

滴滴业务实时监控系統架构及实践

艾毅 2016/12





地球上的一半的人 已和我们同行

美好出行服务触达约**50%**全球人口

覆盖中国近**400座**城市

包括美国，及即将覆盖的东南亚、南亚区域国家，近**400座**海外城市

* 滴滴快的、Lyft、Uber、Ola建立全球共享出行合作联盟，覆盖中国、美国、东南亚、印度提供出行服务，覆盖全球50%人口

2016年6月20日摘自: <http://wq.qq.com/onservice/2016-12-06/detail-ixmihao9076081.shtml>

滴滴出行

- 滴滴实时监控系統演变历程
- 当前架构及服务介绍
- 系統优化方向

滴滴实时监控系统的演变历程



2015年之前

2015年至今

挑战

- 快速满足业务需求

- 业务数据库分库、分表
- 数据、维度、指标越来越多

预计算

Kafka+Druid+Samza

解决方案

瓶颈:

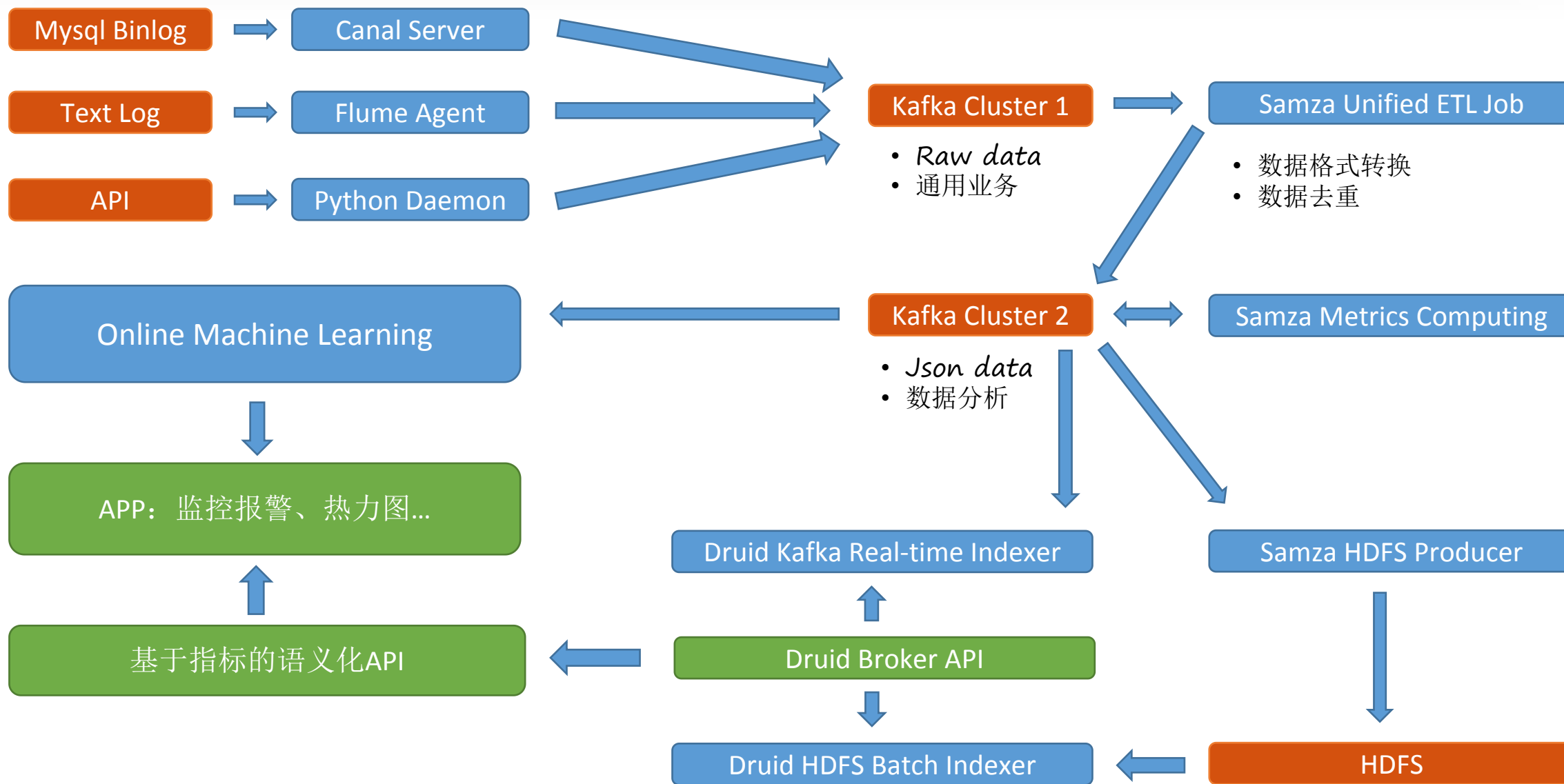
- 计算、存储成本指数级增长
- 不易扩展

优势:

- 实时分析海量数据 (秒级)
- OLAP系统交互式查询

- 滴滴实时监控系統演变历程
- 当前架构及服务介绍
- 系統优化方向

滴滴实时监控系统当前架构概览



当前系统架构特点



- 高可用
- 易扩展
- 高性能
- 支持有状态的实时计算

为何选用Kafka?



