SDCC 2017深圳站
大数据技术实战峰会

CSDN

StreamSql实时计算平台的挑战及解决方案

周建军

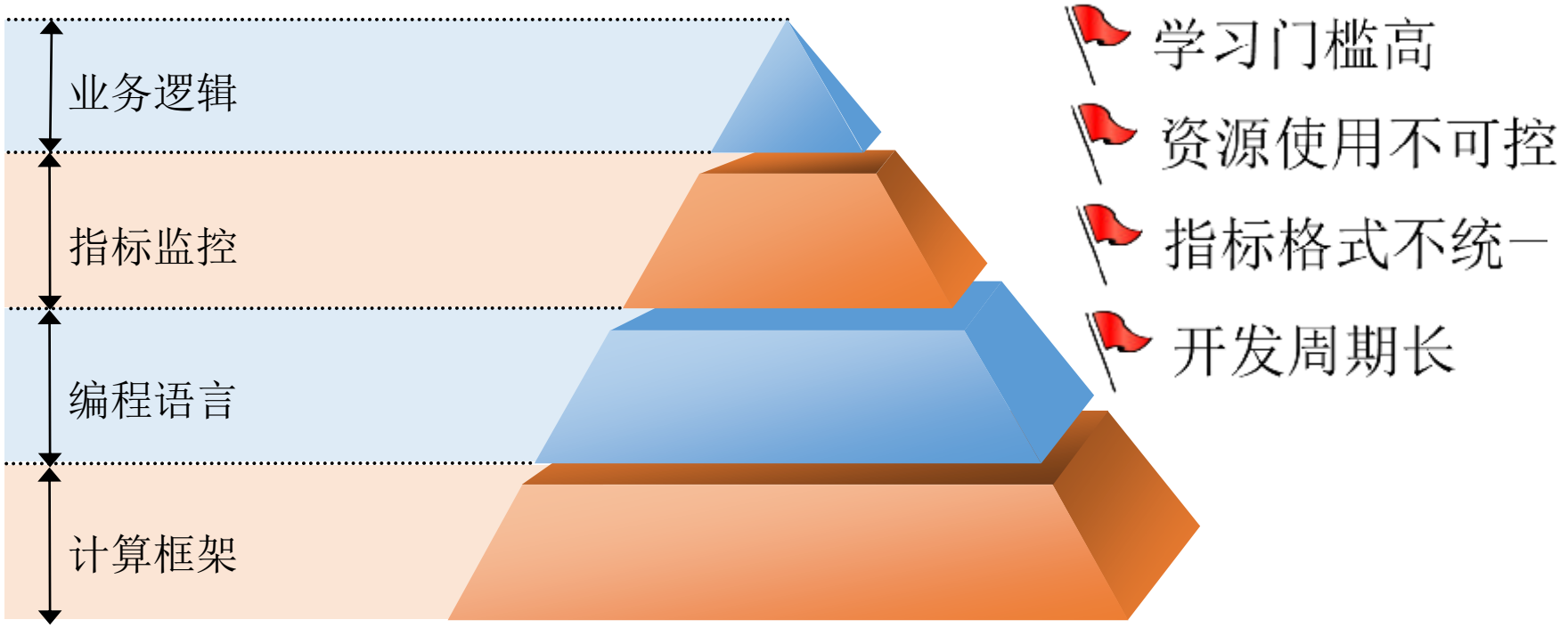
2017-6

目录
CONTENTS

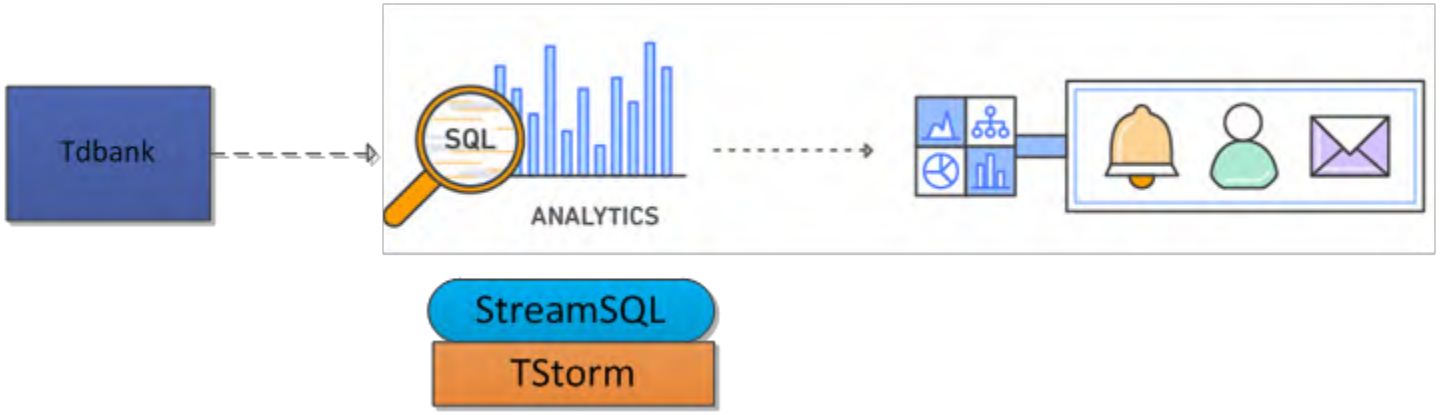
目录
CONTENTS


历史战绩
结果
击败
死亡
助攻
装备
时间
胜利
5 / 3 / 4
排位赛
05-28 14:59
失败
0 / 7 / 4
排位赛
05-28 14:41
MVP
失败
8 / 3 / 5
排位赛
05-28 14:23
失败
0 / 10 / 4
排位赛
05-28 14:06

