

Apache Doris × 阿里云联合 Meetup

🕒 10月26日 (周六) 13:30-17:15



玩心不止基于 Apache Doris 构建游戏业务实时分析平台实践

提升数据驱动决策效率的探索与实现

刘志邓

2024.10

公司介绍

>>

游戏独代

拥有多家游戏独立代理权，提供专业的代理服务



>>

游戏联运

拥有专业的游戏运营团队



公司业务

CORPORATE
BUSINESS

公司介绍

玩心不止手游是一款手游福利平台产品，提供数款热门游戏供玩家下载，涵盖仙侠、策略、卡牌、三国、传奇、挂机等多种类型的热门手游，此外还有满福利转游、火热群聊社区、大神整合攻略等平台服务支持功能。

精品游戏, 应有尽有
传奇、仙侠、魔幻, 你想玩的我们都有

资讯攻略, 一站俱全
游戏活动、大神风采一手掌握

超值福利, 一键领取
礼包、充值、代金券领到手软

游戏

论坛

福利

目录

- 01 业务需求分析
- 02 重点使用场景介绍
- 03 选型思考与优化案例
- 04 核心收益与未来展望



01 业务需求分析

游戏业务场景与数据分析目标

游戏运营和市场监控

实时监控游戏运营状态与市场关键指标



推广效果评估

- 分析广告效果，关于不同渠道不同计划的 LTV, ROI, ARPU, ARPPU 等关键指标，提升回本率



营收数据监控

- 实时查看收入，调整运营策略
- 各个场景下营收效果，合理分配运营重心



活跃用户统计

- 追踪活跃玩家，保持用户粘性



游戏故障排查

- 快速响应故障，保障游戏稳定
- 游戏内部道具，礼包非正常领取



用户行为分析

- 监控玩家行为，优化游戏体验



平台私域分析

- 洞察玩家社交，增强游戏社区
- 开展各式各样活动刺激用户参与，提高玩家黏性



业务背景

业务需求



面向分析场景的 OLAP

对复杂的筛选维度和条件查询，
可以实现实时或近实时的响应。

01



满足多场景的数据存储 和查询

可以高吞吐的数据稳定写入，满足公司上亿实时市场数据在分钟级别写入，支持高并发的点查，保证不同使用场景的查询效率。

02



运维简单

部署和运维成本低，高可用，
系统稳定性高，节点易拓展

03



支持联邦查询

可以跨不同数据中心进行读取
查询，有灵活的数据处理能力。

04

02 选型思考与优化案例

技术选型、实施与持续优化

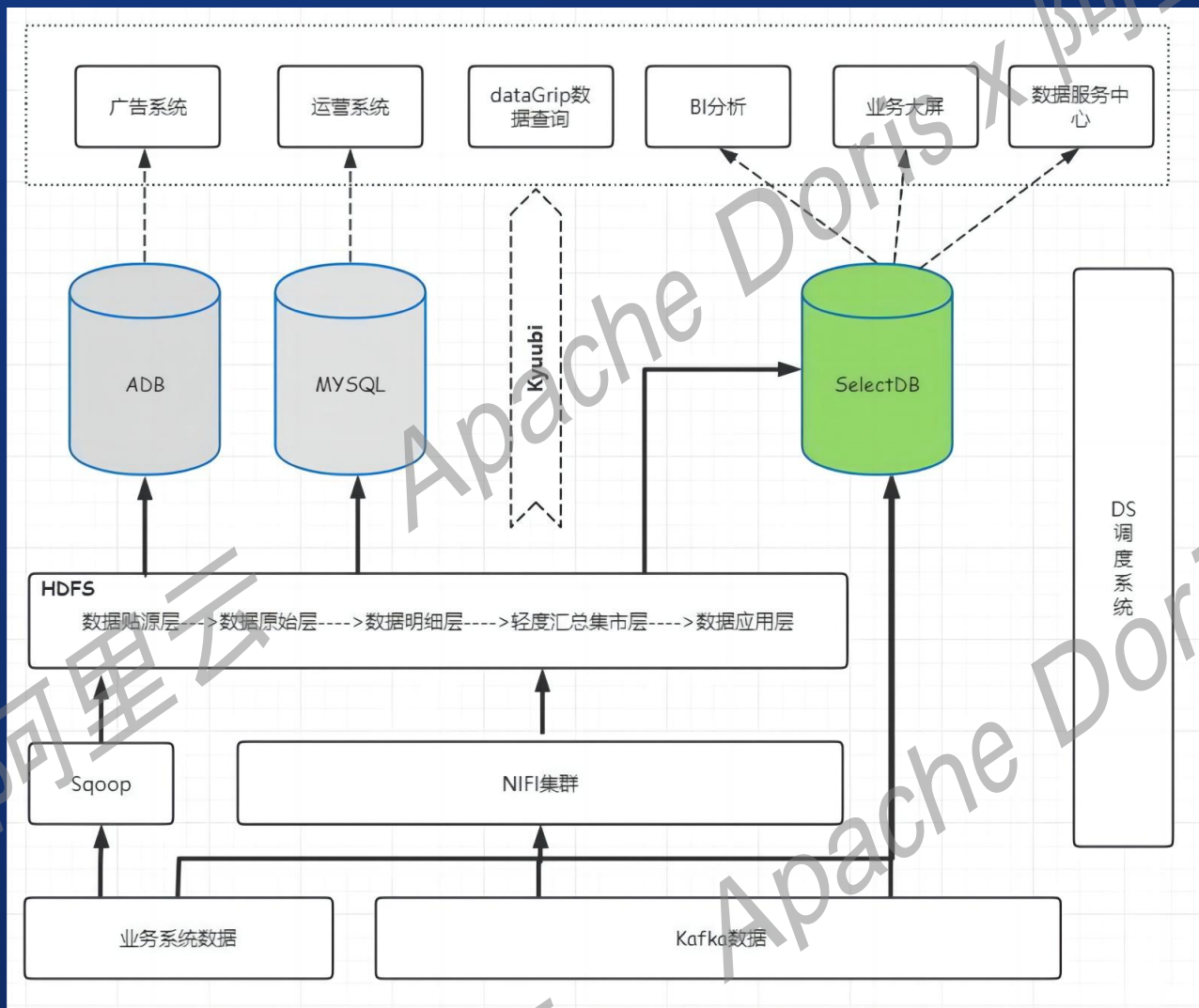
OLAP选型分析

选型因素

