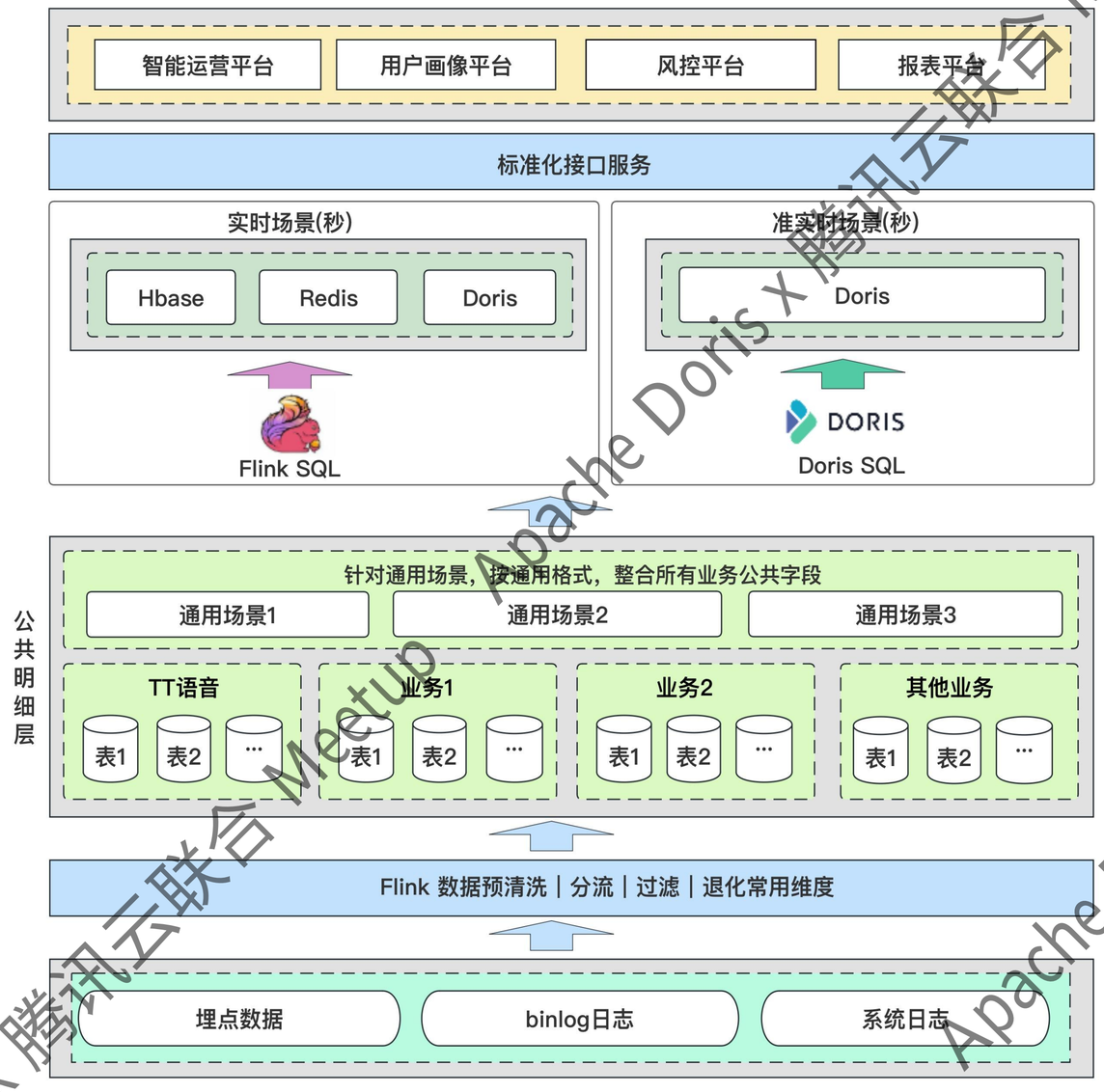
趣丸科技基于Apache Doris的应用实践

陈观潮 高级数仓开发工程师

目录

目录

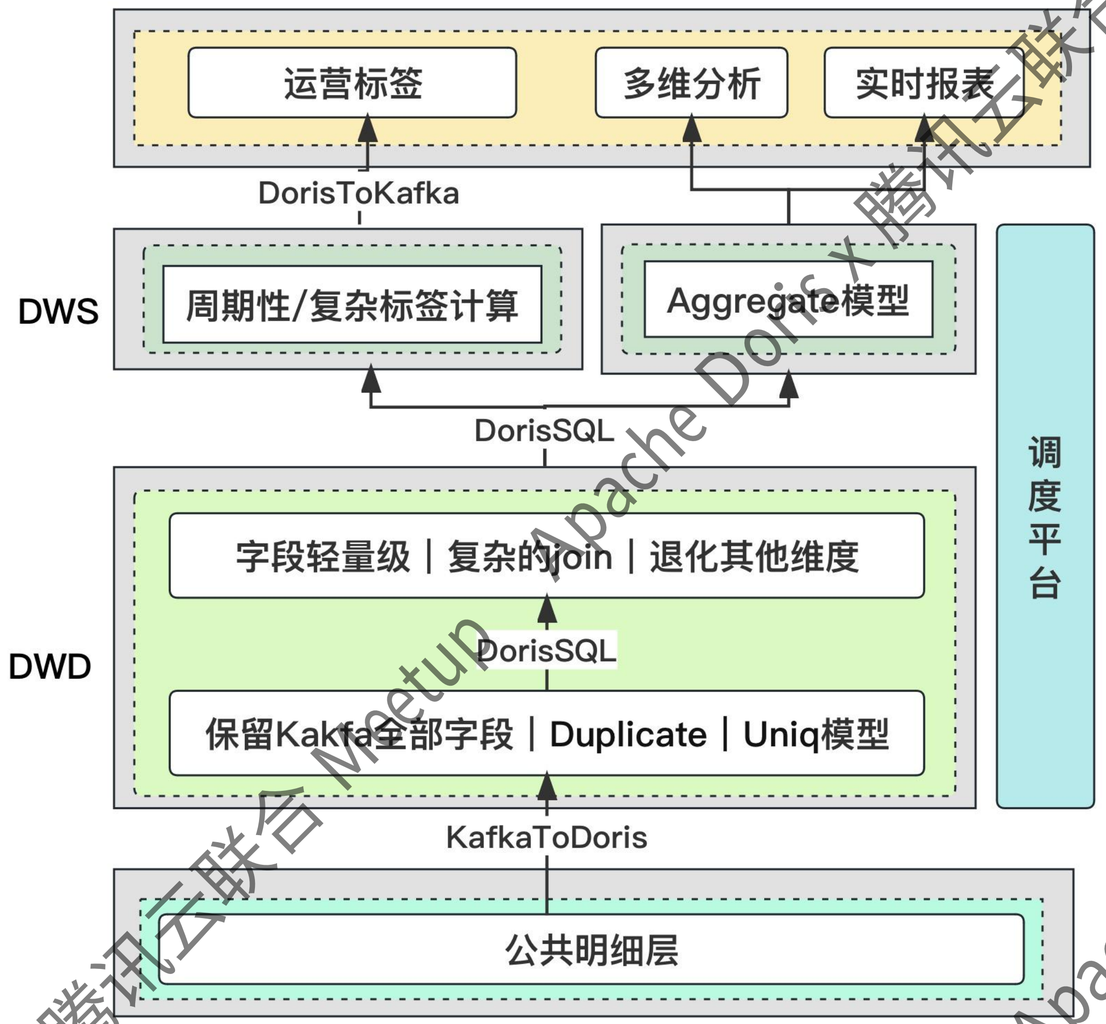
实时数仓架构设计方案



架构设计

- **公共明细层：**
  - 通过 FlinkSQL 对各业务的原始数据做清洗，形成各业务统一标准的 DWD 层数据，供下游使用；
  - 针对通用场景，整个所有业务通用字段，便于下游使用
- **针对时效性高场景：**
  - 基于清洗后的 DWD，通过 FlinkSQL 计算后，写入对应的存储介质，供下游使用；
- **针对准实时场景：**
  - 基于清洗后的 DWD，通过 DorisSQL，分钟级计算，写入 Doris 后，供下游使用。

Doris 在实时场景的架构方案



架构设计

- **DWD 明细层**
  - 基于清洗后的明细数据，通过 FlinkSQL 写入 Doris，作为 Doris 的数据源，通过 Doris 的 Unique 模型，尽可能保留全部字段，方便新增字段时，历史数据的回溯；
  - 基于 Doris 的第一层明细数据，针对复杂的 Join 场景做维度退化，最终字段设计尽可能轻量级；
- **DWS 汇总层**
  - 复杂标签的计算，可在 Doris 计算后，同步至 Kafka 供下游使用；
  - 基于 Doris 的聚合模型，对指标进行预聚合，供上层报表或分析使用；

