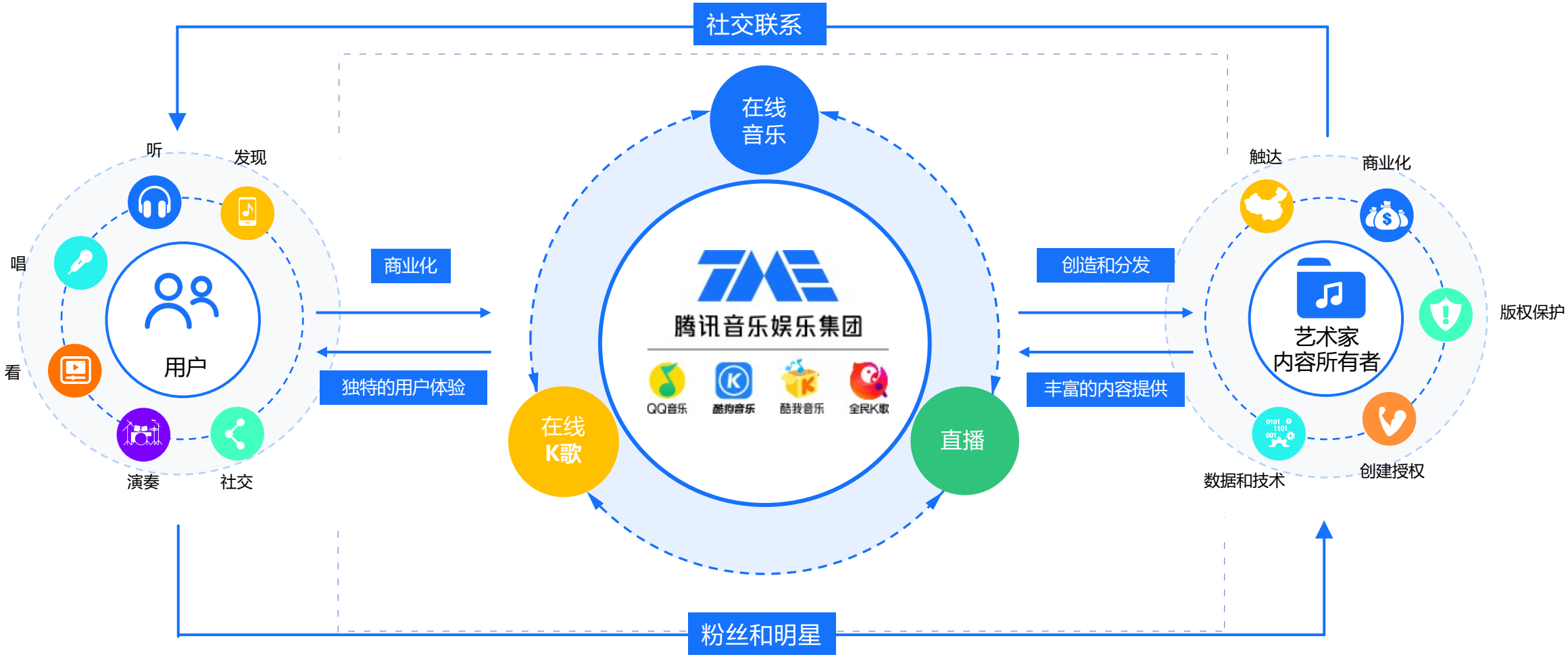
ClickHouse在腾讯音乐敏捷数据分析中的实践和思考

zelus

2021.02

生态全景



一体化的音乐娱乐，高参与度、强社交性和有趣的用户体验

目 录

ClickHouse 小调查

call 1 简单搭建使用，性能测试对比

call 2 落地生产环境大规模应用

call 3 社区源码贡献者

01 快速的业务迭代与激增的数据需求

数据需求中心化处理瓶颈

内容 FT

增长 FT

直播 FT

商业化 FT

……FT

2x+ 业务研发小组

激增的数据需求

VS

X个 数据工程师

BI数据分析

数据科学

内容推荐

增长决策

……

x+ 数据应用团队

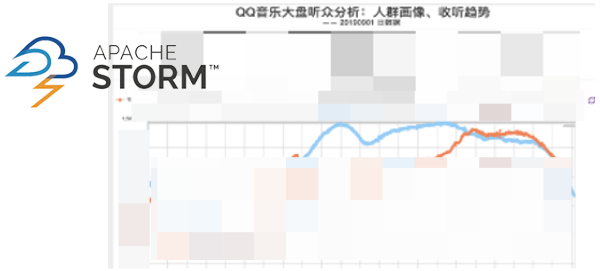
02 数据平台演进实践

数据平台架构演进

1.0 BI数据分析

- 数据需求驱动，罗盘BI数据报表建设

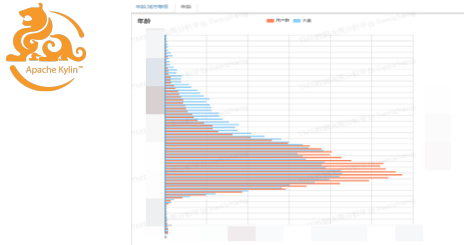
- 探索TRC（Storm）实时计算



2.0 业务数据分析平台

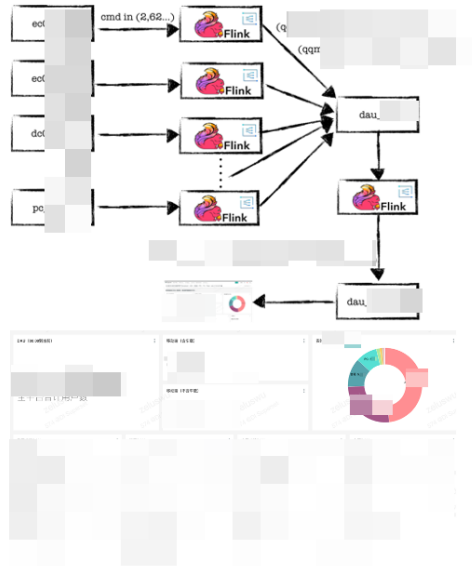
- 大规模实时流式内容分析计算应用