传统实时计算任务开发过程



目录
CONTENTS



practice online

计算规模：万亿级别
计算延迟：秒级

StreamSQL平台易用性

1. 类SQL逻辑描述、学习成本低
2. 指标种类丰富、格式统一

StreamSQL平台稳定性

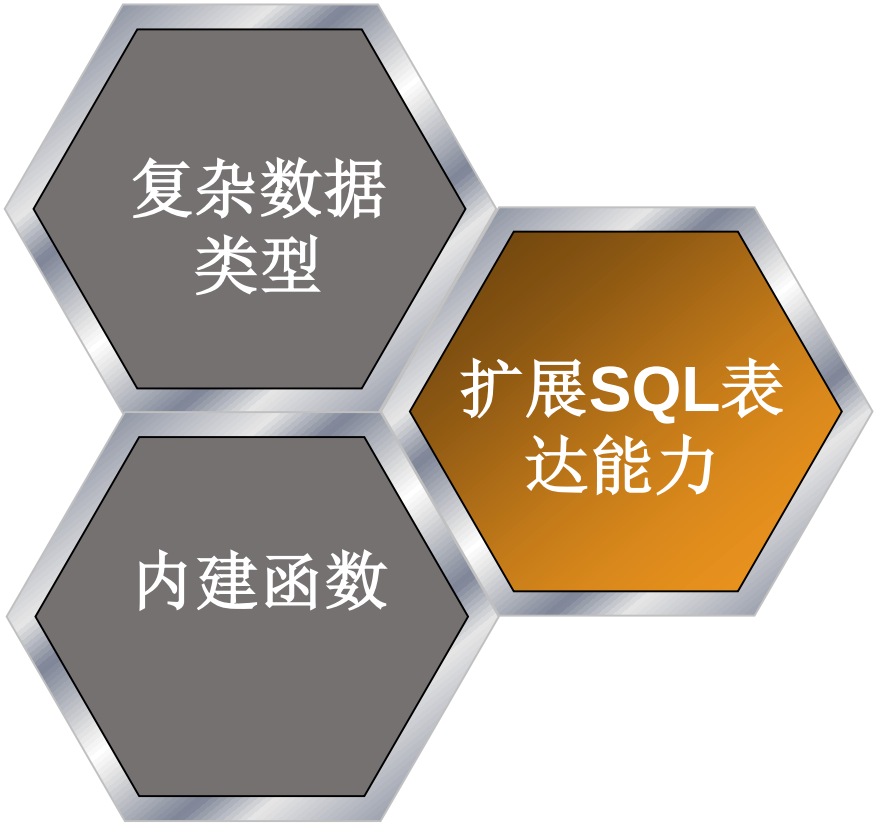
资源不足

过载保护

外部环境

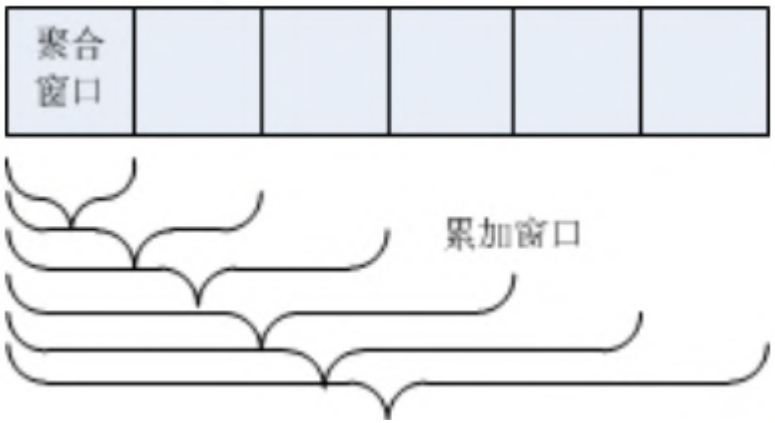
StreamSQL平台计算功能

StreamSQL平台功能强大

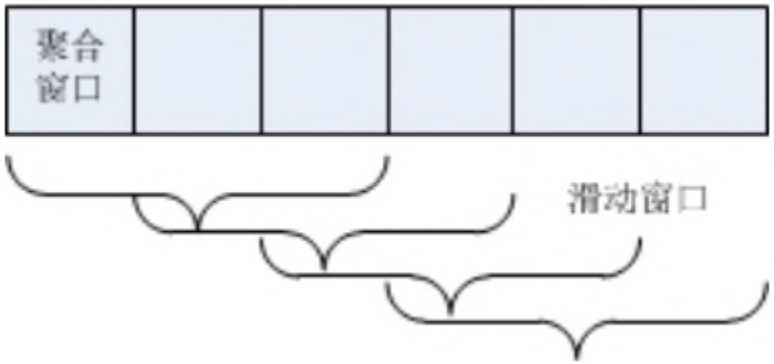


