实时计算平台之挑战与演进

数据平台与应用部 实时计算团队

目录
CONTENTS

01 实时计算平台现状

02 实时平台的发展历程

03 实时计算的难点与挑战

04 实时平台的发展方向

01 实时计算平台现状

规模和提供的服务

实时平台现状

集群稳定性>99.99%

集群：650+节点

应用：300+

集群：350+节点

应用：200+

集群：35+节点

应用：10+

实时平台现状——业务规模

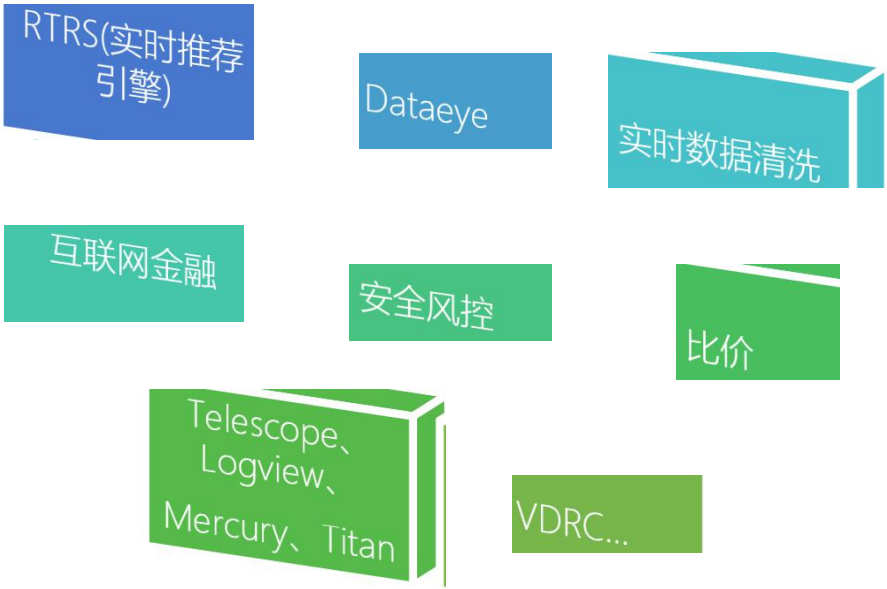
470万 条/s 实时数据采集，1~30s 埋点延迟，<100ms VDP延迟

4000万 条/s 实时计算量， 99.99% 可用

1300万QPS + 25TB 实时存储，<10ms 延迟

1700+ 服务器， 500+ 实时应用

实时平台现状——核心业务


RTRS(实时推荐引擎)
Dataeye
实时数据清洗
互联网金融
安全风控
比价
Telescope、Logview、Mercury、Titan
VDRC...

