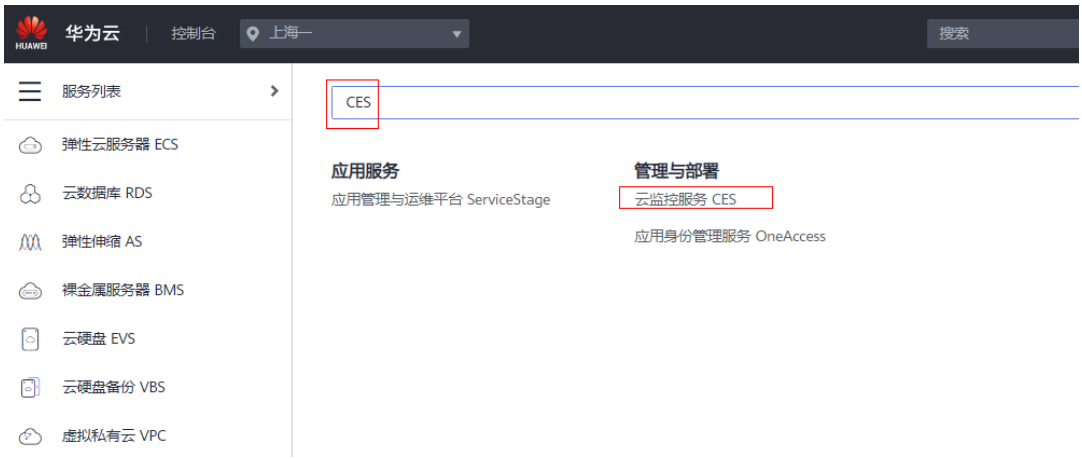
场景概述

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

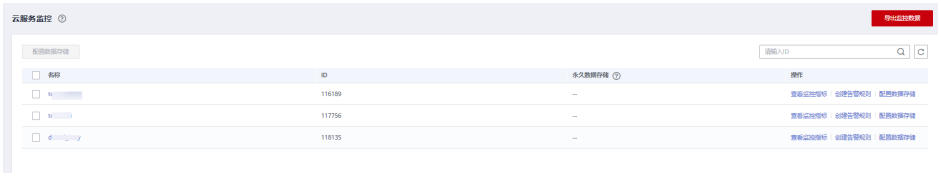
1. 在控制台搜索“云监控服务 CES”。

图 2-4 搜索 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”。

图 2-5 云服务监控



3. 选择队列进行查看。

图 2-6 查看队列负载

2.4 怎样监控 DLI 队列上的作业异常?

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**》。

2.5 怎样将老版本的 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。

2.6 在 default 队列执行 DLI SQL 失败，提示超时异常怎么办?

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。

了解更多新建队列的操作指导请参考**创建弹性资源池并添加队列**。

3 DLI 数据库和表类

3.1 为什么在 DLI 控制台中查询不到表?

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。

3.2 OBS 表压缩率较高怎么办?

当您在提交导入数据到DLI表的作业时，如果遇到Parquet/Orc格式的OBS表对应的文件压缩率较高，超过了5倍的压缩率，您可以通过调整配置来优化作业的性能。

具体方法：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。

但调整这个参数可能会影响到作业的执行效率和资源消耗，因此在做调整时需要根据实际的数据量和压缩率来选择适合的参数值。

3.3 字符码不一致导致数据乱码怎么办?

在DLI执行作业时，为了避免因字符编码不一致导致的乱码问题，建议您统一数据源的编码格式。

DLI服务只支持UTF-8文本格式，因此在执行创建表和导入操作时，用户的数据需要以UTF-8编码。

在将数据导入DLI之前，确保源数据文件（如CSV、JSON等）是以UTF-8编码保存的。如果数据源不是UTF-8编码，请在导入前提前转换为UTF-8编码。

3.4 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吗?

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。

3.5 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怎么办?

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。

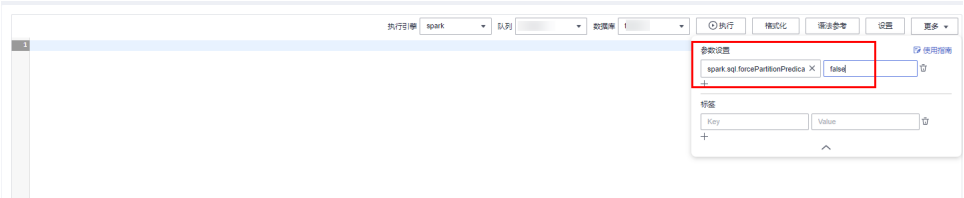
问题根因

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 3-1 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。

3.6 创建 OBS 外表，由于 OBS 文件中的某字段存在换行符导致表字段数据错误怎么办？

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);
```

3.7 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，作业运行失败怎么办?

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)
```

3.8 手动在 OBS 表的分区目录下添加了数据，但是无法查询到数据怎么办?

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。

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

当您使用insert overwrite语句覆盖分区表的数据时，如果发现它覆盖了全量数据而不是预期的分区数据，这可能是因为动态分区覆盖功能没有被启用。

如果需要动态覆盖DataSource表指定的分区数据，您需要先配置参数dli.sql.dynamicPartitionOverwrite.enabled=true，然后通过insert overwrite语句实现。

