

腾讯云数据工坊

快速入门

产品文档



腾讯云

【版权声明】

©2013-2017 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。

【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模式的承诺或保证。

文档目录

文档声明.....	2
快速入门.....	4
案例背景	4
准备工作	5
详细配置步骤.....	8

快速入门

案例背景

Web 日志分析

某移动应用开发商在腾讯云 CDB MySQL 已经存放了大量 Web 用户日志数据。用户希望通过周期性的 Hive 工具对数据进行分析，并将分析结果，存放 CDB 数据格式已经预先经过结构化处理，按照不同字段存放。

希望通过 Hive 工具，计算日志中不同出错类型HTTP返回值（http_code）的数量。

准备工作

配置所用信息

配置所有信息如下图所示：

实体	名称	备注
CDB MySQL实例	game_analysis	数据库实例
CDB MySQL数据库	game	数据库名称
CDB MySQL日志原始表	game_log	原始日志
CDB MySQL分析结果表	game_result	内容通过导入任务，由TDF分析结果表导入
TDF工程名称	game_log	工程中包括表、工作流和任务
TDF数据源服务器标识	*****_game_log_raw	TDF中定义的逻辑实体，关联至CDB MySQL实例game_analysis；前缀*号系统自动生成
TDF数据源服务器说明	game_log原数据	同*****_game_log_raw——一对应的说明
TDF日志原始表	game_tdf_raw	为game_log原始表的子集，仅选取部分字段
TDF分析结果表	game_tdf_dest	存放分析结果，包含http_code，count字段

说明：上图中为该示例所用到的表以及其他逻辑实体。

注册腾讯云账号并开启 TDF 服务

首先，请登录 <https://cloud.tencent.com/>，完成注册。腾讯云使用 QQ 账号体系或邮箱注册，如为企业开发商，建议使用专用非私人账户注册。如用户已经使用腾讯云产品。

在产品介绍页面 <https://cloud.tencent.com/product/tdf> 提交 TDF 开通申请，开通后，可登录腾讯云-控制台，在云产品中选择数据工坊 TDF。

准备日志原始数据

假设日志已经存放在CDB MySQL中，实例名为 game_analysis。

示意图如下所示：



可通过数据库客户端软件，查询当前原始数据的数据库格式，包括字段等，并记录。在本场景中，日志原始数据样本有数千条，单条记录示意如下，存放在腾讯云CDB MySQL数据库实例game_analysis的game_log表中：

```
20160626002522 115.153.150.199 tdf.qcloud.com /tdftest_21_FotAejgyFBMfNWJu-5KNjyf7axH-.jpg?imageView2/1/w/1080/h/1920&e=1466783400&token=Q-hCY0VbL4F6NTX3TgRvE_T3vcpNEo2Gr3S9RA-b:yrQjzCHZH2luqA-ONED8IODBjY= 185801 1465 206 NULL 702 "AndroidDownloadManager/5.1.1 (Linux; U; Android 5.1.1; tdfest R3 A Build/LAT4AV)"
```

可下载日志原始数据样本并通过您熟悉的MySQL客户端软件上载至CDB:TDF_Demo_Sample_Raw_game_log.csv [下载](#)

- 日志字段内容说明：

request-time、user-

Agent、access_time、ip、host、request_uri、size、province、isp、http_code、referer.

其中字段内容已经做了处理，仅作参考。

您可在如下链接下载到上述测试数据至本地，并通过您自己熟悉的数据库客户端软件导入到 game_log 表项中。

另，准备空表用于存放结果，名为 game_result 在本例子中需分析不同http_code对应的记录计数。

结构如下图所示：

Field	Type	Null	Key	Default	Extra
http_code	int(11)	NO		NULL	
count	double	NO		NULL	

