QCon[上海站]
全球软件开发大会2016

网易APM数据处理系统实践

SPEAKER

焦智慧

International Software
Development Conference

主办方 Geekbang. 极客邦科技 InfoQ

- 2010年毕业加入网易

- 2012年网易云相关工作
- 2014年底开始负责APM

安排

- APM简介
- 系统演进过程
- 碰到的问题

Application Performance Management

移动端

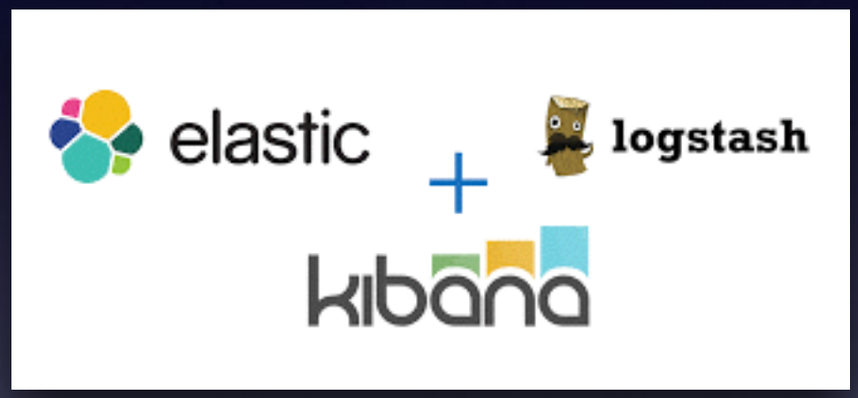
- 线上App的用户体验如何?

网络类型　　运营商

地理位置

服务端

- 如何排查问题?



全链路调用数据

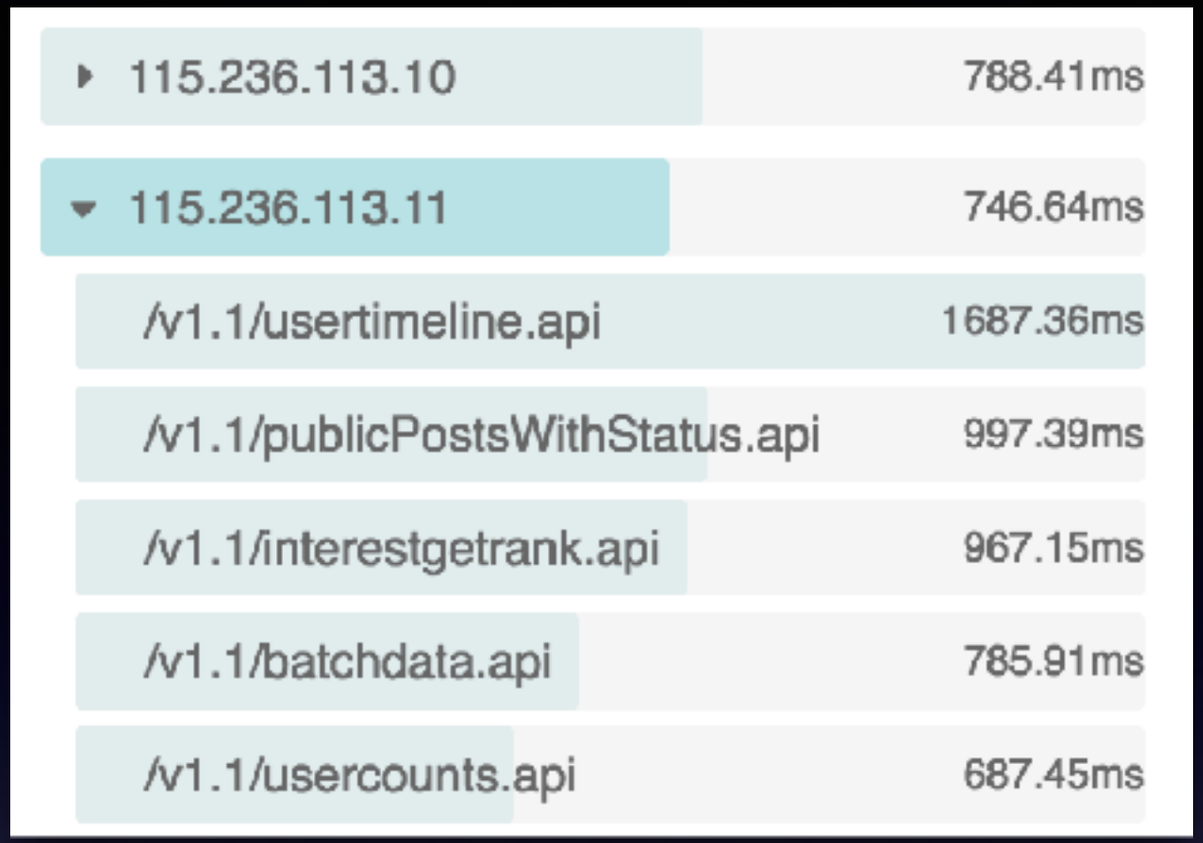
全局拓扑图

提升运维效率

Stage1：单维统计

- Android网络请求性能：DNS时间，首包时间，响应时间，上传下载速率，错误率等
- 单维度：域名，运营商，网络类型，地理位置
- 原始数据：根据用户可查询

Top-N


115.236.113.10 788.41ms
115.236.113.11 746.64ms
/v1.1/usertimeline.api 1687.36ms
/v1.1/publicPostsWithStatus.api 997.39ms
/v1.1/interestgetrank.api 967.15ms
/v1.1/batchdata.api 785.91ms
/v1.1/usercounts.api 687.45ms

