DAMS

中国数据智能管理峰会

DATA & AI MANAGEMENT SUMMIT

爱奇艺大数据分析平台的演进

演讲人：邹兴标

目录

爱奇艺数据现状

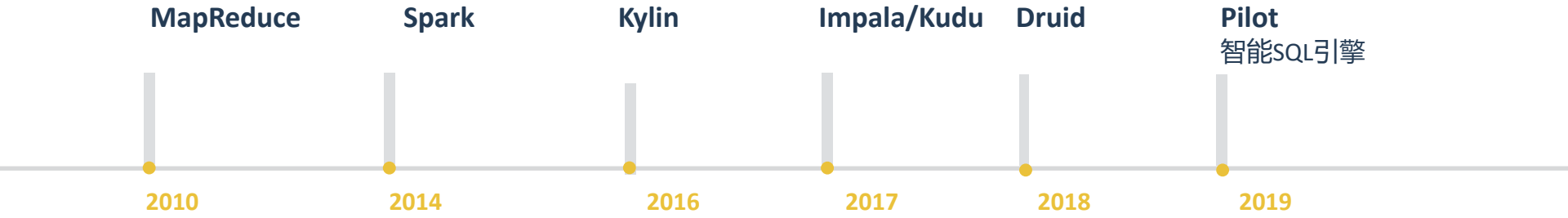
日均上亿独立设备数

离线：日增 500TB

实时：日处理万亿级消息

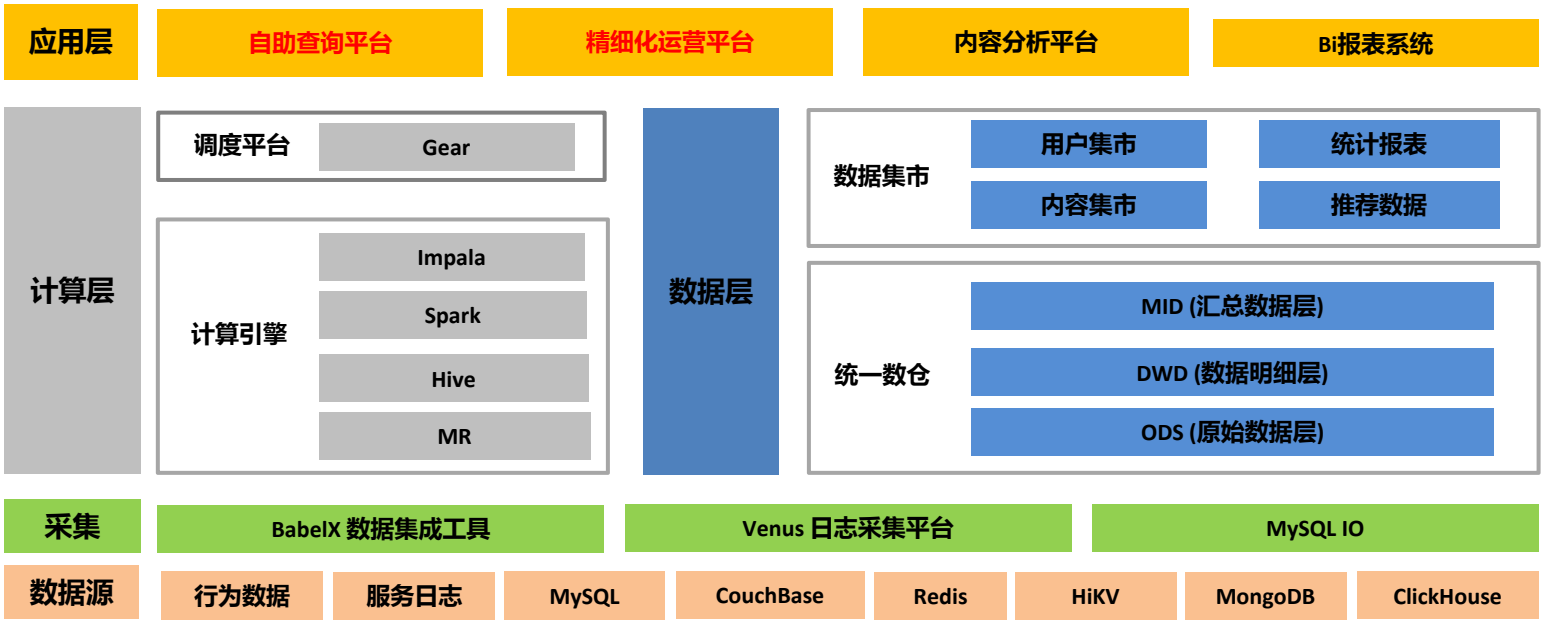
10+ Apps

爱奇艺数据平台发展史


MapReduce
Spark
Kylin
Impala/Kudu
Druid
Pilot
智能SQL引擎
2010
2014
2016
2017
2018
2019