实时平台的职责——IS

实时计算平台

- 提供Storm、Spark、Flink等计算框架
- 监控、稳定性保障
- 开发支持

实时基础数据

- 提供基础数据（埋点、VDP数据）的清洗、打宽、质量保证
- 上游埋点的规范化和新埋点的定义

实时平台的职责——NOT

- 不提供各种API包装
- 以提供基础数据和平台为主，基本不提供数据汇总计算

提供API

- 提供技术协助
- 提供基础数据
- 应用开发由业务方自己完成

提供应用开发

实时平台的发展历程

不断前行

平台发展历程——早期

只提供Storm

无监控管理

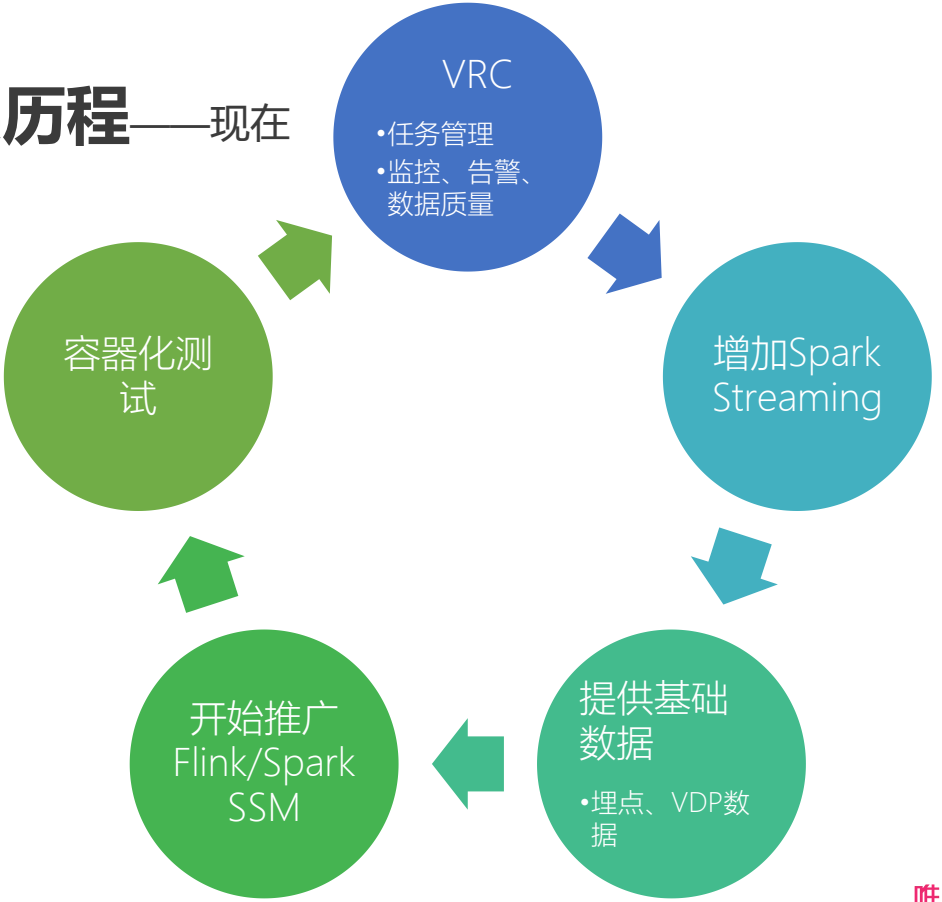
尝试SQL与UI开发

失败——先烈

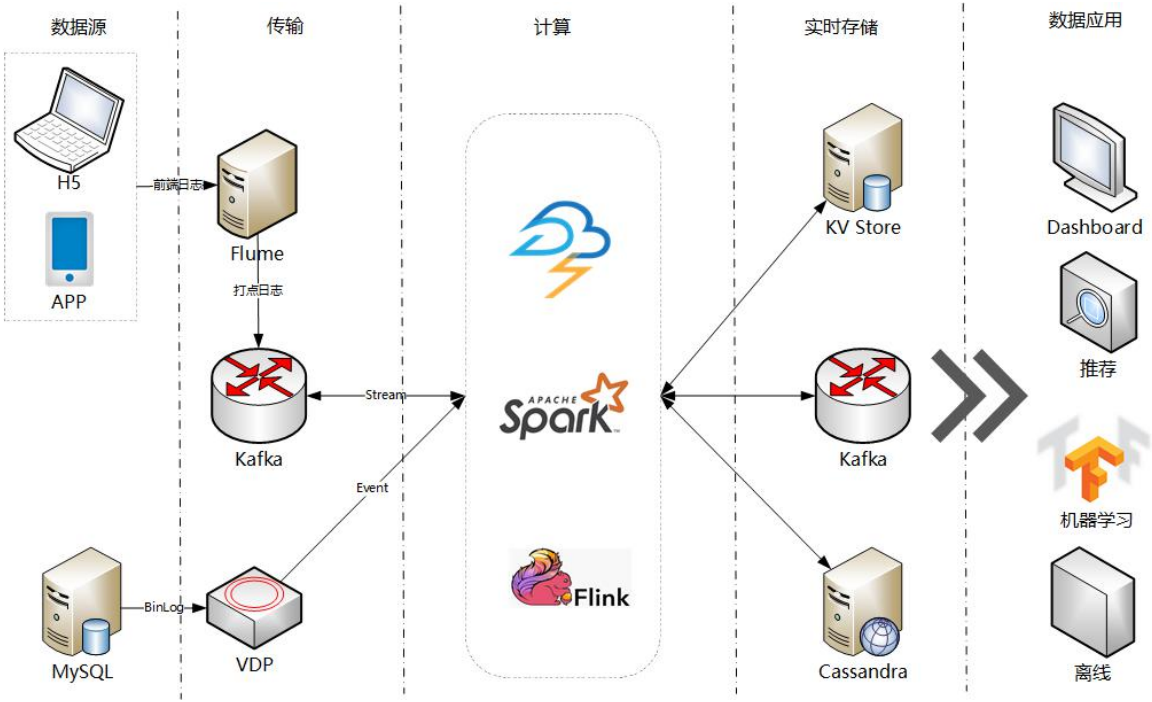
提供应用开发

资源严重不足

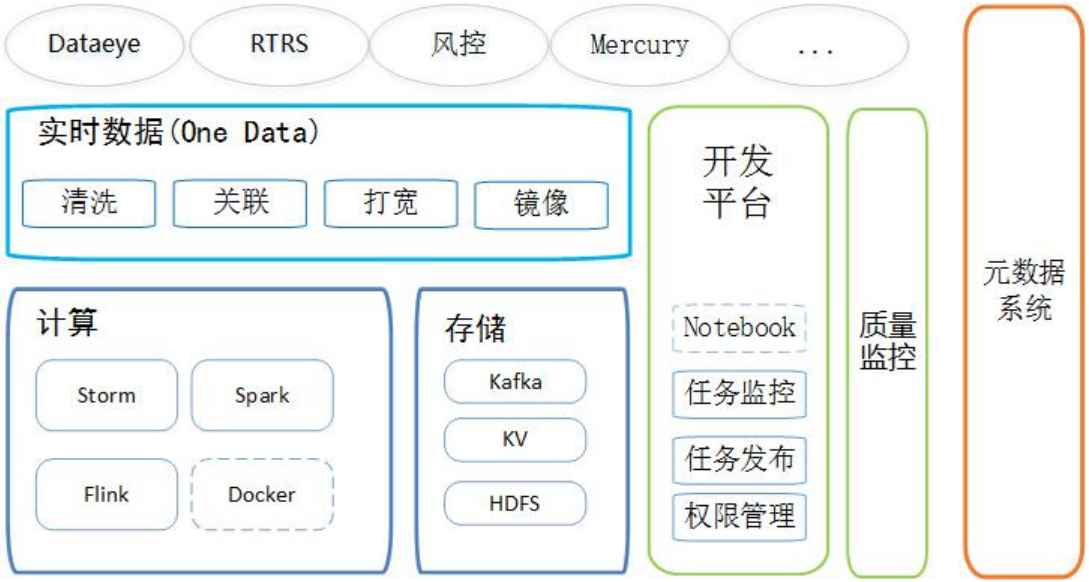
平台发展历程——现在


VRC
•任务管理
•监控、告警、数据质量
增加Spark Streaming
提供基础数据
•埋点、VDP数据
开始推广Flink/Spark SSM
容器化测试

平台发展历程——平台架构


数据源
传输
计算
实时存储
数据应用
H5
APP
前端日志
Flume
打点日志
Kafka
Stream
Event
MySQL
BinLog
VDP
Spark
Flink
KV Store
Kafka
Cassandra
Dashboard
推荐
机器学习
离线

平台发展历程——平台架构


Dataeye
RTRS
风控
Mercury
...
实时数据(One Data)
清洗
关联
打宽
镜像
计算
Storm
Spark
Flink
Docker
存储
Kafka
KV
HDFS
开发平台
Notebook
任务监控
任务发布
权限管理
质量监控
元数据系统