Kafka 是一个高性能、高可用、易扩展的
分布式日志系统

将整个数据处理流程解耦

为何选用Druid?



*Druid*是针对时间序列数据提供低延时的数据写入以及快速交互式查询的的分布式OLAP数据库

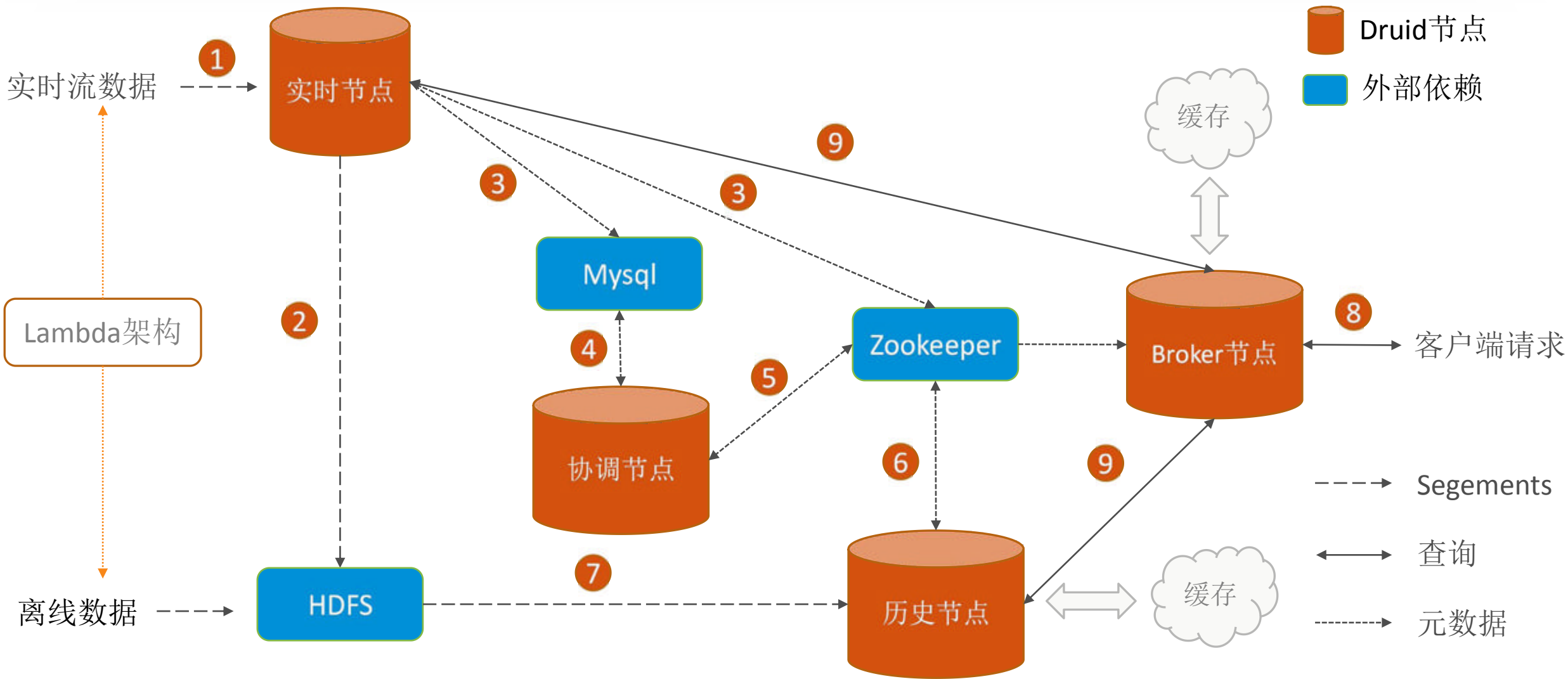
- 为OLAP查询优化过的列式存储结构: *Segment*
- *Segment*中存储聚合计算后的统计结果
- 主要根据时间对*Segment*文件进行分片存储

