Flink China

Flink在唯品会的实践

王新春

唯品会-数据平台与应用部-实时平台

实时平台现状

Flink China

实时平台现状

集群稳定性>99.99%

Storm

集群：650+节点

应用：300+

Spark

集群：350+节点

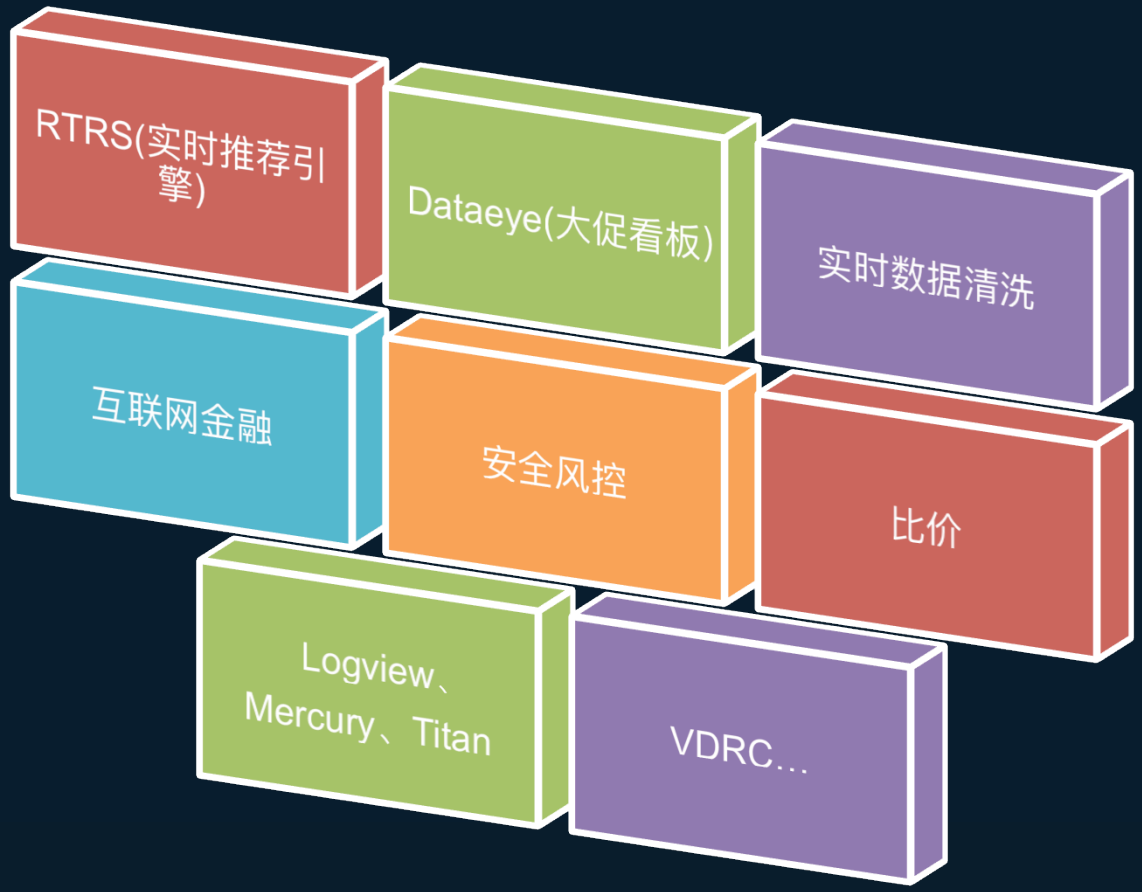
应用：200+

Flink

集群：80+节点

应用：30+

实时平台现状——核心业务