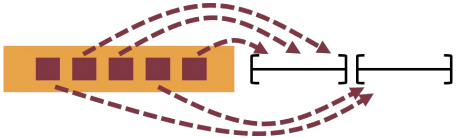
实时计算的难点与挑战

痛并快乐着

难点与挑战——技术挑战

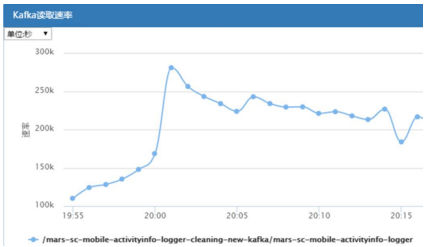
时序

- Event time vs. process time
- JOIN基本没法实现



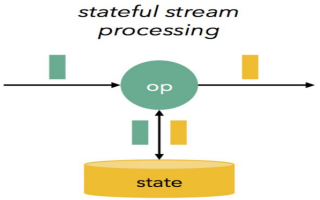
TPS、Latency

- 不能有积压



状态数量

- 从几十可能到上亿



难点与挑战——开发复杂度

思路转变

- 流 vs 静态表
  - e.g. 订单过程

工具链路长

- Kafka
- Storm/Spark Streaming
- Redis
- HBase

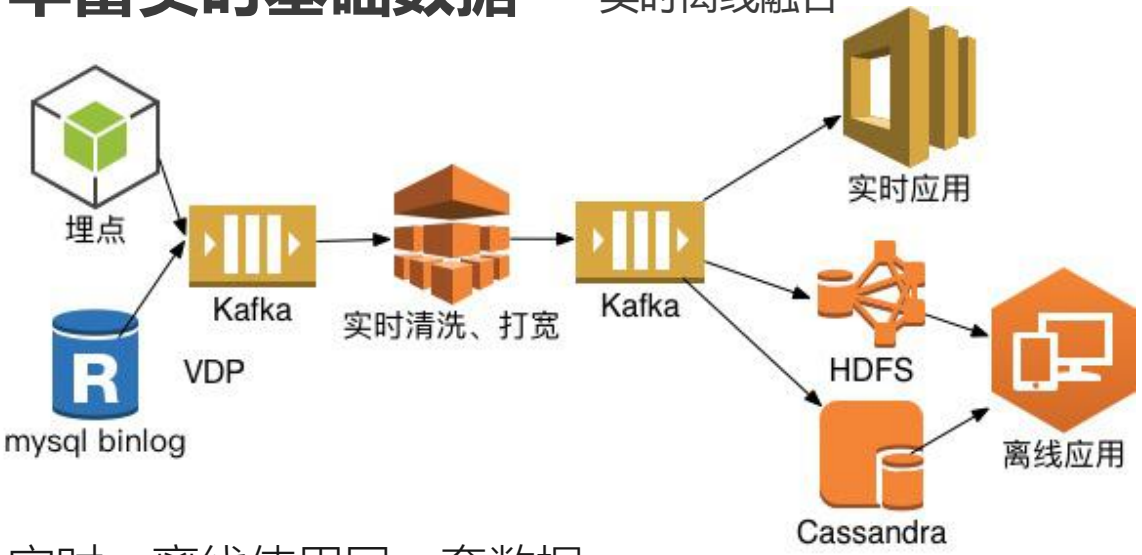
要考虑的问题多

- 并行度是否满足峰值流量?
- 状态数量? 存哪里?
- 几个流? 是否要JOIN?
- 对乱序的流如何处理?
- 输出频率?
- 如何对数?