详细配置步骤

第一步：创建工程和配置基本信息

登录TDF后，首先进入总览页面。如果是初次进入，则无工程和表信息。

单击创建工程，如下图，输入工程名（例如game_log）。工程标识前缀为系统唯一标识符自动生成。工程管理员默认为用户本人；计算引擎为系统默认设置。

创建成功后，可在总览视图看到所创建的工程基本信息,如下图所示：



第二步：创建数据表

数据表可理解为数据处理过程中在 TDF 所存放的位置。

在本示例中，需要用到两个表：原始数据表（用于存放从 CDB MySQL 导入的数据，例如game_tdf_raw），以及结果表（用于存放分析后的结果，例如game_tdf_dest），过程如下：

创建原始数据表

点击管理控制台，在【数据管理-表管理】点击屏幕右侧【新建表】按钮，开始创建数据表。填写表名 game_tdf_raw，选择刚刚创建的工程game_log。

在高级设置中，选择表的记录格式和对应文件格式，按照默认设置即可。

[返回](#) | 新建表

1 填写表信息



表名 *

game_tdf_raw



表名必须为数字字母开头，是小写字母，数字，下划线组合，且不能全为数字或者下划线(30个字符以内)

所属工程 *

game_log



表描述

请输入描述

[显示高级设置...](#)

下一步

点击下一步，设置表中的字段。本例中不使用分区，在本例子中，我们只打算分析用户接入日志中不同 http_code 类型的数量，因此，仅选择如下8个字段进行分析：

access_time,host,http_code,ip,isp,province,request_time,request_uri

初始编辑界面类似如下图所示：

[返回](#) | 新建表

1 填写表信息



2 字段与分区

字段英文名	字段类型	字段描述	是否将该字段设置为分区字段
access_time	string		☐ 是 删除

[添加一行](#)

上一步

提交

创建后，可在【数据管理-表管理】中，查询表结构，如下：

[< 返回](#) | game_tdf_raw

表信息

表结构

记录数趋势

修改历史

增加字段

字段英文名	字段类型	描述	分区字段	操作
access_time	string		否	修改 删除
host	string		否	修改 删除
http_code	int		否	修改 删除
ip	string		否	修改 删除
isp	double		否	修改 删除
province	double		否	修改 删除
request_time	string		否	修改 删除
request_uri	string		否	修改 删除

创建结果表

点击管理控制台，在【数据管理-表管理】点击屏幕右侧【新建表】按钮，开始创建数据表。

填写表名 game_tdf_dest，选择刚刚创建的工程 game_log。

在高级设置中，选择表的记录格式和对应文件格式，按照默认设置即可，

点击下一步，设置表中的字段。在本例子中，我们只打算记录不
http_code类型的数量结果，因此，仅选择如下两个个字段且不使用分区：http_code，count

创建后，可在【数据管理-表管理】中，查询表结构，如下图所示：

[< 返回](#) | game_tdf_dest

表信息

表结构

记录数趋势

修改历史

增加字段

字段英文名	字段类型	描述
http_code	int	
count	double	

此结构需同最终计算完毕后，拟用于存放结果的CDB MySQL表game_result结构相同。

第三步：设置数据源

在【工程管理-数据源管理】，点击【+新建配置】。

选择服务器类型：云数据库CDB-MySQL，编辑服务器标识（前缀由系统自动生成，自行输入部分例如【game_log_raw】），和服务说明（例如【game_log原数据】），责任人默认为当前用户，下拉选择CDB数据库实例【game_analysis】（之前在CDB配置），下拉选择数据库名称【game】（之前在CDB配置），并输入对应用户名密码，连接后，即完成数据库配置。

服务器类型 *	cdb
服务器标识 *	game_log_raw
服务器说明 *	game_log原数据
责任人 *	选择成员 +
实例 *	game_analysis ▼ 暂只支持广州地域
数据库名称 *	game ▼
数据库用户名 *	root 字母开头，由字母、下划线、数字组成，最长16个字符
数据库密码 *	*****

第四步：创建工作流

在【数据开发-工作流开发】界面，下拉选择刚刚创建的工程【game_log】，然后选择新创建工作流列表。

新建工作流

×

工作流名称 *	game_log任务
责任人 *	选择责任人 +
备注	

确定

取消

第五步：创建数据接入任务

拖拽任务节点【数据接入-CDB MySQL导入Hive】，并单击右键编辑，基本信息配置，任务名称命名为【导入game_log任务】，类型默认为【CDB MySQL导入Hive】，责任人为默认当前用户，输入输出配置。

