

数据湖计算 DLC

常见问题



腾讯云

【 版权声明 】

©2013–2024 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分內容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【 商标声明 】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或 95716。

文档目录

常见问题

权限类常见问题

引擎类常见问题

功能类常见问题

Spark 作业类常见问题

常见问题

权限类常见问题

最近更新时间：2023-10-12 16:38:11

为什么要同时开通 CAM 权限和 DLC 权限？

类型	说明
CAM 权限	开通子账号访问 DLC 权限。
DLC 权限	添加 CAM 子账号为 DLC 用户账号后，可对数据库表与引擎的权限进行配置管理。

为什么我开通了子账号，还是使用不了数据探索（数据作业）？

账号开通后，DLC 管理员需要在数据湖计算 DLC 的 [权限管理](#) 功能里，开通子账号的角色属性、引擎使用权限和数据的库表读写权限。

引擎类常见问题

最近更新时间：2024-04-25 17:15:51

共享引擎和独享引擎的区别是什么？

引擎类型	说明	计费模式	特点
共享引擎	为当前地域下所有用户公共使用的引擎	按量计费 ：按扫描量计费，不产生任何费用	<ol style="list-style-type: none">1. 无需配置即可使用2. 适合数据量较小、临时数据计算场景。
独享引擎	为用户独享引擎资源	按量计费 ：按 CU 量计费，没有任务时可挂起集群，挂起时不产生任何费用	<ol style="list-style-type: none">1. 资源独享，支持配置资源规模，弹性伸缩2. 适合有一定任务量但任务周期不规律的数据计算场景
		包年包月 ：按 CU 量计费，集群无需等待随时可用。弹性部分按量计费。	<ol style="list-style-type: none">1. 资源独享，支持配置资源规模，弹性伸缩2. 适合任务量大且稳定的数据计算场景

