

# 数据湖计算 开发指南 产品文档





【版权声明】

©2013-2025 腾讯云版权所有

本文档著作权归腾讯云单独所有,未经腾讯云事先书面许可,任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】



及其他腾讯云服务相关的商标均为腾讯集团下的相关公司主体所有。另外,本文档涉及的第三方主体的商标,依法 由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况,部分产品、服务的内容可能有所调整。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。



## 文档目录

开发指南

SparkJar 作业开发指南 PySpark 作业开发指南 查询性能优化指南 UDF 函数开发指南 物化视图 系统约束 元数据信息 计算任务



# 开发指南 SparkJar 作业开发指南

最近更新时间:2024-07-31 18:02:45

### 应用场景

DLC 完全兼容开源 Apache Spark,支持用户编写业务程序在 DLC 平台上对数据进行读写和分析。本示例演示通过 编写 Java 代码在 COS 上读写数据和在 DLC 上建库表、读写表的详细操作,帮助用户在 DLC 上完成作业开发。

### 环境准备

依赖: JDK1.8 Maven IntelliJ IDEA

### 开发流程

### 开发流程图

DLC Spark JAR 作业开发流程图如下:



### 创建资源

第一次在 DLC 上运行作业,需新建 Spark 作业计算资源,例如新建名称为 "dlc-demo" 的 Spark 作业资源。

1. 登录数据湖计算 DLC 控制台,选择服务所在区域,在导航菜单中单击数据引擎。

2. 单击左上角创建资源,进入资源配置购买页面。

3. 在集群配置 > 计算引擎类型 项选择 Spark 作业引擎。



基础配置										
计费模式	按量计费	包年包月	详细对比 🖸							
	按量计费适合周期性任务场景,	. 不使用时可以挂起资源追	晚存在费用消耗造成浪费。	包年包月适合有稳定、持	续处理的任务需求的场景,无	需等待资源拉起。				
地域				华东地区		西南	地区	美国西部	亚太东南	
	3Ľ			上海	上海金融	成都	重庆	硅谷	新加坡	弗吉尼亚
	Btändes.									
	法兰克福									
	处于不同地域的云产品内网不可	互通,购买后不能更换,请	<b>總谨慎选择。</b> 建议选择最靠	155您客户的地域,可降低	访问时延。					
集群配置	处于不同地域的云产品内网不且	互通,购买后不能更换,请	<mark>總達慎选择,</mark> 建议选择最靠	15近忽客户的地域,可降低	访问时预测。					
集群配置	处于不同地域的云产品内网不正	互通,购买后不能更换,请	部 <mark>建慎选择,</mark> 建议选择最靠	ML您客户的地域,可降低	访问时证。					
集群配置 基本配置 计第引擎类型	处于不同地域的云产品内网不引 SparkSQL	巨通,购买后不能更换,请 Spark作L业	<mark>学校改革、</mark> 達から将着品 Presto	8508第户的地域,可降低	1508155.					
<b>集群配置</b> 基本配置 计算引聊类型	<mark>处于不同地域的云产品内网不</mark> SparkSQL 不同引擎支他不同就最处理场	<mark>王通, 购买后不能更换, 请</mark> Spark/行业 문, SparkSOL送用于稳定	学校正年、建议法将委員 Presto 高校的言語SOL任务、Span	8近想察户的地域,可降低	访问时越。 新心理,Prestol版 计交互式算	18分析。				
<b>集群配置</b> 基本配置 计算引擎类型	<b>处于不同地域的云户品内用不</b> <b>SparkSQL</b> 不可可 医文述不可能器化进行	<mark>互通、购买店不能更换、请</mark> Spark作业 最、SparkSQL通用于稳定	Presto 電効的単純SOL任务, Spar	NE你客户的地域,可降低 一 M作业适用于流动和式任务	访问时延。 时进,Prestor通 行交互式集	陶分析。				
<b>集群配置</b> 基本配置 计第列際类型 内板版本	处于不同地域的云产品内與不引 SparkSOL 자폐非常文地不同對最优进场 Spark 3.2	五通、明天后不能更施、清 Spark/fsil 展、SparkSOL版用于作会 Spark 2.4	(2)建筑法保、建心地将最新 (2) (2) (2) (2) (2) (2) (2) (2) (2) (2)	8년5章户的地域。可译任 《作业》图用于意动中式任命	访问时延。 B代理,Prestol版 于交互式重	<b>滚分析</b> 。				
<b>集群配置</b> 基本配置 计算可顺频型	<mark> 文子不可地域的云产品内與不</mark> SparkSOL 不可引擎文件不可能量的进行。 Spark 3.2 現作业甚子Span版专起,不	<ul> <li>正通、時天后不能更換、清</li> <li>Spark/行业</li> <li>Spark 2.4</li> <li>「現在本対点位表色不同、小</li> </ul>	(2)建筑选择, 建议总将最新 Presto 総分約率減50.15分、Spart 年間が数51表可参见、Spart 并間が数51表可参见、Spart	私在容易中的地域。可降低 **作业返用于完成地域に任	访问时延。 5处理,Prestol版,于交互式多	1933 ff.				
<b>集群配置</b> 基本配置 计算引擎类型	<mark> なす不可能域が云产品内障不引 SparkSOL</mark> 不同引 東支持不同数量化連続 Spark 3.2 我作业基于Spani版年載力、不	<ul> <li>転通、時天后不能更換、着</li> <li>Spark/FsyL</li> <li>Spark/SOLI進用于後空</li> <li>Spark 2.4</li> <li>同新年対应依款起不同、1</li> </ul>	(2)建築(法保)、建立法保養者 Presto 総合約素(核50)(任务、Spar 序句(成)(教)(大学)、Spar	12日日本 - 1946年 1月1日日 1月1日 1月1日 1月1日 1月111日 1月111日 1月111日 1月111日 1月1111 1月1111 1月1111 1月11111 1月111111 1月111111	访问时延。 再处理,Presto通。于交互式重	199 <b>%</b> .				
<b>集群配置</b> 基本配置 计算引辱类型 内板版本	<b>处于不同地域的云产品内闲不</b> <b>SparkSQL</b> 不同引集支持不可数量加速转 <b>Spark 3.2</b> 地作业基于Spark版年載力、不 <b>标准型</b>	五週、時天后不能更換、 第 Spark/F社 Spark 2.4 の時年7回位時起不同、 内存型	(2)建築(法保)、建立法保養者 Presto 単位(法)を、Spar 単位(法法)、Spar 単位(法法)、Spar	私日空間-台的地域。可称低 MPE型道用于流动機式任会 NT機械機構 (2	访问时延。 F处理,Presio版 于交互式重	199 <b>%</b> .				
▲本記置 计算引爆类型 内板版本 集群类型	<b>     此子不可能域的云户最内闲不</b> SparkSQL 不同計算支持不可能開始運動 取作並基于Spank版主範的、不 <b>所定型</b> Spark 3.2 取作並基于Spank版主範的、不 所定型 SparkCU - 164(408, 内容型	<ul> <li>1週、 時天后不能更迭、 着</li> <li>Spark(行业)</li> <li>Spark 2.4</li> <li>Spark 2.4</li> <li>内存型</li> <li>CU = 165086、不同共当</li> </ul>	(2) 建筑法保健、建立法保健者 中であり をなかる単純50に任务、Spar 単価が統予1時可参见、Spar 単価が存在不同、详細可参り	社会学校一部的地域。可称任 	访问时延。 Std理,Presto道 于交互式重	169 ff.				

信息配置 > 资源名称 项填写 "dlc-demo"。新建资源详细介绍请参见购买独享数据引擎。

高级配置			
9 XBLE	+ 80		
集群网段	10 255 0.016 <b>修改</b> 该配置会影响服务之间的网络互通性,在非联邦查询场展下,推荐使用默认配置:使用联邦查询时,引擎网站	這勿与数据源网段重复。	
信息配置			
资源名称	dic-demo		
描述信息			
	1/201750/1-3-14		
	运镇,长度限制为250		
标签	「新元時空 /		
	标签用于从不同健康对资源分类管理,如需了解更多,请前往 <b>标签产品文档</b> 2		

4. 单击**立即开通,**确认资源配置信息。

5. 确认信息无误后,单击**提交,**完成资源配置。

### 上传数据到 COS

创建名称为 "dlc-demo"的存储桶,上传 people.json 文件,用作从 COS 读写数据的示例,people.json 文件的内容如下:

```
{"name":"Michael"}
{"name":"Andy", "age":30}
{"name":"Justin", "age":3}
```



```
{"name":"WangHua", "age":19}
{"name":"ZhangSan", "age":10}
{"name":"LiSi", "age":33}
{"name":"ZhaoWu", "age":37}
{"name":"MengXiao", "age":68}
{"name":"KaiDa", "age":89}
```