1000
800
600
400
200
ms
10-06 20:52
10-06 20:56
10-06 21:00
10-06 21:04
10-06 21:08
10-06 21:12
10-06 21:16
移动
WIFI
联通
电信

时序

Stage1：选择

- 原始数据
- 统计数据
  - 实时
  - 数据量

RowKey

时序

↓

xxx_host_timestamp

Top-N

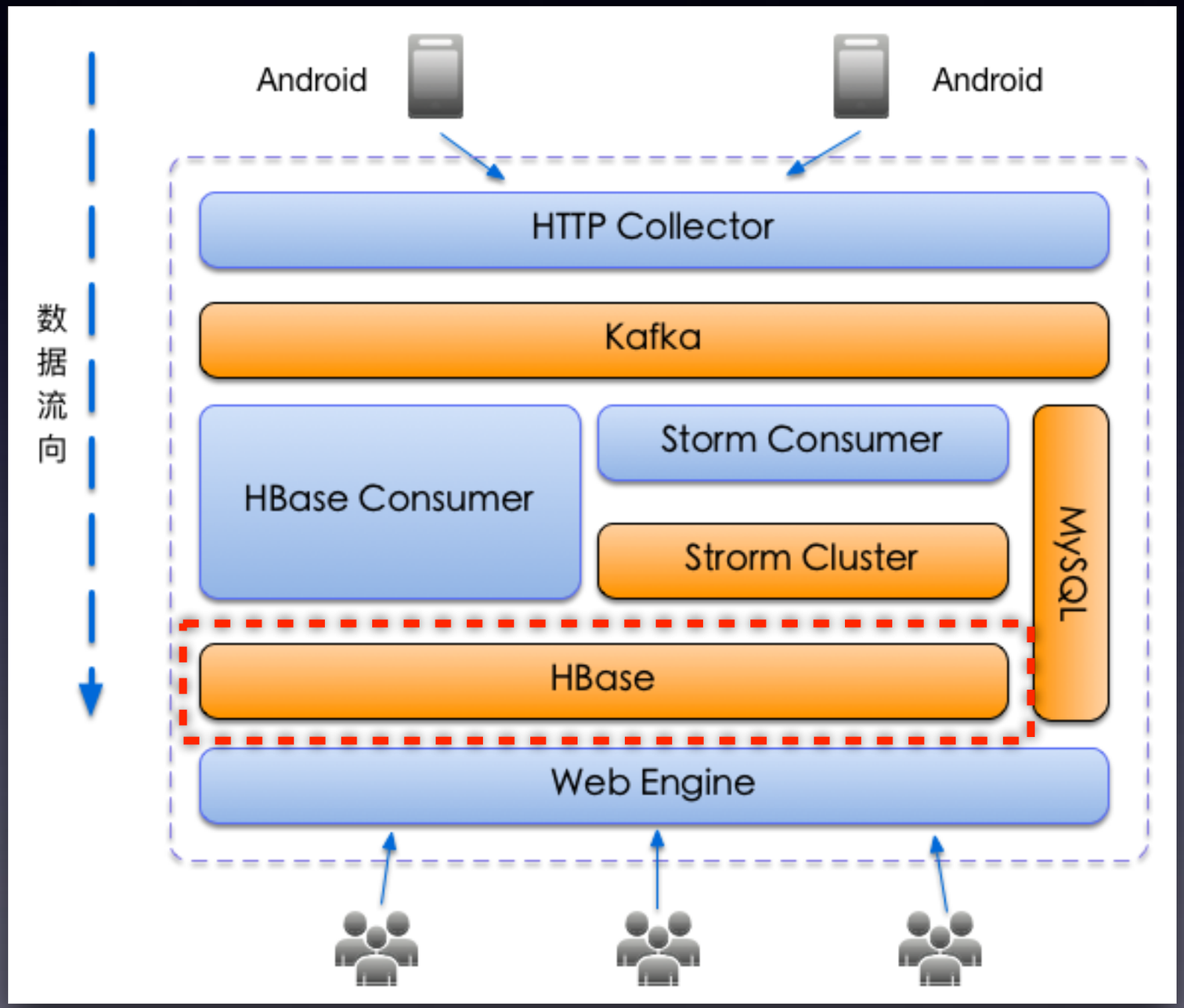
↓

xxx_timestamp_host

RowKey

```
public class StatsConst {
    public static final String TB_HOST_STATS = "TH"; // 指定时间段，所有HOST统计
    public static final String HOST_TB_METRIC = "HT"; // 指定HOST，指定时间段的统计

    public static final String HOST_TB_PATH_STATS = "HTP"; // 指定HOST，以PATH为单位的统计
    public static final String HOST_PATH_TB_METRIC = "HPT"; // 指定HOST，指定PATH，指定时间段的统计统计

    public static final String TB_HTTP_ERROR_STATS = "THE"; // 指定时间段，以HttpError为单位的统计
    public static final String HTTP_ERROR_TB_METRIC = "HET"; // 指定HttpError，指定时间段的统计

    public static final String TB_NETWORK_ERROR_STATS = "TNE"; // 指定时间段，以NetworkError为单位的统计
    public static final String NETWORK_ERROR_TB_METRIC = "NET"; // 指定NetworkError，指定时间段的统计：metric

    public static final String HOST_TB_HTTP_ERROR_STATS = "HTHE"; // 指定HOST，以HttpError为单位的统计
    public static final String HTTP_ERROR_TB_HOST_STATS = "HETH"; // 指定HttpError，指定时间段，以Host为单位的统计