扩展SQL表达能力-窗口设计

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



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



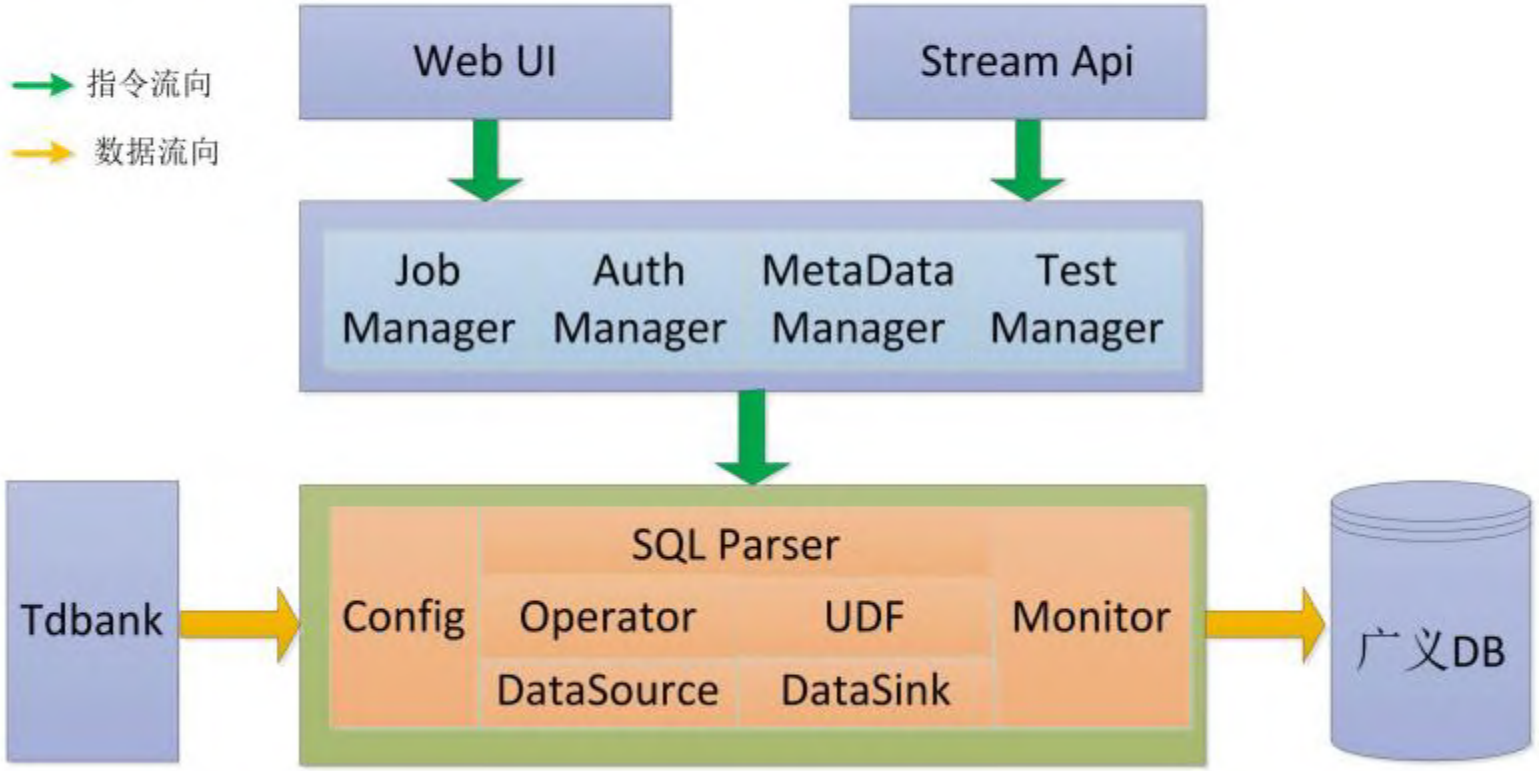
INSERT INTO dest SELECT appId count(qq SW))
FROM src GROUP BY appId COORDINATE BY
dTime WITH AGGR INTERVAL 60SECONDS WITH
SW INTERVAL 180 SECONDS