爱奇艺数据平台架构

数据平台目标：链接数据与业务，提供高效，快捷的分析平台



目录

自助查询平台-诞生背景

遇到的问题

- 运营及老板有越来越多的数据需求
- 固定的报表开发难以满足各方数据需求
- 数据开发工程师逐渐成为获取数据的瓶颈

破局的思路

- 需要赋能运营及分析师自助获取数据的能力

自助查询平台1.0

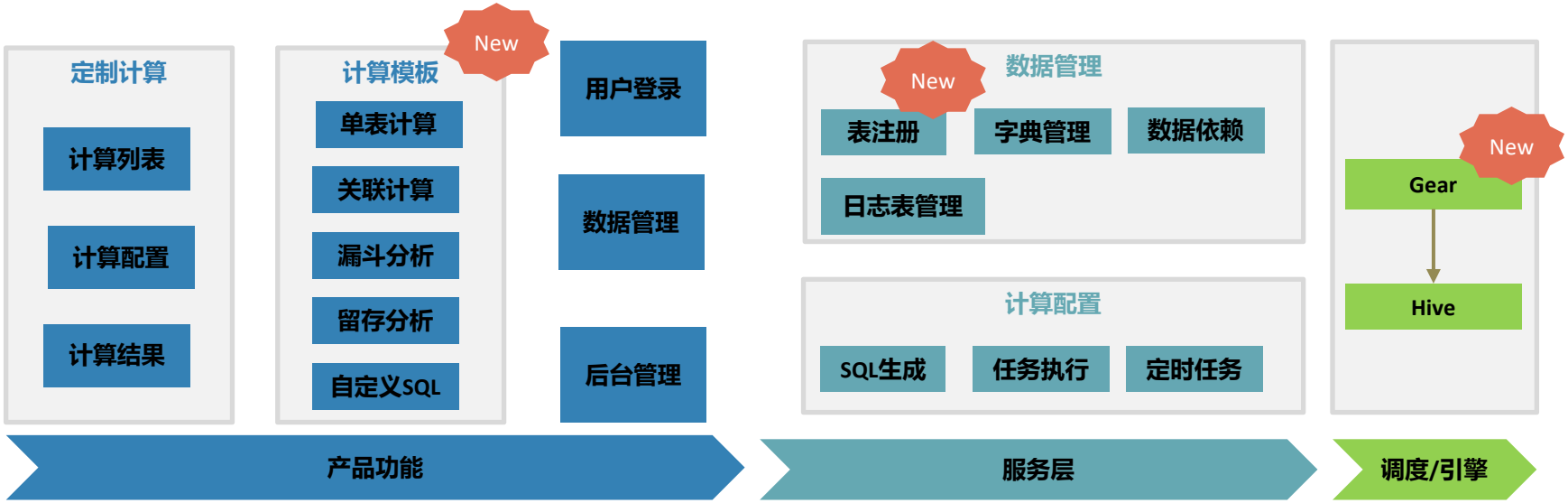

定制计算
计算列表
计算配置
计算结果
用户登录
数据管理
后台管理
数据管理
表管理
字典管理
数据依赖
计算配置
SQL生成
任务执行
定时任务
HIVE
产品功能
服务层
调度/引擎

自助查询平台1.0

缺陷

- 功能单一：支持单表计算，支持计算模板类型少
- 数据源单一：数据来源为原始日志，没有数仓数据
- 引擎单薄：使用hive client作为单一执行引擎，稳定性不强

自助查询平台2.0



改进

- 丰富功能：支持了关联，留存，漏斗等计算类型
- 扩展数据源：支持用户自定义注册数仓表
- 健壮调度：集成基于Apache Oozie的Gear工作流调度系统，去除单点入口机依赖