“dli.sql.dynamicPartitionOverwrite.enabled”默认值为“false”，在不配置时它会覆盖整张表的数据。

详细说明请参考**插入数据**。

3.10 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';
```

TIMESTAMP类型详细可参考**TIMESTAMP数据类型**。

3.11 SQL 作业执行完成后，修改表名导致 datasize 不正确怎么办?

在执行SQL作业后立即修改表名，可能会导致表的数据大小结果不正确。

这是因为DLI在执行SQL作业时，会对表进行元数据更新，如果在作业执行完成前修改了表名，会和作业的元数据更新过程冲突，从而影响对数据大小的判断。

为了避免这种情况，建议在SQL作业执行完成后，等待5分钟后再修改表名。确保系统有足够的时间更新表的元数据，避免因修改表名而导致的数据大小统计不准确的问题。

3.12 从 DLI 导入数据到 OBS，数据量不一致怎么办?

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 3-2 检查读取的数据量

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**》。

3.13 为什么创建 Hudi 表没有在 DLI 控制台上显示?

问题描述

使用Flink SQL语句创建的Hudi表未在DLI控制台显示无法通过控制台对该表进行管理和查询操作。

根因分析

DLI控制台的数据管理默认使用的是Hive Catalog，而Hudi表的元数据没有正确同步到Hive Catalog中。Hudi表的元数据管理是独立于Hive的，因此需要额外的配置来确保Hudi表的元数据能够被Hive Catalog识别和同步。

解决方案

Hudi表持久化是指将Hudi表的元数据和数据信息同步并保存到Hive的元数据存储中，以便 Hive 能够识别和管理这些表。

- **SQL作业**

  如果仅使用SQL来操作Hudi数据，通常不需要配置额外的“持久化参数”，因为SQL作业本身已经提供了对Hudi数据的读写支持。

- **Flink OpenSource SQL作业**

  为了将Hudi表的元数据持久化到DLI的数据管理（Hive Catalog），需要在创建Hudi表时添加一些特定的配置参数，这些参数用于通知Hudi将元数据同步到Hive Catalog。

  详细参数说明如**表3-1**所示。

  在DLI使用Hudi提交Flink SQL作业且配置，配置hive_sync相关参数，实时同步元数据至由DLI的操作示例请参考**在DLI使用Hudi提交Flink SQL作业**。

表 3-1 创建 Hudi 结果表时配置的持久化参数

参数名称	参数描述
hive_sync.enable	是否启用向Hive同步表信息。 当你需要将Hudi表的元数据同步到 Hive元数据存储中，以便通过Hive查询工具或在DLI控制台的数据管理中访问Hudi表时，需要将此参数设置为 true。 • true：Hudi 将会把表的元数据（如表结构、分区信息等）同步到 Hive 中。 • false：不会向Hive同步表信息。 开启向hive同步表信息后会使用catalog相关权限，还需配置访问catalog的委托权限。
hive_sync.mode	Hudi表元数据同步到Hive的方式 根据你的环境和需求选择合适的同步模式。 • **jdbc**：通过JDBC连接到 Hive Server来同步元数据。 • **hms**：通过Hive Meta Client 直接与Hive Metastore交互来同步元数据。 • **hiveql**：通过执行Hive QL语句来同步元数据。
hive_sync.table	同步到Hive的表名称，Hudi表的元数据将同步到这个Hive表中
hive_sync.db	同步到Hive的数据库名称，Hudi表的元数据将同步到这个Hive数据库中的表。

参数名称	参数描述
hive_sync.support_timestamp	是否支持时间戳，用于确保Hudi表的元数据同步到Hive时能够正确处理时间戳字段。 建议配置为True。

- **Spark jar作业**

  在Spark jar中操作Hudi数据时，也需要配置一些参数来实现持久化和同步到Hive。

  详细参数说明如**表3-2**所示。

  在DLI使用Hudi提交Spark Jar作业且开启作业同步配置的示例请参考**在DLI使用Hudi提交Spark Jar作业**。

  表 3-2 Spark Jar 作业同步 Hive 表参数配置

参数	描述	默认值
hoodie.datasource.hive_sync.enable	是否同步hudi表信息到Hive。当使用DLI提供的元数据服务时，配置该参数代表同步至DLI的元数据中。 建议该值设置为true，统一使用元数据服务管理hudi表。	true
hoodie.datasource.hive_sync.database	要同步给Hive的数据库名。	default
hoodie.datasource.hive_sync.table	要同步给Hive的表名，建议和hoodie.datasource.write.table.name的值保证一致。	unknown
hoodie.datasource.hive_sync.partition_fields	用于确定Hive分区列。	""
hoodie.datasource.hive_sync.partition_extractor_class	用于提取hudi分区列值，将其转换成Hive分区列。	org.apache.hudi.hive.SlashEncodedDayPartitionValueExtractor
hoodie.datasource.hive_sync.support_timestamp	当hudi表存在timestamp类型字段时，需指定此参数为true，以实现同步timestamp类型到hive元数据中。 该值默认为false，默认将timestamp类型同步为bigInt，默认情况可能导致使用sql查询包含timestamp类型字段的hudi表出现错误。	true

参数	描述	默认值
hoodie.datasource.hive_sync.fast_sync	Hudi同步Hive分区方式： ● true：从最近一次hive同步后所修改的分区直接向Hive表中做add partition if not exist操作。 ● false：会根据修改的分区去hive表查询是否已存在，不存在的进行添加。	true
hoodie.datasource.hive_sync.mode	hudi表同步Hive表的方式，在使用DLI提供的元数据服务时需要保证为hms： ● hms：通过hive metastore client同步元数据。 ● jdbc：通过hive jdbc方式同步元数据。（DLI元数据服务不支持） ● hiveql：执行hive ql方式同步元数据。（DLI元数据服务不支持）	hms
hoodie.datasource.hive_sync.username	使用jdbc方式同步Hive时，指定的用户名。	hive
hoodie.datasource.hive_sync.password	使用jdbc方式同步Hive时，指定的密码。	hive
hoodie.datasource.hive_sync.jdbcurl	连接hive jdbc指定的连接。	""
hoodie.datasource.hive_sync.use_jdbc	是否使用Hive jdbc方式连接Hive同步Hudi表信息。建议该值设置为false，设置为false后jdbc连接相关配置无效。	true

4 增强型跨源连接类

4.1 增强型跨源连接绑定队列失败怎么办?

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

步骤1 登录DLI管理控制台，选择“全局配置 > 服务授权”。

步骤2 在委托设置页面，按需选择所需的委托权限。

其中“DLI Datasource Connections Agency Access”是跨源场景访问和使用VPC、子网、路由、对等连接的权限。

了解更多DLI委托权限请参考**DLI委托权限**。

步骤3 选择dli_management_agency需要包含的权限后，并单击“更新委托权限”。

图 4-1 更新委托权限

步骤4 委托更新完成后，重新创建跨源连接和运行作业。

----结束

4.2 DLI 增强型跨源连接 DWS 失败怎么办?

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: PSQLException: 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**》。

4.3 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DLI Datasource Connections Agency Access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 4-2 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 4-3 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4-4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 4-4 查看连接日志-1

检查是否为 DLI 授权了 DLI Datasource Connections Agency Access 权限

您可以从连接日志判断是否是由于权限不足导致的连接失败。

如图4-5、图4-6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DLI Datasource Connections Agency Access授权。

了解DLI更新委托权限。

图 4-5 查看连接日志-2

图 4-6 查看连接日志-3

图 4-7 DLI 服务授权

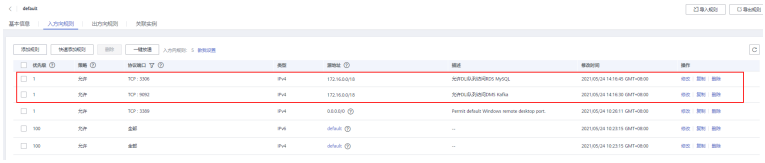
检查对端安全组是否放通队列的网段

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 4-8 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 4-9 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 4-10 查看对等连接

图 4-11 查看队列网段

3. 查看队列对应的VPC的路由表信息。

图 4-12 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。

4.4 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连通**

  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。

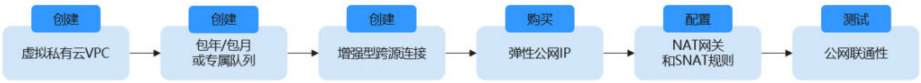
  DLI提供的增强型跨源连接功能，底层采用对等连接的方式打通与目的数据源的vpc网络，通过点对点的方式实现数据互通。

图 4-13 增强型跨源连接配置流程

- **配置DLI队列与公网网络连通**

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 4-14 配置 DLI 队列访问公网流程



4.5 为什么 DLI 增强型跨源连接要创建对等连接?

DLI增强型跨源连接创建对等连接的主要原因是为了实现DLI与不同VPC中的数据源之间的网络连通。

当DLI需要访问外部数据源，而这些数据源位于不同的VPC中时，由于网络隔离，DLI默认无法直接读取这些数据源的数据。通过创建增强型跨源连接，可以采用对等连接的方式打通DLI与数据源的VPC网络，从而实现数据的互通和跨源分析。

增强型跨源连接的优势：

- 网络连通性：直接打通DLI与目的数据源的VPC网络实现数据互通。
- 支持多种数据源：支持DLI与多种数据源的网络连通，例如DWS，RDS，CSS，DCS等数据源。

4.6 DLI 创建跨源连接，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。

4.7 新建跨源连接，显示已激活，但使用时提示 communication link failure 错误怎么办?

根因分析

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。

解决方案

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

1. 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 4-15 创建经典型跨源连接-RDS

2. 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 4-16 检查网段及端口是否配置

       如果不存在，则进行添加。

       图 4-17 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 4-18 检查网段及端口是否配置。

  如果不存在，则进行添加。

  图 4-19 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。
  i. 在DLI服务找到跨源连接绑定队列对应的网段

  图 4-20 查找跨源连接绑定队列对应的网段

  ii. 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 4-21 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 4-22 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。

4.8 跨源访问 MRS HBase，连接超时，日志未打印错误怎么办?

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。

建议您重新配置主机信息后再重试访问MRS HBase。

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：“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。

4.9 DLI 跨源连接报错找不到子网怎么办?

问题现象

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."}
```

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DLI跨源连接报错找不到子网。

处理步骤

步骤1 登录DLI管理控制台，选择“全局配置 > 服务授权”。

步骤2 在委托设置页面，按需选择所需的委托权限。

其中“DLI Datasource Connections Agency Access”是跨源场景访问和使用VPC、子网、路由、对等连接的权限。

了解更多DLI委托权限请参考**DLI委托权限**。

步骤3 选择dli_management_agency需要包含的权限后，并单击“更新委托权限”。

图 4-23 更新委托权限

步骤4 委托更新完成后，重新创建跨源连接和运行作业。

----结束

4.10 跨源 RDS 表，执行 insert overwrite 提示 Incorrect string value 错误怎么办？

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

4.11 创建 RDS 跨源表提示空指针错误怎么办?

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**》。

4.12 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。

4.13 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()。

4.14 通过 DLI 跨源写 DWS 表，非空字段出现空值异常怎么办?

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: PSQLException: 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

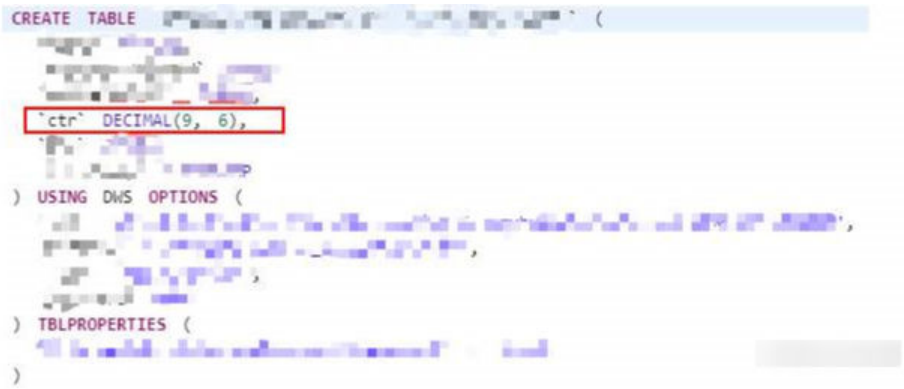
1. DLI源表对应字段cir为double类型。

   图 4-24 创建源表

2. 目标表对应字段类型为decimal(9,6)。

   图 4-25 创建目标表

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。

4.15 更新跨源目的端源表后，未同时更新对应跨源表，导致 insert 作业失败怎么办?

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。

4.16 RDS 表有自增主键时怎样在 DLI 插入数据?

在DLI中创建关联RDS表时，如果RDS表包含自增主键或其他自动填充字段，您在DLI中插入数据时可以采取以下措施：

1. 插入数据时省略自增字段：在DLI中插入数据时，对于自增主键字段或其他自动填充的字段，您可以在插入语句中省略这些字段。数据库会自动为这些字段生成值。例如，如果表中有一个名为id的自增主键字段，您可以在插入数据时不包含这个字段，数据库会自动为新插入的行分配一个唯一的id值。
2. 使用NULL值：如果您需要在插入数据时明确指定某些字段由数据库自动填充，可以在这些字段的位置填写NULL。这样，数据库会识别到这些字段应该由系统自动生成值，而不是由用户指定。

5 SQL 作业类

5.1 SQL 作业开发类

5.1.1 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。

5.1.2 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。

推荐使用临时表进行数据中转 自读自写在突发异常场景下存在数据丢失的风险

执行SQL：

```
INSERT OVERWRITE TABLE tablename
select  * FROM  tablename
DISTRIBUTE BY floor(rand()*20)
```

5.1.3 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**》。

5.1.4 创建 OBS 表时怎样指定 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'
```

典型错误示例分析

