



数领先机 智赢未来

DTCC 第九届中国数据库技术大会
DATABASE TECHNOLOGY CONFERENCE CHINA 2018

2018.05.10 – 12 北京国际会议中心



大数据实时计算 Flink SQL 解密

伍翀 (云邪)

2018.05



姓名：伍翀

花名：云邪

硕士毕业于
北京理工大学

2015

阿里巴巴
实时计算引擎 JStorm
的开发与设计

阿里巴巴
新一代实时计算引擎
BlinkSQL 开发与优化

2016

Flink Committer
2017年2月

2017

Now

目录

1 Background

2 Flink SQL 基本概念

3 Flink SQL 核心功能

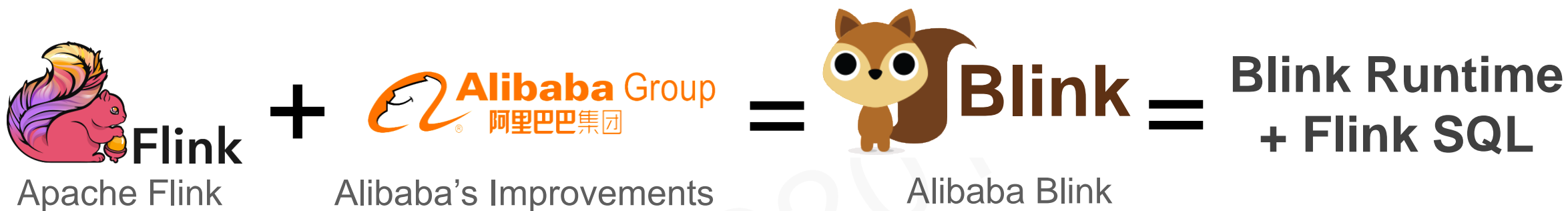
4 Flink SQL 优化

5 阿里云流计算产品

Background

Part I

Alibaba Blink



阿里巴巴Blink团队有 20+ flink contributor , 6 名 committer , 向社区贡献了数百个Commit

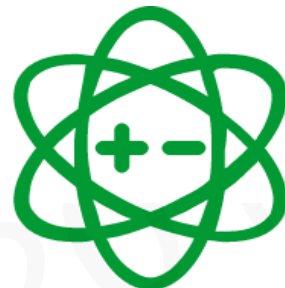
团队工作



主导制定 Flink SQL 语义

- DynamicTable 2016-2017
- Retraction 2016-2017

贡献社区



完善 Flink SQL 功能

- Agg, Join, Window 2017
- 跑通全部 TPCH Query 2018

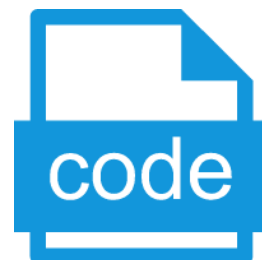
贡献社区



性能提升

- 大量的查询优化 2017-2018

部分贡献社区



资源配置自动化 2018

Flink SQL

Section 2

Why SQL?



Declarative



Optimized



Understandable



Stable



Unify

One Query, One Result

SQL 不是为流设计的

批处理

数据是有限的



批计算查询返回一个结果并结束

没有Retraction

VS.

流处理

流数据是无穷的



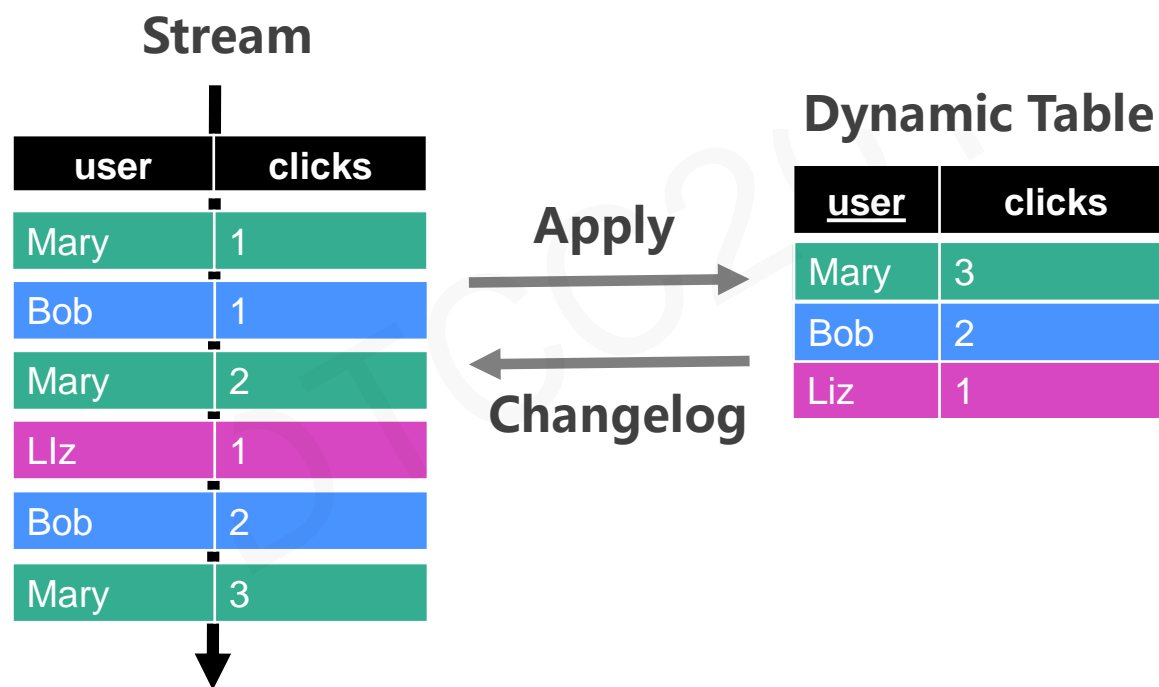
流上的查询不断产生结果且不会结束

有对历史数据的修改(Retraction)

VS.