实时平台的发展方向

降低开发门槛

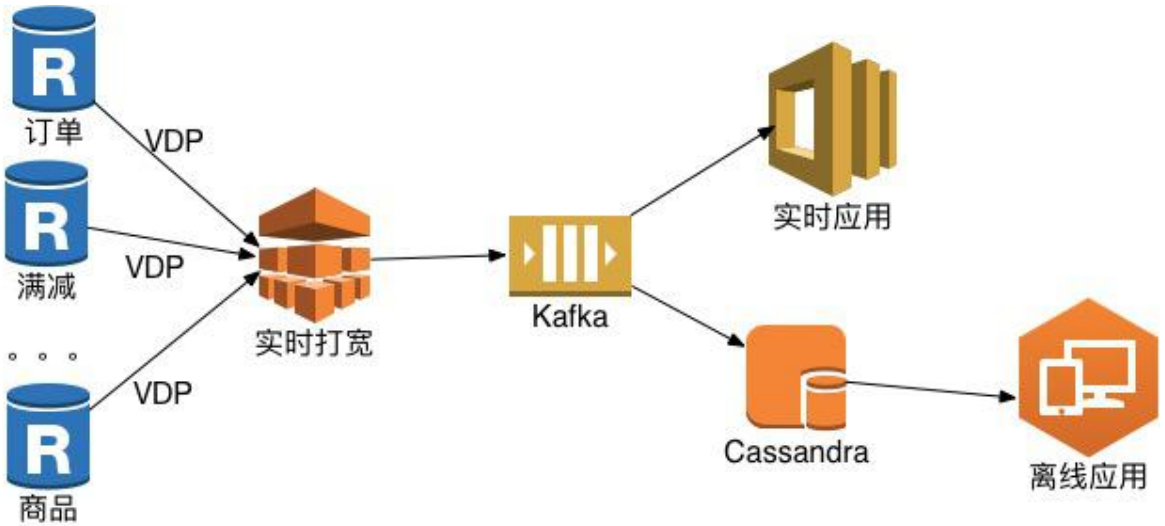
丰富实时基础数据——实时离线融合



实时、离线使用同一套数据
落地离线 xxx_5min Hive表

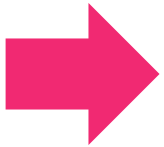
用离线SQL来解决那些需要分钟级时延的需求

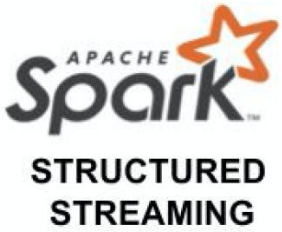
丰富实时基础数据——订单打宽



✓ 解决输入的乱序、时效（秒级）、数据量（全表）、用户消费顺序

新的开发框架





- 代码简化 100X
- 自带状态管理
- 语义保证

预研notebook开发环境、Streaming SQL开发

统一数据源——开发方式

实时、离线Spark代码开发

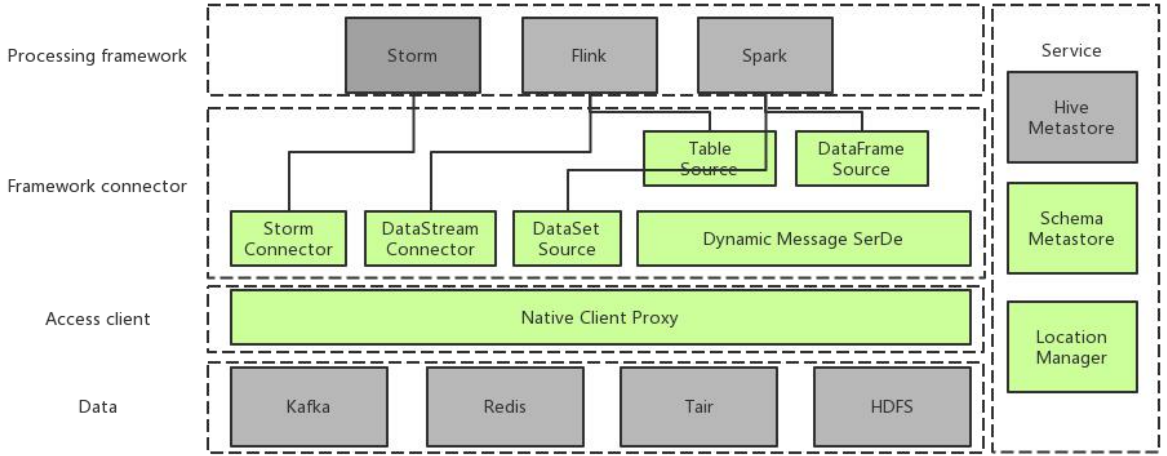
```
// Spark read from Kafka as typed data frame
val dataFrame = spark.readStream
  .option("namespace", "kafka1")
  .option("topic", "pageview")
  .load

dataFrame.groupBy("brandId", "start_time").count()
```