源数据库：选择之前创建的后缀为【game_log_raw】数据库。

源数据库查询SQL语句：例如如下语句

```
select access_time,host,http_code,ip,isp,province,request_time,request_uri from game_log。
```

即仅检索以上8个字段，顺序和字段名严格按照CDB MySQL的顺序和字段名。

目标表所在工程：选择当前TDF工程game_log

目标表：选择之前建立的TDF表game_tdf_raw

目标表列名映射：

```
access_time,host,http_code,ip,isp,province,request_time,request_uri
```

分区格式选择【无分区】，其余设置按照默认值。

目标表所在工程 *

game_log ▾

新建工程

目标表 *

game_tdf_raw ▾

新建表

目标表列名映射 *

access_time,host,http_code,ip,isp,province

填目标表列名，顺序需与源数据查询结果顺序保持一致，例如：column_1,column_2,column_3

分区格式 *

无分区 ▾

分隔符 *

\t

字段分隔符，只能为单个字符，如：\t \x09 或，

数据库入库模式 *

TRUNCATE ▾

脏数据阈值 *

1000

允许失败的条数

是否忽略空数据源 *

是 ▾

是否空数据源的时候任务成功

保存设置

第六步：创建 HQL 计算任务

该任务为计算日志中不同的 http_code 对应的记录数量。任务脚本参数如下图所示：

< 返回 | 任务编辑



任务基本信息

任务名称*
 中文或字母开头，由中文、字母、下划线、数字组成，最长32个字符

任务类型

责任人*

任务说明

计算设置

数据库名

SQL*

```
1 use ;
2 insert overwrite table game_tdf_dest select http_code,count(http_code) from game_tdf_raw group by http_code;
```

其中，SQL脚本代码如下：

use 数据库名称;

insert overwrite table game_tdf_dest select http_code,count(http_code) from game_tdf_raw group by http_code.

其中数据库名称需填入文本框上方的数据库名。此语句将TDF日志原始表中的记录按照http_code不同类别计数，并将结果存放回TDF分析结果表game_tdf_dest。

第七步：创建数据导出任务

拖拽任务节点【数据导出-Hive导出CDB MySQL】，并单击右键编辑。

基本信息配置，任务名称命名为【game导出任务】，类型默认为【Hive导出CDB MySQL】，责任人为默认当前用户。

输入输出配置中，选择源数据库所在工程【game_log】，查询语句为select * from game_tdf_dest，即将上一个任务节点Hive计算得到的结果表game_tdf_dest内容全部回写到CDB

MySQL，下拉选择目标数据库以及目标表（即CDB已有数据库和目标表），目标表列名映射为：

http_code,count，其余按照默认值即可。即如下配置：

输入输出

源数据库所在工程 *

game_log ▾

[新建工程](#)

源数据库查询SQL *

select * from game_tdf_dest;

支持通配符，只可导出单表，不能含有join，不能含有复杂查询语句，不能以分号结尾，如：
select column_1,column_2,column_3 FROM table where p_date='\${YYYYMMDD}'

目标数据库 *

game_log_raw ▾

[新建数据库配置](#)

目标表 *

game_result

[导入MySQL表名](#)