效果

- 任务失败率从5%降低至2%（非语法类）

自助查询平台2.0-表一键替换

问题

- 数据开发提供了注册的数仓表，但是由于用户的习惯，不愿意进行计算的切换，导致高效的表难以推广

解决方案

- 对最核心的用户行为数据进行数仓建模，进行常用维度聚合，平台提供一键切换功能，杜绝使用原始表

效果

- 释放了**50台**服务器资源
- 单任务平均时长**40分钟→20分钟**

自助查询平2.0

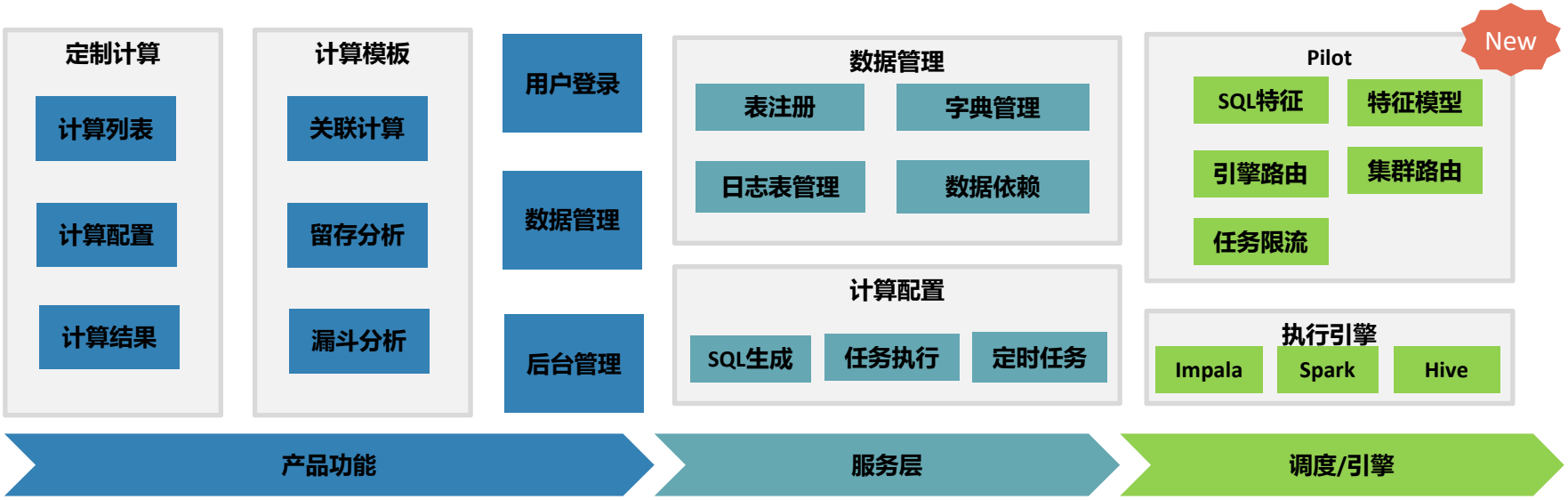
缺陷

- 执行效率：使用Hive作为计算引擎，在日益增长的数据现状下，难以满足数据即时提取需求

新的问题

- 数据量的增加及基础设施的限制，公司开始多机房建设，数据可能分布到多个大数据集群。面临着为每一次查询找到正确的集群的问题。

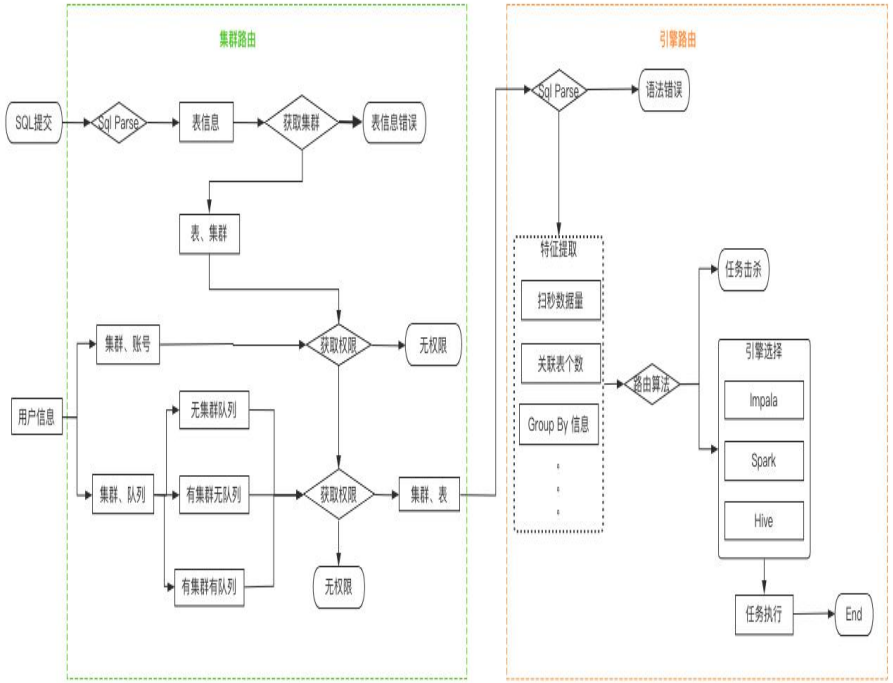
自助查询平台-魔镜3.0



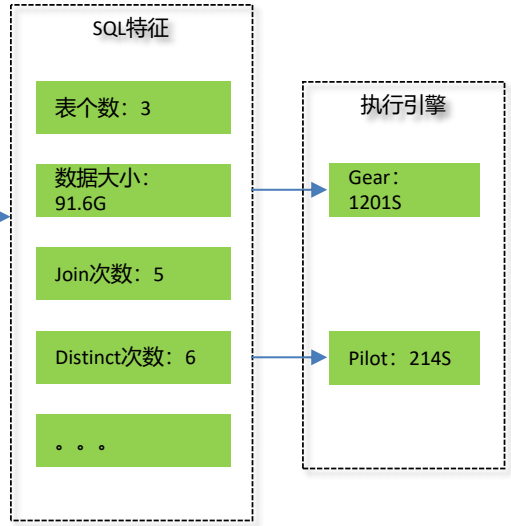
改进及效果

- 自研智能SQL路由引擎Pilot，实现基于机器学习的智能SQL路由、失败降级、审计、错误诊断等功能
- 查询性能从平均**20分钟→16分钟**，提升25%
- 查询失败率降低至从**2%降低至0.3%**

Pilot-查询路由




```
SELECT date_add(AB.dt, 30) AS dt ,
    count(DISTINCT AB.qyid) AS quanyuan_uv,
    count( DISTINCT coalesce(C.device_id, AB.qyid, NULL) ) AS push_chuda_uv,
    count( DISTINCT if(C.device_id IS NOT NULL, AB.qyid, NULL) ) AS push_click_uv,
    count( DISTINCT if( C.device_id IS NOT NULL
                        AND E.device_id IS NOT NULL, AB.qyid, NULL ) ) AS chuda_liucun_uv FROM
 ( SELECT DISTINCT a.dt,
  if( B.u IS NOT NULL AND C.u IS NULL, CASE WHEN A.platform_id = '2_22_254' THEN A.qyid ELSE A.u END, NULL ) qyid