动态表 (Dynamic Table)

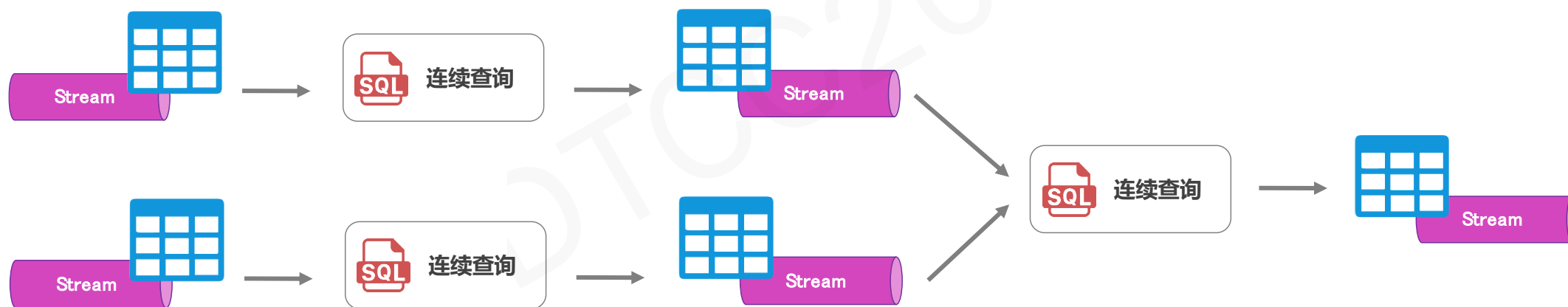
- 动态表 (Dynamic Table) : 数据会随着时间变化的表



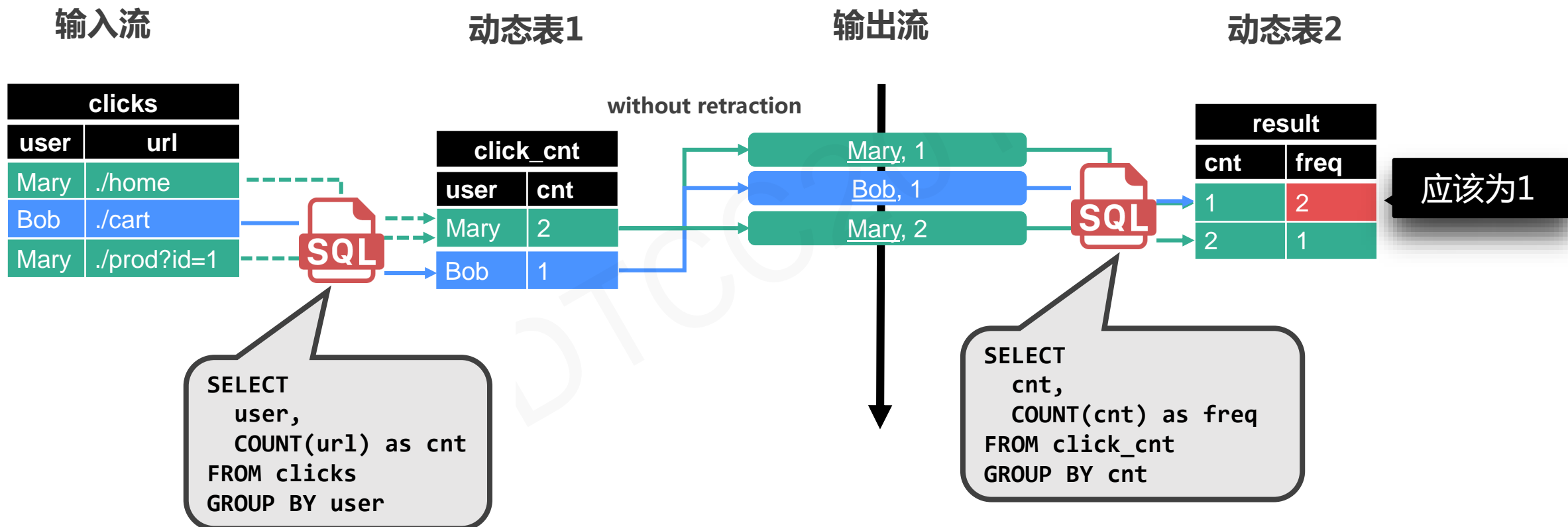
流与动态表的对偶性

动态表 + 连续查询

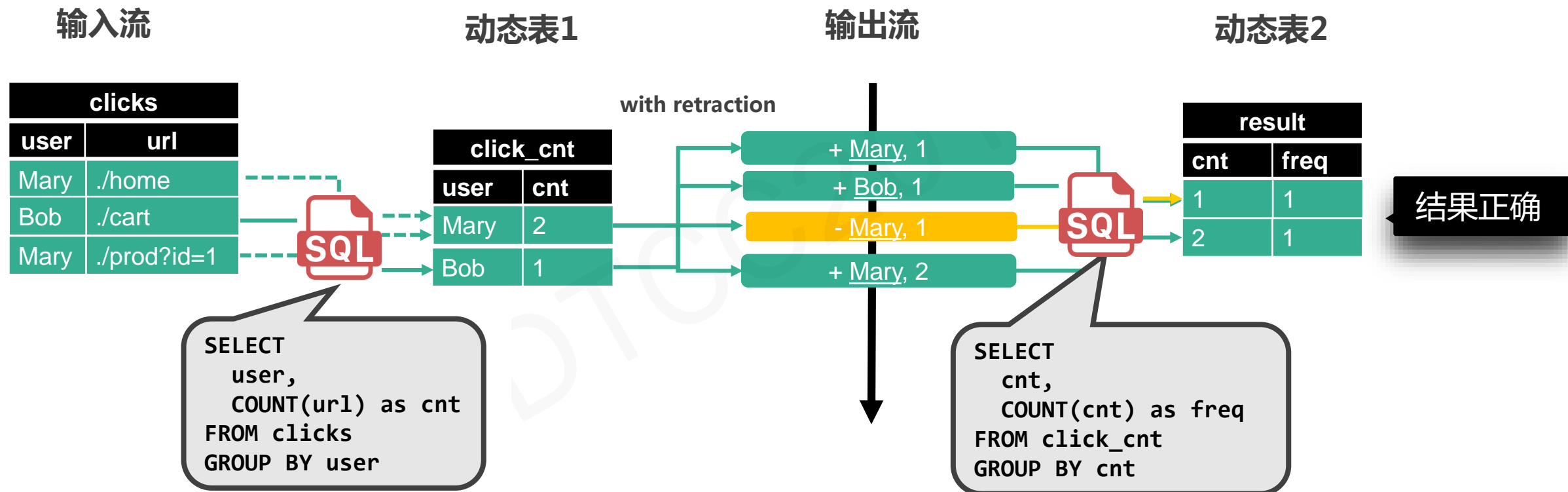
- 连续查询（ Continuous Query ）：持续运行的查询



流计算 Retraction



流计算 Retraction



由查询优化器判断是否需要Retraction，用户无感知。

世界上不需要所谓的 Stream SQL
标准的 ANSI SQL 就可以用来定义流计算

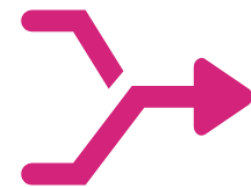
Flink SQL 核心功能



DDL & DML



UDF/UDTF/UDAF



Join



Group Agg



Window Agg



Over Agg

Loading Data

-- 定义数据源表

```
CREATE TABLE clicks (  
  user    VARCHAR,  
  cTime   TIMESTAMP,  
  url     VARCHAR  
) WITH (  
  type = 'kafka',  
  topic = 'click_topic',  
  ...  
);
```

```
SELECT * FROM clicks
```

user	cTime	url
Mary	12:00:00	./home
Bob	12:00:00	./cart
Mary	12:00:05	./prod?id=1