关键指标	Apache Doris (SelectDB)
运维部署	Doris 不依赖外部系统，不存在小文件问题，节点可线性扩展，大大降低了运维复杂度。
使用方式	兼容 MySQL 协议，接入、迁移以及使用成本低。
分析性能	Doris 支持向量化执行引擎和物化视图等，可有效提升查询效率。
社区活跃	Doris 的社区非常活跃，Issue 问题能够得到及时反馈。
湖仓一体	有非常好的 MPP 向量话分布式查询层，可以对湖上数据加速分析。
生态工具	支持多种 Connector，满足离线和实时等多场景的数据传输。

OLAP选型分析

数据流程



离线

• 通过不同采集方式存入 HDFS，通过离线数仓层层模型搭建形成可用数据集，同步到 SelectDB 供下游使用。

实时

• 数据存入 Kafka 后，基于 StreamPark 开发 flink-sql 作业，将数据实时写入 SelectDB 供下游使用。

OLAP 选型分析

数据导入

DataX 离线 数据集成

- 离线数仓数据模型同步
- 主要应用 BI 分析看板场景
- 客群人群包预估和圈选
- 活动效果多维分析

采用 Stream Load

Routine Load

- 消费 Kafka Topic 中的数据
- 支持 Exactly-Once 语义，保证数据不丢不重

采用 Routine Load

实时写入

- Flink CDC 目前支持接入实现自动删除
- flink-SQL、DataStream 的方式
- Lookup Join 的使用等

采用 Stream Load，同时开启部分列更新
partial_columns=true

使用场景

Doris 使用场景

2023 年 Q1 开始引入 Doris



运营业务分析



- 多维数据立方体
- 分钟级别业务分析图表
- 用户行为查询
- 研发数据质量跟踪

实时大屏



- 实时业务趋势变化
- 平台业务KPI监控
- 市场买量效果追踪

数据中心



- 用户运营标签
- 营销活动效果
- dataApi

使用场景

从 Doris 到 SelectDB 的思考

2023年Q1开始引入Doris



2024年Q2开始引入SelectDB



- SelectDB 是 100% 兼容 Doris 的数仓产品
- SelectDB 单副本写&读写 Cache 加速性能，查询性能和写入性能大大提升
- 部署成本大幅度降低，自建 Doris 服务器和服务都需要人力维护，使用初期迭代版本升级频繁，运维成本增高
- 技术服务支持更高效，遇到异常情况时可以快速得到响应