    public static final String HOST_HTTP_ERROR_TB_METRIC = "HHET"; // 指定HOST，指定HttpError，指定时间段的统计

    public static final String HOST_TB_NETWORK_ERROR_STATS = "HTNE"; // 指定HOST，以NetworkError为单位的统计
    public static final String NETWORK_ERROR_TB_HOST_STATS = "NETH"; // 指定NetworkError，指定时间段，以Host为单位的统计
    public static final String HOST_NETWORK_ERROR_TB_METRIC = "HNET"; // 指定HOST，指定NetworkError，指定时间段的统计

    public static final String TB_GEO_STATS = "TG"; // 指定时间段，所有GEO统计
    public static final String GEO_TB_METRIC = "GT"; // 指定GEO，指定时间段的统计

    public static final String TB_OPERATOR_STATS = "TO"; // 指定时间段，所有OPERATOR统计
    public static final String OPERATOR_TB_METRIC = "OT"; // 指定OPERATOR，指定时间段的统计

    public static final String TB_NETWORK_STATS = "TN"; // 指定时间段，所有NETWORK统计
    public static final String NETWORK_TB_METRIC = "NT"; // 指定NETWORK，指定时间段的统计
}
```

Stage1： 架构


Android
Android
数据流向
HTTP Collector
Kafka
HBase Consumer
Storm Consumer
Strorm Cluster
MySQL
HBase
Web Engine

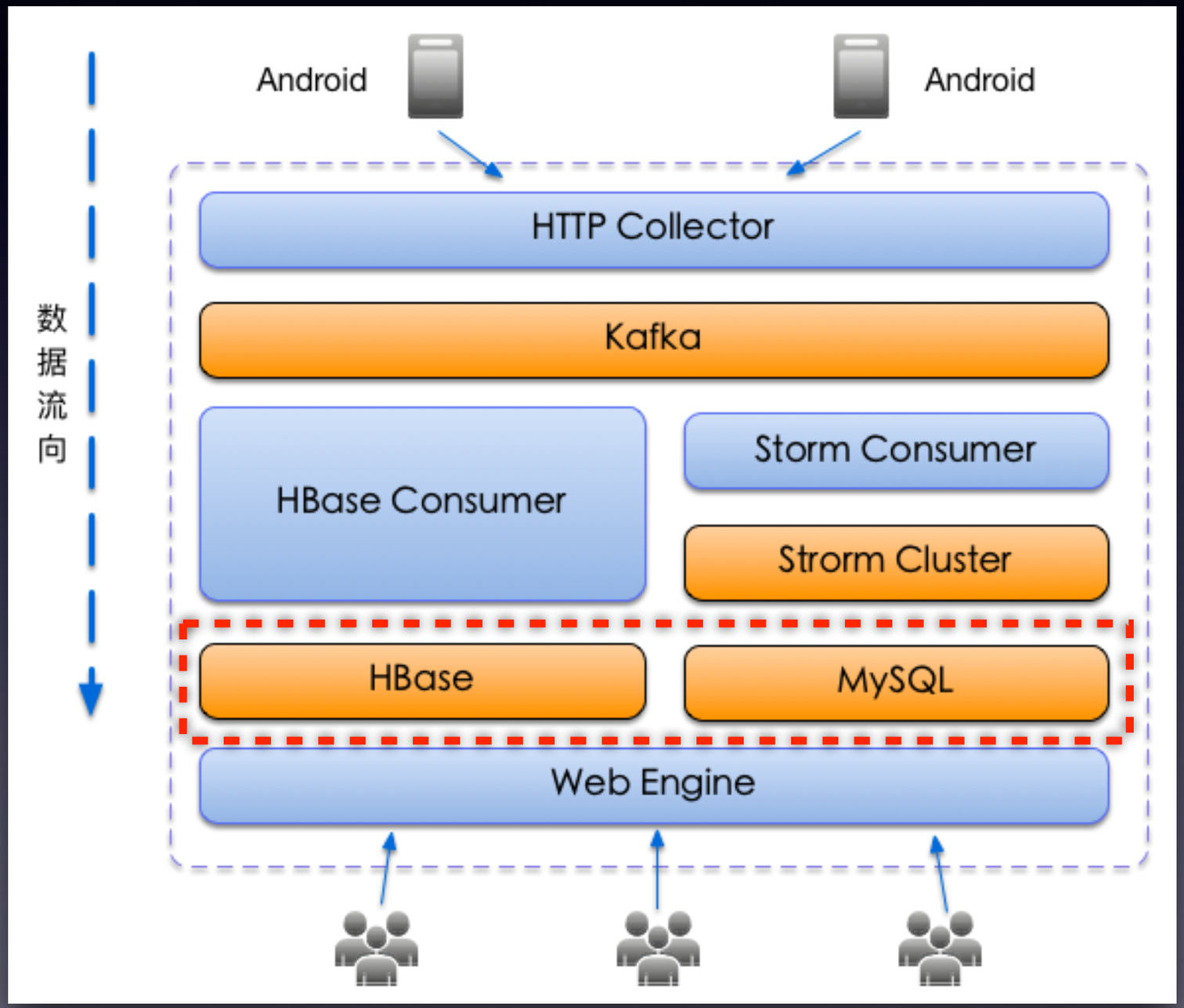
Stage2：多维分析

- 在指定时间段，哪个运营商在哪个区域的哪种网络类型下，性能比较差（错误率比较高）？

移动_WIFI_山东	1151.42ms
移动_4G_山东	872.32ms
移动_WIFI_江苏	791.85ms
移动_WIFI_河南	780.12ms
移动_WIFI_浙江	766.92ms