```
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。

5.1.5 关联 OBS 桶中嵌套的 JSON 格式数据如何创建表

如果需要关联OBS桶中嵌套的JSON格式数据，可以使用异步模式创建表。

以下是一个示例的建表语句，展示了如何使用 JSON 格式选项来指定 OBS 中的路径：

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

- using json：指定使用 JSON 格式。
- options：用于设置表的选项。
- path：指定OBS中JSON文件的路径。

5.1.6 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
```

5.1.7 怎样将一个区域中的 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。
3. 根据需要导入或使用对应的数据。

5.1.8 SQL 作业如何指定表的部分字段进行表数据的插入

如果需要将数据插入到表中，但只想指定部分字段，可以使用INSERT INTO语句结合SELECT子句来实现。

但是DLI目前不支持直接在INSERT INTO语句中指定部分列字段进行数据插入，您需要确保在SELECT子句中选择的字段数量和类型与目标表的Schema信息匹配。即确保源表和目标表的数据类型和列字段个数相同，以避免插入失败。

如果目标表中的某些字段在SELECT子句中没有被指定，那么这些字段也可能被插入默认值或置为空值（取决于该字段是否允许空值）。

5.1.9 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”。

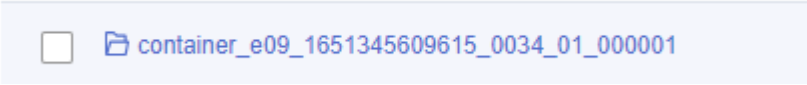
图 5-1 归档日志

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 5-2 带有 driver 的归档日志文件夹名示例

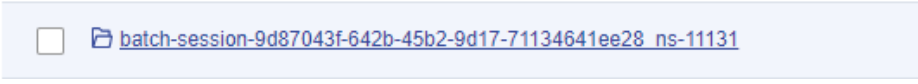
     

     图 5-3 container_*xxx*_000001 归档日志文件夹示例

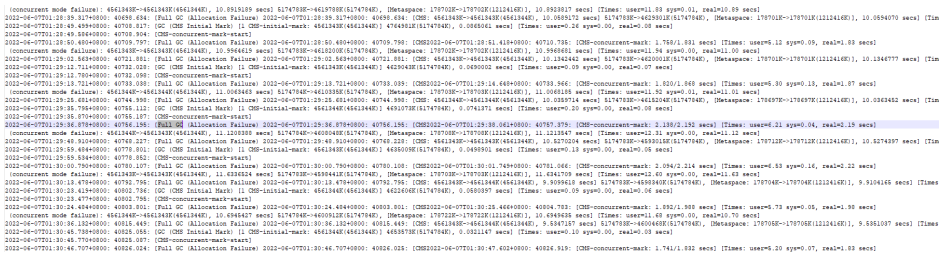
   - Spark Jar作业：
     Spark Jar作业的归档日志文件夹以“batch”开头。

     图 5-4 Spark Jar 作业归档日志文件夹名示例

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。

图 5-5 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。

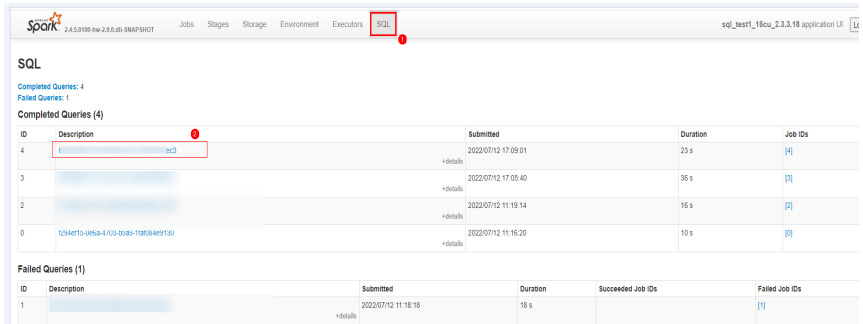
原因2 广播表：广播也可能会造成Driver内存的FullGC。

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 5-6 获取作业 ID

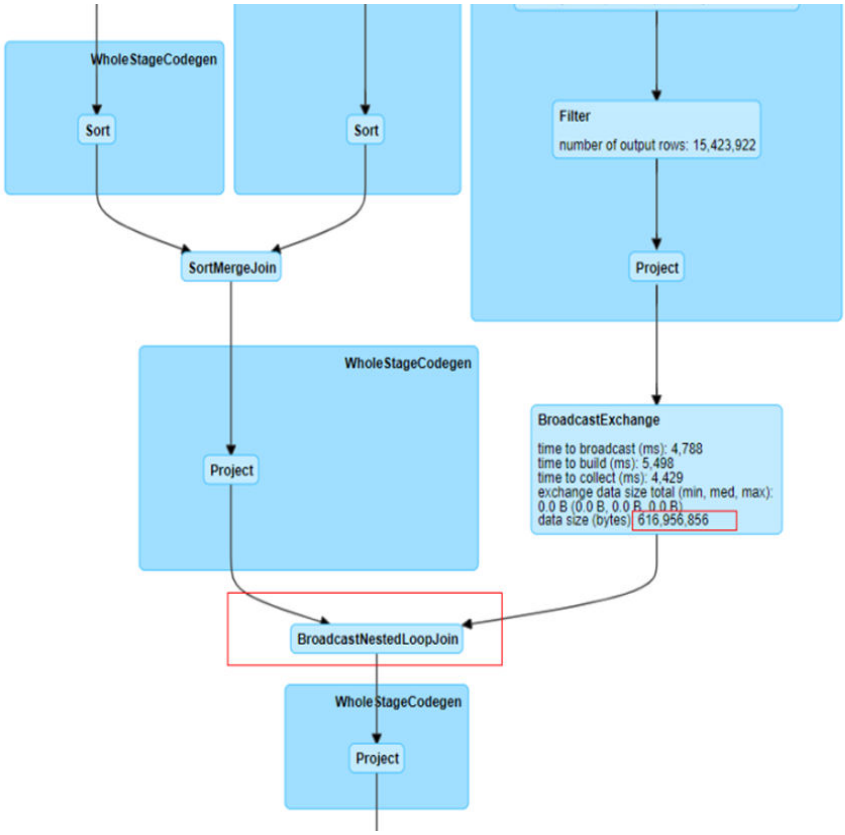
3. 在对应作业的“操作”列，单击“Spark UI”，进入“Spark UI”页面。
4. 在“Spark UI”页面，在上方菜单栏选择“SQL”。参考下图，根据作业ID，单击Description中的超链接。

图 5-7 单击作业链接



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

图 5-8 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。

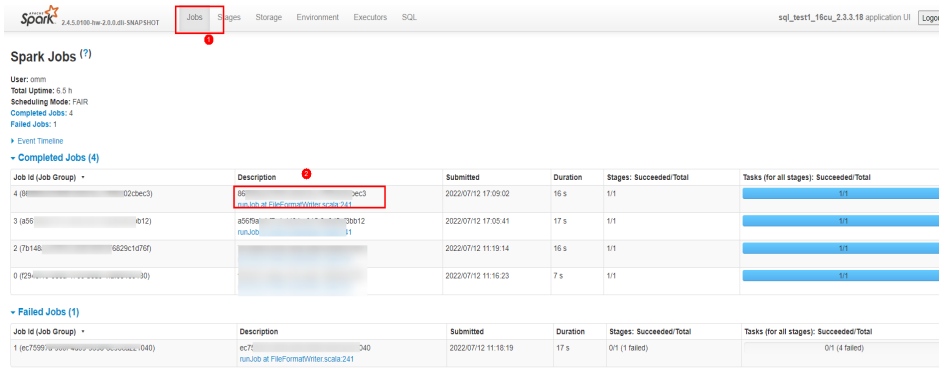
可能原因 2：数据倾斜

判断当前作业运行慢是否是数据倾斜导致：

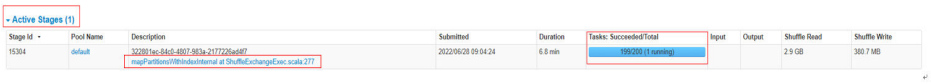
1. 登录DLI控制台，单击“作业管理 > SQL作业”。
2. 在SQL作业页面，在对应作业所在行，单击 按钮，查看作业详情信息，获取作业ID。

图 5-9 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。



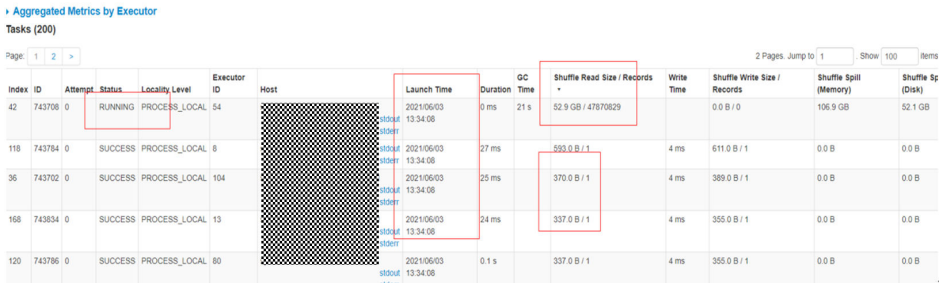
5. 根据Active Stage可以看到当前正在运行的Stage运行情况，单击Description中的超链接。



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5-10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 5-10 数据倾斜示例图



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;
```

   从图5-11可以看出，num为1的数量远大于其他值的数量。

图 5-11 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**执行SQL作业时产生数据倾斜怎么办?** 处理。

5.1.10 怎样查看 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 5-12 查看日志

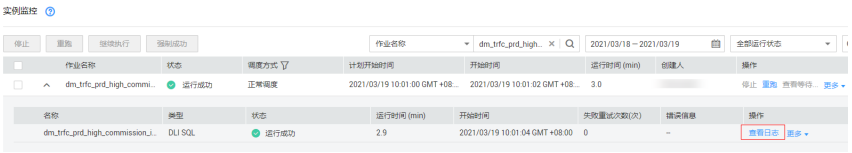
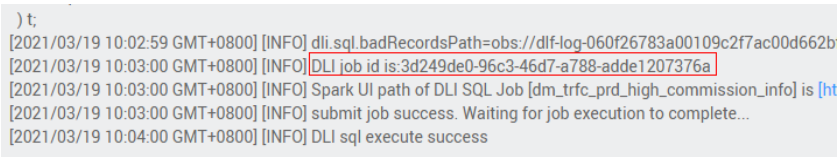


图 5-13 查找 job id



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 5-14 归档日志

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。

5.1.11 怎样查看 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。

5.1.12 执行 SQL 作业时产生数据倾斜怎么办?