一个集群，支持多少任务并行？可以调整吗？

1个集群，任务并行数是5，如果需要调整，可以在 [数据引擎](#) 功能中，选择需要修改的引擎，单击规格配置进行调整。

为什么任务实际使用 CU 核数会小于引擎的集群规模中指定的 CU 核数？

以下几种情况都可能导致实际使用 CU 核数小于引擎的集群规模中指定的 CU 数：

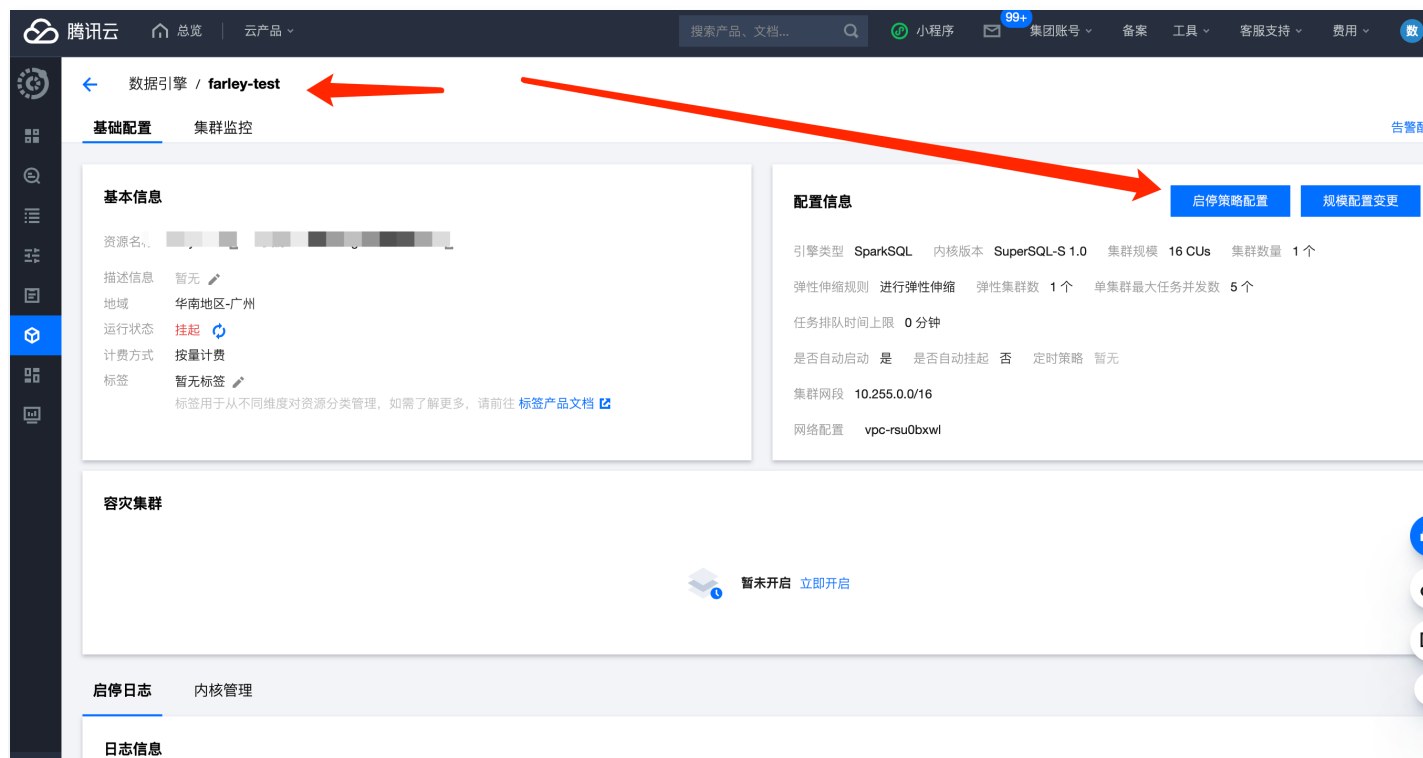
1. 集群中有其他任务正在执行。
2. 批作业集群指定的 driver 资源 + executor 资源总数小于集群规模。
3. 按量计费集群，使用时才会发起资源申请，当 CU 核数较多时，不能确保资源完全满足申请的 CU 数。
4. 批处理作业指定了增强型网络配置用于打通其他 VPC 网络，但是其他 VPC 网络的 IP 数不足以启动所有的 executor。

DLC 集群是否可以访问同地域的其他 VPC 下的 IP/服务？是否可以访问外网？

DLC 引擎可以访问同地域下的其他 VPC，需要在 [数据引擎](#) > [网络配置](#) 中创建一个网络配置和目标 VPC 打通，并在批处理作业中指定使用改网络配置。DLC 引擎默认是无法访问外网的。但是通过增强型网络配置，并且目标 VPC 配置了可以访问外网的路由规则，那么 DLC 可以通过增强型网络配置访问外网。

如何修改集群的自动启停时间？

在 **数据引擎** 功能中，单击引擎列表中需要修改的引擎名称，进入修改页，单击**启停策略配置**进行修改。



若在任务运行中进行集群变配，是否会导致任务失败？

我们为您罗列了各类引擎在常见任务进行中，若发起变配带来的任务影响，请您参考：

引擎类型	购买方式	是否会影响任务失败
SuperSQL-Spark作业	按量计费	任务不会受影响
	包年包月	当发起集群规格变配时： <ul style="list-style-type: none"> 扩容集群规格不会影响任务 缩容集群规格，会在流程中等待pod运行结束后再去隔离/销毁缩容的机器，如果任务运行时间长可能导致流程卡住，导致任务重启
SuperSQL-SparkSQL	按量计费	当发起集群规格变配时，会导致任务重启
	包年包月	当发起集群规格变配 或 减少集群数量时，会导致任务重启
SuperSQL-Presto	按量计费	当发起集群规格变配时，会导致任务重启
	包年包月	当发起集群规格变配 或 减少集群数量时，会导致任务重启

集群规格

测试环境 16 CUs	64 CUs	128 CUs	256 CUs	512 CUs	1024 CUs	工单购买 1536 CUs
工单购买 2048 CUs	工单购买 2560 CUs	工单购买 4096 CUs				

