

# 通向“亿万级”实时流计算之路 ——我们的架构与实践

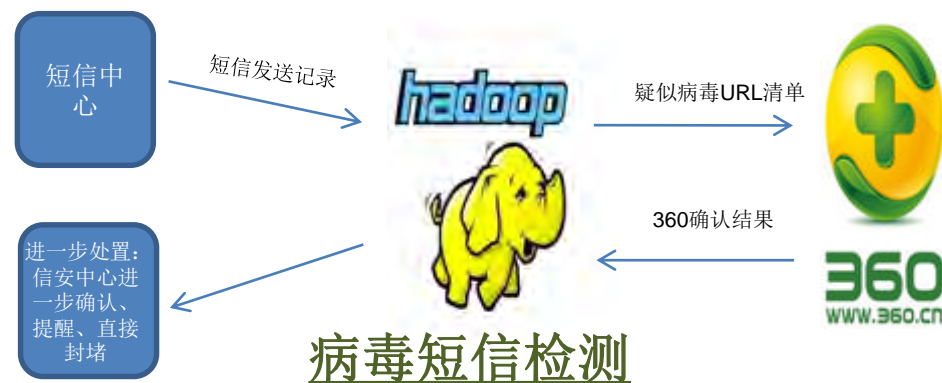
康祖令 博士、数据科学家  
浙江移动网管中心

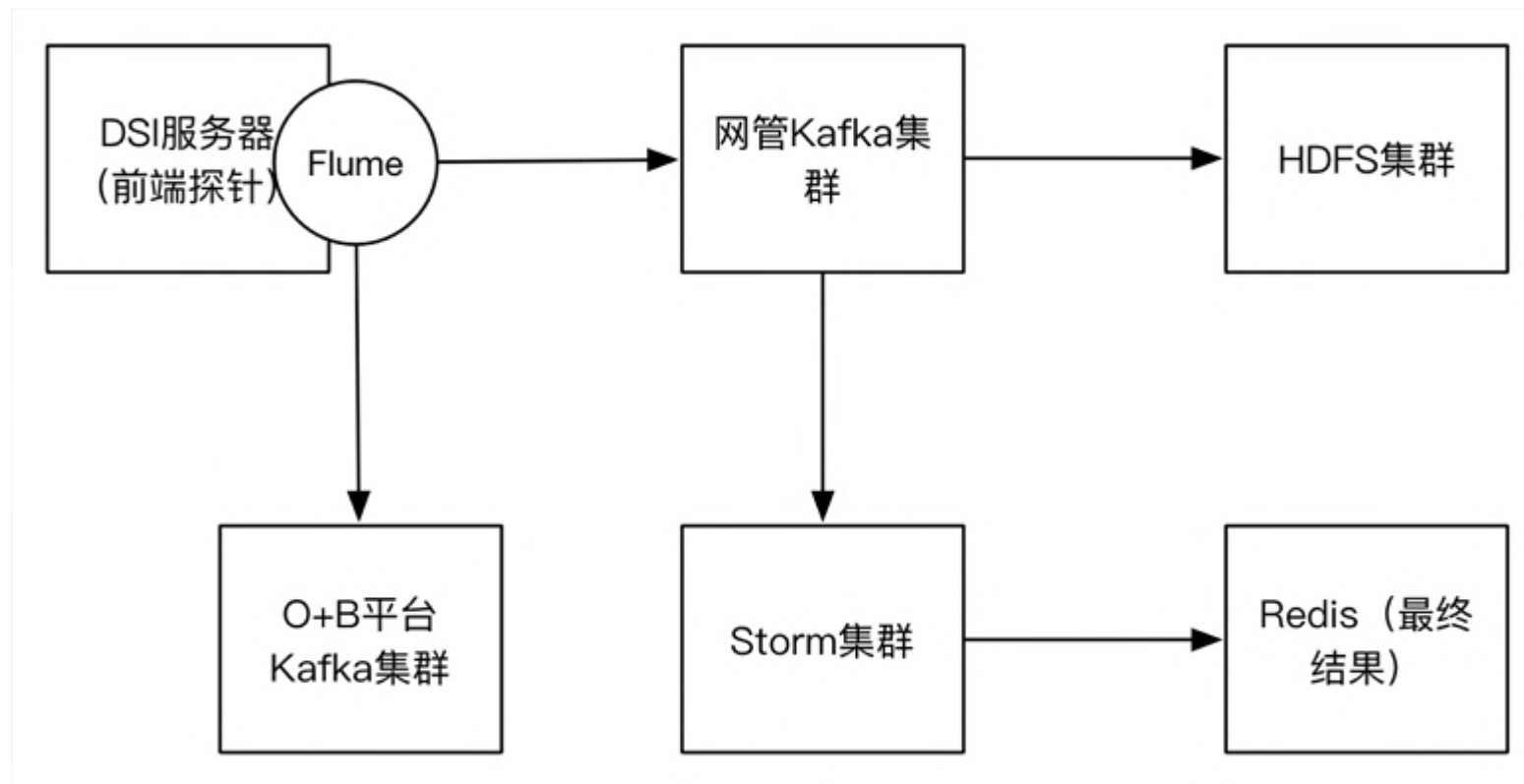
2016年7月

- 应用领域：时延敏感的应用领域
- 网络层面：
  - 实时性能监控
  - 故障预测
- 位置信息服务：
  - 实时人流（标签）
  - 实时标签
  - 用户行为预测

## 故障双向定界

品牌	终端型号	话路	华为	中兴	大唐	贝尔
苹果	Apple iPad Air(A1475)	69.02	70.36	73.96	60.17	59.24
苹果	Apple Mini 2(A1491)	56.56	78.62	69.23	64.89	61.15
苹果	iPhone 5C A1516	55.89	70.62	72.28	56.94	61.59
苹果	iPhone 5C A1526	55.30	64.19	78.45	67.84	74.58
苹果	iPhone 5C A1529	55.89	70.93	66.91	58.70	74.08
苹果	iPhone 5S A1528	55.43	70.97	70.60	67.84	72.49
苹果	iPhone 5S A1529	55.22	68.25	72.24	70.51	67.55
苹果	iPhone 5S A1530	54.45	70.92	66.46	66.28	69.82
苹果	iPhone 6	62.71	71.87	71.95	65.01	60.01
中国移动	CM912	67.67	74.71	73.10	64.84	56.87
中国移动	CM911	64.53	68.82	71.97	64.62	67.13
中国移动	CM911	57.69	74.25	64.26	68.78	66.79
中国移动	CM912	65.73	68.82	68.21	68.17	61.58
中国移动	CM909	68.51	61.86	69.82	68.24	