Segment包含的三种列类型

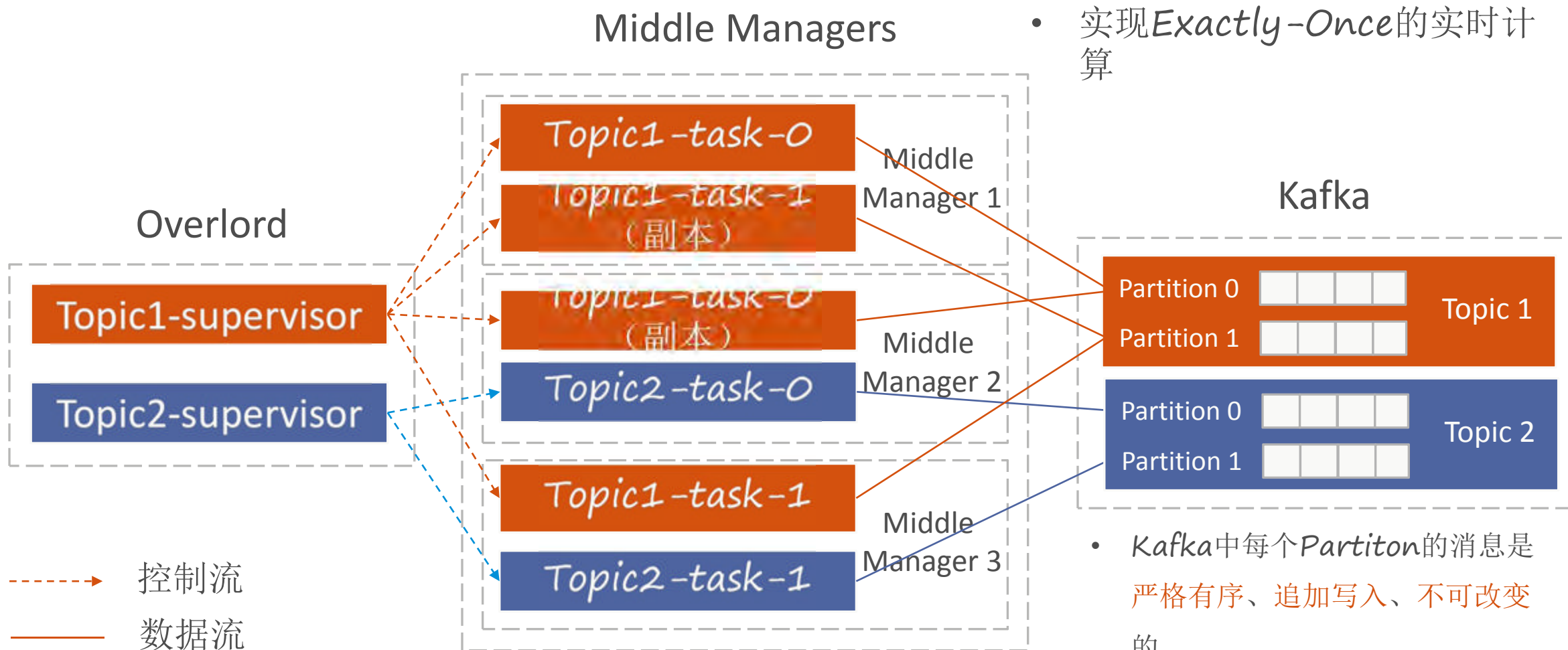


- 时间戳列
 - 作为数据分发、存储、查询的依据
- 维度列
 - 支持过滤和分组
 - 使用字典编码压缩
 - 使用*BitMap*索引压缩
- 指标列
 - 用来聚合计算
 - 使用*LZ4*压缩

Druid的数据处理流程介绍



Druid Kafka Indexing Service介绍



- 实现 *Exactly-Once* 的实时计算

- *Kafka* 中每个 *Partiton* 的消息是严格有序、追加写入、不可改变的
- 可“回退”到任意的 *Partition-*

Druid支持近似统计算法



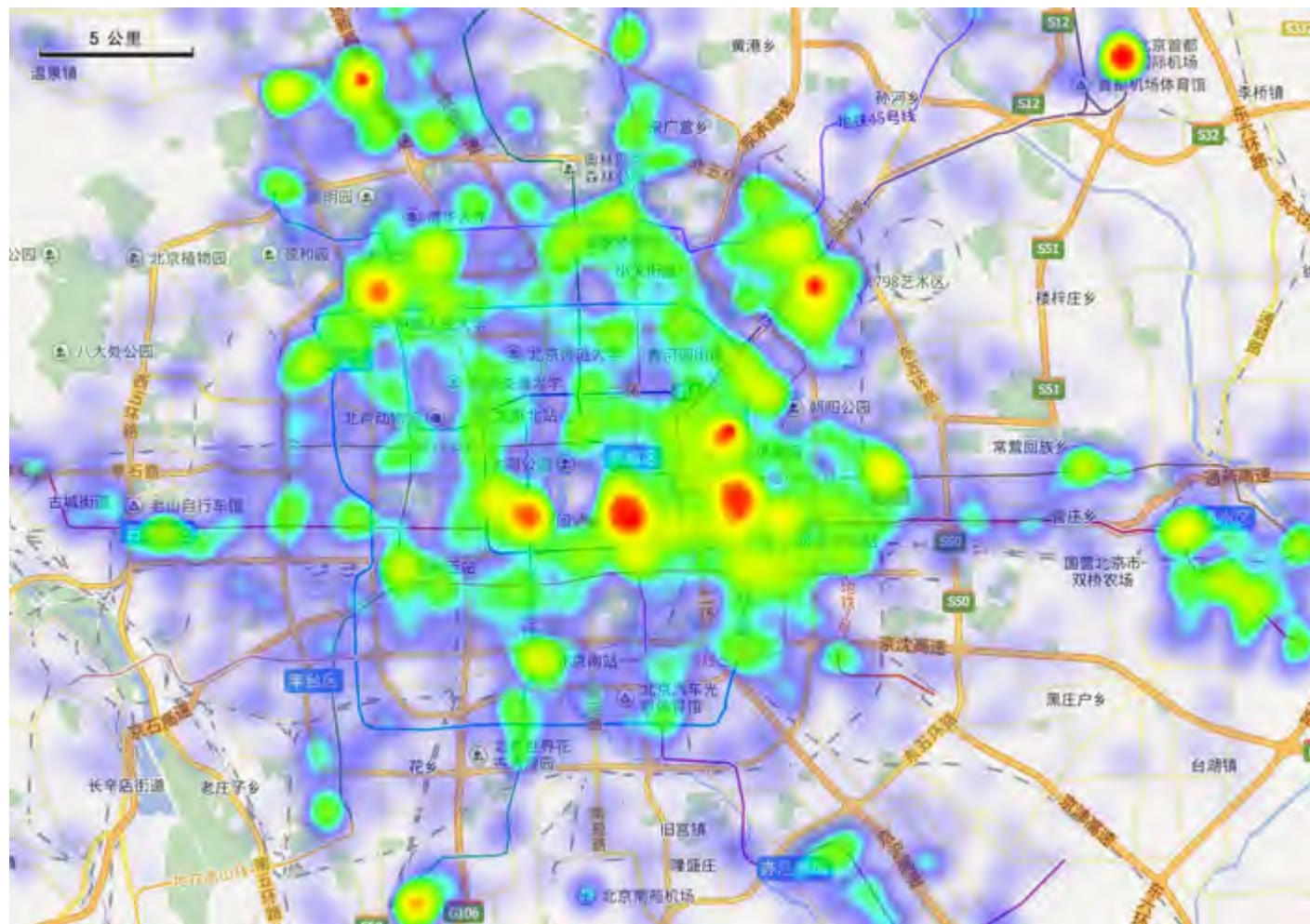
- 为什么要使用近似算法?
 - 计算速度快
 - 误差可控
 - 显著降低计算、存储需求
- *Druid*是如何实现近似算法的?
 - 基于Yahoo开发的 [datasketches](#) 库
 - 使用Theta Sketch近似算法
 - 支持集合操作（并集、交集、差集）