1CU ≈ 1核4GB,指定集群运行的最小规格,单个任务的计算能力受此影响。为保证大量数据任务稳定运行,生产环境建议使用128CUs以上规格。如需购买超过256CUs的引擎,为避免资源不足,请提前[通过工单](#) 与我们联系为您预留资源。

集群数量

- 1 +

数据引擎下支持多个集群,每个集群规格固定,影响任务并发数量,多个集群可以提供更高的任务并发处理能力

功能类常见问题

最近更新时间：2023-07-26 17:00:41

外表和原生表的区别是什么？

- 原生表：是您存放于 DLC 托管存储上的表，默认为 Iceberg 格式。使用原生表无需关注 Iceberg 底层文件，而且具备数据优化等能力帮助构建数据湖。
- 外表：文件为您自己账号下的 COS 桶或其他第三方数据存储的表。DLC 可以直接建立外表进行分析，无需额外加载数据。

数据湖计算 DLC 支持数据调度吗？

如您有数据调度的需求，可前往咨询并开通 [数据开发治理平台 Wedata](#)。

数据湖计算 DLC 是否支持自助上传 jar 包，自定义 spark 函数？

支持。

元数据加速桶支持 CDN 吗？

加速桶不支持，普通桶支持。

元数据加速桶与普通 COS 桶的区别是什么？

元数据加速桶与普通桶的区别在于：对大数据分析中常见的 rename，list 文件操作等会有明显的加速效果。

数据存储需要绑定引擎权限吗？

元数据加速桶需要绑定引擎权限，普通的对象存储不需要绑定引擎权限。

一个 COS 的 bucket 可以绑定多个 DLC 数据引擎么？

可以。

Spark 作业类常见问题

最近更新时间：2024-01-05 14:21:01

PySpark 任务数据倾斜导致 python+jvm 内存占用超过 K8s request 内出现 OOMkilled?

问题描述：PySpark 任务执行时，executor 日志出现“k8s执行OOMKilled”，使用内存超过 k8s 限制的内存。

原因分析：K8s 申请的内存是根据 Spark executor 内存乘以 memoryOverheadFactor 计算出来的，如果 Python 处理的数据有倾斜，或单条数据过大，可能导致使用内存超过 K8s 分配的内存。

解决方案：添加任务参数 spark.kubernetes.memoryOverheadFactor=0.8，默认值0.4。

操作步骤：登录 [DLC 控制台](#)，进入数据作业（Spark 作业）> 编辑作业，按照如下配置：



Insert into/overwrite 后如何自动添加 repartition 命令对数据做分区以减少小文件数量?

解决方案：开启自动重分区，配置如下参数：

```
spark.sql.adaptive.enabled: true
spark.sql.adaptive.insert.repartition: true
spark.sql.adaptive.insert.repartition.forceNum: 300 (指定了具体需要分区的值)
```

操作步骤：

- 程序内配置到 SparkConf:

```
val spark: SparkSession = args(0) match {
  case "1" =>
    SparkSession.builder().getOrCreate()
  case "2" =>
    SparkSession.builder().config("spark.sql.adaptive.insert.repartition", "true").getOrCreate()
  case "3" =>
    SparkSession.builder().config("spark.sql.adaptive.insert.repartition", "true")
      .config("spark.sql.adaptive.coalescePartitions.enabled", "true")
      .config("spark.sql.adaptive.coalescePartitions.initialPartitionNum", "300")
      .getOrCreate()
  case "4" =>
    SparkSession.builder().config("spark.sql.adaptive.insert.repartition.forceNum", "300").getOrCreate()
  case _ =>
    SparkSession.builder().getOrCreate()
}
```

- 程序内设置 SQL SET:

创建作业
✕

基本信息 ▲

作业名称 * 支持中文、英文、数字与“_”，最多100个字符

作业类型 * 批处理 流处理 SQL作业

数据引擎 * 计费以所选数据引擎计费模式为准，可至[数据引擎](#) 查看管理。数据引擎的网络配置信息可至[网络配置](#) 查看管理

SQL脚本 SQL编辑 DLC查询文件

```
SET spark.sql.adaptive.enabled=true
SET spark.sql.adaptive.insert.repartition=true
SET spark.sql.adaptive.insert.repartition.forceNum=300
INSERT OVERWRITE DataLakeCatalog.demo.test SELECT * FROM
DataLakeCatalog.demo.test2
```

作业参数 (--config) -config信息，spark开头的参数信息，多条配置信息换行填写

PySpark 任务高并发写 COS 存储桶时返回503错误?

