Blink SQL关键技术及实现原理

孙金城(金竹)

About Me

计算平台事业部–实时计算部

孙金城(金竹)

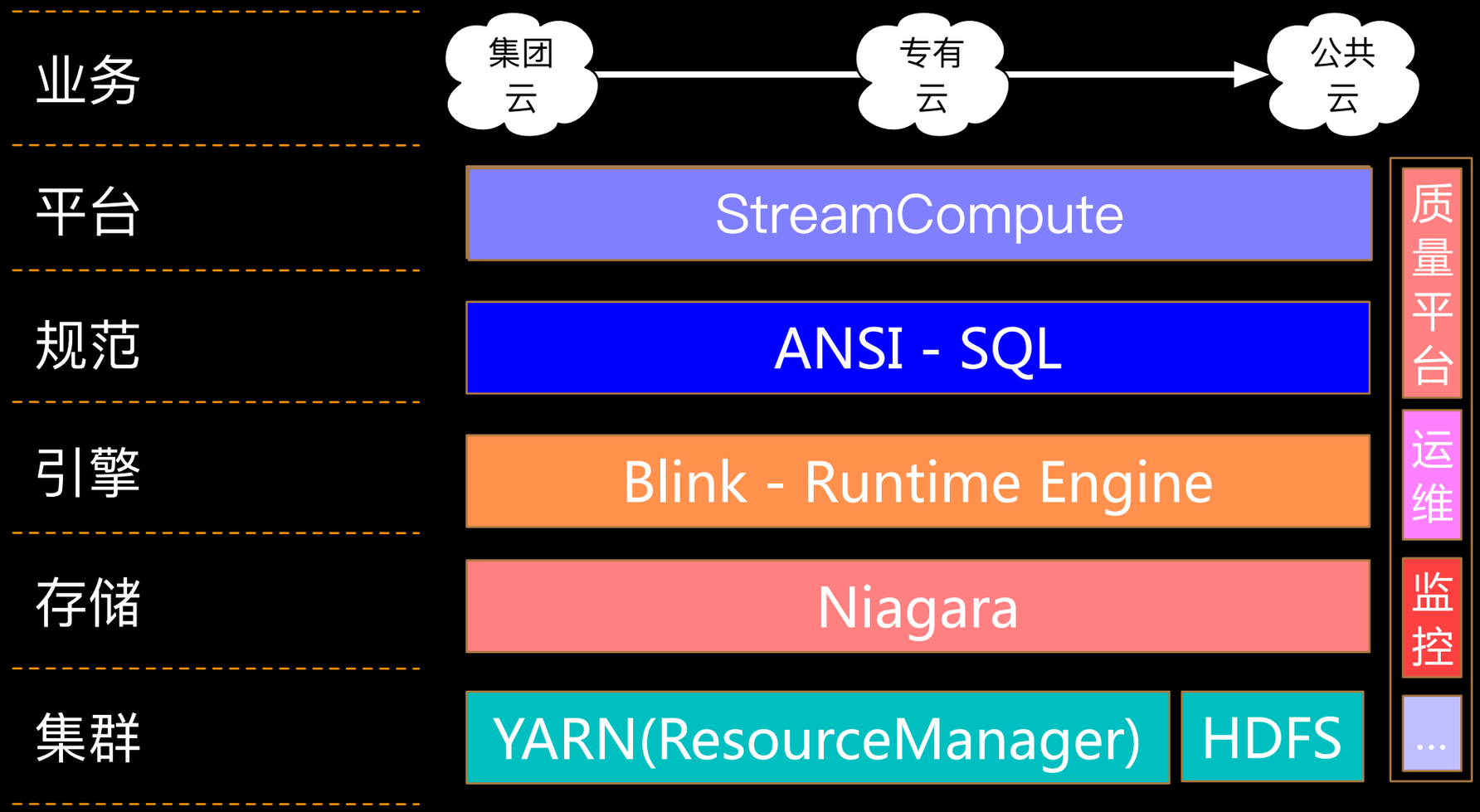
Blink

Flink

- 2011：加入Alibaba 底层技术平台研发
- 2016：Alibaba Blink SQL 设计研发
- 2017：Apache Flink Committer
- Now：Alibaba Blink 生态

目录

Blink 生态


业务
集团云
专有云
公共云
平台
StreamCompute
规范
ANSI - SQL
引擎
Blink - Runtime Engine
存储
Niagara
集群
YARN(ResourceManager)
HDFS
质量平台
运维
监控
...

Blink 特点

低延时

弹性扩容

正确
处理乱序

快速容错

高性能

支持
大数据量

什么是BlinkSQL

Apache FlinkSQL

+

Alibaba Group 阿里巴巴集团

Alibaba's Improvements

=

Alibaba BlinkSQL

Why SQL

Declarative

Optimized

Understandable

Stable

SQL是经得起时间考验的标准...

Why SQL

流 vs 批

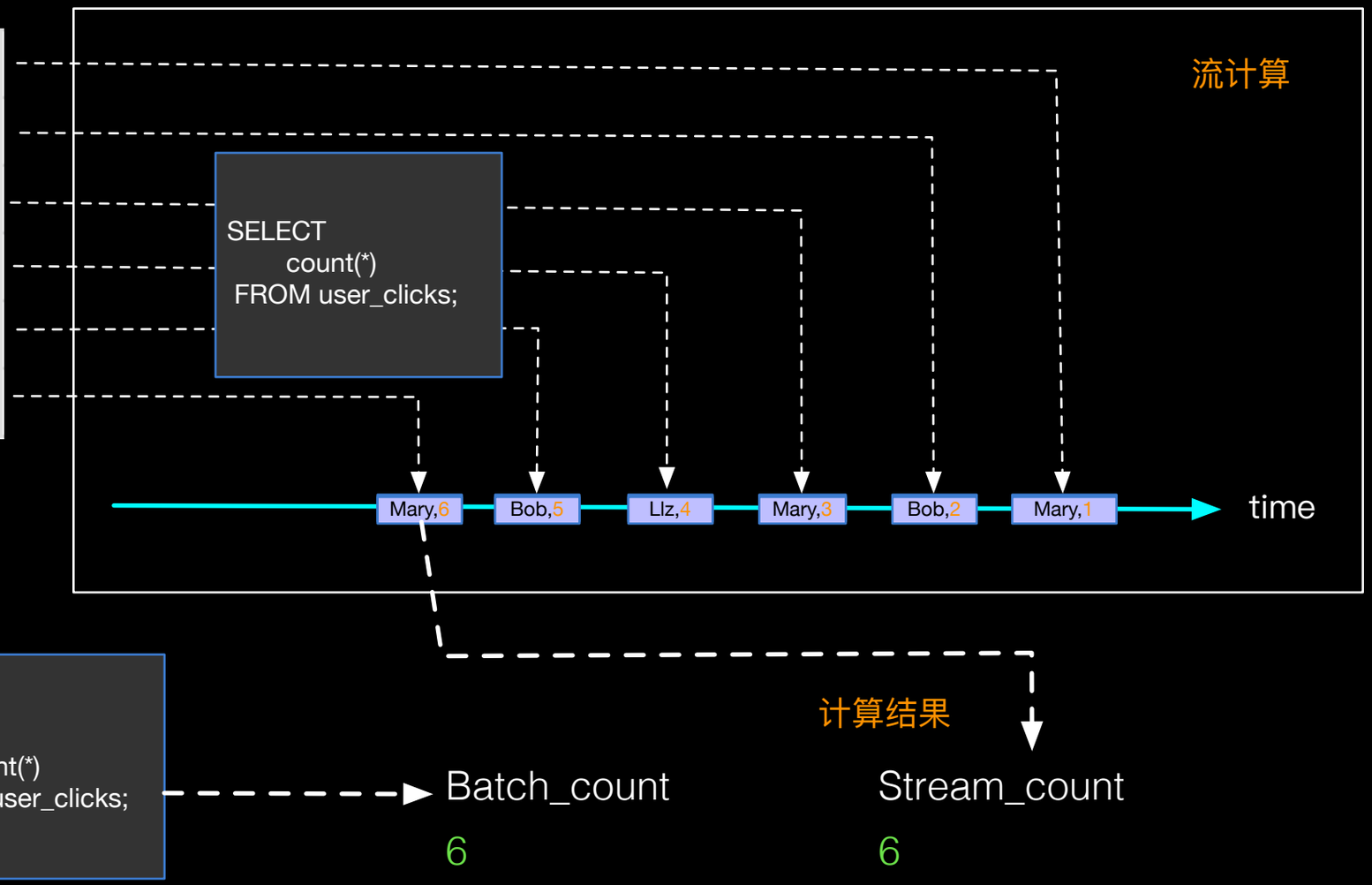
流处理	批处理
数据集合是无穷的	数据集合是有限的
流计算不断 产生结果且不结束	批计算返回 一个结果并结束
需要对历史结果 进行修正	

SQL 真的适用于流计算场景吗?