1. 登录 对象存储 COS 控制台,在左侧菜单导航中单击 存储桶列表。

2. 创建存储桶:单击左上角创建存储桶,名称项填写 "dlc-dmo",单击下一步完成配置。

3. 上传文件:单击**文件列表 > 上传文件**,选择本地"people.json"文件上传到"dlc-demo-1305424723"桶里

(-1305424723是建桶时平台生成的随机串),单击上传,完成文件上传。新建存储桶详情可参见创建存储桶。

← 返回桶列表	dlc-demo-1305424723 /							Z
投表菜単名称 Q		件夹 文件碎片 清空存储桶 更多操	¢; ▼				在线编辑器 國	
文件列表 1	前缀搜索 ▼ 只支持部	素当前去拟目录下的对象	別新 共0个文件			每页 100 个对象	K < 1 → Ξ	
基础配置	文件名 #	上传文件		×	数时间 \$		操作	
安全管理		送择上传对象 之 设置对象	属性					
权限管理 ✓		3						
城名与传输管理 🚽		选择文件 上传至 dlc-demo-1305424723/						
容错容灾管理		若上传路径中存在同名文件,上传将覆盖原有文件。						
日志管理		上传操作将产生请求次数和上行流量, 其中请求 國務	·万次收费,上行流量 II 免费。详情请查看计费项 II 及产品定价 II					
内容审核 KOT · · ·		文件/文件夹	大小	操作				
数据处理		people.json	241.00B	删除				
任务与工作流 HOT >								
数据监控								
的数计算								
CVM 注载 COS								
			<b>多如西</b> <u>上传</u> 4					

### 新建 Maven 项目

- 1. 通过 IntelliJ IDEA 新建一个名称为"demo"的 Maven 项目。
- 2. 添加依赖:在 pom.xml 中添加如下依赖:

```
<dependency>
<groupId>org.apache.spark</groupId>
<artifactId>spark-core_2.12</artifactId>
<version>3.2.1</version>
<scope>provided</scope>
</dependency>
<dependency>
<groupId>org.apache.spark</groupId>
<artifactId>spark-sql_2.12</artifactId>
<version>3.2.1</version>
<scope>provided</scope>
</dependency>
```



### 编写代码

编写代码功能为从 COS 上读写数据和在 DLC 上建库、建表、查询数据和写入数据。 1. 从 COS 上读写数据代码示例:

```
package com.tencent.dlc;
import org.apache.spark.sql.Dataset;
import org.apache.spark.sql.Row;
import org.apache.spark.sql.SaveMode;
import org.apache.spark.sql.SparkSession;
public class CosService {
   public static void main( String[] args )
    {
       //1.创建SparkSession
       SparkSession spark = SparkSession
               .builder()
               .appName("Operate data on cos")
               .config("spark.some.config.option", "some-value")
               .getOrCreate();
        //2.读取cos上的json文件生成数据集,支持多种类型的文件,如 json,csv,parquet,orc,text
       String readPath = "cosn://dlc-demo-1305424723/people.json";
       Dataset<Row> readData = spark.read().json(readPath);
       //3.对数据集做业务计算操作生成结果数据,计算支持API和SQL形式,这里生成临时表用sql读数据
       readData.createOrReplaceTempView("people");
       Dataset<Row> result = spark.sql("SELECT * FROM people where age > 3");
       //4.结果数据保存到cos
        String writePath = "cosn://dlc-demo-1305424723/people_output";
       //写入支持多种类型的文件, 如 json, csv, parquet, orc, text
       result.write().mode(SaveMode.Append).json(writePath);
       spark.read().json(writePath).show();
       //5.关闭session
       spark.stop();
    }
}
```

2. DLC 上建库、建表、查询数据和写入数据:

package com.tencent.dlc; import org.apache.spark.sql.SparkSession; public class DbService {

```
🔗 腾讯云
```

```
public static void main(String[] args) {
    //1.初始化SparkSession
    SparkSession spark = SparkSession
            .builder()
            .appName("Operate DB Example")
            .getOrCreate();
    //2.建数据库
    String dbName = " `DataLakeCatalog`.`dlc_db_test` ";
    String dbSql = "CREATE DATABASE IF NOT EXISTS" + dbName + " COMMENT 'demo t
    spark.sql(dbSql);
    //3.建内表
    String tableName = "`test`";
    String tableSql = "CREATE TABLE IF NOT EXISTS " + dbName + "." + tableName
            + "(`id` int,`name` string, `age` int)";
    spark.sql(tableSql);
    //4.写数据
    spark.sql("INSERT INTO " + dbName + "." + tableName + "VALUES (1, 'Andy', 12)
    //5.查询数据
    spark.sql(" SELECT * FROM " + dbName + "." + tableName).show();
    //6.建外表
    String extTableName = "`ext_test`";
    spark.sql(
            "CREATE EXTERNAL TABLE IF NOT EXISTS " + dbName + "." + extTableNam
                    + " (`id` int, `name` string, `age` int) "
                    + "ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerD
                    + "STORED AS TEXTFILE LOCATION 'cosn://dlc-demo-1305424723/
    //7.写外表数据
    spark.sql("INSERT INTO " + dbName + "." + extTableName + "VALUES (1,'LiLy',
    //8.查询外表数据
    spark.sql(" SELECT * FROM " + dbName + "." + extTableName).show();
    //9.关闭Session
    spark.stop();
}
```

建外表时,需按照上传数据到COS的步骤先在桶里建对应表名文件夹保存表文件

}



dlc-demo-1305424	723 /							文林
上传文件	1           创建文件夹         文件碎片	<b>清空存储桶</b> 更多操作	v					在线编辑器
前缀搜索 ▼	只支持搜索当前虚拟目录下的对象	<b>©</b> Q	刷新	共1个文件		每页 100 个对象	₩ 4 1 ►	
文件名 \$		大小 \$		存储类型 🔻	修改时间 💲		操作	
ext_test/	1						设置权限 统计	● 更多 ▼

### 调式、编译代码并打成 JAR 包

通过 IntelliJ IDEA 对 demo 项目编译打包, 在项目 target 文件夹下生成JAR包 demo-1.0-SNAPSHOT.jar。

### 上传 JAR 包到 COS

登录 COS 控制台,参考 上传数据到 COS 的步骤将 demo-1.0-SNAPSHOT.jar 上传到 COS。

### 新建 Spark Jar 数据作业

创建数据作业前,您需先完成数据访问策略配置,保证数据作业能安全地访问到数据。配置数据访问策略详情请参见配置数据访问策略。如已配置数据策略名称为:qcs::cam::uin/100018379117:roleName/dlc-demo。

1. 登录数据湖计算 DLC 控制台,选择服务所在区域,在导航菜单中单击数据作业。

2. 单击左上角创建作业,进入创建页面。

3. 在作业配置页面, 配置作业运行参数, 具体说明如下:

配置参数	说明
作业名称	自定义 Spark Jar 作业名称,例如:cosn-demo
作业类型	选择 <b>批处理类型</b>
数据引擎	选择 创建资源 步骤创建的 dlc-demo 计算引擎
程序包	选择 COS,在 上传 JAR 包到 COS 步骤上传的 JAR 包 demo-1.0-SNAPSHOT.jar
主类(Main Class)	根据程序代码填写,如: 从 COS 上读写数据填: com.tencent.dlc.CosService 在 DLC 上建库、建表等填: com.tencent.dlc.DbService
数据访问策略	选择该步骤前创建的策略 qcs::cam::uin/100018379117:roleName/dlc-demo



### 其他参数值保持默认。

编辑作业		
基本信息 ▲ 作业名称 *	<b>cosn_demo</b> 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 2 查看管理。数据引擎的网络	各配置信息可至网络配置 🖸
程序包 *	◯ 对象存储COS ○ 本地上传	
	cosn://1305424723/demo-1.0-SNAPSHOT.jar	选择COS位置
	需具备COS相关权限,可选择jar/py文件	
主类(Main Class) *	com.tencent.dlc.CosService	
程序入口参数	请输入程序入口参数,多个参数空格分隔。不超过65536个字符	
作业参数 (config)	示例: spark.network.timeout=120s	
保存取	消	

4. 单击保存,在Spark 作业页面可以看到创建的作业。

### 运行并查看作业结果

1. 运行作业:在Spark 作业页面,找到新建的作业,单击运行,即可运行作业。

2. 查看作业运行结果: 可查看作业运行日志和运行结果。

### 查看作业运行日志

1. 单击**作业名称 > 历史任务**,查看任务运行状态。



数据作业 🤇	〕广州 ▼				Spark作业详情						
Spark作业	作业配置 Session	管理			作业信息	历史任务	2 监控告警				
创建作业		Q	全部	请选择创建人	请选择执行状;	لــــــــــــــــــــــــــــــــــــ	近7天 近30天	2023-0	)7-07 ~ 2023-07-13		
作业名称	作业ID	作业类型	作业文件	当前任务数	任务ID 3	执行状态	任务提交时间 🗲	计算耗时	操作		
<b></b> 6	r <mark>a</mark>	-		0	ed56fcd2-5		2023-07-12 16:04:48	2min36s	查看详情 Spark UI		
6	6			0	-				查看详情 Spark UI		
_1									查看详情 Spark UI		
	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~			0					查看详情 Spark UI		
cosn_demo 🖬	batch_366ae4	批处理	cosn: 1305424	1	共 4 条				10 ▼ 条/页	₩ 4 1	/1页 )

2. 单击任务ID > 运行日志, 查看作业运行日志:



基本信息 作业名称				Б			
控制台最多展	示最近1000条	·					
近7天	近30天	2023-08-01 17:36:02	~ 2023-08-07 17:36:02	i đ	安时间降序 🔻	φ	创建下载
日志名称:			• •	日志级别:	All		•
23/08/07 1 +++   2 Lucy  3	7:32:24 INFO \$ -+ 3	SparkUI: Stopped Spark	lulerBackend: Shutting dov web UI at http://spark-5f64	vn all execu	tors 6bc-driver-svc.d	efault.svc:	4040
23/08/07 1 +++   2 Lucy  3   1 LiLy  12 +++	7:32:24 INFO \$ -+ 3  2  -+	SparkUI: Stopped Spark	lulerBackend: Shutting dov	vn all execu	tors 6bc-driver-svc.d	efault.svc:	4040
23/08/07 1 +++   2 Lucy  3   1 LiLy  12 +++   id name a	7:32:24 INFO \$ -+ 3  2  -+ ge	SparkUI: Stopped Spark	lulerBackend: Shutting dov	vn all execu	tors 6bc-driver-svc.d	efault.svc:	4040
23/08/07 1 +++   2 Lucy  3   1 LiLy  12 +++   id name a +++	7:32:24 INFO \$ -+ 3  2  -+ ge  -+	SparkUI: Stopped Spark	iulerBackend: Shutting dov	vn all execu 1289cf55a6	tors Sbc-driver-svc.d	efault.svc:	4040
23/08/07 1 +++   2 Lucy  3   1 LiLy  12 +++   id name a +++ 23/08/07 1 23/08/07 1	7:32:24 INFO \$ -+ 3  2  -+ ge  -+ 7:32:24 INFO   7:32:24 INFO	SparkUI: Stopped Spark DAGScheduler: Job 5 fin TaskSchedulerImpl: Killin	lulerBackend: Shutting dov web UI at http://spark-5f64 ished: show at DbService.	vn all execu 1289cf55a6 java:37, tool e 5: Stage fi	tors Sbc-driver-svc.d k 0.161365 s nished	efault.svc:	4040
23/08/07 1 +++   2 Lucy  3   1 LiLy  12 +++   id name a +++ 23/08/07 1 23/08/07 1	7:32:24 INFO \$ -+ 3  2  -+ 7:32:24 INFO   7:32:24 INFO   7:32:24 INFO	SparkUI: Stopped Spark DAGScheduler: Job 5 fin TaskSchedulerImpl: Killin DAGScheduler: Job 5 is	ished: show at DbService. ig all running tasks in stage	vn all execu 1289cf55a6 java:37, tool e 5: Stage fii tial speculati	tors Sbc-driver-svc.d k 0.161365 s nished ive or zombie ta	efault.svc:	4040 s job
23/08/07 1 +++   2 Lucy  3   1 LiLy  12 +++   id name a +++ 23/08/07 1 23/08/07 1 23/08/07 1	7:32:24 INFO \$ -+ 3  2  -+ 7:32:24 INFO   7:32:24 INFO   7:32:24 INFO	DAGScheduler: Job 5 fin TaskScheduler: Job 5 fin DAGSchedulerImpl: Killin DAGScheduler: Job 5 is	lulerBackend: Shutting dov web UI at http://spark-5f64 ished: show at DbService. ig all running tasks in stage finished. Cancelling potent age 5 (show at DbService	vn all execu 1289cf55a6 java:37, tool e 5: Stage fin tial speculati .java:37) fini	tors Sbc-driver-svc.d k 0.161365 s nished ive or zombie ta ished in 0.157 s	efault.svc: asks for this	4040 s job

1. 运行从 COS 读写数据示例,则到 COS 控制台查看数据写入结果。



← 返回桶列表		-1305424723 / people_output					
搜索菜单名称	Q	上传文件 创建文件夹 更多操作 ▼					在线编
概览							
文件列表		前缀搜索 ▼ 只支持搜索当前虚拟目录下的对象	☑ Q 刷新	共2个文件	每页 100 个对象		
基础配置	~	文件名 \$	大小 \$	存储类型 🔻	修改时间 💠	操作	
安全管理	~	_SUCCESS 🖋 🗖	0.00B	标准存储	2023-07-12 16:51:49	详情 预览	下载
权限管理	~						
は々とは絵範囲	~	part-00000-4695aae9-f286-4e97-8d97-bcf190f16910-c000.csv	11.00B	标准存储	2023-07-12 16:51:48	详情 预览	下载

2. 运行在 DLC 上建表、建库,则到 DLC 数据探索页面查看建库、建表。

数据探索 🔇 广州 👻	SOLI	语法参考 🖸
<b>库表</b> 查询 🗘 +	ख़ात sst_sql ● + ▼	
数据目录 DataLakeCatalog ▼	部分运行 🖾 保存 🖸 刷新 💼 格式化 😡 😂 请选择默认数据库 ▼ 🔚 fa	ırley-test(Suj
dlc_db_test	<pre>33 select * from `DataLakeCatalog`.`dlc_db_test`.`test` 34 35 select * from `DataLakeCatalog`.`dlc_db_test`.`ext_test`</pre>	
<ul> <li>● 82 初題</li> <li>▶ 62 函数</li> </ul>	查询结果     统计数据       Task ID SOLi洋倩     号出结果     优化建议       查询邦时     2.75s     数据扫描量     21 B       共2<	运行
	id name age	
	1 Andy 12	
	2 Justin 3	



# PySpark 作业开发指南

最近更新时间:2025-03-07 15:52:30

### 应用场景

DLC 支持 Python 语言编写的程序运行作业。本示例演示通过编写 Python 代码在 对象存储(COS)上读写数据和在 DLC 上建库表、读写表的详细操作,帮助用户在 DLC 上完成作业开发。

### 环境准备

依赖: PyCharm 或其他 Python 编程开发工具。

### 开发流程

### 开发流程图

DLC Spark JAR 作业开发流程图如下:



### 创建资源

第一次在 DLC 上运行作业,需新建 Spark 作业计算资源,例如新建名称为 "dlc-demo" 的 Spark 作业资源。

1. 登录数据湖计算 DLC 控制台,选择服务所在区域,在导航菜单中单击数据引擎。

2. 单击左上角创建资源,进入资源配置购买页面。

3. 在集群配置 > 计算引擎类型选项选择 Spark 作业引擎。



>~J/[]/9] V										
基础配置										
计费模式	按量计费	包年包月	详细对比 🛛							
	按量计费适合周期性任务场累,	不使用时可以挂起资源递	<sup>整</sup> 免存在费用消耗适成浪费。	包年包月适合有稳定、持	续处理的任务需求的场景,无	需等待资源拉起。				
地域			_	华东地区		西南	地区	美国西部		
	北			上海	上海金融	成都	重庆	硅谷	新加坡	弗吉尼亚
	Bt%#\$									
	法兰克福									
	法兰克福处于不同地域的云产品内网不可	<b>三通,购买后不能更换,</b> 诸	<b>青恕谨慎选择。</b> 建议选择最常	11近您客户的地域,可降低	访问时班。					
41:2427.02	法兰克福	且通,购买后不能更换,请	<b>弗尔道慎选择。</b> 建议选择最新	副后您客户的地域,可降低	行问时就后。					
集群配置	法兰克福	豆通,购买后不能更换,诸	<mark>尊欲達慎选择、</mark> 建议选择最新	<u>敏近您客户的地域,可</u> 得低	访问时起。					
<b>集群配置</b> 基本配置	法兰弗福	互通,购买后不能更换,读	<b>客心谨慎选择,</b> 建议选择最繁	<b>到近你客户的地域,可降低</b>	初日时就。					
集群配置 基本配置 计第引擎类型	法兰莞福 处于不同地域的云产品内网不到 SparkSQL	巨通,购买后不能更换,诸 Spark作业	<mark>春空谨慎选择。</mark> 建议选择最新 Presto	<u>新活思馨</u> 产的地域,可降低	12510199785.					
<b>集群配置</b> 基本配置 计算引擎类型	法兰费福 处于不同地域的云产品内网不可 SparkSQL 不同引擎支持不同就语处理场器	五通、 駒天后不能更快、 諸 Spark作业 副、 SparkSQL道用于稳定	<mark>春尔谨慎选择,</mark> 建议选择最新 Presto 海觉的高级SQL任务,Spa	14.50%客户的地域。可降低 14.50%。 14.50%。14.50%。14.50%。14.50%。14.50%。14.50%。14.50%。14.50%。14.50%。14.50%。14.50%。14.50%。14.50%。14.50%。14.50%	访问时延。 时处理,Presto道,行交互式)	重运分析。				
<b>集群配置</b> 基本配置 计第9家类型	法兰费福 处于不同地域的云产品内网不到 SparkSQL 不同引擎支持不可数据处理场路	日道、開天后不能更換、彼 Sparkfs业 長、SparkSQL版用于保定	など連続には、 違い也得要的 Presto 変が的事代SOL任务、Span	勤后於聯合的始始。可降低 1996年 1997年 1997年 1997年 1997年 1997年 1997年 1997年 1997年 1997年 1997年 1997年 1997年 1997年 1996年 1996年 1996年 1996年 1996年 1996年 1996年 1996年 1996年 1996年 1996年 1996年 1996年 1996年 1996年 1996年 1996年 1996年 1996年 1996年 1996年 1996年 1996年 1996年 1996年 1996年 1996年 1996年 1996年 1996年 1996年 1996年 1996年 1996年 1996年 1996年 1996年 1997 1997 1997 1997 1997 1997 1997 199	访问时猛。 R处理,Prestolla 开交互式	副局分析。				
<b>集群配置</b> 基本配置 计算引家类型	法兰党福 处于不同地域的云产品内网不到 SparkSQL 不同引擎支持不同数据处理场器	E通、购完后不能更换、留 Spark(行业 E、SparkSQL适用于信定 Spark 2.4	<b>1923年105年</b> 、違い送岸最新 <b>Presto</b> 電気効素成501に分・Spa	NEOSA-BOBA, JAG NEOSA-BOBA, JAG	的同时能。 RM是 Presiou推 于文王式	<b>王国分析</b> 。				
<b>集群配置</b> 基本配置 计算引导类型	<b>法兰党福</b> 处于不同地域的云产品内网不到 <b>SparkSQL</b> 不同引擎支持不同数据处理场器 <b>Spark 3.2</b> 批作业盛于Spark版本能力、不	EIL, 對大后不能更終。 Sparkfby B, SparkSQL适用于稳定 Spark 2.4 词版本对点说表起不同。	<ul> <li>(1) (1) (1) (1) (1) (1) (1) (1) (1) (1)</li></ul>	1925年から出来、可称任 小作业道用子売式税式任会 nt环境時間 2	的同时能。 我处理,Prestou能 于交互式	<b>王氏分析</b> 。				
<b>集群配置</b> 基本配置 计算引导类型 内核版本	法兰党福 处于不同地域的云产品内网不到 SparkSQL 不同引擎支持不同数据处理语言 文件上述基于Spark版本能力,不	EIL,解究后不能更换,读 Sparkfeit EL、SparkSOL适用于稳定 Spark 2.4 词版本对应依赖也不同,	<ul> <li>(1) (1) (1) (1) (1) (1) (1) (1) (1) (1)</li></ul>	1958年始始逝,可降后 小作业道用于流动规式任命	说明时能。 乾燥、Prestoum 于文三式	王成分析,				
集群配置 基本配置 计算引导类型 内板版本 集群类型	法兰费福 处于不同地域的云产品内网不到 SparkSQL 不同引 家交持不同数据处理语言 犯作业强于Spark怎本能力,不 码准型	EIL,對天后不能更強,當 Spark作业 El、SparkSQL道用于稳定 Spark 2.4 词版本对应该数也不同, 内存型	<ul> <li>(1) (1) (1) (1) (1) (1) (1) (1) (1) (1)</li></ul>	社经容察产始始越,可降低 小作业适用于流动和武任者 n5环境进程 [2	的时候。 乾燥、Presida 于文王式	王明分析。				
集群配置 基本配置 计算引导映型 内级版本	法兰党福 处于不同地域的云产品内网不到 SparkSQL 不同引擎支持不同数据处理语言 处作业基于Spark系和 规作业基于Spark系和 系力	3. 购完后不能更美。前           Spark(于业)           8. SparkSQL道用于稳定           Spark 2.4           "调版本对应依景色不调,           内存型           HOU > 1t6308, 不同送起	また、 学校の 学校の また、 なた、 なた、 なた、 なた、 なた、 なた、 なた、 な	11日の事件的地域、可降低 小市业活用于流动和式任务 れ研究規則【 の: 计表数注【	的明时廷。 Prestor 新子交互式	2001f.				

信息配置 > 资源名称填写 "dlc-demo"。新建资源详细介绍请参见购买独享数据引擎。

高级配置		
参数配置	+版11	
集群网段	10.255.0.016 <b>修改</b> 该配置会影响服务之间的间络互通性。在4联邦查询场展下,推荐使用款认配置:使用联邦查询时,引擎网级语勿与数据源网级重复。	
信息配置		
资源名称	dic-demo	
描述信息	不超过250个字符	
	运稿。长速网机为250	
标签	123月、153月11日77-2500	

4. 单击**立即开通**,确认资源配置信息。

5. 确认信息无误后,单击**提交**,完成资源配置。

### 上传数据到 COS

创建名称为 "dlc-demo"的存储桶,上传people.json文件,供从COS 读写数据示例用,people.json 文件的内容如下:

```
{"name":"Michael"}
{"name":"Andy", "age":30}
{"name":"Justin", "age":3}
```



```
{"name":"WangHua", "age":19}
{"name":"ZhangSan", "age":10}
{"name":"LiSi", "age":33}
{"name":"ZhaoWu", "age":37}
{"name":"MengXiao", "age":68}
{"name":"KaiDa", "age":89}
```

1. 登录 对象存储 COS 控制台,在左侧菜单导航中单击 存储桶列表。

2. 创建存储桶:

单击左上角 创建存储桶,名称项填写 "dlc-dmo",单击下一步完成配置。

3. 上传文件:

单击**文件列表 > 上传文件**,选择本地"people.json"文件上传到"dlc-demo-1305424723"桶里(-1305424723是建桶时 平台生成的随机串),单击**上传**,完成文件上传。新建存储桶详情请参见创建存储桶。

← 返回桶列表	dic-demo-1305424723 /							Ż
捜索菜単名称 Q 概処		件夫 文件碎片 <b>清空符储桶</b> 更多操作	•				在线编辑器 🔤	
文件列表 1		建秦当前虚拟目录下的对象 🛛 Q	刷新 共0个文件			每页 100 个对象	K < 1 → Ξ	
基础配置	文件名 ≄	上传文件		×	牧时间 \$		操作	
安全管理		→ 选择上传对象 2 设置对象属性						
权限管理    ✓		<u> </u> ۲						
域名与传输管理 ✓		选择文件 上传至 dlc-demo-1305424723/						
容错容灾管理		若上传路径中存在同名文件,上传将覆盖原有文件。						
日志管理		上传操作将产生请求次数和上行流量, 其中请求 🗹 按万次战	z费,上行流量 LE 免费。详情请查看计费项 LE 及产品定价 LE					
内容审核 NOT ・		文件/文件夹 フ	大小操	Pe				
数据处理		people.json 2	241.00B ##	\$				
任务与工作流 HOT >								
数据监控								
函数计算								
CVM 挂载 COS								
		参数	<u>武</u> (上传)					

### 新建 Python 项目

通过 PyCharm 新建一个名称为"demo"的 项目。

### 编写代码

1. 新建 cos.py 文件,编写代码,功能为从 COS 上读写数据和在 DLC 上建库、建表、查询数据和写入数据。

```
import sys
from pyspark.sql import SparkSession
from pyspark.sql import Row
if __name__ == "__main__":
    spark = SparkSession\\
    .builder\\
```



```
.appName("Operate data on cos") \\
         .getOrCreate()
     # 1.读cos上的数据 支持多种类型的文件 如 json,csv,parquet,orc,text
     read_path = "cosn://dlc-demo-1305424723/people.json"
     peopleDF = spark.read.json(read_path)
     # 2.对数据做操作
     peopleDF.createOrReplaceTempView("people")
     data_src = spark.sql("SELECT * FROM people WHERE age BETWEEN 13 AND 19")
     data src.show()
     # 3. 写数据
     write_path = "cosn://dlc-demo-1305424723/people_output"
     data_src.write.csv(path=write_path, header=True, sep=",", mode='overwrite')
     spark.stop()
2. 新建 db.py 文件,编写代码,功能为 DLC 上建库、建表、查询数据和写入数据。
 from os.path import abspath
 from pyspark.sql import SparkSession
 if __name__ == "__main__":
     spark = SparkSession \\
         .builder \\
         .appName("Operate DB Example") \\
         .getOrCreate()
     # 1.建数据库
     spark.sql("CREATE DATABASE IF NOT EXISTS `DataLakeCatalog`.`dlc_db_test_py` COM
     # 2.建内表
     spark.sql("CREATE TABLE IF NOT EXISTS `DataLakeCatalog`.`dlc_db_test_py`.`test`
     # 3.写内数据
     spark.sql("INSERT INTO `DataLakeCatalog`.`dlc_db_test_py`.`test` VALUES (1,'And
     # 4.查内数据
     spark.sql("SELECT * FROM `DataLakeCatalog`.`dlc_db_test_py`.`test` ").show()
     # 5.建外表
     spark.sql("CREATE EXTERNAL TABLE IF NOT EXISTS `DataLakeCatalog`.`dlc_db_test_p
     # 6. 写外数据
     spark.sql("INSERT INTO `DataLakeCatalog`.`dlc_db_test_py`.`ext_test` VALUES (1,
```

```
©2013-2025 Tencent Cloud International Pte. Ltd.
```

# 7.查外数据



spark.sql("SELECT \* FROM `DataLakeCatalog`.`dlc\_db\_test\_py`.`ext\_test` ").show(
spark.stop()

建外表时,可按照 上传数据到 COS 的步骤 先在桶里建对应表名文件夹保存表文件。

llc-demo-1305424	723 /				ک
上传文件	创建文件夹文件码	<b>許計 清空存储桶</b> 更多操作 ▼			在线编辑
前缀搜索 ▼	只支持搜索当前虚拟目录下的	对象 <b>2</b> 刷新	共1个文件	每页 100 个对象	₩ 4 1 >
文件名 \$		大小 ≄	存储类型 🕈	修改时间 🕈	操作
ext_test/	1		-		设置权限统计更多