问题描述: PySpark 任务高并发写 COS 存储桶时，executor 有非常多的 COS 返回503错误。

问题原因: Spark 任务写 COS 时的并行核数为 `*fs.cosn.trsf.fs ofs.data.transfer.thread.count` 指定。例如，4096核下不做调优，默认并发度为 $4096 \times 32 = 131072$ ，导致 COS 瓶颈。

解决方案:

1. COS 新建一个元数据加速桶，避免 spark 任务写时的 list 和 rename 超频。
2. 通过 COS 调整元数据加速桶带宽限制。
3. 任务添加以下参数降低高并行度时对 COS 访问压力过大。

```
fs.cosn.trsf.fs ofs.data.transfer.thread.count=8
fs.cosn.trsf.fs ofs.block.max.file.cache.mb=0
spark.hadoop.fs.cosn.trsf.fs ofs.data.transfer.thread.count=8
```

```
spark.hadoop.fs.cosn.trsf.fs ofs.block.max.file.cache.mb=0
```

常用数据治理 SQL 有哪些?

- 关闭库治理 SQL

```
ALTER DATABASE DataLakeCatalog.demo_db
SET
DBPROPERTIES (
  'dlc.ao.data.govern.inherit' = 'none',
  'dlc.ao.merge.data.enable' = 'disable',
  'dlc.ao.expired.snapshots.enable' = 'disable',
  'dlc.ao.remove.orphan.enable' = 'disable',
  'dlc.ao.merge.manifests.enable' = 'disable'
)
```

- 开启库治理 SQL

```
ALTER DATABASE DataLakeCatalog.db_name
SET
DBPROPERTIES (
  'dlc.ao.data.govern.inherit' = 'none',
  'dlc.ao.merge.data.enable' = 'enable',
  'dlc.ao.merge.data.engine' = 'bda-sinker',
  'dlc.ao.merge.data.min-input-files' = '10',
  'dlc.ao.merge.data.target-file-size-bytes' = '536870912',
  'dlc.ao.merge.data.interval-min' = '90',
  'dlc.ao.expired.snapshots.enable' = 'enable',
  'dlc.ao.expired.snapshots.engine' = 'bda-sinker',
  'dlc.ao.expired.snapshots.retain-last' = '5',
  'dlc.ao.expired.snapshots.before-days' = '2',
  'dlc.ao.expired.snapshots.max-concurrent-deletes' = '4',
  'dlc.ao.expired.snapshots.interval-min' = '150',
  'dlc.ao.remove.orphan.enable' = 'enable',
  'dlc.ao.remove.orphan.engine' = 'bda-sinker',
  'dlc.ao.remove.orphan.before-days' = '3',
  'dlc.ao.remove.orphan.max-concurrent-deletes' = '4',
  'dlc.ao.remove.orphan.interval-min' = '600',
  'dlc.ao.merge.manifests.enable' = 'enable',
  'dlc.ao.merge.manifests.engine' = 'bda-sinker',
  'dlc.ao.merge.manifests.interval-min' = '1440'
)
```

- 关闭表治理 SQL

```
ALTER TABLE
`DataLakeCatalog`.`db_name`.`tb_name`
SET
TBLPROPERTIES (
'dlc.ao.data.govern.inherit' = 'none',
'dlc.ao.merge.data.enable' = 'disable',
'dlc.ao.expired.snapshots.enable' = 'disable',
'dlc.ao.remove.orphan.enable' = 'disable',
'dlc.ao.merge.manifests.enable' = 'disable'
)
```