性能优化实践

查询优化

```
select
  case when `first_day_pay_money` = 0 then '0'
  when `first_day_pay_money` > 0 and `first_day_pay_money` <= 10 then '0-10'
  when `first_day_pay_money` > 10 and `first_day_pay_money` <= 50 then '10-50'
  when `first_day_pay_money` > 50 and `first_day_pay_money` <= 100 then '50-100'
  when `first_day_pay_money` > 100 and `first_day_pay_money` <= 500 then '100-500'
  when `first_day_pay_money` > 500 and `first_day_pay_money` <= 1000 then '500-1000'
  when `first_day_pay_money` > 1000 and `first_day_pay_money` <= 2000 then '1000-2000'
  when `first_day_pay_money` > 2000 then '2000+'
end as `付费分段_0`,
sum(`is_register`) as `是否注册_sum_0`,
sum(case when (`date_diff` + 1) = 2 then `is_ret` end) / sum(`is_register`) as `次日留存_nul_1`,
sum(case when (`date_diff` + 1) = 3 then `is_ret` end) / sum(`is_register`) as `3日留存_nul_2`,
sum(case when (`date_diff` + 1) = 4 then `is_ret` end) / sum(`is_register`) as `4日留存_nul_3`,
sum(case when (`date_diff` + 1) = 5 then `is_ret` end) / sum(`is_register`) as `5日留存_nul_4`,
sum(case when (`date_diff` + 1) = 6 then `is_ret` end) / sum(`is_register`) as `6日留存_nul_5`,
sum(case when (`date_diff` + 1) = 7 then `is_ret` end) / sum(`is_register`) as `7日留存_nul_6`,
sum(case when (`date_diff` + 1) = 8 then `is_ret` end) / sum(`is_register`) as `8日留存_nul_7`,
case max(case when `first_day_pay_money` = 0 then '0'
  when `first_day_pay_money` > 0 and `first_day_pay_money` <= 10 then '0-10'
  when `first_day_pay_money` > 10 and `first_day_pay_money` <= 50 then '10-50'
  when `first_day_pay_money` > 50 and `first_day_pay_money` <= 100 then '50-100'
  when `first_day_pay_money` > 100 and `first_day_pay_money` <= 500 then '100-500'
  when `first_day_pay_money` > 500 and `first_day_pay_money` <= 1000 then '500-1000'
  when `first_day_pay_money` > 1000 and `first_day_pay_money` <= 2000 then '1000-2000'
  when `first_day_pay_money` > 2000 then '2000+'
end) when '0' then 0 when '0-10' then 1 when '10-50' then 2 when '50-100' then 3 when '100-500'
then 4 when '500-1000' then 5 when '1000-2000' then 6 when '2000+' then 7 else 8 end as `付费分段
__max__15`, 0 as `__grouping_id`
from `dwm_user_first_pay_grade_ret_data_d14`
where register_date >= '2023-01-01' and register_date < '2023-02-01'
```

背景

- 数据量 2 亿左右
- 查询时间 5s

优化过程

- 分桶从 8 个到 20 个，查询速度提升 2s 左右
- 并行度设置从 4 调整为 8，查询速度提升 1.5s
- 对 group by 字段类型调整，从 String 类型调整为 Decimal，速度提升 1s 左右

