大数据处理套件

操作指南

产品文档

腾讯云

文档目录

操作指南
实时计算
实时计算概述

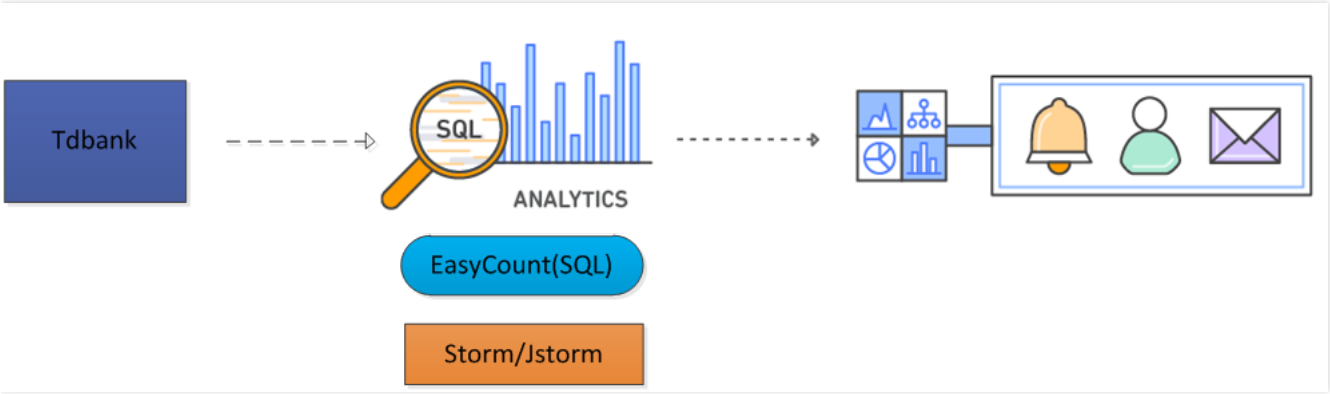
最近更新时间：2017-12-07 16:47:22

实时计算一般针对海量数据进行实时的计算和分析，具备秒级甚至毫秒级响应的特性。实际应用中数据源往往是实时的、不间断的，用户的响应也是实时的。实时计算可以帮您实时分析并动态刷新用户访问数据，展示网站实时流量的变化情况和用户分布情况等。

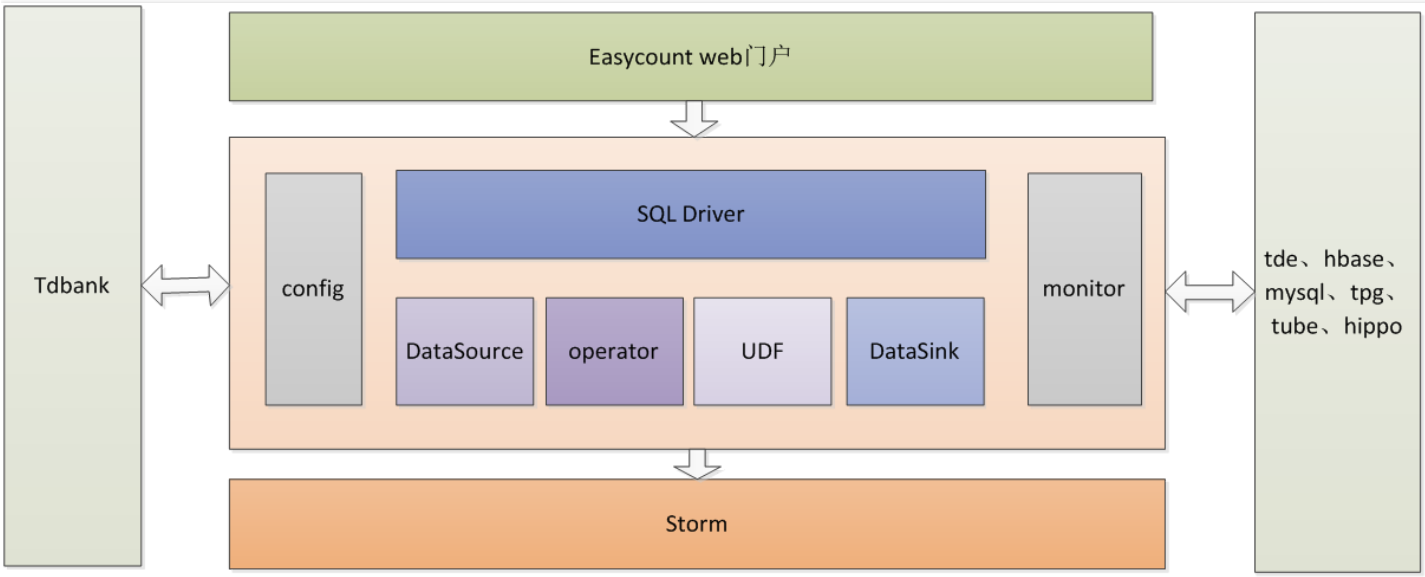
腾讯大数据处理套件采用的实时计算平台是 EasyCount。 EasyCount 使用 SQL 描述业务实时计算的需求，并将 SQL 转化为基于 storm 的 topology。相对于传统的 SQL，EasyCount 加入了窗口的概念，使得数据可以一直保存在内存中，快速地进行大量内存计算。EasyCount 的输出结果为数据流在某一时刻的计算结果。

EasyCount 的设计目标就是，用纯粹的 SQL 语句再加上一些命令，就可以完成所有的任务发布以及执行。这样，就可以通过 EasyCount Web 门户配置和管理实时计算任务。对于有一定 SQL 基础的用户，只需要掌握一些 EasyCount SQL 比较特殊的语法，比如窗口或者临时表定义的语法，就可以配置出可运行的实时计算任务，大大降低了实时计算上手难度。

EasyCount 从 kafka 或 hippo 读取数据同时关联第三方维表数据（MySQL、HBASE等）进行实时分析。计算结果数据有多重分发途径，可以回流到 TDBank，KV，DB（TPG，MYSQL），ES，以及 HBASE 等。以下为 EasyCount 运行示意图：



EasyCount 实时计算平台主要分为两大块组成：第一块是 EasyCount web 门户，管理 EasyCount 脚本和 EasyCount 任务；第二块是 EasyCount core，SQL Driver 负责将 SQL 解析成 AST、DataSource 负责计算数据源接入的实现、DataSink 负责将计算结果会写到配置的结果表、Operator 负责计算算子的实现、UDF 负责系统自定义函数的实现、config 负责系统的配置管理、monitor 负责监控指标的统计及上报。



EasyCount 系统关键概念定义

最近更新时间：2017-12-07 16:47:42

表

EasyCount 系统处理的基本单元是“表”，也就是说需要有明确的 schema 定义的结构化数据。不过相对于传统数据库的表中的字段定义，EasyCount 中的表支持 map，list，struct 以及 binary 等复杂数据类型。根据数据表的功能，EasyCount 系统中的表分为四种，流水表、维表、结果表、临时表。

- 流水表是一组（无穷）元素的集合。流水表上的每个元素都属于同一个 schema。每个元素都和逻辑时间有关，即流包含了元组和时间的双重属性。流水表与传统数据表不同，传统数据表的计算是针对整张表数据进行的，而流水表数据的计算只针对当前时间的数据进行的，最多针对截止到当前时间的所有数据进行计算。流水表数据的统计是实时进行的，根据我们大多数应用需求场景，流水表一般都是按照某个小的时间粒度进行数据统计的，例如 10 s，1 min，5 min 等。因此，EasyCount 系统在进行实时计算的时候，要求数据中必须有一个时间字段作为协调。如果数据中确实没有时间字段，那么 EasyCount 系统就按照接收到数据的时间进行协调。流水表是 EasyCount 实时计算系统的重要数据源，目前系统支持的流水表有 kafka 和 hippo。
- 维表是指静态数据表。用户在实际应用过程中往往需要新增或更改配置需求。所以流水表中的信息是不完全的，在实际的统计中需要关联一些配置信息，之后再进行数据统计。EasyCount 系统将这类数据表抽象为维表，维表数据一般存储在 KV 系统中。如果维表数据量较小，也可以存储到 DB，甚至以配置文件的方式抽象为内存表，直接从内存中关联。维表目前支持以下几种：hbase、MySQL。
- 结果表是用来存储计算结果的表。结果表可以是类似 kafka、hippo 的流水表， 也可以是类似 hbase 的 KV 表，还可以是类似 MySQL 的 db。具体的计算结果流向由业务根据计算结果的数据量及使用情况做权衡。
- 临时表是供计算逻辑临时使用的表。当多组计算逻辑使用同一结构的数据集时可以将该数据集使用 WITH 关键字定义为临时表。

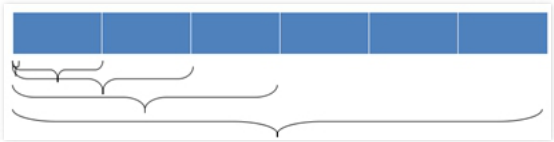
窗口

窗口（window）是流处理中解决事件的无边界性和流动性的一种重要手段，它把事件流在某一时刻变成静态的视图，以便进行类似数据库表的各种查询操作。EasyCount 系统中的窗口需要和聚合函数配合使用。传统 SQL 对于聚合函数的定义都是针对整张数据表进行的，然而对于实时计算，数据是流式进入的，聚合统计只能做到对当前某个时间段的数据进行。因此这里定义了聚合窗口（AGGR INTERVAL）的概念，是指进行数据聚合所采用的时间粒度。假设聚合窗口为 60 s，那就表示每一分钟进行一次聚合计算，聚合计算的结果是针对这 1 分钟数据进行的。这里提到时间窗口，那么这个时间是按照什么标准进行协调的呢，默认情况下，如果不使用 coordinate by 子句，系统按照接收到数据时的系统时间进行协调，如果用户指定了 coordinate by 子句，就按照用户指定的字段进行协调。注意，用户指定的字段必须是一个毫秒数，如果不是可以通过函数转化为毫秒数。

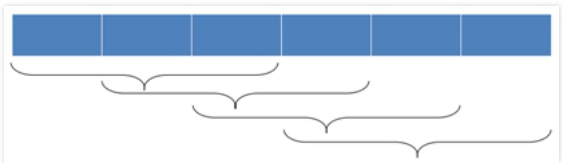
EasyCount SQL 窗口有三种类型，聚合窗口、滑动窗口和累加窗口。累加窗口和滑动窗口，这两个窗口和聚合窗口一样是两个聚合时间粒度。不过这里要求累加窗口和滑动窗口必须是聚合窗口的整数倍。

1）普通聚合：和传统聚合函数一致，对每个聚合窗口进行一次聚合计算。

2）累加聚合：在累加窗口内的每个聚合窗口进行一次聚合计算，不过计算的数据是针对从累加窗口起始直到当前聚合窗口的聚合值。如下图所示：



3）滑动窗口聚合：在每个聚合窗口结束的时候计算，从当前聚合窗口向前推到滑动窗口大小内的数据进行聚合计算。如下图所示：



累加聚合和滑动窗口聚合的具体实现是比较耗费资源的，因此一般累加窗口和滑动窗口不要比聚合窗口大太多。但是在某些场景下，统计任务又无法实现，因此提供了一个新的全局累加功能（ACCUGLOBAL）。这个功能是一个累加聚合功能，区别是启动这个功能以后，所有的聚合函数都是被设置为累加功能，并且同一条子查询中的滑动窗口聚合的功能被关闭。

表达式

表达式是符号和运算符的一种组合。EasyCount 解析引擎处理表达式以获取单个值。简单表达式可以是常量、变量或者函数，可以用运算符将两个或者多个简单表达式联合起来构成更复杂的表达式。

EasyCount S-QL开发指南

最近更新时间：2018-06-12 10:14:09

S-QL（Stream Query Language），流式查询语言。这里面的 S 和标准 SQL 中的 S 含义不同。为了便于区分，使用 S-QL 表示 EasyCount 系统中的流式查询语言。

语法规范

下表给出 S-QL 的语法规范：

语法约定	语法说明	示例	备注
回车	代表一条 sql 逻辑结束。		
WITH (SELECT ...),(SELECT ...)	定义临时表，多个临时表直接使用逗号分隔。	`WITH (SELECT ... FROM...)tmpa, (SELECT ..... FROM .... ) tmpb`	
INSERT INTO	将计算结果写入到结果表。	`INSERT INTO xxx SELECT .... FROM tmpa, INSERT INTO yyy SELECT ..... FROM tmpb'	
COORDINATE BY column_x	指定按照某个时间字段划分统计窗口，默认值为系统时间。	`COORDINATE BY unix_timestamp(dtevetTime, 'yyyy-MM-dd HH：mm：ss')*1000`	按照 dtevetTime 这个时间字段划分统计窗口。
AGGR	定义普通聚合窗口。	`WITH AGGR INERVAL 60 SECONDS`（聚合窗口每隔 60 秒做一次统计输出）	累加窗口和和滑动窗口必须以普通聚合窗口为基础，时间跨度为普通聚合窗口的整数倍。
ACCU	定义累加窗口。	`WITH ACCU INERVAL 120 SECONDS`（累加窗口每隔 120 秒做一次统计输出）	
SW	定义滑动窗口。	`WITH SW INERVAL 120 SECONDS`（滑动窗口 120 秒做一次统计输出）	

以下为一条带有滑动、累加和聚合窗口的 sql 示例：

```
NSERT INTO test_result
SELECT activityId, count(1) pv, countd_hllp(uiUin ACCU) huv,countd_hllp(uiUin SW) fmuv, from_unixtime(AGGRTIME DIV 1000, "yyyy-MM-dd HH：mm：ss")
FROM test_src_kafka
GROUP BY activityId
COORDINATE BY unix_timestamp(dtevetTime, 'yyyy-MM-dd HH：mm：ss')*1000 WITH AGGR INTERVAL 60 SECONDS
WITH ACCU INTERVAL 3600 SECONDS
WITH SW INTERVAL 300 SECONDS
```

通常编写一条 S-QL，可以按照如下步骤进行：

1. 将所有的中间查询结果，通过 WITH 语法，以子查询的方式作为临时表写在最前面，并用英文逗号分隔。
2. 使用 INSERT 语法将计算结果写入目标表，多条 INSERT 语句使用英文逗号分隔。
   可见，一条 S-QL 语句支持多个输入，多个输出，用户可以在一条 S-QL 内部充分发挥。S-QL 支持这种操作，主要原因是 EasyCount 系统中提交一条 S-QL 以后，任务将一直运行，其所占用的系统资源一直不会被释放。在系统资源总数固定的条件下，系统能够承载的总任务数目是有限的，因此提交一个 S-QL 需要相当的谨慎，一条 S-QL 应该可以做更多的事情，从总体上减少系统资源的占用。
   以下是多条 sql 示例：

```
with (select iActivityId, hllp(uiUin) uvb, from_unixtime(AGGRTIME DIV 1000, "yyyy-MM-dd HH:mm:ss") ts,
concat_ws('-', 'd', from_unixtime((AGGRTIME DIV 1000), 'yyyy-MM-dd 00:00:00'), cast(iActivityId as string)) dk
from src
GROUP BY iActivityId
COORDINATE BY unix_timestamp(dteventTime, 'yyyy-MM-dd HH:mm:ss')*1000 WITH AGGR INTERVAL 60 SECONDS) tmp, (select iActivityId, hllp_merge(tmp.uvb, dim.uvball) uvball, tmp.dk k, ts
from tmp
left join dim on tmp.dk=dim.k) jd
insert into dim with k as KEY select jd.k k, jd.uvball uvball from jd,
insert into dest select jd.iActivityId, hllp_get(jd.uvball) duv, jd.ts from jd
```

数据类型

基本数据类型

数据类型	描述	示例
TINYINT	1 字节（8 位）有符号整数（从 -128 到 127），后缀 Y 用来表示小范围的数字。	10Y
SMALLINT	2 字节（16 位）有符号整数（从 -32,768 到 32,767），后缀 S 用来表示一个 egular descriptive number。	10S
INT	4 字节（32 位）有符号整数（从 -2,147,483,648 到 2,147,483,647）。	10
BIGINT	8 字节（64 位）有符号整数（从 -9,223,372,036,854,775,808 到 9,223,372,036,854,775,807），后缀为 L。	100L
FLOAT	4 字节（32 位）单精度浮点数，范围在 1.40129846432481707e-45 到 3.40282346638528860e+38 (正负值)。暂时不支持科学计数法，用它来存储会非常接近数字值。	1.2345679
DOUBLE	8 字节（64 位）双精度浮点数，范围在 4.94065645841246544e-324d 到 1.79769313486231570e+308d (正负值)。暂时不支持科学计数法，用它来存储会非常接近数字值。（numeric values）	1.2345678901234567
DECIMAL 十进制	DECIMAL 数据类型存储数据的精确值，它的范围在 1-1038 到 1039 -1 之间，默认定义格式是 decimal (10,0)。decimal (a,b) 中 a 代表小数点左边的最大位数，b 代表小数点右边的最大位数。	DECIMAL (3,2) for 3.14
BINARY	它只支持与 STRING 类型的转换，反之亦然。	1011
BOOLEAN	TRUE 或 FALSE。	TRUE
STRING	它使用单引号（'）或者双引号（"）来表达包含的字符串。Hive 使用 C 语言格式的字符串，最大溢出大小在 2 G 左右。	'Books' 或 "Books"
DATE	用来指定年月日，格式是 YYYY-MM-DD，范围从 0000-01-01 到 9999-12-31。	'2013-01-01'
TIMESTAMP	从 Hive 0.8.0 开始便支持该类型，它用来描述指定的年，月，日，时，分，秒，毫秒。格式是 YYYY-MM-DD HH：MM：SS[.fff...]。	'2013-01-01 12：00：01.345'

复杂数据类型

EasyCount 有 3 个主要的复杂数据类型： ARRAY，MAP 和 STRUCT。这些数据类型是建立在基本数据类型基础之上的。STRUCT 是一个 Record 类型，允许包含任意类型的字段。复杂数据类型允许嵌套类型。

复杂数据类型	描述	语法示例
ARRAY	数组是一组具有相同类型和名称的有序变量的集合。这些变量成为数组的元素，每个数组元素有一个编号，而且从 0 开始。例如：fruit[0]='apple'。	['apple','orange','mango']
MAP	Map 是一组无序的键值对元组的集合，使用数组表示法可以访问元素。键的类型必须是原子的，值可以是任何类型，同一个映射的键的类型必须相同，值的类型也必须相同。如果某个列的数据类型是 Map，其中 Key->value paris 对应的是 'first'->'John' 和 'last'->'Doe'， 那么可以通过字段名 ['last'] 获取最后一个元素： fruit['last']='Doe'。	map('first', 'John', 'last', 'Doe')
STRUCT	一组命名的字段。字段类型可以不同。STRUCT 和 C 语言中的 struct 或者"对象"类似，都可以通过"点"分隔符访问元素内容。默认情况下，STRUCT 字段名可以是 col1，col2，..... 您可以通过 structs_name.column_name 来访问具体的值： fruit.col1=1。	info struct< name:STRING, age:INT> (info.name 获取 name info.age 获取 age)

表达式

表达式是符号和运算符的一种组合，EasycCount 解析引擎处理该组合以获取单个值。简单表达式可以是常量、变量或者函数，可以用运算符将两个或者多个简单表达式联合起来构成更复杂的表达式。

通用表达式

通用表达式可以出现在 select 子句中，也可以出现在 where 或者 group by 子句中。
表达式的优先级从高到底如下表所示：

表达式	说明
IS [NOT] NULL， [NOT] LIKE， [NOT] BETWEEN，[NOT] IN	是否为空之类判断表达式。
AND、OR	多个条件之间"且"或者"或"的逻辑关系。

特殊表达式

FOREACH & EXECUTE 处理复杂数据类型时，可以使用 foreach 和 execute 语法进行循环和迭代处理。

函数

EasyCount 内部提供了很多函数给开发者使用，包括数学函数，类型转换函数，条件函数，字符函数，聚合函数，表生成函数等。其中大部分函数继承自 hive,另外也有一部分是 EasycCount 系统的自定义函数。

> **注明：__innertable(1000) 每个 1000 ms 输出一次，用于测试使用。innertable 中没有真实数据，可以用于验证函数。**

数值函数

函数名	语法	返回值类型	说明	示例
取整函数：round	round(double a)	bigint	返回 double 类型的整数值部分（遵循四舍五入）。	select round(3.1415926) from __innertable(1000) 结果：3
指定精度取整函数： round	round(double a, int d)	double	返回指定精度 d 的 double 类型。	select round(3.1415926,4) from __innertable(1000) 结果：3.1416
向下取整函数： floor	floor(double a)	bigint	返回小于或者等于该 double 变量的最大的整数。	select floor(3.1415926) from __innertable(1000) 结果：3
向上取整函数： ceil	ceil(double a)	bigint	返回大于或者等于该 double 变量的最小的整数。	select ceiling(3.1415926) from __innertable(1000) 结果：4
向上取整函数：ceiling	ceiling(double a)	bigint	与 ceil 功能相同。	select ceiling(3.1415926) from __innertable(1000) 结果：4
取随机数函数：rand	rand()，rand(int seed)	double	返回一个 0 到 1 范围内的随机数。如果指定种子 seed，则会得到一个稳定的随机数序列。	select rand() from __innertable(1000) 结果：0.5577432776034763 select rand(100) from __innertable(1000) 结果：0.7220096548596434 select rand(100) from __innertable(1000) 结果：0.7220096548596434
自然指数函数：exp	exp(double a)	double	返回自然对数 e 的 a 次方。	select exp(2) from __innertable(1000) 结果：7.38905609893065
自然对数函数：ln	ln(double a)	double	返回 a 的自然对数。	select ln(7.389) from __innertable(1000) 结果：2.0
以 10 为底对数函数：log10	log10(double a)	double	返回以 10 为底的 a 的对数。	select log10(100) from __innertable(1000) 结果：2.0
以 2 为底对数函数：log2	log2(double a)	double	返回以 2 为底的 a 的对数。	select log2(8) from __innertable(1000) 结果：3.0
对数函数：log	log(double base, double a)	double	返回以 base 为底的 a 的对数。	select log(4,256) from __innertable(1000) 结果：4.0

函数名	语法	返回值类型	说明	示例
幂运算函数：power	power(double a, double p)	double	返回 a 的 p 次幂，与 pow 功能相同。	select pow(2,4) from __innertable(1000) 结果：16.0
幂运算函数：pow	pow(double a, double p)	double	返回 a 的 p 次幂。	select power(2,4) from __innertable(1000) 结果：16.0
开平方函数：sqrt	sqrt(double a)	double	返回 a 的平方根。	select sqrt(16) from __innertable(1000) 结果：4.0
二进制函数：bin	bin(BIGINT a)	string	返回 a 的二进制代码表示。	select bin(7) from __nnertable(1000) 结果：111
十六进制函数：hex	hex(BIGINT a)	string	如果变量是 int 类型，那么返回 a 的十六进制表示；如果变量是 string 类型，则返回该字符串的十六进制表示。	select hex(17) from __innertable(1000) 结果：11 select hex(‘abc’) from __innertable(1000) 结果：616263
进制转换函数：conv	conv(BIGINT num, int from_base, int to_base)	string	将数值 num 从 from_base 进制转化到 to_base 进制。	select conv(17,10,16) from __innertable(1000) 结果：11
绝对值函数：abs	abs(double a) abs(int a)	double	返回数值 a 的绝对值。	select abs(-3.9) from __innertable(1000) 结果：3.9 select abs(10) from __innertable(1000) 结果：10
正取余函数：pmod	pmod(int a, int b),pmod(double a, double b)	int或double	返回正的 a 除以 b 的余数。	select pmod(9,4) from __innertable(1000) 结果：1 select pmod(-9,4) from __innertable(1000) 结果：3
正弦函数： sin	sin(double a)	double	返回 a 的正弦值。	select sin(0.8) from __innertable(1000) 结果：0.7173560908995228
反正弦函数：asin	asin(double a)	double	返回 a 的反正弦值。	select asin(0.7173560908995228) from __innertable(1000) 结果：0.8
余弦函数：cos	cos(double a)	double	返回 a 的余弦值。	select cos(0.9) from __innertable(1000) 结果：0.6216099682706644
反余弦函数：acos	acos(double a)	double	返回 a 的反余弦值。	select acos(0.6216099682706644) from __innertable(1000) 结果：0.9
positive 函数： positive	positive(int a)，positive(double a)	int或double	返回 a。	select positive(-10) from __innertable(1000) 结果：-10 select positive(12) from __innertable(1000) 结果：12