调式

PyCharm 调式无语法错误。

### 上传 py 文件到 COS

登录 COS 控制台,参考上文上传数据到 COS 的步骤将 cos.py、db.py上传到 COS。

### 新建 Spark Jar 数据作业

创建数据作业前,您需先完成数据访问策略配置,保证数据作业能安全地访问到数据。配置数据访问策略详情请参见配置数据访问策略。如已配置数据策略名称为:qcs::cam::uin/100018379117:roleName/dlc-demo

1. 登录数据湖计算 DLC 控制台,选择服务所在区域,在导航菜单中单击数据作业。

2. 单击左上角创建作业按钮,进入创建页面。

3. 在作业配置页面, 配置作业运行参数, 具体说明如下:

配置参数	说明
作业名称	自定义 Spark 作业名称,例如:cosn_py
作业类型	选择 <b>批处理类型</b>
数据引擎	选择 创建资源 步骤创建的 dlc-demo 计算引擎
程序包	选择 COS,在 <b>上传 py 文件到 COS</b> 步骤的上传 py 文件: 从 COS 上读写数据就选择: cosn://dlc-demo-1305424723/cos.py



 在 DLC 上建库、建表等选择:cosn://dlc-demo-1305424723/db.py

 数据访问策略
 选择该步骤前创建的策略 qcs::cam::uin/100018379117:roleName/dlc-demo

其他参数值保持默认。

编辑作业		
基本信息 ▲		
作业名称*	db_py	
	支持中文、英文、数字与"_", 最多100个字符	
作业类型*	<b>批处理</b> 流处理 SQL作业	
数据引擎*	(按量计费)	
	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	息可至网络配置
程序包 *	O 对象存储COS ○ 本地上传	
	cosn:/, -1305424723/db.py 选择C	OS位置
	需具备COS相关权限,可选择jar/py文件	
程序入口参数	请输入程序入口参数,多个参数空格分隔。不超过65536个字符	
作业参数 (config)	示例: spark.network.timeout=120s	
保存	取消	

4. 单击保存,在 Spark 作业页面可以看到创建的作业。

### 运行并查看作业结果

1. 运行作业:在Spark 作业页面,找到新建的作业,单击运行,即可运行作业。
 2. 查看作业运行结果:可查看作业运行日志和运行结果。

### 查看作业运行日志

1. 单击**作业名称 > 历史任务** 查看任务运行状态:



<b>发据作业</b>	○ 广州 ▼				Spark作业详情					
Spark作业	作业配置 Session	管理			作业信息	2 历史任务	监控告警			
创建作业		Q	全部	▼ 请选择创建人	请选择执行状态		近7天	近30天	2023-07-08 ~ 2023-07-14	
作业名称	作业ID	作业类型	作业文件	当前任务数	任务ID 2	执行状态	任务提交时间	<b>\$</b> ंग	算耗时 操作	
db_py 🚡	batch_3ae998	批处理	cosn://c1305424	0	02f21ed4-1	成功	2023-07-12 1	7:54:24 1n	min47s 查看详情 Spark UI	
cons_py 🗗	batch_df8ee8 <b>Г</b>	批处理	cosn:/ -1305424	0	565ecb8a-2	成功	2023-07-12 1	7:48:19 1n	nin46s 查看详情 Spark UI	
					b28c614d-4	失败	2023-07-12 1	7:45:42 47	7.2s 查看详情 Spark UI	

2. 单击任务ID > 运行日志,查看作业运行日志。



基本信息	运行日志						
作业名称 <sup>坎判公是名用</sup>	est <b>□</b> 作	F业ID:		6			
近7天	近30天	四本 2023-08-01 17:36:(	02 ~ 2023-08-07 17:36:0	2	按时间降序 ▼	¢	创建下载任务
日志名称:			ferials Lana	日志级别	J: All		•
+++	7:32:24 INFO \$ .+	SparkUI: Stopped Spa	rk web UI at http://spark-5	641289cf55a	a6bc-driver-svc.d	efault.svc:	4040
+++   2 Lucy  3   1 LiLy  12 +++	7:32:24 INFO \$ .+ }  !  .+	SparkUI: Stopped Spa	rk web UI at http://spark-5	641289cf55a	a6bc-driver-svc.d	efault.svc:	4040
+++   2 Lucy  3   1 LiLy  12 +++   id name ar	7:32:24 INFO \$ .+ 3  '  .+ ge  .+	SparkUI: Stopped Spa	rk web UI at http://spark-5	641289cf55a	a6bc-driver-svc.d	efault.svc:	4040
23/08/07 17 +++   2 Lucy  3   1 LiLy  12 +++   id name a +++ 23/08/07 17	7:32:24 INFO 5 -+ 3  -+ ge  -+ 7:32:24 INFO [	SparkUI: Stopped Spa DAGScheduler: Job 5	finished: show at DbServio	e.java:37, to	a6bc-driver-svc.d	efault.svc:	4040
23/08/07 17 +++   2 Lucy  3   1 LiLy  12 +++   id name ar +++ 23/08/07 17 23/08/07 17	7:32:24 INFO \$ .+ 3  :+ ge  .+ 7:32:24 INFO [ 7:32:24 INFO ]	SparkUI: Stopped Spa DAGScheduler: Job 5 TaskSchedulerImpl: Kil	finished: show at DbServio lling all running tasks in sta	e.java:37, to age 5: Stage	a6bc-driver-svc.d nok 0.161365 s finished	efault.svc:	4040
23/08/07 17 +++   2 Lucy  3   1 LiLy  12 +++   id name ar +++ 23/08/07 17 23/08/07 17	7:32:24 INFO \$ .+ 3  2  .+ 7:32:24 INFO [ 7:32:24 INFO ] 7:32:24 INFO [	SparkUI: Stopped Spa DAGScheduler: Job 5 TaskSchedulerImpl: Kil DAGScheduler: Job 5	finished: show at DbServio lling all running tasks in sta	e.java:37, to age 5: Stage	a6bc-driver-svc.d a6bc - driver-svc.d ook 0.161365 s finished ative or zombie ta	efault.svc:	4040 s job
23/08/07 17 +++   2 Lucy  3   1 LiLy  12 +++   id name a +++ 23/08/07 17 23/08/07 17 23/08/07 17	7:32:24 INFO \$ .+ 3  2  .+ 7:32:24 INFO [ 7:32:24 INFO [ 7:32:24 INFO [ 7:32:24 INFO [	SparkUI: Stopped Spa DAGScheduler: Job 5 TaskSchedulerImpl: Kil DAGScheduler: Job 5 DAGScheduler: Result	finished: show at DbServio lling all running tasks in sta is finished. Cancelling pote	ee.java:37, to age 5: Stage ential specula ce.java:37) fi	utors a6bc-driver-svc.d ok 0.161365 s finished ative or zombie ta nished in 0.157 s	efault.svc:	4040 s job

1. 运行从 COS 读写数据示例,则到 COS 控制台查看数据写入结果。



← 返回桶列表		-1305424723 / people_output				
搜索菜单名称	Q	上传文件				在线编
既览		前缀搜索 ▼ 只支持搜索当前虚拟目录下的对象	② Q 刷新	共 2 个文件	每页 100 个对象	
文件列表						
基础配置	~	又件名 \$	大小 \$	存储类型 『	修改时间 \$	操作
安全管理	~	_SUCCESS 🖍 🗖	0.00B	标准存储	2023-07-12 16:51:49	详情 预览 下载
权限管理	~					
ま々とは絵範囲	÷	part-00000-4695aae9-f286-4e97-8d97-bcf190f16910-c000.csv	11.00B	标准存储	2023-07-12 16:51:48	详情 预览 下载

2. 运行在 DLC 上建表、建库,则到 DLC 数据探索页面查看建库、建表。

数据探索		
<b>库表</b> 查询 🗘 +	调度-cry_test_sql ● + ▼	
数据目录 DataLakeCatalog ▼	部分运行 日 保存 〇、刷新 白 格式化 気	请选择默认;
dlc_db_test_py 💌	<pre>32 33 select * from `DataLakeCatalog`.`dlc_db_test_py`.`test`</pre>	
请输入表名称	34 35	
▼	36 37	
▼ 囲 表		
□   ext_test     □   test		
▶ 品 视图		
▶ 四 函数 —	Task ID SQL详情 导出结果优化建议       ☑         查询耗时 10.43s 数据扫描量 21 B       共2条数据(控制台最多可展示1000条数据)复制数据 □	
	id name	age
	1 Andy	12
	2 Justin	3



# 查询性能优化指南

最近更新时间:2025-03-07 15:52:30

### 前言

为了提升任务执行效率,DLC 引擎在计算过程中有许多优化措施,例如数据治理、lceberg 索引、缓存等。正确使用 不仅可以减少不必要的扫描费用,甚至可以提升几倍甚至几十倍的效率。下面提供一些不同层面的优化思路。

### 优化 SQL 语句

场景:SQL 语句本身不合理,导致执行效率不高。

### 优化 JOIN 语句

当查询涉及 JOIN 多个表时, Presto 引擎会优先完成查询右侧的表的 JOIN 操作,通常来说,先完成小表的 JOIN, 再用结果集和大表进行 JOIN,执行效率会更高,因此 JOIN 的顺序会直接影响查询的性能,DLC presto 会自动收集 内表的统计数据,利用 CBO 对查询中的表进行重排序。