Saving Data

```
-- 定义数据结果表
CREATE TABLE last_clicks (
    user    VARCHAR,
    cTime   TIMESTAMP,
    url     VARCHAR,
    PRIMARY KEY (user)
) WITH (
    type = 'mysql',
    ...
);

INSERT INTO last_clicks
SELECT * FROM clicks
```

Multi Output

```
CREATE TABLE mysql_clicks (  
    user    VARCHAR,  
    cTime   TIMESTAMP,  
    url     VARCHAR,  
    PRIMARY KEY (user)  
) WITH (  
    type = 'mysql',  
    ...  
);
```

```
CREATE TABLE hbase_clicks (  
    user    VARCHAR,  
    cTime   TIMESTAMP,  
    url     VARCHAR,  
    PRIMARY KEY (user)  
) WITH (  
    type = 'hbase',  
    ...  
);
```

```
CREATE VIEW taobao_clicks AS  
SELECT * FROM clicks WHERE url LIKE 'http://taobao.com%'
```

```
INSERT INTO mysql_result  
SELECT * FROM taobao_clicks
```

```
INSERT INTO hbase_result  
SELECT * FROM taobao_clicks
```

Group Aggregate

从历史到现在每个用户点击的次数

clicks		
user	cTime	url
Mary	12:00:00	./home
Bob	12:00:00	./cart
Mary	12:00:05	./prod?id=1
Mary	12:01:45	./prod?id=7

```
SELECT
  user,
  COUNT(url) as cnt
FROM clicks
GROUP BY user
```

result	
user	cnt
Mary	3
Bob	1

Window Aggregate

每小时每个用户点击的次数

clicks		
user	cTime	url
Mary	12:00:00	./home
Bob	12:00:00	./cart
Mary	12:02:00	./prod?id=2
Mary	12:55:00	./home
Bob	13:01:00	./prod?id=4
Liz	13:30:00	./cart
Liz	13:59:00	./home

```
SELECT
  user,
  TUMBLE_END(
    cTime,
    INTERVAL '1' HOURS)
  AS endT,
  COUNT(url) AS cnt
FROM clicks
GROUP BY
  user,
  TUMBLE(
    cTime,
    INTERVAL '1' HOURS)
```

result		
user	endT	cnt
Mary	13:00:00	3
Bob	13:00:00	1
Bob	14:00:00	1
Liz	14:00:00	2

双流 JOIN : 支持 INNER, LEFT, RIGHT, FULL, SEMI, ANTI

Orders		
orderId	productId	orderTime
5	30	10:17:00
6	10	10:17:05
9	10	11:02:00
12	10	11:24:11

Shipments	
orderId	shipTime
5	10:55:00
6	10:20:00
9	11:58:00
12	11:44:00

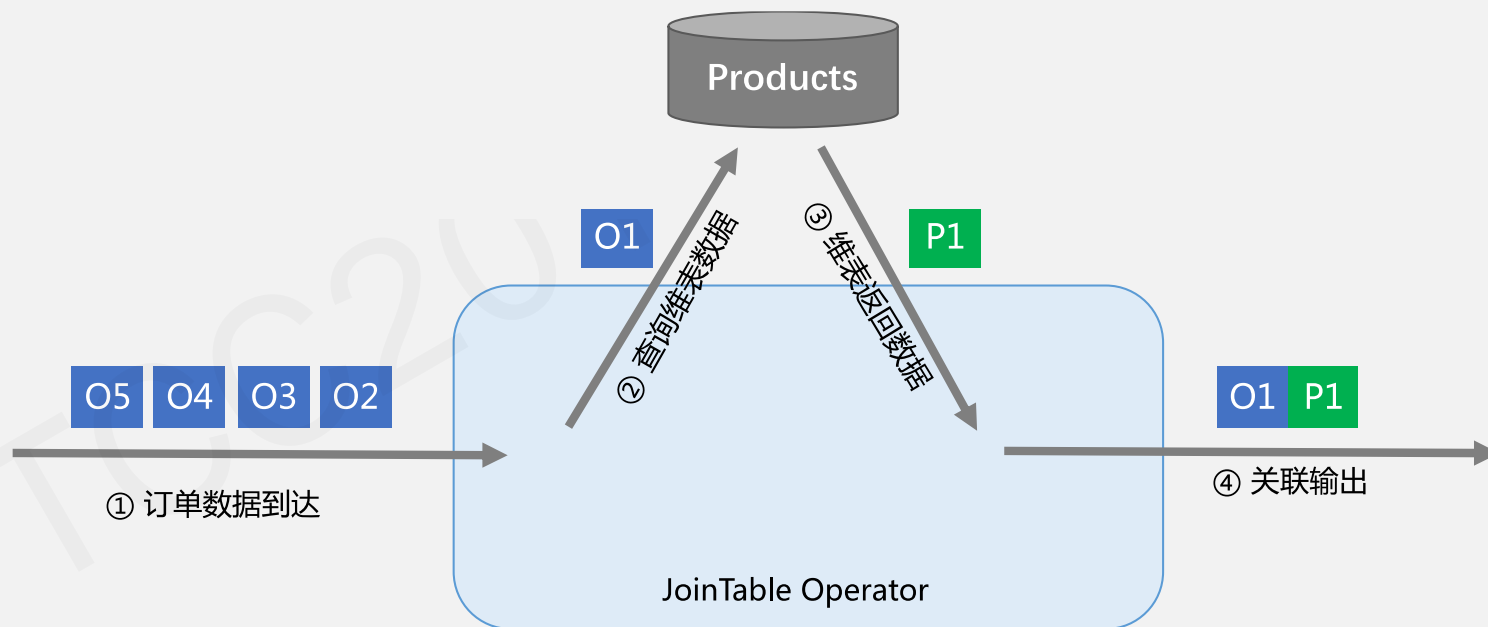
```
SELECT
  o.orderId,
  o.productId,
  o.orderTime,
  s.shipTime
FROM Orders AS o
JOIN Shipments AS s
ON o.orderId = s.orderId
```

result			
orderId	productId	orderTime	shipTime
5	30	10:17:00	10:55:00
6	10	10:17:05	10:20:00
9	10	11:02:00	11:58:00
12	10	11:24:11	11:44:00