- Hermes画像在线分析千万级用户圈层
- Kylin预计算解决内容类多维度分析需求

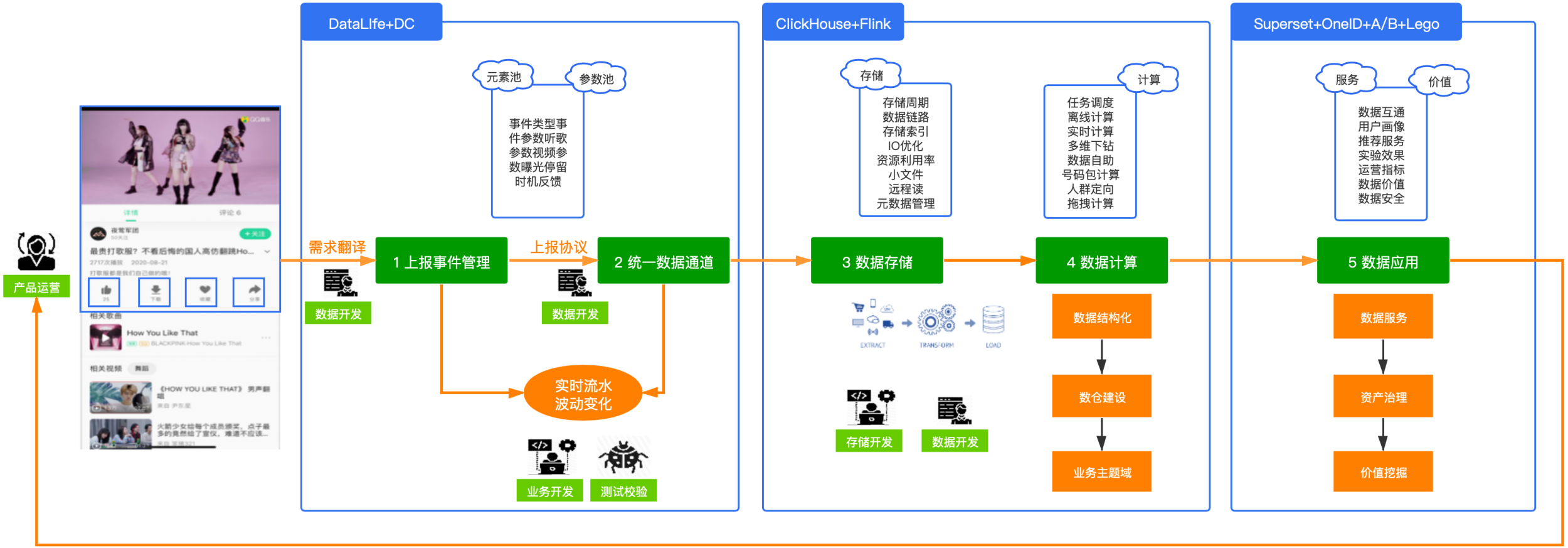


3.0 自助数据分析平台

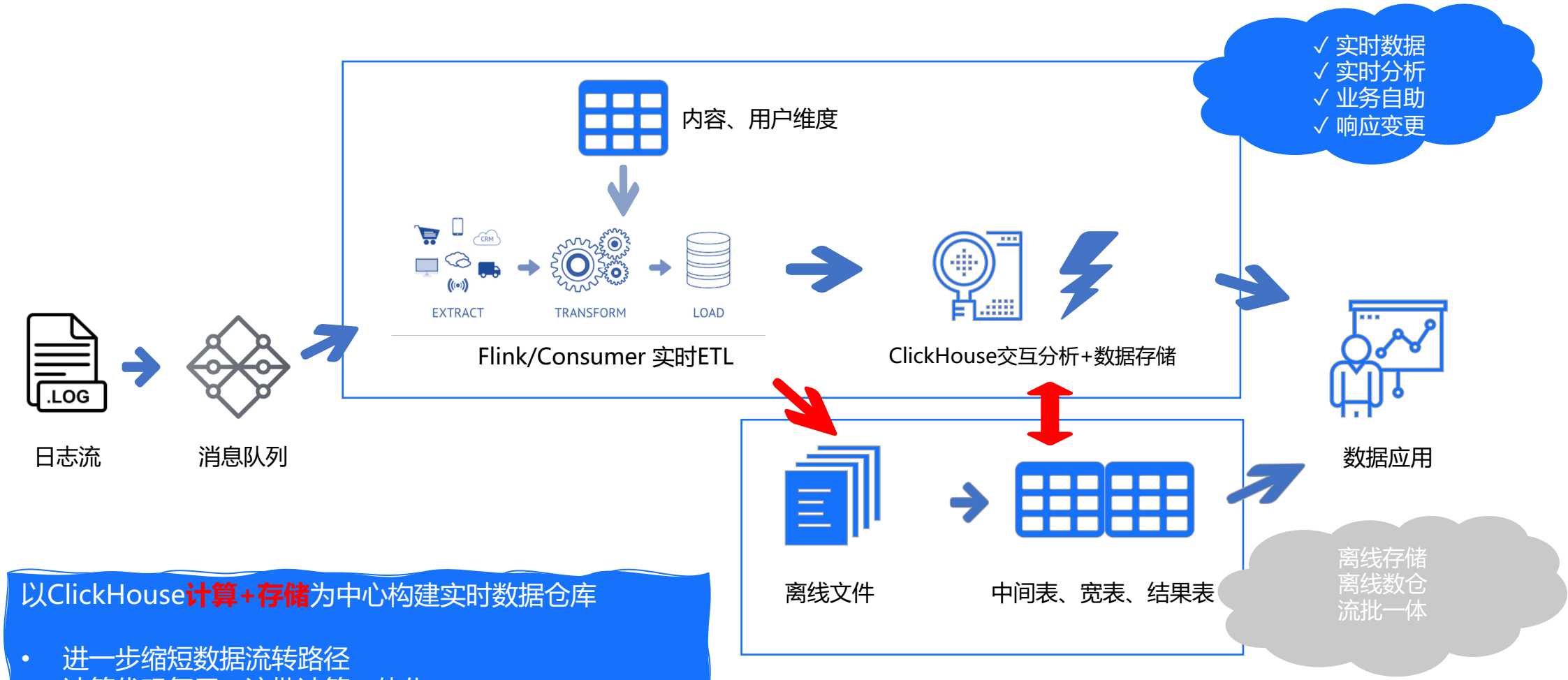
- Clickhouse同时满足大数据量即席查询与实时计算
- Superset一站式的数据探索+数据可视化平台
- Flink实时基础实时数仓建设



数据全链路生产加工


DataLIfe+DC
ClickHouse+Flink
Superset+OneID+A/B+Lego
元素池
参数池
事件类型事件参数听歌参数视频参数曝光停留时机反馈
存储
存储周期
数据链路
存储索引
IO优化
资源利用率
小文件
远程读
元数据管理
计算
任务调度
离线计算
实时计算
多维下钻
数据自助
号码包计算
人群定向
拖拽计算
服务
价值
数据互通
用户画像
推荐服务
实验效果
运营指标
数据价值
数据安全
产品运营
需求翻译
1 上报事件管理
上报协议
2 统一数据通道
3 数据存储
4 数据计算
5 数据应用
数据开发
数据开发
实时流水
波动变化
EXTRACT
TRANSFORM
LOAD
存储开发
数据开发
数据结构化
数仓建设
业务主题域
数据服务
资产治理
价值挖掘
业务开发
测试校验