Stage2：选择

Stage2：架构


Android
Android
数据流向
HTTP Collector
Kafka
HBase Consumer
Storm Consumer
Strorm Cluster
HBase
MySQL
Web Engine

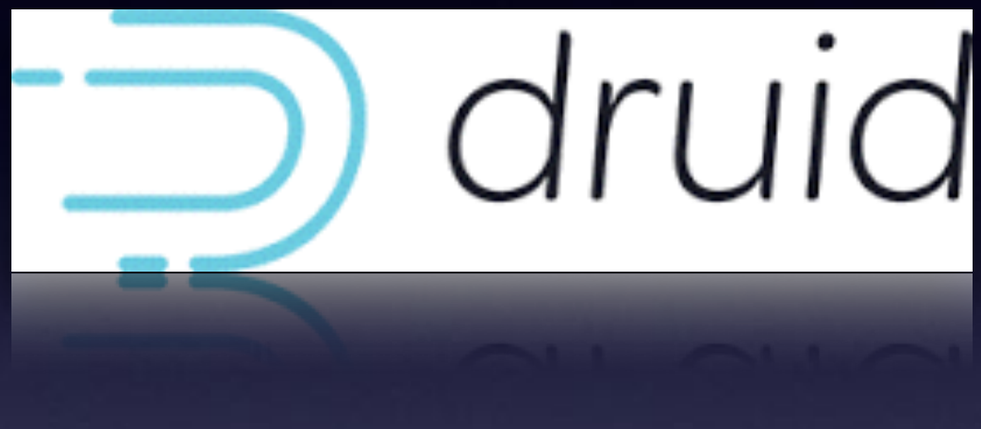
Stage2：问题

写入效率

查询效率

存储需求

- 性能：实时大数据量写入，毫秒级或者秒级查询，多维分析
- 架构：可水平扩展，高可用

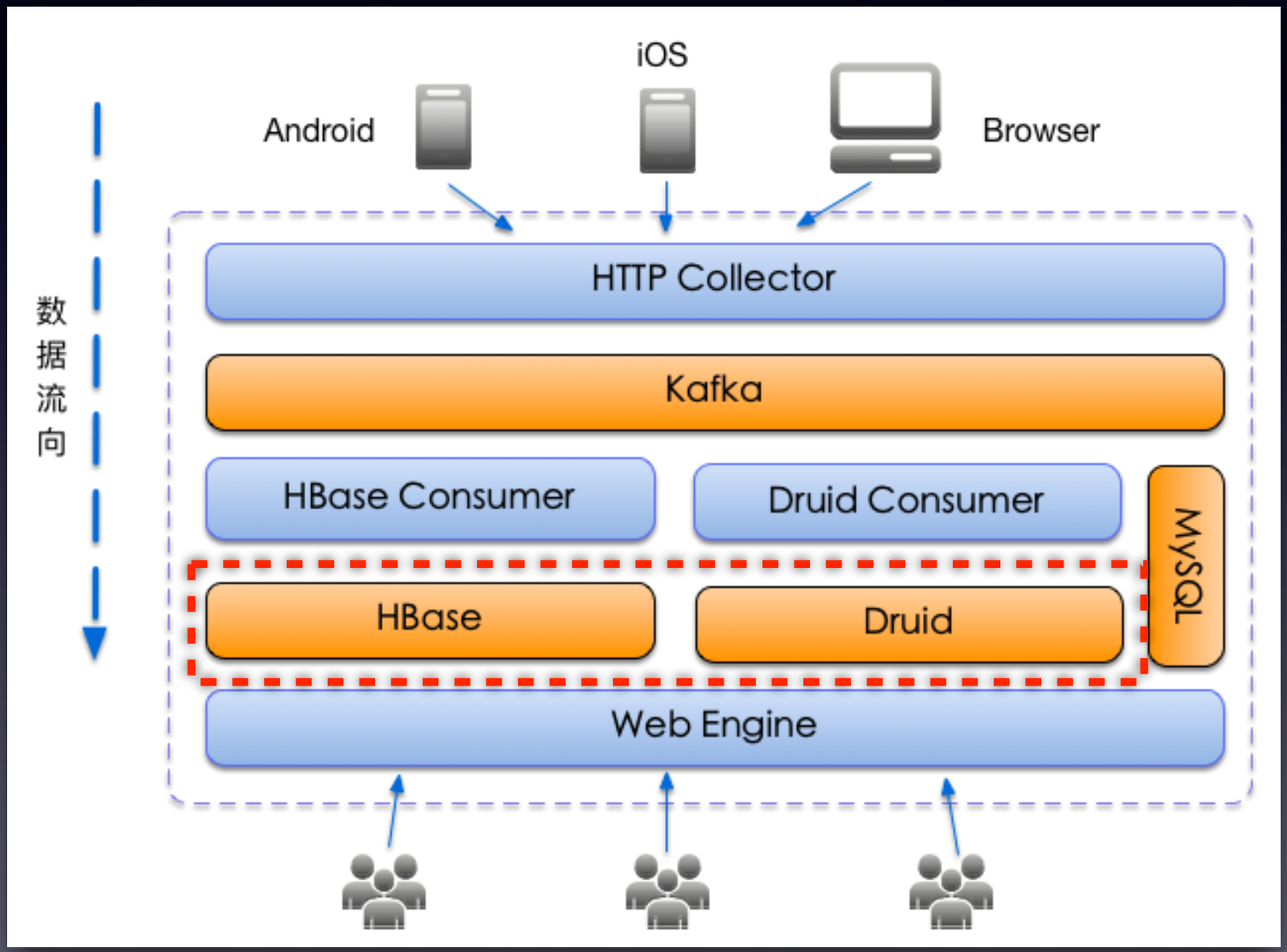