对于外表,通常用户可以通过analyze语句完成统计数据的收集,或者手动指定 JOIN 的顺序。如需手动指定请按表的大小顺序,将小表放在右侧,大表放在左侧,如表A > B > C,例如:select \* from A Join B Join C。需要注意的 是,这不能保证所有场景下都能提升效率,实际上这取决于 JOIN 后的数据量大小。

### 优化 GROUP BY 语句

合理安排 GROUP BY 语句中字段顺序对性能有一定提升,请根据聚合字段的基数从高到低进行排序,例如:

```
//高效的写法
SELECT id,gender,COUNT(*) FROM table_name GROUP BY id, gender;
//低效的写法
SELECT id,gender,COUNT(*) FROM table_name GROUP BY gender, id;
```

另一种优化方式是,尽可能地使用数字代替具体分组字段。这些数字是 SELECT 关键字后的列名的位置,例如上面的 SQL 可以用以下方式代替:

SELECT id,gender,COUNT(\*) FROM table\_name GROUP BY 1, 2;

### 使用近似聚合函数

对于允许有少量误差的查询场景,使用这一些近似聚合函数对查询性能有大幅提升。 例如,Presto可以使用 APPROX\_DISTINCT()函数代替 COUNT(distinct x),Spark 中对应函数为 APPROX\_COUNT\_DISTINCT。该方案缺点是近似聚合函数有大概2.3%的误差。



### 使用 REGEXP\_LIKE 代替多个 LIKE

当 SQL 中有多个 LIKE 语句时,通常可以使用正则表达式来代替多个 LIKE,这样可以大幅提升执行效率。例如:

SELECT COUNT(\*) FROM table\_name WHERE field\_name LIKE '%guangzhou%' OR LIKE '%beiji

可以优化成:

SELECT COUNT(\*) FROM table\_name WHERE regexp\_like(field\_name, 'guangzhou|beijing|ch

### 数据治理

#### 数据治理适用场景

场景:实时写入。Flink CDC 实时写入通常采用 upsert 的方式写入,该流程在写入过程中会产生大量的小文件,当 小文件堆积到一定程度后会导致数据查询变慢,甚至超时无法查询。 可以通过以下方式查看表文件数量和快照信息。

SELECT COUNT(\*) FROM [catalog\_name.][db\_name.]table\_name\$files; SELECT COUNT(\*) FROM [catalog\_name.][db\_name.]table\_name\$snapshots;

例如:

```
SELECT COUNT(*) FROM `DataLakeCatalog`.`db1`.`tb1$files`;
SELECT COUNT(*) FROM `DataLakeCatalog`.`db1`.`tb1$snapshots`;
```

表文件、快照数量过多时,可以参考文档开启数据优化 启用数据治理功能。

#### 数据治理效果

开启数据治理后,查询效率得到显著提升,例如下表对比了合并文件前后的查询耗时,该实验采用16CU presto,数据量为14M,文件数量2921,平均每个文件0.6KB。

执行语句	是否合并文件	文件数量	记录条数	查询耗时	效果
SELECT count(*) FROM tb	否	2921个	7895条	32s	油麻林02%
SELECT count(*) FROM tb	是	1个	7895条	2s	<b>歴度 仄33</b> %

分区



分区能够根据时间、地域等具有不同特征的列值将相关数据分类存储,这有助于大幅减少扫描量,提升查询效率。 关于 DLC 外表分区更多详情信息,请参考一分钟入门分区表。下表展示了在数据量为66.6GB,数据记录为14亿条, 数据格式为 orc 的单表中,分区和不分区时查询耗时和扫描量的效果对比。其中 dt 是含有1837个分区的分区字段。

本询语句	未分区		分区		耗时对	扫描量对比	
重肉佔內	耗时	扫描量	耗时	扫描量	比	扫捆重刈比	
SELECT count(*) FROM tb WHERE dt='2001-01-08'	2.6s	235.9MB	480ms	16.5 KB	快81%	少99.9%	
SELECT count(*) FROM tb WHERE dt<'2022-01-08' AND dt>'2001-07-08'	3.8s	401.6MB	2.2s	2.8MB	快42%	少99.3%	

从上表中可以看出,分区可以有效地降低查询延时和扫描量,但过度分区可能适得其反。如下表所示。

本海运句	未分区	未分区		分区		扫描量对比	
重闽后り	耗时 扫描量		耗时	扫描量	▲七中J /J LL	扫描重刈比	
SELECT count(*) FROM tb	4s	24MB	15s	34.5MB	慢73%	爹30%	

建议您在 SQL 语句中通过 WHERE 关键字来过滤分区。

### 缓存

在如今分布式计算和存算分离的趋势下,通过网络访问元数据以及海量数据将会受到网络 IO 的限制。DLC 默认开启 以下缓存技术大幅降低响应延时,无需您介入管理。

Alluxio: 是一种数据编排技术。它提供缓存,将数据从存储层移动到距离数据驱动型应用更近的位置从而能够更容易被访问。Alluxio内存至上的层次化架构使得数据的访问速度能比现有方案快几个数量级。

RaptorX:是Presto的一个连接器。它像 Presto 一样运行在存储之上,提供亚秒级延迟。目标是为 OLAP 和交互式用例提供统一、廉价、快速且可扩展的解决方案。

结果缓存:Result Cache,对于重复的同一查询进行缓存,极大提高速度和效率

DLC Presto 引擎默认支持 RaptorX 和 Alluxio 分级缓存,在短时间内相同任务场景中可以有效地降低延时。Spark、 Presto引擎均支持结果缓存。

下表是在总数据量为1TB的 Parquet 文件中的 TPCH 测试数据,本次测试选用16CU Presto。因为测试的是缓存功能,所以主要从 TPCH 中选择 IO 占用比较大的 SQL,涉及的表主要有 lineitem、orders、customer 等表,涉及的



SQL 为 Q1、Q4、Q6、Q12、Q14、Q15、Q17、Q19 以及 Q20。其中横坐标表示SQL语句,纵坐标表示运行时间 (单位秒)。



需要注意的是,DLC Presto 引擎会根据数据访问频率动态加载缓存,所以引擎启动后首次执行任务无法命中缓存, 这导致首次执行仍受网络 IO 限制,但随着**执行次数增加**,该限制明显得到缓解。如下表展示了 presto 16cu 集群三 次查询的性能比较。

查询语句	查询	耗时	数据扫描量
SELECT * EBOM table namewhere	第一次查询	3.2s	40.66MB
udid='xxx';	第二次查询	2.5s	40.66MB
	第三次查询	1.6s	40.66MB

您可以在DLC控制台数据探索功能中查看执行的SQL任务的缓存命中情况。

← 查询结果 统计数据	5		运行历史 下载历史
数据扫描量 <b>798</b> B	自适应shuffle ① shuffle字节数 溢出到COS	螺存加速 ① ResultCache 螺存命中 否 FragmentCache命中車 Alluxio 缓存命中	物化現型加速 ① 自动改写到 DataLakeCatalog.mv_test4.mv_student 利用数据 4468
■ 预处理: 2.86s		■ 执行:8.51s	■ 获取结果:192.00ms

### 索引

内表+索引的建表方式相对于外表,在时间和扫描量上均会大幅减小,关于创建表的更多详细信息,请参考数据表管理。



创建表后根据业务使用频率在 insert 前建立索引, WRITE ORDERED BY 后的索引字段。

alter table `DataLakeCatalog`.`dbname`.`tablename` WRITE ORDERED BY udid;

下表展示了 presto 16cu 集群在外表和内表(加索引)上查询性能比较

表类型	查询	耗时	数据扫描量
	第一次查询	16.5s	2.42GB
外表	第二次查询	15.3s	2.42GB
	第三次查询	14.3s	2.42GB
	第一次查询	3.2s	40.66MB
内表(索引)	第二次查询	2.5s	40.66MB
	第三次查询	1.6s	40.66MB

从表中可以看出,内表+索引的建表方式相对于外表,在时间和扫描量上均会大幅减小,并且由于缓存加速,执行时 间也会随着执行次数的增加而减少。

### 同步查询和异步查询

DLC 针对于 BI 场景进行了特别的优化,可以通过配置引擎参数dlc.query.execution.mode来开启同步模式或者异步模式(只支持 presto 引擎)。取值介绍如下。

async(默认):该模式任务会完成全量查询计算,并将结果保存到 COS,再返回给用户,允许用户在查询完成后 下载查询结果。

sync:该模式下,查询不一定会执行全量计算,部分结果可用后,会直接由引擎返回给用户,不再保存到 COS。因此用户可获得更低查询延迟和耗时,但结果只在系统中保存30s。推荐不需要从 COS 下载完整查询结果,但期望更低查询延迟和耗时时使用该模式,例如查询探索阶段、BI 结果展示。

配置方式:选择数据引擎后,支持对数据引擎进行参数配置,选择数据引擎后,在高级设置单击添加即可进行配置。



	📄 请选择默认数据库 🍷 🔚 🖡	oublic-engine 💮 presto 🔻
🗄 数据引擎		
public-engine		▼
⑦ 引擎 先选择数据引擎	<sup>全</sup> 才可选择引擎	
presto		T
① 创建引擎		
高级设置 🔺		配置说明 🖸