函数名	语法	返回值类型	说明	示例
negative 函数： negative	negative(int a)， negative(double a)	int或 double	返回 -a。	select negative(-5) from __innertable(1000) 结果：5 select negative(8) from __innertable(1000) 结果：-8

日期函数

函数名	语法	返回值	说明	示例
UNIX时间戳转日期函数：from_unixtime	from_unixtime(bigint unix_timestamp, string format)	string	转化 UNIX 时间戳（从1970-01-01 00：00：00 UTC 到指定时间的秒数）到当前时区的时间格式。	select from_unixtime(1493864893,'yyyy-MM-dd HH：mm：ss') from __innertable(1000) 结果：2017-05-04 10：28：13
获取当前 UNIX 时间戳函数：unix_timestamp	unix_timestamp()	bigint	获得当前时区的 UNIX 时间戳。	select unix_timestamp() from __innertable(1000) 结果：1493864893
日期转UNIX时间戳函数：unix_timestamp	unix_timestamp(string date)	bigint	转换格式为 "yyyy-MM-dd HH：mm：ss" 的日期到 UNIX 时间戳。如果转化失败，则返回0。	select unix_timestamp('2017-05-04 10：28：13') from __innertable(1000) 结果：1493864893
指定格式日期转UNIX时间戳函数：unix_timestamp	unix_timestamp(string date, string pattern)	bigint	转换 pattern 格式的日期到 UNIX 时间戳。如果转化失败，则返回 0。	select unix_timestamp('20170504 10：28：13','yyyyMMddHH：mm：ss') from __innertable(1000) 结果：1493864893
日期时间转日期函数：to_date	to_date(string timestamp)	string	返回日期时间字段中的日期部分。	select to_date('2017-05-04 10：03：01') from __innertable(1000) 结果：2017-05-04
日期转年函数： year	year(string date)	int	返回日期中的年。	select year('2013-12-08 10：03：01') from __innertable(1000) 结果：2013 select year('2012-12-08') from __innertable(1000) 结果：2012
日期转月函数： month	month (string date)	int	返回日期中的月份。	select month('2011-12-08 10：03：01') from __innertable(1000) 结果：12
日期转天函数： day	day (string date)	int	返回日期中的天。	select day('2017-05-04 10：03：01') from __innertable(1000) 结果：4
日期转小时函数： hour	hour (string date)	int	返回日期中的小时。	select hour('2017-05-04 10：03：01') from __innertable(1000) 结果：10
日期转分钟函数：minute	minute (string date)	int	返回日期中的分钟。	select minute('2017-05-04 10：03：01') from __innertable(1000) 结果：3
日期转秒函数：second	second (string date)	int	返回日期中的秒。	select second('2017-05-04 10：03：01') from __innertable(1000) 结果：1
日期转周函数：weekofyear	weekofyear (string date)	int	返回日期在当前的周数。	select weekofyear('2011-12-08 10：03：01') from __innertable(1000) 结果：49
日期比较函数：datediff	datediff(string enddate, string startdate)	int	返回结束日期减去开始日期的天数。	select datediff('2012-12-08','2012-05-09') from __innertable(1000) 结果：213

函数名	语法	返回值	说明	示例
日期增加函数：date_add	date_add(string startdate, int days)	string	返回开始日期 startdate 增加 days 天后的日期。	select date_add('2012-12-08',10) from __innertable(1000) 结果：2012-12-18
日期减少函数：date_sub	date_sub (string startdate, int days)	string	返回开始日期 startdate 减少 days 天后的日期。	select date_sub('2012-12-08',10) from __innertable(1000) 结果：2012-11-28

字符串函数

函数名	语法	返回值	说明	示例
字符串长度函数：length	length(string A)	int	返回字符串 A 的长度。	select length('abcedfg') from __innertable(1000) 结果：7
字符串反转函数：reverse	reverse(string A)	string	返回字符串 A 的反转结果。	select reverse(abcedfg’) from __innertable(1000) 结果：gfdecba
字符串连接函数：concat	concat(string A, string B…)	string	返回输入字符串连接后的结果，支持任意个输入字符串。	select concat(‘abc’ ,'def’ ,'gh’) from __innertable(1000) 结果：abcdefgh
带分隔符字符串连接函数：concat_ws	concat_ws(string SEP, string A, string B…)	string	返回输入字符串连接后的结果，SEP 表示各个字符串间的分隔符。	select concat_ws('-','abc','def','gh') from __innertable(1000) 结果：abc-def-gh
字符串截取函数：substr，substring	substr(string A, int start)，substring(string A, int start)	string	返回字符串 A 从 start 位置到结尾的字符串。	select substr('abcde',3) from __innertable(1000) 结果：cde
字符串截取函数：substr，substring	substr(string A, int start, int len)，substring(string A, int start, int len)	string	返回字符串 A 从 start 位置开始，长度为 len 的字符串。	select substr('abcde',3,2) from __innertable(1000) 结果：cd
字符串转大写函数：upper，ucase	upper(string A) ，ucase(string A)	string	返回字符串 A 的大写格式。	select upper('abSEd') from __innertable(1000) 结果： ABSED
字符串转小写函数：lower，lcase	lower(string A)，lcase(string A)	string	返回字符串 A 的小写格式。	select lower('abSEd') from __innertable(1000) 结果：absed
去空格函数：trim	trim(string A)	string	去除字符串两边的空格。	select trim(' abc ') from __innertable(1000) 结果：abc
正则表达式替换函数：regexp_replace	regexp_replace(string A, string B, string C)	string	将字符串 A 中的符合 Java 正则表达式 B 的部分替换为 C。注意，在有些情况下要使用转义字符,类似 oracle 中的 regexp_replace 函数。	`select regexp_replace('foobar', 'oo | ar', '-') from __innertable(1000 结果：f-b-
正则表达式解析函数：regexp_extract	regexp_extract(string subject, string pattern, int index)	string	将字符串 subject 按照 pattern 正则表达式的规则拆分，返回 index 指定的字符。	select regexp_extract('foothebar', 'foo(.*?)(bar)', 1) from__innertable(1000) 结果：the	

函数名	语法	返回值	说明	示例
URL 解析函数：parse_url	parse_url(string urlString, string partToExtract [, string keyToExtract])	string	返回 URL 中指定的部分。partToExtract 的有效值为：HOST，PATH，QUERY，REF，PROTOCOL，AUTHORITY，FILE 和 USERINFO。	select parse_url('http://facebook.com/path1/p.php?k1=v1&k2=v2#Ref1', 'HOST') from _innertable(1000) 结果：facebook.com select parse_url('http://facebook.com/path1/p.php?k1=v1&k2=v2#Ref1', 'QUERY','k1') from _innertable(1000) 结果：v1
json 解析函数：get_json_object	get_json_object(string json_string, string path)	string	解析 json 的字符串 json_string，返回 path 指定的内容。如果输入的 json 字符串无效，那么返回 NULL。	select get_json_object('{"fruit":apple,"ower":"tim"}','$.owner') 结果：tim
空格字符串函数：space	space(int n)	string	返回长度为 n 的空格字符串。	
重复字符串函数：repeat	repeat(string str, int n)	string	返回重复 n 次后的 str 字符串。	select repeat('abc',5) from _innertable(1000) 结果：abcabcabcabcabc
首字符 ascii 函数：ascii	ascii(string str)	int	返回字符串 str 第一个字符的 ascii 码。	select ascii('abcde') from _innertable(1000) 结果：97
分割字符串函数： split	split(string str, string pat)	array	按照 pat 字符串分割 str，会返回分割后的字符串数组。	select split('abtcdtef','t') from _innertable(1000) 结果：["ab","cd","ef"]
集合查找函数：find_in_set	find_in_set(string str, string strList)	int	返回 str 在 strlist 第一次出现的位置，strlist 是用逗号分割的字符串。如果没有找该 str 字符，则返回 0。	select find_in_set('ab','ef,ab,de') from _innertable(1000) 结果：2

条件函数

函数名	语法	返回值	说明	示例
If 函数：if	if(boolean testCondition, T valueTrue, T valueFalseOrNull)	T	当条件 testCondition 为 TRUE 时，返回 valueTrue；否则返回 valueFalseOrNull。	select if(1=2,100,200) from _innertable(1000) 结果：200 select if(1=1,100,200) from _innertable(1000) 结果：100
nvl 函数：nvl	nvl(T value, T default_value)	T	如果 value 值为 NULL 就返回 default_value，否则返回 value。	select nvl(null,100) from _innertable(1000) 结果：100
isnull 函数：isnull	isnull(T value)	true 或 false	如果 value 值为 NULL 就返回 true，否则返回 false。	select isnull(null) from _innertable(1000) 结果：true
isnull 函数：isnotnull	isnotnull(T value)	true 或 false	如果 value 值为 NULL 就返回 true，否则返回 false。	select isnotnull(null) from _innertable(1000) 结果：false

函数名	语法	返回值	说明	示例
条件判断函数：CASE	CASE WHEN a THEN _a [WHEN b THEN _b]* [ELSE _c] END	T	如果 a 为 TRUE，则返回 _a；如果 b 为 TRUE，则返回 _b；否则返回 _c 。	select case when 1=2 then 'tom' when 2=2 then 'mary' else'tim' end from __innertable(1000) 结果：mary

类型转换函数

函数名	语法	返回值	说明	示例
类型转换函数：cast	cast(expr as < type >)	Expected "=" to follow "type"	返回 array 类型的长度。	select cast(1 as bigint) from __innertable(1000) 结果：1

集合函数

标题1	标题2	标题3	标题2	标题3
Map 类型长度函数： size(Map< K.V >)	size(Map< K.V >)	int	返回 map 类型的长度。	select size(map('100','tom','101','mary')) from __innertable(1000) 结果：2
Array 类型长度函数： size(Array< T >)	size(Array< T >)	int	返回数组类型的长度。	select size(array(‘aa’ ,’ bb’)) from __innertable(1000) 结果：2

聚合函数

函数名	语法	返回值	说明	示例
个数统计函数：count	count(col)	int	count(expr) 统计检索出的行的个数，返回指定字段的个数。	
总和统计函数：sum	sum(col)	double	sum(col) 统计结果集中 col 的相加的结果；sum(DISTINCT col) 统计结果中 col 不同值相加的结果。	select sum(t) from lxw_dual 结果：100
平均值统计函数：avg	avg(col)	double	avg(col) 统计结果集中 col 的平均值；avg(DISTINCT col) 统计结果中 col 不同值相加的平均值。	select avg(t) from lxw_dual 结果：50
最小值统计函数：min	min(col)	double	统计结果集中 col 字段的最小值。	select min(t) from lxw_dual 结果：20
最大值统计函数：max	max(col)	double	统计结果集中 col 字段的最大值。	select max(t) from lxw_dual 结果：120

扩展函数

函数名	语法	返回值	说明
个数统计函数：countd，countd_hllp	countd(col)，countd_hllp(col)	int	count(col) 去重统计检索出的行的个数，返回指定字段的个数。该函数为非精确去重统计，精确度在 99.5% 左右。
去重合并统计函数：hllp_merge	hllp_merge(a,b)，参数为 binary	binary	去重合并统计函数是将两个二进制集合去重合并生成以一个新的集合。
获取去重后的结果函数：hllp_get	hllp_get(a)	bigint	获取去重后的统计结果。

说明：以上三个函数都是基于 HyperLogLog 算法的实现，为了做大规模去重统计，降低存储空间，精确度上有一点误差。

Sql示例：

```
with (select iActivityId, hllp(uiUin) uvb, from_unixtime(AGGRTIME DIV 1000, "yyyy-MM-dd HH:mm:ss") ts,
concat_ws('-', 'd', from_unixtime((AGGRTIME DIV 1000), 'yyyy-MM-dd 00:00:00'), cast(iActivityId as string)) dk
from src GROUP BY iActivityId
COORDINATE BY unix_timestamp(dteventTime, 'yyyy-MM-dd HH:mm:ss')*1000 WITH AGGR INTERVAL 60 SECONDS) tmp,
(select iActivityId, hllp_merge(tmp.uvb, dim.uvball) uvball, tmp.dk k, ts from
tmp left join dim on tmp.dk=dim.k) jd
insert into dim with k as KEY select jd.k k, jd.uvball uvball from jd,
insert into dest select jd.iActivityId, hllp_get(jd.uvball) duv, jd.ts from jd
```

以上为 sql 部分，下面是 sql 中使用的表的描述信息。该信息根据用户在页面上库表配置的内容自动生成。

```
[tabledesc-1]
table.name=src
table.fields=iActivityId,int,:dteventTime,string,:uiUin,string,
table.field.splitter=|

[tabledesc-dimtable-tde]
table.name=dim
table.fields=k,string,:uvball,binary,
table.field.key=k

[tabledesc-destination]
table.type=tpg
table.name=dest
table.fields=iActivityId,int,:duv,bigint,:dteventTime,string
```

说明：改向配置用户描述每条数据的结构，统计计算过程会根据结构解析每一行数据。

任务管理

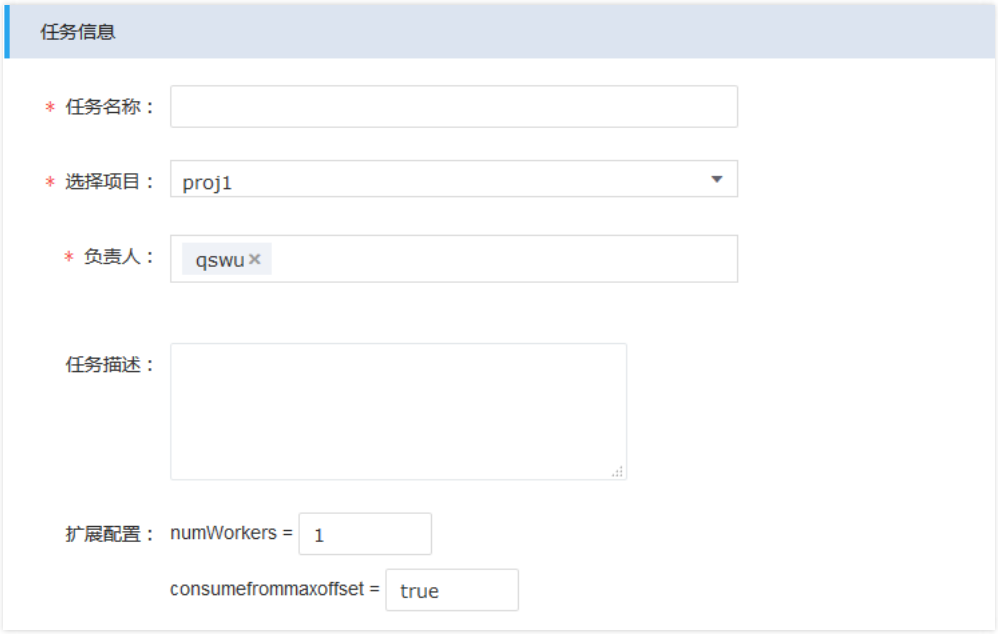
最近更新时间：2018-05-25 18:16:05

任务管理模块主要用于对 EasyCount 任务进行管理，主要包括 EasyCount 任务的创建、运行、停止、删除、复制和任务导入导出。下面通过一个操作流程来介绍各个功能。
首先登录腾讯大数据处理套件，单击【实时计算】模块。

单击【任务管理】导航进入任务管理列表。

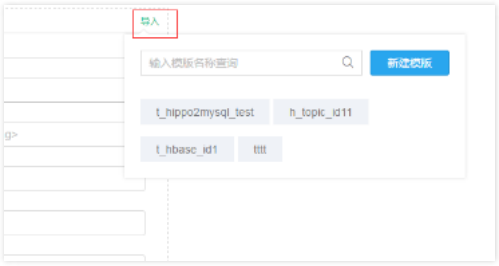
单击【新建任务】按钮，进入新建任务页面。新建任务的信息分为三块：任务信息、库表配置和脚本编辑。

- 任务信息：任务的基本配置信息，包括任务名称、任务描述、选择任务的责任人及所属项目。***扩展配置 numWorkers 代表任务在 storm 上需要启动多少个 worker。***



- 库表配置：配置脚本信息中需要使用的表信息，库表分为源表、目标表和纬表三类。源表只能为流水表，代表计算数据的主要来源。目标表是计算的存储位置。纬表是一些静态数据，在计算过程需要关联的表。

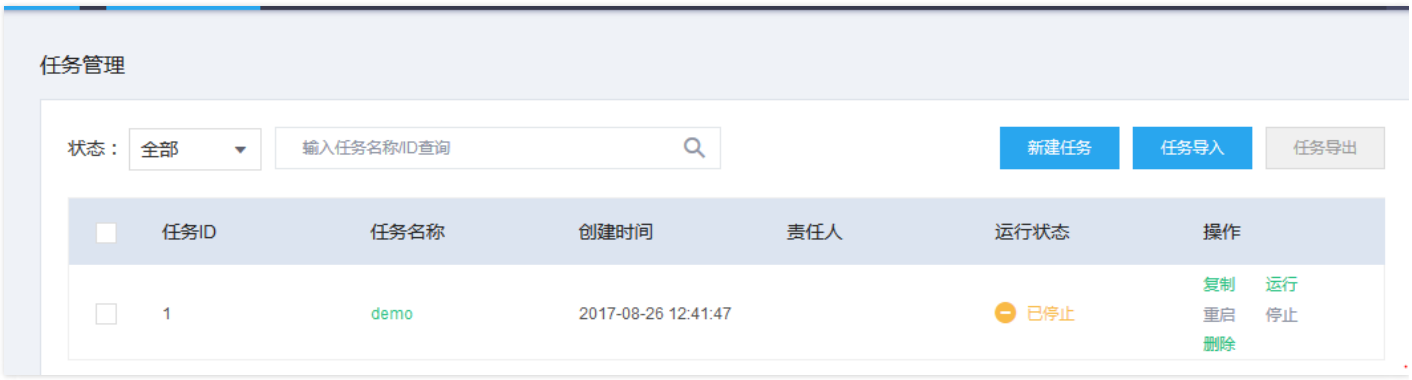
库表配置可以单击【导入】从模板中选择，或进一步单击【新增模版】。每个库表信息填写完整后可以单击测试验证该库表是否连通。



注意：在申请使用 hippo 表时需要配置生产组和消费组权限，在使用 kafka 表时需要配置生产组权限。Hippo 和 kafka 在使用之前，分别需要先申请配置生产组和消费组权限，生产组权限。Hippo 权限申请参考《tbds-hippo使用手册》。
Kafka topic 和 hbase 表的权限分配需要联系管理员申请。以上所需的 kafka、hippo 和 hbase 表的权限申请通过之后才能在 SQL 中使用，否则在任务运行过程中会有权限异常抛出。

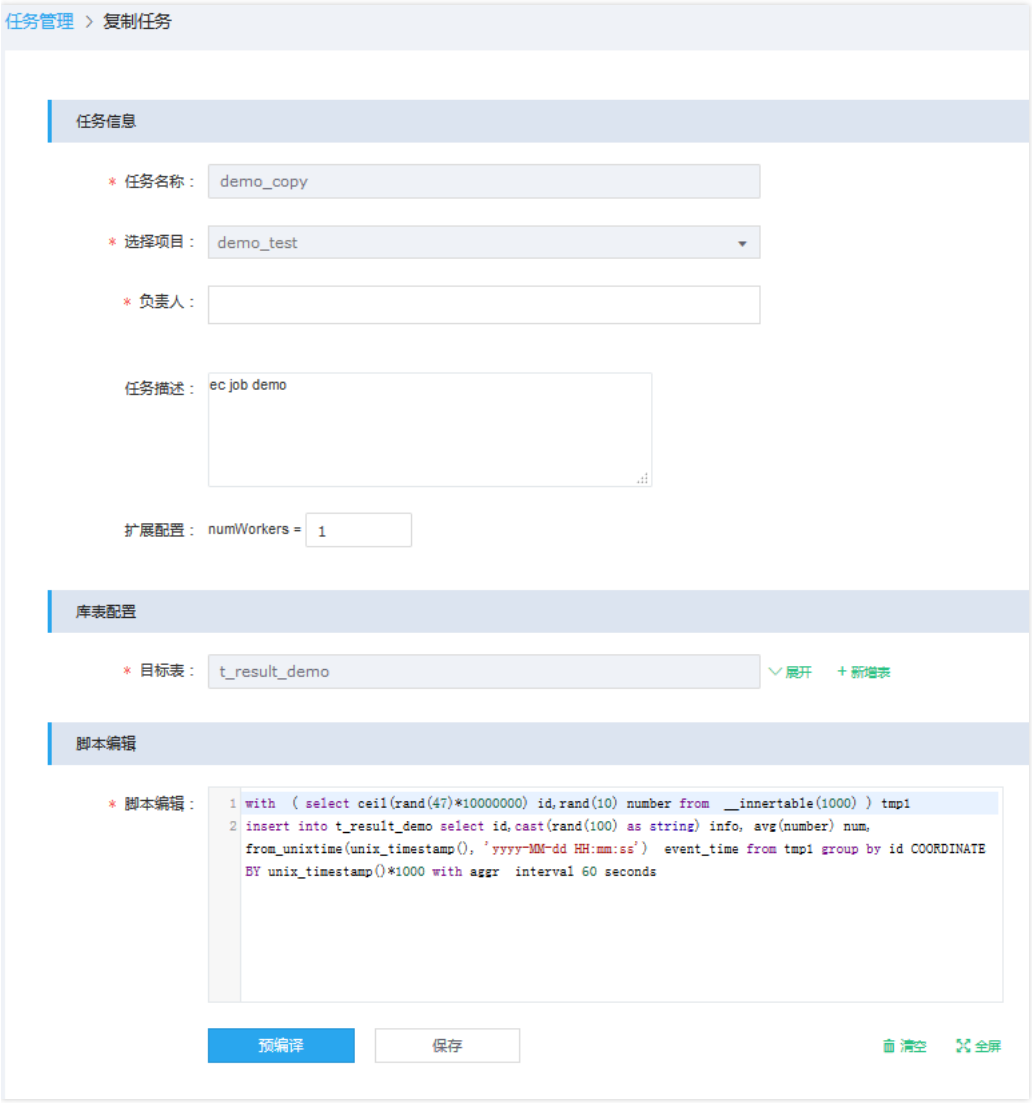
- 脚本编辑：编辑计算的 sql 脚本。

以上三个模块的信息都填写完整后，单击【预编译】按钮，查看脚本是否能编辑通过。脚本编译异常会有错误提示，脚本编辑通过后单击【保存】，进入任务列表页面单击对应任务的【运行】按钮来启动任务。任务会异步下发到 storm 集群，任务的状态会每隔 30 s 刷新一次。