维表 JOIN：支持 INNER, LEFT

```
CREATE TABLE Products (  
  productId VARCHAR,  
  productName VARCHAR,  
  price DECIMAL,  
  PRIMARY KEY (productId),  
  PERIOD FOR SYSTEM_TIME  
) WITH (  
  type = 'hbase'  
  ...  
);
```

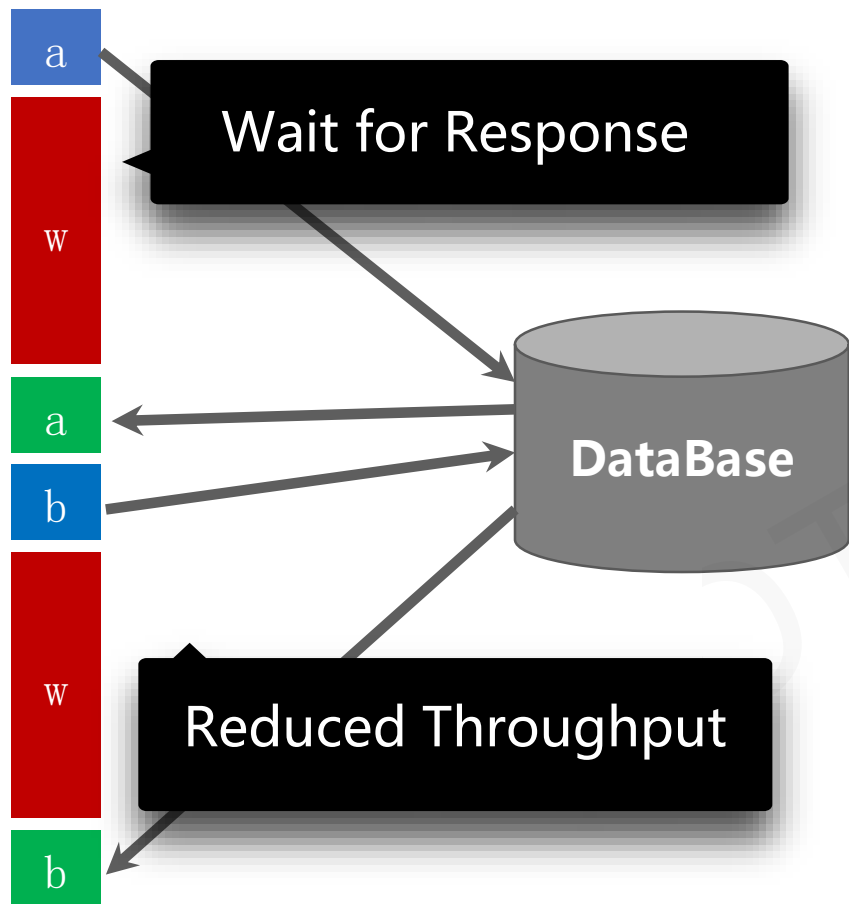
```
SELECT o.*, p.*  
FROM Orders AS o  
JOIN Products FOR SYSTEM_TIME AS OF PROCTIME() AS p  
ON o.productId = p.productId
```