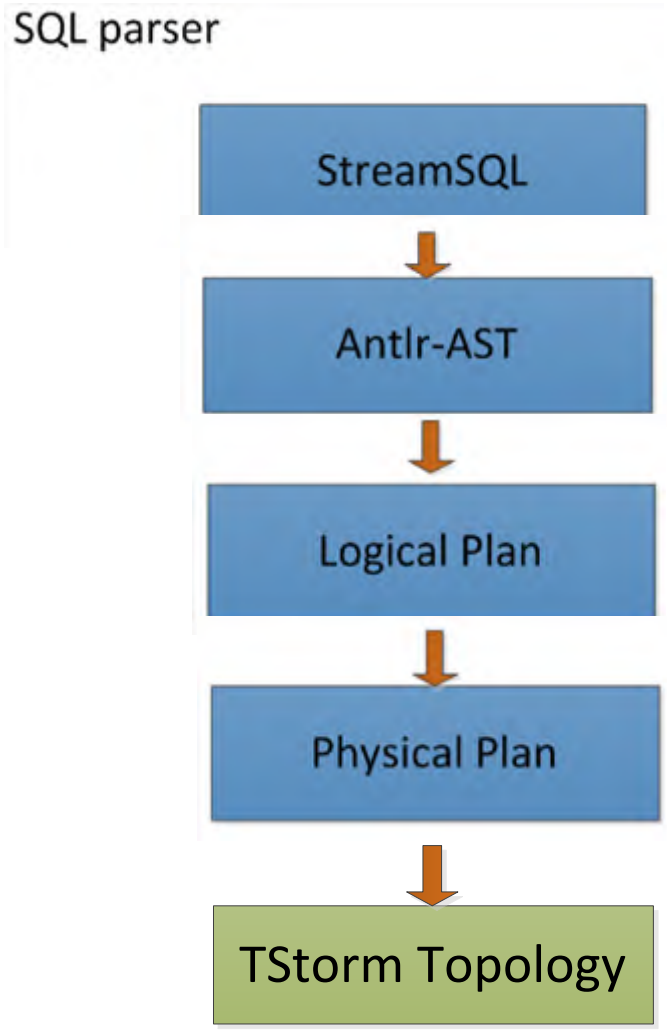
扩展SQL表达能力-表达式

For Each

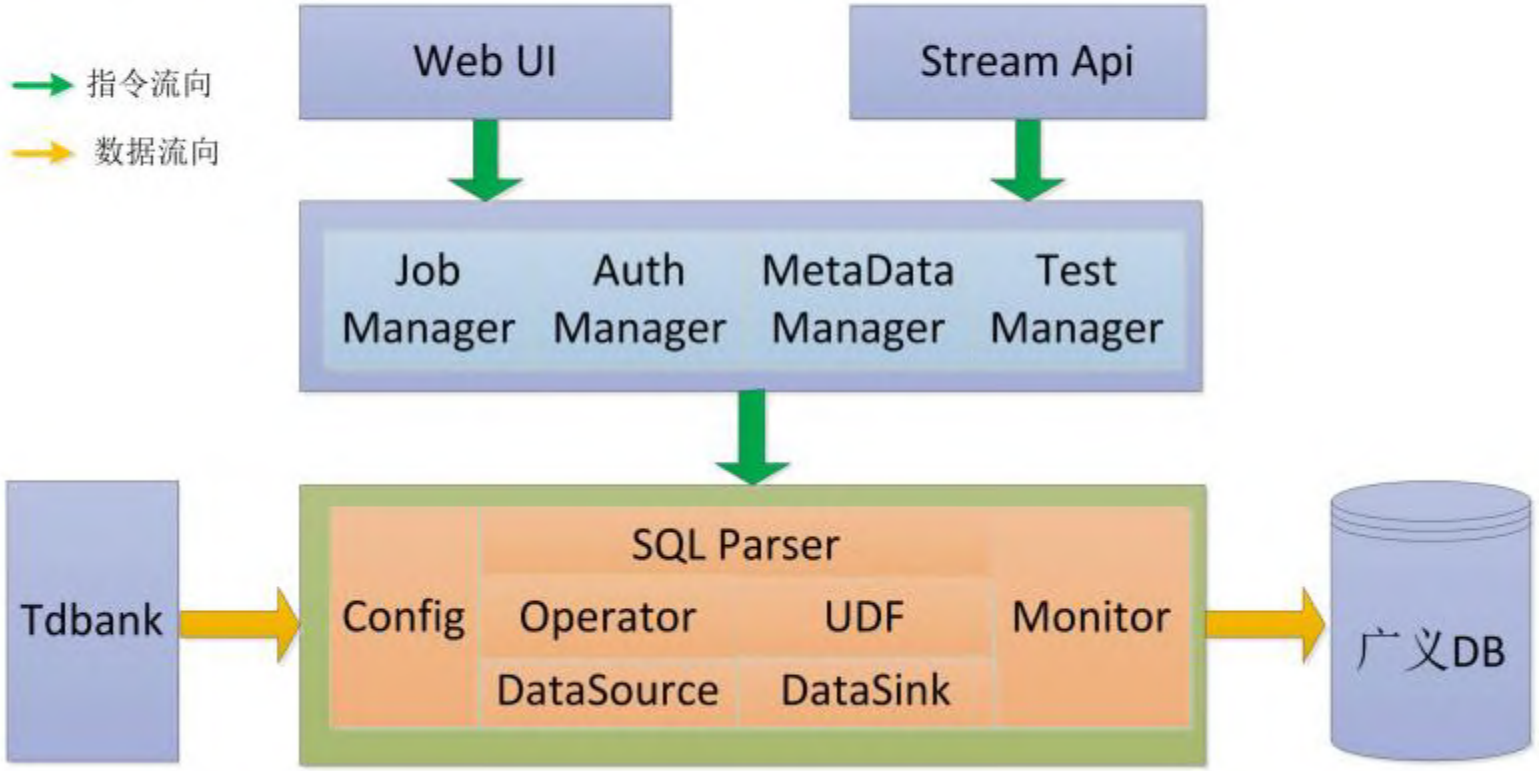
Execute

平台架构


指令流向
数据流向
Web UI
Stream Api
Job Manager
Auth Manager
MetaData Manager
Test Manager
Tdbank
Config
SQL Parser
Operator
UDF
DataSource
DataSink
Monitor
广义DB