 FROM
  ( SELECT dt, platform_id, device_id AS u, qyid
   FROM yd_dwd.yd_dwd_dd_fact_hive_new_user
   WHERE dt = date_sub('2020-08-17', 30)
    AND platform_id IN ('2_22_254', '2_22_253') ) A
 LEFT JOIN
  ( SELECT DISTINCT dt, device_id AS u
   FROM yd_dwd.yd_dwd_dd_fact_view_actv
   WHERE dt >date_sub('2020-08-17', 28)
    AND platform_id IN ('2_22_254', '2_22_253') ) B ON A.u = B.u
 LEFT JOIN
  ( SELECT DISTINCT dt, device_id AS u
   FROM yd_dwd.yd_dwd_dd_fact_view_actv
   WHERE dt >= date_sub('2020-08-17', 28)
    AND dt <= '2020-08-17'
    AND platform_id IN ('2_22_254', '2_22_253') ) C ON A.u = c.u ) AB
LEFT JOIN
 ( SELECT *
  FROM mbd_backend_feige.other_apps_feige_message
  WHERE app_id = '111'
   AND deliver_time IS NOT NULL
   AND dt = '2020-08-17' ) C ON AB.qyid = C.device_id
LEFT JOIN
 ( SELECT *
  FROM mbd_backend_feige.other_apps_feige_message
  WHERE app_id = '111'
   AND click_time IS NOT NULL
   AND dt = '2020-08-17' ) D ON AB.qyid = d.device_id
LEFT JOIN
 ( SELECT *
  FROM yd_dwd.yd_dwd_dd_fact_view_actv
  WHERE dt = '2020-08-17' ) E ON AB.qyid = e.qyid
GROUP BY date_add(AB.dt, 30)
```