Druid is a high-performance, column-oriented, distributed data store

Druid

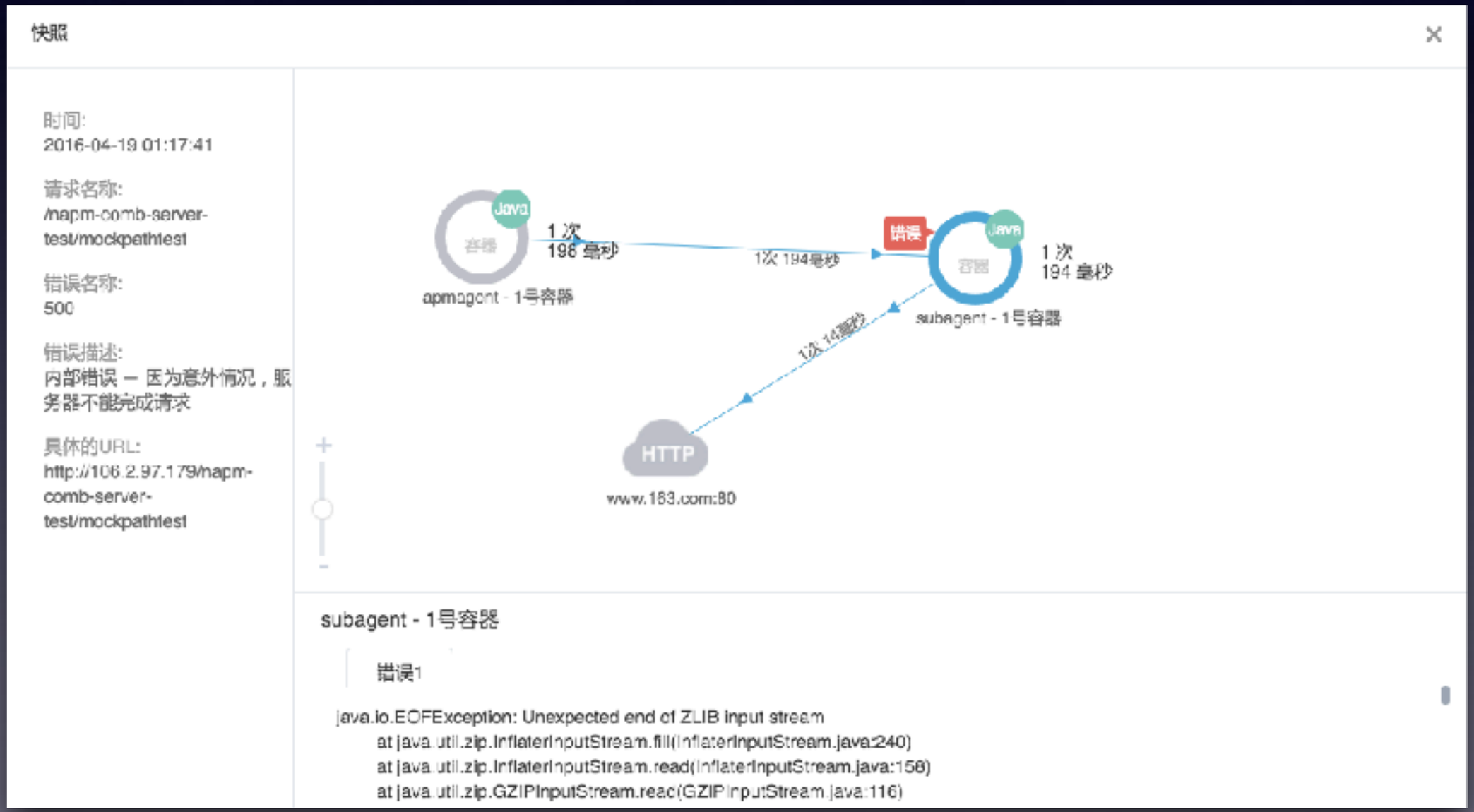
- 实时数据写入，预聚合
- Sub-Second查询性能
- 可水平扩展至PB级
- 高可用

Stage3：架构


iOS
Android
Browser
数据流向
HTTP Collector
Kafka
HBase Consumer
Druid Consumer
MySQL
HBase
Druid
Web Engine