- 开启继承库治理 SQL

```
ALTER TABLE `DataLakeCatalog`.`db_name`.`tb_name`
SET TBLPROPERTIES ('dlc.ao.data.govern.inherit' = 'default')
```

- 开启表治理 SQL

```
ALTER TABLE
`DataLakeCatalog`.`db_name`.`tb_name`
SET
TBLPROPERTIES (
'dlc.ao.data.govern.inherit' = 'none',
'dlc.ao.merge.data.enable' = 'enable',
'dlc.ao.merge.data.engine' = 'bda-sinker',
'dlc.ao.merge.data.min-input-files' = '10',
'dlc.ao.merge.data.target-file-size-bytes' = '536870912',
'dlc.ao.merge.data.interval-min' = '90',
'dlc.ao.expired.snapshots.enable' = 'enable',
'dlc.ao.expired.snapshots.engine' = 'bda-sinker',
'dlc.ao.expired.snapshots.retain-last' = '5',
'dlc.ao.expired.snapshots.before-days' = '2',
'dlc.ao.expired.snapshots.max-concurrent-deletes' = '4',
'dlc.ao.expired.snapshots.interval-min' = '150',
'dlc.ao.remove.orphan.enable' = 'enable',
'dlc.ao.remove.orphan.engine' = 'bda-sinker',
'dlc.ao.remove.orphan.before-days' = '3',
'dlc.ao.remove.orphan.max-concurrent-deletes' = '4',
'dlc.ao.remove.orphan.interval-min' = '600',
'dlc.ao.merge.manifests.enable' = 'enable',
'dlc.ao.merge.manifests.engine' = 'bda-sinker',
)
```

```
'dlc.ao.merge.manifests.interval-min' = '1440'  
)
```

- 不指定 Where 条件全表合并 SQL

```
CALL `DataLakeCatalog`.`system`.`rewrite_data_files`(  
  `table` => 'tb_name',  
  `options` => map(  
    'min-input-files',  
    '10',  
    'target-file-size-bytes',  
    '536870912',  
    'delete-file-threshold',  
    '1',  
    'max-concurrent-file-group-rewrites',  
    '20'  
  )  
)
```

- 支持 Where 条件增量合并 SQL

```
CALL `DataLakeCatalog`.`system`.`rewrite_data_files`(  
  `table` => 'tb_name',  
  `options` => map(  
    'min-input-files',  
    '10',  
    'target-file-size-bytes',  
    '536870912',  
    'delete-file-threshold',  
    '1',  
    'max-concurrent-file-group-rewrites',  
    '20'  
  ),  
  `where` => 'field_date > "2022-01-01" and field_date <= "2023-01-01"  
)
```

- 快照过期 SQL

```
CALL `DataLakeCatalog`.`system`.`expire_snapshots`(  
  `table` => 'tb_name',  
  older_than => TIMESTAMP '2023-02-28 16:06:35.000',  
  retain_last => 1,  
  max_concurrent_deletes => 4,
```