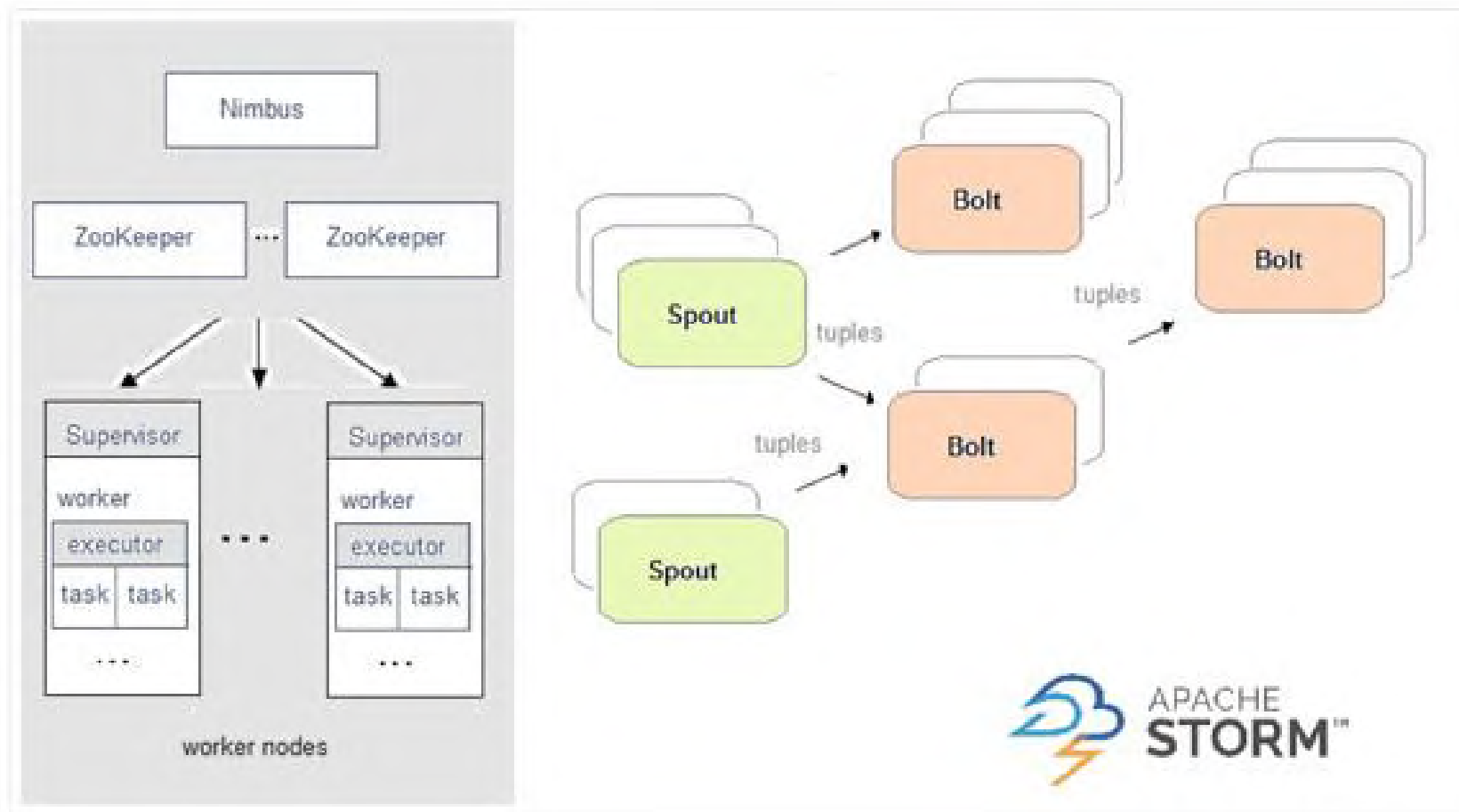


APP#  
3

CPU CORE  
480

数据量  
约50TB/天

记录数  
约1200亿/天

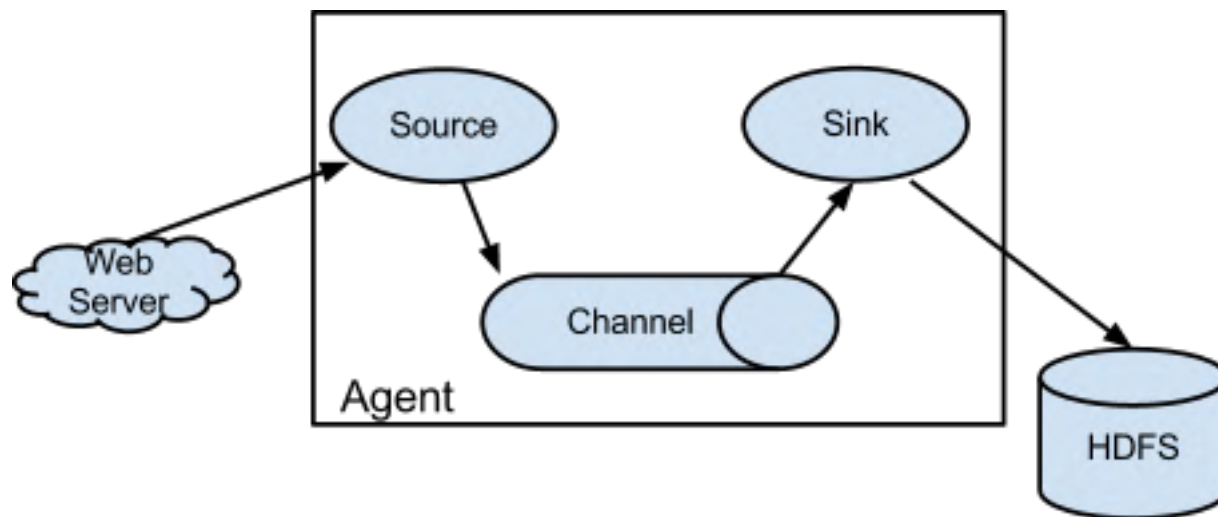


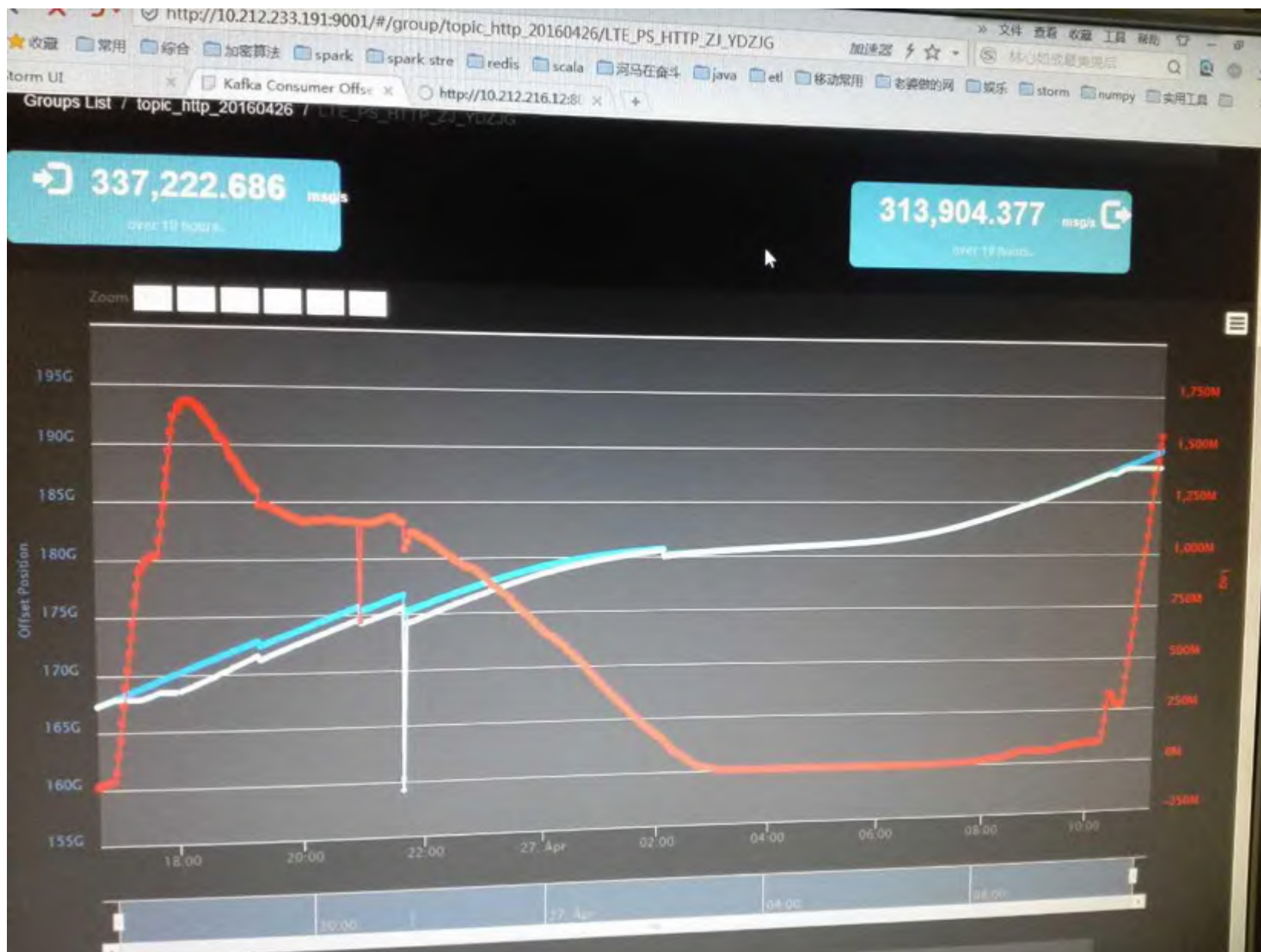
Storm平台的  
物理结构

一个Topology的  
逻辑结构

【问题描述】在Flume出现流量瓶颈，说的更准确一些，是在Flume的前一级服务器出现数据堆积