SQL parser
StreamSQL
Antlr-AST
Logical Plan
Physical Plan
TStorm Topology

平台架构


指令流向
数据流向
Web UI
Stream Api
Job Manager
Auth Manager
MetaData Manager
Test Manager
Tdbank
Config
SQL Parser
Operator
UDF
DataSource
DataSink
Monitor
广义DB

设计理念：一切皆是表

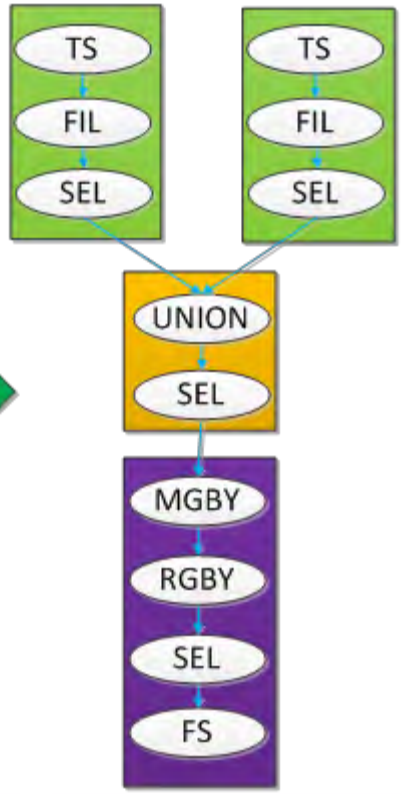
目录
CONTENTS

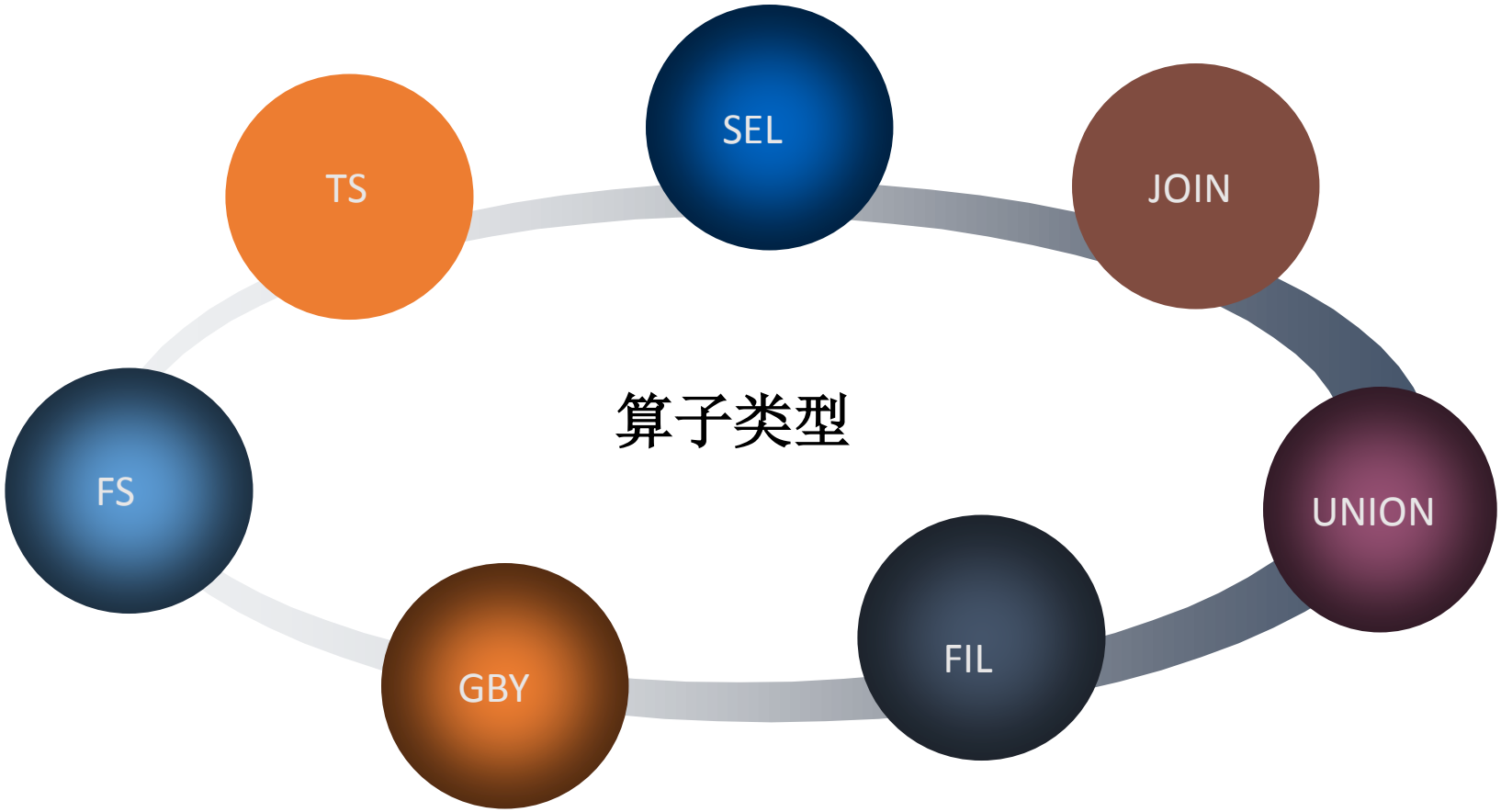
StreamSql实时计算平台的挑战-SQL转化

```
with(select "ylzt" name, qq from tbl_ylzt where flag=0) ylzt,
     (select "lol" name, qq from tbl_lol where flag=0) lol ,
(select name, qq from  ylzt union all select name, qq from lol) utbl
insert into result select name, count(qq) from utbl group by name
  coordinate by unix_timestamp() with aggr interval 60 seconds
```