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

Flink China

实时平台的职责

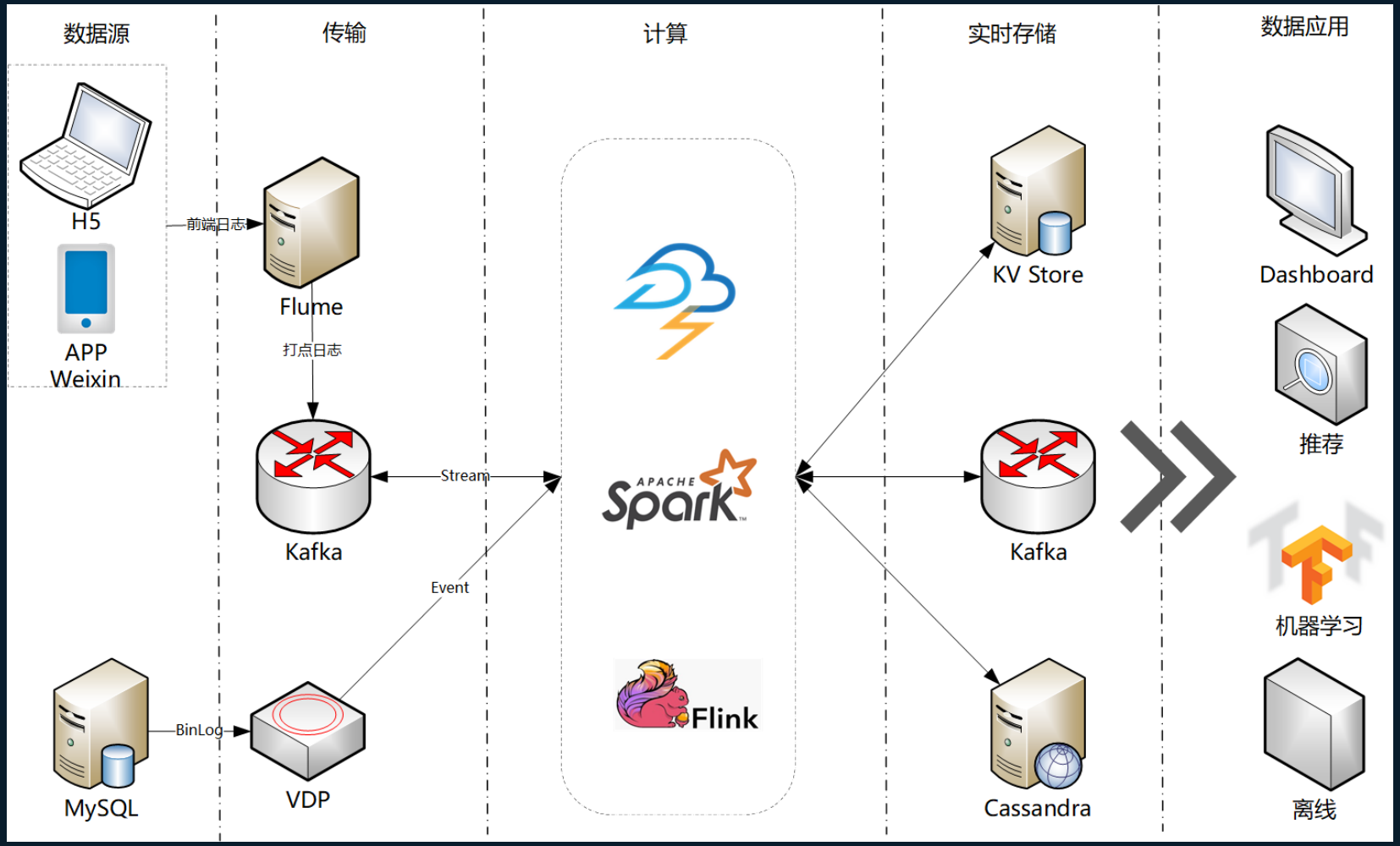
实时计算平台

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

实时基础数据

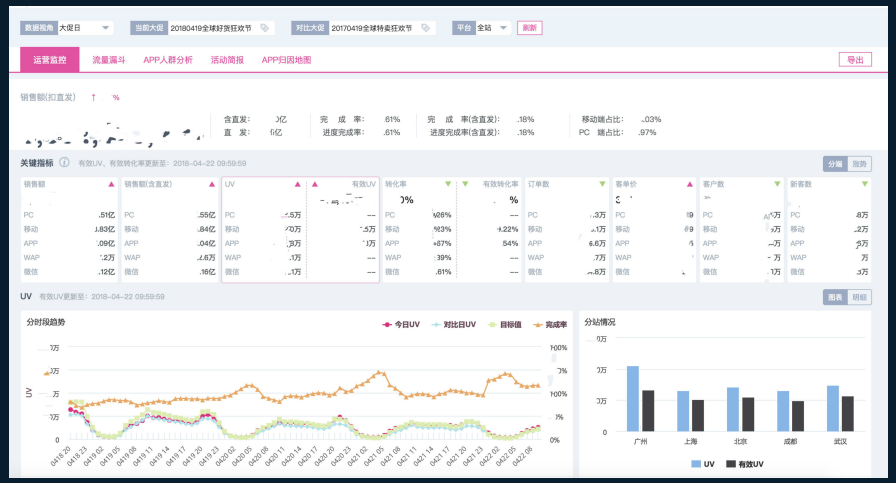
- 提供基础数据（埋点、Binlog数据）的清洗、打宽、质量保证
- 上游埋点的规范化和新埋点的定义