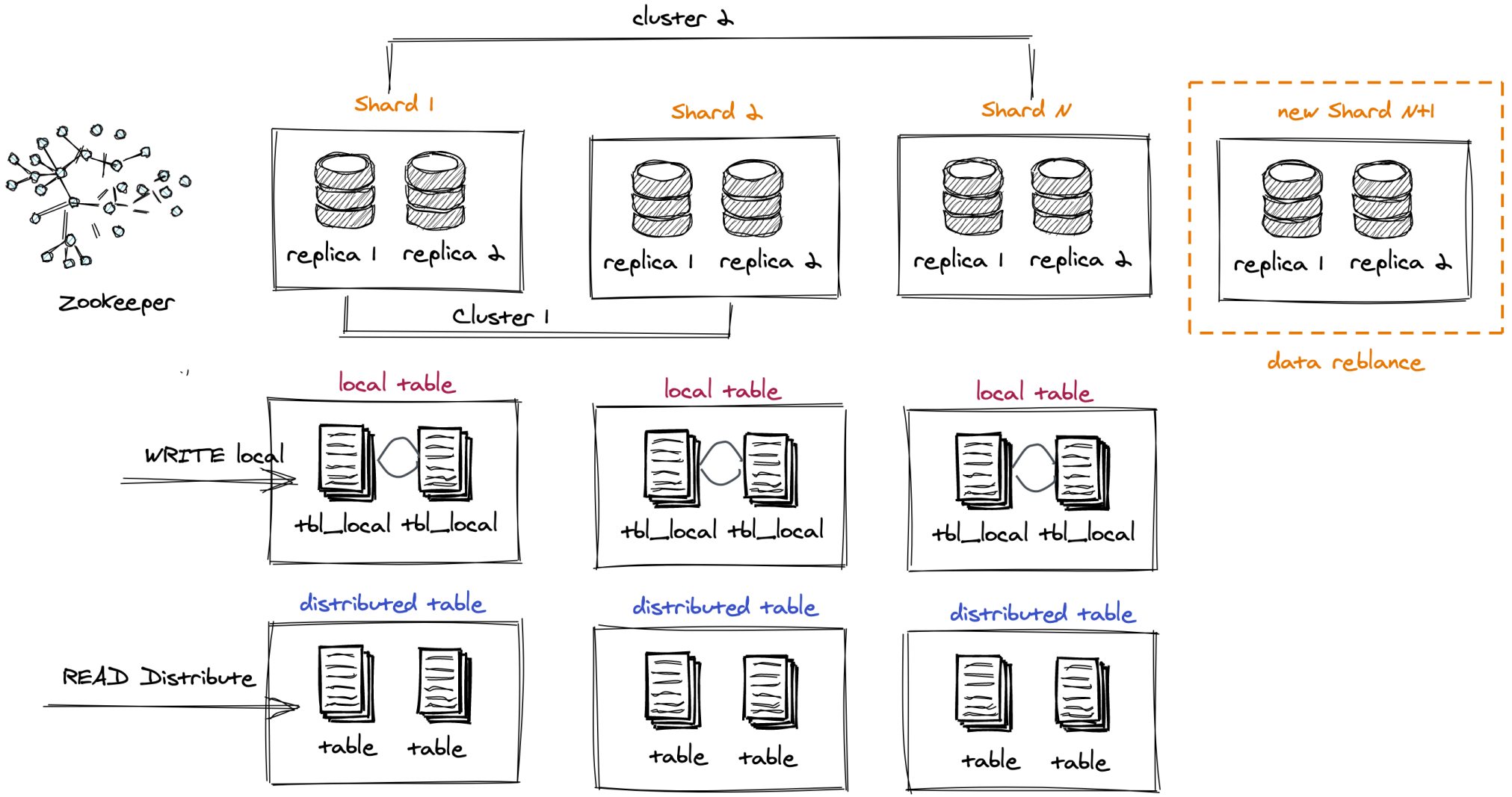
3.0交互式数据流批一体计算



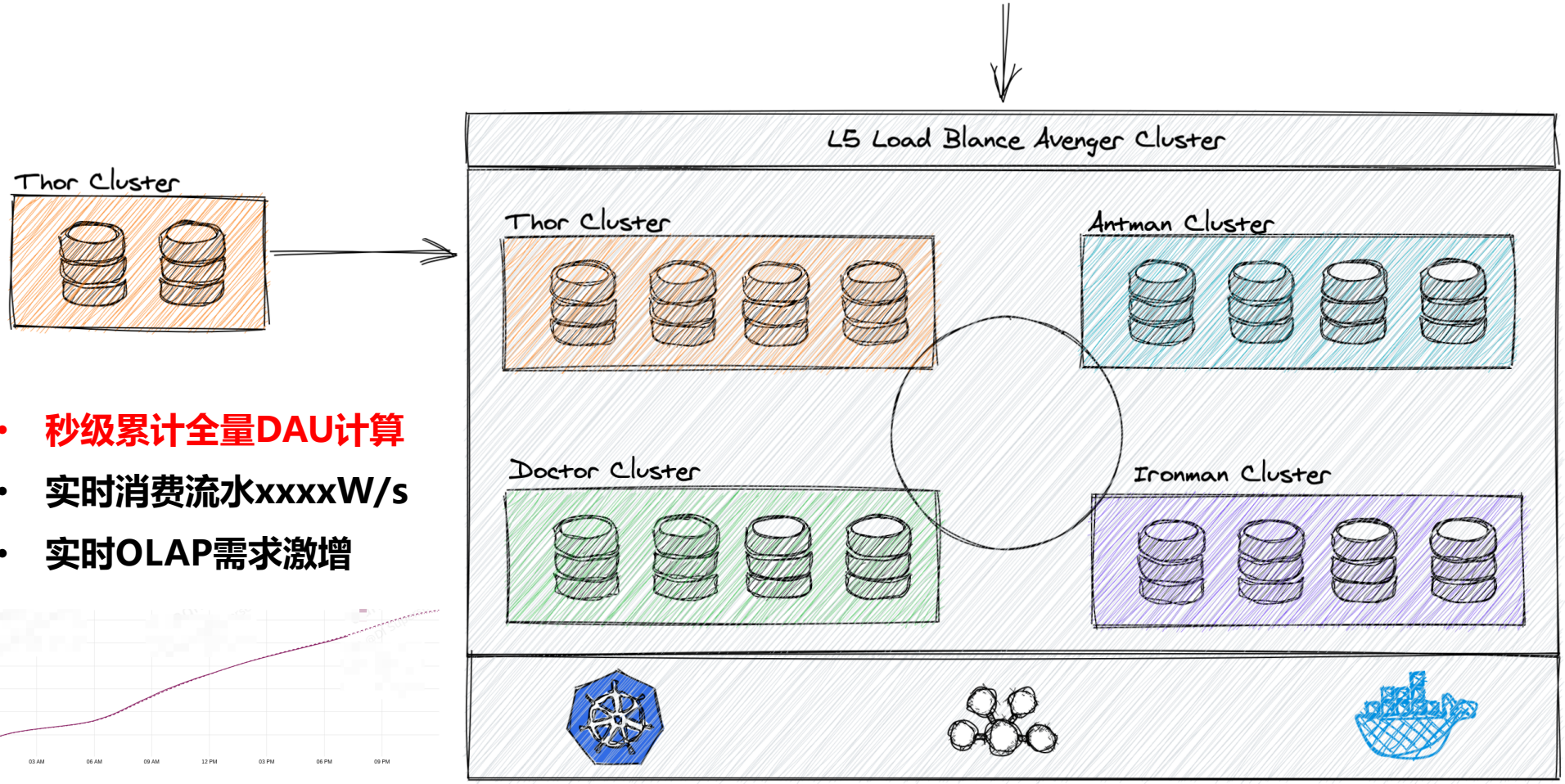
以ClickHouse**计算+存储**为中心构建实时数据仓库

- 进一步缩短数据流转路径
- 计算代码复用，流批计算一体化
- 离线大规模存储

K8容器化多集群架构部署



K8容器化多集群架构部署



- **秒级累计全量DAU计算**
- **实时消费流水xxxxW/s**
- **实时OLAP需求激增**

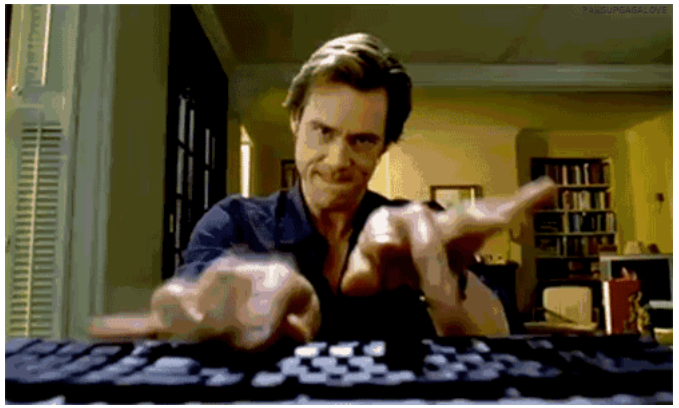
1. **数据多分片**
2. **多数据中心**
3. **多集群相互可见**
4. **K8容器化管理**
5. **状态存活检测**
6. **Server熔断**
7. **资源限制**
8. **负责均衡**
9. **多维度监控**

为什么用K8s

v1.0

clickhouse(Docker手动管理)

juno(bash手动管理)