Why SQL

timestamp user clicks

2018/4/30 22:36:53	Mary
2018/4/30 22:37:6	Bob
2018/4/30 22:37:15	Mary
2018/4/30 22:37:27	Llz
2018/4/30 22:37:36	Bob
2018/4/30 22:37:45	Mary



Why SQL

2018/4/30 22:36:53	Mary
2018/4/30 22:37:6	Bob
2018/4/30 22:37:15	Mary
2018/4/30 22:37:27	Llz
2018/4/30 22:37:36	Bob
2018/4/30 22:37:45	Mary

流 vs 批

流计算

Llz,4 Mary,3 Bob,2 Mary,1 time

Unify

批计算

```
count()
FROM order_tab;
```

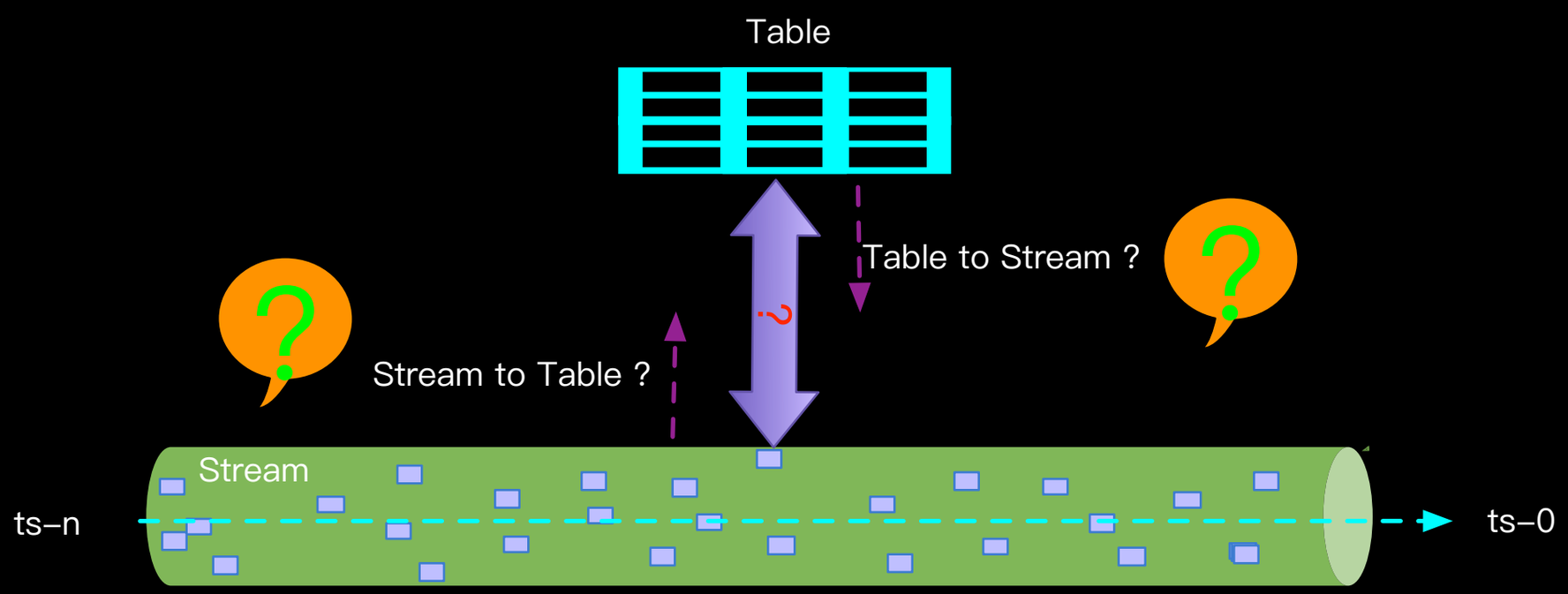
Batch_count	Stream_count
6	6

One Query One Result – 相同语义，最终结果一致!!!