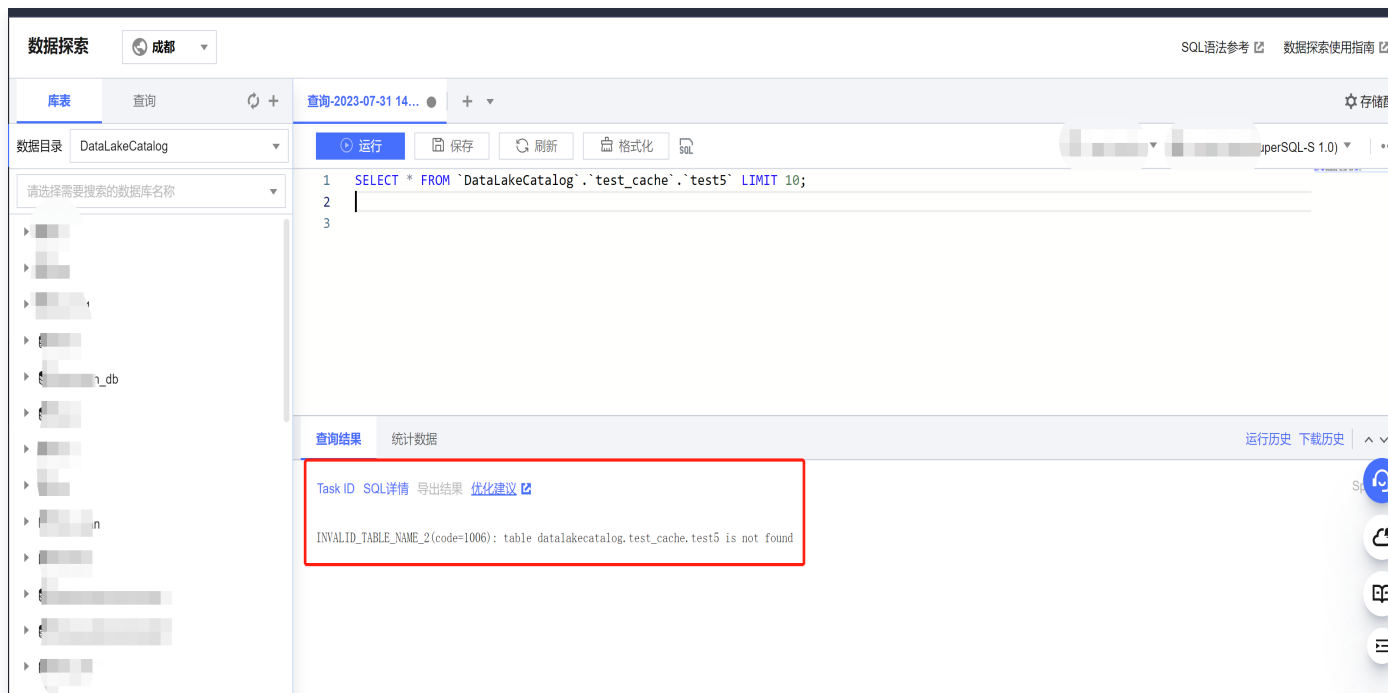
```
stream_results => true
)
```

如何查看 SQL 执行计划和 SQL 执行的日志?