Log	Node-X	Ch-X	Event-X
日志	节点监控	ch监控	事件监控

监控报警

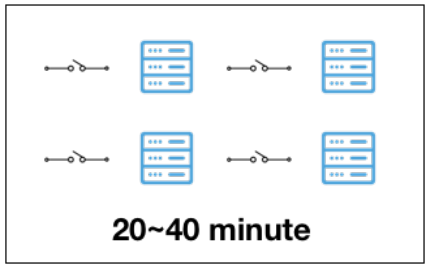
工作重心转移

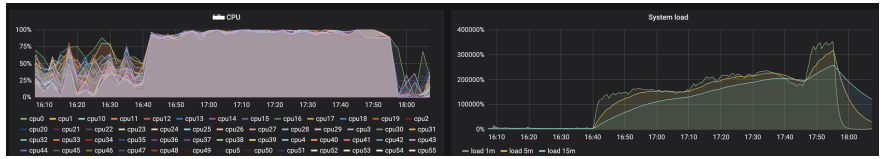
Clickhouse

自动化熔断修复

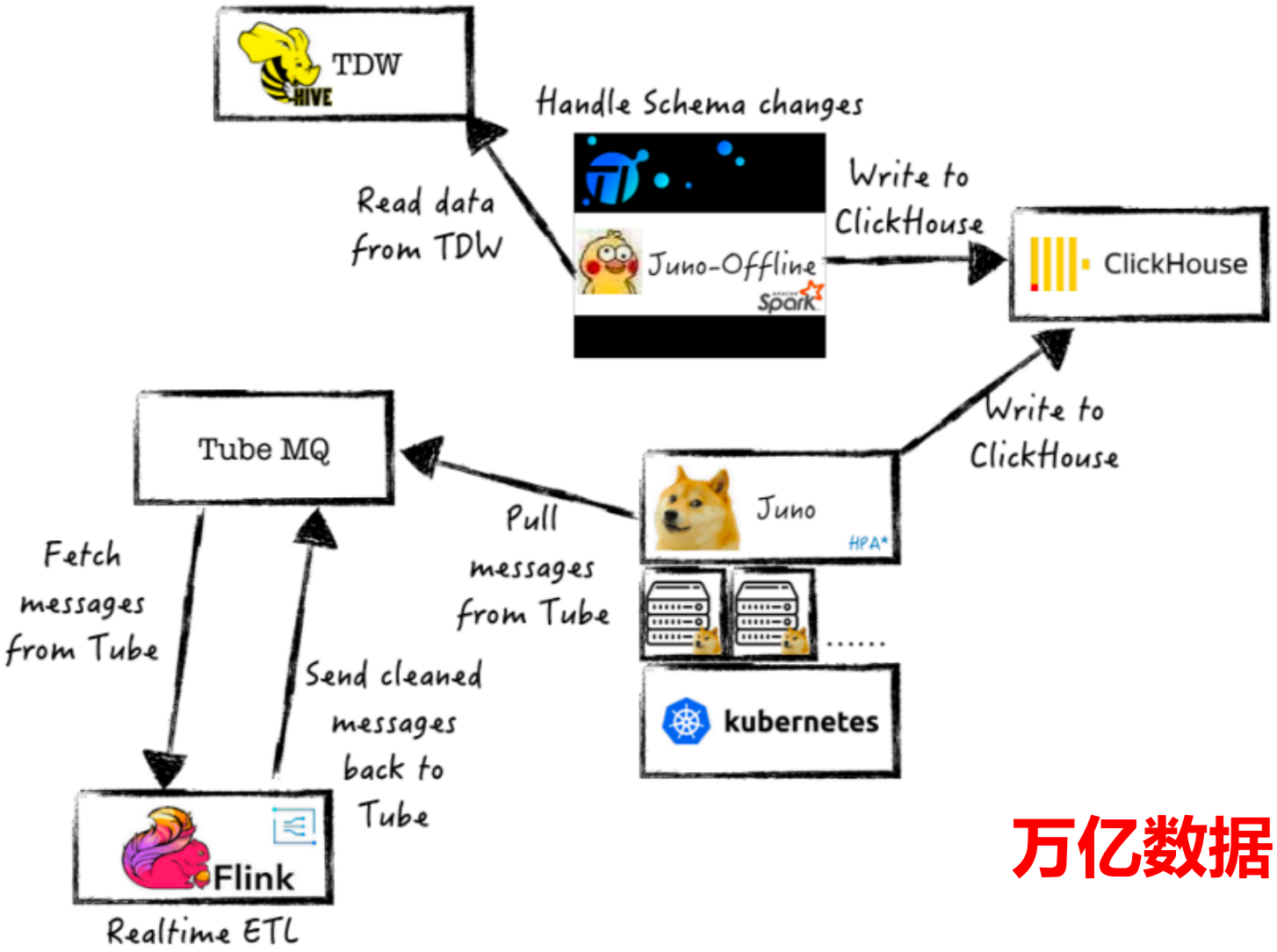
限 资源使用

熔 断/就绪检测



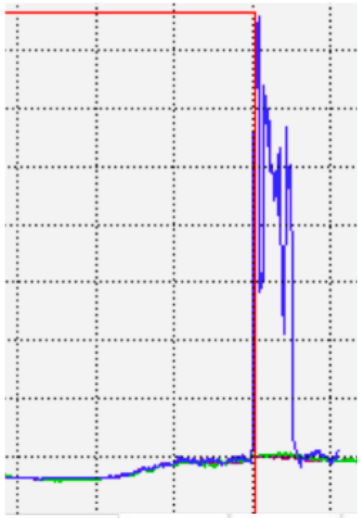


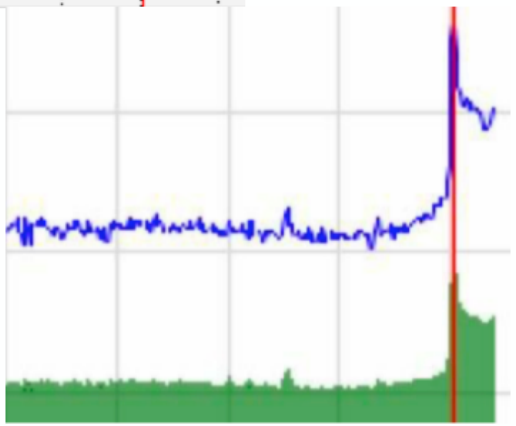
离线+实时 写入方案



万亿数据同步方案

HPA 弹性伸缩





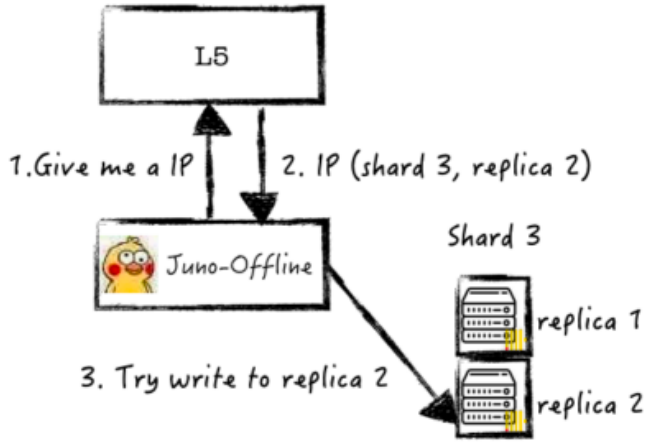
QQ音乐直播：陈奕迅《Live is so much better with music 》慈善音乐会

07–11 17:00

TME live:五月天 突然好想见到你 mayday 2020 live in the sky 线上演唱会

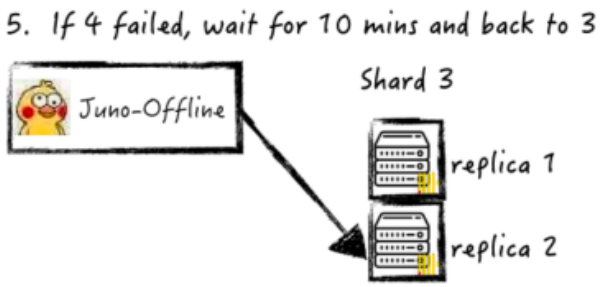
05–31 20:00

spark 离线写



Each block of data is written atomically. The INSERT query is divided into blocks up to `max_insert_block_size = 1048576` rows. In other words, if the `INSERT` query has less than 1048576 rows, it is made atomically.

Data blocks are deduplicated. For multiple writes of the same data block (data blocks of the same size containing the same rows in the same order), the block is only written once. The reason for this is in case of network failures when the client application doesn't know if the data was written to the DB, so the `INSERT` query can simply be repeated. It doesn't matter which replica INSERTs were sent to with identical data. `INSERTs` are idempotent. Deduplication parameters are controlled by merge_tree server settings.



写性能瓶颈

1、分区数过多

> **Info**
>
> A merge only works for data parts that have the same value for the partitioning expression. This means **you shouldn't make overly granular partitions** (more than about a thousand partitions). Otherwise, the `SELECT` query performs poorly because of an unreasonably large number of files in the file system and open file descriptors.