目录

流表对偶性

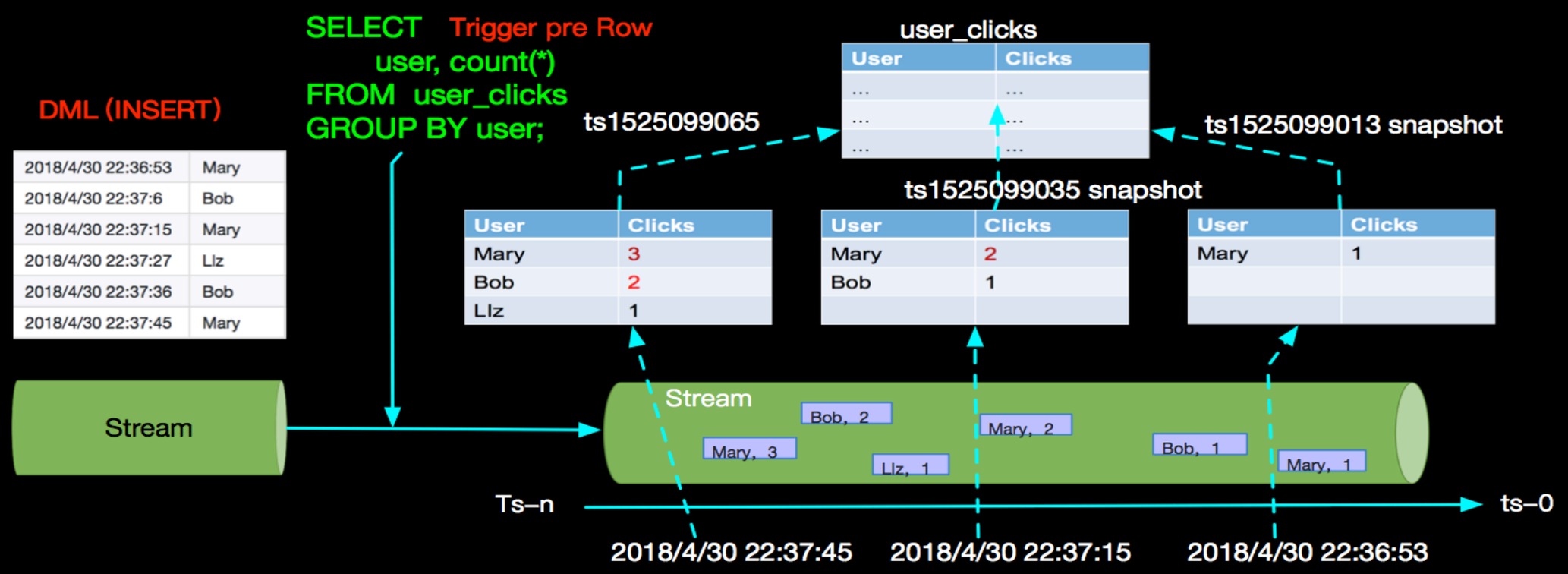
- 流表对偶性 – 流表转换信息无损，具有对偶性



流表对偶性

表特征： Schema，Data，？

流特征： Schema，Data，Time

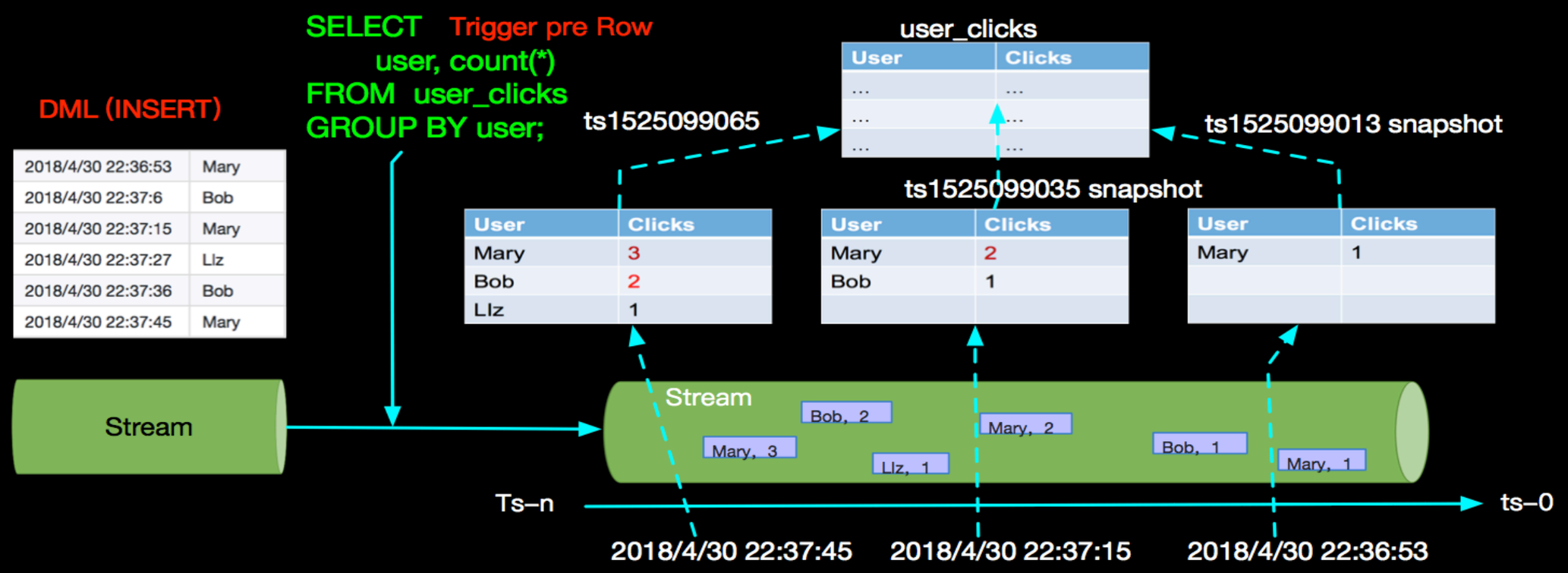


流表对偶性

表特征： Schema，Data，DML操作时间

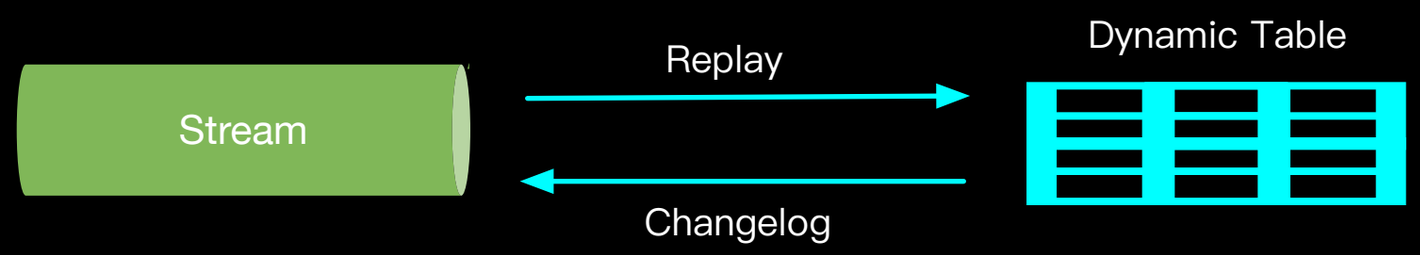
流特征： Schema，Data，Time

流表等价



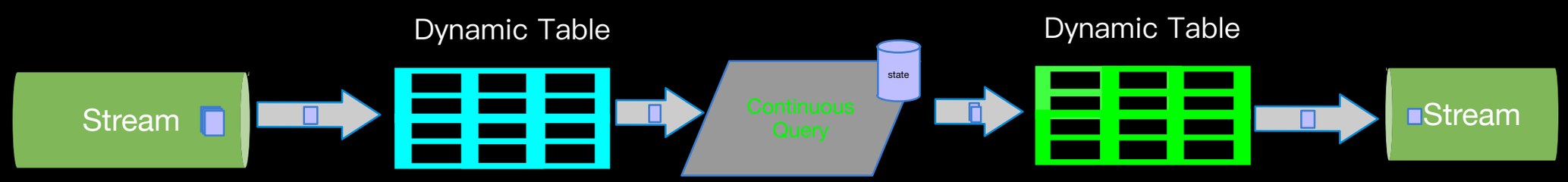
动态表(Dynamic Table)

- 动态表(Dynamic Table)– 随着时间变化不断变化的表



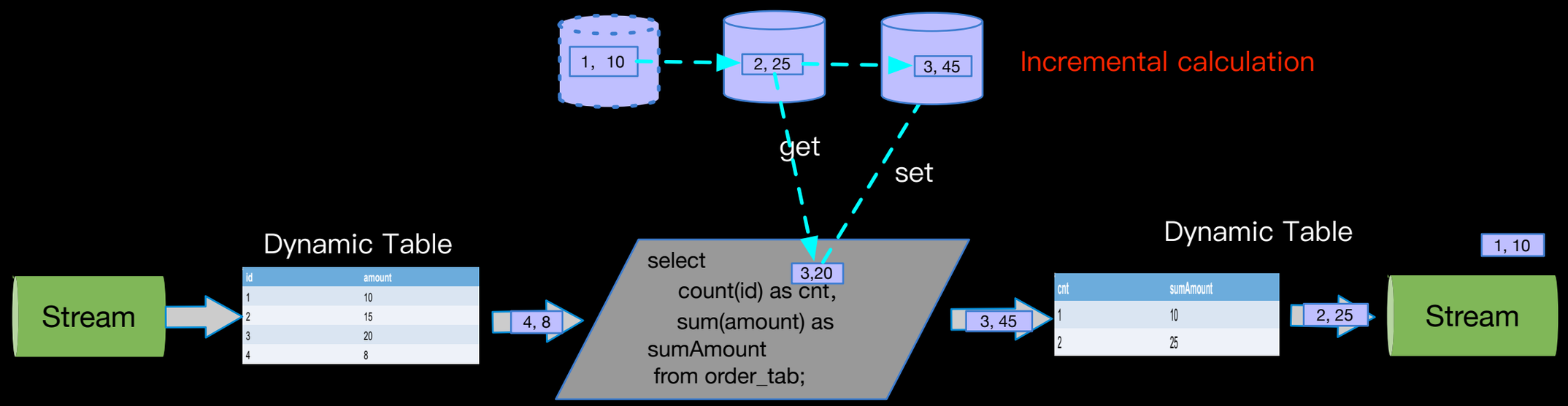
持续查询(Continuous Query)

- 查询查询(Continuous Query)– 流计算特有的，不断更新结果，持续运行的查询



增量计算(Incremental calculation)

- 增量计算(Incremental Calculation) – 一种可以提高流计算性能的计算实现



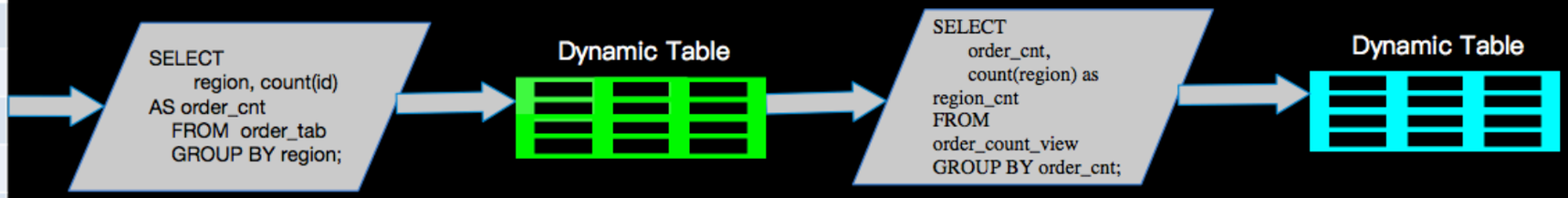
Early Emit & Retraction

- Early Emit – 为保障流计算的低延时需求，将计算结果尽早发出
- Retraction – 为保障流计算语义正确性，将Early Emit的计算结果撤回并发出正确的计算结果

```sql
-- 按订单数量分组统计具有相同订单数量的地区数量
SELECT
    order_cnt,
    count(region) as region_cnt
FROM (
    -- 按地区分组计算每个地区的订单数量
    SELECT
        region, count(id) AS order_cnt
    FROM  order_tab
    GROUP BY region
)
GROUP BY order_cnt;
```

order_tab

id	region
1	SH
2	BJ
3	SH
4	SZ
5	BJ
6	BJ



Early Emit & Retraction

order_tab

id	region
1	SH
2	BJ
3	SH
4	SZ
5	BJ
6	BJ

```
SELECT
    region, count(id) AS
order_cnt
  FROM  order_tab
  GROUP BY region;
```

BJ, 3 | BJ, 2 | SZ, 1 | SH, 2 | BJ, 1 | SH, 1

order_cnt_tab

region	order_cnt
SH	1
BJ	1
SH	2
SZ	1
BJ	2
BJ	3

```
SELECT
    order_cnt,
    count(region) as  region_cnt
FROM order_count_view
GROUP BY order_cnt;
```

3, 1 | 2, 2 | 1, 3 | 2, 1 | 1, 2 | 1, 1

region_cnt_tab

order_cnt (pk)	region_cnt
1	3
2	2
3	1

without retraction

Early Emit & Retraction

order_tab

id	region
1	SH
2	BJ
3	SH
4	SZ
5	BJ
6	BJ

```
SELECT
    region, count(id) AS order_cnt
  FROM  order_tab
  GROUP BY region;
```

+(BJ, 3) -(BJ, 2) +(BJ, 2) -(BJ, 1) +(SZ, 1) +(SH, 2) -(SH, 1) +(BJ, 1) +(SH, 1)

with retraction

order_cnt_tab

region	order_cnt
+SH	1
+BJ	1
-SH	1
+SH	2
+SZ	1
-BJ	1
+BJ	2
-BJ	2
+BJ	3

```
SELECT
    order_cnt,
    count(region) as  region_cnt
FROM order_count_view
GROUP BY order_cnt;
```