性能提升：80%

目录

用户分析平台-诞生背景

Q1

看过隐没的角落的人群数据分析、提取	2周
策略上线	2周
效果计算 申请分析师资源进行时间片对比分析	1周

深入实践

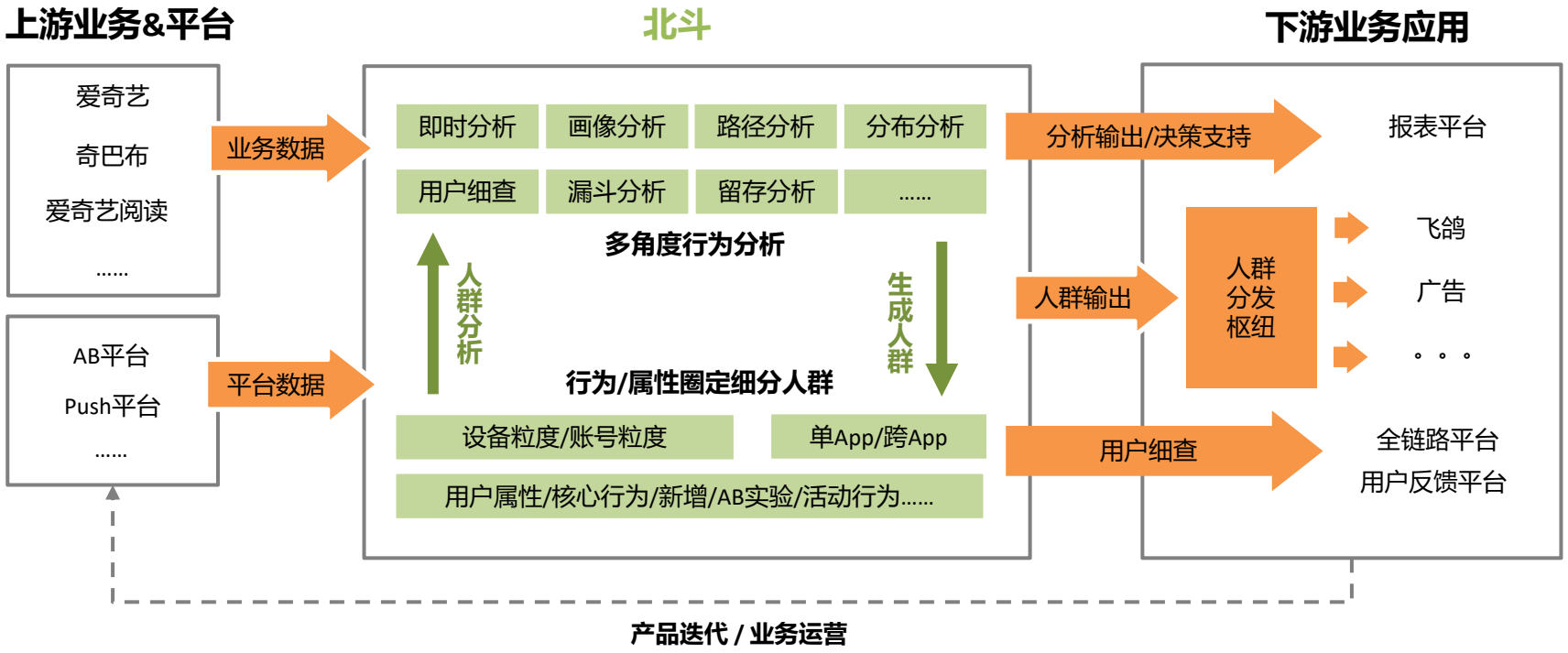
死循环局面：

- 策略细分靠临时开发
- 跨APP、甚至APP内跨场景数据不统一，无法规模化复用

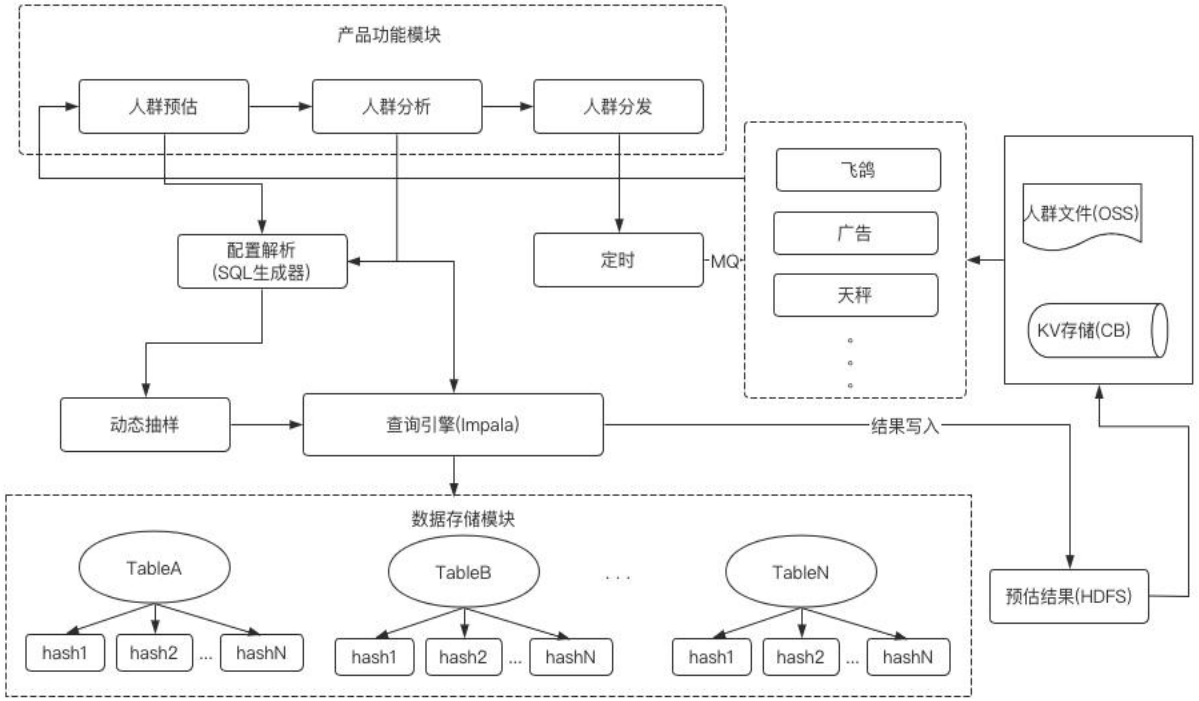
底层、工具、应用层同时升级

- 人群圈选→行为分析→定向运营
- 数据可快速 可分析→可决策→可行动
- 不但提供表的服务，进一步提供解决方案服务

用户分析平台-产品架构


上游业务&平台
北斗
下游业务应用
爱奇艺
奇巴布
爱奇艺阅读
.....
AB平台
Push平台
.....
业务数据
平台数据
即时分析
画像分析
路径分析
分布分析
用户细查
漏斗分析
留存分析
.....
多角度行为分析
人群分析
生成人群
行为/属性圈定细分人群
设备粒度/账号粒度
单App/跨App
用户属性/核心行为/新增/AB实验/活动行为.....
分析输出/决策支持
人群输出
用户细查
报表平台
人群分发枢纽
飞鸽
广告
。。。
全链路平台
用户反馈平台
产品迭代 / 业务运营

用户分析平台-技术架构



数据层:

将数据通过统一的Hash算法切片为100份存储，基于用户行为库定制的用户事件，属性模型。

引擎层:

基于动态抽样（根据样本数量决定Hash分片数）+ Impala查询引擎，提升单次分析效率从平均**70S提升至9S以内**

产品模块:

人群结果通过文件及接口方式分发至公司90%的触达用户平台

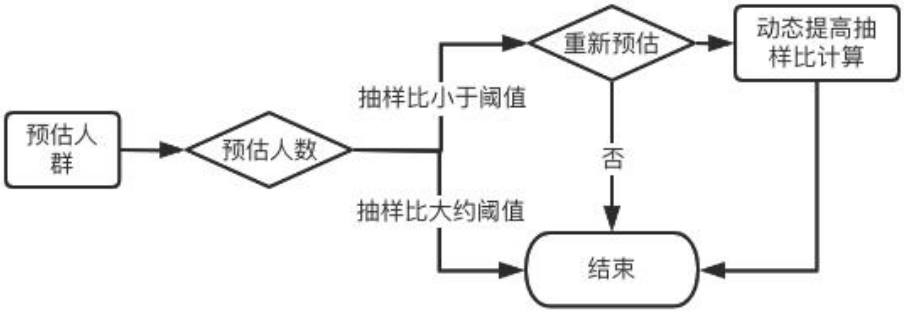
用户分析平台-技术选型

VS

VS

	Kylin	Spark	Impala
单表查询(10 亿)	好(命中Cube)	差	中
关联查询	差	中	好
并发控制	中	好	差

用户分析平台-抽样实现



背景

需要从900亿数据中圈选出入左图条件人群

解决方案

对数据进行抽样，保证千分之五以下的误差

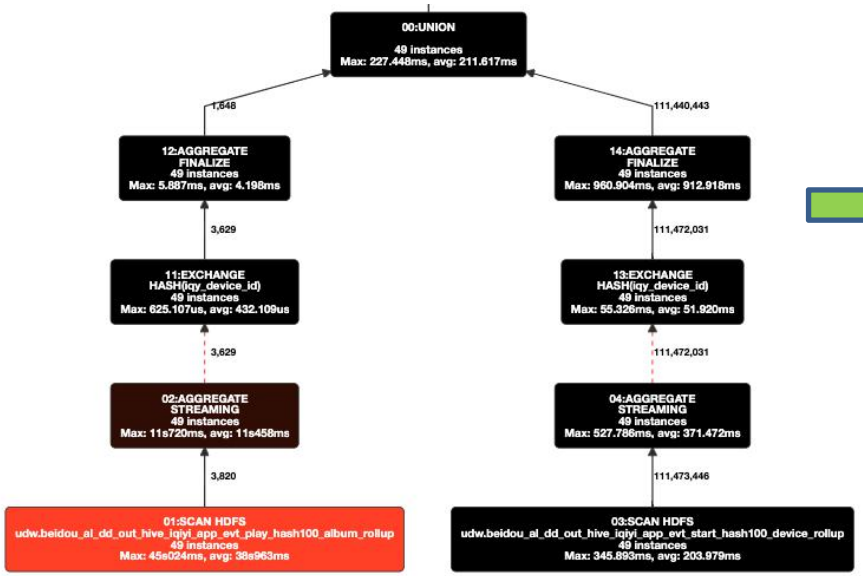
性能提升

性能从70S→9S

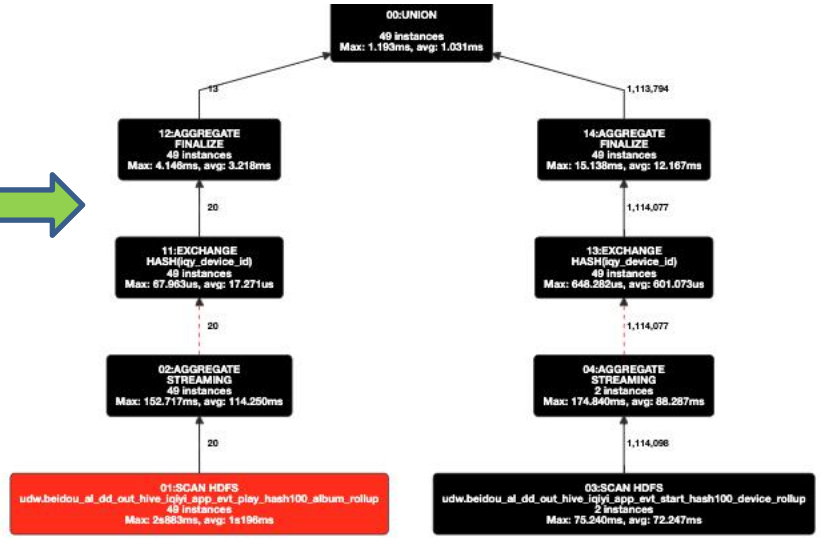
全量和抽样的误差千分之一

用户分析平台-抽样实现

全量分析



抽样分析



用户分析平台-结构数据使用

背景

每个用户基本都有多个剧集偏好，需要计算用户对于剧集每个用户基本都有多个剧集偏好

数据存储：

每个用户存多条记录

存储成本高，scan hdfs耗时

用户ID	偏好
AAAA	言情
AAAA	家庭
AAAA	喜剧
BBBB	惊悚
BBBB	喜剧

优化提升

修改数据存储类型，一个用户存一条特征