什么是数据倾斜?

数据倾斜是在SQL作业执行中常见的问题，当数据分布不均匀的情况下，一部分计算节点处理的数据量远大于其他节点，从而影响整个计算过程的处理效率。

例如观察到SQL执行时间较长，进入SparkUI查看对应SQL的执行状态，如图5-15所示，查看到一个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 5-15 数据倾斜样例

常见数据倾斜场景

- Group By聚合倾斜

  在执行Group By聚合操作时，如果某些分组键对应的数据量特别大，而其他分组键对应的数据量很小，在聚合过程中，数据量大的分组会占用更多的计算资源和时间，导致处理速度变慢，出现数据倾斜。

- JOIN 操作倾斜

  在执行表JOIN操作时，参与JOIN的键在某个表中分布极不均匀，导致大量数据集中在少数几个任务中处理，而其他任务则已完成，造成数据倾斜。

Group By 数据倾斜解决方案

取部分数据执行select count(*) as sum,Key from tbl group by Key order by sum desc查询具体是哪些key引起的数据倾斜。

然后对于倾斜Key单独做处理，加盐让其先将他分为多个task分别统计，最后再对分开统计结果进行结合统计。

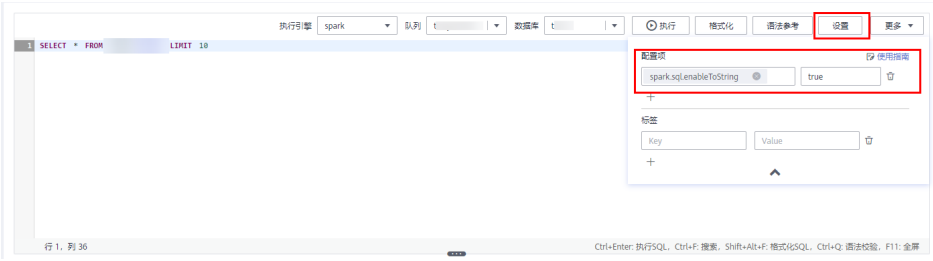
例如：如下SQL示例，假设已知倾斜key为'Key01',导致单个task处理大量数据，做如下处理：

```
SELECT
  a.Key,
  SUM(a.sum) AS Cnt
```

```
FROM
  (
    SELECT
      Key,
      count(*) AS sum
    FROM
      tbl
    GROUP BY
      Key,
      CASE
        WHEN KEY = 'Key01' THEN floor(random () * 200)
        ELSE 0
      END
  ) a
GROUP BY
  a.Key;
```

JOIN 数据倾斜解决方案

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。

5.1.13 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。

5.1.14 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。

5.2 SQL 作业运维类

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

该提示信息说明您将数据导出到一个已经存在的OBS路径。

解决方案：

- 新建OBS目录。
  您可以新建一个不存在的OBS目录用于存储导出的数据。
- 删除已存在的OBS目录。
  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。
- 检查目录权限
  确保您已具备访问和写入该OBS路径的权限。如果权限缺失可以联系管理员添加对应的OBS桶权限。

5.2.2 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;
```

5.2.3 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.

该提示信息说明您可能因账户欠费或余额不足导致操作受限。

解决方案：

1. 检查账户状态。
   请先确认是否欠费，如有欠费请充值。
2. 重新登录账户。
   如果充值后仍然提示相同的错误，请退出账号后重新登录。

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

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。

解决方案

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。

5.2.5 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

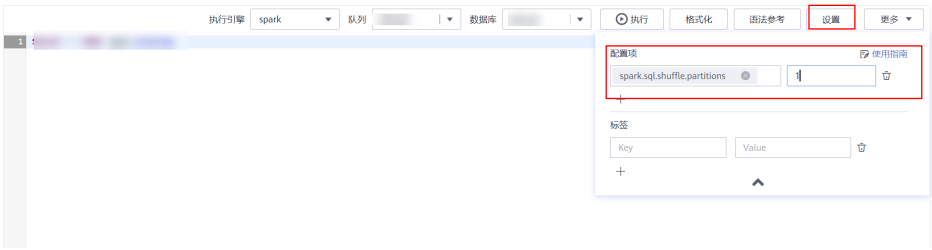
上述报错可能原因是当前导入的文件数据量较大，同时因为spark.sql.shuffle.partitions参数设置的并行度过大，导致缓存区大小不够而导入数据报错。

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。

   图 5-16 设置参数



3. 重新执行作业。

5.2.6 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。

5.2.7 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。

您可以通过“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。

5.2.8 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。

5.2.9 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. 选择“资源管理 > 弹性资源池”，选择已购买的弹性资源池，单击操作列的“添加队列”，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。

5.2.10 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。

可能原因

- 可能由于文件路径错误或文件不存在导致系统无法找指定文件路径或文件。
- 文件被占用。

解决措施

- 检查文件路径、文件名。
  检查文件的路径是否正确，包括目录名称和文件名。
- 文件被占用
  文件被占用导致的文件报错找不到，一般是读写冲突产生的，建议查询SQL查询报错表的时候，是否有作业正在覆盖写对应数据。

5.2.11 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

请检查OBS桶权限，确保账号有权限访问报错信息中提到的OBS桶。

如果没有，需要联系OBS桶的管理员添加桶的访问权限。

5.2.12 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

请检查OBS桶权限，确保账号有权限访问报错信息中提到的OBS桶。

如果没有，需要联系OBS桶的管理员添加桶的访问权限。

5.2.13 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。

解决方案

1. 检查账户状态。
   请先确认是否欠费，如有欠费请充值。
2. 重新登录账户。
   如果充值后仍然提示相同的错误，请退出账号后重新登录。

5.2.14 预览 SQL 作业结果报错：预览结果内容超过预览阈值

问题现象

执行SQL作业，因查询结果过大导致作业结果预览失败。

根因分析

获取作业结果失败，抛出runException。已解析输入的长度超过了解析器设置中定义的最大字符数限制。

解决方案

您可以导出作业结果到指定的桶，再从OBS桶下载作业结果。

导出查询结果的操作入口有两个，分别在“SQL作业”和“SQL编辑器”页面。

- 在“作业管理”>“SQL作业”页面，可单击对应作业“操作”列“更多 > 导出结果”，可导出执行查询后的结果。
- 在“SQL编辑器”页面，查询语句执行成功后，在“查看结果”页签右侧，单击“导出结果”，可导出执行查询后的结果。

说明

- 如果查询结果中无数值列，则无法导出查询结果。
- 确保执行导出作业结果的用户具备该OBS桶的读写权限。

6 Flink 作业类

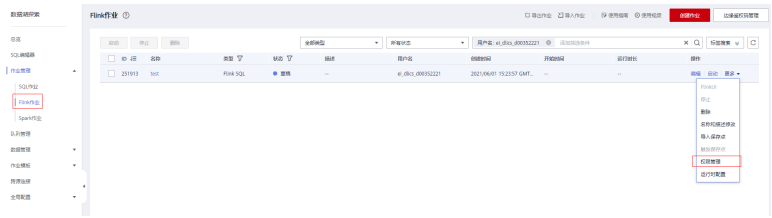
6.1 Flink 作业咨询类

6.1.1 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：

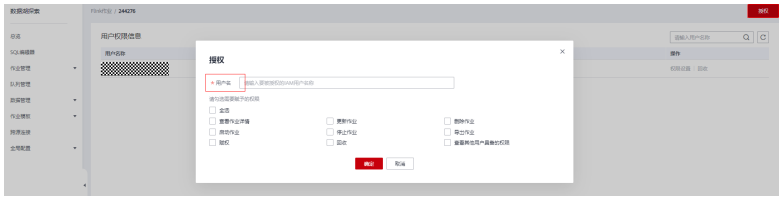
- DLI授权管理
  a. 使用租户账号，或者作业owner账号，或有DLI Service Administrator权限的账号，登录DLI控制台。
  b. 在“作业管理”>“Flink作业”页面找到对应的作业。
  c. 在对应作业的“操作”栏中选择“更多”>“权限管理”。

  图 6-1 Flink 作业权限管理

  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。

  图 6-2 授权

- IAM授权管理

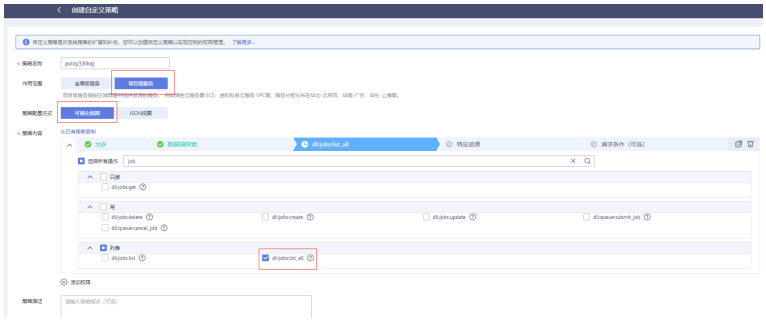
a. 登录统一身份认证IAM控制台，在“权限”页面，单击“创建自定义权限”。

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。

图 6-3 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 Administrator”（权限较大，拥有DLI所有权限）也可以实现Flink作业的查看。

6.1.2 Flink 作业怎样设置“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6-4**所示。

图 6-4 Flink SQL 作业编辑界面

6.1.3 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**》。

6.1.4 Flink 作业管理界面对用户进行授权时提示用户不存在怎么办?

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提示“No such user. userName:xxxx.”错误。

解决方案

以上问题可能是由于系统未能识别新用户信息。

请按以下步骤进行排查：

1. 请先确认下当前用户名是否存在。
2. 如该用户存在，请重新登录管理控制台，系统才能对该用户进行授权操作。

6.1.5 手动停止了 Flink 作业，再次启动时怎样从指定 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作业怎样从指定Checkpoint恢复。

解决方案

由于Flink Checkpoint和Savepoint生成机制及格式一致，因此可以通过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。

6.1.6 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。

如果遇到提示SMN主题不存在您可以按照以下步骤进行排查：

1. 确认SMN主题是否已经创建。

   如果未创建，请在SMN服务管理控制台创建一个新的主题。

   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。
2. 检查IAM权限。

   如果SMN主题已经存在，但仍然提示不存在，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，确保该用户组已添加相应Region的SMN策略。

3. 确认主题名称和区域。

   确保您在DLI中配置的SMN主题名称和区域与实际创建的SMN主题一致。如果SMN主题名称不一致也会导致系统提示SMN主题不存在。

6.2 Flink SQL 作业类

6.2.1 怎样将 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，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;
   ```