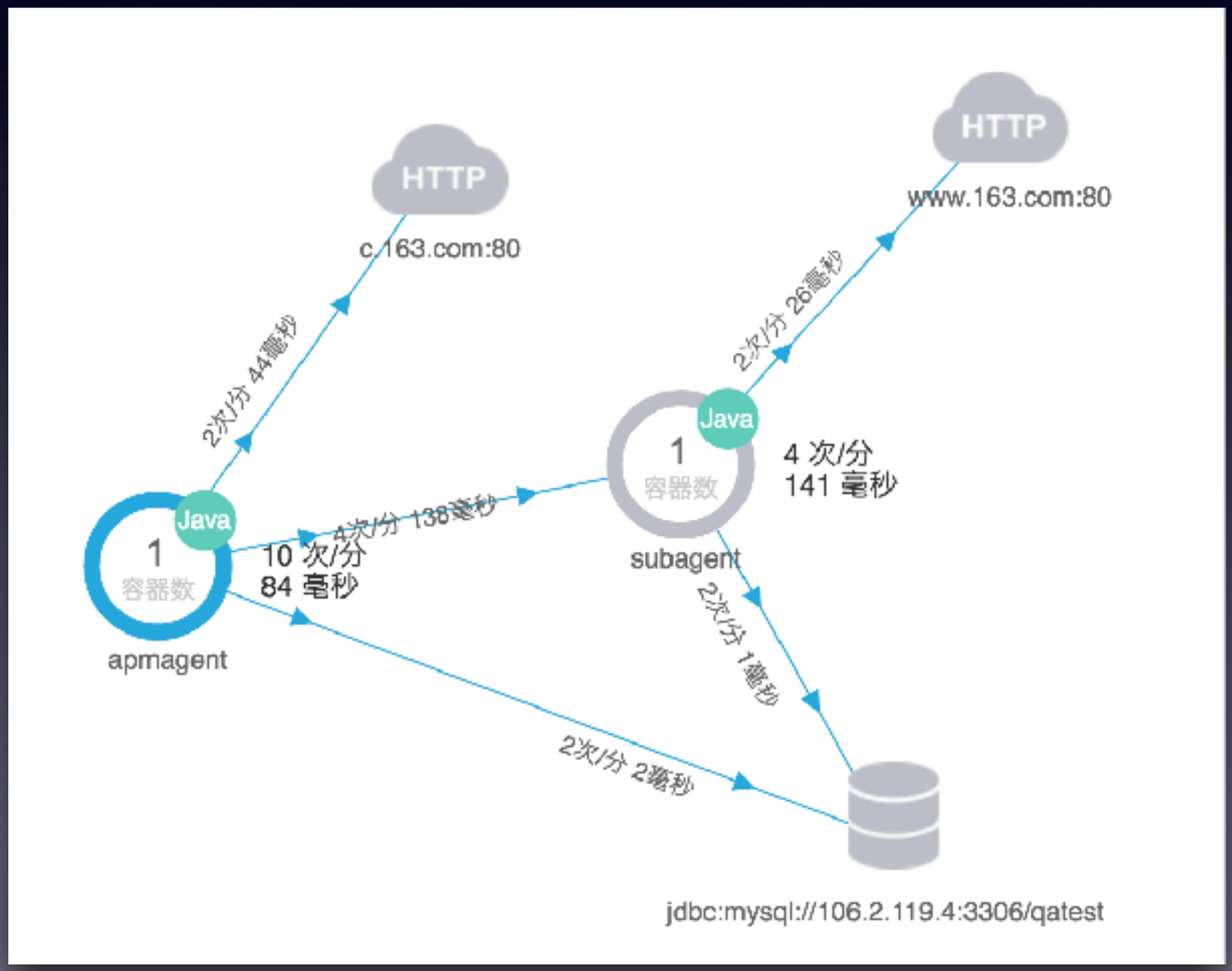
Stage4：服务端

- 全链路调用数据

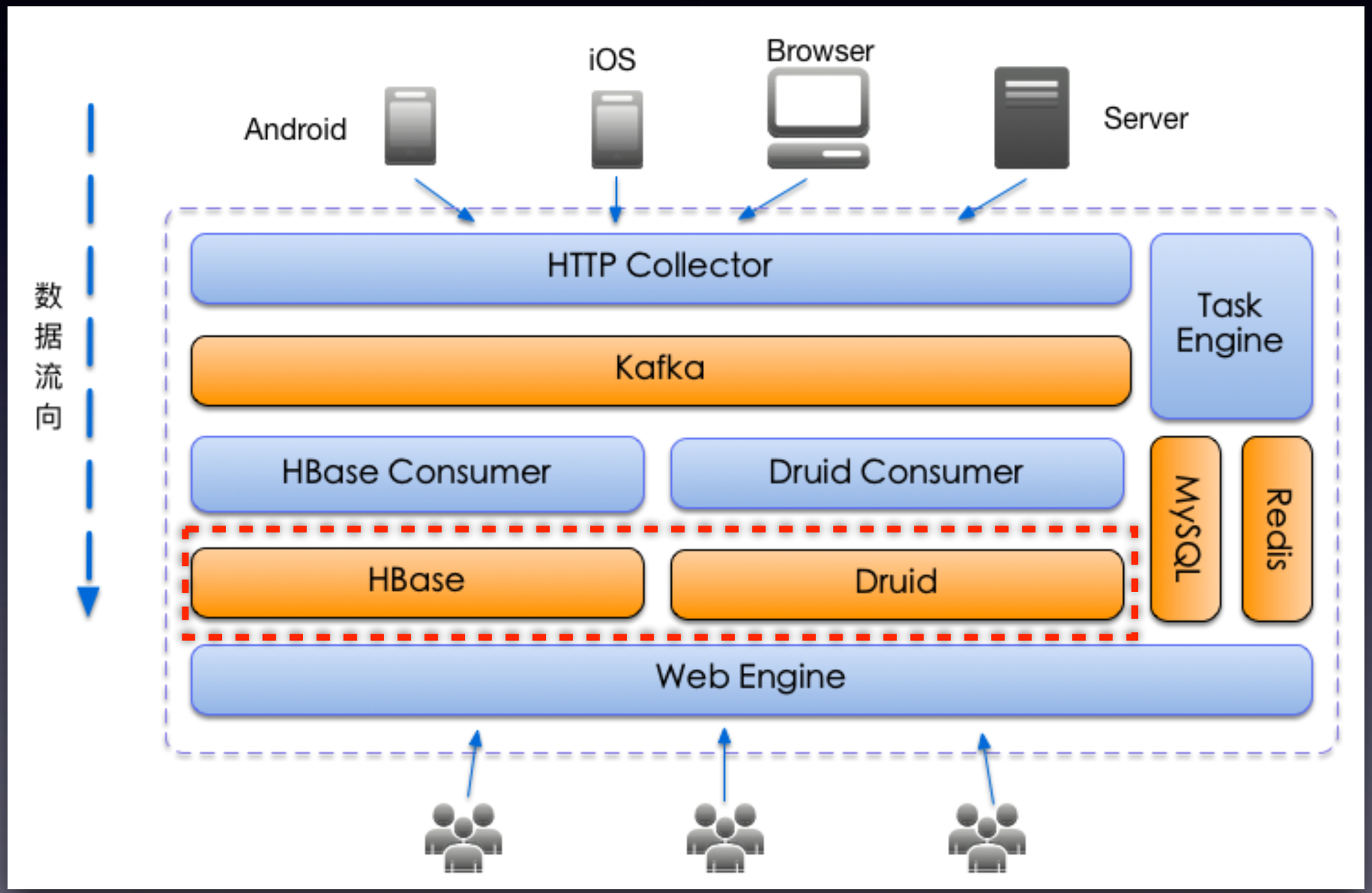


Stage4：服务端

- 全局拓扑图

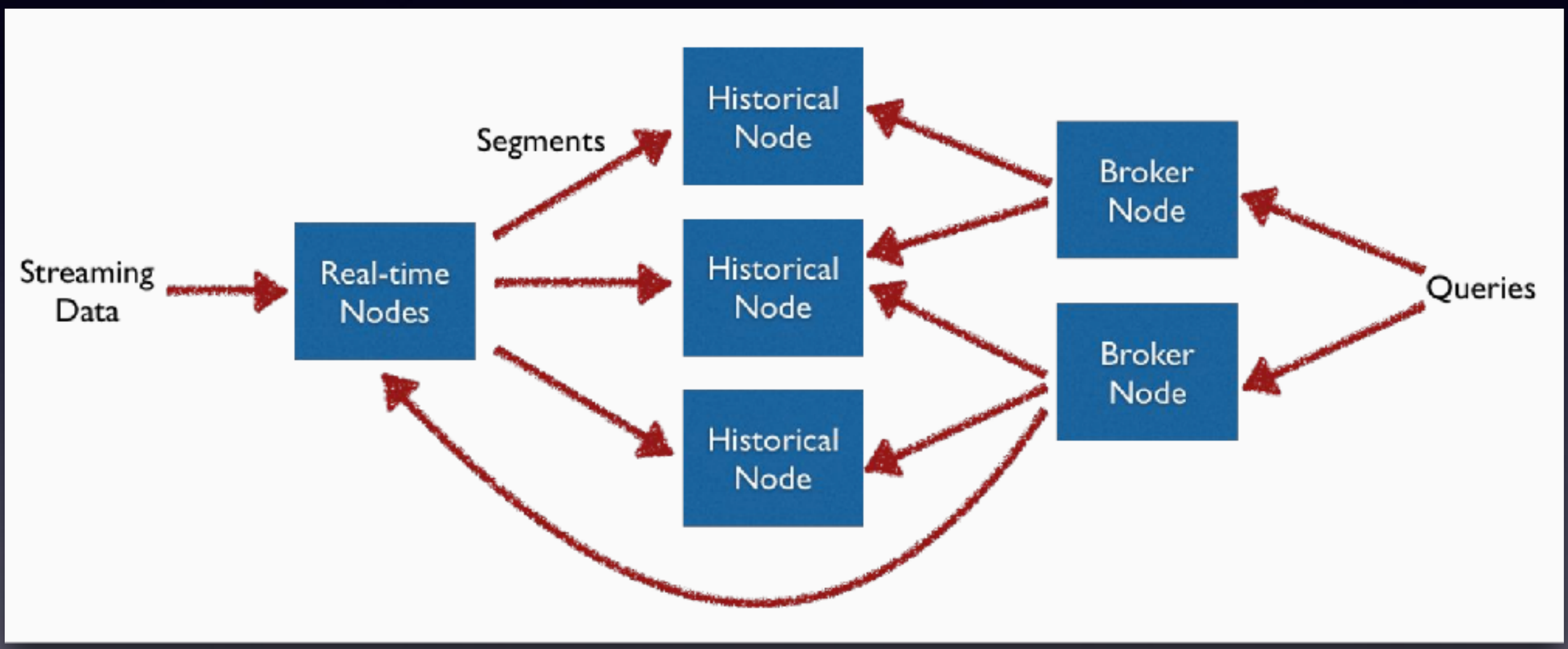


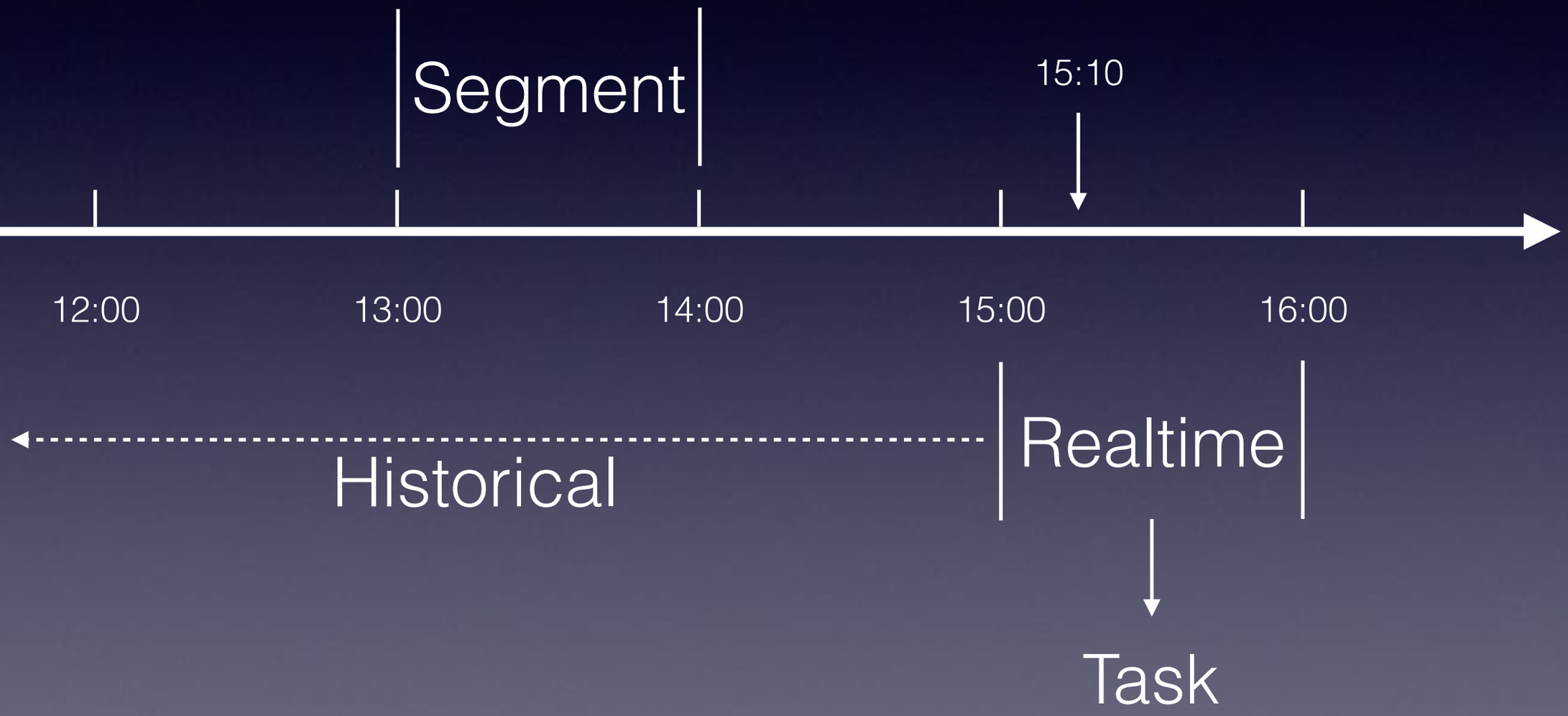
Stage4：架构


Android
iOS
Browser
Server
数据流向
HTTP Collector
Kafka
Task Engine
HBase Consumer
Druid Consumer
MySQL
Redis
HBase
Druid
Web Engine

千万 → 1亿 ——→ 10亿 ——→ 20+亿