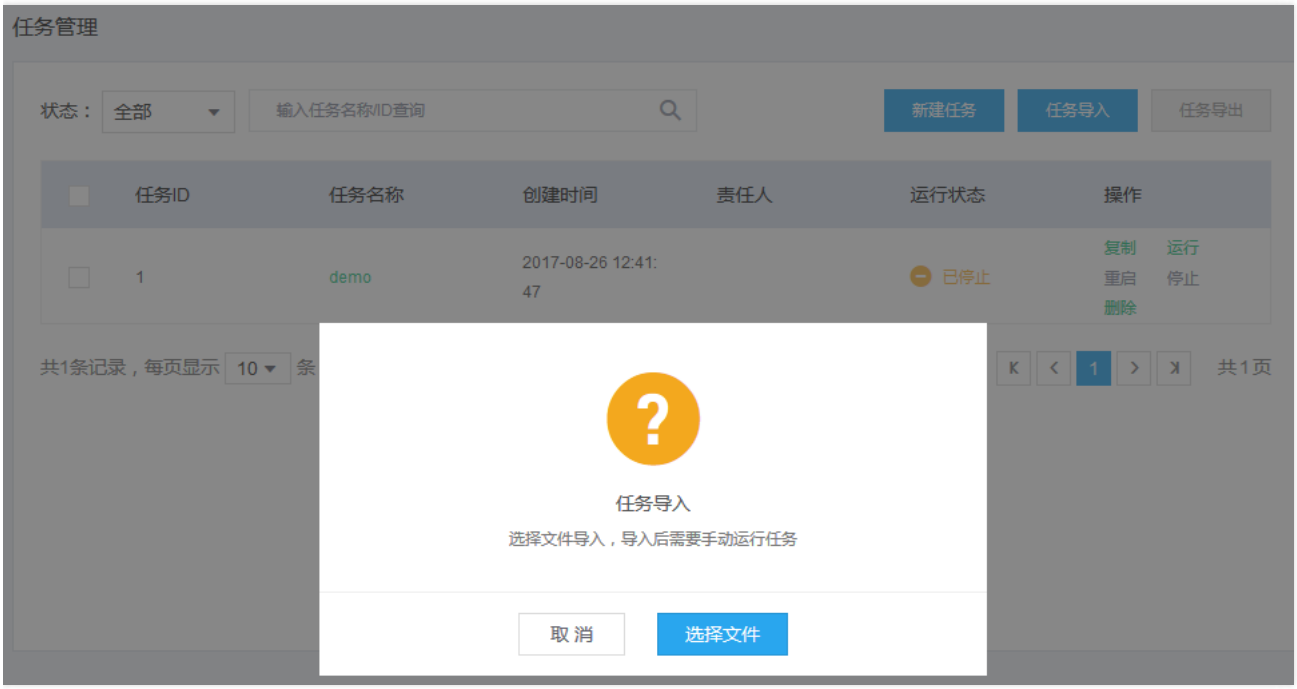
进入任务列表页面单击对应任务的【停止】或【删除】按钮，即可停止或删除一个任务。

如果对某一个任务单击【复制】按钮，则会复制一个执行相同操作的任务。配置信息和原任务相同。



任务导出功能：为了将已有的任务方便的迁移到其他平台，可以选择任务单击【任务导出】，任务脚本会打包下载，解压后可以看到任务脚本。

任务导入功能：选择任务单击【任务导入】，将已有的任务脚本导入到系统生成任务。



库表模版管理

最近更新时间：2018-12-19 11:18:42

库表模板可以减少用户在创建任务时对表信息重复编辑。对于经常使用的表，用户可以先创建库表模板，在创建任务时从库表模板中选择库表。

> ⚠ **注意：**
> 用户在创建任务选择从库表模板中导入库表时建议对表做必要修改，例如修改表名，否则多个任务将同时访问一个表，此表将会成为性能瓶颈。

操作步骤

1. 单击【库表模板管理】进入库表模块。

2. 单击【创建模块】进入创建库表页面，填写库表内容，内容填写完整后可以先单击测试，验证库表是否连通。测试连通后，单击【新建】保存模板。

Stream S-QL 实战案例

最近更新时间：2018-05-25 18:27:28

前置条件：sql 使用的所有非临时表必须事先定义好，并且测试通过，否则在任务运行过程会有异常抛出。

案例1：星级游戏实时指标统计

需求描述

实时统计多款星级游戏的注册人数。每款游戏的日志数据记录在各自的注册流水表，以 1 分钟为单位实时统计每款游戏的注册人数，并将结果数据写入到用户指定结果数据表。 目前对以下 4 款游戏进行计算：御龙在天，英雄联盟。

实现方案

首先对每一款游戏的各个指标分别进行统计，然后将每款游戏相同的结果指标通过 UNION 的方式合并并存储结果表。完整的 sql 如下：

```
with(select "ylzt" bname, iUin qqid, dtEventTime agtime from ylzt_dsl_createrole_fht0) ylzt_cr,
(select "lol" bname, iUin qqid, dtEventTime agtime from lol_dsl_createrole_fht0) lol_cr,
(select "bns" bname, iUin qqid, dtEventTime agtime from bns_dsl_createrole_fht0) bns_cr,
(select "nizhan" bname, iUin qqid, dtEventTime agtime from nizhan_dsl_createrole_fht0) nizhan_cr
insert into rtc_idata_createrole_m select from_unixtime(AGGRTIME DIV 1000, 'yyyyMMddHHmm') vdatetime_m, bname vgame, count(qqid) from (select
bname, qqid, agtime from ylzt_cr union all select bname, qqid, agtime from lol_cr union all select bname, qqid, agtime from bns_cr union all select bna
me, qqid, agtime from nizhan_cr) utbl group by bname coordinate by agtime*1000 with aggr interval 60 seconds
```

1. 在上述 sql 中，首先对每个表选出需要使用的字段。对于目前的需求，只用到 iUin 和 dtEventTime 字段，因此将这两个字段选出来。同理，对于收入流水表，选出收入相关字段。

   > 要点：这段sql中使用了with语法，将一段子查询逻辑抽象出来作为临时表，以供后面的逻辑使用。

2. 将每款游戏的数据 UNION 到一起，然后进行聚合统计。首先从上一段 sql 中的临时表中把每个游戏的注册信息进行 UNION，然后进行聚合统计计算，最后将数据写入目标表。Group by bname 定义了聚合分组字段为 bname，coordinate by 定义了时间协调坐标，这里要求一个毫秒数，而 agtime 是源数据表中定义的 dtEventTime 字段是一个秒数，所以这里乘以 1000。Aggr interval 60 seconds 表示聚合粒度为 1 分钟。Count(qqid) 以及 sum(m) 就是聚合统计。另外AGGRTIME 是一个系统关键字，表示聚合时间的毫秒数，并且是 aggr interval 的整数倍。这个字段只有在当前查询（子查询）中包含 group by 时才有意义，否则使用这个关键字会报错。

案例2：累加聚合和滑动窗口聚合

需求描述

假定某款业务需要按照如下规则进行实时统计： 每1分钟进行一次数据统计，输出当前分钟的登录人数，当前小时截止到当前分钟的登录人数，截止到当前分钟的连续30分钟的登录人数。

实现方案

使用累加聚合和滑动窗口聚合两种方法来实现。完整的 sql 如下：

```
INSERT INTO dest SELECT appId, count(qq), count(qq ACCU), count(qq SW)
FROM src
GROUP BY appId
COORDINATE BY dTime WITH AGGR INTERVAL 60 SECONDS WITH ACCU INTERVAL 3600 SECONDS WITH SW INTERVAL 1800 SECONDS
```

说明：在 sql 中通过 WITH AGGR INTERVAL 60 SECONDS 指定聚合窗口为 1 分钟，通过 WITH ACCU INTERVAL 3600 SECONDS 指定累加聚合窗口为 1 小时，通过 WITH SW INTERVAL 1800 SECONDS 指定滑动窗口为 30 分钟。 Count(qq) 表示当前分钟的计数，count（qq ACCU) 表示当前小时截止到当前分钟的计数，count(qq SW) 表示当前分钟及向前 30 分钟的滑动窗口计数。

案例3：关联纬表实时指标统计

需求描述

1. 实时计算微信在不同国家和地区，不同的设备的连接数和平均连接耗时。微信连接数据中，每条记录包含一个连接 IP，连接设备类型（Android，ios），连接耗时等信息。
2. 存在一张 IP 表，包含 IP 段到地区的映射关系（纬表）。
3. 要求按照 1 分钟的统计粒度计算每个国家每种设备的连接数和平均连接耗时。

实现方案

对于每条数据首先针对 IP 通过纬表关联的方式从 IP 表中读取纬表中的国家和地区信息，然后按照国家和设备进行聚合统计。完整的 SQL 如下：

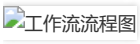
```
INSERT INTO weixin_res_meta
SELECT from_unixtime(AGGRTIME DIV 1000, 'yyyyMMddHHmm'), ttt.Device_, ip_table.country_id,
COUNT(1), avg(CostTimeref_) FROM (SELECT Device_, CostTimeref_, CAST ((ClientIP_ - ClientIP_ % 256) AS STRING) AS ipstr
FROM log_10410 WHERE Funid_='138') ttt LEFT JOIN ip_table ON ttt.ipstr = ip_table.ipint GROUP BY ttt.Device_, ip_table.country_id
COORDINATE BY TimeStamp_*1000 WITH AGGR INTERVAL 60 SECONDS
```

工作流

工作流概述

最近更新时间：2017-12-07 16:51:38

工作流是腾讯自研的任务调度系统，是 TBDS（大数据处理套件）最重要的功能之一，具有毫秒级下发任务，高可靠的特性，同时支持插件式扩展任务类型。其主要工作流程如下：



我们在快速入门中已经对工作流做过简单介绍，后续教程将详细介绍工作流的具体操作。

工作流操作
创建工作流

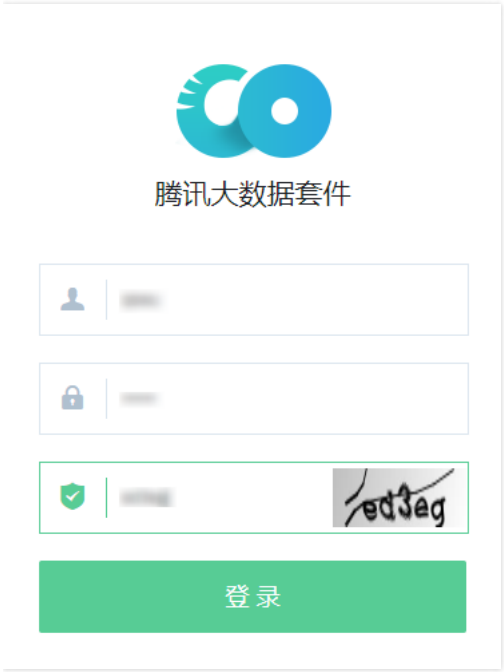
最近更新时间：2017-12-07 16:52:01

工作流是腾讯自研的任务调度系统。作为 TBDS 最重要的功能之一，工作流对于数据处理的基本流程包含数据接入，计算和数据导出三部分，如下图：

创建工作流的步骤如下：

1. 进入工作流

首先登录 腾讯大数据处理套件。



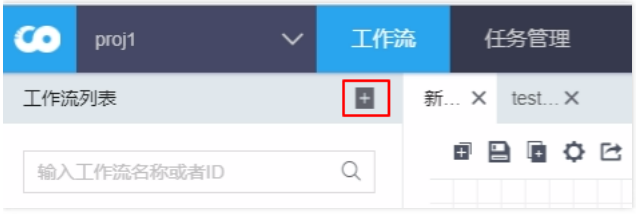
进入业务主界面后，单击【工作流】，即可进行工作流和任务的创建。

2. 项目选择

下拉框会显示当前登录账号加入的项目，选择一个项目，在项目中创建工作流。

3. 创建新工作流

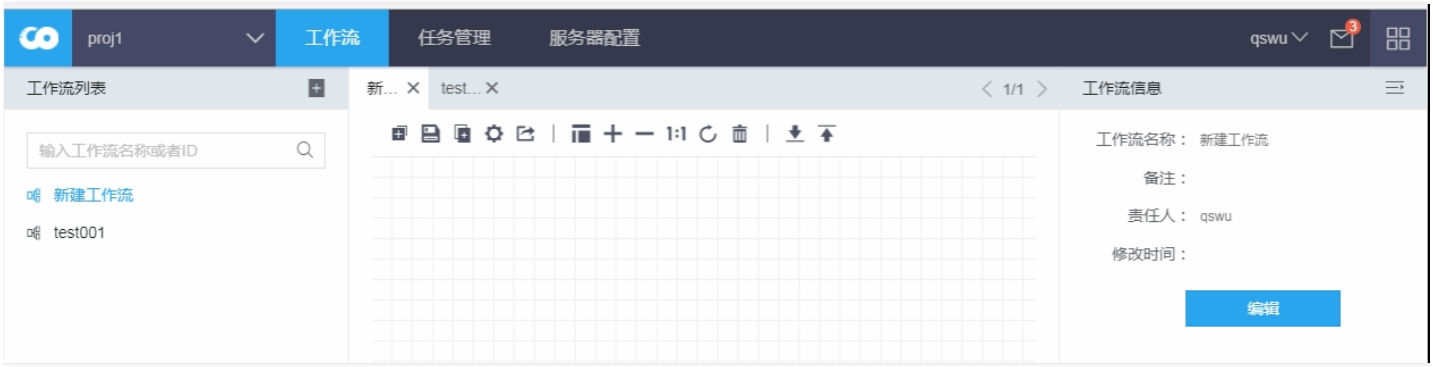
在工作流的页面中，有创建工作流的入口，单击【+】即可新建一个工作流，如图：



依次填写信息后，单击【确定】按钮，如下图：

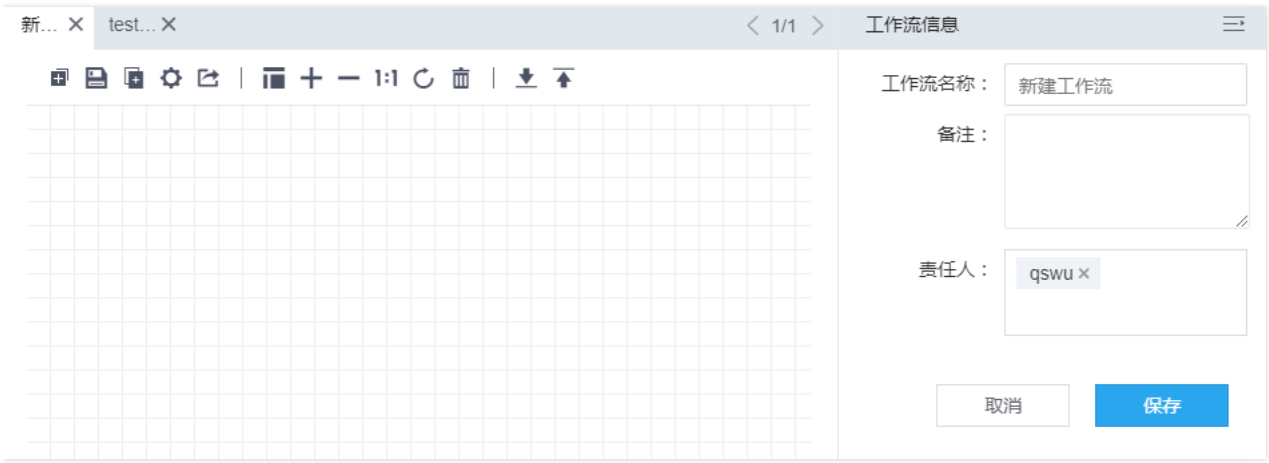
4. 工作流配置

创建工作流后工作界面如下：



单击右侧【编辑】按钮，配置工作流，编辑完成后单击【保存】即可。

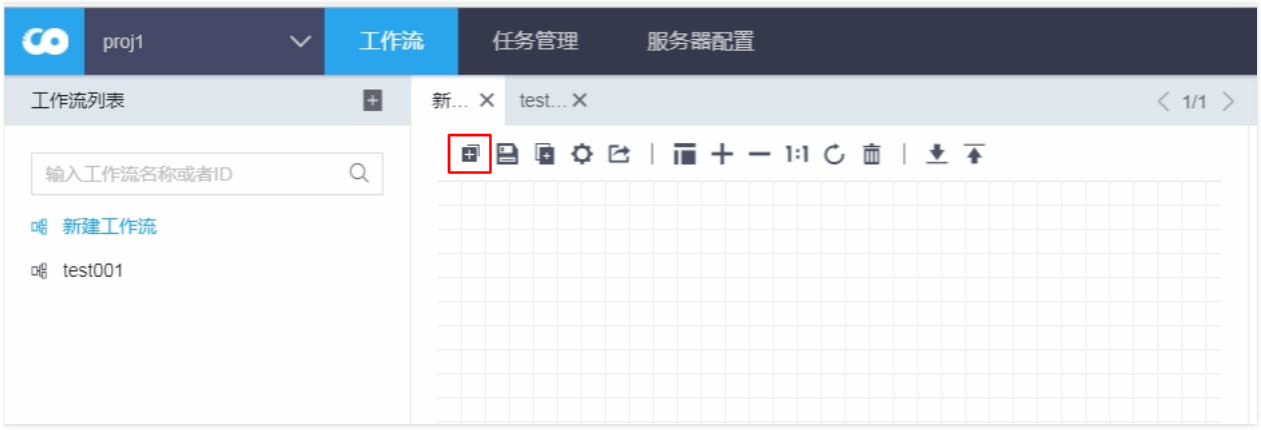
- 工作流名称（可修改）：工作流的名称，不唯一；
- 备注（可修改）：工作流的描述信息；
- 责任人（必填，多选）：责任人只能是"所属项目"中的项目成员，可以填多个；
- 修改时间（只读）：展示最近修改时间。


新...
test...
1/1
工作流信息
工作流名称：
新建工作流
备注：
责任人：
qswu
取消
保存

创建任务

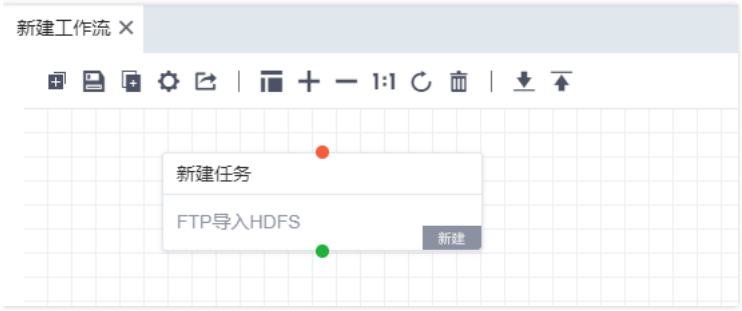
最近更新时间：2017-12-07 16:52:12

创建工作流之后，得到一个空白的工作流画布，可以在该画布上部署不同类型的任务。拖动工作流列表右侧的【+】图标，摆放到任意位置，就会弹出所有的工作流任务。



在搜索框里搜索常用的任务，选择一个任务后，单击【确定】，任务即创建成功，如下图：

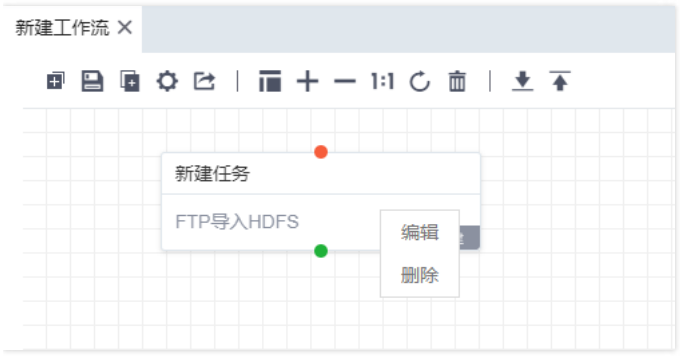
任务创建成功后，工作界面会弹出新建任务的小窗口，如下图：



编辑任务

最近更新时间：2018-05-25 18:35:19

在当前任务窗口右键单击【编辑】。



填写配置信息，任务编辑完成后将在下次运行时生效。

配置说明：

（1）源服务器和目标服务器：用户做数据分析处理时需将外部数据从源服务器导入到平台，最终将处理结果导出到目标服务器。配置详情请参见 服务器配置。

（2）源目录：FTP 上面源文件所在的目录，例： /stage/outface/sng/test_table/${YYYYMMDD}/ 。

（3）文件名正则表达式：支持时间格式${YYYYMMDD}, .*为所有文件，基于 Java 正则规则。

（4）是否遍历子目录：选择是，则递归遍历子目录。

（5）最大遍历层数：默认递归遍历 5 层。

（6）目标目录：HDFS 上存放文件的目录，例： /stage/outface/sng/test_table/${YYYYMMDD}/ 。

（7）重试次数：重试次数。（重试次数为 0，任务不下发）

（8）步长：间隔多少周期执行一次。(若是天任务，步数为 2，每两天执行一次)。

（9）代理 IP：任务执行所在机器 IP。

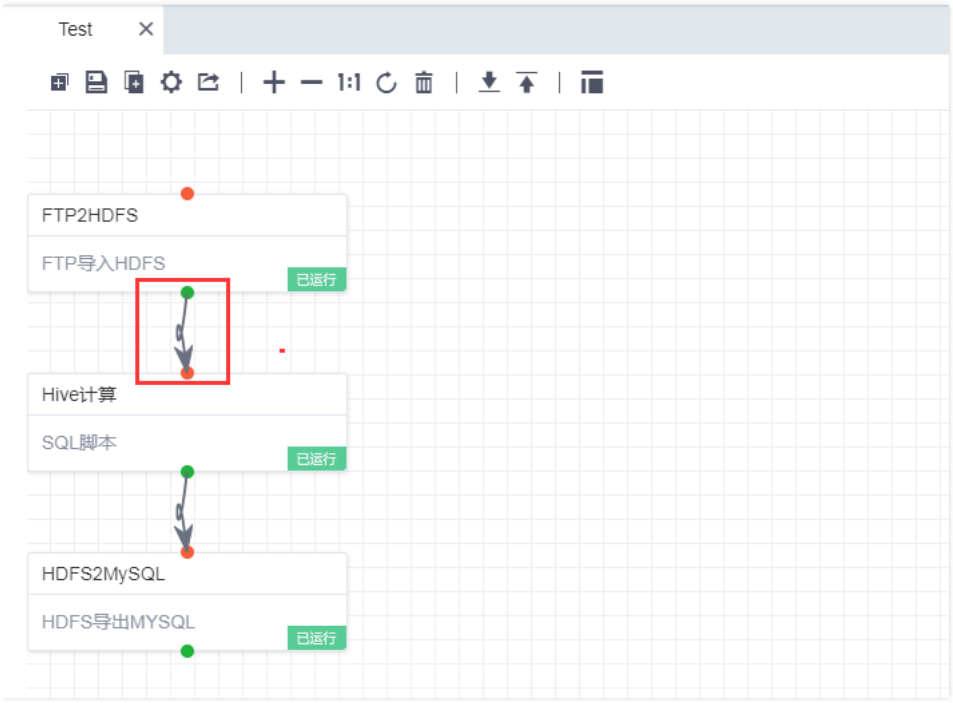

基本信息
调度设置
参数配置
* 周期类型：
天
* 起始数据时间：
2017-09-21
周期类型和起始数据时间一旦选定保存后将不允许再修改。
任务获取数据的时间起点，任务将于下一个周期（执行时间）调度运行。例如：数据时间为3点的小时任务，将会在下一个周期4点运行。
* 自身依赖：
是
否
并行
调度时间：
0点
0分
任务到了执行时间后，会按照这里的配置延迟一定时间才会开始执行。


基本信息
调度设置
参数配置
参数
值
* 终止时间：
2019-09-22
是否可重试：
是
* 源服务器：
petertestftp
编辑服务配置
* 目标服务器：
petertesthdfs
编辑服务配置
* 源目录：
* 文件名正则表达式：
FTP上面源文件所在的目录，例：/stage/outface/sng/test_table/${YYYYMMDD}/
* 是否遍历子目录：
否

工作流任务继承关系建立

最近更新时间：2017-12-07 16:52:33

各个工作流任务之间可通过拖动任务块下方箭头，建立任务之间的继承关系。建立任务间的继承关系，就是建立任务间的依赖关系。例如任务 A 和任务 B 有依赖关系，那么同周期 A 任务实例成功运行后，同周期 B 任务实例才会开始运行，否则为等待下发状态。