- 透传，单线程
- 落盘，多线程





【问题描述】Kafka出现瓶颈

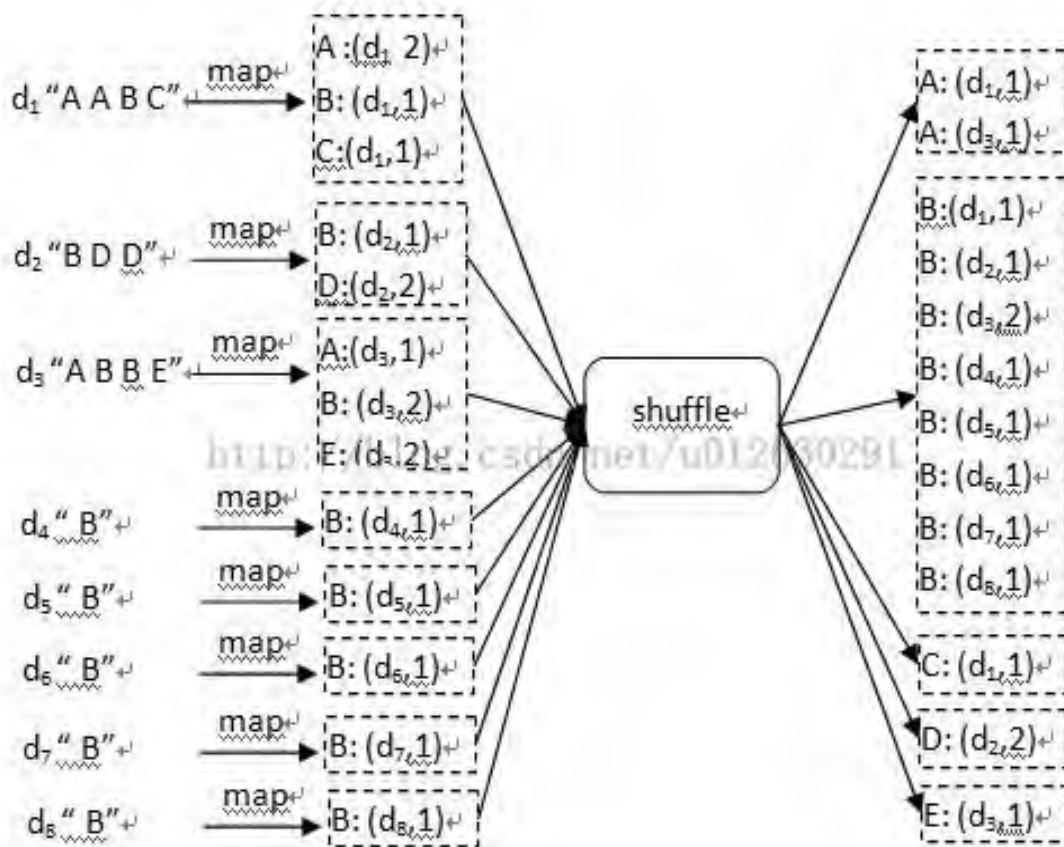


**【问题描述】**严重的结果稳定性问题。以某省会城市的5分钟流量为例，比如一般都在800GB上下，但有时候某一次计算结果突然下降到400GB左右

- ❑ 在不考虑数据延迟的情况下，为了计算5分钟的流量，我们需要在Bolt中设置一个变量hz\_bytes以保存中间结果。比如在10:03的时候，hz\_bytes中实际保存了10:00 ~ 10:03的该市流量。当10:05时，该Bolt就将hz\_bytes的值输出到Redis中。
- ❑ 如果这个Bolt所在的Executor在10:03挂掉，Storm会重新选择一个Worker，并重新调度该Bolt——这是Storm的一个最基本的HA机制。但是当Bolt重新启动后，hz\_bytes却已经丢失了之前暂存的10:00~10:03流量。
- ❑ 所以，最后写出到Redis的，实际上是10:03~10:05的流量。这就是结果中流量会突然减少的原因。

	增量式	覆盖式
性能	?	?
健壮性	?	?