聊几个优化

异步维表 JOIN

Sync. IO



Send Request

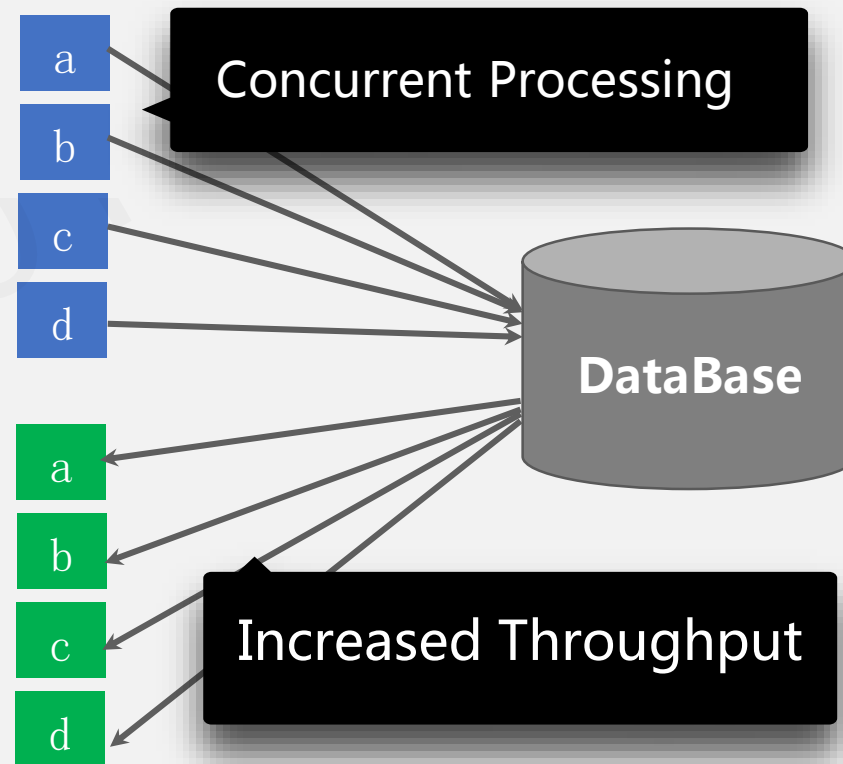


Receive Request



Wait

Async. IO



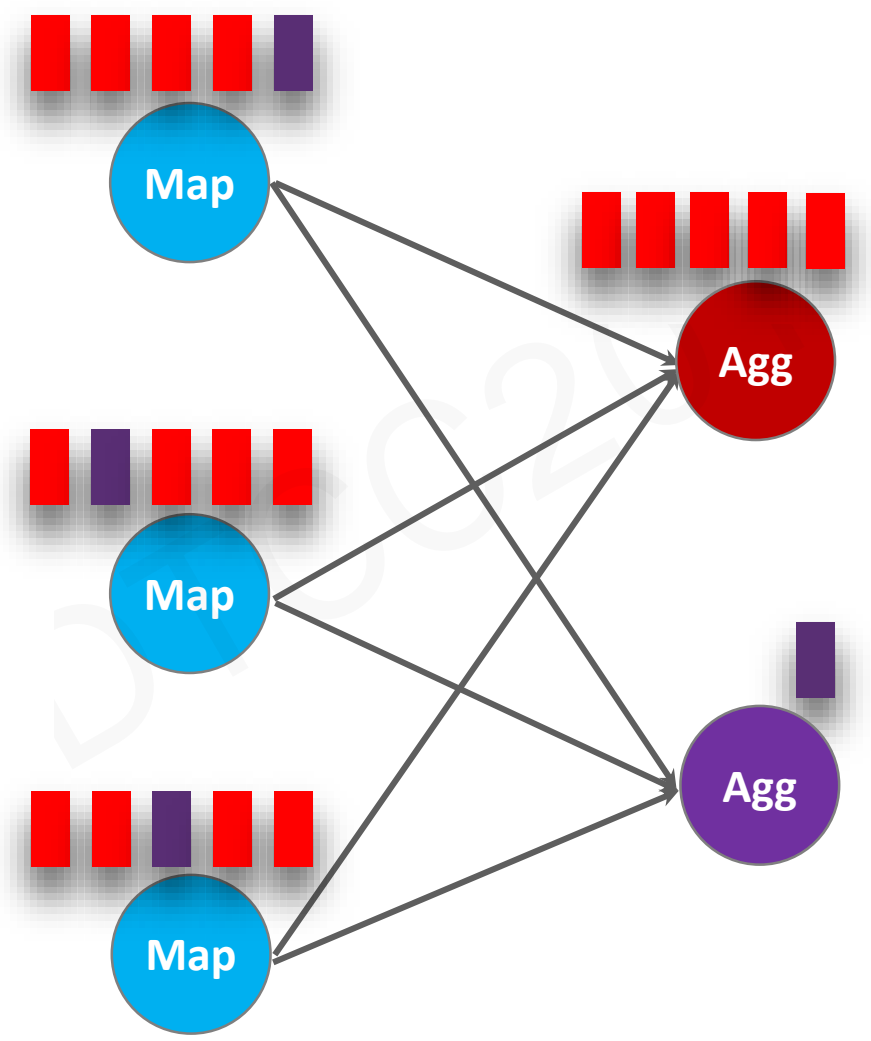
异步维表 JOIN

```
CREATE TABLE Products (  
    productId VARCHAR,  
    productName VARCHAR,  
    price DECIMAL,  
    PRIMARY KEY (productId),  
    PERIOD FOR SYSTEM_TIME  
) WITH (  
    type = 'hbase',  
    async = 'true'  
    ...  
);
```

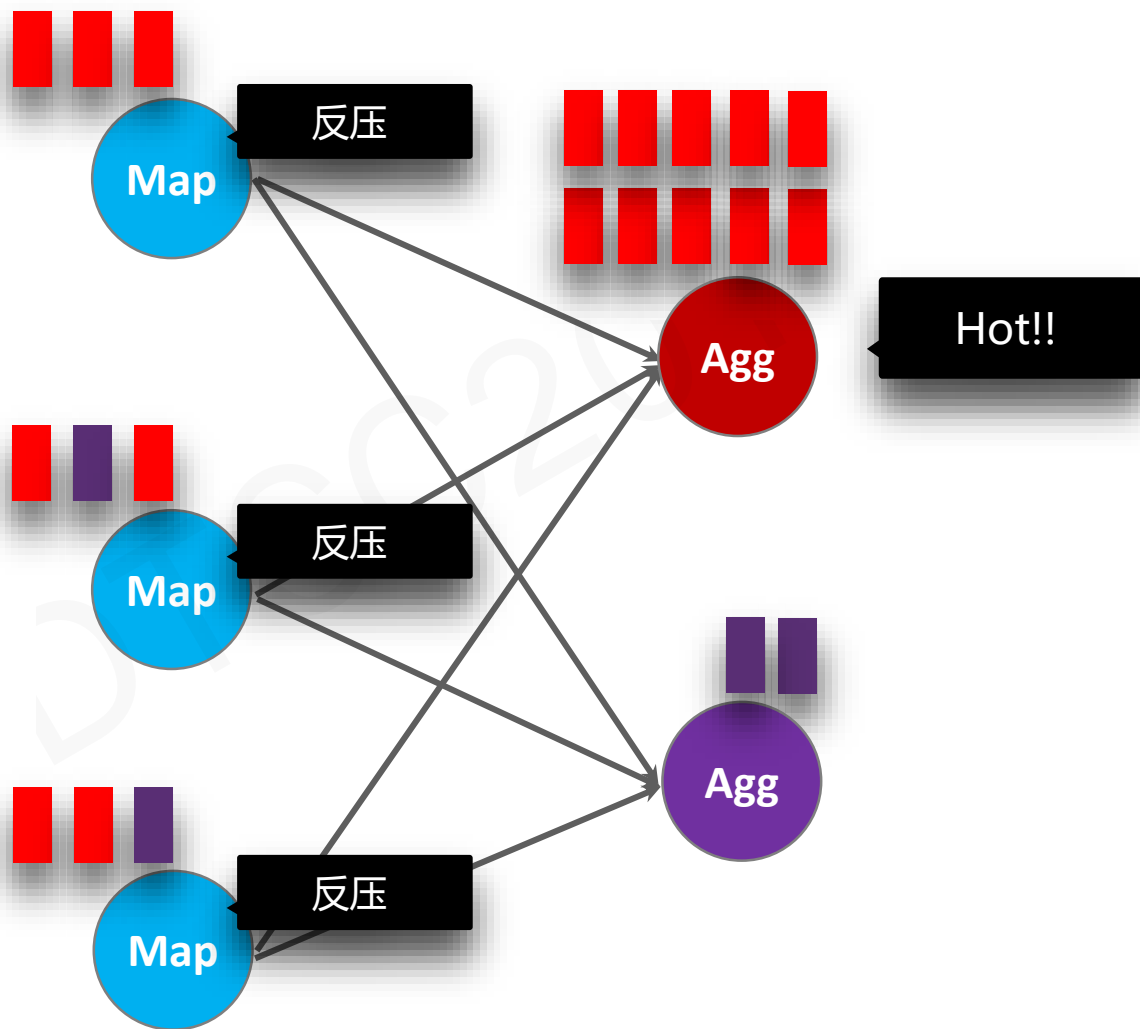
一行配置的改动

```
SELECT o.*, p.*  
FROM Orders AS o  
JOIN Products FOR SYSTEM_TIME AS OF PROCTIME()  
AS p  
ON o.productId = p.productId
```