3, 1 2, 1 2, 2 1,1 1, 2 2, 1 1,1 1, 2 1, 1

region_cnt_tab

order_cnt (pk)	region_cnt
1	1
2	1
3	1

为什么要介绍 五 个概念

流表对偶性	流表转化才有理论支撑，流表才能互转
动态表	有了动态表的概念，SQL才能作用于流上，才能进行持续查询
持续查询	有了持续查询概念，才能真正解释流上SQL计算永远不结束，结果永远在更新
增量计算	有了增量计算 + 查询优化+引擎优化，才有了Blink的高性能
Early Emit Retraction	有了Early Emit 才可以无限逼近秒级毫秒级的低延时 有了Retraction才可以保证流上语义的正确性

高性能

快速容错

低延时

数据乱序

BlinkSQL功能概览

DML & DDL & Query Conf

UDF/UDTF/UDAF

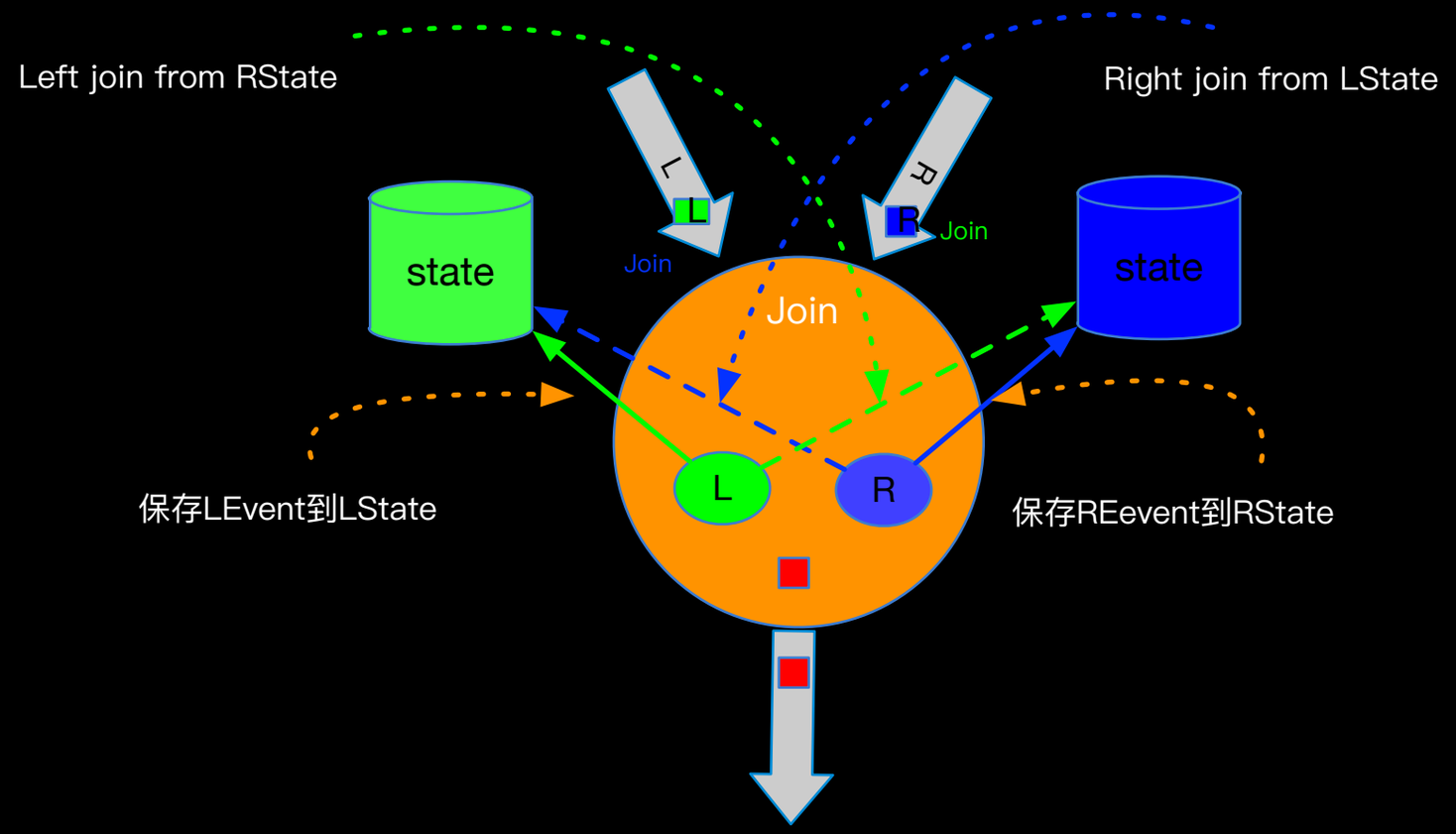
JOIN

Retraction

Window AGG

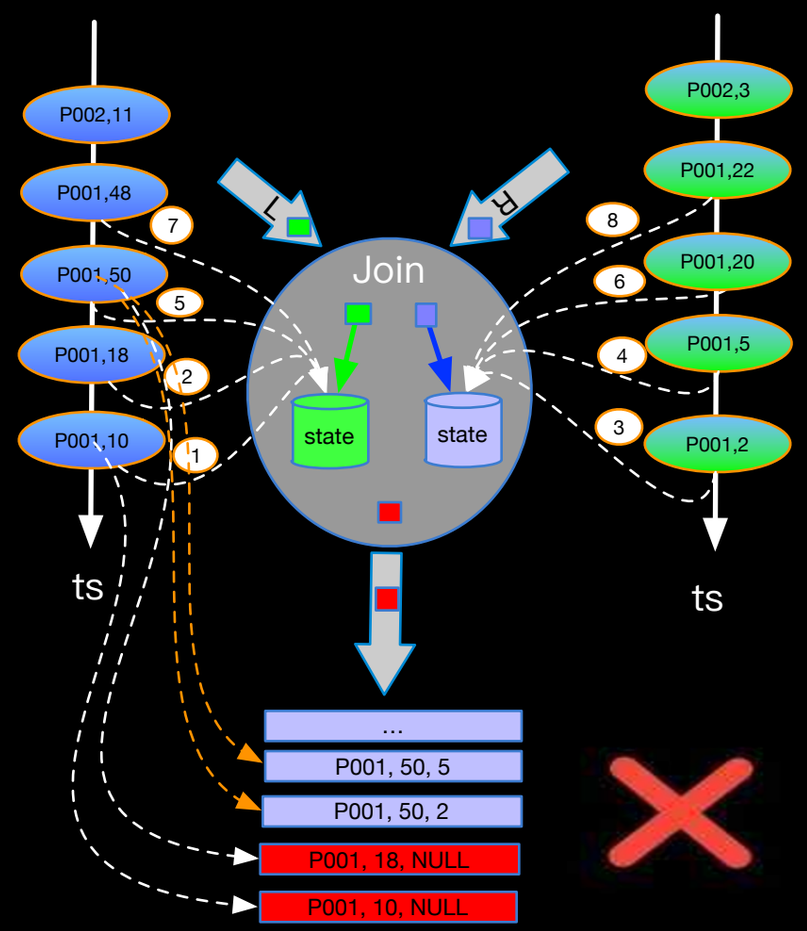
Query Optimization

双流JOIN实现


Left join from RState
Right join from LState
L
R
L
R
Join
Join
state
state
Join
L
R
保存LEvent到LState
保存REevent到RState

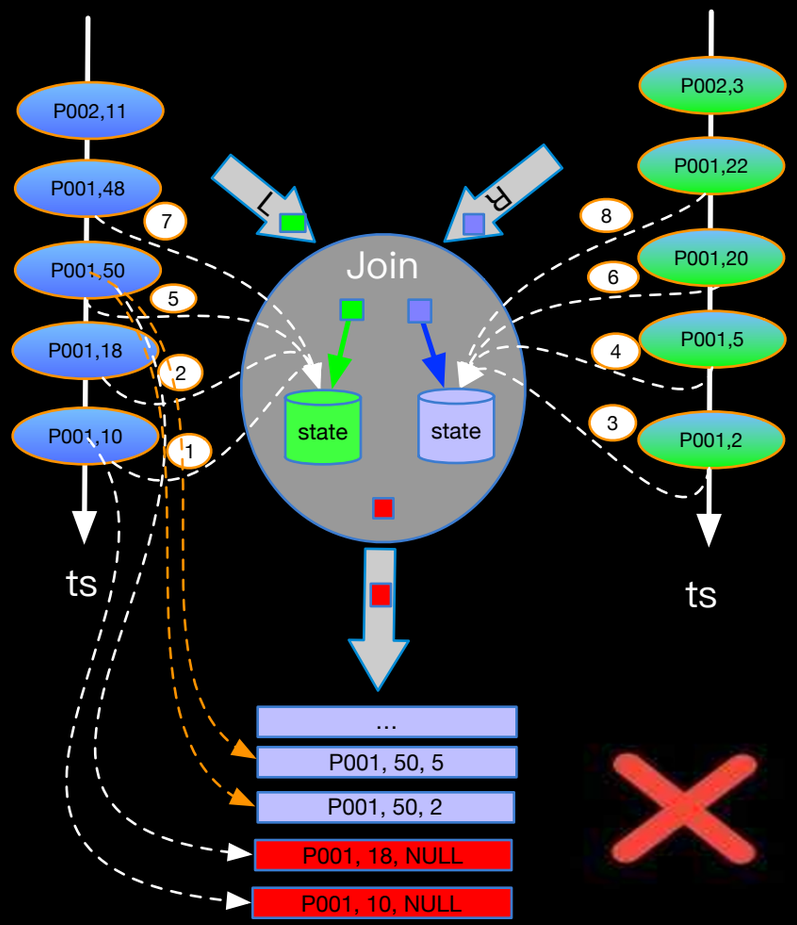
双流JOIN实现

```
SELECT
    product_id, product_count, sales_count
FROM inventory_tab I LEFT JOIN sales_tab s ON I.product_id = s.product_id;
```