如果要删除某个继承关系，只需将鼠标移动到该继承关系相应的箭头，右键单击【删除】即可。

任务运行和运行过程查看

最近更新时间：2017-12-07 16:52:42

运行任务

工作流任务编辑完成后，就可以开启任务的运行。在任务上右击选择【运行】即可触发任务开始运行。

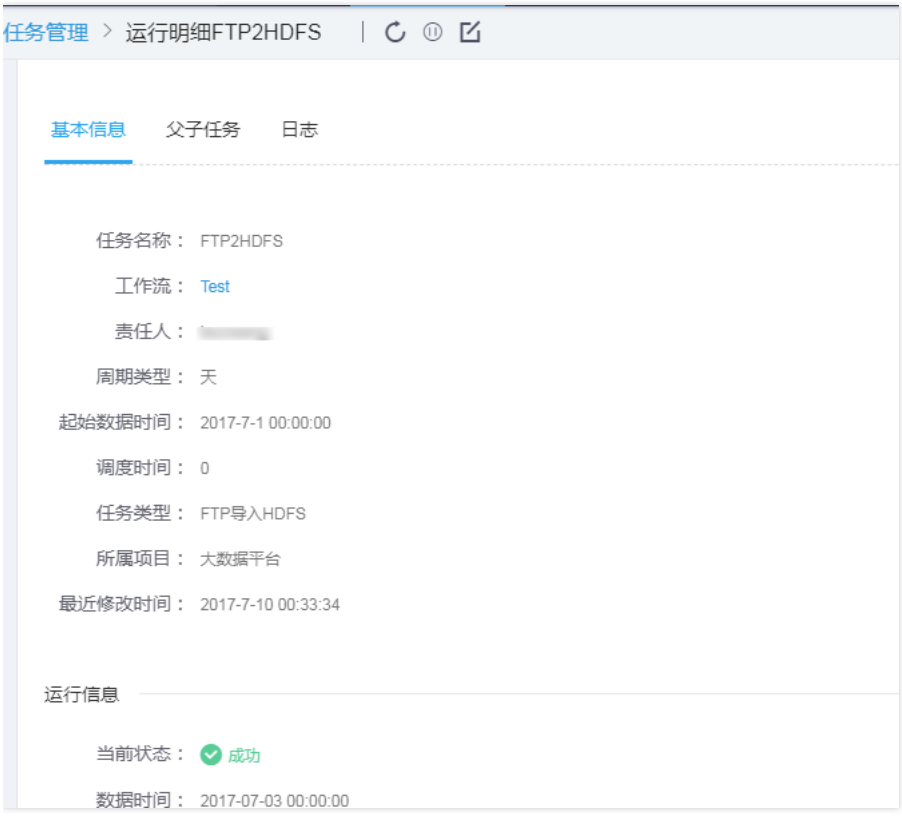
单击【运行】后，会提示用户任务需要审批才可运行；审批权限为项目管理员所有，可选择审批通过后自动运行，单击【确定】。

查看运行状态

右击工作流任务选择【查看运行状态】，可以打开查看该任务各个周期任务运行的详细状态。

查看运行明细

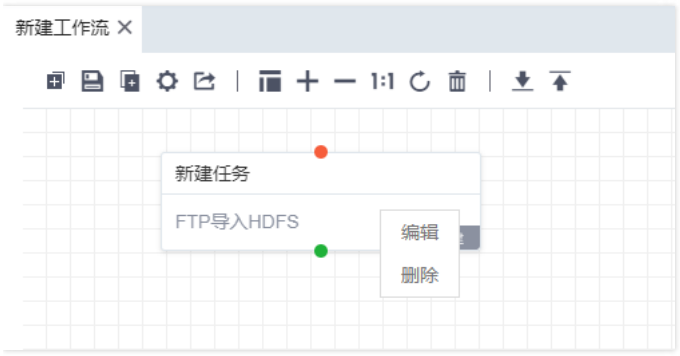
单击日志下的【查看】可以查看详细运行内容，也可以单击【更多实例】>【任务名称】下的内容，到详细的父子任务查看页面。



删除任务

最近更新时间：2017-12-07 16:52:51

对新建未运行的任务、已经停止的任务可以进行删除，右键单击【删除】即可，如下图：



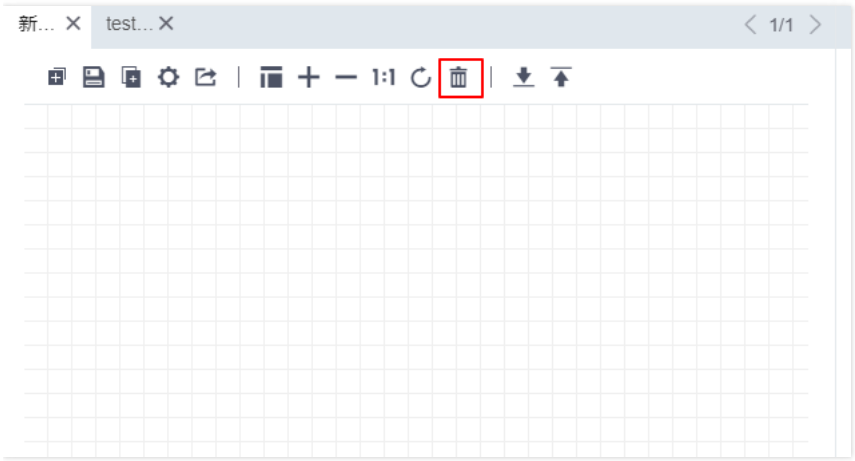
注意：如果任务正处于运行状态，则不能删除。

单击【删除】后弹出窗口，单击【确定】完成删除，如下图：

删除工作流

最近更新时间：2017-12-07 16:53:02

首先在画布的工具栏中，单击【删除】图标。



单击【删除】图标后，弹出确认删除窗口，单击【确定】即可删除整个工作流。

工作流开发
服务器配置

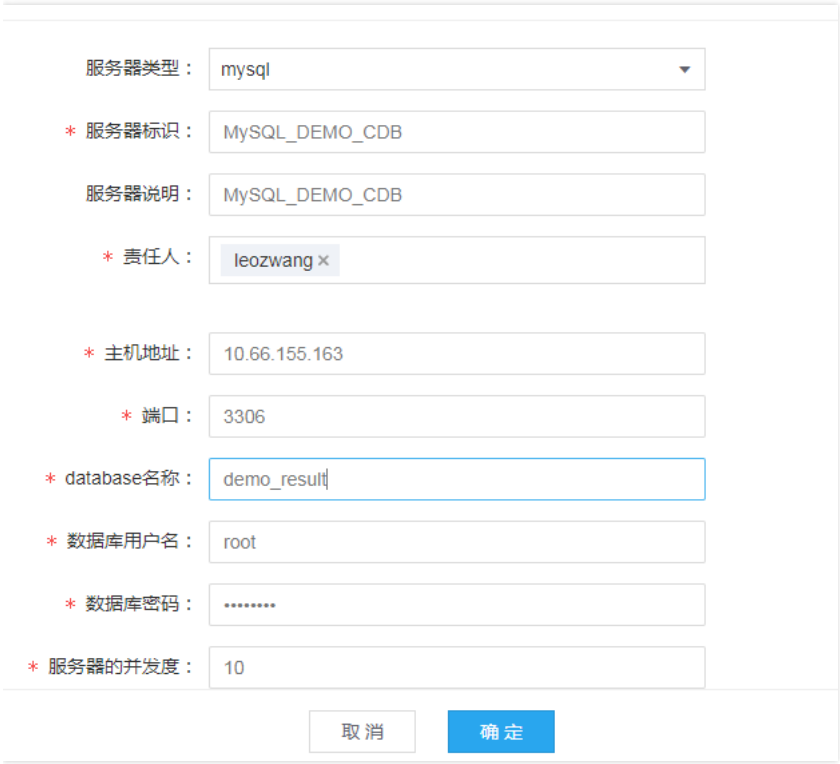
最近更新时间：2017-12-07 16:53:25

腾讯大数据处理套件是集项目管理，资源管理，计算工具，资源调度等功能为一体的平台工具。本身不创生数据也不消灭数据，故用户需要做数据分析处理时需将外部数据导入到平台，这样就需要配置各种各样的数据源，以及配置套件内的计算资源等。配置入口如图：

新增 MySQL 配置

此配置一般用于外部数据源或最终计算结果数据仓库。
配置说明：
（1）服务器标识：即系统里显示名称；
（2）服务器说明：备注说明；
（3）责任人：设置该项配置的负责人；
（4）端口：MySQL 服务远程连接端口；
（5）数据库用户名：可远程访问 MySQL 服务并可读写相关库表的用户；
（6）主机地址：MySQL 服务远程地址；
（7）database 名称：数据源数据库名称；
（8）数据库密码：远程访问 MySQL 服务时该用户密码；

（9）服务器并发度：远程连接 MySQL 服务的最大并发连接数。



新增 HDFS 配置

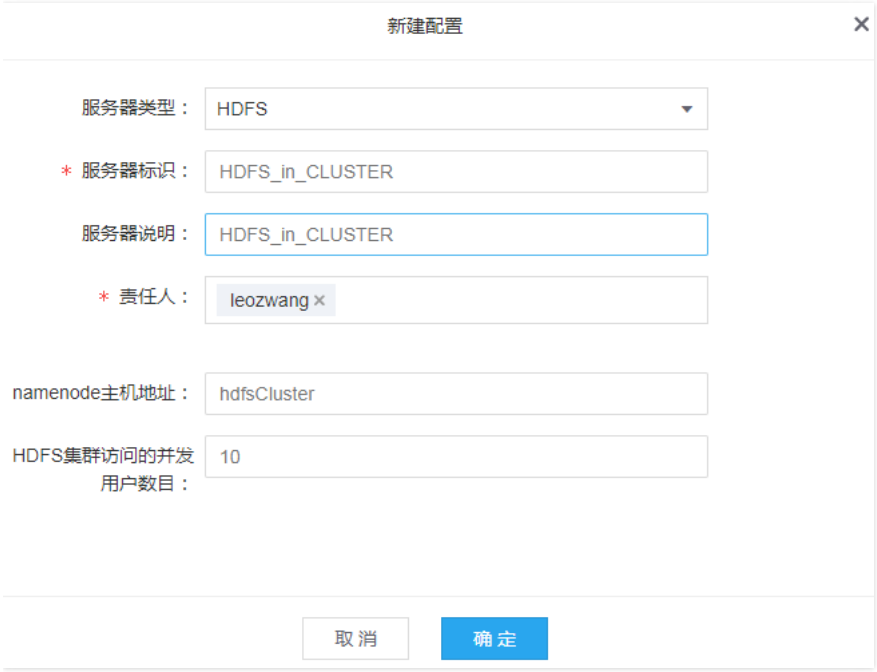
HDFS配置即套件内的资源配置。

配置说明：

（1）服务器标识，说明即系统内标识和备注信息；

（2）namenode 主机地址：套件对 namenode 有做高可用性集群，故填写 hdfsCluster 即可；

（3）HDFS集群访问的并发用户数目：即通过 namenode 访问 HDFS 资源最大并发用户数。



新增 hive 配置

hive配置即套件内的HIVE资源配置，配置说明：
（1）服务器标识，说明即系统内标识和备注信息；
（2）端口即 HiveServer 的端口（默认10000）；
（3）主机地址：即 HiveServer 部署的主机（默认已将集群的 HiveServer 地址自动填充）。

> 注意：这里 hive 的认证信息均使用集群的 LDAP 进行。

新增其他配置

其他的数据源可设置 Postgre，Oracle，mrgroup 等，配置方法同前。

离线工作流的使用

最近更新时间：2018-05-25 18:36:25

离线工作流即对实时性要求不高，可接受 T+1 的数据计算结果的任务类型，下面以一个实际的离线工作流为例子进行介绍。

示例任务介绍

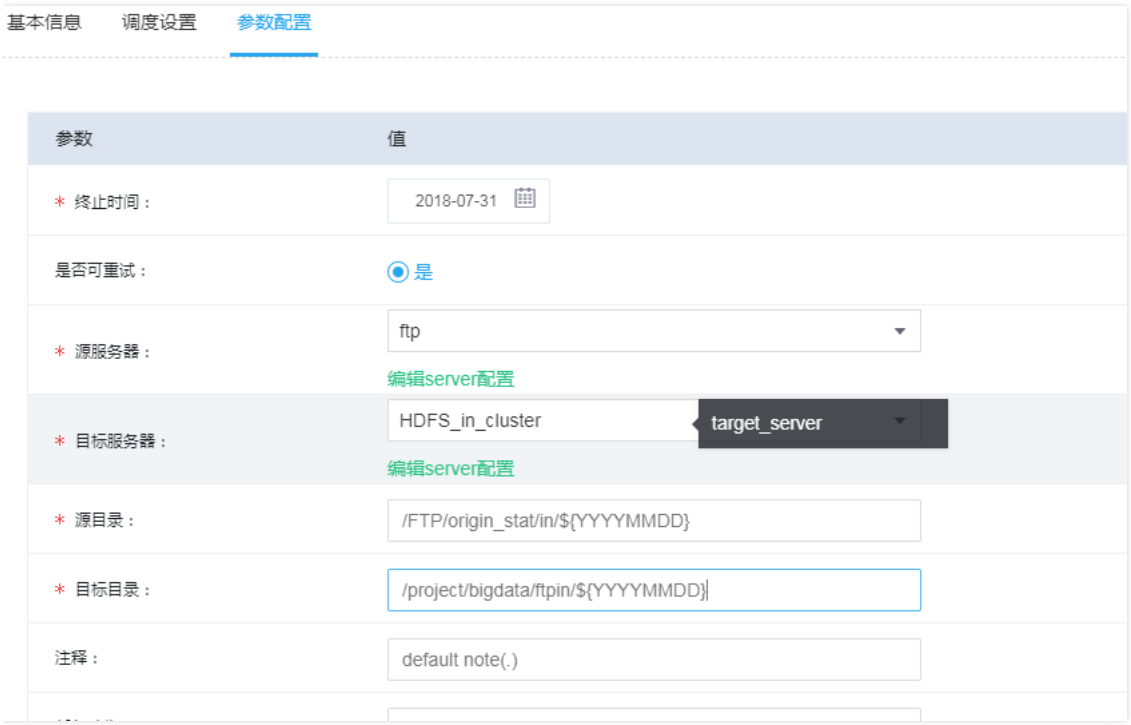
首先创建一个 FTP 导入 HDFS 任务。

- 离线工作流以某业务场景下的按天处理测速日志为例。
- 数据流向为测速 agent—>FTP—>HDFS—>HIVESQL—>MySQL。
- 原始日志已采集并按天存入到 FTP。（集群内 FTP，也可以外部数据源，设置方法参见【服务器配置】）
- HIVESQL 做运营商维度的 PV 统计，并最终导出到 MySQL 结果数据库。
- FTP 存储位置：`/FTP/origin_stat/in/${YYYYMMDD}`。（注:这里变量表示按日期分目录）

数据接入

数据接入示例说明

右击新建的离线接入任务模块，然后选择【编辑】，根据详细说明设置离线接入的相关内容。



详细说明：

（1）任务名称：任务标识名称。

（2）任务类型：数据导入集群的方式。

（3）告警：即依据任务状态设置告警，目前支持任务失败或执行超时告警。

（4）周期类型：数据导入任务的执行周期。支持每月，每周，每天，每小时，每分钟或一次性非周期和持续性非周期方式。
（5）起始数据时间：以数据的更新时间作为任务执行的起始时间。例如现在是 8 月 21 日，但任务需要从昨天的（8 月 20 日产生的数据）数据开始计算，那么这里选择起始数据时间为 8 月 20 日。
（6）自身依赖：同一个任务不同周期产生的调度任务之间是否有依赖关系。例如离线导入任务按天执行，那每天都会生成一个数据导入任务，如果这里选择“是”，那只有前一天的导入任务成功完成，第二天的才会顺序执行。
（7）调度时间：与任务周期相关的参数，即任务下发的相对时间，例如这里选择 0 点 0 分，任务周期选为“天”时，则每天的 0 点 0 分下发任务执行，以此类推。
（8）责任人：工作流负责人，可修改删除工作流。
（9）任务说明：备注信息。
（10）基础设置-源服务器：【服务器设置】里增加的服务器配置，这里直接选择即可。介绍示例为 FTP 导入到 HDFS，故这里选择事先加入的 FTP 服务器即可。
（11）基础配置-目标服务器：【服务器配置】里增加的服务器配置，这里直接选择即可。介绍示例为 FTP 导入到 HDFS，故这里选择事先加入的 HDFS 服务器即可。
（12）源目录：由于这里的选择是 FTP 导入到 HDFS，故选择的是 FTP 的目录。这里支持时间内置变量 ${YYYYMMDDhhmm}，即年，月，日，小时，分钟。
（13）目标目录：由于这里选择的是 FTP 导入到 HDFS，故填的是 HDFS 的主目录。这里同样支持内置时间变量 ${YYYYMMDDhhmm} 。
（14）其他配置：其他配置里选项默认即可。

其他离线数据导入方式介绍

多种导入方式的基础配置基本类似，任务周期选择均相同。差异点即取源数据的方式不同，根据需要选择不同方式即可。
目前离线数据导入的方式支持如下图所示：

数据计算

数据计算示例说明

本示例以 HIVESQL 进行数据计算，首先新建一个 HIVE SQL脚本 任务。

右键单击【编辑】之后，根据详细说明填写相关参数信息。

详细说明：

（1）任务名称、类型、周期类型、起始数据时间等和数据导入类似，不再赘述。

（2）源服务器：即在【服务器设置】里增加的服务器配置，这里为 HIVESQL 计算，故选择之前配置的 HIVEserver 即可。

（3）SQL脚本：即该任务执行的 SQL 脚本，单击【上传脚本】，选择本任务执行的脚本即可。

示例脚本如下：

```
use demo_stat_db;
ALERT TABLE demo_stat_tb_day ADD PARTITION(date_id = '${YYYYMMDD}')
LOCATION 'hdfs://hdfsCluster/project/demo/in/${YYYYMMDD}/';

insert into table demo_stat_db.isp_conn_total_times
SELECT date_id,isp,sum(conn_succ)
FROM demo_stat_tb_day
WHERE 1.date_id = '${YYYYMMDD}'
GROUP BY isp,1.date_id;
```

简单解释：第一步使用外部表的方式使 HDFS 与 HIVE 建立关联；第二步即常用的SQL语法（详细使用可通过搜索引擎学习），进行简单的运算并存入到 HIVE 结果表。

其他数据计算方法说明

目前支持的计算方法包括 HIVESQL，Map-Reduce，Shell 脚本，Spark 和 JStorm，每种计算方法的基础配置如周期性等均完全一样。

差异在于 Spark 和 JStorm 上传脚本时，可以自定义参数，同时 JStorm 需要指定 Topology 类和名称，如下图所示：

数据导出

数据导出示例说明

本示例中，到目前为止，数据经过计算已经将结果存入到 HIVE 表，即 HDFS 的文件中，导出即选择从 HDFS 导出到 MySQL 永久存储。
首先新建一个 HDFS 导出 MYSQL 任务。

右键单击【编辑】，填写相关配置信息。

配置说明：

（1）任务名称，周期类型，起始数据时间，自身依赖，调度时间，责任人，任务说明均与数据导入，数据计算的含义一致。

（2）任务类型：本例使用 HDFS 导出到 MySQL，其他可使用的导出方式后面后续介绍。

（3）源服务器：即【服务器配置】里增加的服务器，这里示例中选择配置好的 HDFS 服务器即可。

（4）目标服务器：即【服务器配置】里增加的服务器，这里示例中选择配置好的 MySQL 服务器即可。

（5）源目录：即源服务器中 HIVESQL 表在 HDFS 上目录位置。

（6）目标表：即目标 MySQL 服务器的数据库中对应 table。

（7）目标表列名：目标 MySQL 服务数据库中 table 的表名称列表以,分隔，注意需要和 HIVESQL 表的列数量一一对应。

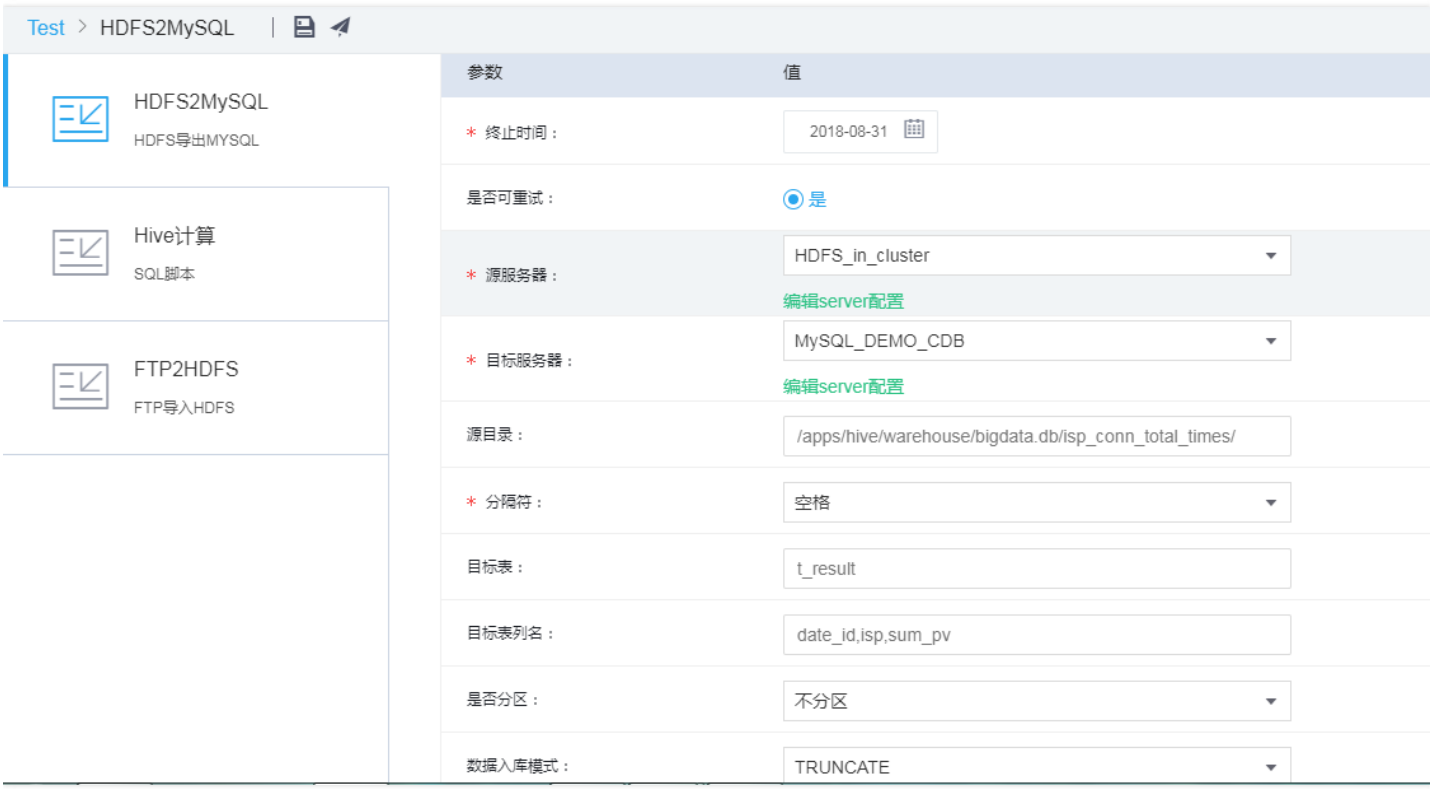
（8）分隔符：HIVESQL 表的字段之间的分隔符，这里要注意的是 HIVESQL 表在创建时指定好这里的分隔符。

（9）是否分区：即导出到 MySQL 表时，MySQL 表是否需要分区，目前支持按照时间分区，分区的格式支持 P*${YYYYMM}*，*P*${YYYYMMDD}和P_${YYYYMMDDHH}。

（10）数据库入库模式：目前支持两种入库模式：TRUNCATE 和 APPEND，其中 TRUNCATE 每次都会清理掉目标 MySQL 数据表全表，将 HIVESQL 全表重新导入；APPEND 方式会在 MySQL 表中增加一个 etl_timestamp 字段，填入数据导入时间戳，当导出任务重跑时会根据这个时间戳来进行覆盖。


Test > HDFS2MySQL
HDFS2MySQL
HDFS导出MYSQL
Hive计算
SQL脚本
FTP2HDFS
FTP导入HDFS
基本信息
调度设置
参数配置
* 任务名称：
HDFS2MySQL
* 任务类型：
HDFS导出MYSQL
* 负责人：
leozwang ×
任务将用第一个责任人相关权限来执行任务，画布责任人也具有任务修改相关权限。
任务说明：
告警：


Test > HDFS2MySQL
HDFS2MySQL
HDFS导出MYSQL
Hive计算
SQL脚本
FTP2HDFS
FTP导入HDFS
基本信息
调度设置
参数配置
* 周期类型：
天
* 起始数据时间：
2017-07-01
周期类型和起始数据时间一旦选定保存后将不允许再修改。
任务获取数据的时间起点，任务将于下一个周期（执行时间）调度运行。例如：数据时间为3点的小时任务，将会在下一个周期4点运行。
* 自身依赖：
是
否
并行
调度时间：
0点
0分
任务到了执行时间后，会按照这里的配置延迟一定时间才会开始执行。