**【问题描述】** 因某个Bolt处理的数据量特别大，导致集群性能严重不足。



**【例】** 如果计算每个地市的总流量（某省的话，总共是11个地市），一种做法是直接根据地市对数据进行分片，然后直接汇总。但是由于某些大城市的流量特别大，从而导致处理这些地区Bolt的压力特别大，从而大幅度延缓了整个Topology的处理性能。

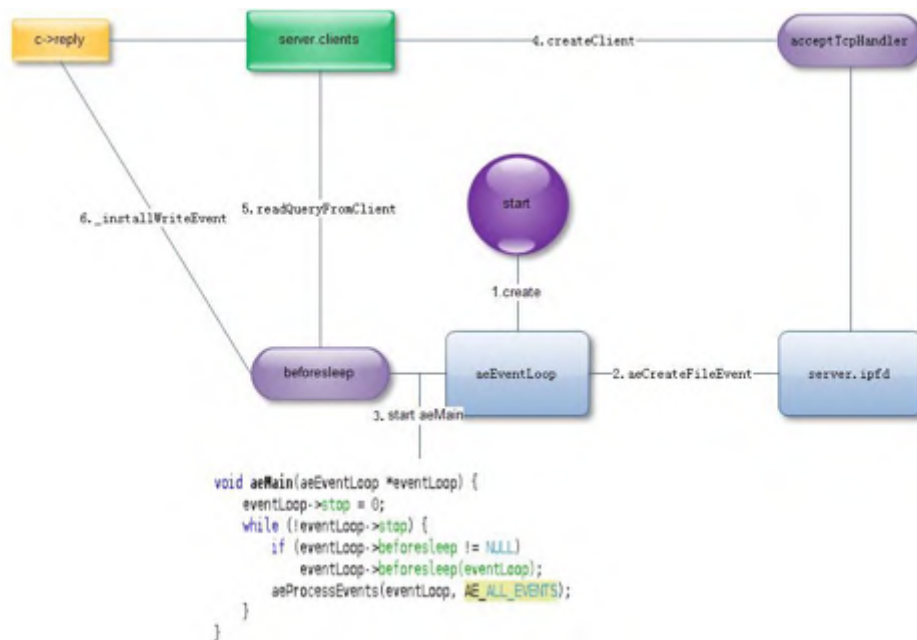
**【解决办法】**

先CELL级别汇总，  
后地市级别二次汇总

**【额外的性能开销？】**

【问题描述】因Redis读写能力不足，导致批量写Redis的Bolt出现性能瓶颈。

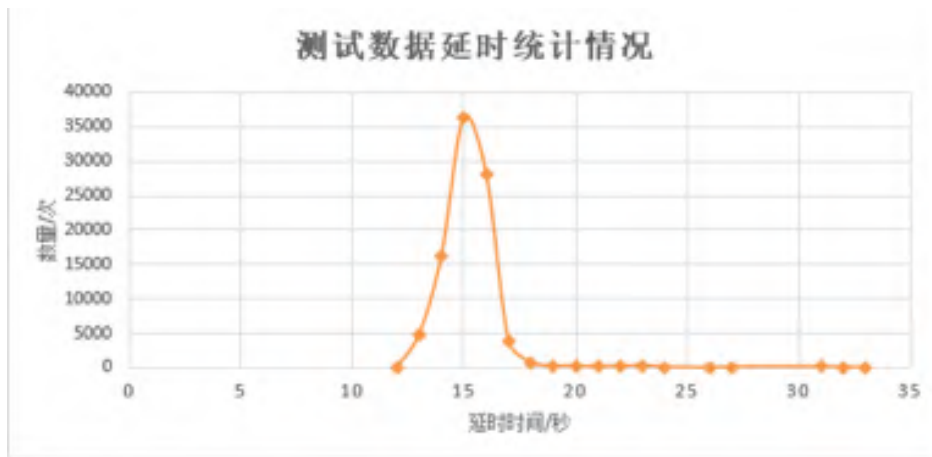
- ✓ 内存操作
- ✓ 单线程
- ✓ 非阻塞IO（EPOLL）



## 如何定位

- Storm UI中“读写Redis”的Bolt会频繁抛出类似“Redis连接池满，无法获得连接”、“Redis连接被复位”的Exception
- Redis采用的是单线程模型，因此，当出现性能瓶颈时，会有一个CPU CORE的使用率接近100%，这个可以通过“mpstat -P ALL”看到

【问题描述】在现网调测和后期运行的过程中，如何确定数据延时的“分布”