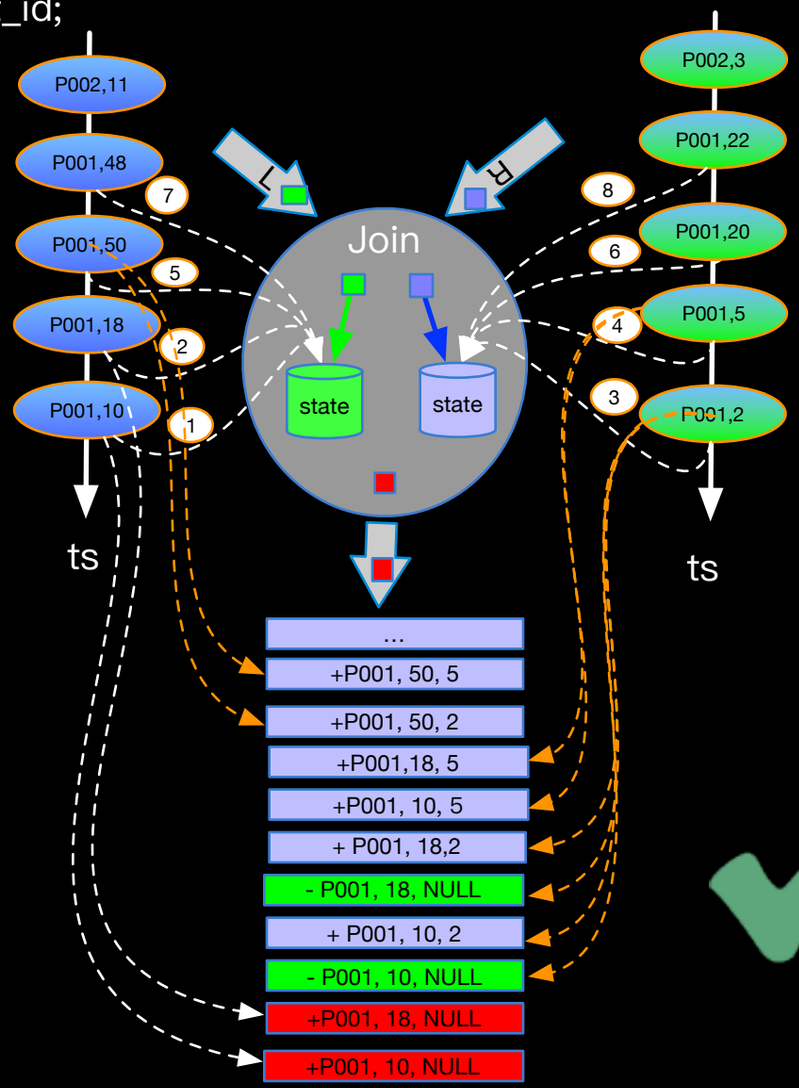


双流JOIN实现

```
SELECT
    product_id, product_count, sales_count
FROM inventory_tab I LEFT JOIN sales_tab s ON I.product_id = s.product_id;
```