其他数据导出方法说明

目前支持的数据导出方法如下图所示：

任务基本信息的设置均一致，差异如下：

数据导出到 HBase，列的对应信息填写准确即可。（其余均为导出到类 SQL 数据库）

实时工作流的使用

最近更新时间：2017-12-07 16:53:45

示例任务介绍

本示例以一个实时任务流介绍实时任务处理的操作方法。
数据流向：kafka—>JStorm—>MySQL。
kafka 作为消息缓存队列，JStorm 做逻辑处理，MySQL 做数据永久存储。
按照【快速入门->创建工作流】的做法，新建一个工作流。

数据导入与计算

1. 这里默认消息已经导入到 kafka 里，故基于实时消息接入后，数据计算直接使用 JStorm 落地，存入到 MySQL。
   JStorm 代码需要打包为 jar 包的形式，示例代码如下：

```
public class StormKafkaMysql{
public static void main(String []args) throws Exception{
//配置Zookeeper地址
//BrokerHosts brokerHosts = new ZkHosts("a:2181,b:2181,c:2181,d:2181,e:2181");
BrokerHosts brokerHosts = new ZkHosts("192.168.100.25:2181,192.168.100.64:2181,192.168.100.97:2181");
//配置Kafka订阅的的Topic,以及zookeeper中数据节点目录和名字
SpoutConfig spoutConfig = new SpoutConfig(brokerHosts,"cd123","/zkkafkaspout-tim","kafkaspout");

//配置KafkaBolt中的kafka.broker.properties
Config conf = new Config();
HashMap<String,String> map = new HashMap<String,String>();
//配置Kafka,broker地址
//map.put("metadata.broker.list","cloud203:9092,cloud204:9092,cloud206:9092,cloud207:9092,cloud208:9092")
map.put("metadata.broker.list","192.168.100.117:6667,192.168.100.131:6667,192.168.100.159:6667,192.168.100.166:6667")
//serializer.class为消息的序列化类
map.put("serializer.class","kafka.serializer.StringEncoder");
conf.put("kafka.broker.properties",map);

spoutConfig.scheme = new SchemeAsMultiScheme(new MessageScheme());
TopologyBuilder builder = new TopologyBuilder();
builder.setSpout("spout",new KafkaSpout(spoutConfig));
builder.setBolt("MysqlBolt",new MysqlBolt()).shuffleGrouping("spout");
conf.setNumberWorker(3);
StormSubmitter.submitTopology(StormKafkaMysql.class.getSimpleName()+"-timtest",conf,builder.creatTopology());
}
}
```

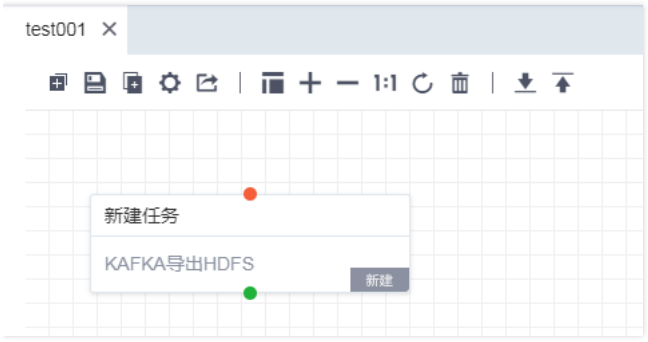
2. 在工作流画布新建一个 JStorm 任务，右击【编辑】，详细配置如下：

填入 JStorm 的 Topology 名称，Topology 类以及参数，程序包从本地上传脚本即可。（Storm相关编程知识可通过搜索引擎查询）

数据导出

数据导出示例介绍

本示例中直接用 JStorm 落地存储，故没有使用到工作流的实时数据导出。如下为实时数据导出的示例，在工作流画布新建一个 KAFKA 导出 HDFS 任务。



右击【编辑】，填写详细配置信息。

配置说明：

（1）任务名称，责任人，任务说明和所有工作流任务一致。

（2）任务类型：目前支持 tube（kafka）导出到 HDFS 和 HBase，本例以导出到 HDFS 为例。

（3）周期类型：实时任务只可选择一次性非周期。

（4）消息中间件 topic：即 tube（kafka）的 topic 名称。

（5）消息中间件group：即 tube（kafka）的消费组名称。

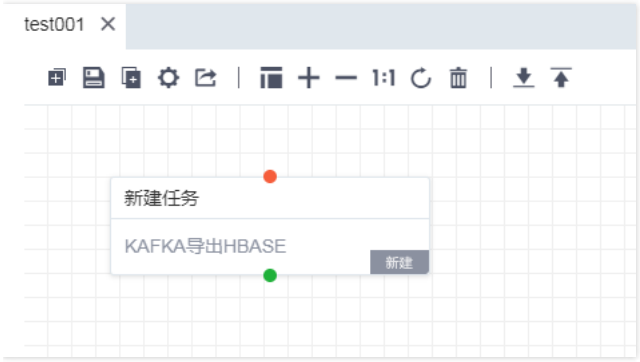
（6）HDFS 目标目录：导出到 HDFS 的目录。

（7）HDFS 地址：即 HDFS 的 activenamenode 地址（本套件已做了 HA，地址为 hdfsCluster）。

（8）文件最大落地大小：即消息中间件的消息积累的消息量才落地的 HDFS 的最大值。

其他实时数据导出方式

目前支持的其他实时导出方式为导出到 HBase，首先新建一个 KAFKA 导出 HBASE 任务。



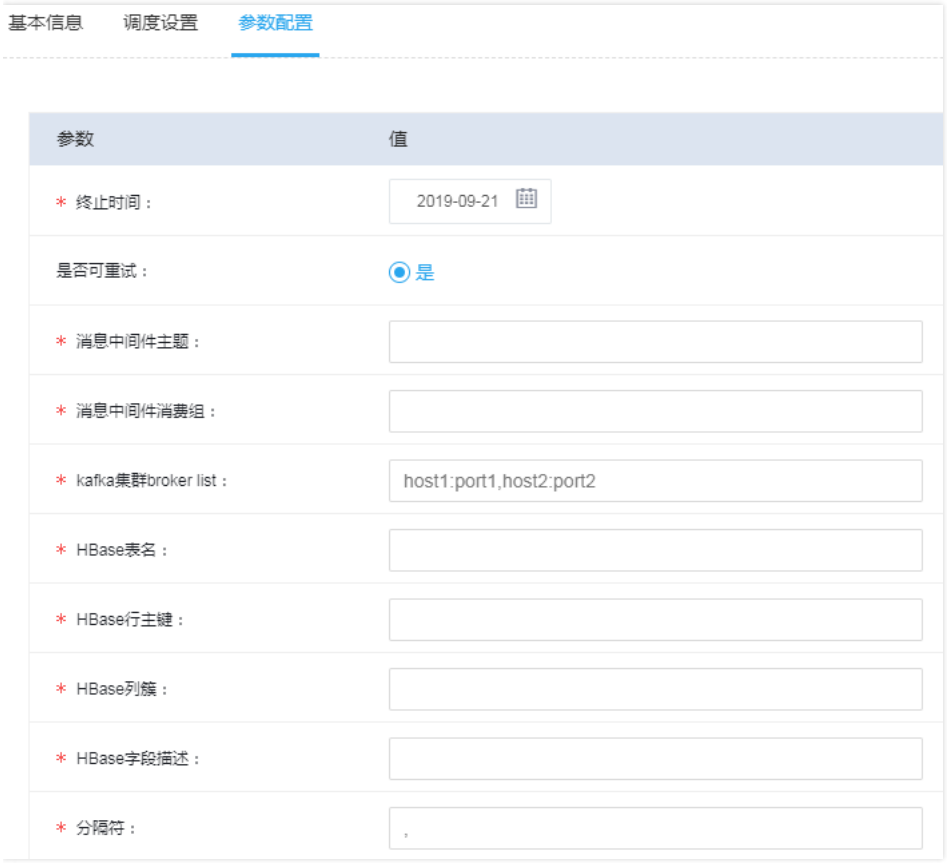
右击选择【编辑】，填写详细配置信息。

配置说明：

（1）消息中间 topic，消息中间件 group 和导出到 HDFS 类似。

（2）HBase 表名，行主键，列簇，字段描述，分隔符即 HBase 表相关信息。

（3）Zookeeper 根节点：即 HBase 在上 zookeeper 上注册的 znode 名称。



数据展现
数据展现概述

最近更新时间：2019-03-21 19:23:50

数据展现模块是运营数据分析结果的展示环节。本模块提供数据可视化自助报表——黄金眼报表组件，一款可视化运营报表门户制作工具。
本组件主要帮助用户解决“去开发化”，为业务决策者等数据使用者，提供轻量、自助化、可视化的报表分析服务。本组件有以下优点：

- 无需 coding 基础，可视化数据源配置（数据指标/维度指标）。
- 提供2种报表类型，覆盖日常报表需求。
- 标准报表页面模板，即编即用。
- 报表权限灵活分配。

相关术语：

- 数据源：报表展示的数据来源，一般指数据库表。
- 维度：报表展示分析的角度。报表展示可指定不同值的对象的描述性属性或特征。
  例如，网页浏览统计的浏览器类型维度，该维度包含 IE、Chrome、Firefox 等浏览器，地理分析中的城市维度，该维度包含北京、上海、广州等地点。用户可以使用维度来整理、细分和分析数据，也可以通过添加和删除维度来查看数据的不同统计特征。
- 指标：报表展示的聚合统计对象。指标通常是指数据字段，可以按总数或比值衡量的具体维度元素。例如，维度“城市”可以关联指标“人口”，其值为具体城市的居民总数。

配置报表方式，大致需要以下三个步骤：

1. 填写报表相关信息。
2. 选择对应的数据源和页面设备。
3. 预览发布，即可完成一个数据报表门户。根据报表类型的不同，数据源配置、页面设计过程中会有细微差别。

自助报表提供如下基础版和高级版两种，在报表配置页面提供参考两种 demo 可供用户学习和体验。

类型	基础版	高级版
适用对象	运营开发	产品经理、运营开发等
定位	快速创建在线运营报表门户，支持多维对比筛选	标准数据报表门户，拥有完整网站信息导航，能自定义页面元素

自助报表基础版创建指引

最近更新时间：2018-06-12 10:17:04

自助报表基础版创建适用于运营开发，能够快速创建在线运营报表门户；支持多维对比筛选。下面通过一个示例介绍自助报表基础版创建流程。

首先登录腾讯大数据处理套件，单击【数据展现】模块。

单击【新建产品运营门户】。

编辑门户基本信息，单击【创建】。

创建成功后，单击【设计报表页面】，进行报表设计。

基础版报表门户采用直观的树形目录结构，建议按业务内容分组创建文件夹，方便批量管理页面。支持创建无限层级文件夹/页面，满足业务快速成长需要。

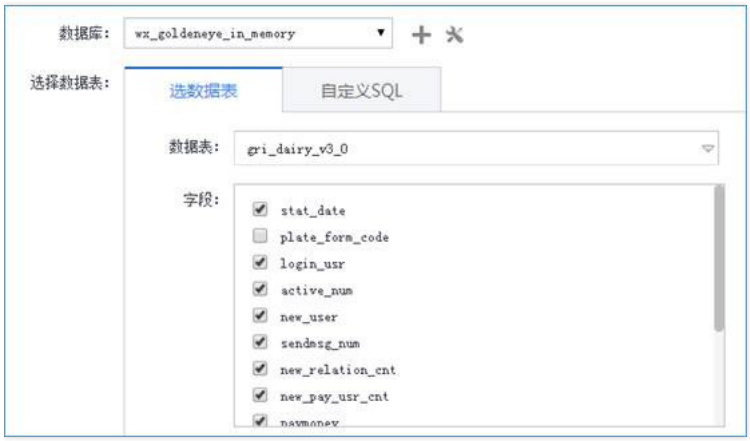
选择数据来源：

- 选择数据库：每张报表页面均需单独配置数据源。目前支持 Postgre、MySQL、Oracel 三种数据驱动。

- 选择字段数据库：支持可视化配置、自定义 SQL 两种字段选择方式。

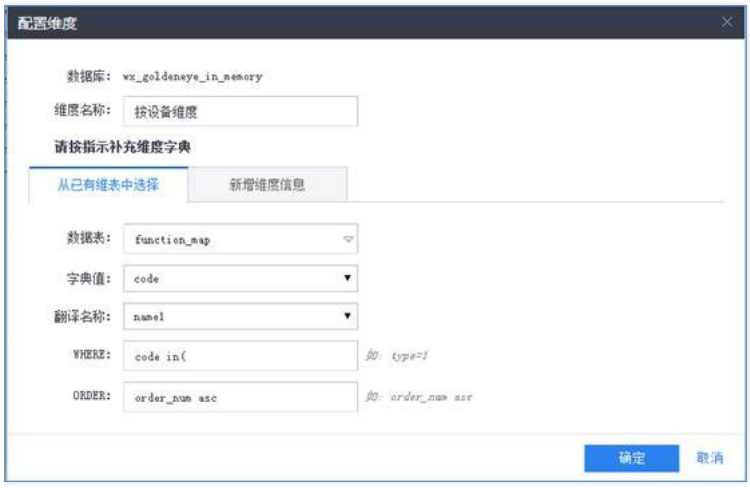
目前可视化配置只能选择在一张表中选择字段。对于 Postgre 存在 schema 需要在选择数据库后在数据表中使用 schema.table 方式。

对于 MySQL 可以直接选择库表字段。



自定义 SQL ，输入语句后，单击【解析SQL】生效。

设置指标展示信息，支持3种指标类型：时间维度、数据指标、维度指标。

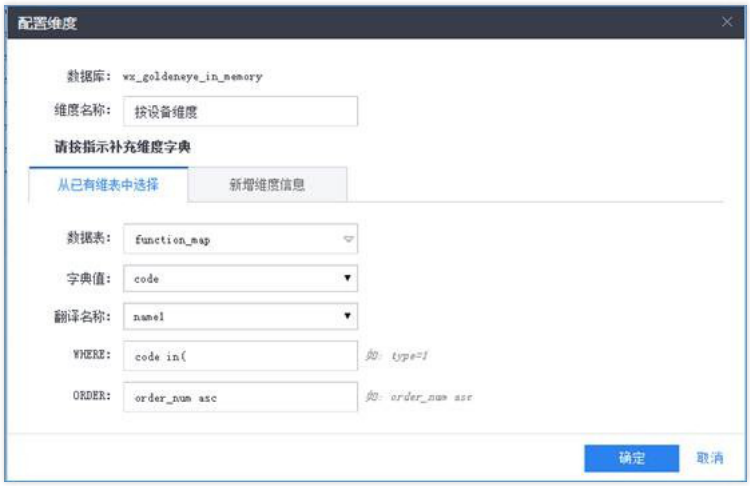


- 时间维度。目前支持日、周、月 3 种周期，请注意匹配数据库中的时间格式。

- 数据指标。支持整数、小数、百分比、字符型；高级配置中，可以更加细致的定义数据格式，方便前台浏览。

- 维度指标。

单击【配置维度】，补充维度值信息：



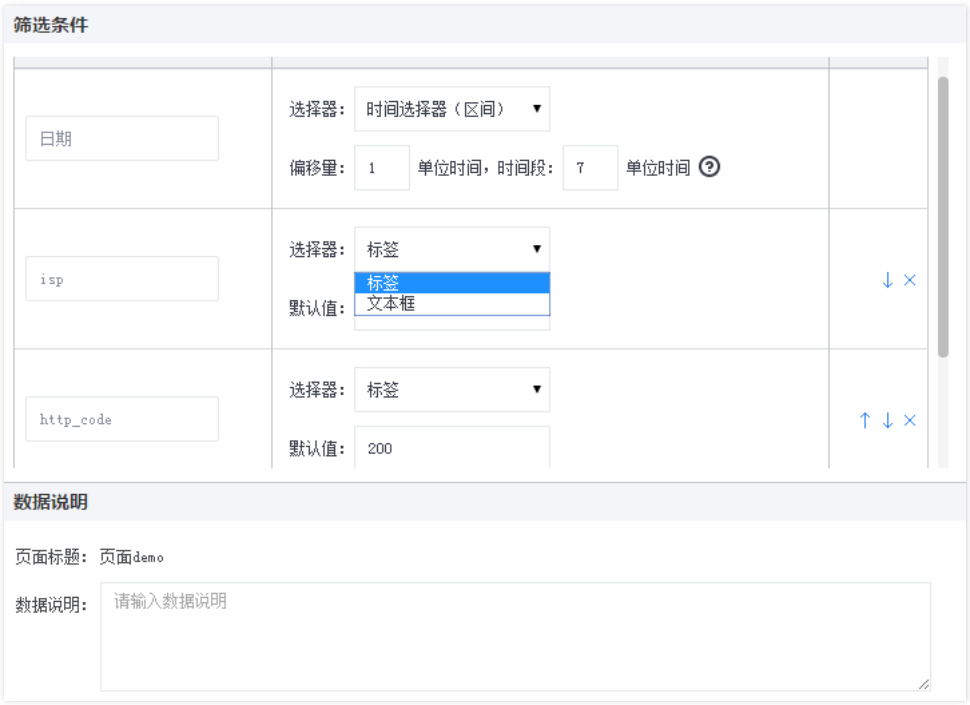
添加后，可统一在【数据源管理】-【维表管理】中修改。

自定义指标，除了原始指标，目前支持自定义比率类指标：

编辑页面元素。在完成数据源选择和展示的纬度指标后单击下一步，进行页面元素编辑。基础版页面由 3 部分组成，分别是筛选条件区、统计数据图区、数据表格区。左侧为

报表页面名称，可以新增和删除页面，修改页面的名称。

- 筛选条件区，来源于已选中的维度指标信息，已默认关联表单选择控件。单击【↑】、【↓】调整维度排序，可以配置页面的标题。在数据说明区可以为数据关键点补充说明，解释数据异常信息。



- 统计数据图区，支持折线、柱状、饼图控件，如选定条件有内容，能看到图表预览效果。
- 数据表格区，展示选择条件和纬度的数据。

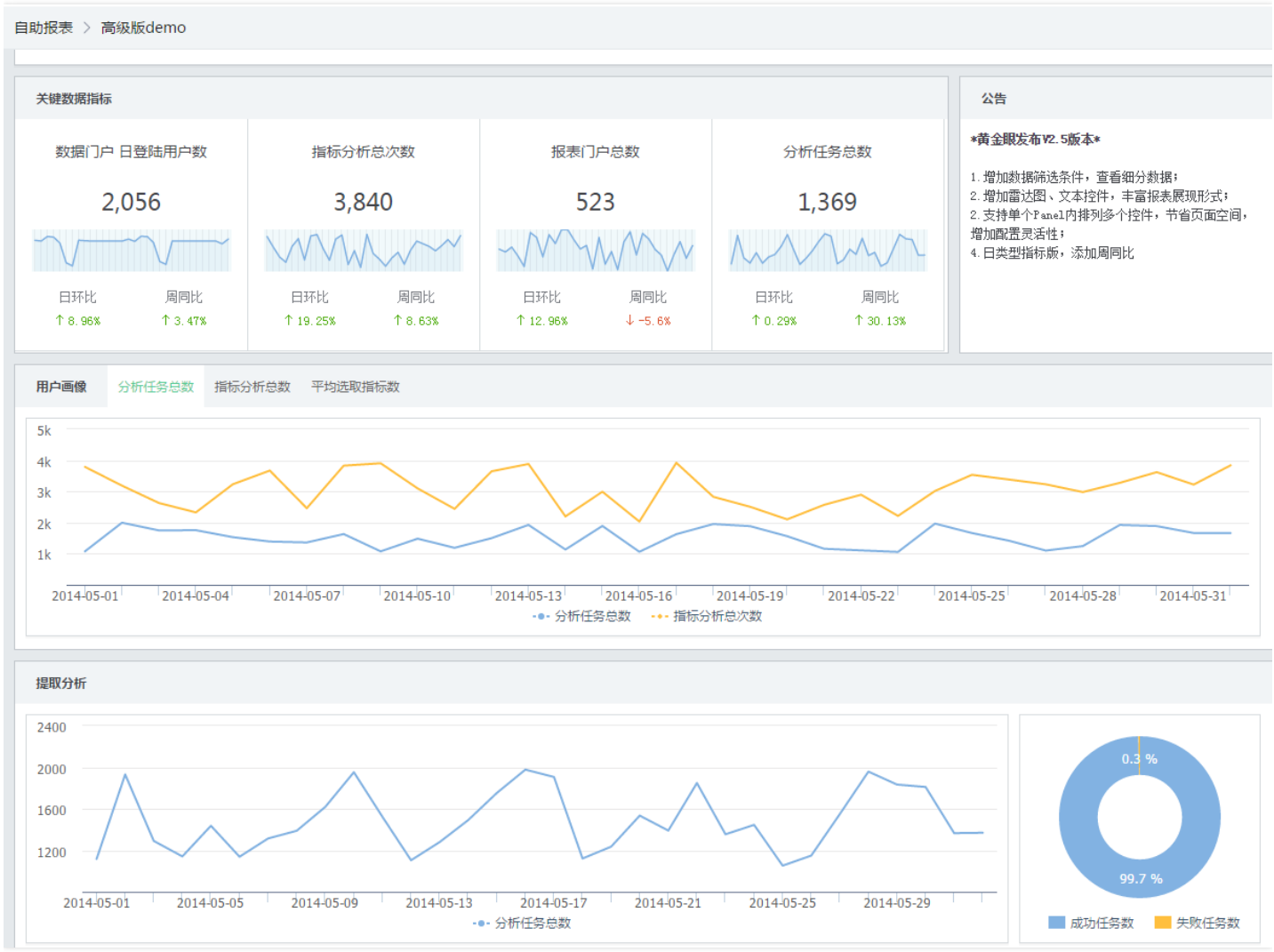
至此基础版报表就建立完成，单击【预览】即可看到创建的报表。

自助报表高级版创建指引

最近更新时间：2018-06-12 10:21:45

自助报表高级版通常用来沉淀产品运营指标，观察产品长趋势。您可以从概览页到详情页，立体、多维地展现产品运营状况。新建高级版运营报表，需要以下几个步骤：

套件自带的高级版 demo 如下：



新建报表填写基本信息

首先登录 腾讯大数据处理套件，单击【数据展现】模块。

单击【新建产品运营门户】。

在新建页填写门户基本信息。

- 门户名称、LOGO用于前台展示，能体现产品业务特征。
- 英文简写作为默认门户地址标识符，不能重复。
- 选择"高级版"（高级版适用对象是产品经理、运营开发等）。

建立运营指标

提交基本信息后，开始建立运营指标。本部分需要运营开发预先接入数据源，将指标信息配置完整，为页面设计准备好素材。

新建数据库

在【数据库管理】标签页中单击【新建数据库】，以预先接入指标所在的数据库信息。在弹出窗口中填写相应内容以配置数据库，数据源支持 Postgre、MySQL、Oracle。

新建指标

在【指标管理】页面中单击【新建指标】，可从已有数据库中选择数据表，亦可自定义 SQL。这一步让您从分散的数据源中，聚合产品所有数据指标，形成指标体系，实现指标的复用。