6.2.2 Flink SQL 作业 Kafka 分区数增加或减少，怎样不停止 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。

6.2.3 在 Flink SQL 作业中创建表使用 EL 表达式，作业运行提示 DLI.0005 错误怎么办？

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*}**。

修改前：

```
t_user_message_input_#{date_format(date_sub(current_date(), 1), 'yyyymmddhhmmss')}
```

修改后：

```
t_user_message_input_${date_format(date_sub(current_date(), 1), 'yyyymmddhhmmss')}
```

修改后，Flink SQL作业能够正确解析表名，并根据EL表达式动态生成表名。

6.2.4 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 6-5 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;
```


6.2.5 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。

6.2.6 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

6.2.7 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。

6.2.8 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?

问题描述

Flink Opensource SQL从RDS数据库读取的时间和RDS数据库存储的时间为不一致

根因分析

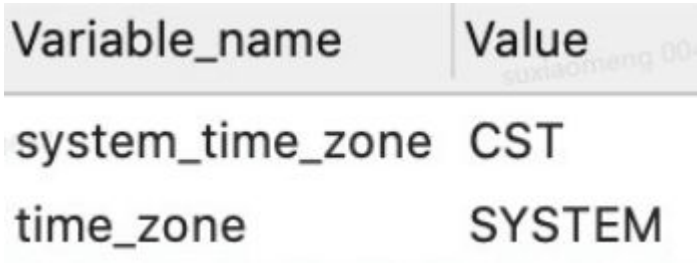
该问题的根因是数据库设置的时区不合理，通常该问题出现时Flink读取的时间和RDS数据库的时间会相差13小时。

请在RDS数据库内执行如下语句

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

执行结果如下：

图 6-6 执行结果



Variable_name	Value
system_time_zone	CST
time_zone	SYSTEM

表 6-1 参数说明

参数	说明
system_time_zone	**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，最终数据库时区才是 CST。
time_zone	**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。

问题根因：在Mysql的time_zone是SYSTEM，system_time_zone是CST的情况下会造成bug。

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

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

6.2.9 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。

6.2.10 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"
);
```

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

在Flink 作业中，可以使用CREATE语句来定义Source表和Sink表，并指定它们的连接器类型以及相关的属性。

如果需要将数据写入到不同的Elasticsearch集群，您需要为每个集群配置不同的连接参数，并确保Flink作业能够正确地将数据路由到各个集群。

例如本例中分别对es1和es2定义连接器类型以及相关的属性。

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;
```

6.2.12 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流作业是否处于同一区域。

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

Flink JobManager提示 "Timeout expired while fetching topic metadata" ，说明Flink作业在尝试获取Kafka主题的元数据时超时了。

此时您需要先检查Flink作业和Kafka的网络连通性，确保执行Flink作业所在的队列可以访问Kafka所在的VPC网络。

若果网络不可达，请先配置网络连通后再重新执行作业。

6.3 Flink Jar 作业类

6.3.1 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/fileName")加载该文件。
   - ClassName”为需要访问该文件的类名。
   - userData为固定文件路径名，不支持修改或自定义其他路径名。
   - fileName为需要访问的文件名。

说明

本节示例适用于Flink 1.12版本。Flink 1.15版本的Jar作业开发指导请参考**Flink Jar写入数据到OBS开发指南**。

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"));
        vaildProps();
    } catch (Exception e) {
        LOGGER.info("load kafka conf failed");
        e.printStackTrace();
    }
}
```

图 6-7 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"));
        vaildProps();
    } catch (Exception e) {
        LOGGER.info("load kafka conf failed");
        e.printStackTrace();
    }
}
```

图 6-8 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
```

6.3.2 Flink Jar 包冲突，导致作业提交失败

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

首先您需要排除是否有冲突的Jar包。

含DLI Flink提供了一系列预装在DLI服务中的依赖包，用于支持各种数据处理和分析任务。

如果您上传的Jar包中包含DLI Flink运行平台中已经存在的包，则会提示Flink Jar 包冲突，导致作用提交失败。

请参考DLI用户指南中提供的依赖包信息先将重复的包删除后再上传。

DLI内置依赖包请参考《**数据湖探索用户指南**》。

6.3.3 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**。

6.3.4 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

因为账号没有在全局配置中配置服务授权，导致该账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 在委托设置页面，按需选择所需的委托权限。

其中“DLI Datasource Connections Agency Access”是跨源场景访问和使用VPC、子网、路由、对等连接的权限。

了解更多DLI委托权限请参考**DLI委托权限**。

步骤3 选择dli_management_agency需要包含的权限后，并单击“更新委托权限”。

图 6-9 更新委托权限

步骤4 委托更新完成后，重新创建跨源连接和运行作业。

----结束

6.3.5 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

6.3.6 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

6.3.7 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。

6.4 Flink 作业性能调优类

6.4.1 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 6-10 创建主题

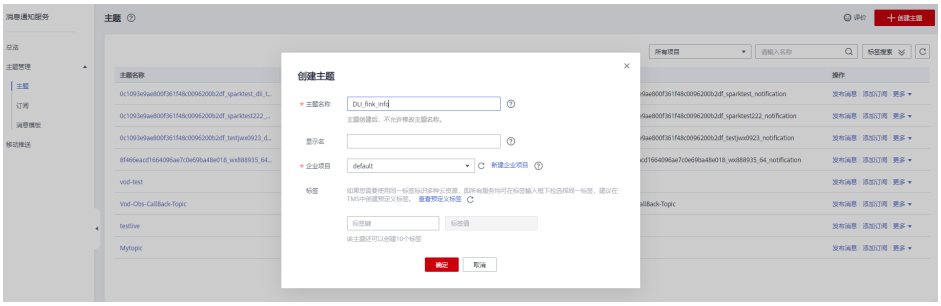


图 6-11 添加订阅

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”：

CU数量 = 管理单元 + (算子总并行数 / 单TM Slot数) * 单TM所占CU数

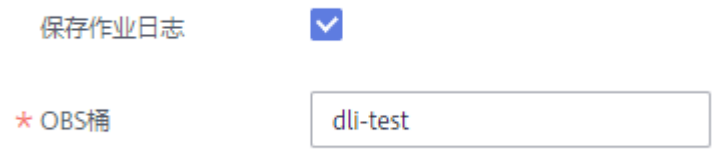
例如：CU数量为9CU，管理单元为1CU，最大并行数为16，则计算单元为8CU。

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 6-12 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 6-13 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 6-14 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，**建议大于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 6-15 云服务监控

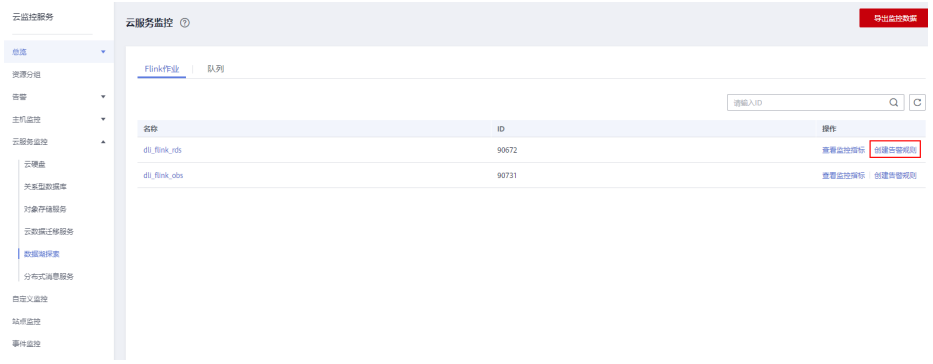
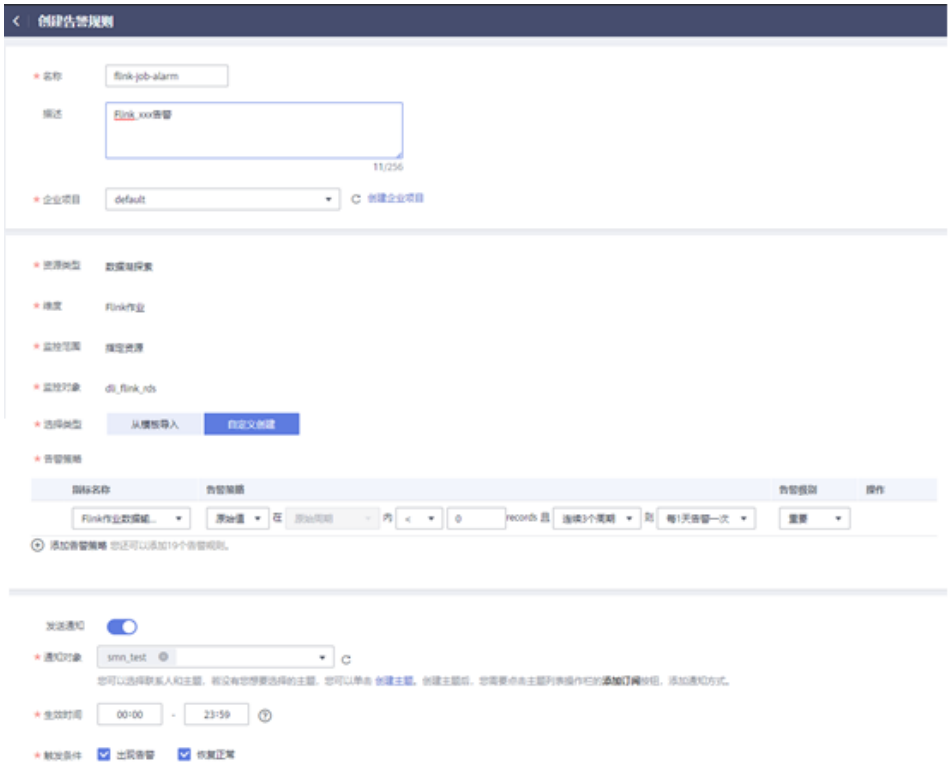