实时平台架构


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

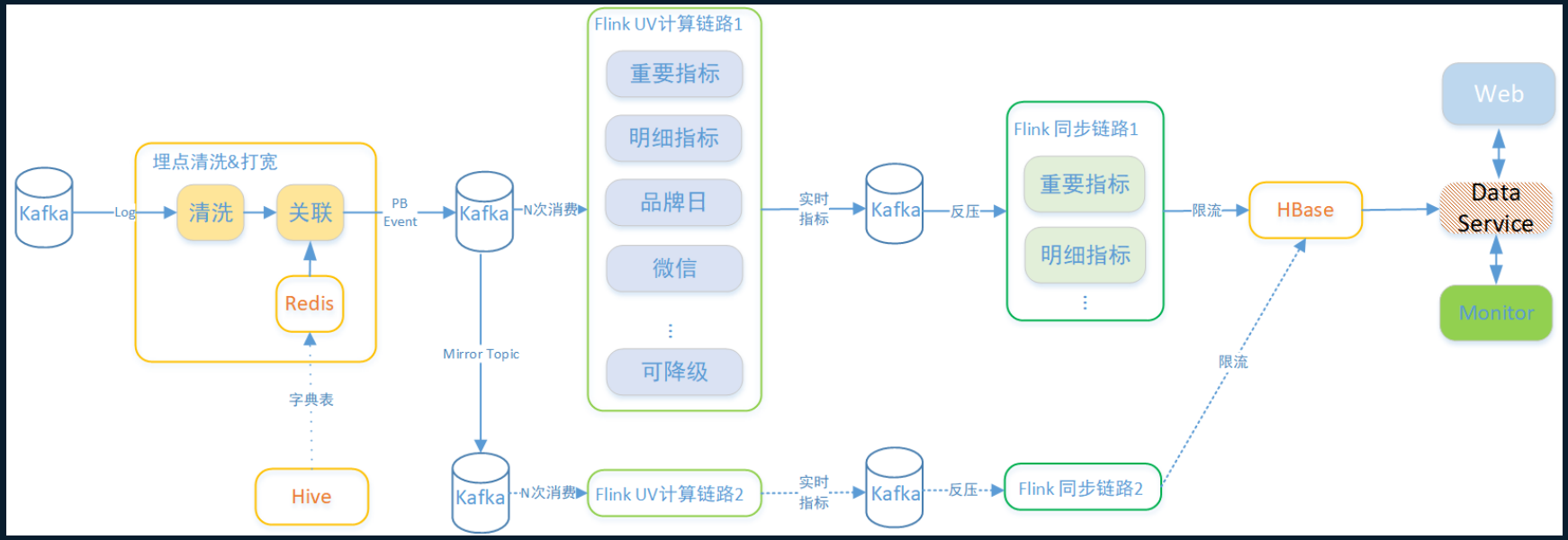
Flink的实践

场景1：Dataeye实时看板



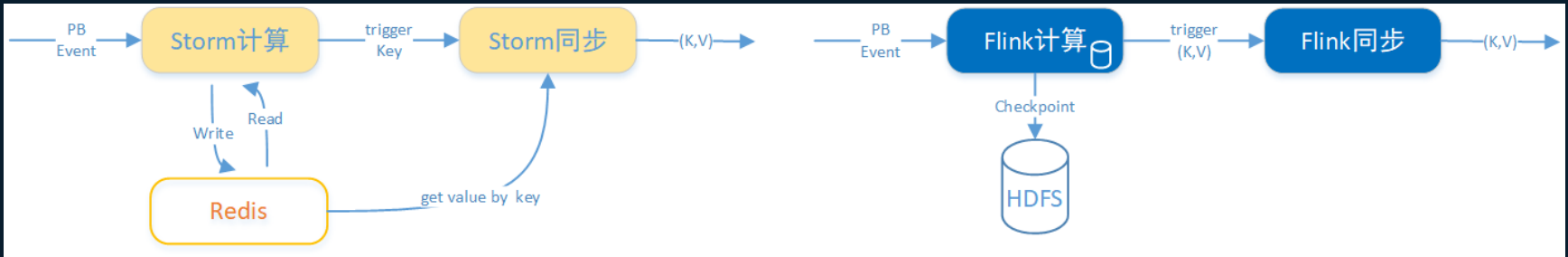
- 任务特点：
  - 数据量大
  - 统计维度多：全站、二级平台、部类（一级、二级）、档期、人群、分站、分省、活动、时间维度（大促、天、小时）、新客...