小时分区 => 天分区

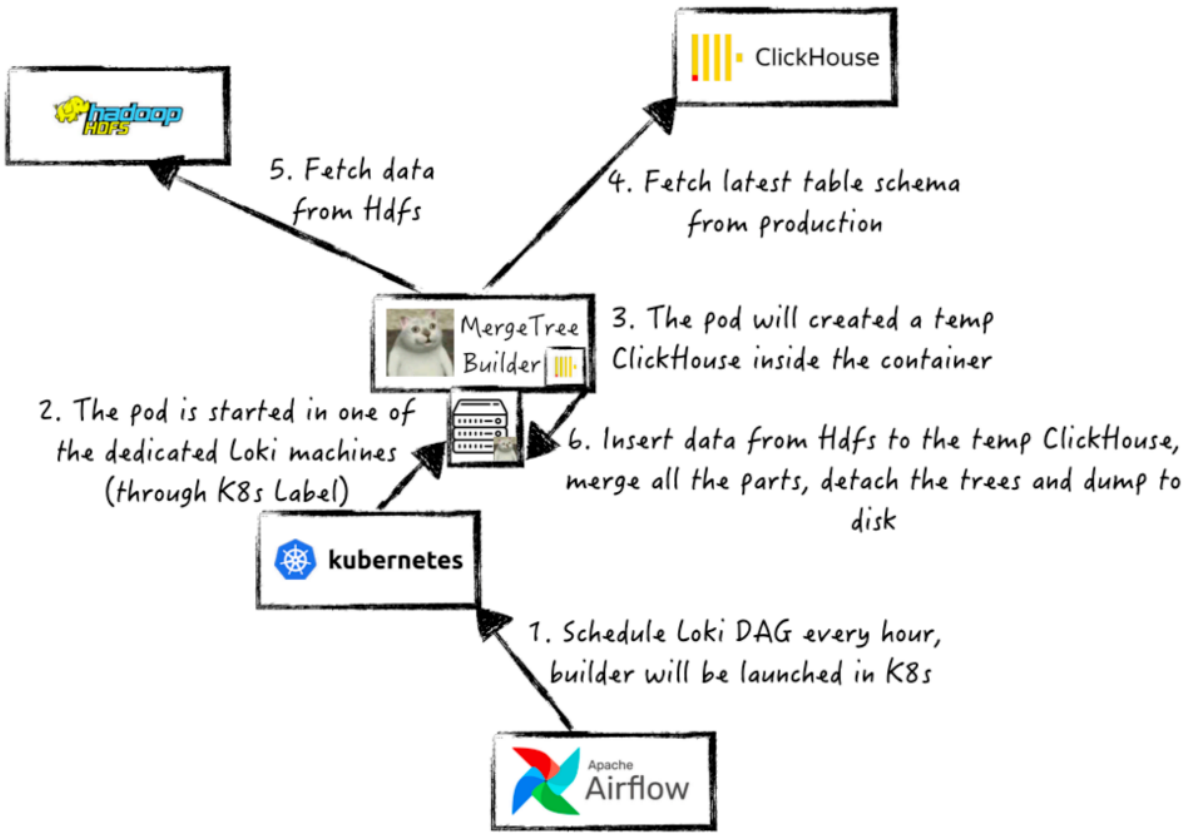
2、zk同步瓶颈

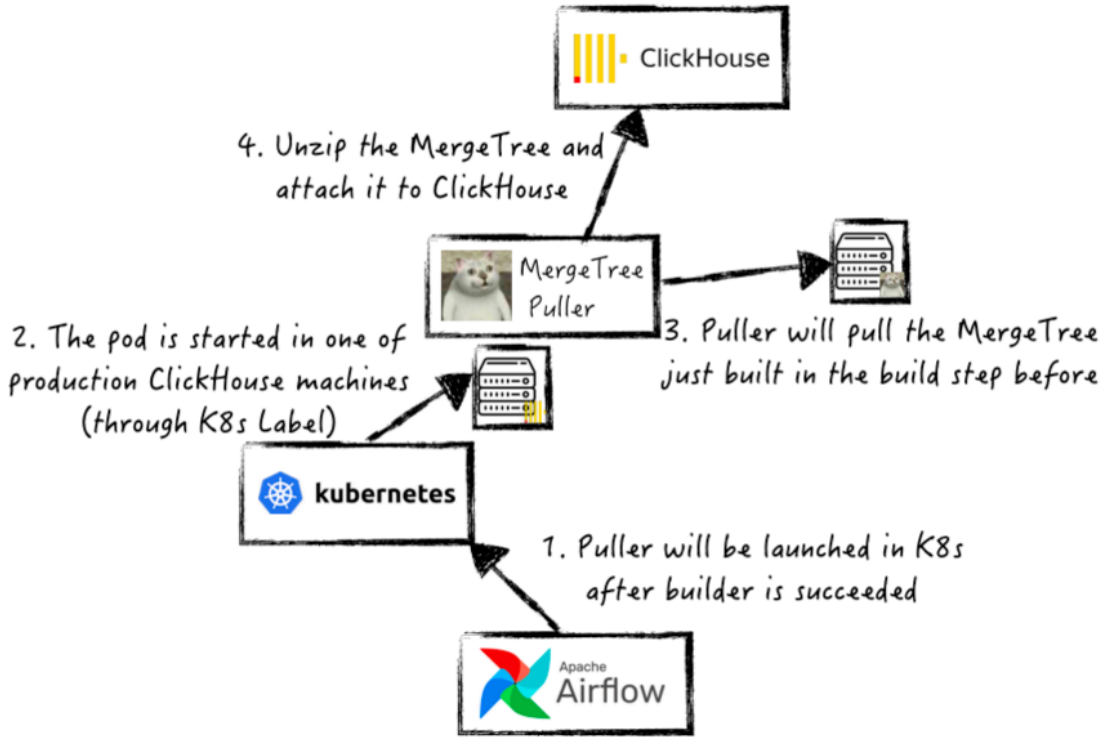
SSD 提升IO读写性能，JVM资源调优
拆分集群
多zk observer 节点
不同表指向不同zk集群，auxiliary_zookeepers
自定义同步环

3、大表批量，网卡打满，Merge Parts太慢

读写分离方案

读写分离

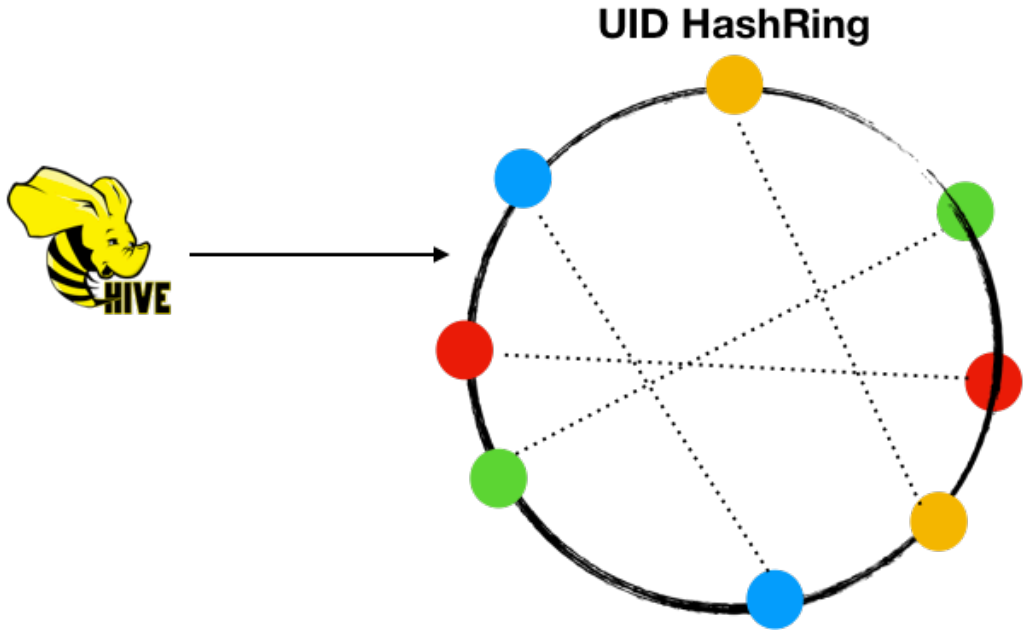

ClickHouse
hadoop HDFS
5. Fetch data from Hdfs
4. Fetch latest table schema from production
MergeTree Builder
3. The pod will created a temp ClickHouse inside the container
2. The pod is started in one of the dedicated Loki machines (through K8s Label)
6. Insert data from Hdfs to the temp ClickHouse, merge all the parts, detach the trees and dump to disk
kubernetes
1. Schedule Loki DAG every hour, builder will be launched in K8s
Apache Airflow


ClickHouse
4. Unzip the MergeTree and attach it to ClickHouse
MergeTree Puller
3. Puller will pull the MergeTree just built in the build step before
2. The pod is started in one of production ClickHouse machines (through K8s Label)
kubernetes
1. Puller will be launched in K8s after builder is succeeded
Apache Airflow