如何处理数据倾斜Data-Skew



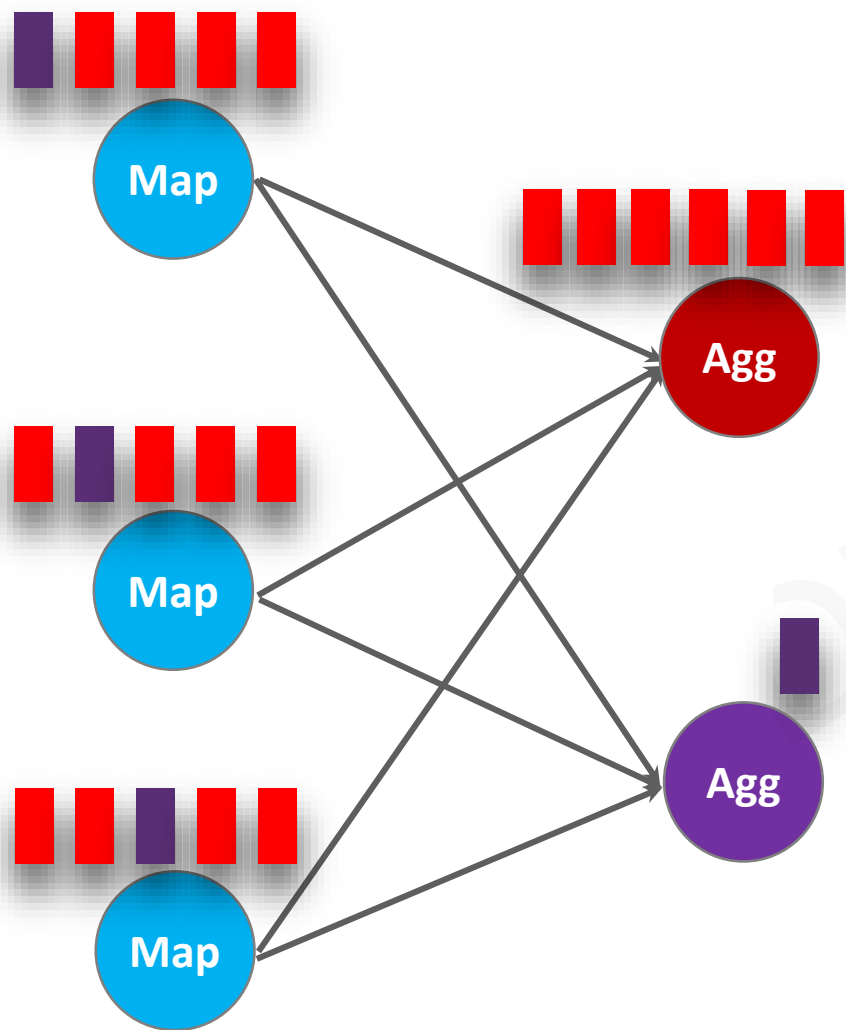
如何处理数据倾斜Data-Skew



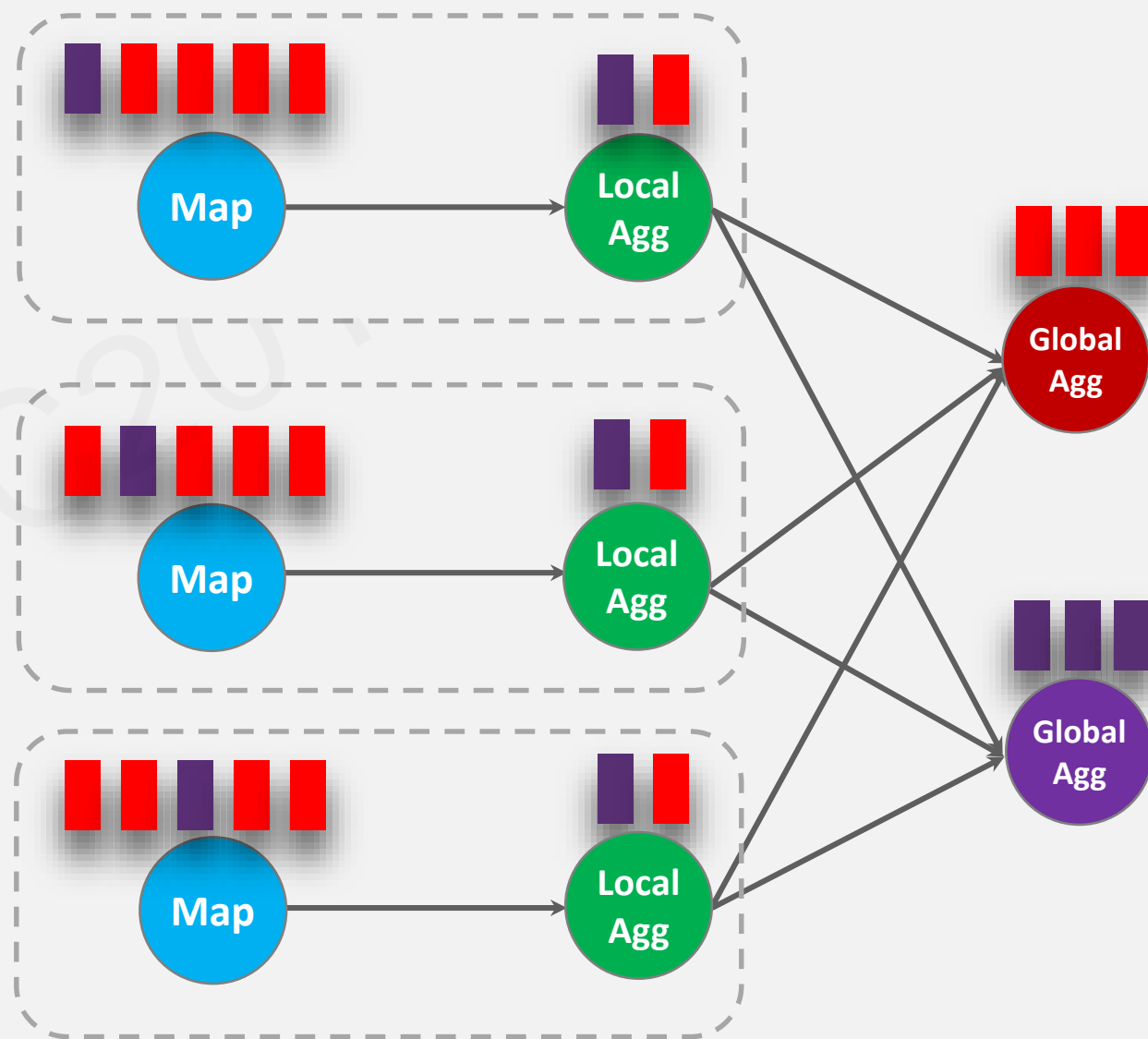
Local-Global Aggregation 优化

如何处理数据倾斜Data-Skew

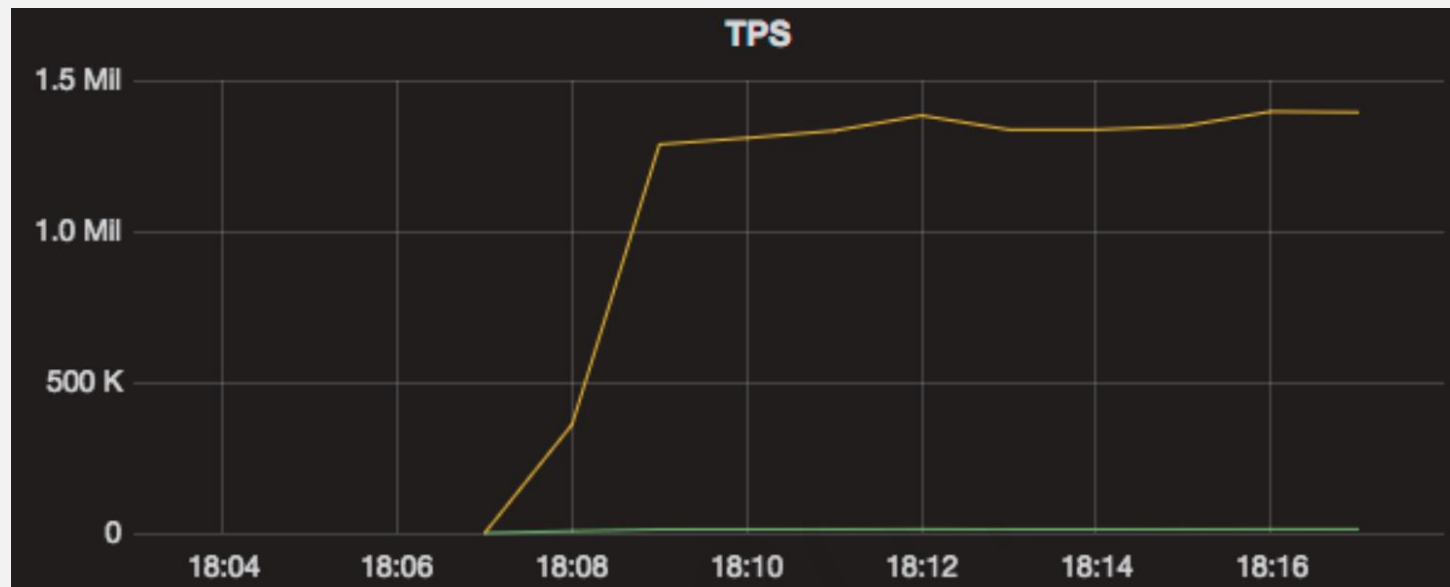
Simple Aggregation



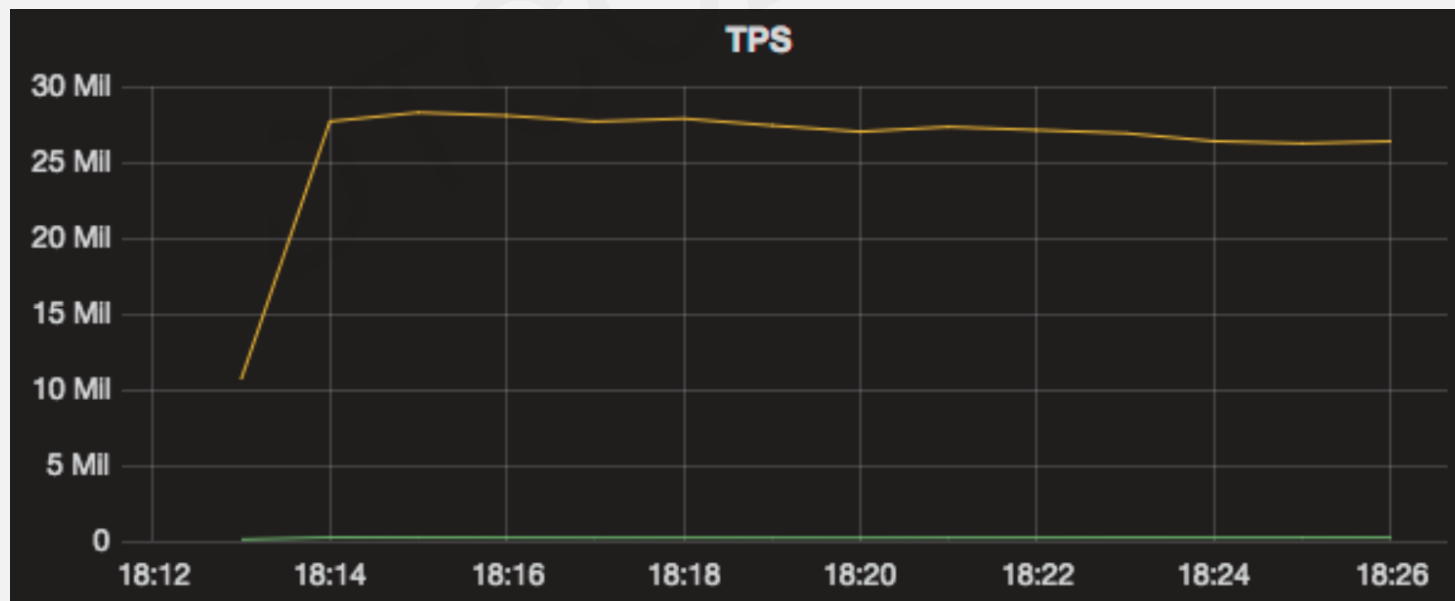
Local-Global Aggregation



Local-Global
带来 20X
的性能提升



优化前



优化后



阿里云流计算产品

Section 3

阿里流计算开发平台

blink_user

总览 开发 运维 帮助

开发作业

资源引用

运行引擎

新建 保存 另存为 撤销 重做 查找 调试 上线 运维 设置

window_test

```
7
8 CREATE TABLE tt_stream(
9   a varchar,
10  b varchar,
11  c as PROCTIME()
12 ) with (
13   type = 'tt',
14   topic = 'datahub-test',
15   accessId = 'oss-11718330000000000000',
16   accessKey = 'a62c7c86-ba5a-4ec7-8e52-b97ad2618e7d'
17 );
18
19 CREATE TABLE rds_output(
20   id varchar,
21   c TIMESTAMP,
22   f TIMESTAMP,
23   cnt BIGINT
24 ) with (
25   type = 'rds',
26   url = 'jdbc:mysql://11.239.142.113:3306/test',
27   tableName = 'datahub2rds',
28   userName = 'root',
29   password = '123456'
30 );
31
32 INSERT INTO rds_output