语法树

算子树

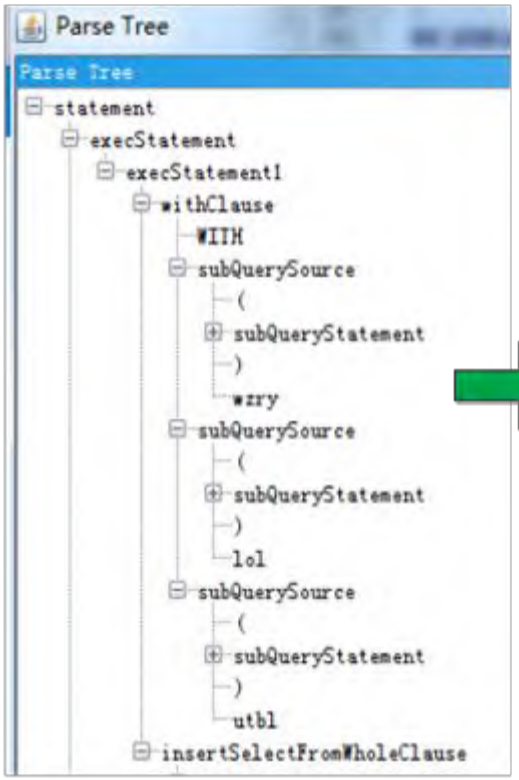



SEL
TS
JOIN
算子类型
FS
UNION
GBY
FIL

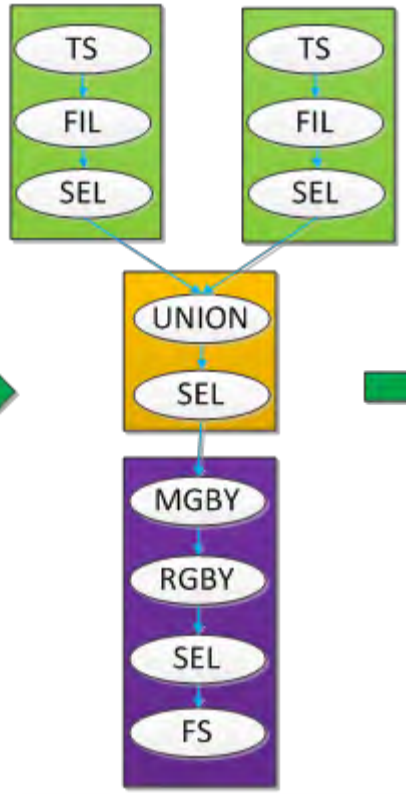
StreamSql实时计算平台的挑战-SQL转化

```
with(select "ylzt" name, qq from tbl_ylzt where flag=0) ylzt,
     (select "lol" name, qq from tbl_lol where flag=0) lol ,
(select name, qq from  ylzt union all select name, qq from lol) utbl
insert into result select name, count(qq) from utbl group by name
  coordinate by unix_timestamp() with aggr interval 60 seconds
```