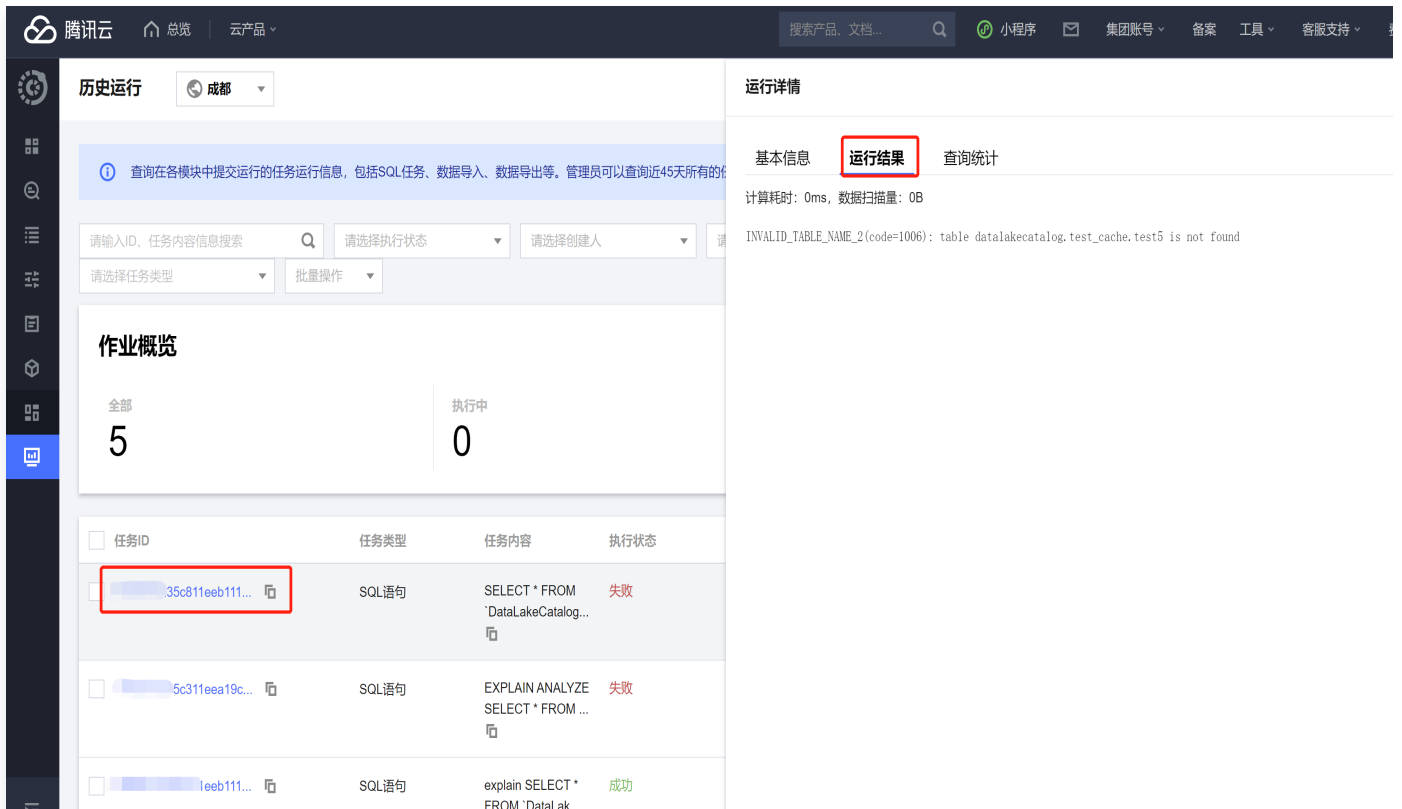
查看 SQL 执行计划: 用 explain 关键字在数据探索中查看 SQL 执行的物理计划, explain 详细使用请参考 [SQL 统一语法 > EXPLAIN](#)

查看SQL执行日志:

1. 数据探索执行 SQL, 运行结果展示 SQL 执行日志。



2. DLC 控制台 > 数据运维 > 历史运行可以查看 SQL 执行日志。



CAST 未自动转换精度导致数据写入失败?

问题描述: hive sql 迁移 spark sql 时, 报错 Cannot safely cast 'class_type': string to bigint。

问题定位: Spark 3.0.0 开始, Spark SQL 在处理类型转换时有 3 种安全策略:

- ANSI: 不允许 Spark 进行某些不合理的类型转换, 如: string 转换成 timestamp。
- LEGACY: 允许 Spark 进行类型强制转换, 只要它是有效的 Cast 操作。
- STRICT: 不允许 Spark 进行任何可能有损精度的转换。默认策略是 ANSI。

解决方案: 修改策略为 LEGACY, 设置 spark.sql.storeAssignmentPolicy=LEGACY。

QUERY_PROGRESS_UPDATE_ERROR(code=3060): Failed to update statement progress 错误

问题描述: 数据探索中提交 spark sql 任务, 执行过程中, 提示 Failed to Update statement progress 错误

问题定位: 当有多个 Spark SQL 任务提交时, 需要持续的异步跟进每个 SQL 的执行进度, 这里异步处理的队列有限制, 默认值是100 (2024.1.14以后的版本更新为300)。所以当某个任务被提交后一直没有执行完成, 而后续新增的任务超过了队列上限会导致该错误。如果您出现这种错误, 一般表示该 SQL 任务可能是个长尾任务, 需要关注对其他任务资源占用是否合理。

解决方案: 可以在引擎上调整配置 livy.rsc.retained-statements, 调整到大于默认值的值。注意调整后引擎会重启。具体值可以根据任务的并发量来设置, 该参数对集群影响较小, 同时提交的 SQL 并发量达到100-200/min 时, 该参数调整到6000也实测没有影响。