Druid支持地理查询



- 矩形查找
- 圆形查找
- 任意多边形查找
 - 滴滴贡献给了社区
 - 包含在Druid 0.9.2版本

滴滴实时订单热力图

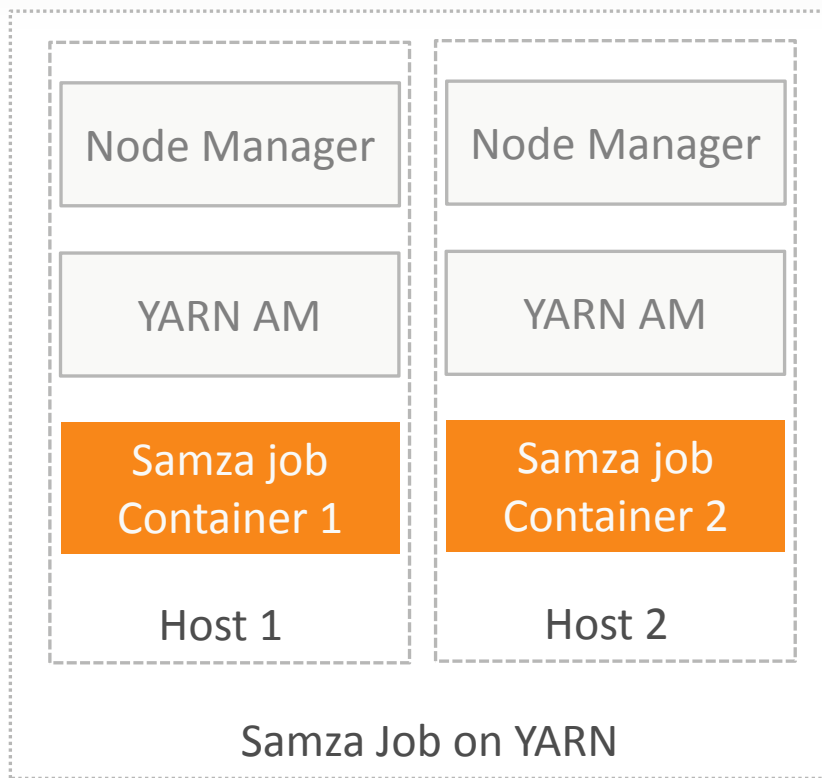


为何选用Samza?