图 6-16 创建告警规则



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** 》。

6.4.2 Flink 作业性能调优

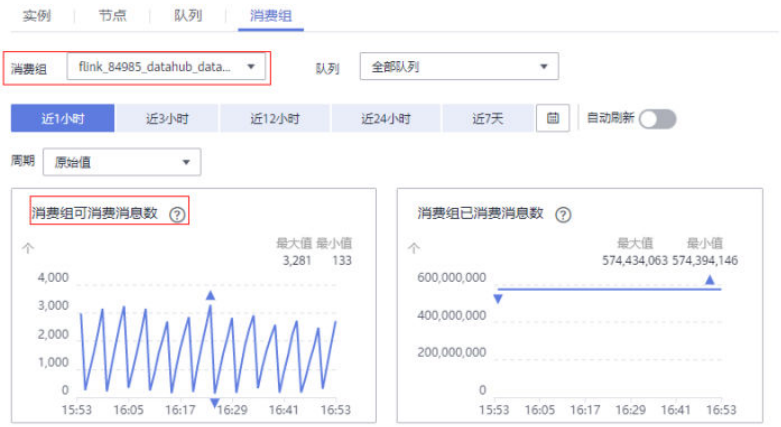
性能调优相关基本概念

- 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 6-17 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 6-18 反压状态和时延

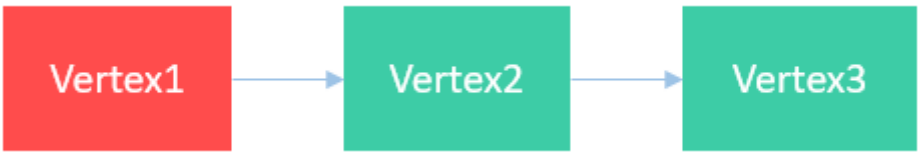
性能分析

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）

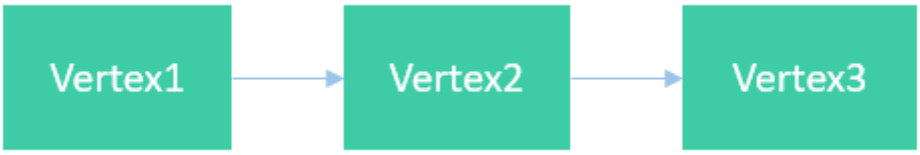
该场景说明性能瓶颈点在sink，此时需要根据具体数据源具体优化，比如对于JDBC数据源，可以通过调整写出批次(connector.write.flush.max-rows)、JDBC参数重写(rewriteBatchedStatements=true)等进行优化。

- 作业非倒数第二个算子反压高（红色）



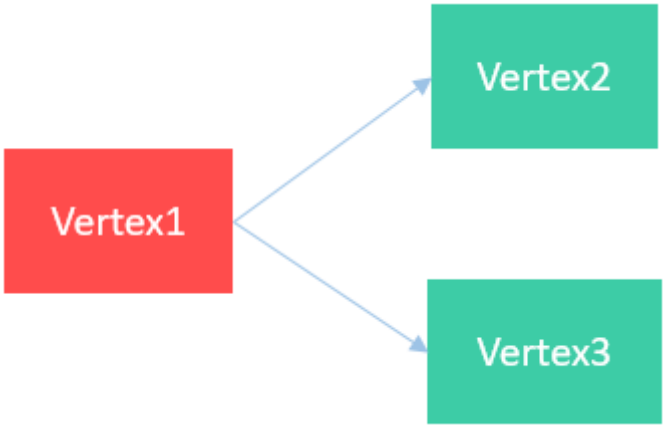
该场景说明性能瓶颈点在Vertex2算子，可以通过查看该算子描述，确认该算子具体功能，以进行下一步优化。

- 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。

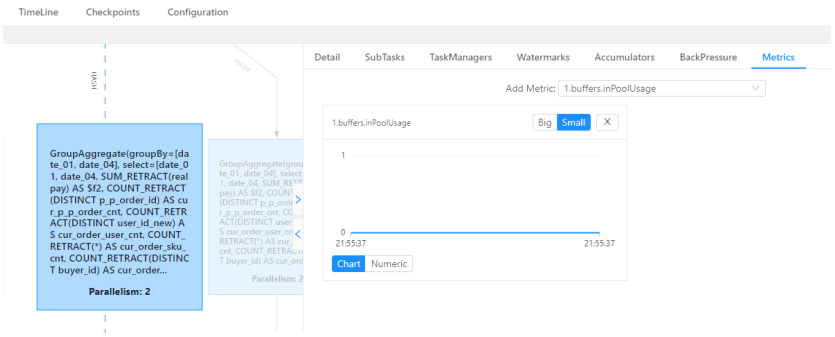
- 作业一个算子反压高（红色），而其后续的多个并行算子都不存在反压（绿色）



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 6-19 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:

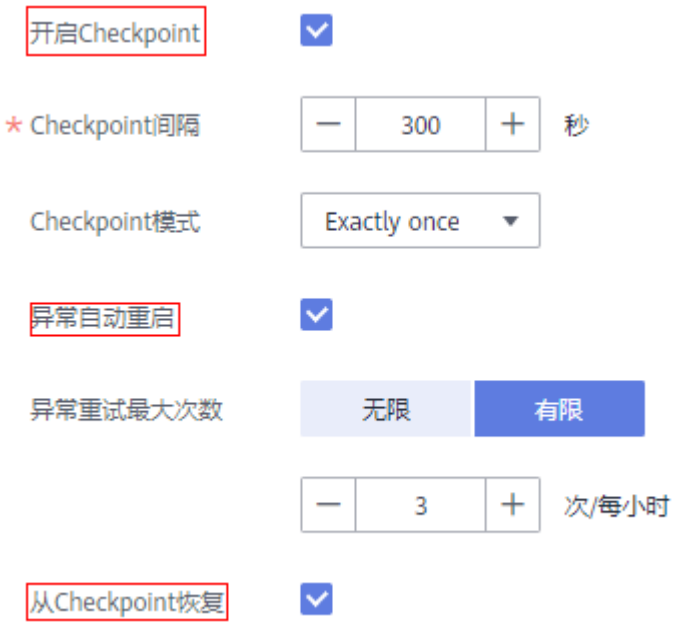
  - 增加JVM内存并增加缓存记录条数
  - 维表设置索引，加快查询速度

6.4.3 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）。

    图 6-20 Flink 作业配置参数

  - 对于Flink Jar作业，您需要在代码中开启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 6-21 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 6-22 停止作业

图 6-23 恢复保存点

6.4.4 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 6-24 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。

图 6-25 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。

6.4.5 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

本文列举了一些常见的从checkpoint恢复的场景供您参考，如**表6-2**所示。

更多场景可以使用**从checkpoint恢复的原则**结合实际情况进行判断。

表 6-2 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	未修改作业参数和算法逻辑。

相关操作：怎样从 checkpoint 恢复作业

由于Flink Checkpoint和Savepoint生成机制及格式一致，因此可以通过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。

6.4.6 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l：“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。

6.4.7 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。

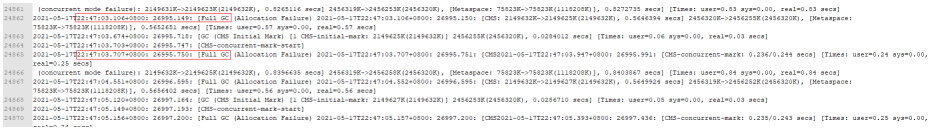
图 6-26 异常信息

```
Jobmanager. log

2021-05-17 22:44:37.312 INFO  [70] org.apache.flink.runtime.checkpoint.CheckpointCoordinator    - Triggering checkpoint 223 @ 1621262677310 for job 00d04eb6e7147e59f2d2877bdb48ce4d.
// 22:46:01.729  Jobmanager 打印心跳超时
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.
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

// 可以发现zookeeper的client 也超时了
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: 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 6-27 Full GC



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。

7 Spark 作业类

7.1 Spark 作业开发类

7.1.1 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。

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。

7.1.2 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)
```

7.1.3 通用队列操作 OBS 表如何设置 AK/SK

（推荐）方案 1：使用临时 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。

表 7-1 DLI 获取访问凭据相关开发指南

类型	操作指导	说明
Flink作业场景	Flink Opensource SQL使用DEW管理访问凭据	Flink Opensource SQL场景使用DEW管理和访问凭据的操作指导，将Flink作业的输出数据写入到Mysql或DWS时，在connector中设置账号、密码等属性。
	Flink Jar 使用DEW获取访问凭证读写OBS	访问OBS的AKSK为例介绍Flink Jar使用DEW获取访问凭证读写OBS的操作指导。
	用户获取Flink作业委托临时凭证	DLI提供了一个通用接口，可用于获取用户在启动Flink作业时设置的委托的临时凭证。该接口将获取到的该作业委托的临时凭证封装到com.huaweicloud.sdk.core.auth.BasicCredentials类中。 本操作介绍获取Flink作业委托临时凭证的操作方法。
Spark作业场景	Spark Jar 使用DEW获取访问凭证读写OBS	访问OBS的AKSK为例介绍Spark Jar使用DEW获取访问凭证读写OBS的操作指导。
	用户获取Spark作业委托临时凭证	本操作介绍获取Spark Jar作业委托临时凭证的操作方法。