Dataeye实时看板——UV计算



- 计算任务Storm迁移到Flink
  - 任务双跑
  - 计算任务和同步任务分离
  - Data Service层自动切换使用那一路任务的数据

UV计算效率提升



UV计算Storm->Flink

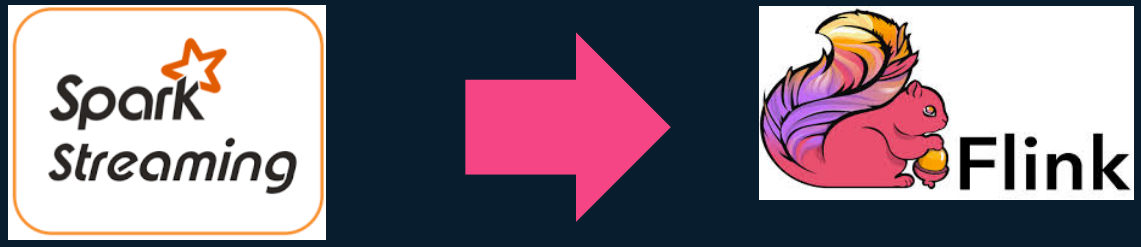
稳定性可靠性提升；计算资源消耗降低2/3

Flink China

UV计算问题点

- FlinkKafkaConsumer
  - 支持offsetAutoCommit
  - *支持Kafka集群切换：恢复任务的时候指定offset消费，skip checkpiont的state（todo）*
- FlatMapFunction
  - 不带Window的State数据需要手动清理
- 数据倾斜处理
  - keyBy容易倾斜
- 同步任务追数
  - Storm计算：根据Key自动取Redis里面的统计指标的最新值
  - Flink计算：等！

场景2: Kafka数据落地HDFS



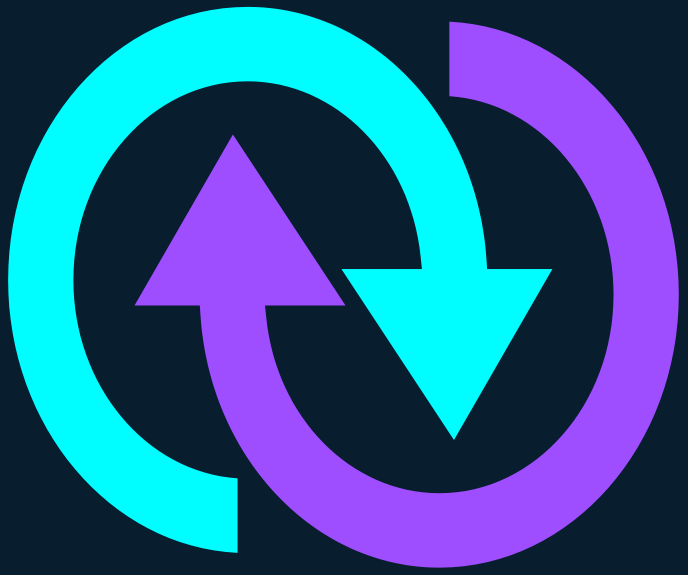
- 实现*OrcBucketingTableSink*将埋点数据5分钟落地HDFS为Hive表
- 计算资源消耗降低90%；延迟30s降低到3s以内
- 单Task Write ~3.5K/s
- *支持Spark Bucket Table (todo)*