进一步缩短数据导入链路

老链路

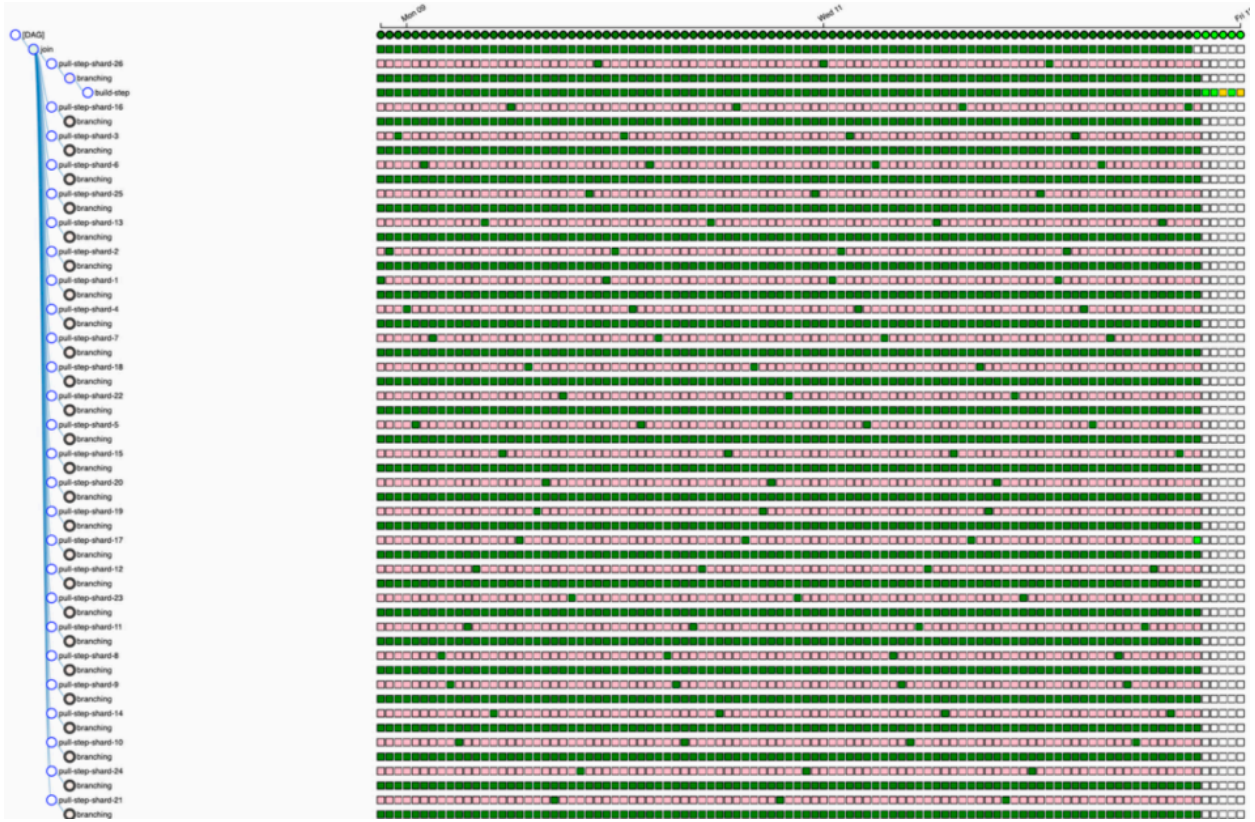
新链路

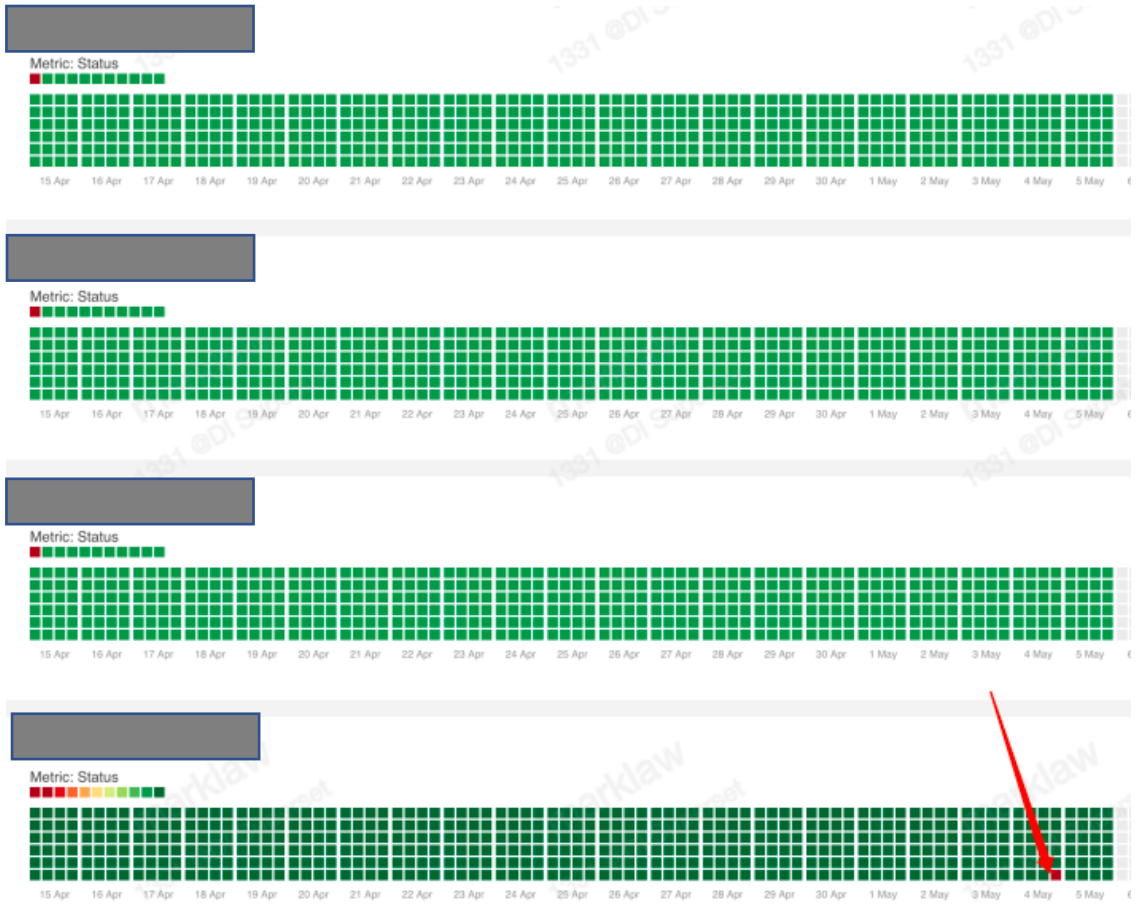


1. 一致性Hash算法，按用户uid分成1024桶
2. 同一用户行为数据落在同一个节点，提升用户历史行为聚合性能
3. 一致性hash架构，加速集群扩容和数据迁移
4. 写入端完成数据双写，不依赖zookeeper

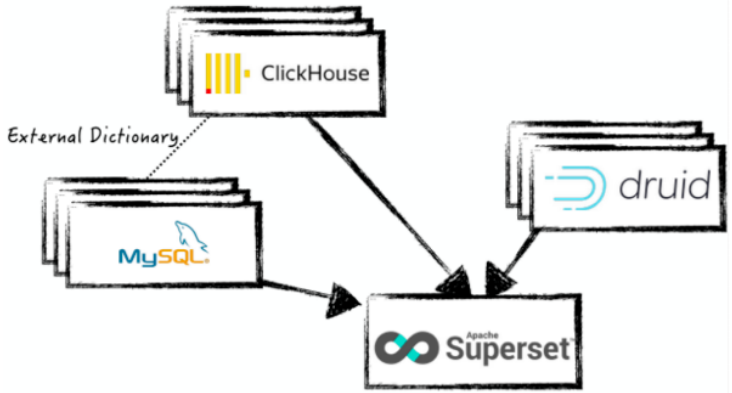
https://github.com/ClickHouse/ClickHouse/issues/16798