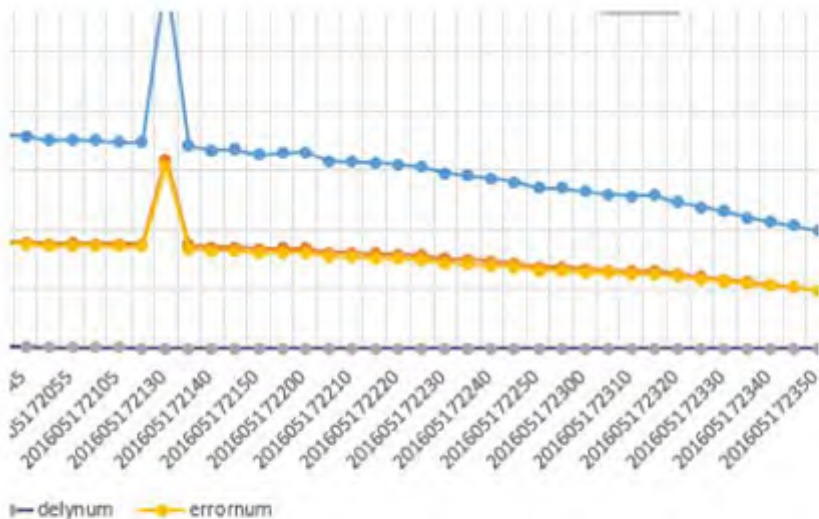
## 2类值得关注的数据延时

- 从Flume进，到Kafka出
- 从Flume进，到Storm输出最终结果（端到端延时）

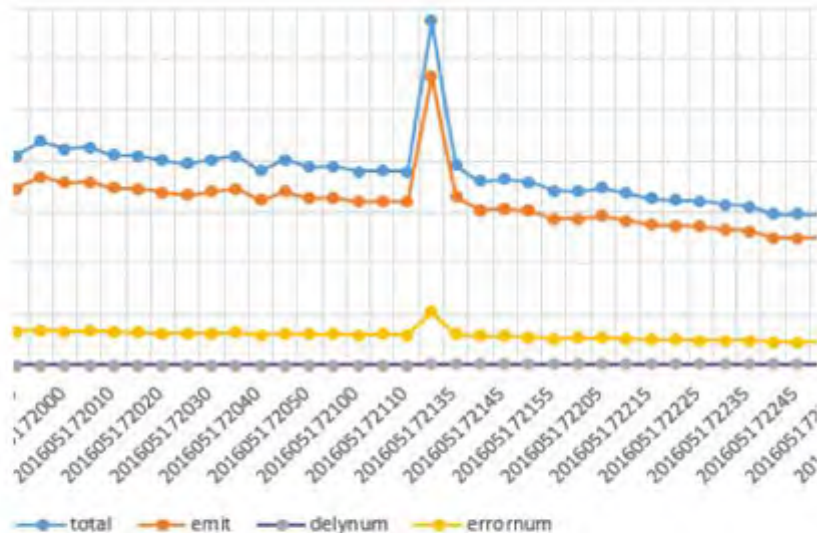
【基本思想】Flume源端随机生成测试数据，Storm端发现并保存数据



【问题描述】处理的数据量严重偏少，统计出来的流量值只有其他途径的1/2”



过条件判断时的数据质量



正常条件的数据质量

在Spout端统计Storm处理的数据总量、过滤数量及由Spout最终Emit出来的量，以文件的形式保存，并通过脚本将数据进行统计处理

**习惯与“脏数据”共事，对大数据从业者而言很重要！**

【问题描述】在尽可能少消耗系统资源前提下，在误差允许的范围内，实时计算基站小区的活跃用户数（类似于UV的实时计算？）



最准确的方法——用HashSet  
保存手机号码

	Bloom Filter	HyperLogLog
资源消耗		★
误差	★	
统计结果的可聚合性		★

## 重点关注

- ▶ Flume、Kafka、Storm、Redis等各主要进程运行是否正常？
- ▶ 数据在“流转”过程中在任何节点都不会出现“积压”？
- ▶ 混入的测试数据的计算结果是否正确？

## Flume

- ▶ 数据积压
- ▶ 测试记录

## Kafka

- ▶ Kafka Manager
- ▶ Kafka Offset Monitor

## Storm : Storm UI

- ▶ Spout和Bolt所发射的数据量，由此可以发现整个集群当前的数据处理量
- ▶ 当某个Spout或者Bolt代码中有Exception抛出时，会在Storm UI中统一显示出来

## Redis

- ▶ Redis进程的CPU使用率



中国移动  
China Mobile

*谢谢!*

中国移动内部资料，  
未经允许不得复制、转发、传播。