### 资源瓶颈

评估资源是否达到瓶颈,DLC 提供引擎的 CPU、内存、云盘、网络等资源监控。您可以根据业务规模调整资源规格,变配请参考调整配置费用说明。查看引擎资源使用情况步骤如下:

1. 打开左侧数据引擎标签页。

2. 单击相应引擎的右侧监控按钮。

**3**. 跳转到腾讯云可观测平台,可以查看到所有监控指标,如下图所示。详细操作以及监控指标请参考数据引擎监控。同时您也可以针对每个指标进行告警配置,详细介绍请参考监控告警配置。

() ::	数据引擎 数据引擎	<b>⑤ 北京</b> 网络配置	Ŧ									数据引撃を	使用指南 12 212 ・
© ≣	<ol> <li>数据引持 将不会/</li> </ol>	擎包括独享数据 产生任何费用,	引擎与共享数据引擎 操作步骤及注意事项	。public_engine(共享) 可参见管理独享数据引导	)筆) 按扫描量计费。 1 C	由系统进行管理,无需操作及授权	: 独享引擎支持按量计费及	2.包年包月两种计费模式,	更多计费信息可参见 计最概述 🖸	。您可对按量计费数据引	學配置自动挂起或定时挂起策略,	推起后	×
±	创建资源	订单查询 🗹	续费管理 🖸							选择资源标签或输入关键	字(多个关键字请用)分割)进行弱;		Q Ø
	资源名称/ID		0	使英型	是否自动模费	启停策略	集群规模	网络配置	创建时间 \$	创建人	描述	操作	
6		<b>.</b> ,		2181 2181	Ŧ	于动启动,于动挂起	16 CU 标准型		2023-02-07 20:41:18		test	「监控」 現象 「更多」	化工
	Datasargene 100			34MI	書	手动启动,手动挂起	16 CU 标准型		2023-01-30 10:56:06		test	国拉 规格 更多,	17. M -





### 其他因素

### 自适应 shuffle

为了提高稳定性,DLC 默认开启自适应 shuffle,这是一套即能支持有限本地磁盘的常规 shuffle,又能保证在大 shuffle和数据倾斜等场景下的稳定性。自适应 shuffle 带来的优势:

降低存储成本:集群节点的磁盘挂载量进一步降低,一般规模集群每节点只要50G、大规模集群也不超200G。
 稳定性:对于 shuffle 数据量剧增或数据倾斜场景任务执行的稳定性不会再因本地磁盘限制而失败。
 尽管自适应shuffle带来存储成本的降低和稳定性提升,但在某些场景下,如资源不足时,会带来约15%的延时。

### 集群冷启动

DLC 支持自动或者手动挂起集群,挂起后不再产生费用,所以在集群启动后,首次执行任务可能存在"正在排队"的 提示,这是因为集群冷启动中正在拉起资源。如果您频繁提交任务,建议购买包年包月集群,该类型集群不存在冷 启动,能在任何时间快速执行任务。



# UDF 函数开发指南

最近更新时间:2024-07-31 18:03:25

### UDF 说明

用户可通过编写 UDF 函数,打包为 JAR 文件后,在数据湖计算定义为函数在查询分析中使用。目前数据湖计算 DLC 的 UDF 为 HIVE 格式,继承 org.apache.hadoop.hive.ql.exec.UDF,实现 evaluate 方法。 示例:简单数组 UDF 函数。

```
public class MyDiff extends UDF {
    public ArrayList<Integer> evaluate(ArrayList<Integer> input) {
        ArrayList<Integer> result = new ArrayList<Integer>();
        result.add(0, 0);
        for (int i = 1; i < input.size(); i++) {
            result.add(i, input.get(i) - input.get(i - 1));
            }
            return result;
        }
    }
}</pre>
```

```
pom 文件参考:
```

```
<dependencies>
<dependency>
<groupId>org.slf4j</groupId>
<artifactId>slf4j-log4j12</artifactId>
<version>1.7.16</version>
<scope>test</scope>
</dependency>
<dependency>
<groupId>org.apache.hive</groupId>
<artifactId>hive-exec</artifactId>
<version>1.2.1</version>
</dependency>
</dependency>
```

### 创建函数

### 注意:

如您创建的是 udaf/udtf 函数,需要在函数名相应加上 \_udaf/\_udtf 后缀。



若您了解 SQL 语法,可通过数据探索执行 CREATE FUNCTION 语法完成函数创建,或通过可视化界面创建,流程如下:

1. 登录数据湖计算控制台,选择服务地域。

2. 通过左侧导航菜单进入**数据管理**,选择需要创建的函数的数据库,如果需要创建新的数据库,可参见数据目录及数据库管理。

数据管理	◎ 广州 ▼					使用指引 🗈	数据管理使用指
数据目录	<b>数据库</b> 在储桶列表						任罰
<ul><li>○ 管理数据目</li><li>○</li></ul>	目录下的数据库,可以点击数据库名称	,管理数据库下的表、视图等数据对象,	手动导入数据到数据表。 <b>了解更多                                    </b>	。数据操作需要相应的数据权限,	更多权限操作指导	引,请参考 <b>用户权限管</b> :	理指南 ×
创建数据库	DataLakeCatalog v				请输	入名称搜索	Q
数据库名称 💠	创建时间	✿ 描述信息		创建人	操作		
_test	2023-03-2	1 10:05:48			编辑 删除		
ıfig 🗗	2023-02-2	0 11:11:15			编辑 删除		
<b>宛数</b> 讲入顶	家教管理页面。						

4. 单击**创建函数**进行创建。

හි	腾讯云 🕜 总览 📔 云产品 🗸				搜索	产品、文档…	Q	⑦ 小程序	☑ 集团	1账号 ~   备:	案 工具 ~	客服支持 🗸 🛛 费
1	← 数据库 /				创建函数							
	数据表视图 函数				函数名称 *	raw_dff	]					
© 	数据库下的Spark自定义函数。支	寺将JAR包上传到DLC,或者挂	圭在用户的COS存储上,建议使	用上传到DLC方式,	描述信息	选填	_					
iii E	创建函数 更新时间 全部	近7天 近30天		Ē								
Ē	函数名称	创建时间 🕈	更新时间	描述信息	存储方式	<ul> <li>上传到系统</li> </ul>	<b>依保存</b> 〇 :	指定COS位置挂	载			
Ø	in in			-		函数包的存储》	方式,可以选	择将函数包上传	到系统并保存	(推荐) , 或者直	直接指定函数包	保存的对象存储COS路
5	<b>— •</b>				程序包来源	○ 本地上传	🔾 数据存	储COS				
	共 2 条				文件路径*	cosn:/ 程序包只支持、	<b>130542</b> JAR类型,文	4723/demo-1.0- 件最大支持100M	SNAPSHOT.jar		选择COS	立置
					函数类名 *	com.tencent	t.dlc.MyDiff					

UDF 的程序包支持本地上传或选择 COS 路径(需具备 COS 相关权限),示例为选择 COS 路径创建。 函数类名包含"包信息"及"函数的执行类名"。

### 函数使用

1. 登录数据湖计算控制台,选择服务地域。



2. 通过左侧导航菜单进入数据探索,选择计算引擎后即可使用 SQL 调用函数。

کة 🕑	日 保存 □ 保存 □ 品 格式化 □
35 se	elect raw_diff(Array[1,2,3])
36 37	
查询结果	统计数据
Task ID	ー SQL详情 - 导出结果 优化建议 IZ
查询耗时	5.21s
共1条数	据(控制台最多可展示1000条数据)复制数据 🕞
test_ca	che.raw_diff(array(1, 2, 3))
[0,1,1]	



### 物化视图

最近更新时间:2024-07-31 18:03:40

#### 注意:

目前数据湖计算 DLC 物化视图只支持 SparkSQL 引擎和 Presto 引擎。

物化视图(Materialized View)是数据库中的一种特殊对象,它是一个预先计算和存储的查询结果集。物化视图在处理大量数据和复杂查询时可以提供快速的查询性能。

物化视图提高查询性能的同时也引入了存储成本和计算成本。我们建议您在以下场景使用物化视图:

源表变更不频繁

相比于源表,物化视图表的字段和结果数量有明显的减少

DLC 支持普通物化视图和映射物化视图,以下是介绍和完整的使用示例,支持的语法列表可以参考物化视图语法。

### 普通物化视图

普通物化视图的基本使用流程包括创建、刷新、使用。 以下基于 Presto 引擎操作举例完整流程。

#### 准备数据

执行 SQL创建库表,并插入数据。以下语句创建了一个名为 student 的表。

```
CREATE DATABASE IF NOT EXISTS mv_test3;
create table student(id int, name string, score int);
insert into student values (1,'zhangsan', 90);
insert into student values (2,'lisi', 100);
insert into student values (3,'wangwu', 80);
insert into student values (4,'zhaoliu', 30);
select * from student order by id;
```