维表管理

在【维表管理】标签页中单击【新增维表】，指定前台显示的维度指标与数据库中字符的对应关系。维表可以关联到报表中，也支持在线补充维度值。

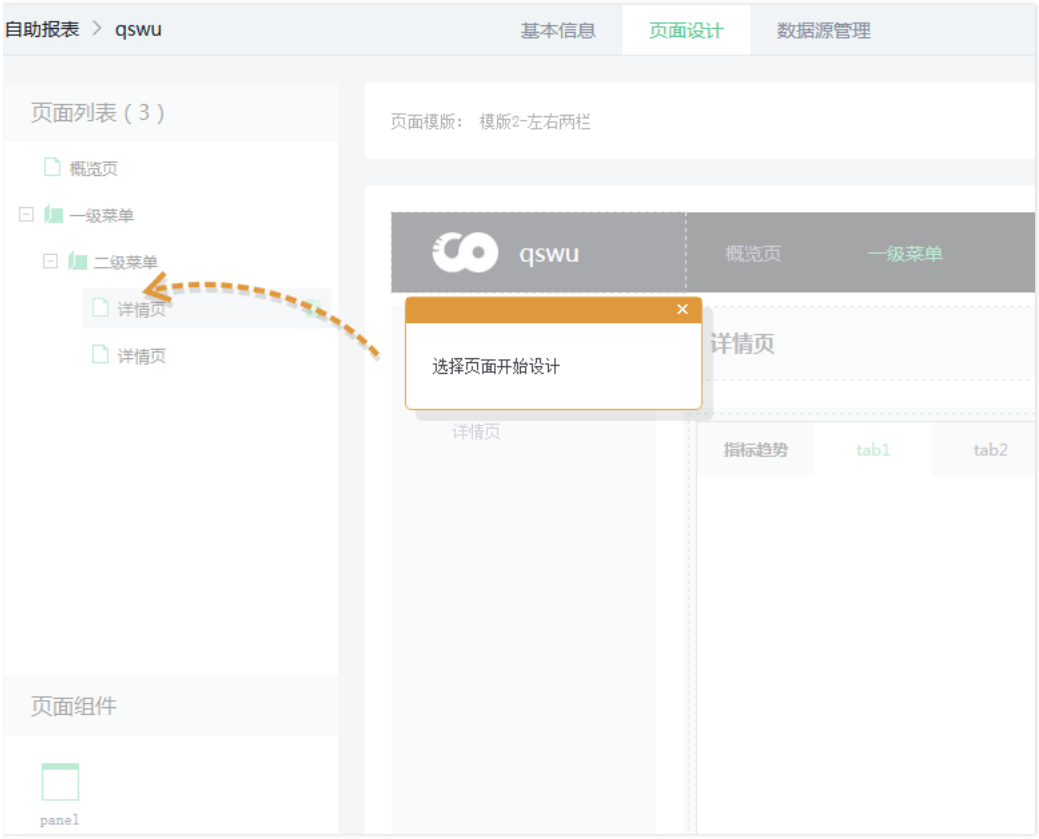
设计门户页面

搭建好数据指标，相当于备好了盖房的原材料。接下来，就可以设计门户页面了，相当于把这些原材料进行加工，并搭建房子。

注：报表页面模板宽度固定在 1200 px，建议使用宽度大于 1920 px 的显示器，在 Chorme、Firefox、IE10 等高版本浏览器下设计，以获得最佳设计体验。

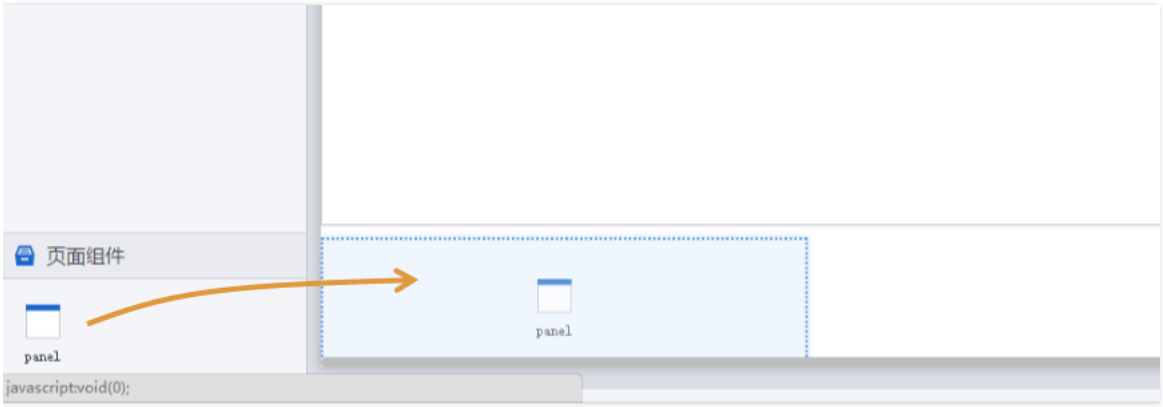
选择页面开始设计

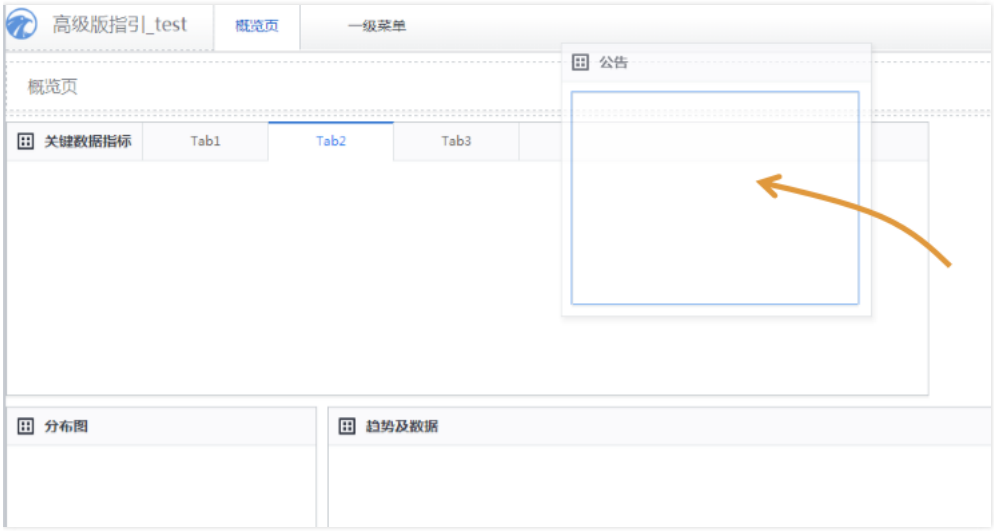
您可以在左侧页面列表区域按需要新建页面或文件夹，支持无限层级菜单。页面列表结构会同步映射为报表网站导航；页面区分发布、未发布状态，每次调整后需发布方可生效。



新建 panel

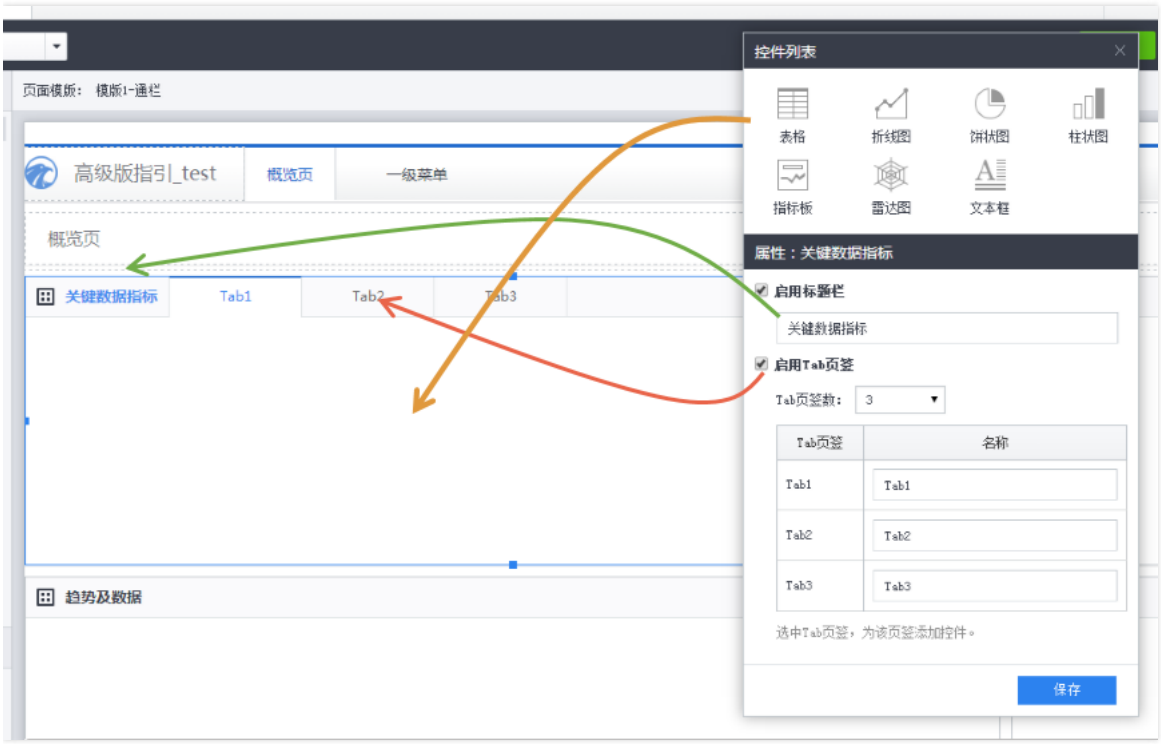
将 panel 图标拖入右侧页面设计区域下方空白处，即可新建 panel。Panel 可自由拖拽，自行布局，满足个性化需求。





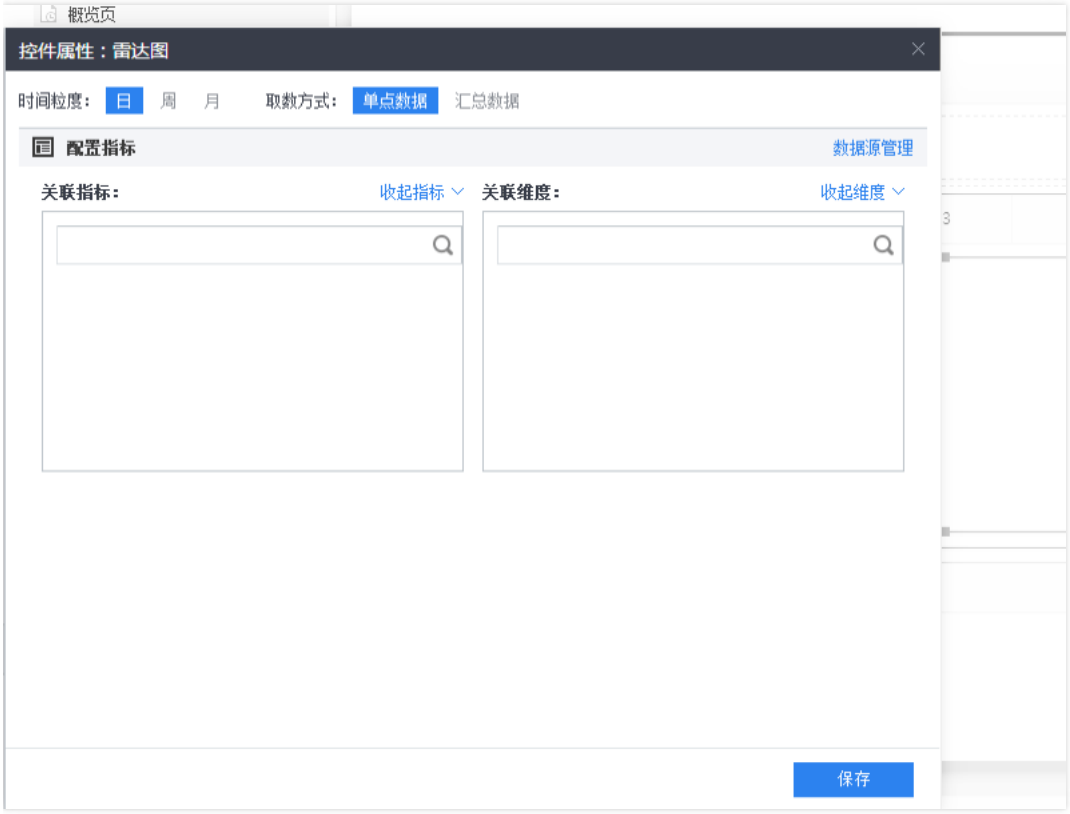
添加图表控件并绑定数据内容

单击 panel 区，即可在弹框内修改 panel 标题，设置 tab 标签（可在统一页面内自由切换）；图表包含了文本控件、表格、曲线图、柱状图、饼图和雷达图等，您只需将图表控件拖入 panel 区，即可实现丰富的报表展现形式。



接着，单击 panel 区，即可配置控件属性：可以选择时间维度、取数方式，并对指标进行配置，数据效果即时生成。

“关联指标”是指该图表所关联的数据指标，例如日登录用户数、活动参与人数等；“关联维度”是指该图表所关联的维度指标，例如平台类型、用户性别等。



完成新建

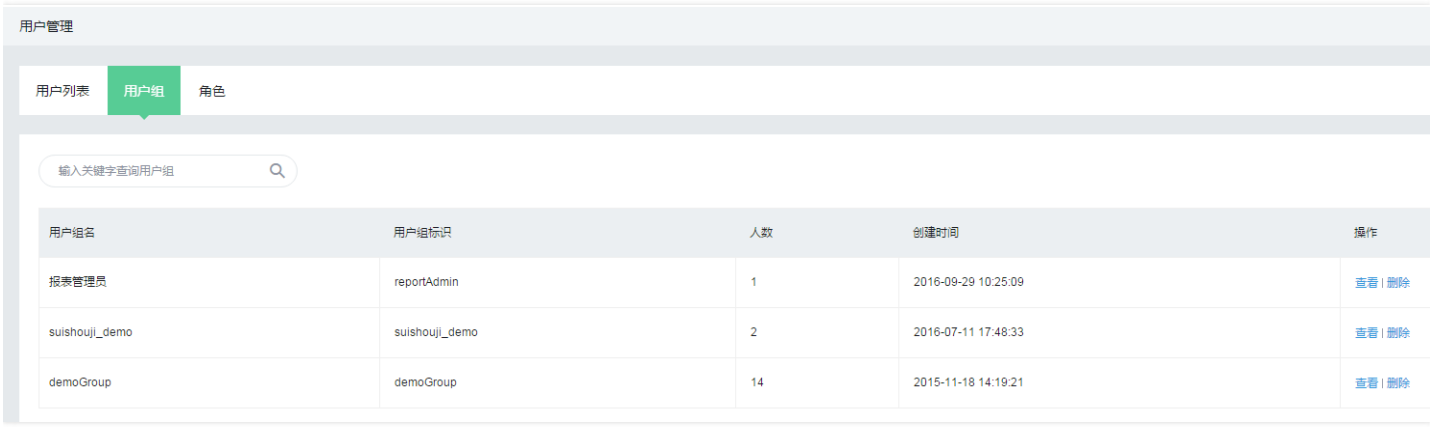
此时，新建运营报表的基本步骤已经完成。单击【预览】，可以实时查看；单击【发布】，即可发布报表。

建议在页面设计过程中，随时预览效果。

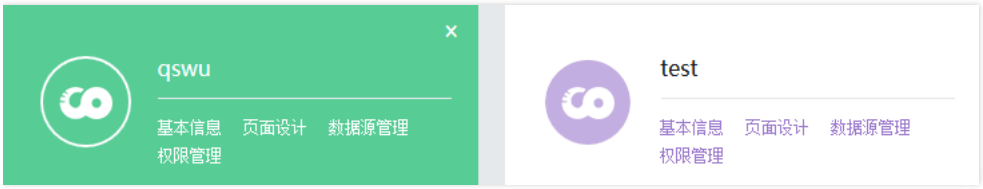
报表权限管理

最近更新时间：2017-12-07 16:54:37

单击对应报表的【权限管理】，可以配置当前报表的读取和编辑权限。此处支持按用户组授权，报表默认权限为报表创建对应的用户所有，为方便管理报表权限，可以在用户管理中增加用户组。



单击对应报表的【权限管理】，选择用户组添加或删除到读取和编辑列表中即可。



数据分析

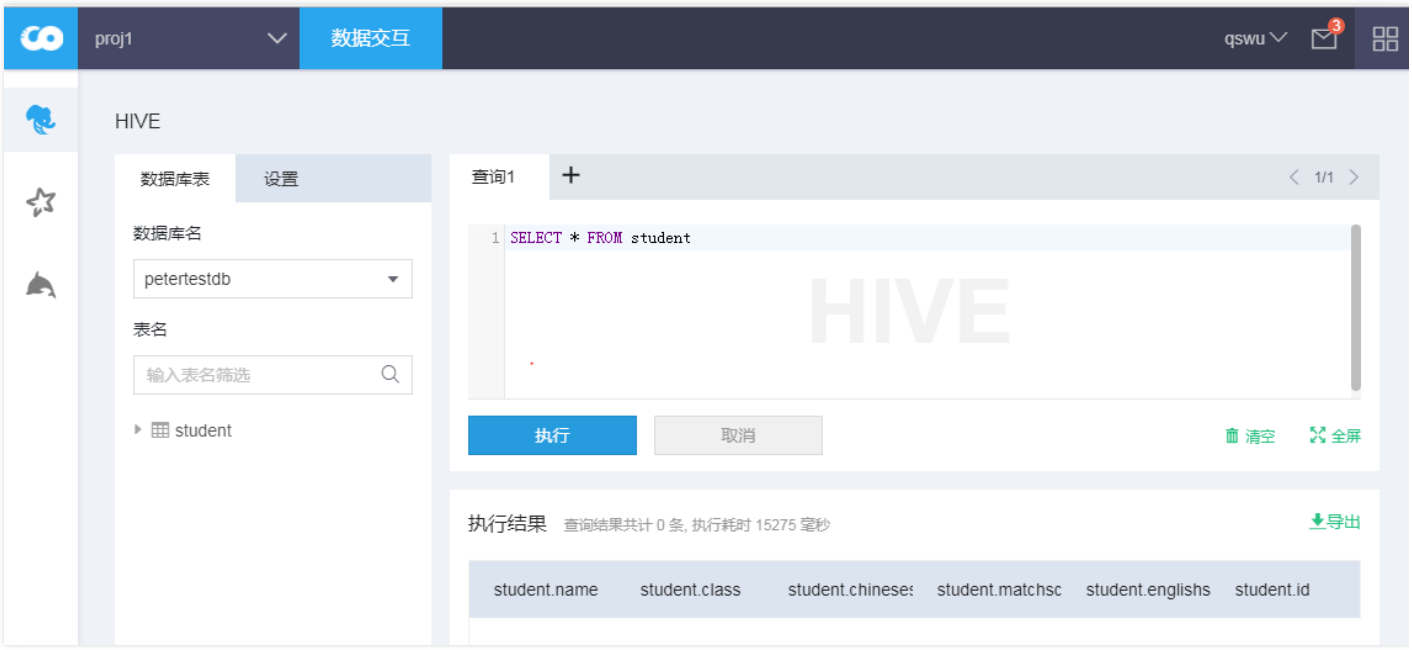
最近更新时间：2018-05-25 18:28:25

数据交互即通过 WebUI 的方式对数据库表进行查询，更新等操作的功能。本文档将以 Hive 交互为例子，介绍数据交互功能。

1. 登录 腾讯大数据套件 ，在主界面单击【数据分析】，进入数据交互模块的界面。

2. 具体操作步骤：
   （1）在数据库表快捷选择窗口选择要查询操作的库表（也可以在编辑窗口使用 use 命令指定）。
   （2）在编辑窗口输入查询命令。
   （3）单击【运行】按钮，执行命令。

（4）导出查询结果。



3. Spark 交互和 HBase 交互

Spark 交互和 HBase 交互的操作与 Hive 交互的操作相似，差异是查询命令有所不同。

4. 其他

（1）Spark 的查询目前支持 Scala,PyShark（python），R语言和 shark（sparkSQL）。

（2）Spark 方式查询时实际会向 yarn 申请资源，请注意资源的使用情况。

（3）SuperSQL 为腾讯自研的以 SQL 执行 HBase 表的接口。

数据资产
数据资产概述

最近更新时间：2017-12-07 16:55:20

概述和组件构成

腾讯大数据平台提供细而全的数据权限验证和授权，保障企业数据资产安全，支持血缘、直系和重要性分析。用户能够直观地把握数据资产状况，清晰了解全局信息，同时可以多渠道全方位进行统一维护。数据资产主要提供了平台的库表管理、数据血缘管理和数据提取功能。

相关术语

- **可管理库表**：数据库表的责任人具有对库表的可管理权限。
- **可读写库表**：数据库表所属项目的成员对库表具有可读写权限，其它项目成员可通过申请库表权限功能开通可读写权限。
- **无归属库表**：系统扫描到的数据库表（一般是用户在登录机器命令行创建），无责任人和项目归属关系。
- **数据血缘**：数据产生的链路或者路径，比如通过数据 A 数据 B 产生了数据 C，那么 C 的父血缘就是 A 和 B，反之亦然。在大数据套件中描述数据“父子”关系，以思维导图形式展现了数据变化影响和数据生产溯源，清晰刻画表与表之间、任务与任务之间的关系。
- **直系关系**：父级到子级数据表的任务处理流程关系。
- **血缘回溯**：分析原始数据到目标数据表的计算路径。
- **数据字典**：对数据的数据项、数据结构和数据存储等进行定义和描述，其目的是对数据流程图中的各个元素做出详细的说明，使用数据字典为简单的建模项目。简而言之，数据字典是描述数据的信息集合，是对系统中使用的所有数据元素的定义的集合。

数据库管理

最近更新时间：2018-06-12 10:23:32

创建数据库

进入【库表管理】界面

单击【库表管理】界面，即可进行库表管理。库表管理是以用户视角，而不是以项目视角进行管理，左侧菜单标签页面是可管理库表、可读写表、未归属库表（只有管理员权限用户才可以看到）、Hbase 库表，右侧主要是创建新库表和申请库表权限。

新建库

在【库表管理】界面中单击【新建库】，将会进入到数据库的创建页面。依次填写数据库信息后，单击【确定】按钮即可。

- **所属项目**（必填）：该数据库属于哪一个项目，只能填写创建数据库的用户所在的项目；
- **类型**（必填）：目前支持 hive 和 hbase 类型；
- **库名/名称**（必填）：类型选择 hive 时，填写 hive 数据库名称，库名必须全局唯一；选择 hbase 类型时，填写 hbase 命名空间(namespace)名称
- **责任人**（必填）：该数据库或者命名空间的责任人，该字段会与【所属项目】联动，责任人只能是【所属项目】中的项目成员，可以填多个；
- **库描述**（非必填）：数据库的描述信息。

查看数据库信息

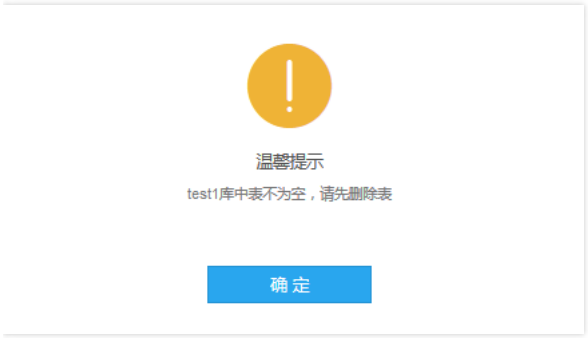
新创建的数据库可以在【可管理库表】的【数据库】标签页中找到，新创建数据库只有责任人才能看到。

单击【库名】可查看该数据库的基本信息和所包含的表信息。

删除数据库

只有对数据库具有可管理权限才能删除数据库，在【可管理库表】的【数据库】标签页中，找到对应数据库，单击【删除】即可。

删除数据库前需清空数据库下所有表，否则会删除失败。您需要单击到数据库名找到对应的数据表删除后再操作，未清空表时有下图所示：



数据表管理

最近更新时间：2018-06-12 10:25:17

创建数据表

进入【库表管理】界面

单击【库表管理】界面，即可进行库表管理，库表管理是以用户视角，而不是以项目视角。

新建表

在【库表管理】界面中单击【新建表】，将会进入到库表的创建页面。表信息包括三部分，【填写表信息】，【高级设置】和【字段与分区】，填写完成后单击【提交】即完成创建数据表。需要注意表类型需要区分 hive 和 hbase，hbase 表创建时不提供高级设置功能，如需要对表列簇有属性设置，可以参考本文档后续 hbase 表管理章节进行建