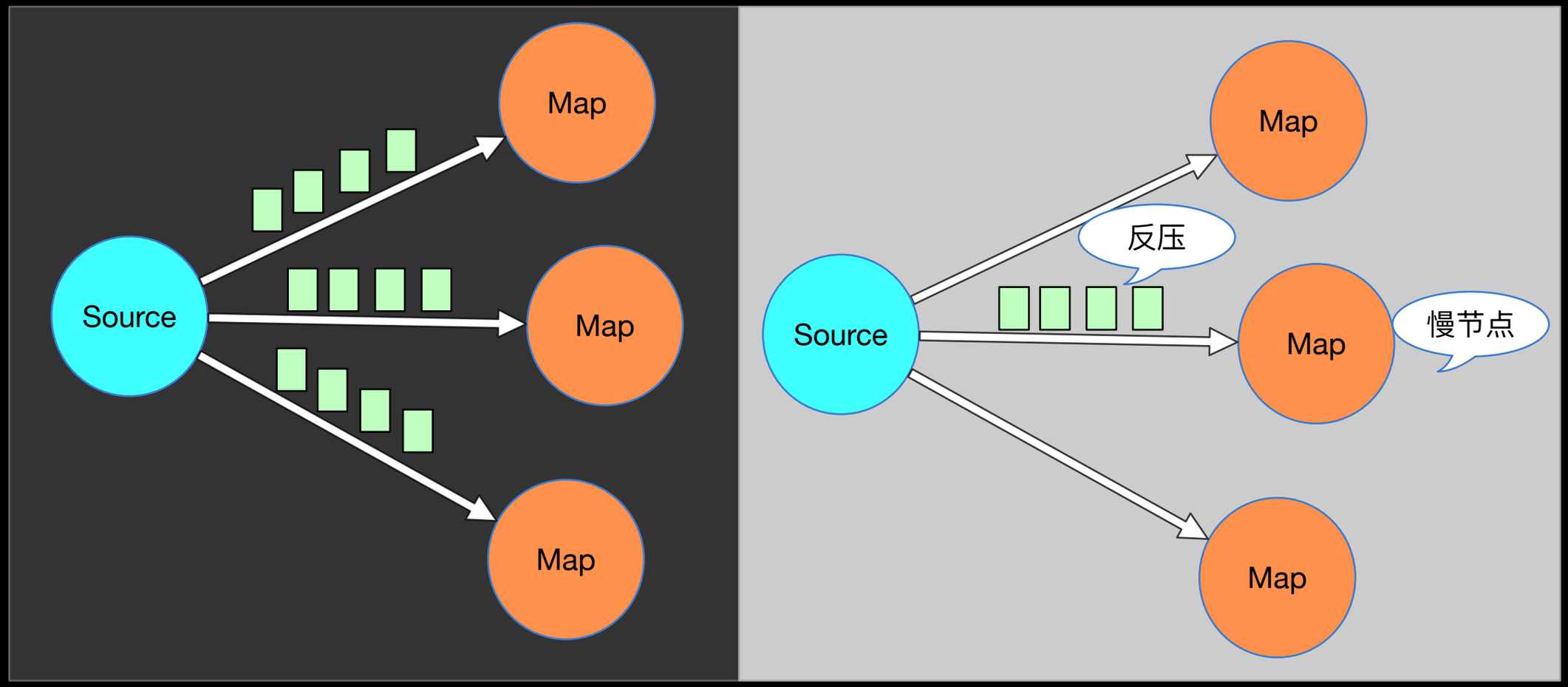


Retract机制



目录

案例1–无Key热点



均衡处理

不均衡处理

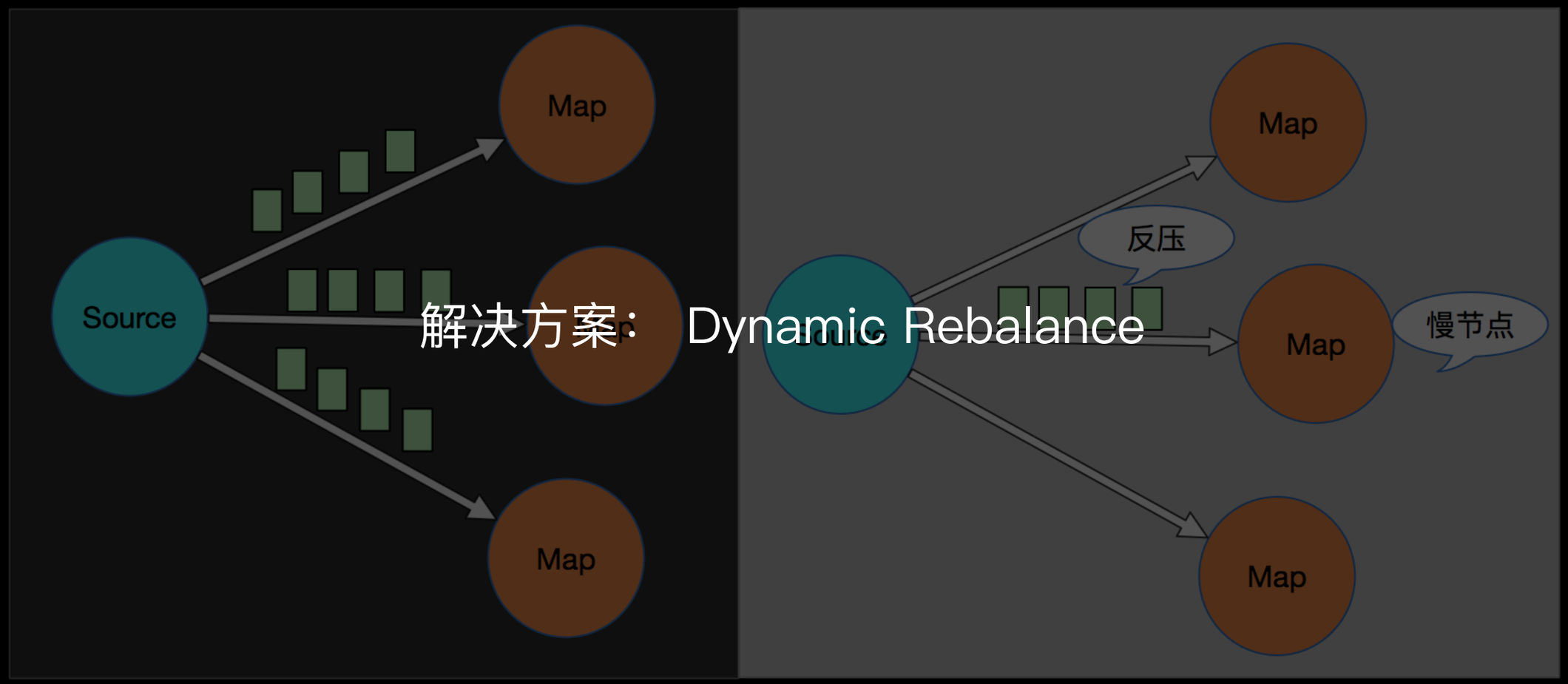
案例1–无Key热点

Blink WebMonitor

OverView | Metrics | Accumulators

▲ID	StartTime	StopTime	Duration(s)	InQueue	OutQueue	Latency(ms)	RecvCnt	SendCnt	TPS	Retries	Ho
global search ...											
0	2018-01-26 16:13:08	-	1,999.983	399 (100.00%)	0 (0.00%)	2,070.00	10,942,801	1,094	5,265	0	hadoop5047.e
1	2018-01-26 16:13:08	-	1,999.98	0 (0.00%)	0 (0.00%)	548.00	10,946,889	1,094	5,313	0	hadoop5047.e
2	2018-01-26 16:13:08	-	1,999.979	399 (100.00%)	0 (0.00%)	1,923.00	10,942,501	1,094	5,265	0	hadoop5047.e
3	2018-01-26 16:13:08	-	1,999.978	0 (0.00%)	0 (0.00%)	762.00	10,946,889	1,094	5,313	0	hadoop5047.e
4	2018-01-26 16:13:08	-	1,999.978	399 (100.00%)	0 (0.00%)	1,962.00	10,942,501	1,094	5,260	0	hadoop5047.e
5	2018-01-26 16:13:08	-	1,999.977	0 (0.00%)	0 (0.00%)	594.00	10,946,889	1,094	5,313	0	hadoop5047.e
6	2018-01-26 16:13:08	-	1,999.977	399 (100.00%)	0 (0.00%)	1,760.00	10,942,801	1,094	5,265	0	hadoop5047.e
7	2018-01-26 16:13:08	-	1,999.977	0 (0.00%)	0 (0.00%)	665.00	10,946,888	1,094	5,313	0	hadoop5047.e
8	2018-01-26 16:13:08	-	1,999.976	399 (100.00%)	0 (0.00%)	1,725.00	10,943,101	1,094	5,265	0	hadoop5047.e
9	2018-01-26 16:13:08	-	1,999.976	0 (0.00%)	0 (0.00%)	512.00	10,946,888	1,094	5,313	0	hadoop5047.e