### 创建普通物化视图

使用 CREATE MATERIALIZED VIEW 语句来创建物化视图。指定物化视图的名称和查询语句,可以选择性地指定 查询的来源表和条件。

以下例子,使用了一个简单的 SELECT 语句从表 student 中选择所有分数,并对它们进行求和操作。然后将这个 求和结果作为物化视图 mv\_student\_sum 的内容。

```
CREATE MATERIALIZED VIEW mv_student_sum AS (
   select sum(score) from student
);
```



### 查看物化视图详情

使用 DESCRIBE MATERIALIZED VIEW 语句来查看物化视图的详细信息,包括名称、查询语句和刷新状态等。

DESCRIBE MATERIALIZED VIEW mv\_student\_sum;

查询结果	运行历史 下载历史
expr_0 bigint	
MaterializedView Detail:	
state NORMAL	
mvType SINGLE	
viewOriginalText SELECT SUM(`score`)	
FROM `student`	
autoRewrite ENABLE	
dataLatest FRESH	
freshType RUNTIME	
updateType FULL	
createTime Sat May 06 19:57:52 CST 2023	
modifiedTime Sat May 06 19:58:00 CST 2023	
그리수도 빠른 입고 거의 보험	

手动刷新物化视图

使用 REFRESH MATERIALIZED VIEW 语句来手动刷新物化视图的数据。

此处仅作演示,大部分情况下,您并不需要手动刷新物化视图,只要 SQL 命中了源表有变更的物化视图就会自动刷新。

REFRESH MATERIALIZED VIEW mv\_student\_sum;

#### 查看物化视图的执行任务列表

使用 SHOW MATERIALIZED VIEW JOBS 语句来查看物化视图的执行任务列表,可以了解到物化视图的刷新历史和状态。

SHOW MATERIALIZED VIEW JOBS IN mv\_student\_sum;

查询结果						运行历史 下载历史
Task ID SQL详情         导出结果           查询耗时         490 ms         数据扫描的           共 2 条数据(控制台最多可展示)         100 mg	优化建议 🖸 📱 0 B 示1000条数据)复制数据 💼					
xid	taskld	state	buildType	execEngine	createTime	updateTime
19b61c931fa646159ea	95284167ec0511ed9ae	FINISHED	MANUAL_REFRESH	PRESTO	2023-05-06 20:00:27	2023-05-06 20:00:3
6bb622691e68471b909	328b5149ec0511ed9ae	FINISHED	CREATE	PRESTO	2023-05-06 19:57:47	2023-05-06 19:58:0

SQL 改写执行

使用 SELECT 语句查询数据,期望自动改写并命中物化视图。可以通过查询结果里的统计数据,查看是否自动改写 到了物化视图上。

```
select sum(score) from student;
```



← 查询结果 统计数据			运行历史 下载历史	
数据扫描量	自适应shuffle(i)	缓存加速 🛈	物化视图加速 访	
<b>798</b> <sup>₿</sup>	shufile字节数 溢出到COS	ResultCache 缓存命中 否 FragmentCache命中率 Alluxio 缓存命中	自动改写到 DataLakeCatalog.mv_test4.mv_student 利用数据 446B	
■ 预处理:2.86s		■ 执行: 8.51s	■ 获取结果:192.00ms	

### 删除物化视图

DROP MATERIALIZED VIEW mv\_student\_sum;

### 映射物化视图

映射物化视图是一种特殊类型的物化视图,它与现有的表进行映射关联。通过映射物化视图,可以将物化视图的查询结果与现有表的数据进行关联,从而实现对现有表的查询性能优化。

### 限制

物化视图相对于普通物化视图有以下限制:

映射物化视图不支持刷新操作,即无法通过REFRESH MATERIALIZED VIEW语句来刷新物化视图的数据。因此,物 化视图的数据只能与映射表的数据保持一致,无法自动更新。

映射物化视图不进行自动SQL改写,即查询语句不会自动转换为使用物化视图。需要手动指定使用物化视图的查询语句。

删除映射物化视图时,只会删除与映射表的关联关系,而不会删除映射表本身。映射表仍然存在,可以继续使用。

### 推荐场景

推荐您在以下场景使用映射物化视图:

当已经存在一个数据量较大的表,并且该表的查询性能较低时,可以通过映射物化视图来优化查询性能。 当需要保持物化视图的数据与现有表的数据保持一致,并且不需要自动刷新物化视图时,可以使用映射物化视图。

### Iceberg 类型的源表

lceberg 表为源表时,完整示例如下:

### 基于 CTAS 创建映射物化视图

映射物化视图需要与待映射的表保持名称一致。以下例子先基于CTAS创建表,用于映射MV的创建。数据的准备可以参考普通物化视图中完整示例中的数据准备一节。



```
CREATE TABLE link_mv_student AS (
   select sum(score) from student
);
--创建映射物化视图:使用CREATE MATERIALIZED VIEW语句创建映射物化视图。
--在创建物化视图时,使用WITH META LINK子句,并指定映射表的名称作为关联。
CREATE MATERIALIZED VIEW link_mv_student WITH META LINK AS (
   select sum(score) from student
);
```

#### 查看映射物化视图

使用 DESCRIBE MATERIALIZED VIEW 语句可以查看映射物化视图的详细信息,包括名称、查询语句和刷新状态等。

DESCRIBE MATERIALIZED VIEW link\_mv\_student; SHOW MATERIALIZED VIEW JOBS IN link\_mv\_student;

#### 映射物化视图不支持刷新操作

映射物化视图不支持 REFRESH 操作,即无法通过 REFRESH MATERIALIZED VIEW 语句来刷新物化视图的数据。因此,物化视图的数据只能与映射表的数据保持一致,无法自动更新。

#### SQL 改写

映射物化视图不会自动对查询语句进行 SQL 改写。

如执行 select sum(score) from student; 不会命中映射物化视图。 可以通过使用 Hint 或 TaskConf 参数来指定允许基于映射物化视图进行 SQL 改写。

--手动指定需要改写SQL
select /\*+ OPTIONS('eos.sql.materializedView.enableRewrite'='true') \*/
sum(score) from student;

#### 删除映射物化视图

使用 DROP MATERIALIZED VIEW 语句来删除映射物化视图。删除映射物化视图后, 仅会删除与映射表的关联关系, 映射表本身仍然存在。

```
DROP MATERIALIZED VIEW link_mv_student;
DESCRIBE link_mv_student; --可查看源表还存在
```

#### Hive 类型的源表

Hive 表为源表时,完整示例如下:

#### 准备初始化数据



首先,需要准备初始化数据并创建Hive基表。使用 CREATE EXTERNAL TABLE 语句创建 Hive 基表,并通过 INSERT 语句手动插入数据。

```
CREATE EXTERNAL TABLE student_2(id int, name string, score int)
LOCATION 'cosn://guangzhou-test-1305424723/mv_test4/student_2';
insert into student_2 values (1,'zhangsan', 90);
insert into student_2 values (2,'lisi', 100);
insert into student_2 values (3,'wangwu', 80);
insert into student_2 values (4,'zhaoliu', 30);
select * from student_2;
```

创建被映射的 Hive 外表

使用 CREATE EXTERNAL TABLE 语句创建一个被映射的 Hive 外表。

```
CREATE EXTERNAL TABLE link_mv_student_hive (
   sum_score BIGINT
) LOCATION 'cosn://guangzhou-test-1305424723/mv_test4/link_mv_student_hive';
```

向映射表插入数据,使用 INSERT OVERWRITE 语句将查询结果插入到映射表中,确保映射表的数据与 Hive 基表的数据保持一致。

```
--向映射表插入数据
INSERT OVERWRITE link_mv_student_hive
select sum(score) from student;
```

#### 基于 Hive 外表创建映射物化视图

使用 CREATE MATERIALIZED VIEW 语句创建映射物化视图。在创建物化视图时,使用 WITH META LINK 子句,并指定上述 Hive 外表的名称作为关联。

```
CREATE MATERIALIZED VIEW link_mv_student_hive WITH META LINK AS (
select sum(score) from student_2
```

);

# 系统约束 元数据信息

最近更新时间:2024-07-17 18:17:37

#### 数据库、数据表、属性列、分区的个数。

项目	最大数量
每个账户的数据库数量	1,000
每个账户的数据表数量	10,000
每个数据库的表数	4,096
每个数据表的列数	4,096
每个表的分区数	10,000
每个主账户的分区数	1,000,000
每张表字段数量上限	4096
每个账户的自定义函数数量	100
可创建 catalog 数量	20

### 数据库

名称:127个字符,同一个数据链接下,不允许有相同的数据库名称。 描述:2048 个字符。 外表的数据地址(COS 地址):888个字符(COS 路径长度限制)。 参数: Map<string:string> 的形式,每个参数的长度限制为127个字符,总长度限制为3000个字符。

### 数据表/视图

名称:127个字符,同一个数据库下,不允许有相同的数据表名称。 描述:1000个字符。 外表的数据地址(COS地址):888个字符(COS路径长度限制)。 参数: Map<string:string> 的形式,每个参数的长度限制为127个字符,总长度限制为512000个字符。

### 属性列

名称:**127**个字符,同一个数据表下,不允许有相同的属性列名称。 描述:**256**个字符。



字段对应的值:131072个字符。若超过将无法创建。

### 分区

分区字段名称:127个字符。



# 计算任务

最近更新时间:2025-01-03 15:27:27

单条 SQL 语句大小限制为2MB。