33 SELECT a AS id,
34        TUMBLE_START(c, INTERVAL '1' MINUTE) AS c,
35        TUMBLE_END(c, INTERVAL '1' MINUTE) AS f,
36        COUNT(a) AS cnt
37 FROM tt_stream
38 GROUP BY TUMBLE(c, INTERVAL '1' MINUTE), a
39
```

资源配置 作业属性 作业参数

开发作业

资源引用

运行引擎

hbase_dim 10/17

test_123 08/04 1

test_view 09/11

tt_test_123 08/29

window_test 09/1

阿里流计算开发平台

blink_user

总览 开发 运维 帮助

开发作业

资源引用

运行引擎

新建 保存 另存为 撤销 重做 查找 调试 上线 运维 设置


window_test

```
7
8 CREATE TABLE tt_stream(
9   a varchar,
10  b varchar,
11  c as PROCTIME()
12 ) with (
13   type = 'tt',
14   topic = '...',
15   accessId = '...',
16   accessKey = '...',
17 );
18
19 CREATE TABLE rds_output (
20   id varchar,
21   c TIMESTAMP,
22   f TIMESTAMP,
23   cnt BIGINT
24 ) with (
25   type = 'rds',
26   url = 'jdbc:mysql://...',
27   tableName = 'datahub2101',
28   userName = 'root',
29   password = '123456',
30 );
31
32 INSERT INTO rds_output
33 SELECT a AS id,
34        TUMBLE_START(c, INTERVAL '1' MINUTE) AS c,