同步状态监控





敏捷自助数据分析


ClickHouse
External Dictionary
druid
MySQL
Apache Superset

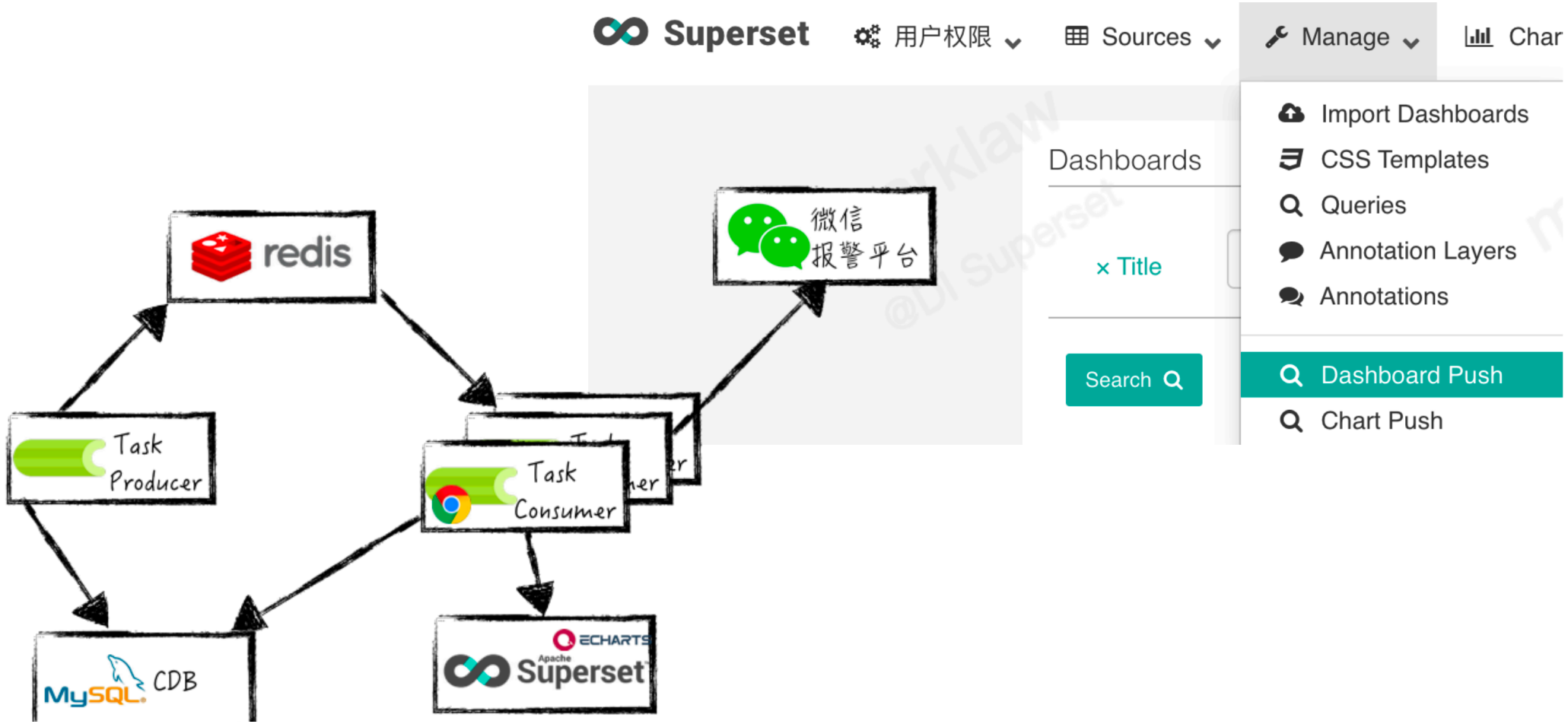

215
+7.0% WoW
80.7M
Line Chart
Big Number with Trendline
Table
Filter Box
Bar Chart
Area Chart
Time-series Bar Chart
deck.gl Polygon
Pie Chart
Time-series Table
Pivot Table
Histogram
Big Number
deck.gl Scatterplot
deck.gl 3D Hexagon
Time-series Period Pivot
deck.gl Arc
Heatmap
deck.gl Grid
Dual Line Chart
deck.gl Screen Grid
Multiple Line Charts
Treemap
Box Plot


QQMusicBU大数据团队
3/10/20 12:00 AM
2020-03-09 全民K歌
3/11/20 12:00 AM
2020-03-10 全民K歌
3/12/20 12:00 AM
2020-03-11 全民K歌


Q音BU数据助手
2020-03-15 QQ 音樂全平台規模日報

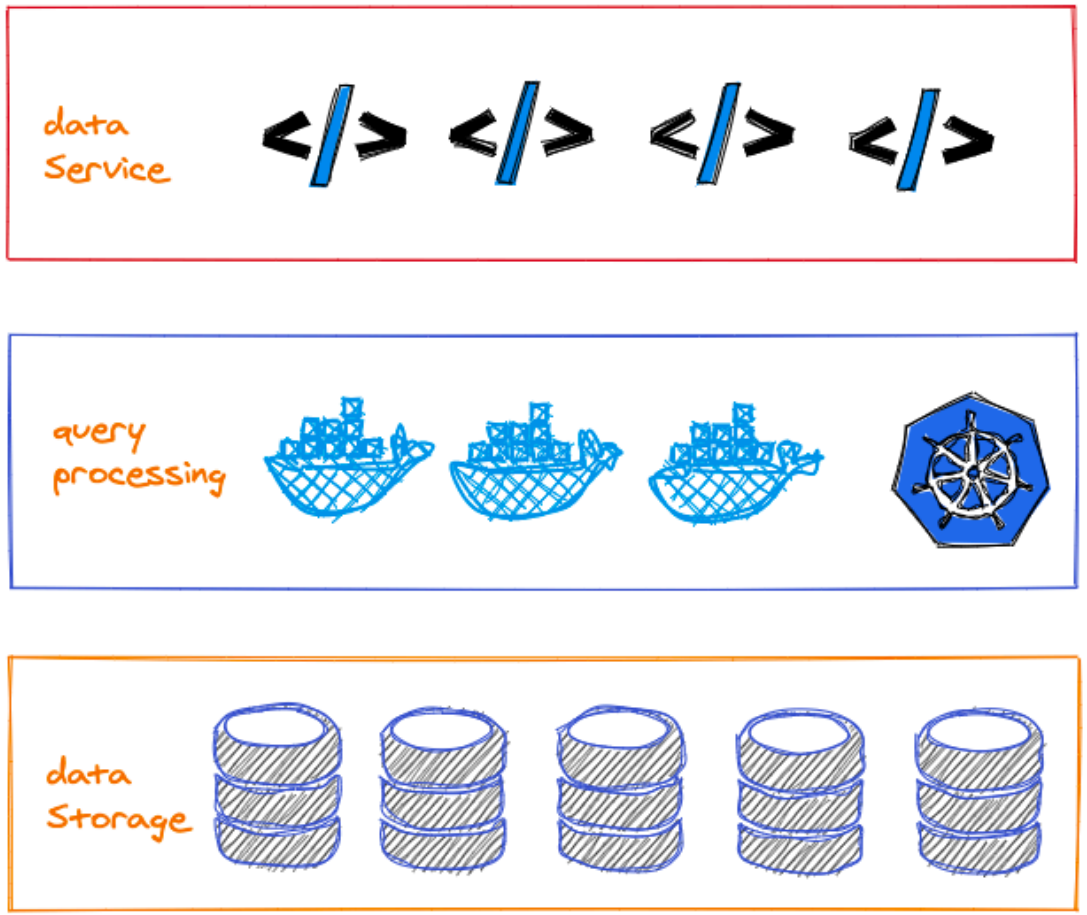

自己动手！
丰衣足食

数据订阅推送


Superset
用户权限
Sources
Manage
Char
Import Dashboards
CSS Templates
Queries
Annotation Layers
Annotations
Dashboard Push
Chart Push
Dashboards
× Title
Search
redis
微信
报警平台
Task
Producer
Task
Consumer
MySQL
CDB
ECHARTS
Apache
Superset

03 平台思考

云原生存储计算分流



Local Machine 96c256g 30T

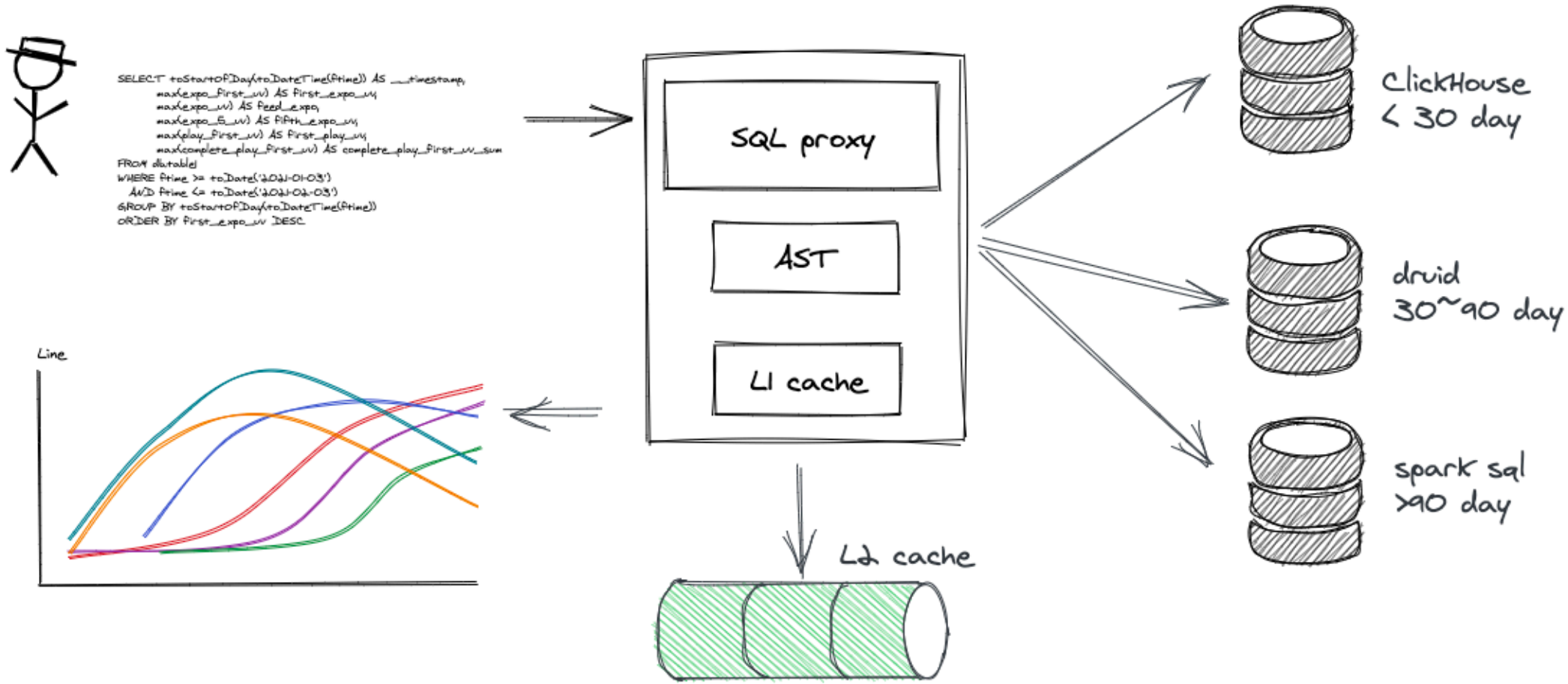
```
CREATE TABLE lineorderflat_multithread_insert_bx2
ENGINE = MergeTree
PARTITION BY toYear(LOORDERDATE)
ORDER BY (LOORDERDATE, LOORDERKEY) AS
SELECT *
FROM lineorderflat
SETTINGS max_insert_threads = 60

Ok.

0 rows in set. Elapsed: 1605.129 sec. Processed 600.04 million rows, 140.52 GB (373.83 thousand rows/s., 87.55 MB/s.
)
```