案例1–无Key热点

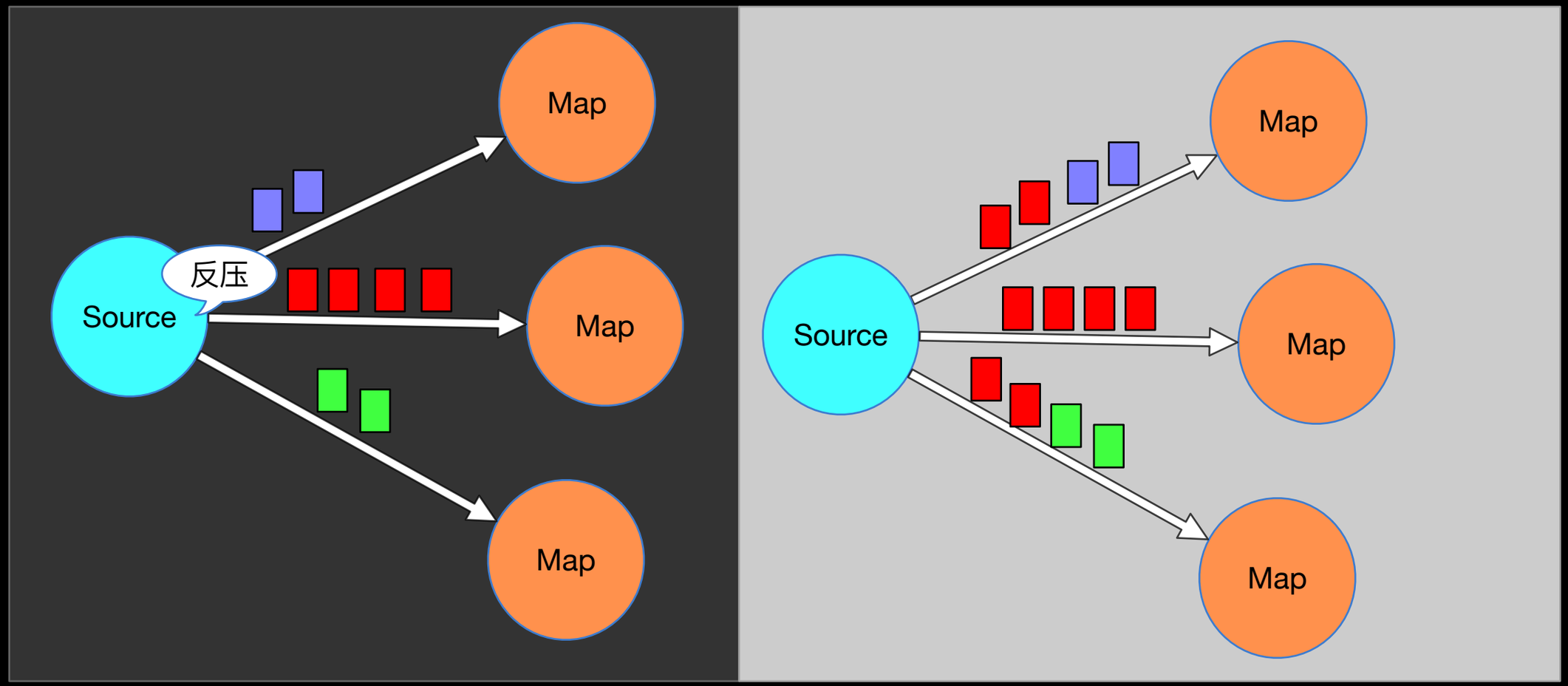


解决方案： Dynamic Rebalance

均衡处理

不均衡处理

案例1–无Key热点

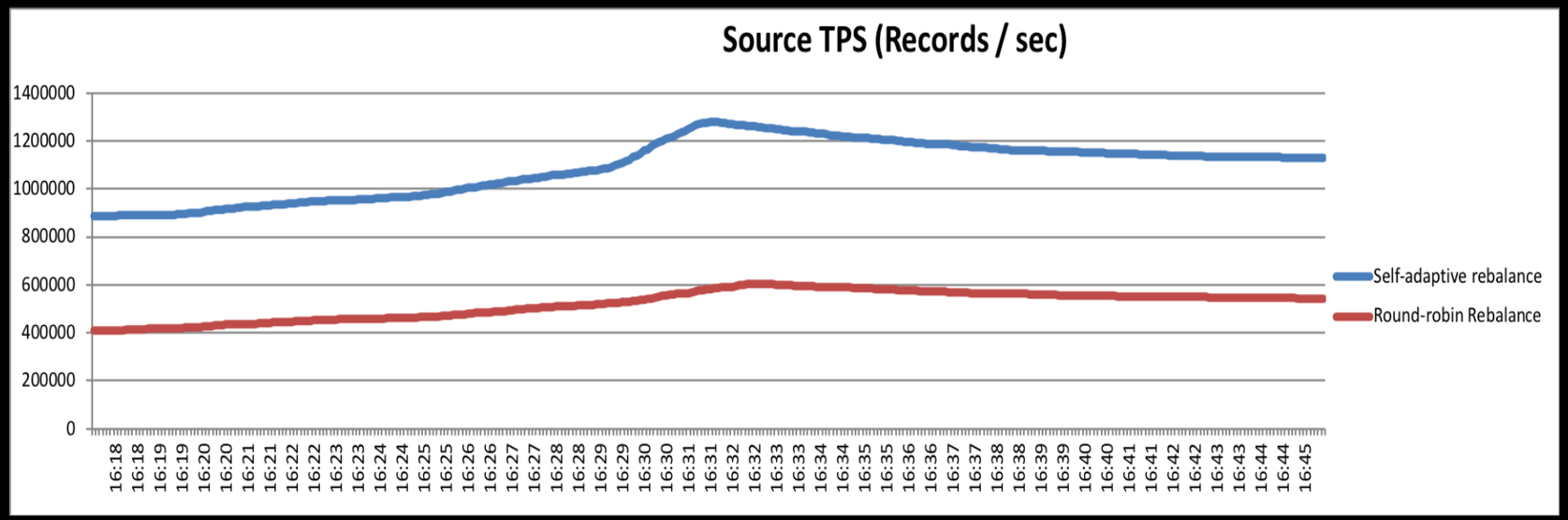


Round Robin

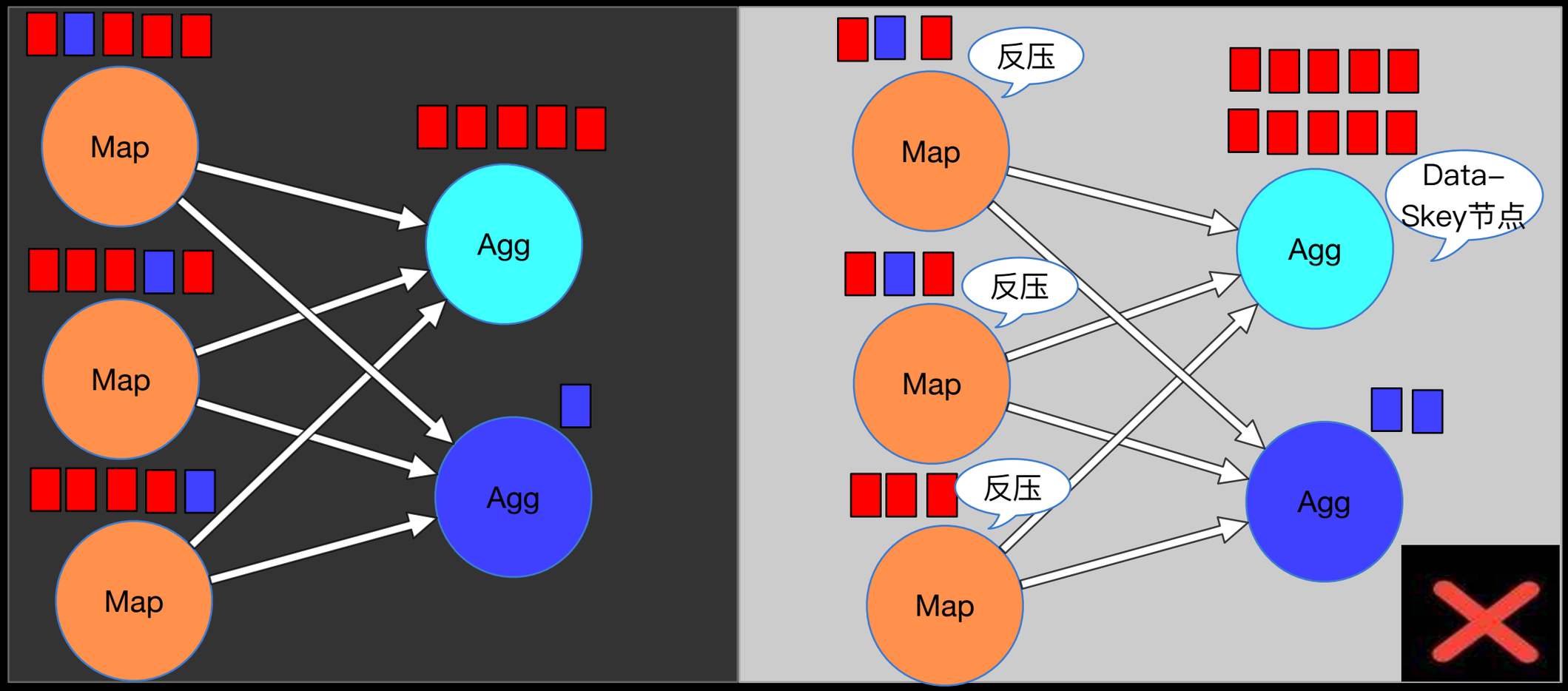
Dynamic Rebalance

案例1–无Key热点

Dynamic Rebalance带来2x的性能提升

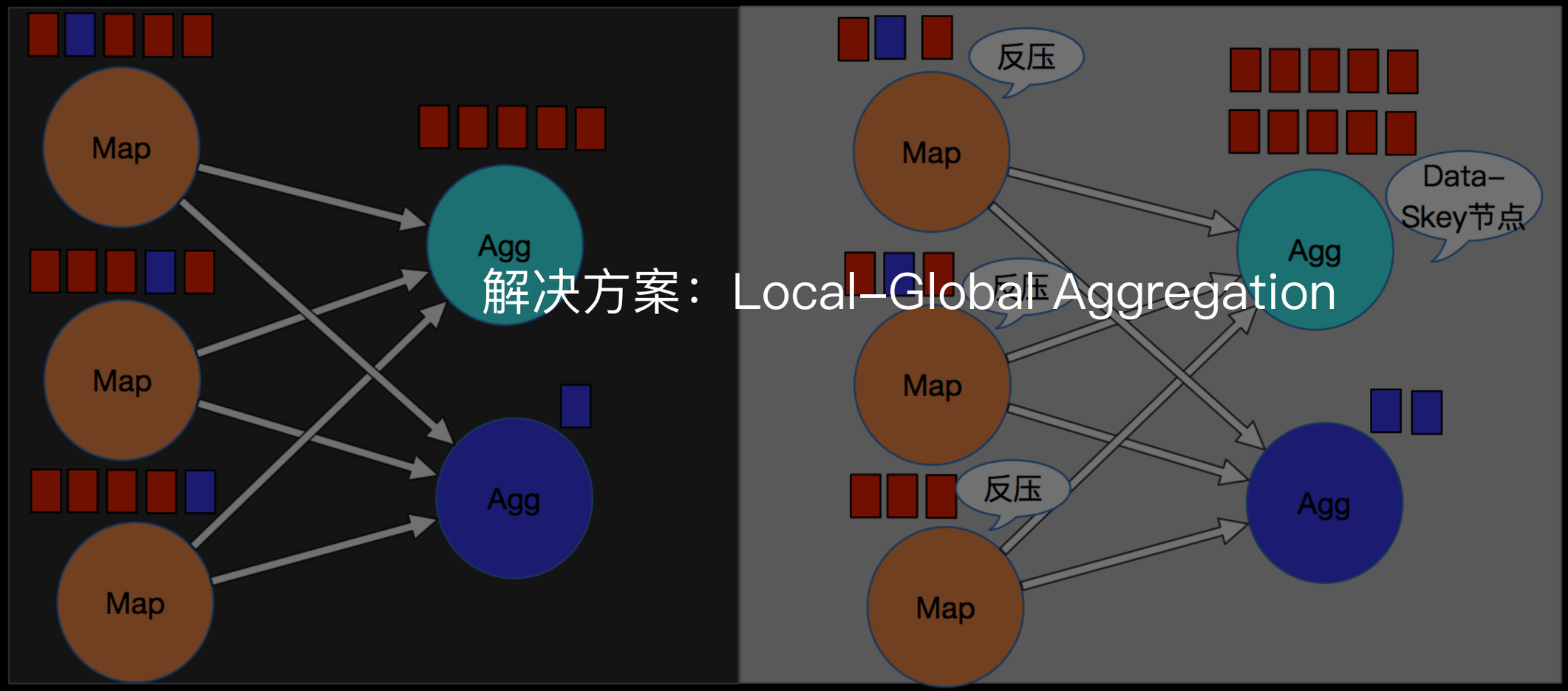


案例2–Keyed热点



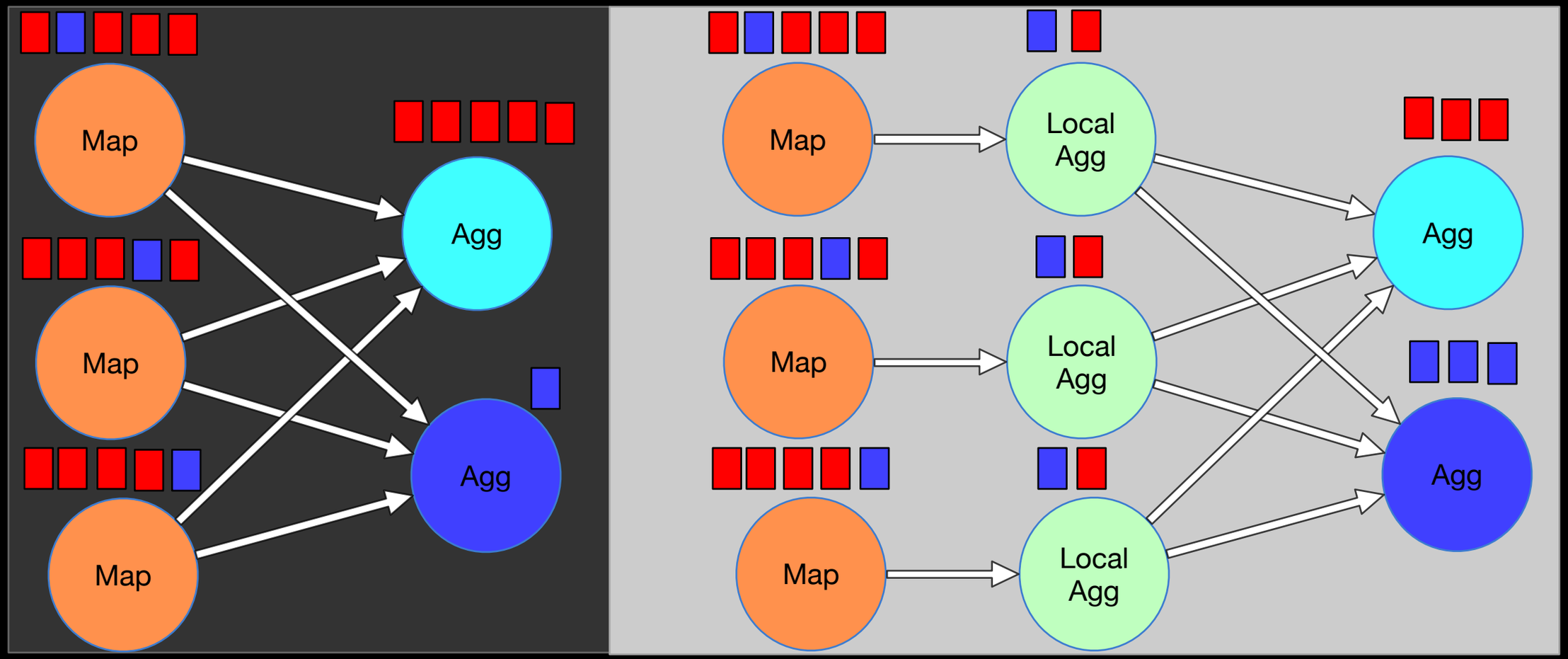
Keyed数据倾斜问题，不能用Dynamic Rebalance解决...

案例2–Keyed热点



解决方案：Local–Global Aggregation

案例2–Keyed热点



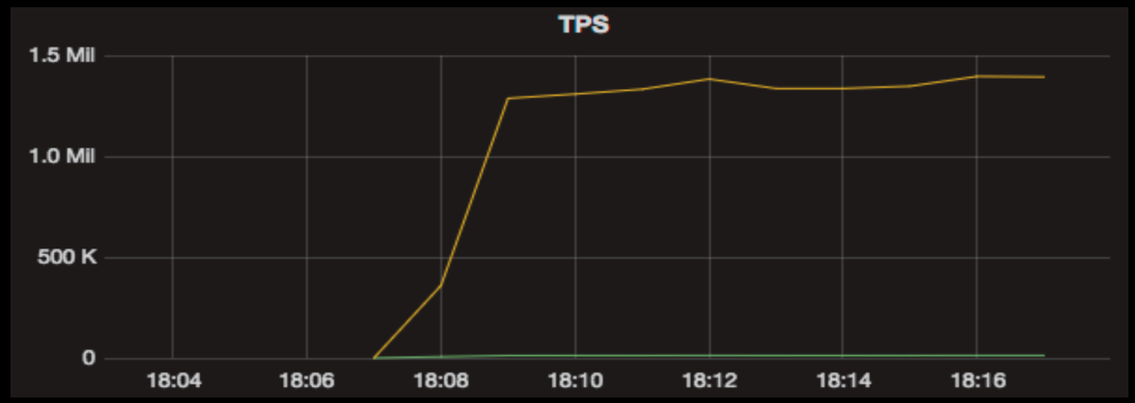
Simple Aggregation

Local–Global Aggregation