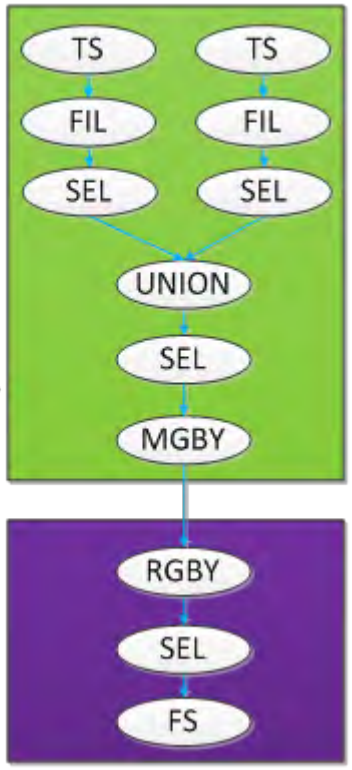
语法树



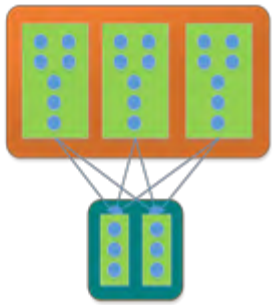
算子树



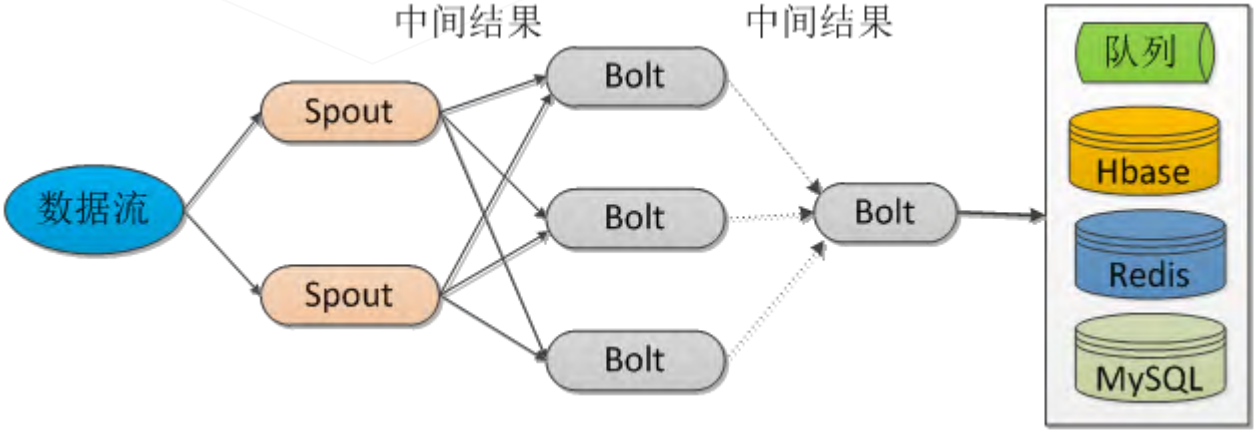
任务树



执行拓扑

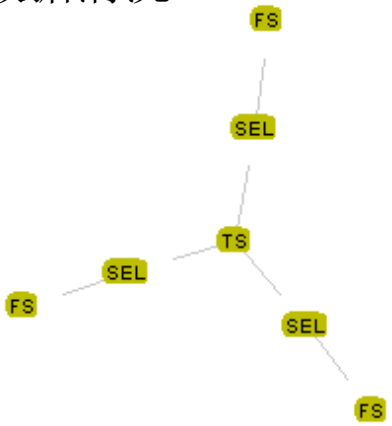


划分原则：尽量减少拓扑的长度

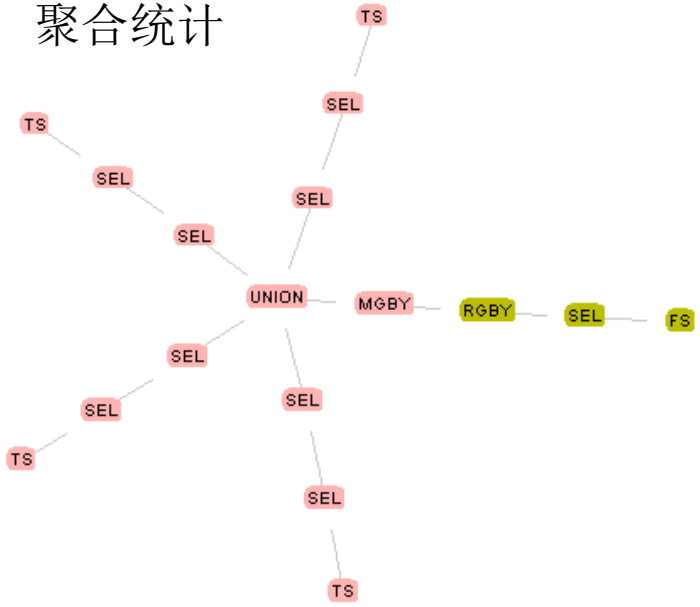


1.实现简单
2.减少网络传输开销
3.减少数据的序列化与反序列化成本开销

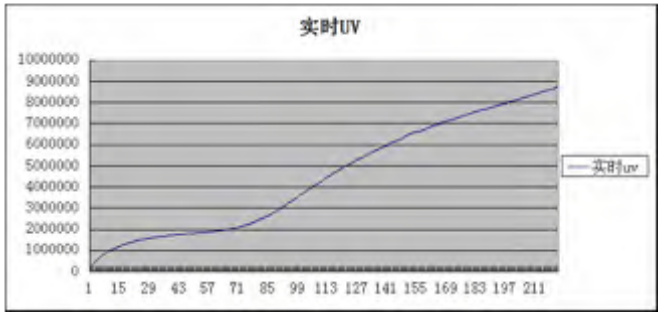
数据清洗

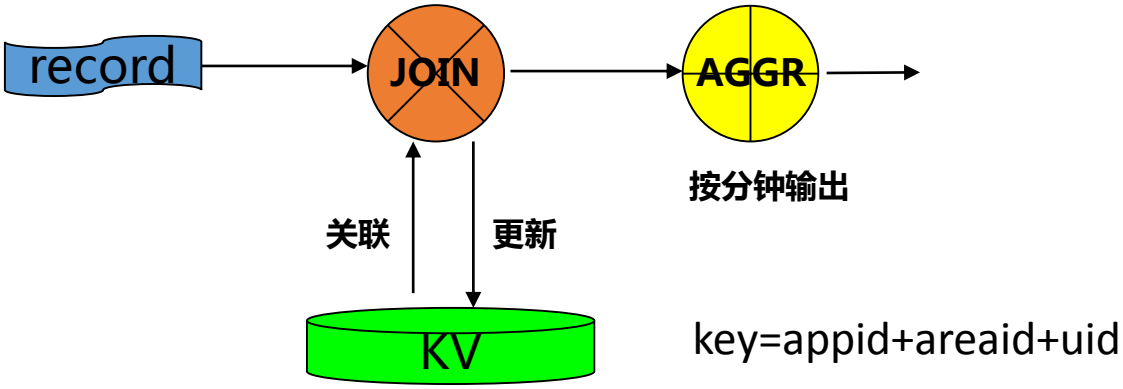


聚合统计



- APP累加UV统计：
  - 对100个APP在10个地区每分钟统计累加UV。
  - 累计UV: 从当天0点开始到当前分钟的独立登录数
  - 假设每个APP在每个地区的天UV约为1千万





方案分析：

KV内存占用：100*10*1千万*60Byte=600GB+

KV查询：100亿次，12万次/s

基于BitMap做去重统计

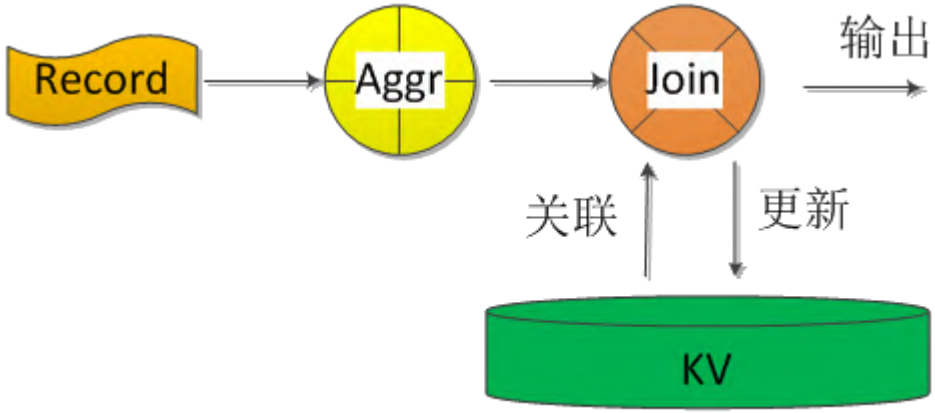
0-7这8个数字

1.使用java byte表示， 占用8*1B=8B

2.使用一个8bit的位图就可以表示， 占用1B