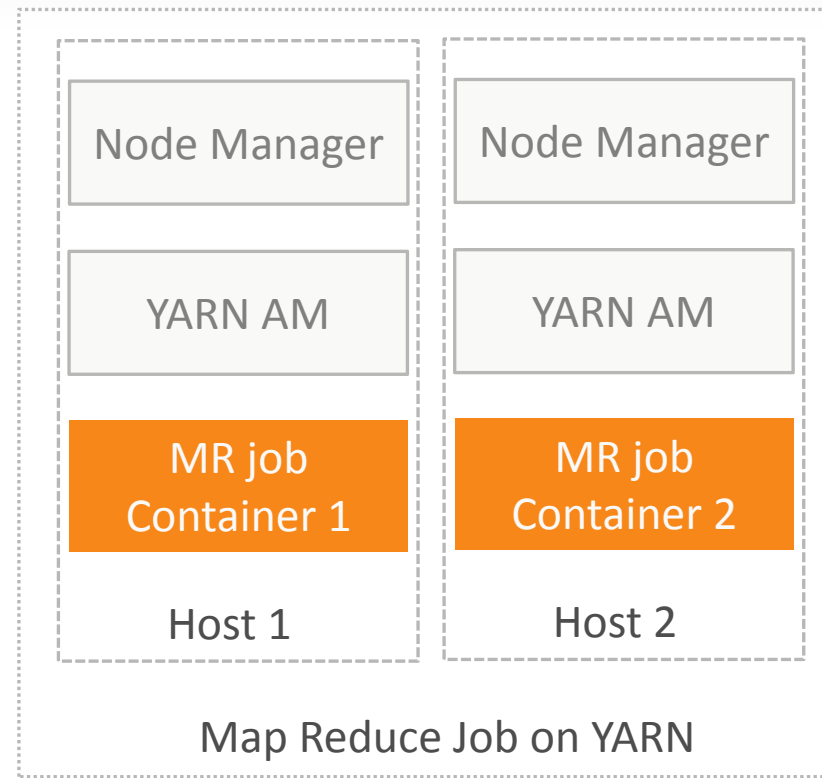
Samza是一个分布式的实时计算框架；
支持低延时的、有状态的实时计算

Samza运行机制介绍



Kafka

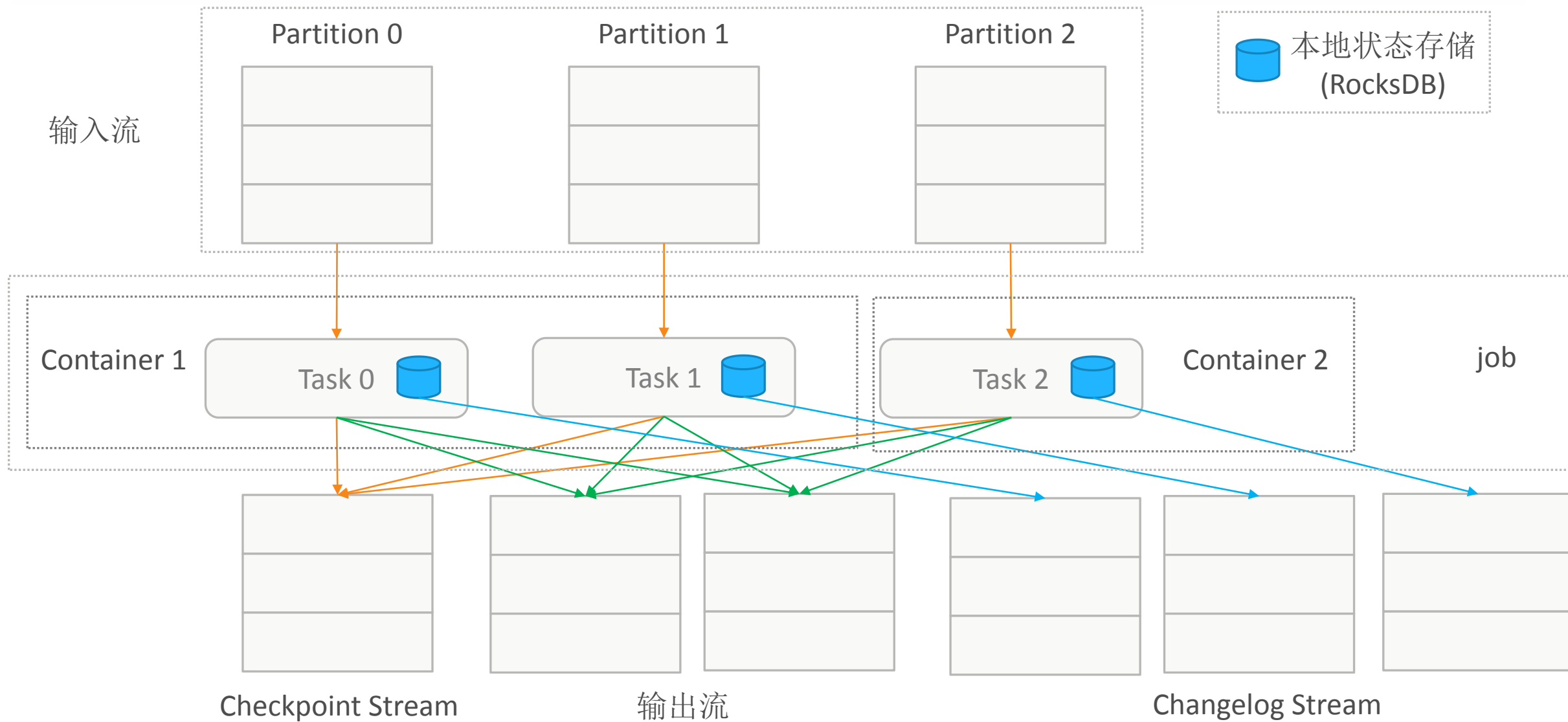
VS



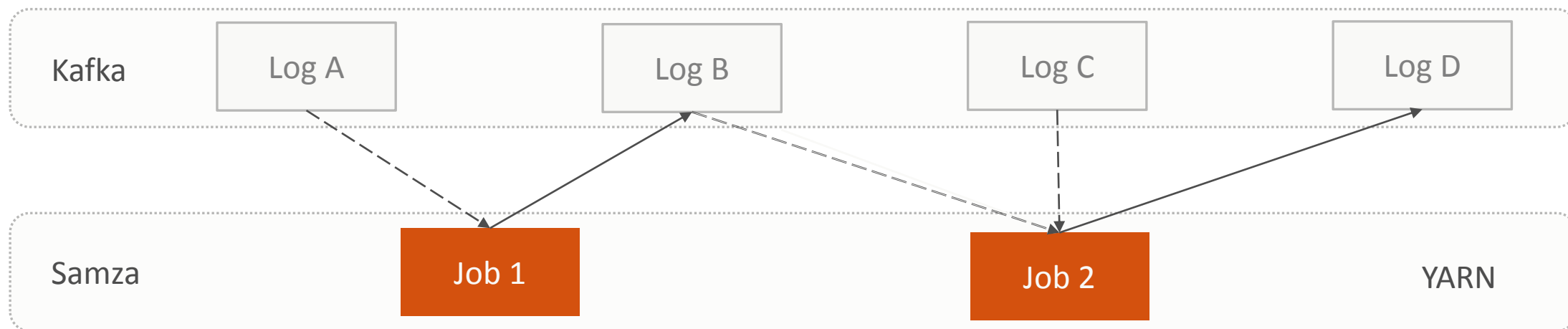
HDFS

Samza Job可看作一个实时计算版的Map/Reduce Job

Samza数据处理流程介绍



Samza的高可用性