优化结果

- 查询时间从 5S 到 600ms

性能优化实践

实时模型按列更新

```
CREATE TABLE `dws_box_user_behavior_stat_dm_di_wanxin` (  
  `user_id` VARCHAR(64) NOT NULL COMMENT '用户id',  
  `date_key` VARCHAR(64) NOT NULL COMMENT '日期',  
  `max_order_time` DATETIME MAX COMMENT '最后一次支付事件',  
  `min_login_time` DATETIME MIN COMMENT '首次登录时间',  
  `is_login_box` VARCHAR(64) REPLACE_IF_NOT_NULL NULL COMMENT '是否登录盒子',  
  `game_maps` JSON REPLACE_IF_NOT_NULL NULL COMMENT '登录游戏app_key集合<app_key:注册时间>',  
  `game_names` TEXT REPLACE_IF_NOT_NULL NULL COMMENT '登录游戏名称集合',  
  `market_game_maps` JSON REPLACE_IF_NOT_NULL NULL COMMENT '登录游戏app_key集合',  
  `market_game_names` TEXT REPLACE_IF_NOT_NULL NULL COMMENT '登录游戏名称集合',  
  `box_game_maps` JSON REPLACE_IF_NOT_NULL NULL COMMENT '登录游戏app_key集合',  
  `box_game_names` TEXT REPLACE_IF_NOT_NULL NULL COMMENT '登录游戏名称集合',  
  `task_id_arr` TEXT REPLACE_IF_NOT_NULL NULL COMMENT '今日完成任务id',  
  `gift_ids` TEXT REPLACE_IF_NOT_NULL NULL COMMENT '今日领取积分礼物id',  
  `remain_credit` BIGINT REPLACE_IF_NOT_NULL NULL COMMENT '剩余积分数',  
  `is_trans_user` BIGINT REPLACE_IF_NOT_NULL NULL COMMENT '今日转游',  
  `trans_games` TEXT REPLACE_IF_NOT_NULL NULL COMMENT '转游游戏名称',  
  `today_is_sign` BIGINT REPLACE_IF_NOT_NULL NULL COMMENT '今日是否签到',  
  `contin_days` BIGINT REPLACE_IF_NOT_NULL NULL COMMENT '截止今日连续签到天数',  
  `trans_gift_ids` TEXT REPLACE_IF_NOT_NULL NULL COMMENT '兑换转游礼包id',  
  `trans_gift_names` TEXT REPLACE_IF_NOT_NULL NULL COMMENT '兑换转游礼包名称',  
  `is_trontable_active` BIGINT REPLACE_IF_NOT_NULL NULL COMMENT '是否参与转盘活动',  
  `is_648_active` BIGINT REPLACE_IF_NOT_NULL NULL COMMENT '是否参与648活动',  
  `is_pay_active` BIGINT REPLACE_IF_NOT_NULL NULL COMMENT '是否参与累充活动',  
  `is_week_week_active` BIGINT REPLACE_IF_NOT_NULL NULL COMMENT '是否参与周周红包活动',  
  `behavior_events` TEXT REPLACE_IF_NOT_NULL NULL COMMENT '盒子行为记录'  
) ENGINE=OLAP  
AGGREGATE KEY(`user_id`, `date_key`)  
COMMENT 'OLAP'  
DISTRIBUTED BY HASH(`user_id`, `date_key`) BUCKETS 5  
PROPERTIES (
```

业务场景

- 用户每日行为特征集市，拓展性要求比较高
- 某个指标更新频率不一致，互相不可影响

使用过程

- 采用AGGREGATE模型，配置聚合类型为REPLACE_IF_NOT_NULL，MIN，MAX

实现效果

- 通过不同用户每日行为集市筛选出特定人群
- 查询用户行为解决部分客诉问题

性能优化实践

OLAP 物化视图

```
create materialized view user_game_day_view as
select
  date_key,
  nvl(game_id,'') as game_id,
  if(nvl(game_id,'') = '', '全部', max(game_name)) as game_name,

  count(distinct if(event='register_active_detail', user_id, null)) as register_active_detail_num,
  count(distinct if(event='register_active_detail_t_r_button' and task_type =1, user_id, null )) as t_r_button_1_num,
  count(distinct if(event='register_active_detail_t_r_button' and task_type =3, user_id, null )) as t_r_button_3_num,
  count(distinct if(event='register_active_detail_t_r_button' and task_type =5, user_id, null )) as t_r_button_5_num,

  count(distinct if(event='register_active_detail_r_popup' and scene_type=1, user_id, null )) as r_popup_num
from event_data
where event in (
  'register_active_detail'
  , 'register_active_detail_t_r_button'
  , 'register_active_detail_r_popup'
)
group by date_key, game_id
```

业务场景

- 提高固定化粗粒度的聚合数据的查询效率
- 查询时，自动匹配最优的物化视图，减少开发成本

使用原则

- 从查询语句中抽象出，多个查询共有的分组和聚合方式作为物化视图的定义
- 不需要给所有维度组合都创建物化视图
- 为了保证物化视图使用效率，在正式使用的时候也会分析命中物化视图情况，合理构建高价值的物化视图，避免无效视图占用资源