35        TUMBLE_END(c, INTERVAL '1' MINUTE) AS f,
36        COUNT(a) AS cnt
37 FROM tt_stream
38 GROUP BY TUMBLE(c, INTERVAL '1' MINUTE), a
39
```

资源引用


作业属性

作业参数




Develop


- hbase_dim 10/17
- test_123 08/04 1
- test_view 09/11
- tt_test_123 08/29
- window_test 09/



Debug



Deploy

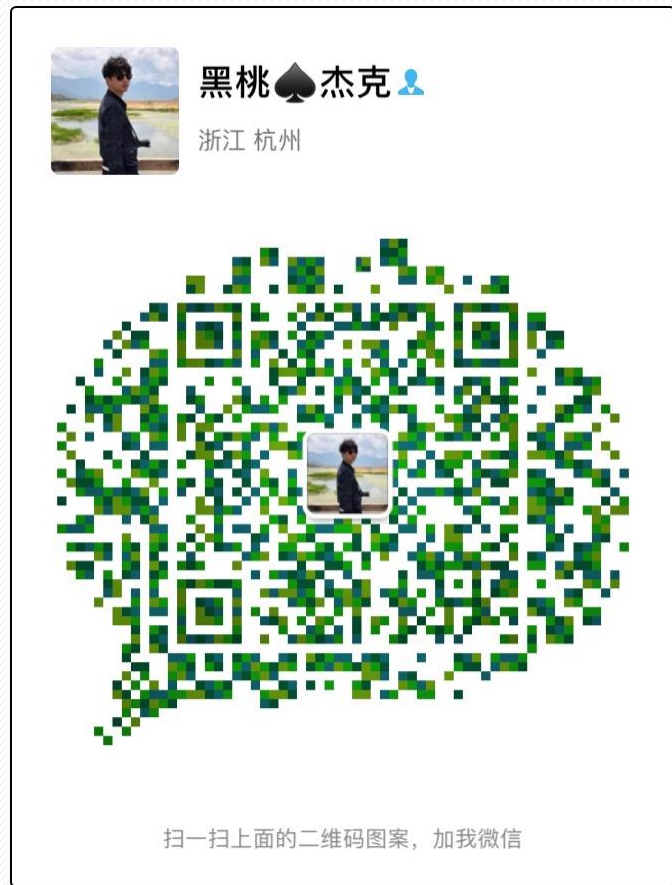


Monitor

拥抱开源，做世界第一的计算平台

Thanks

Q&A





讲师申请

联系电话（微信号）：18612470168

关注“ITPUB”更多
技术干货等你来拿~

与百度外卖、京东、魅族等先后合作系列分享活动



让学习更简单

微学堂是以ChinaUnix、ITPUB所组建的微信群为载体，定期邀请嘉宾对热点话题、技术难题、新产品发布等进行移动端的在线直播活动。

截至目前，累计举办活动期数60+，参与人次40000+。

ITPUB学院

ITPUB学院是盛拓传媒IT168企业事业部（ITPUB）旗下
企业级在线学习咨询平台
历经18年技术社区平台发展
汇聚5000万技术用户
紧随企业一线IT技术需求
打造全方式技术培训与技术咨询服务
提供包括企业应用方案培训咨询（包括企业内训）
个人实战技能培训（包括认证培训）
在内的全方位IT技术培训咨询服务

ITPUB学院讲师均来自于企业
一些工程师、架构师、技术经理和CTO
大会演讲专家1800+
社区版主和博客专家500+

培训特色

无限次免费播放
随时随地在线观看
碎片化时间集中学习
聚焦知识点详细解读
讲师在线答疑
强大的技术人脉圈

八大课程体系

基础架构设计与建设
大数据平台
应用架构设计与开发
系统运维与数据库
传统企业数字化转型
人工智能
区块链
移动开发与SEO



联系我们

联系人：黄老师
电话：010-59127187
邮箱：edu@itpub.net
网址：edu.itpub.net
培训微信号：18500940168