目标表列名映射 *

http_code,count

填目标表列名，顺序需与源数据查询结果顺序保持一致，例如：
column_1,column_2,column_3

数据库入库模式 *

TRUNCATE ▾

脏数据阈值 *

1000

允许失败的条数

是否忽略空数据源 *

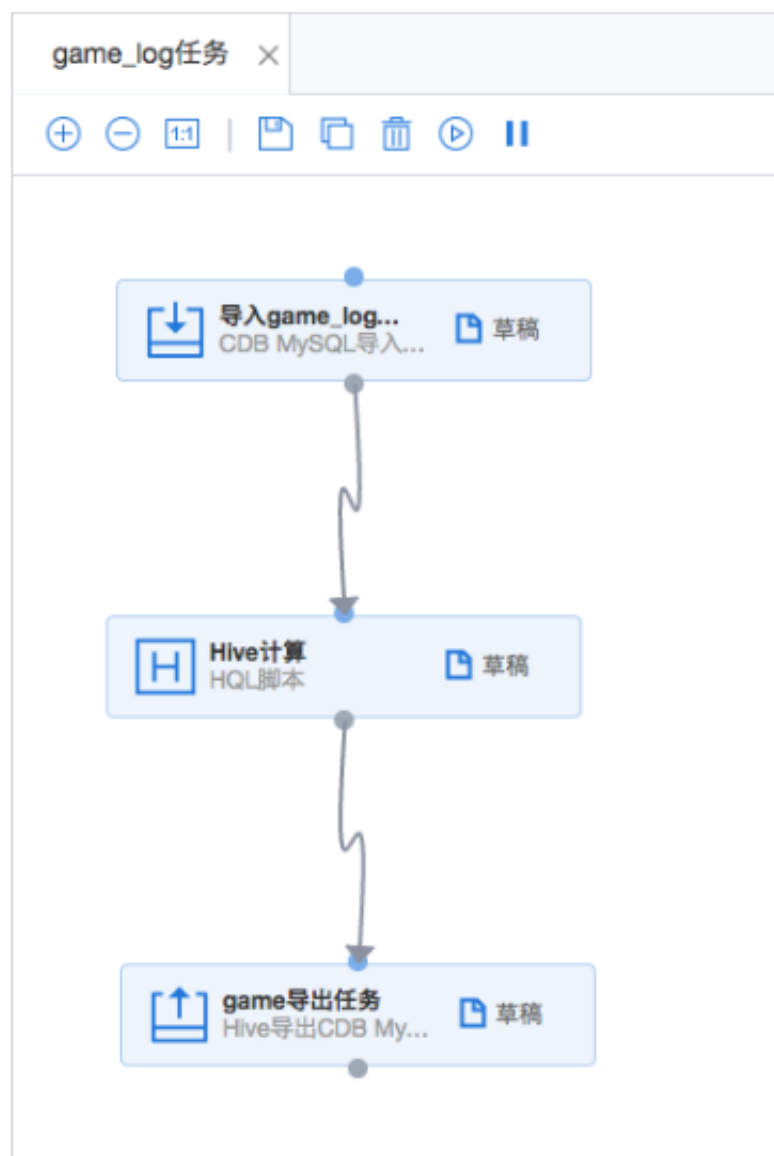
是 ▾

是否空数据源的时候任务成功

保存设置

第八步：保存工作流并发布

任务节点编辑完毕后，检查节点间的依赖关系，如下图连线可见。



点击保存，和发布任务按钮，系统将提示确认调度信息，如下，选择单次执行。

编辑 workflow

×

workflowID

4c7bdb1d-208

workflow名称 *

game_log任务

责任人 *

选择责任人 +

备注

game_log任务

调度设置

☒ 单次

☐ 周期执行

单次任务将在 workflow 发布后立即执行

确定

取消

发布后，在运维中心的任务列表视图可以按照工程维度查看当前的 workflow 的实时状态。当 workflow 状态切换为成功后，任务即全部运行成功（根据数据量和系统负载不同，任务运行时间将会略有区别）。

任务列表 game_log

任务视图

workflow视图

终止 workflow

重新 workflow

筛选

2016-12-19 至 2016-12-26

workflow名称

搜索

	workflowID	workflow名称	数据时间	任务数量	状态	责任人
搜索"game"，找到1条结果。返回数据列表						
☐	4c7bdb1d-208	game_log任务	2016-12-25 00:00:00	3	成功	

第九步：分析结果展示

按照前序设定，查询结果将首先写入TDF表，随后再写入CDB对应数据库。当 workflow 状态成功后，可使用数据库管理客户端，对CDB表进行查询。例如，执行完如下语句 `select * from game_result` 后，系统显示结果如下：

http_code	count
200	2166
206	160
304	1
405	4

即在所有日志中，http_code不同类型类型对应的计数结果如上。

也可以选择通过可视化报表软件，将最终结果进行图形化展示。