最终效果

- 无需人为指定物化视图，查询自动选择最优视图
- 上卷分析场景命中物化视图超 80%，查询效率提升 30%

遇到的问题

遇到的问题和解决

- **doris AGGREGATE 模型 多副本时 sum值每次结果不一致:**

原因: 开启了 按行合并的方式改变为按列组合并, 每次参与合并的粒度变成列组, 降低单次 compaction 内部参与的数据量, 减少 compaction 期间的内存使用, 从而导致每个副本的聚合结果不一致

解决: 将 `enable_vertical_compaction=false` 关闭后, 重新导入数据, 数据 sum 结果一致

- **1.2.4 版本中 日期 和 非标准格式的日期 进行比较时会返回 null, 例如 `register_date < date_format(date_key,'%Y-%m')`**

原因: Doris 遵守 MySQL 5.0 协议, 导致日期比较时需要两个日期格式才能比较, 在 MySQL 8.0 中是支持的

解决: 反馈给社区后, Doris 在1.2.5版本修复完成

- **Flink 程序写入 Doris 一段时间后会产 ck 异常**

原因: Flink 程序流式写入 Doris, 每次数据进入都会跟 Doris 进行交互, 交互频率过于频繁, 且为多副本, 多次写入后导致产生被压, 从而产生 ck 异常, 导致 flink 程序失败

解决: 1. 开启 `enable_single_replica_load`, 写入时单副本写入, 减轻写入时的压力

2. `flinksql` 写入 Doris 开启攒批模式写入

遇到的问题

遇到的问题和解决

- **多层嵌套 不合理的 SQL在 where 条件中, 导致 FE 节点挂掉**

例如: `SELECT EXTRACT(MONTH FROM date_key) AS month, COUNT(DISTINCT user_id) AS user_count FROM dws_box_user_behavior_stat_dm_di_wanxin WHERE date_key >= '2023-10-02' AND date_key <= '2024-01-31' AND is_login_box = '1' AND user_id NOT IN ((SELECT user_id FROM dws_box_user_behavior_stat_dm_di_wanxin WHERE game_names LIKE '%yzsc%') AND user_id NOT IN (SELECT user_id FROM dws_box_user_behavior_stat_dm_di_wanxin WHERE game_names LIKE '%dwcq%')) OR user_id IN (SELECT user_id FROM dws_box_user_behavior_stat_dm_di_wanxin WHERE game_names LIKE '%tg%') AND date_key NOT IN ('2023-10-01', '2023-11-01', '2023-12-01', '2024-01-01') GROUP BY EXTRACT(MONTH FROM date_key) ORDER BY month LIMIT 30`

原因: SQL由于是业务用 gpt 编写, 存在逻辑不合理之处, 但是不能导致 FE 节点挂掉, 从而导致整体服务受影响

解决: 在 2.0.5 中存在该问题, 会导致集群节点挂掉, 升级2.0.7后该问题修复

- **1.2.5 版本中 使用 union all 存在查询结果不唯一的问题**

原因: Bug

解决: 升级 1.2.6 版本问题解决

04 核心收益与未来展望

实时分析平台带来的业务价值与发展趋势

核心收益

承载OLAP数据分析体系，搭建实时数仓底座

01

平台用户标签

- 涉及用户多业务线属性，并且标签计算周期和更新频率不一致。
- 通过Aggregate模型，按列replace_if_not_null实现不同频次的更新且互不影响
- 相对之前的大宽表的业务模型开发效率提升 30%，数据排查效率提高 50%+。

02

多场景查询

- 满足公司所有业务线的复杂分析查询，在排序，多维等复杂指标场景下达到秒级或者毫秒级的响应，整体查询效率提升 20%+。
- 提升游戏质量反馈速率，从之前小时级别提升到分钟级别，开发量减少 30%。
- 固定维度的查询场景，通过物化视图等相关实现，命中物化视图超 80%，整体查询效率有明显提升。

03

平台运维

- 之前自己搭建doris集群，后来整体采用 SelectDB,针对运维成本至少减少 20%，使更多精力在处理业务上。

未来技术展望

后续未来发展方向



01

构建湖仓一体

- 结合 Paimon + Flinksql + SelectDB 构建湖仓一体架构，实现多链路数据共享。



02

资源隔离

- 物理隔离 (Resource Group)
- 逻辑隔离 (Workload Group)



03

AI大模型融合应用

- 交互式助力业务数据诉求。
- 减少机械化 构建清洗流程，提高人效。



04

CCR 跨集群数据同步

- 用户多集群的数据库表自动同步。



谢谢观看

刘志邓

2024.10