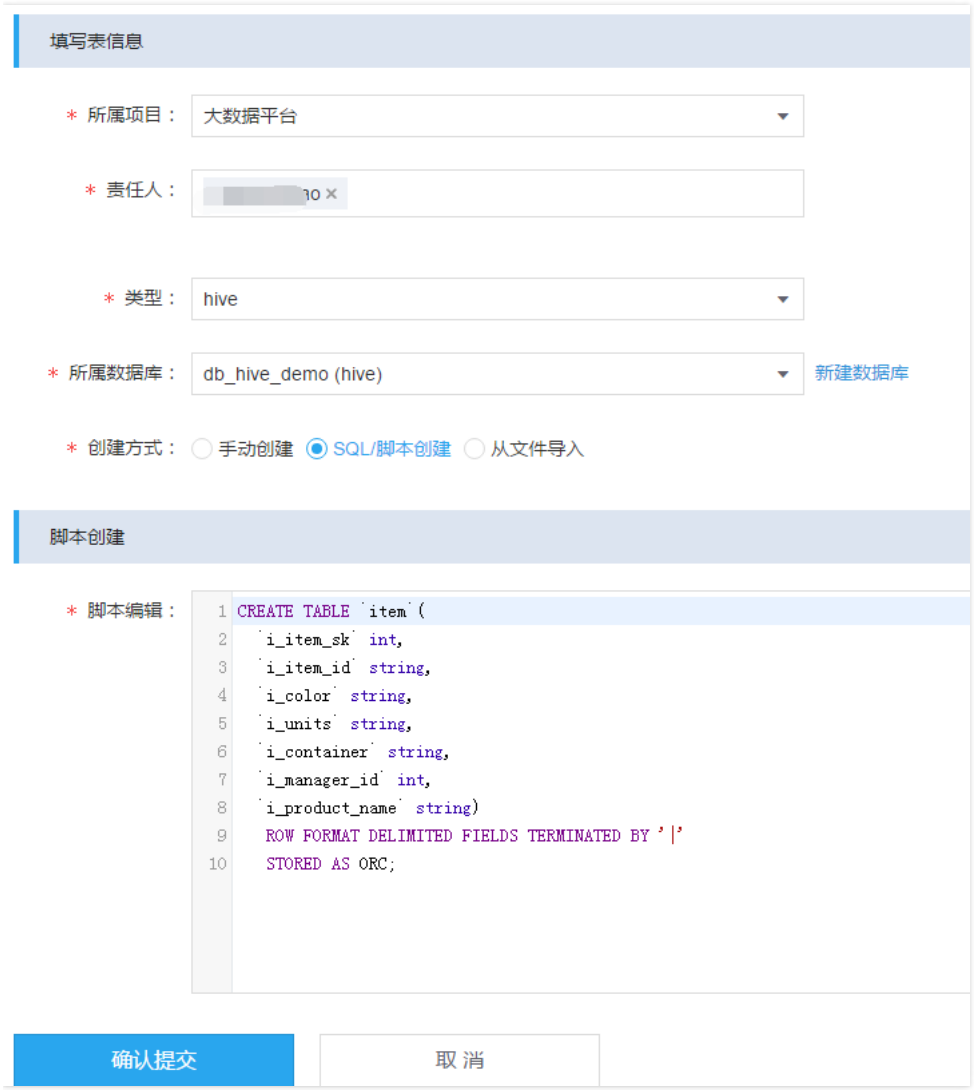
表。

填写表信息

- **所属项目**（必填）：该表属于哪一个项目，只能选择创建数据表的用户所在的项目；
- **责任人**（必填）：该表的责任人，该字段会与【所属项目】联动，责任人只能是【所属项目】中的项目成员，可以填多个；
- **类型**（必填）：支持 hive 类型和 hbase 类型；
- **所属数据库/所属命名空间**（必填）：表所属的数据库，该字段会与【所属项目】联动，由于数据库在创建的时候是有填所属项目的，因此这里只能选取到【所属项目】下的数据库；类型选择 hive 时，选择 hive 数据库名称；选择 hbase 类型时，则选择 hbase 命名空间 (namespace) 名称；
- **创建方式**：默认是为手动创建表，在字段和分区中填写对应的表字段或者列簇名称即可，同时创建 hive 类型的表也提供 SQL 语句创建和文本文件导入创建表，hbase 不支持这两种方式；
- **表名**（必填）：表名称，数据库内唯一；

- **表描述**（非必填）：表的描述信息。

使用SQL建表：
创建方式选择 SQL/脚本创建方式，这里仅支持 hive SQL 建表语句，如下图所示：



使用文件导入建表：
创建方式选择从文件导入创建方式，下载示例模版文件修改字段名称，上传到系统目录中即可选取字段进行建表。

填写表名和表描述后，在【字段与分区】中，单击【选取】，找到上传的建表文件。该文件需要提前在【基础文件管理】上传。

单击【基础文件管理】，跳转到新页面选择当前用户有权限操作的 hdfs 路径（用户根目录或者项目目录）上传建表文件。

在库表管理文件导入建表选取建表文件后，系统会自动选取第一行作为字段名称，同时还可以选择增加或者删除字段名称，单击【确认提交】即可创建完成。

高级设置

在建表方式选择手动创建后，单击【高级设置】，会展开【高级设置】的信息框，主要是 hive 需要的一些高级建表信息，一般情况下这些信息都可以采用默认值，可不填写。

- 记录格式：已分格（数据文件使用分隔符，如逗号 (CSV) 或制表符。），SerDe（自定义序列化反序列化的字段分隔方式）；

- SerDe 名称：记录格式选择 SerDe 时的选项，SerDe 的 Java 类名称；
- Serde properties：要传递给（反）序列化机制的属性；
- 字段终止符：记录格式选择字段间的分隔符；
- 集合终止符：数组类型字段的分隔符；
- Map 键终止符：map 类型字段的分隔符；
- 文件格式：TextFile（文本文件），SequenceFile（hadoop 二进制文件），InputFormat（自定义输入输出格式）；
- inputFormat 类：文件格式选择 InputFormat 时选项，用于读取数据的 Java 类；
- outputFormat 类：文件格式选择 InputFormat 时选项，用于写入数据的 Java 类。

设置记录格式，设置按照专用序列号分割和设置文件格式 InputFormat。

上述高级特性配置实际是 web 化配置 hive 建表的属性，不选择高级特性时默认存储为 TEXTFILE 类型，分隔符为上述选择中默认的分隔符。高级配置旨在为用户提供表特定属性，需要用户对 hive 表有一定了解基础上进行选择配置，hive 建表可以参考 官网说明。

字段与分区

填写表的字段信息，单击【删除】可删除该字段信息，单击【新增一行】则新增一个字段。单击【确认提交】后即可创建表。
hive 类型表：

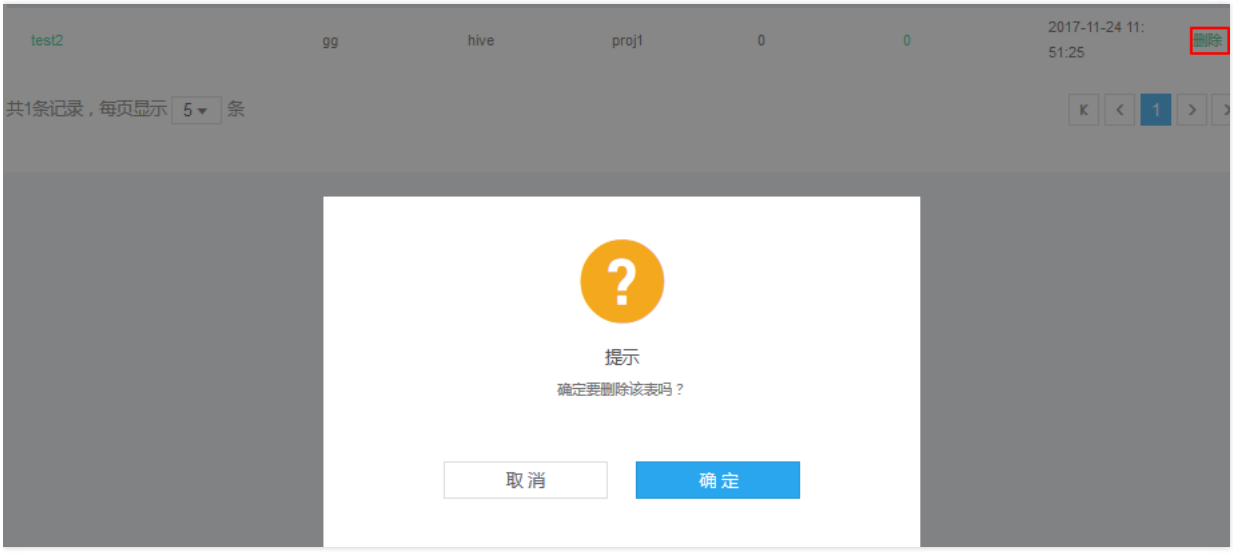
hbase 类型表：

查看数据表信息

新创建的数据表在【可管理库表】的【数据表】中可以查看到，单击【表名】可以查看详细信息，如下图：

删除数据表

只有具有对表的可管理权限才能删除表，在【可管理库表】的【数据表】标签页中，单击【删除】即可删除对应数据表，如下图：



库表权限申请

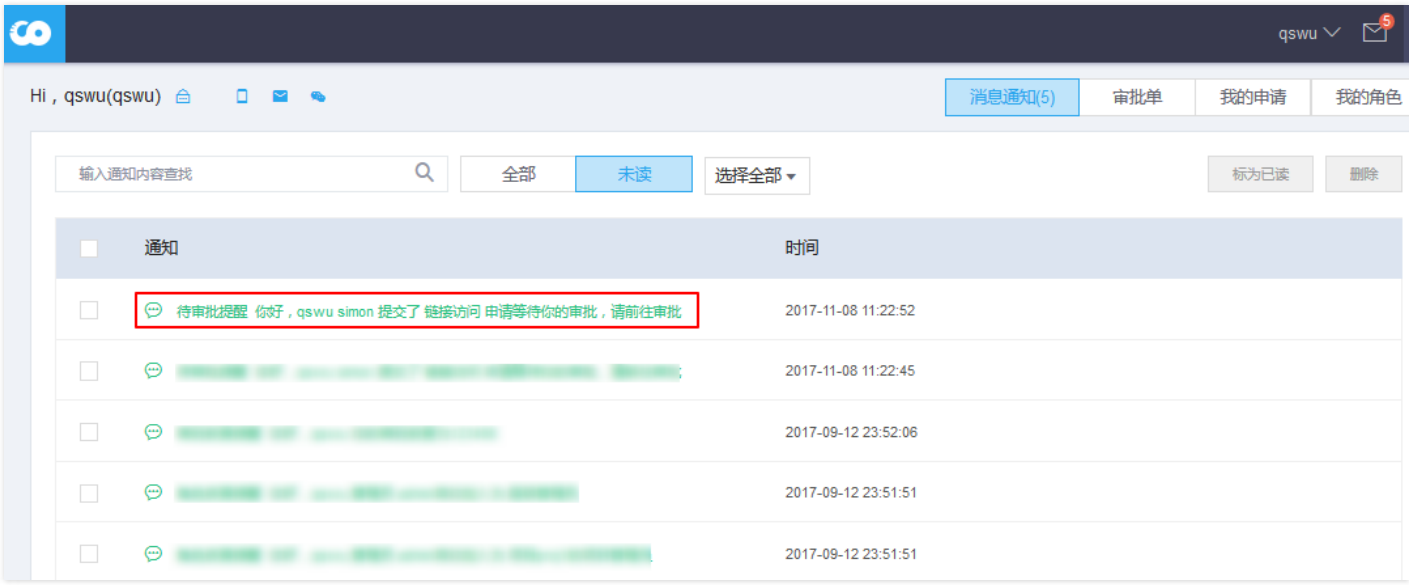
最近更新时间：2018-06-12 10:27:01

对于某个用户来说，与自己无关的项目下的库表，默认是没有权限访问的。如果需要访问，可以通过申请库表权限来实现。

1. 在【库表管理】界面，单击【申请库表权限】。

2. 填写申请信息，单击【提交申请】，申请提交后需要库表管理人员进行审批。
3. 申请库名（必填）：数据库名称，可申请 tbds 中所有数据库的权限；
4. 申请表名（必填）：可申请库名对应数据库下所有数据表；
5. 权限关联项目（必填）：申请的权限属于哪一个项目，只能填写申请用户所在的项目；
6. 申请权限（必填）：申请的权限。

1. 审批申请信息。用户在提出了对库表的申请之后，对该库表具有管理权限的用户将会收到申请信息，然后开始进行审批。
2. 进入个人中心消息通知。单击页面右上角用户旁的邮件标识，进入到【个人中心】，在【消息通知】标签页中有需要用户关心的系统消息，其中就包括待审批的申请，如下图：



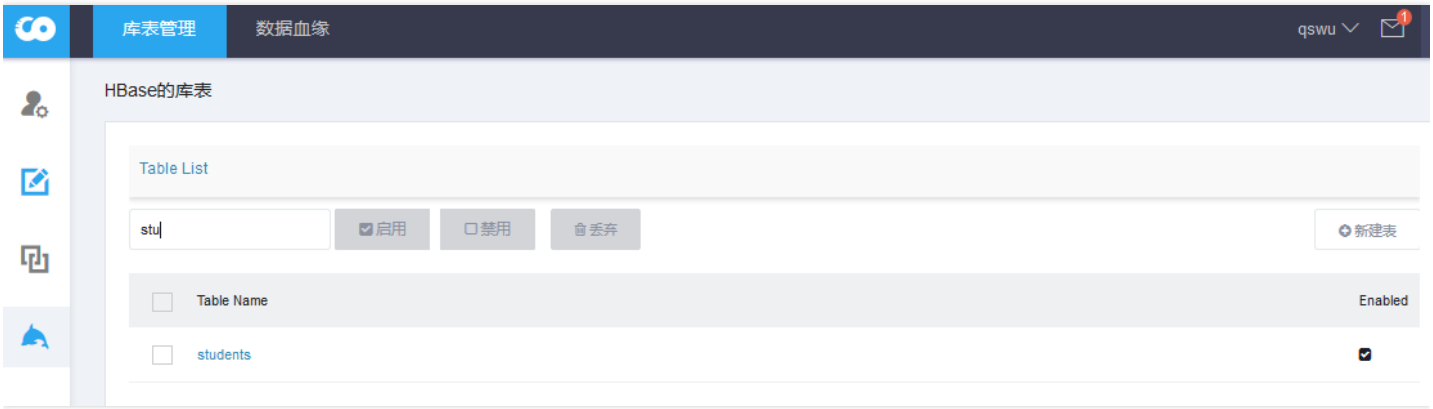
3. 审核进度。单击【我的申请】可以进入审核页面，如下图：

Hbase表管理

最近更新时间：2018-06-12 10:30:26

搜索 hbase 表

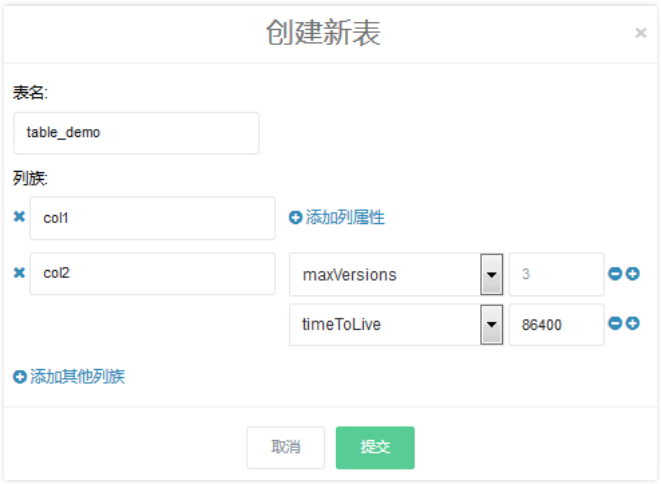
在库表管理左侧标签页中单击【Hbase 的库表】进入 hbase 表管理的单独页面，可以通过表名称搜索表。



新建 hbase 表

单击【库表管理】标签页右上角的【新建表】，填写配置信息，单击【提交】。

- 表名：以命名空间：表名称构成，如果不填写命名空间，默认为 default，普通用户没有权限在该命名空间中建表；需要在库表管理新建库表时，创建属于当前用户的 hbase 命名空间。
- 列族：可以添加多个，并且可以设置对应列族的属性。

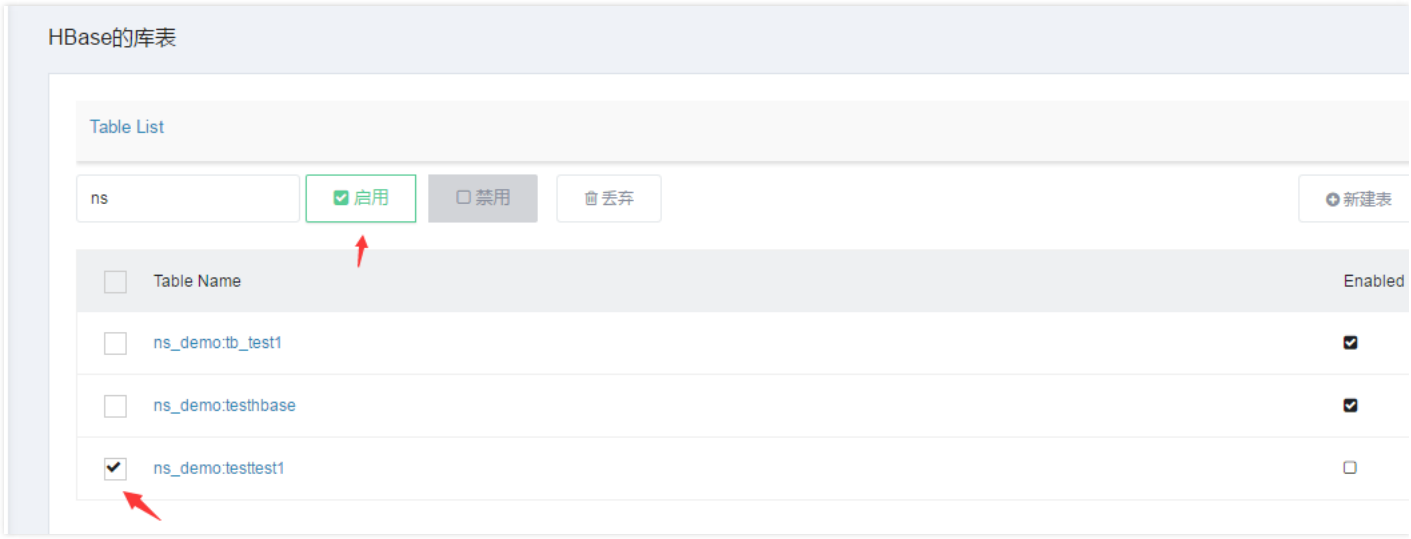


查看 hbase 表

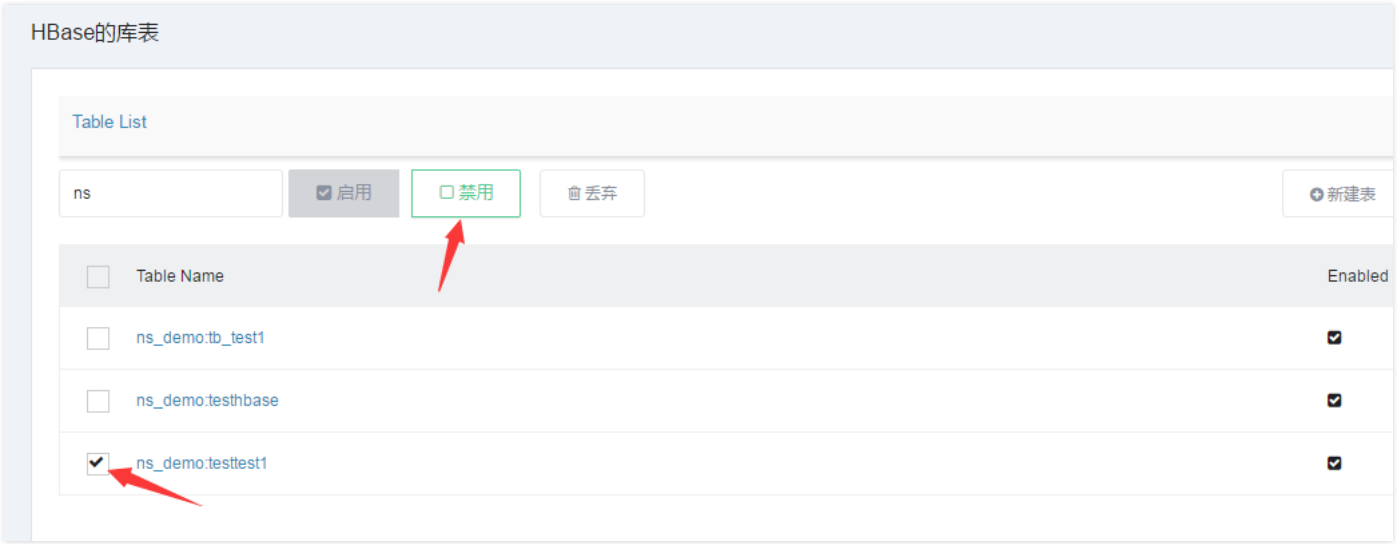
新建 hbase 表完成后即可在页面中看到当前创建的表，也可以搜索 ns_demo 这个命名空间的表查询。

hbase 表操作

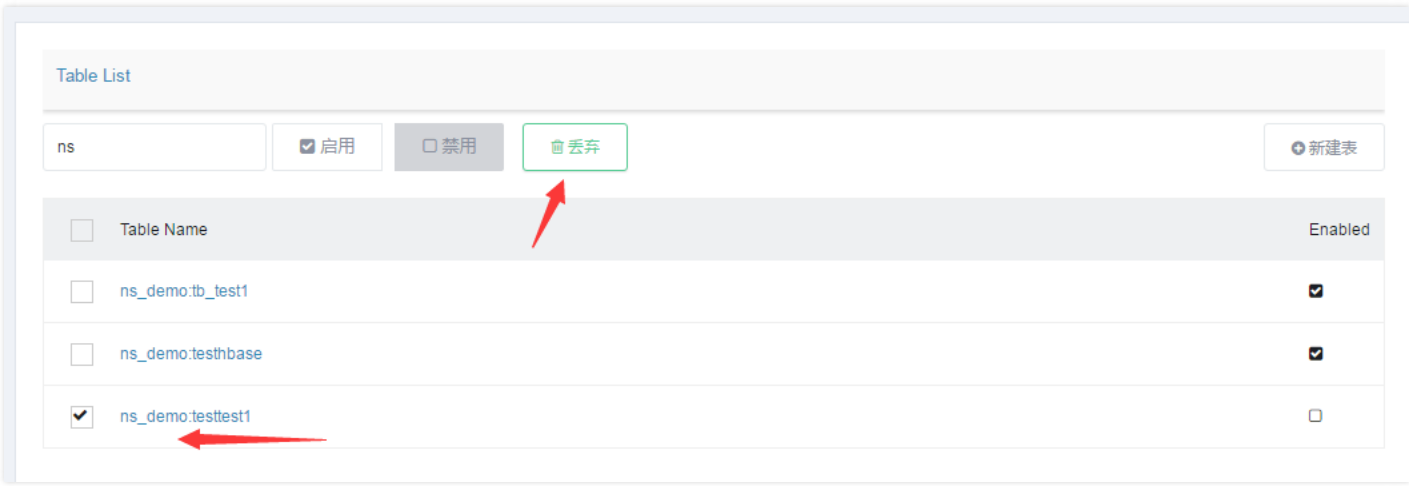
对有权限的表可以进行启用、禁用和丢弃操作。
启用（enable）操作：在库表列表选中要启用的列表，单击【启动】按钮。