- 不会出现“雪崩”
- 不会丢失数据
- 缓存队列：基于磁盘，不受内存限制
- YARN为Samza Job提供了容错机制

- 处理单条信息

- *Filter*: 根据特定条件筛选匹配的记录
- *Map*: $Log = f(Log)$

- 处理多条信息

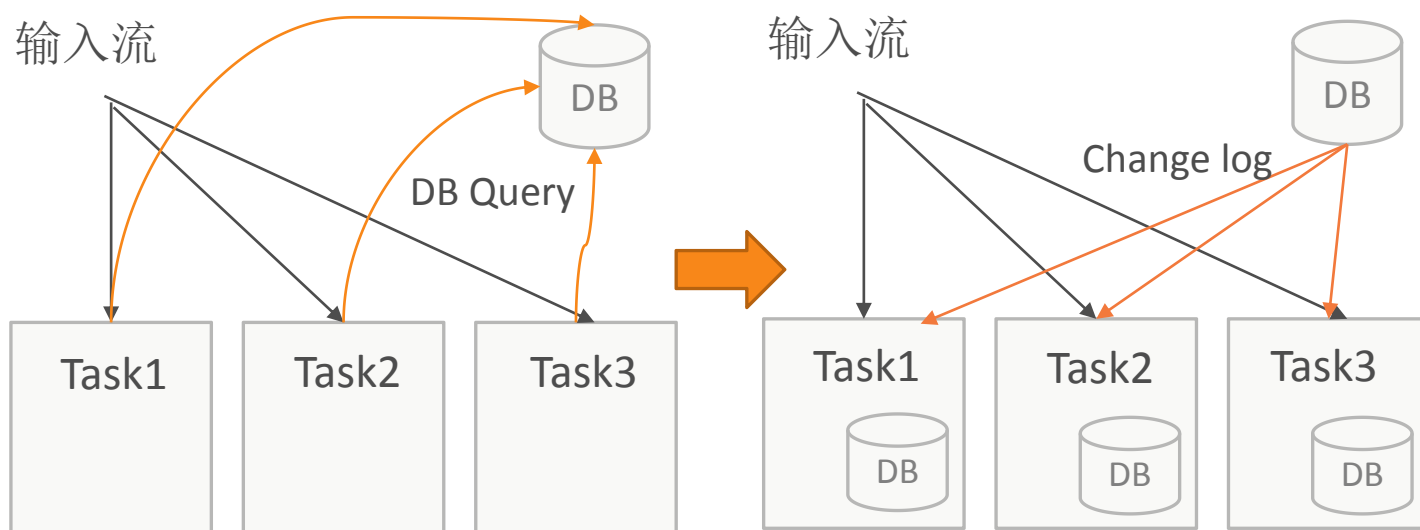
- *Join*: 对多个数据流做关联计算
- *Group*: 对记录进行分组
- *Aggregate*: 对处于相同*Group*的数据进行聚合计算