方案 2：Spark Jar 作业设置获取 AK/SK

- **获取结果为AK/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()
```

7.1.4 如何查看 DLI Spark 作业的实际资源使用情况

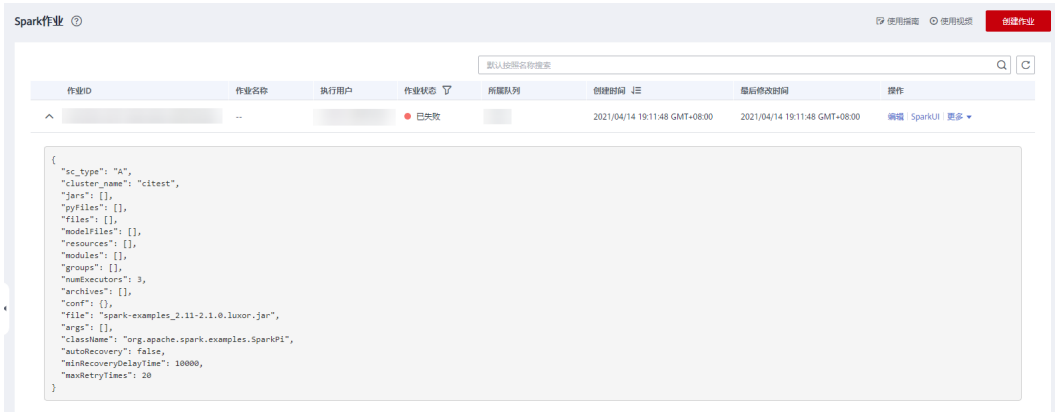
查看 Spark 作业原始资源配置

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 7-1 查看 Spark 作业原始资源配置

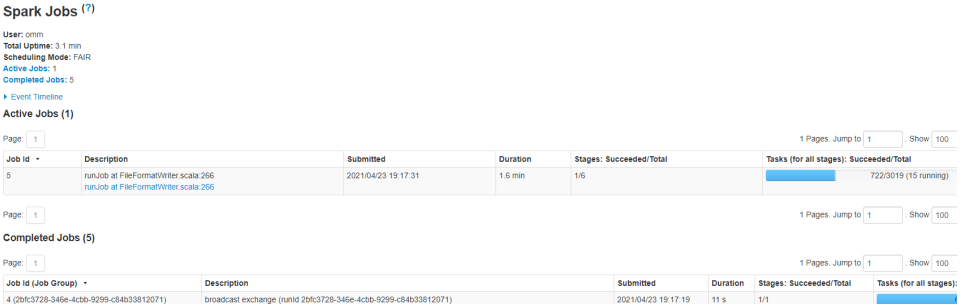


查看 Spark 作业实时运行资源

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 7-2 SparkUI



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 7-3 Driver 信息

spark.driver.cores	2
spark.driver.extraJavaOptions	-XX:CICc XX:+ExitC 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 7-4 Executor 信息

spark.executor.cores	2
spark.executor.extraClassPath	hadoop/c
spark.executor.extraJavaOptions	-XX:CICo XX:+ExitC XX:+Use Dlog4j.co Dcarbon. 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

7.1.5 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.a**egg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**》。

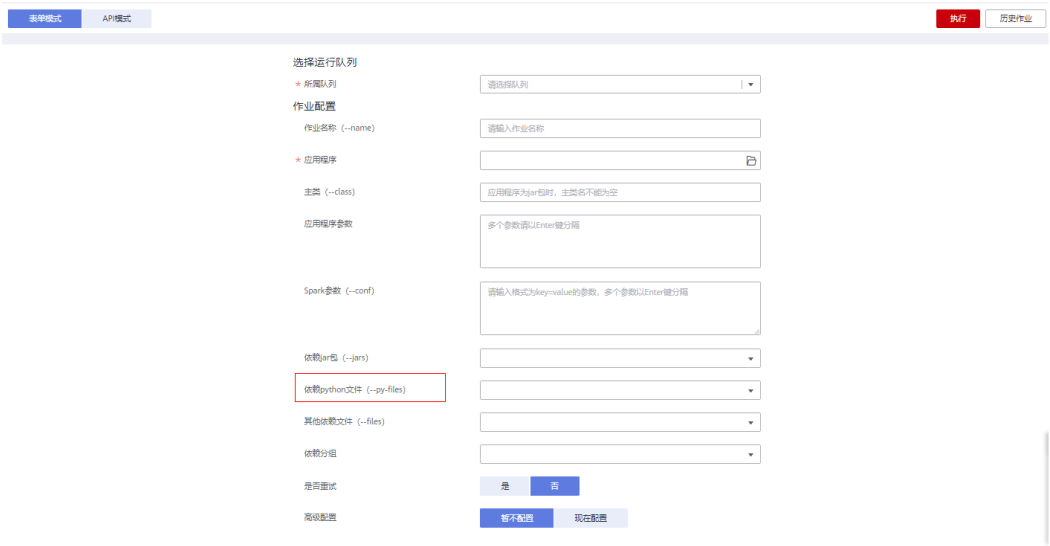
7.1.6 如何在 DLI 中运行复杂 PySpark 程序?

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，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 7-5 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 7-6 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。

7.1.7 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")
```

7.1.8 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 7-7 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()
  }
}
```

7.1.9 为什么 Spark 3.3.1 客户端视图属性查询为空字符串

问题描述

使用Spark 3.3.1版本执行作业，在客户端查询视图属性时，发现某些字段的默认属性值显示为空字符串，而使用Spark 3.1.1版本时返回为null。

根因分析

自Spark 3.3.0版本起，开源社区对CreateViewStatement语法进行了升级，调整了该语法调用接口的版本，以此来避免在查询视图属性时出现空指针异常。然而，这也导致了DESC TABLE命令在处理comment属性时，如果该属性未被配置，会显示为空字符串。

该问题影响所有使用Spark 3.3.1版本客户端查看视图默认属性配置的操作。

解决方案

针对有使用Rest接口调用视图查询的场景，在使用Spark 3.3.1执行作业时需要对业务逻辑进行必要的修改匹配空字符串结果。

7.1.10 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。

7.1.11 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。

7.1.12 Spark Jar 任务使用 LakeFormation 元数据时报错

问题描述

通过Spark Jar作业读写LakeFormation元数据时，作业启动时提示以下错误：

field 'location' must match '^(obs|har)://.+/.+$'

作业无法继续，表或库创建失败。

详细错误提示信息如下：

```
2025-07-25 09:47:55,782 | INFO  | [main] | Send request, http method: GET, url:
https://172.16.xxx/v1/02f6e35xxxxx/instances/default/catalogs/lgl/databases/default, request Id: 75d0ee10-
xxxx| com.huawei.cloud.dalf.lakecat.client.ApiClient.getResponse(ApiClient.java:392)
2025-07-25 09:47:56,259 | INFO  | [main] | Send request, http method: GET, url:
https://172.16.xxx/v1/02f6e35xxxxx/instances/default/catalogs/lgl/databases/default, request Id: eef1c48f-
xxx| com.huawei.cloud.dalf.lakecat.client.ApiClient.getResponse(ApiClient.java:392)
2025-07-25 09:47:56,433 | INFO  | [main] | Send request, http method: POST, url:
https://172.16.xxx/v1/02f6e35xxxxx/instances/default/catalogs/lgl/databases, request Id: e6b4e588-xxx |
com.huawei.cloud.dalf.lakecat.client.ApiClient.getResponse(ApiClient.java:392)
2025-07-25 09:47:56,495 | ERROR | [main] | field 'location' must match '^(obs|har)://.+/.+$' |
com.huawei.cloud.dalf.lakecat.client.hiveclient.impl.HiveExceptionHandler.printLog(HiveExceptionHandler.jav
a:51)
com.huawei.cloud.dalf.lakecat.client.exception.LakeFormationLakeCatClientException:
400 Bad Request: "{"error_code":"common.00000400","error_msg":"field 'location' must match '^(obs|
har)://.+/.+$'"}"
    at
com.huawei.cloud.dalf.lakecat.client.exception.LakeFormationExceptionHandler.exceptionHandler(LakeForma
tionExceptionHandler.java:57) ~
```

根因分析

该异常表明某个数据库或表的location字段未通过LakeFormation的正则校验。

default数据库缺失，Spark自行创建default库提示错误。

- Spark Catalog启动时会检查当前Catalog下是否存在default数据库：
  - 存在，读取default 数据库数据。
  - 不存在，Spark 会自动创建。

    但Spark默认生成的location不满足LakeFormation规则，导致后续引用default库的操作报错。
- 创建数据库语句中指定的location需要以obs:// 开头，否则不符合LakeFormation校验规则。

解决方案

在正式使用新Catalog前，请先手动创建default数据库并指定合法OBS路径，可避免Spark自动创建带来的隐藏风险。

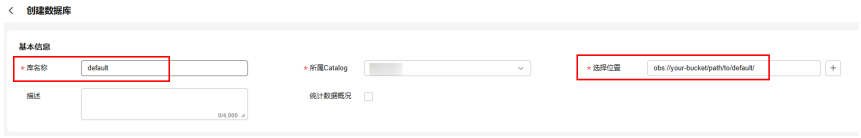
- 检查已有自建库/表报错
  - 检查并修改对应库/表的location，确保以 obs://开头。
  - 如无法修改，可删除后重建，并在建表/库语句中指定正确的OBS路径。
- 手动创建并初始化default数据库

a. 登录LakeFormation控制台，选择“元数据 > 数据库”。
b. 选择待使用的Catalog，点击 “创建数据库”。

图 7-8 创建数据库

c. 数据库名称填写default，“位置”填写符合规则的OBS路径，例如：obs://your-bucket/path/to/default/

图 7-9 创建 default 数据库并配置正确的 OBS 路径



d. 确认创建后，重新提交 Spark Jar 作业。

7.2 Spark 作业运维类

7.2.1 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")
```