Flink China

场景3：实时ETL——Stream Join Batch

HDFS字典表

OrcFormatStream TableSource



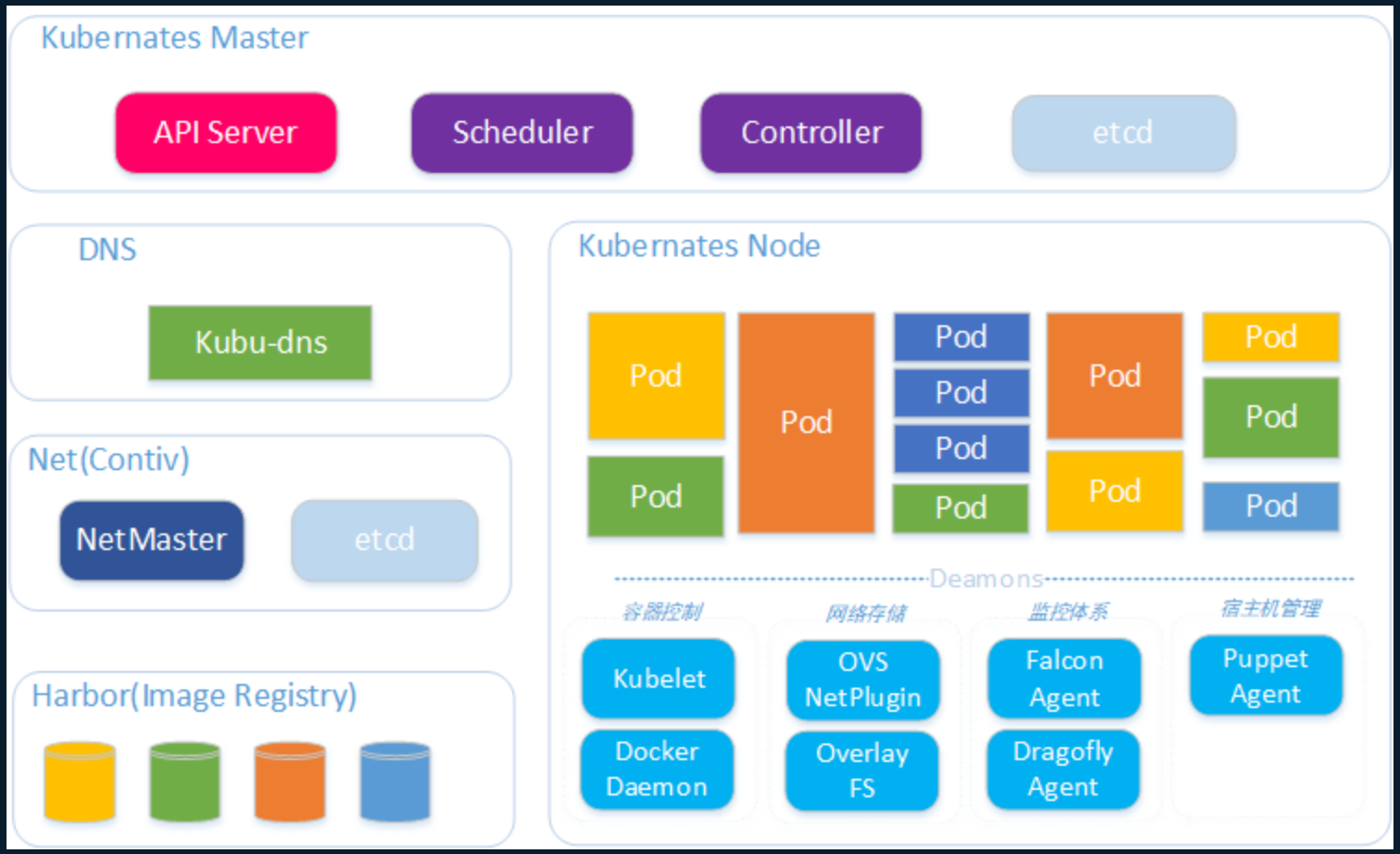
实时数据

Kafka08TableSource

- 使用ContinuousFileMonitoringFunction和ContinuousFileReaderOperator定时监听HDFS数据变化
- *支持Hive表和Stream Join(todo)*