Druid


Historical Node
Segments
Broker Node
Streaming Data
Real-time Nodes
Historical Node
Queries
Broker Node
Historical Node


Segment
15:10
12:00
13:00
14:00
15:00
16:00
Historical
Realtime
Task

Task

- 一个Task对应一个JVM进程
- 一个Task只能处理一个Datasource的数据
- 以机器为单位分配Task内存资源

碰到的问题

- 不同的DataSource资源需求不一样
  - 基于机器分配Datasource
- 相同的DataSource，但Task类型不一样，资源需求不一样：实时任务，批量任务（降采样）
  - 任务前更新配置，任务后恢复配置

碰到的问题

- 多租户
  - 共享Datasource，通过租户ID区分数据
  - 独立Datasource，每个租户一个Datasource

碰到的问题

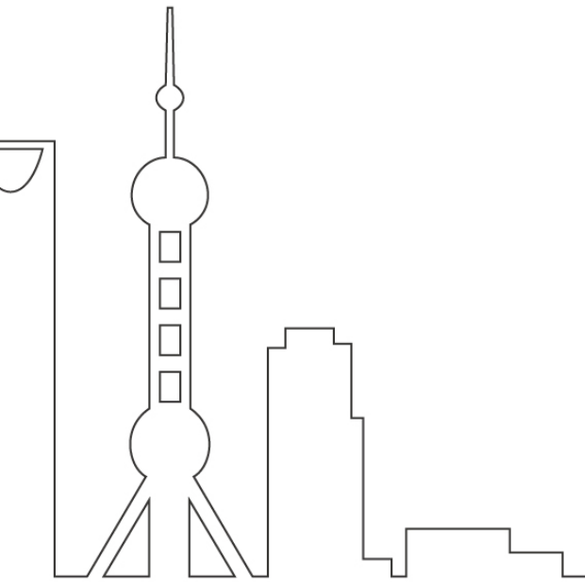
- 共享Datasource
  - 数据混合，没有物理隔离
  - 数据增长对性能的影响

碰到的问题

- 独立Datasource
  - 每个Datasource都需要有一个JVM
  - 资源需求不一致

改进思路

- 灵活的资源分配方案
- 灵活的任务分配方案



Thanks!

International Software Development Conference

主办方 Geekbang 极客邦科技 InfoQ