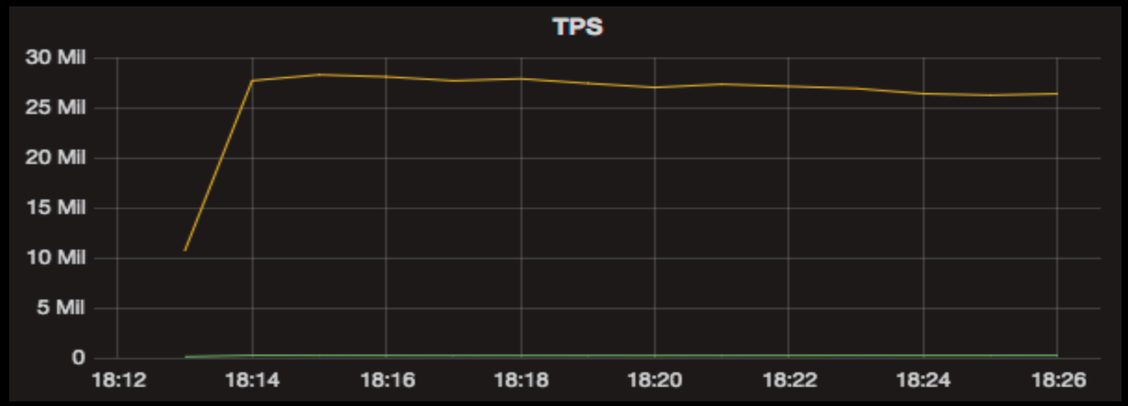
案例2–Keyed热点

Local–Global Agg带来20x的性能提升

Simple Aggregation



Local–Global Aggregation

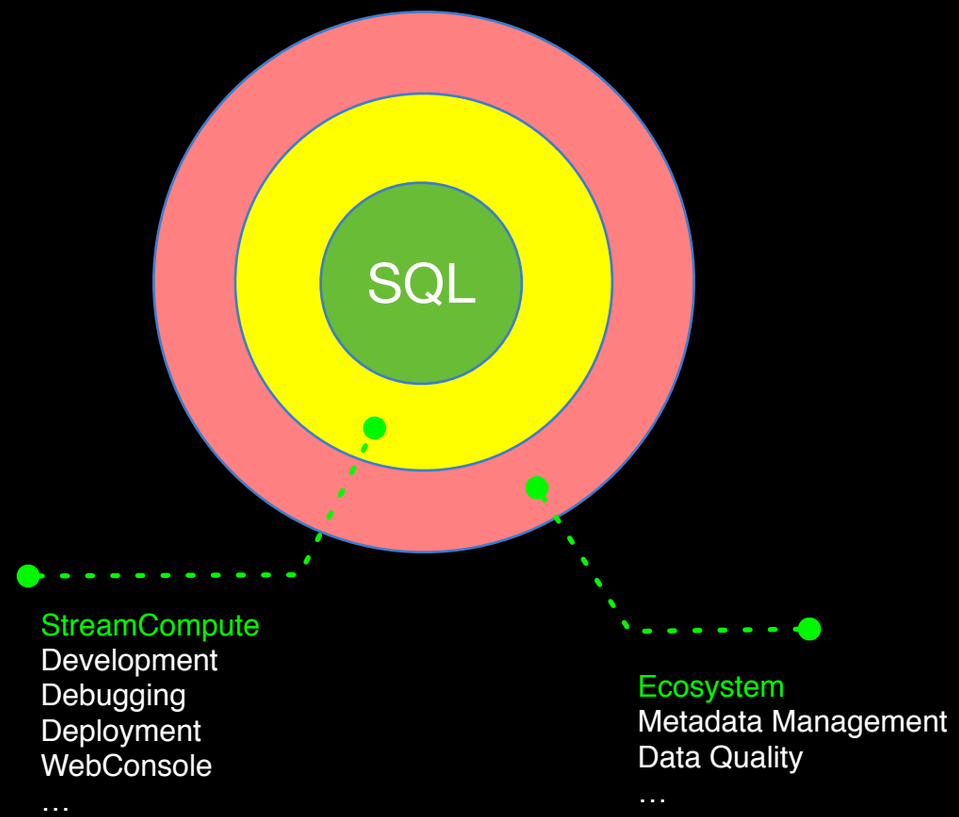


Blink Provide More

Just SQL ?

SQL Only Solves Part of the Problem

We Provide More!



总结

Blink SQL

标准的语义

Blink SQL 支持标准的ANSI SQL，批和流计算使用一套统一的SQL

丰富的功能

支持DDL, DML, Retraction, 内置常用Connector, UDC，4种Group Window，4种Over Window，双流JOIN，Temporal Table JOIN, UDF, UDTF, UDAF 等等

更优的性能

在支持标准ANSI SQL的各个语法功能的基础上，StreamCompute SQL做了大量的性能优化

一站式平台

StreamCompute 集开发，测试，部署，监控，升级于一体，让用户只专注自己的业务

Thanks & QA

2018.6.15