Flink On K8S

统一计算资源——基于Kubernates管理实时和AI平台



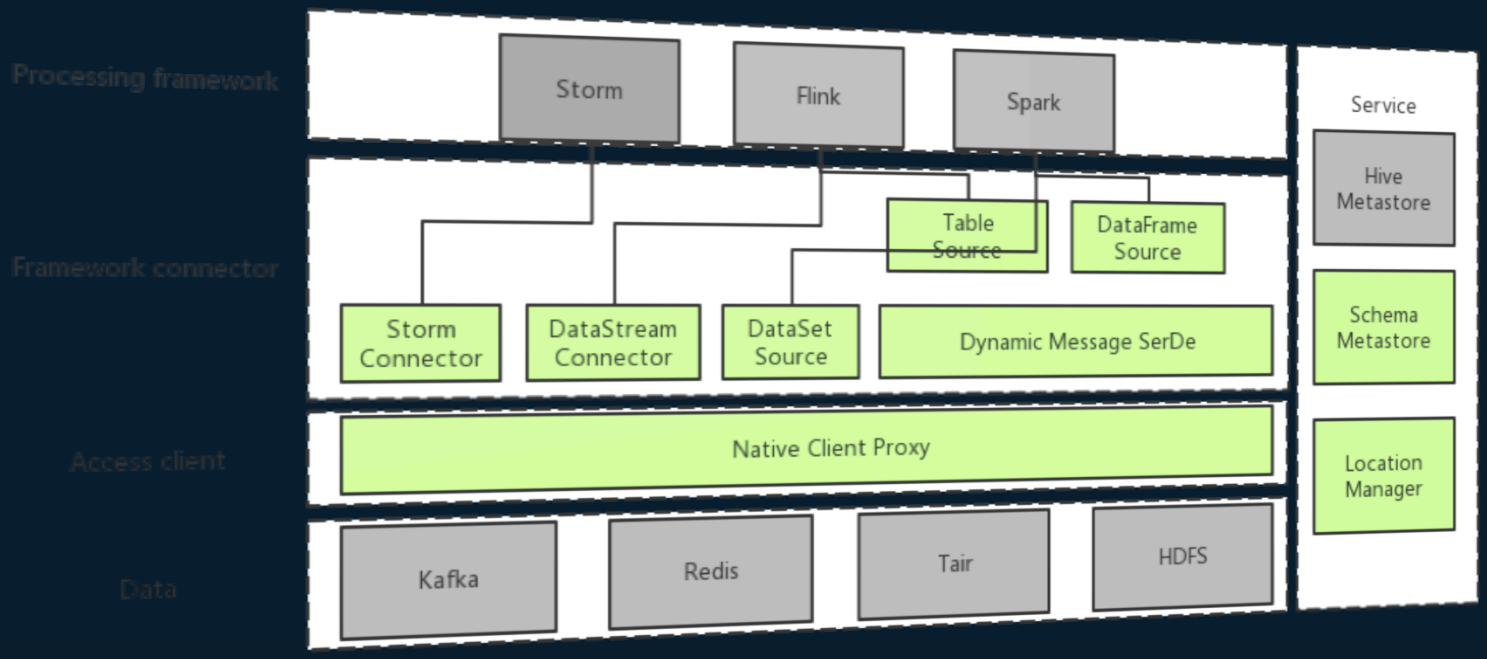
Flink China

Flink On K8S

- 使用Kubernates StatefulSet模式部署
  - FlinkClusterInfo createFlinkCluster(String namespace,
    String clusterName,
    String version,
    Integer taskManagers,
    Integer cpuCorePerTaskManager,
    Integer memoryPerTaskManager) throws Exception;
- 一个Job一个mini cluster，支持HA
  - JobManager：job-[0,1].job.flink.svc.rt.vip.vip.com
  - TaskManager：job-[2..n].job.flink.svc.rt.vip.vip.com
- docker-entrypoint.sh根据环境变量设置flink-conf.yaml
  - jobmanager.rpc.address
  - jobmanager/taskmanager.heap.mb
  - taskmanager.numberOfTaskSlots
  - high-availability.zookeeper.quorum
  - high-availability.cluster-id
  - high-availability.storageDir
  - state.backend.fs.checkpointdir

Doing

统一数据源——UDM架构


Processing framework
Framework connector
Access client
Data
Storm
Flink
Spark
Table Source
DataFrame Source
Storm Connector
DataStream Connector
DataSet Source
Dynamic Message SerDe
Native Client Proxy
Kafka
Redis
Tair
HDFS
Service
Hive Metastore
Schema Metastore
Location Manager

Flink China

统一数据源——基于UDM的开发模式

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;
```

Flink China

统一数据源——Flink相关特性和工作

- UDMExternalCatalog
  - VDP(MySQL Binlog Pipeline) Schema注册
  - Hive Table Schema管理
  - Kafka protobuf数据格式Schema
  - Redis/Tair 数据Schema注册
- VDPSource开发
- KafkaSource增强
- HiveTableSource开发
- 实时数据和Hive表Join场景下，由离线任务调度系统通知Hive表数据变更


Flink China
Thanks !