} 需要状态管理

Samza实现有状态的实时计算

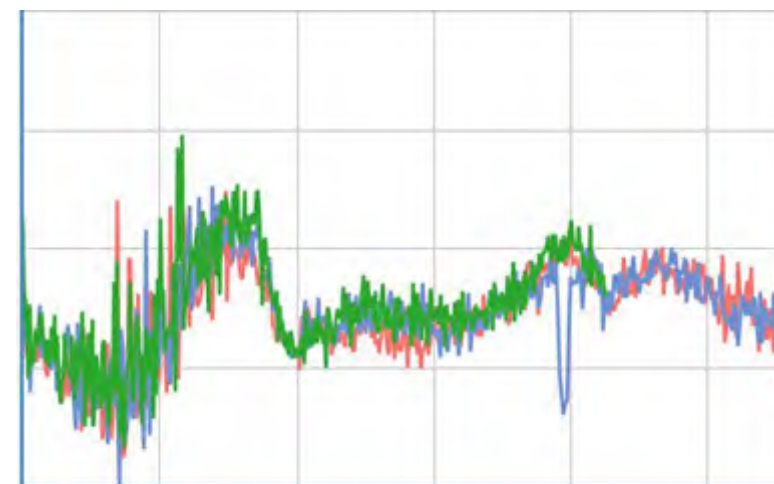


- 基于过去的的数据状态来处理当前的数据：本质是数据缓存机制
- *Samza*基于*RocksDB*实现本地状态存储



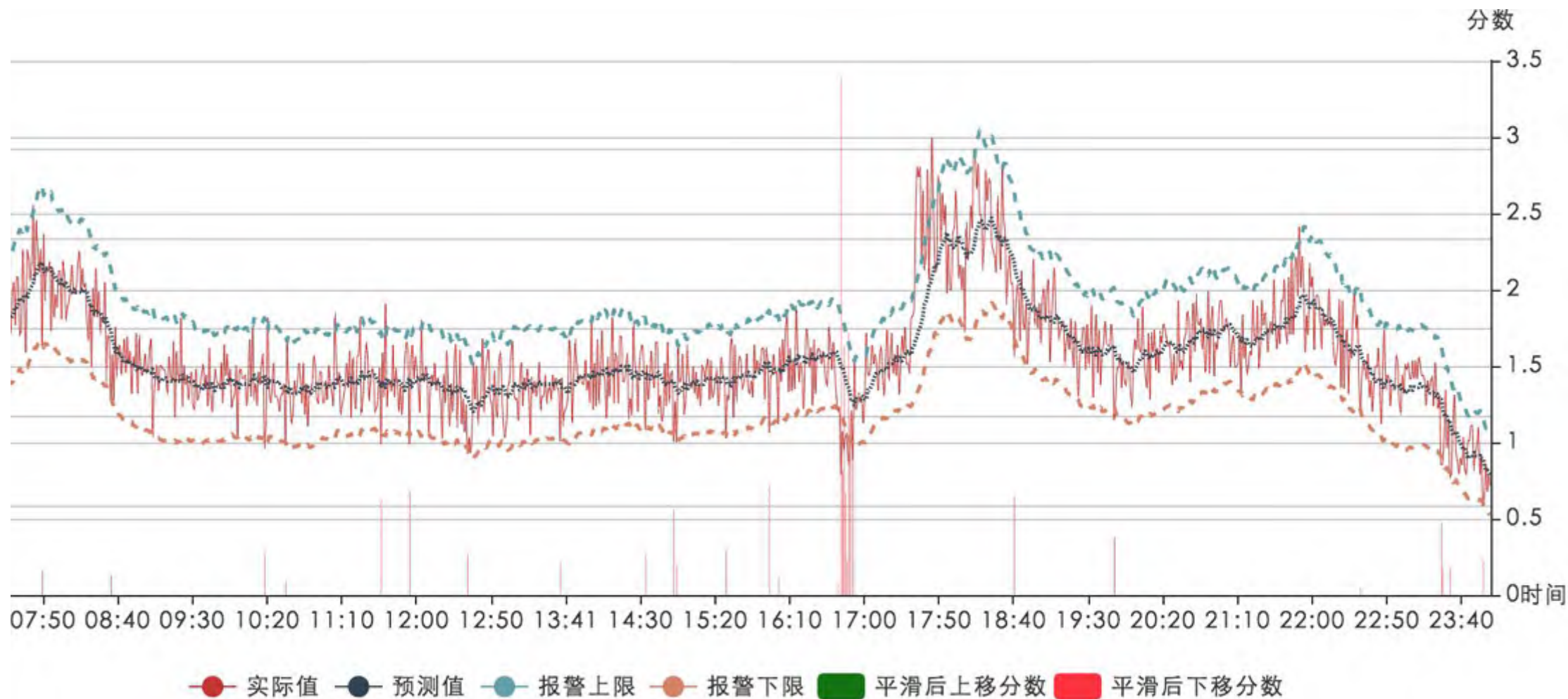
大数据实时计算场景下
移动数据，离计算资源近一点

滴滴实时监控可视化界面

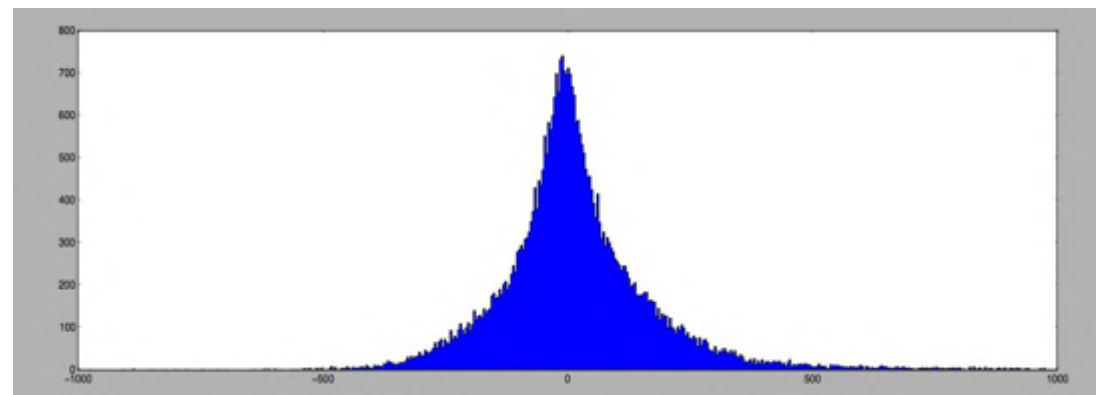
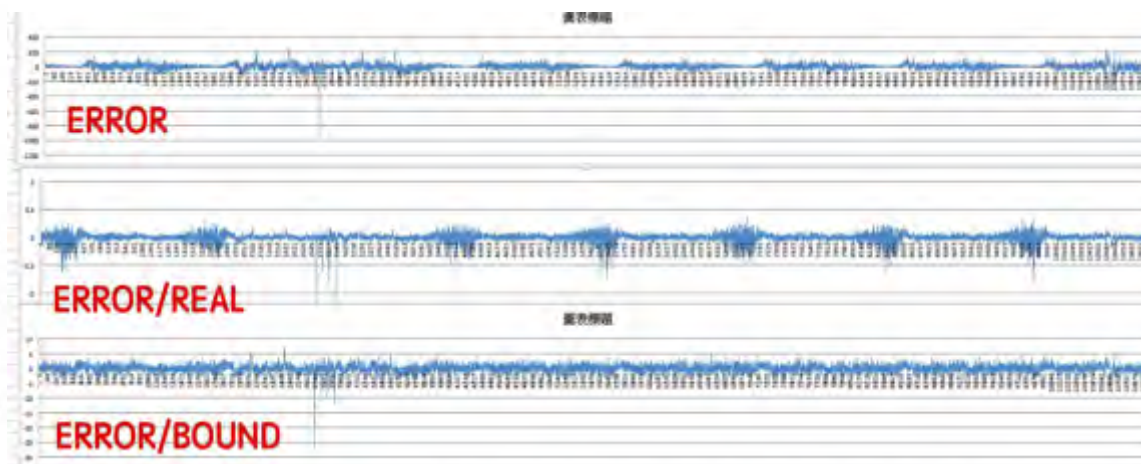
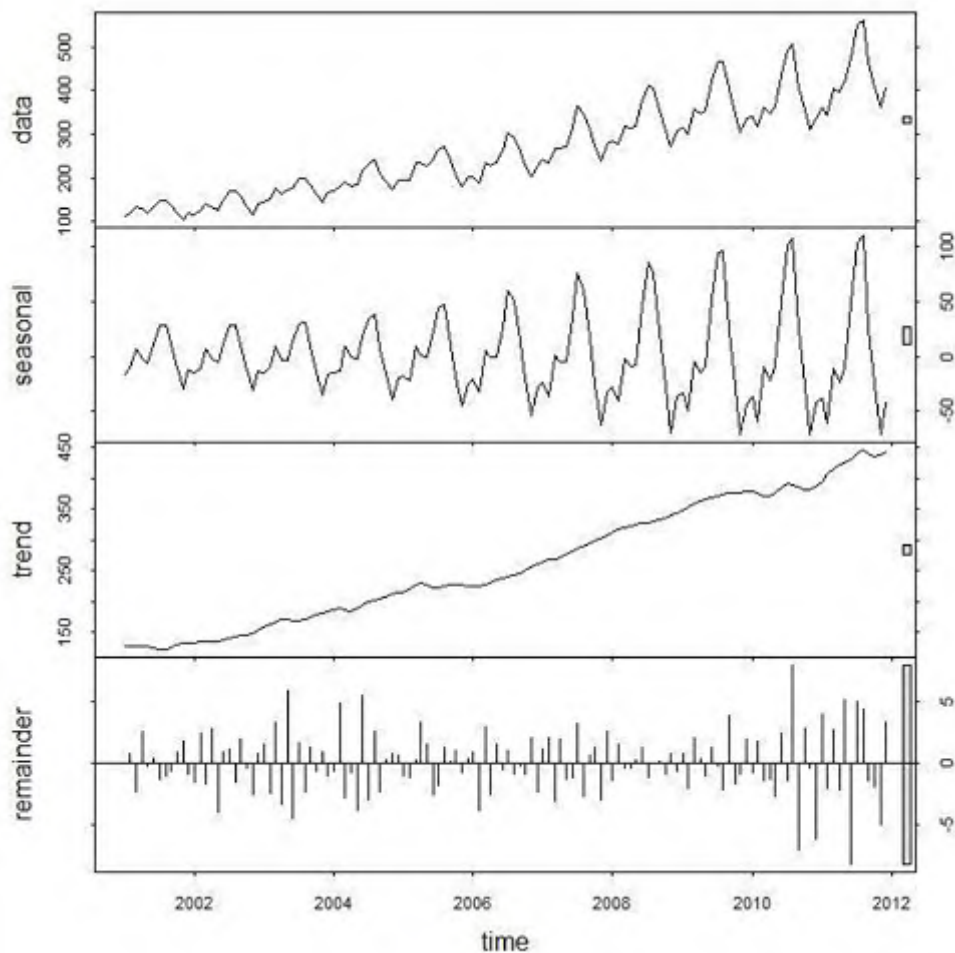


- 监控覆盖滴滴全部核心业务线

基于Holt-Winters时间序列分析模型



Holt-Winters时间序列分析模型介绍



- 滴滴实时监控系統演变历程
- 当前架构及服务介绍
- 系統优化方向

Lambda架构的问题

- 同样的业务逻辑需要维护实时和离线计算两套代码
- 重新处理数据只能依赖离线计算，计算较慢

- 实现“端到端”的*Exactly-Once*实时数据处理，不再需要离线修正
 - *Samza Local Cache*
 - 智能感知*Kafka Partiton*变化
 - *Druid Kafka Indexing Service*
- 数据的重新处理机制：
 - *Kappa* : 新起一个实时计算任务，新旧任务并行处理
 - *Liquid* : 停止当前实时计算任务，修改*Offset*后，重启任务

THANK YOU

北京嘀嘀无限科技发展有限公司

北京市海淀区东北旺路8号院尚东·数字山谷B2号楼

www.xiaojukeji.com