7.2.2 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**。

7.2.3 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]

该报错信息可能是由于OBS桶被设置为了DLI日志桶，而日志桶不能用于DLI的其他业务功能。

您可以按以下操作步骤进行查询：

1. 检查该OBS桶是否被设置为了DLI日志桶。

   在DLI管理控制台的“全局配置 > 作业配置” 页查看对应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。

2. 确认桶是否应用于其他业务功能。

   如果是，您可以在DLI管理控制台页面更改作业配置，选择其他未被占用的OBS桶用于DLI日志存储。

7.2.4 Spark 作业运行大批量数据时上报作业运行超时异常错误

当Spark作业运行大批量数据时，如果出现作业运行超时异常错误，通常是由于作业的资源配置不足、数据倾斜、网络问题或任务过多导致的。

解决方案：

- 设置并发数：通过设置合适的并发数，可以启动多任务并行运行，从而提高作业的处理能力。

  例如访问DWS大批量数据库数据时设置并发数，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。

- 调整Spark作业的Executor数量，分配更多的资源用于Spark作业的运行。

7.2.5 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。

1. 上传数据到OBS桶：通过OBS管理控制台或者使用命令行工具将存储在sftp中的文件数据上传到OBS桶中。

   Spark读取OBS文件数据，详见**使用Spark Jar作业读取和查询OBS数据**。
2. 配置Spark作业：配置Spark作业访问OBS中存储的数据。
3. 提交Spark作业：完成作业编写后，提交并执行作业。

7.2.6 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。

7.2.7 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。

7.2.8 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 7-10 avro 类型的 OBS 创建失败

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。

7.2.9 使用 Spark3.3.1 的 SQL 查询语句使用 rand 函数报错，提示“Input argument to rand must be a constant”

问题描述

使用Spark3.3.1查询SQL语句使用rand函数报错，提示“Input argument to rand must be a constant”。

图 7-11 rand 函数报错

执行语句	select c1_int from Test_Boundary where rand(c1_int) IS NULL or rand(c1_int) IS NOT N
创建时间	2023/07/06 11:50:10 GMT+08:00
参数设置	{"spark.sql.shuffle.partitions":"50"}
结果条数	0
结果状态	执行失败
失败原因	DLI.0005: Input argument to rand must be a constant.; Project [c1_int#137135] +- Filter (isnull(rand(c1_int#137135)) OR isnotnull(rand(c1_int#137135))) +- SubqueryAlias spark_catalog.offlinebva.test_boundary +- HiveTableRelation [`offlinebva`.`test_boundary`, org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, Data Cols: [c1_int#137135, c2_Bigint#137136L, c3_Decimal#137137, c4_double#137138, c5_string#137139, c6_Time..., Partition Cols: []]

解决方案

Spark3.3.1版本中rand函数只能接受常量参数。

因此请将SQL查询语句中rand函数中的参数为常量参数后再次执行。

7.2.10 使用 Spark 3.3.1 客户端创建 view 同时 join 查询数据写入报错，提示 Not allowed to create a permanent view

问题描述

使用Spark 3.3.1客户端创建子查询带聚合函数的视图，在数据写入报错，提示“Not allowed to create a permanent view without explicitly assigning an alias for expression count”。

问题分析

为了解决不同版本视图中默认别名不一致导致schema兼容性问题，Spark 3.2.0版本新增对视图的约束导致在创建视图同时join查询数据写入报错。

解决方案

为了与Spark 3.2.0之前版本使用行为一致，可以通过客户端设置参数“spark.sql.legacy.allowAutoGeneratedAliasForView=true”解决。

8 DLI 资源配额类

8.1 什么是用户配额?

配额是指云平台预先设定的资源使用限制，包括资源数量和容量等。设置配额是为了确保资源合理的分配和使用，避免资源过度集中和资源浪费。

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。

在通过审理之后，系统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**。

8.2 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 8-1 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。

8.3 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 8-2 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”。

9 DLI 权限管理类

9.1 队列引擎版本升级后，在创建表时，提示权限不足怎么办?

问题描述

队列版本从Spark 2.x版本切换至Spark 3.3.x版本时，或切换使用HetuEngine后，如果已经赋予IAM用户的建表权限，但是在创建表时候仍然提示权限不足。

根因分析

DLI队列的引擎版本不同，校验的权限范围不同：

HetuEngine不支持通过IAM用户授权，需使用DLI资源授权。

解决方案

请参考《 数据湖探索用户指南 》中的**数据库权限管理章节**授予用户创建表的权限。

9.2 什么是 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。

9.3 更新程序包时提示权限不足怎么办?

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。

9.4 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。

授权方式请参考资料**表权限管理**。

9.5 已经给表授权，但是提示无法查询怎么办?

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，此时应先检查当前表的权限是否还存在，

- 检查权限是否仍然存在：
  如用户权限被取消则可能导致提示权限缺失无法查询表数据。
- 查看表的创建时间：
  查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。

9.6 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。

因为继承的权限已经足够使用，重复授权还可能导致表权限管理上的混乱。

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。

9.7 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。

当前权限分配情况

- 用户A具备的权限：Table1的管理员权限。
- 用户B具备的权限：View1的管理员权限。
- 用户C具备的权限：Table1的查询表权限。

解决措施

不同版本的Spark引擎对View表的权限要求不同：

- **Spark2.4.x**：用户C具备View查询权限，且用户B具备Table1的查询权限。
  对权限的要求如下：
  - 用户B具备的权限：View1的管理员权限、Table1的查询权限（当前缺失）。
  - 用户C具备的权限：Table1的查询表权限。

  针对该场景请补充用户B对Table1的查询表权限后，用户C重试查询View1。
- **Spark 3.3.x**：用户C具备View查询权限，且用户C具备Table1的查询权限。
  对权限的要求如下：
  - 用户C具备的权限：Table1的查询表权限。View1的查询权限（当前缺失）。

  针对该场景请补充用户C对View1的查询权限后，用户C重试查询View1。

9.8 提交作业时提示作业桶权限不足怎么办?

问题描述

已经配置DLI作业桶，且完成Flink桶授权后在提交作业时仍然提示桶未授权怎么办?

根因分析

使用DLI作业桶需要确保已完成DLI作业桶的权限配置。

您需要在OBS管理控制台中检查DLI作业桶的桶策略，确保策略中包含了允许DLI服务进行必要操作的授权信息。

确保没有任何策略明确拒绝了DLI服务对桶的访问。IAM策略是优先考虑拒绝（deny）权限的，即使有允许（allow）权限，如果有拒绝权限存在，也会导致授权失败。

排查方案

1. 在OBS管理控制台找到DLI作业桶。
2. 查看所选桶的桶策略。

   DLI Flink作业所需要使用的桶授权信息如下，其中*domainId*和*userId*分别为DLI的账号和子账号，*bucketName*为用户桶名，*timeStamp*为策略创建时的时间戳。

```
{
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": {
            "ID": [
               "domain/domainId:user/userId"
            ]
         },
         "Action": [
            "GetObject",
            "GetObjectVersion",
            "PutObject",
            "DeleteObject",
            "DeleteObjectVersion",
            "ListMultipartUploadParts",
            "AbortMultipartUpload",
            "GetObjectAcl",
            "GetObjectVersionAcl"
         ],
         "Resource": [
            "bucketName/*"
         ],
         "Sid": "未命名的桶策略-Timestamp-0"
      },
      {
         "Effect": "Allow",
         "Principal": {
            "ID": [
               "domain/domainId:user/userId "
            ]
         },
         "Action": [
            "HeadBucket",
            "ListBucket",
            "ListBucketVersions",
            "ListBucketMultipartUploads",
            "GetBucketAcl",
            "GetBucketLocation",
            "GetBucketLogging",
            "GetLifecycleConfiguration"
         ],
         "Resource": [
            " bucketName "
         ],
         "Sid": "未命名的桶策略-Timestamp-1"
      }
   ]
}
```

3. 在管理控制台检查以下权限相关内容，查看策略名称是否与**2**一致。
   - 效力：允许
   - 授权资源：按需授权桶和对象。
   - 授权操作：与**2**中Action一致

   常用检查项：

- 检查是否配置了所有账号的某些拒绝操作，且这些操作是上述DLI所需要的授权操作。
- 检查是否对DLI的被授权用户配置了某些拒绝操作，且这些操作是上述DLI所需要的授权操作。

9.9 提示 OBS Bucket 没有授权怎么办?

DLI更新委托后，将原有的dli_admin_agency升级为dli_management_agency。

dli_management_agency包含跨源操作、消息通知、用户授权操作所需的权限，除此之外的其他委托权限需求，都需自定义DLI委托。

授权DLI读写OBS的权限并不包含在的DLI委托dli_management_agency中。需要您创建自定义委托，并将委托配置在作业中（使用Flink 1.15和Spark 3.3及以上版本的引擎执行作业时需要配置）。

了解dli_management_agency请参考**DLI委托概述**。

创建自定义委托并在作业中配置委托的操作步骤请参考**自定义DLI委托权限**。

10 DLI API 类

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 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。

10.5 API 接口返回的中文字符为乱码，如何解决?

当API接口返回的中文字符出现乱码时，通常是因为字符编码格式不匹配。

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"))
```