Cloud CFS 3 shard 1 replication

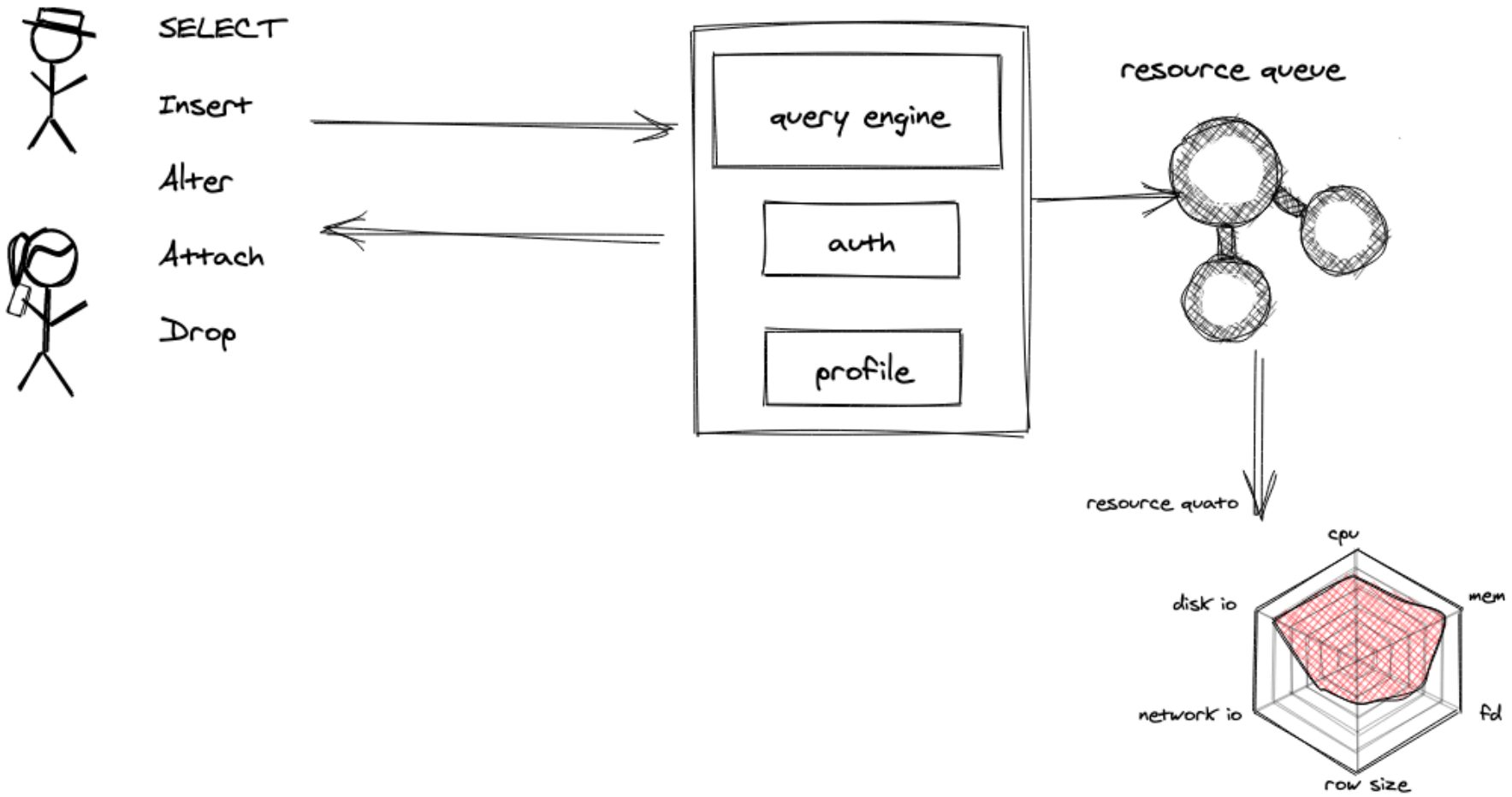
```
0 rows in set. Elapsed: 70.678 sec. Processed 600.04 million rows, 140.41 GB (
8.49 million rows/s., 1.99 GB/s.)
```

```
0 rows in set. Elapsed: 71.301 sec. Processed 600.04 million rows, 140.41 GB (
8.42 million rows/s., 1.97 GB/s.)
```

Cache加速


SELECT toStartOfDay(toDateTime(ftime)) AS __timestamp,
max(expo_first_uv) AS first_expo_uv,
max(expo_uv) AS feed_expo,
max(expo_5_uv) AS fifth_expo_uv,
max(play_first_uv) AS first_play_uv,
max(complete_play_first_uv) AS complete_play_first_uv_sum
FROM dlutable1
WHERE ftime >= toDate('2021-01-03')
AND ftime <= toDate('2021-02-03')
GROUP BY toStartOfDay(toDateTime(ftime))
ORDER BY first_expo_uv DESC
SQL proxy
AST
L1 cache
L2 cache
Line
ClickHouse
< 30 day
druid
30~90 day
spark sql
>90 day

多用户资源隔离


SELECT
Insert
Alter
Attach
Drop
query engine
auth
profile
resource queue
resource quato
cpu
mem
fd
row size
network io
disk io

平台演进方法论

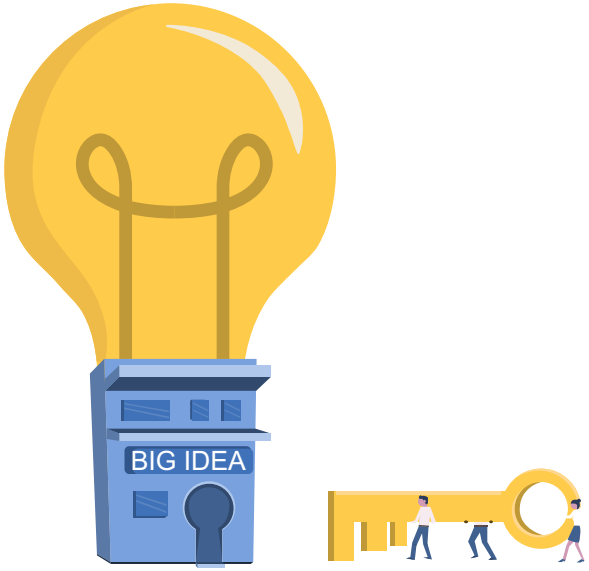
平台建设

数据时效 + 数据质量

组织保障

高效组织 + 有效方法

= 价值倍增



- 平台的演进贴合业务并赋能业务
- 平台的演进兼顾多方协同共享共建
- 平台的演进促进组织架构优化

More Heros

分享到微信

欢迎自荐/推荐各类大数据人才

创 造 音 乐 无 限 可 能

CREATING ENDLESS
OPPORTUNITIES WITH MUSIC

腾讯音乐娱乐集团
TENCENT MUSIC ENTERTAINMENT