1	1	1	1	1	1	1	1

key=appid+areaid,value使用位图字节数组存储userId



方案分析

KV内存占用：100*10*1.192MB=1.164GB

KV查询：10*100*1440=144万次/天，~20次/s

利用基数统计技术（**HyperLogLogPlus算法**）做去重统计

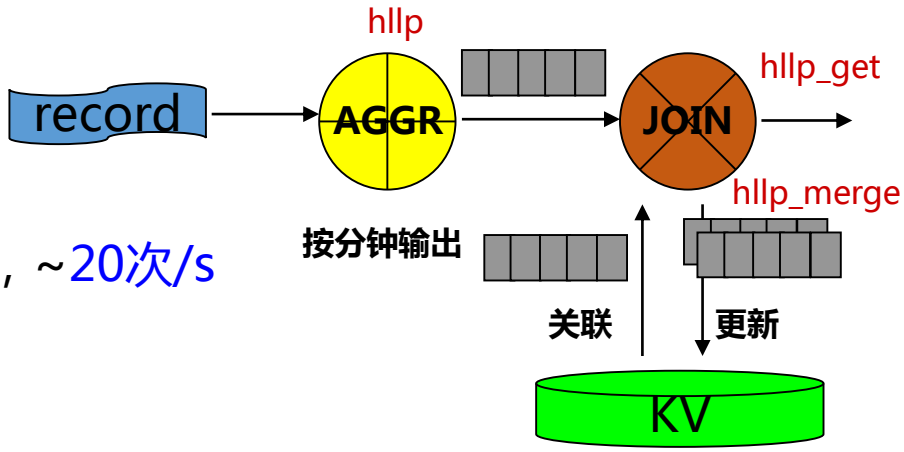
```
with (select appid, areaid, hllp(userId) b from tbl group by appid, areaid ... 60 seconds) t1
(select appid, areaid, jkv.k, hllp_merge(t1.b, j.ball) ball, t1.g g from t1 left join jkv on concat(appid, areaid)=jkv.k)) jt
insert into jkv select k, ball from jt
insert into dest select appid, areaid, hllp_get(ball) from jt
```

方案分析：

KV内存占用：10*100*45KB=45MB

KV查询：10*100*1440=144万次/天，~20次/s

非精确UV统计（99.5%准确率）



Q&A

SDCC 2017深圳站
大数据技术实战峰会

CSDN