SQL开发（实时、离线）

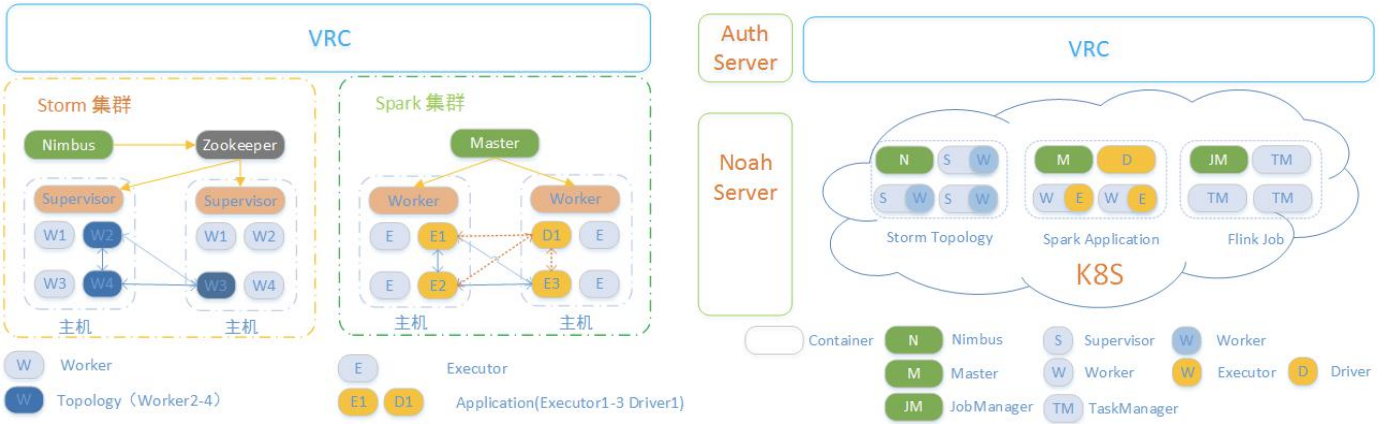
```
select brandId, start_time, count(*) as c from kafka1.pageview group by brandId, start_time;
```

统一数据源——架构



让实时、离线、机器学习可以无缝的访问到所有的数据

实时平台容器化


VRC
Storm 集群
Nimbus
Zookeeper
Supervisor
Supervisor
W1
W2
W1
W2
W3
W4
W3
W4
主机
主机
Spark 集群
Master
Worker
Worker
E
E1
D1
E
E
E2
E3
E
主机
主机
W
Worker
W
Topology（Worker2-4）
E
Executor
E1
D1
Application(Executor1-3 Driver1)
Auth Server
VRC
Noah Server
N
S
W
S
W
S
W
M
D
W
E
W
E
JM
TM
TM
TM
Storm Topology
Spark Application
Flink Job
K8S
Container
N
Nimbus
S
Supervisor
W
Worker
M
Master
W
Worker
W
Executor
D
Driver
JM
JobManager
TM
TaskManager

感谢聆听

—

THANKS!

唯品会
一家专门做特卖的网站