禁用（disable）操作：在库表列表选中要启用的列表，单击【禁用】按钮。

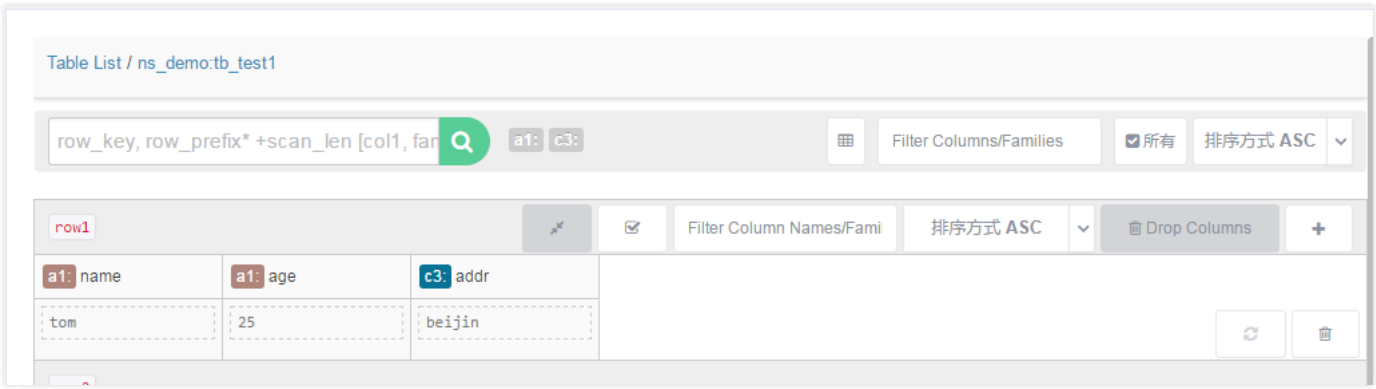


丢弃（delete）操作：在库表列表选中要启用的列表，单击【丢弃】按钮。



查看 hbase 表数据

单击表名称后，进入新页面可以看到当前表中的数据，并提供表数据查询、过滤、排序展示等功能。如下图所示：

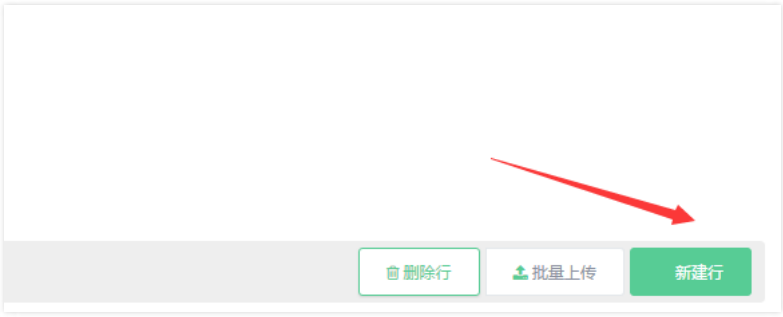


hbase 表数据操作

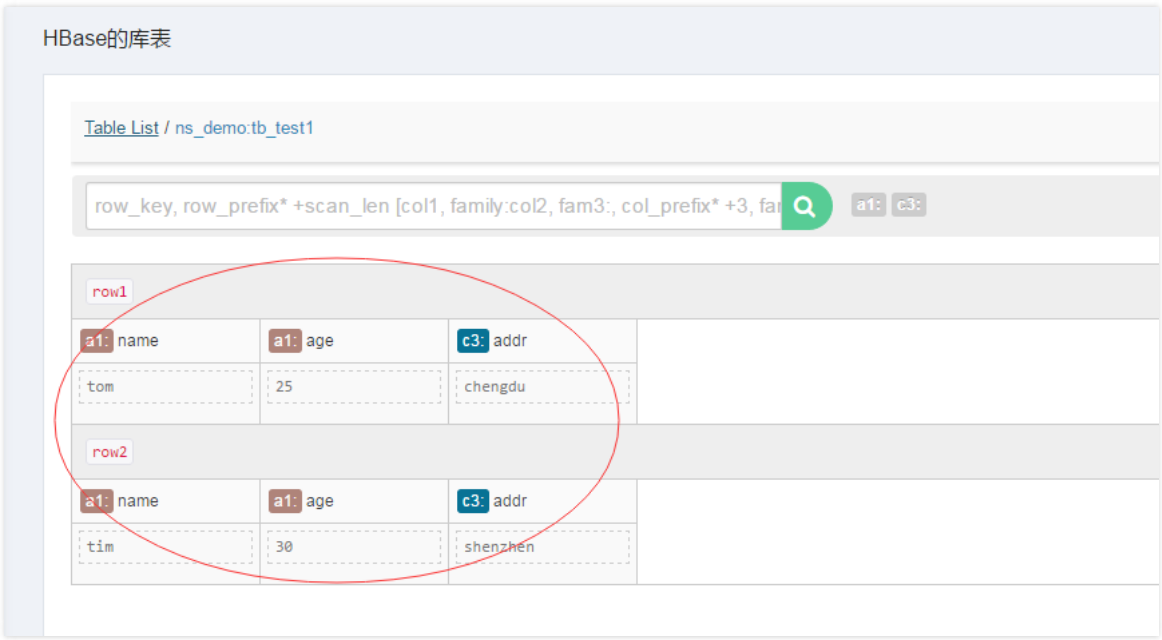
创建新行

单击【新建行】，添加行键、字段以及对应的值。

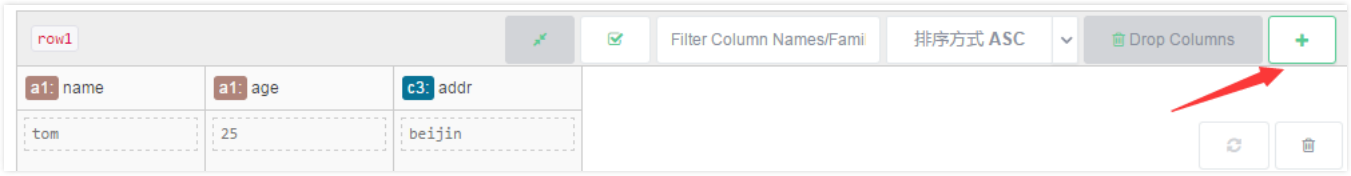
- 行键：rowkey，hbase 行唯一 ID。
- 字段：按照列族：列字段命名填写对应的 value 值即可。



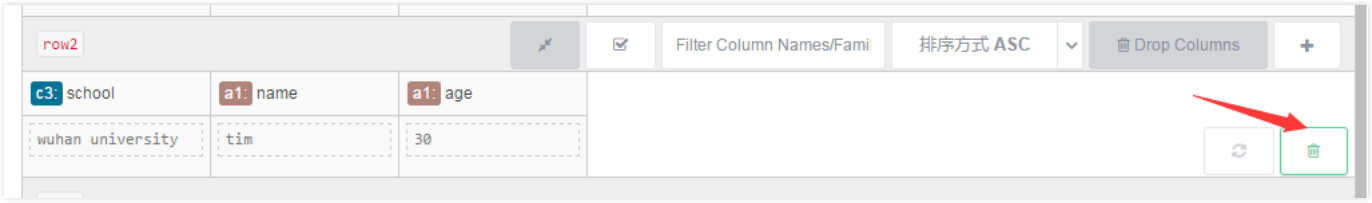
提交后可以看到表中的数据如下：



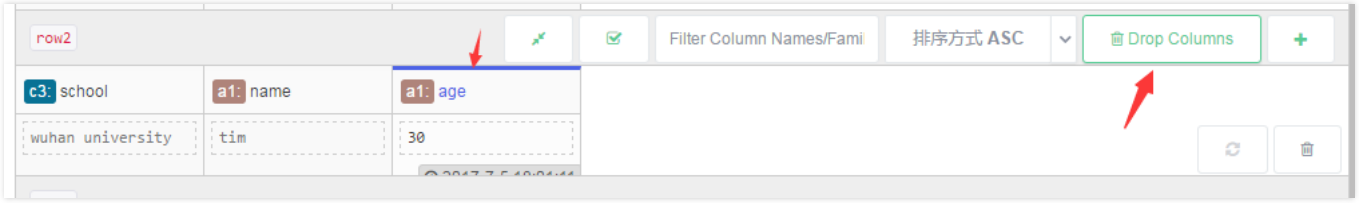
创建新列

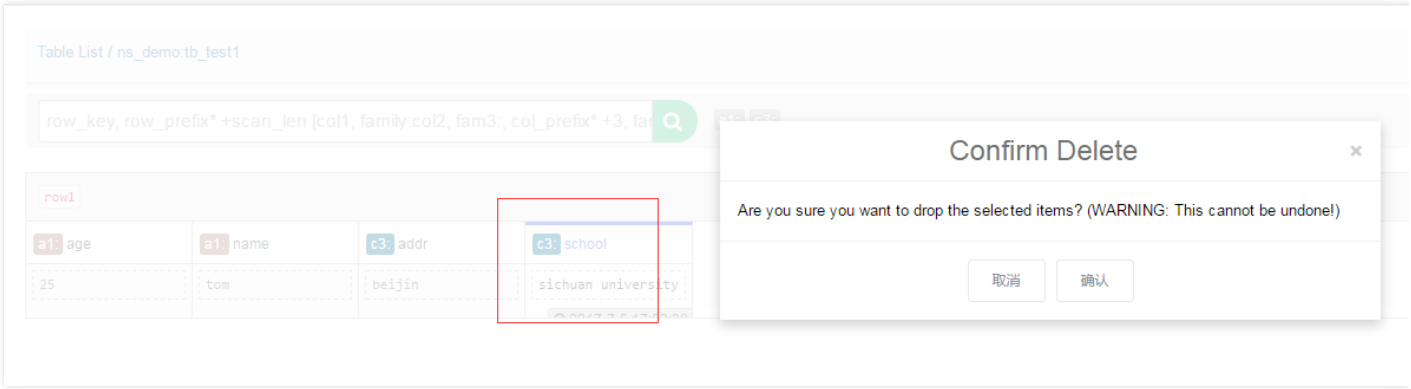


删除行

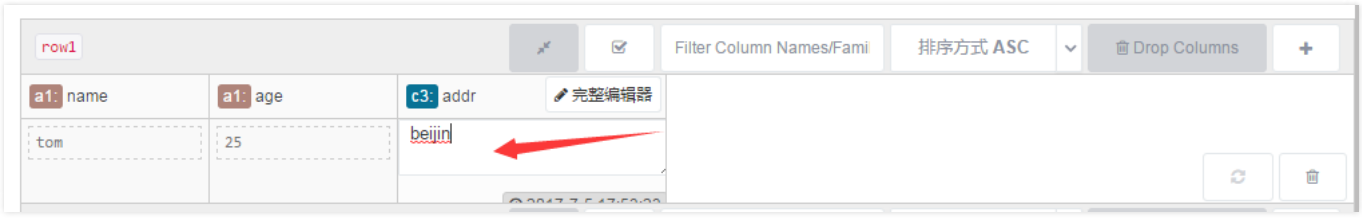


删除列





修改行数据

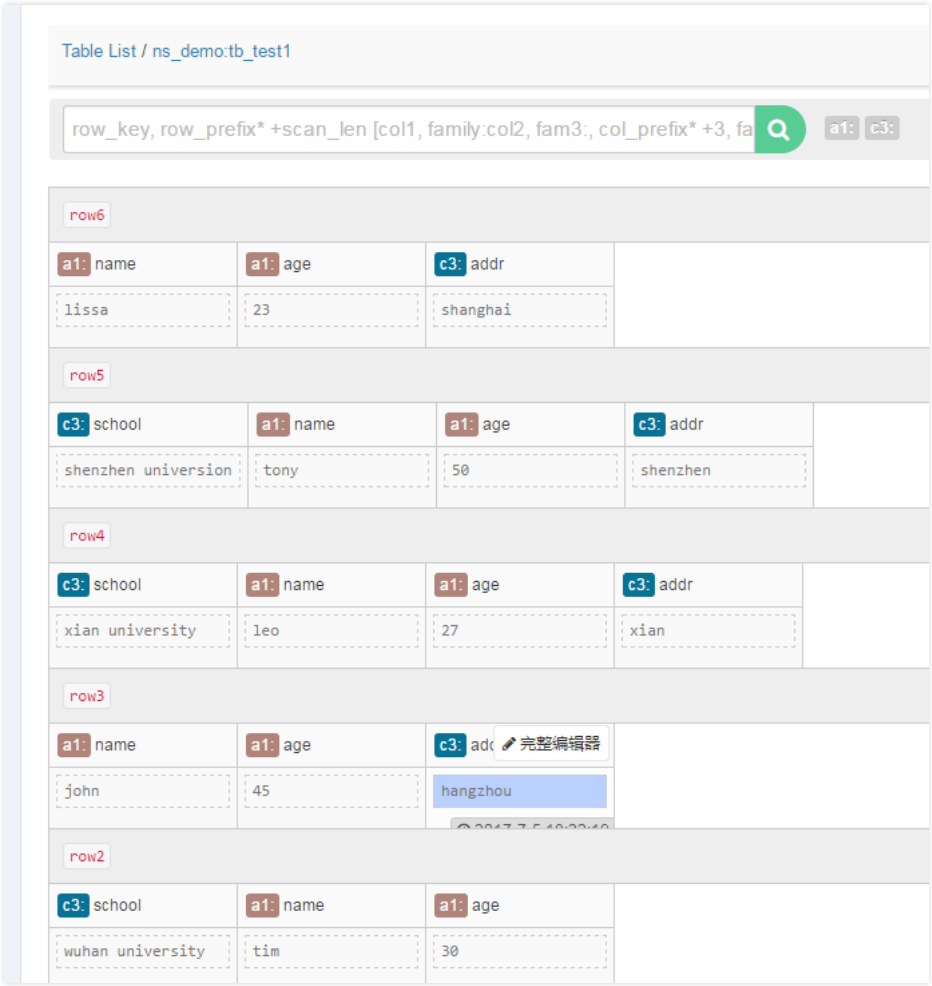


批量上传数据

csv文件示例：

A	B	C	D	E	F
row	a1:name	a1:age	c3:addr	c3:school	
row3	john	45	hangzhou		
row4	leo	27	xian	xian university	
row5	tony	50	shenzhen	shenzhen universion	
row6	lissa	23	shanghai		

上传选择该文件，即可查看到当前表中的数据如下：



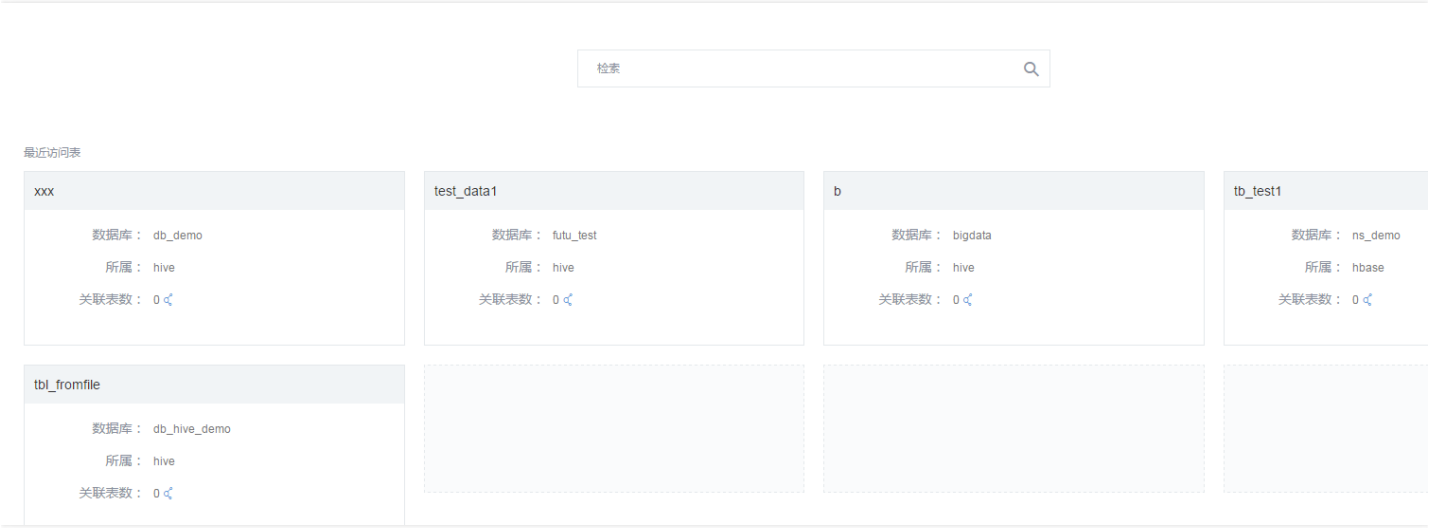
数据血缘

最近更新时间：2017-12-07 16:56:11

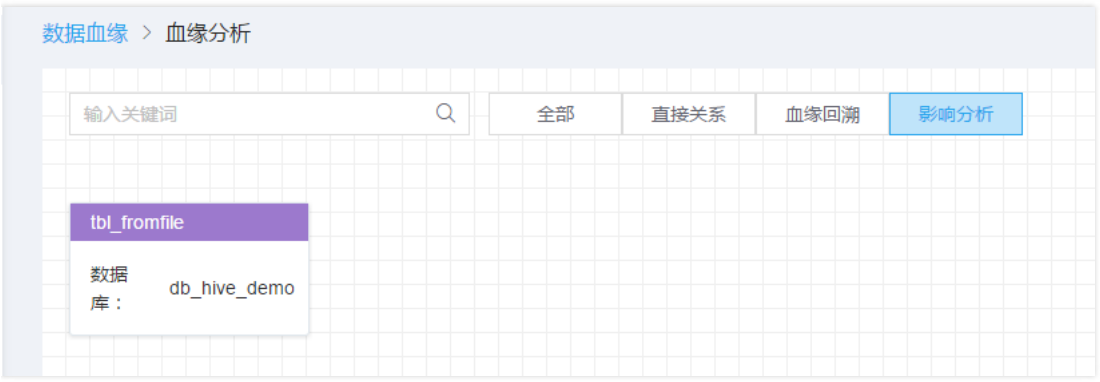
单击【数据血缘】进入数据血缘分析页面，可以看到基于库表管理和工作流分析的表与任务、表与表之间的关联分析。

全文检索

单击左侧菜单【全文检索】，可以在检索框中输入表名称进行检索。

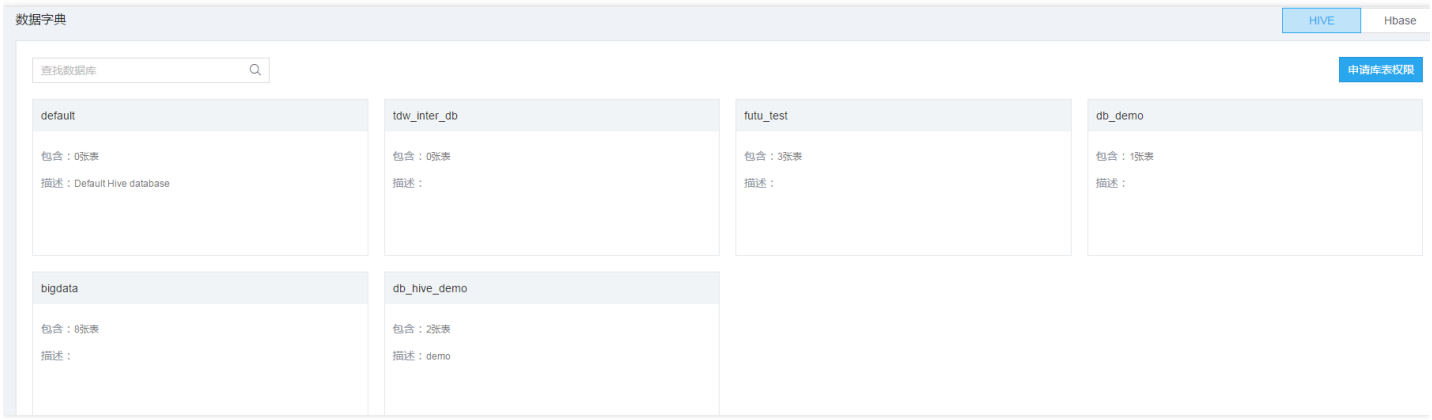


单击管理表的关联表数，跳转到该表的血缘分析页面，可以查询表的直系关系、血缘回溯以及影响分析，如下图所示：



数据字典

数据字典是数据库的视角来统计的，单击左侧菜单【数据字典】，可以在检索框中输入库名称进行检索。



单击其中一个数据库，可以列出当前数据库中的表列表信息，如下图所示：

单击表名称，可以查询当前表的表结构，表记录数，修订历史信息，如下图所示：

记录数可以展示最近7天、14天、30天的表记录数变化信息，如下图所示：

重要度排行

重要度排行是以数据库+表，关联关系和访问频度分析出来的结果，旨在标志这些库表的重要性，帮助用户做好此类关键库表的维护和使用。

平台管理
平台管理概述

最近更新时间：2017-12-07 16:56:35

腾讯大数据平台的平台管理提供了项目管理、资源管理、用户管理和系统设置服务，支持资源动态调整和资源申请，支持项目任务的可视化运维，形成了一个多角色大数据项目管理模型，为您提供点、线、面全方位的用户管理策略。

下面简单介绍下平台管理中的相关术语：

- **项目**：项目是大数据平台所有资源管理的基础，所有应用系统上线，都必须运行在分配好的项目之内，项目可以认为是一个大数据开发项目，也可以按照组织部门逻辑划分项目。项目包含唯一的资源队列，包括 CPU、内存、存储空间等，不同资源队列之间分配优先级。
- **用户**： 大数据平台的使用用户，隶属于项目，拥有开发、运维、管理员等数种项目成员权限。
- **角色**：分配给用户在大数据平台的权限的划分，包括系统管理员（超级用户）、项目管理员、项目开发、项目运维。
- **资源**：包括计算资源和存储资源，计算资源是 yarn 资源可以调度分配的 CPU 和内存资源；存储资源是 HDFS 的存储空间。
- **资源池**：是 yarn 分配指定资源队列，提供计算任务时需要的资源。

项目管理

最近更新时间：2018-06-12 10:31:46

大数据平台基于项目进行业务系统上线和管理，项目包含：项目成员、资源队列（CPU、内存、HDFS 空间）、项目优先级、项目内运行的最大程序数量和项目内数据存储目录等。

新建项目

如下图所示，单击右侧【创建项目】。

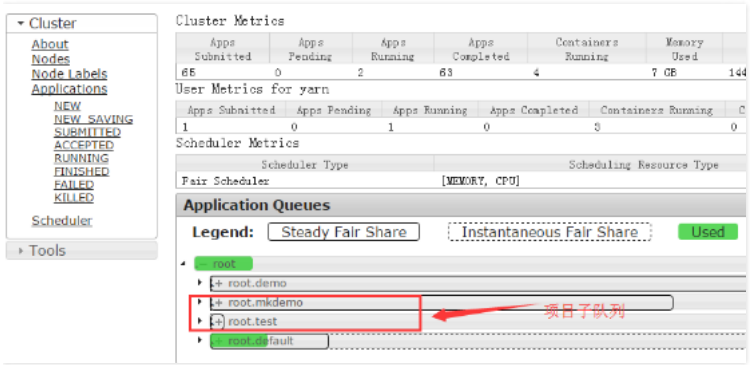
单击进入后，可以看到创建一个项目包含三个步骤，添加项目基本信息、新建资源池、配置存储资源。基本信息说明：

- 项目名称：项目的展示名称。
- 项目标识：项目唯一标识。
- 项目管理员：需要指定项目管理员，管理员用户有权对项目成员、资源进行调整。

新建资源池

资源池即为 yarn 的队列名称，大数据平台提供默认的资源队列 root.default（默认分配整个资源池的10%）。

项目创建后，系统会自动为项目创建一个 yarn 的子队列用于项目使用，如下所示在 yarn 的资源管理界面中可以看到资源池名称为 test、demo、mkdemo 的示例，每创建一个项目则会分配一个资源队列。



配置存储资源

通过拖动鼠标自定义存储空间大小，单击【创建】。存储目录（HDFS 路径）平台按照 */project/项目名称/* 进行命名，该项目的成员可以在当前的目录下进行文件操作。

资源管理

最近更新时间：2017-12-07 16:56:54

资源状态查看

资源状态查看提供一个总体的集群资源查看视图，右侧可以按照不同的时间粒度进行查询，同时提供了项目使用的资源状态，方便系统管理员调整项目资源大小。

资源配置

单击左侧【资源配置】，进入资源配置页面。

资源管理在首次创建项目分配计算池资源和存储池资源后，也可以在此处动态调整项目拥有的资源。通常在集群扩容，或者资源优先级调度会使用到此处。在相应资源池的操作栏中，单击【配置】，编辑资源池配置信息后，单击【确定】，即可调整项目拥有的资源。

在多个应用使用的情况下，大数据平台资源管理可以通过限制每个应用占用的系统资源，避免出现整个平台的资源被一个应用锁死的情况。