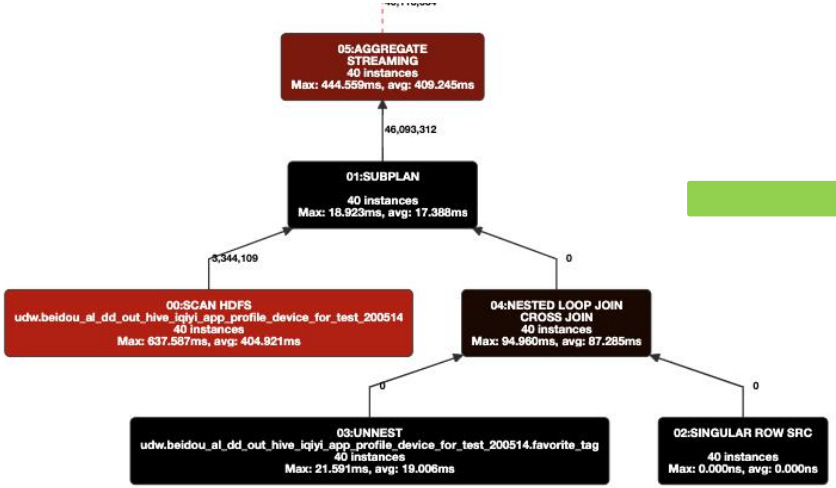
```
array<struct<
  entity_id:string,
  entity_name:string,
  entity_uv:bigint,
  base_uv:bigint,
  entity_uv_rate:doubl
  entity_weight:double
  entity_uv_high:bigin
  entity_uv_mid:bigint
  base_uv_high:bigint,
  base_uv_mid:bigint,
  c1:string,
  c1_name:string
```

性能提升

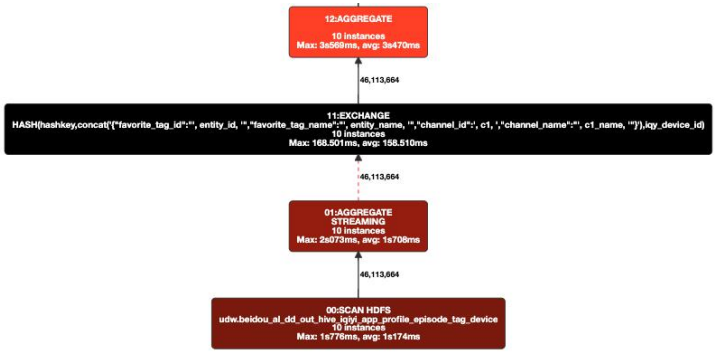
性能从**8.66S→2.89S**

用户分析平台-结构数据使用

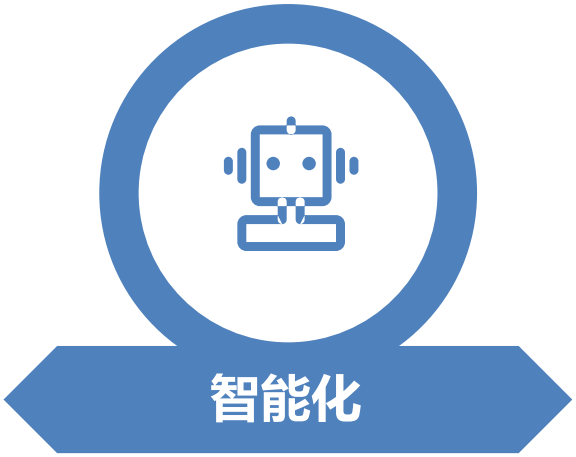
单条记录



结构体方式



未来规划



DAMS
中国数据智能管理峰会
DATA & AI MANAGEMENT SUMMIT

Q&A

DAMS

中国数据智能管理峰会

DATA & AI MANAGEMENT SUMMIT

THANK YOU !