Doris 使用场景

2022 年上半年引入 Doris

DORIS

主集群规模

- 近百个节点

- 数百个实时任务导入数据
- 数百个微批 ETL 调度任务
- 单表数据量最大增量超百亿/天
- 数百 T 存储

运营活动

- 运营标签
- 营销活动效果分析

风控

- 风控标签
- 风控审核

业务报表

- 数据驾驶舱
- 多维度分析报表
- 分钟趋势报表

算法

- 数据预测
- 模型训练

目录

应用实践

数据同步

数据开发

数据治理

开发优化

数据同步-实时导入

使用 FlinkSQL 实时写入 Doris

Flink Doris Connector特性

1. 落表延迟为Checkpoint间隔
2. 默认开启两阶段提交

Stream Load label前缀必填且全局唯一

衍生问题

- 重启需要修改 Label

取消两阶段提交保证Exactly Once方法

1. 为每条Kafka数据赋id

- ID 规则：Kafka 元数据 timestamp+partition+offset

2. 明细表设为Unique模型，将id纳入key值

```
CREATE TABLE if not exists dwd_table (
    data_date date comment '数据日期'
    ,id varchar(128) comment '唯一键。规则：数据时间(yyyyMMddHHmmss)+分区+offset'
    ...
)
engine=olap
unique key(data_date, id)
...
```

数据同步-离线导入

离线导数方式

缺点

1. 不支持分布式同步
2. 不支持 Exactly Once 语义

数据同步-离线导入

离线导数方式

优点

1. 支持分布式同步
2. 支持 Exactly Once 语义
3. 支持 CDC 同步（探索中）

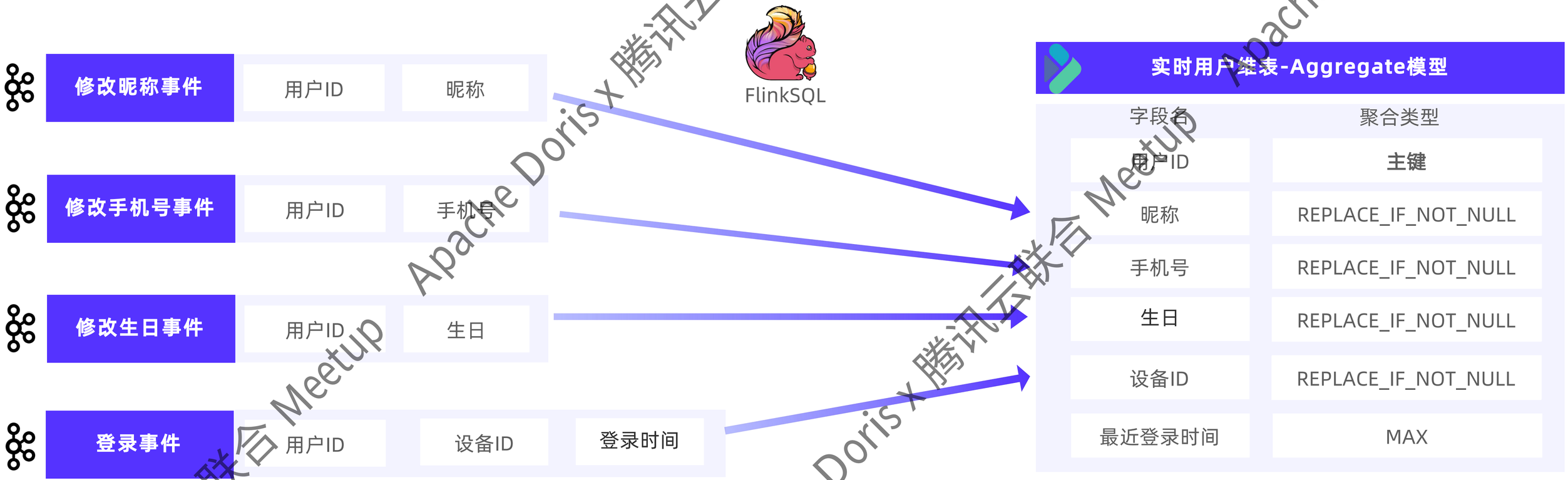
数据开发

实时维表部分列更新

微批调度

1. 延迟=Checkpoint时间+微批调度时间
2. 开发逻辑复杂

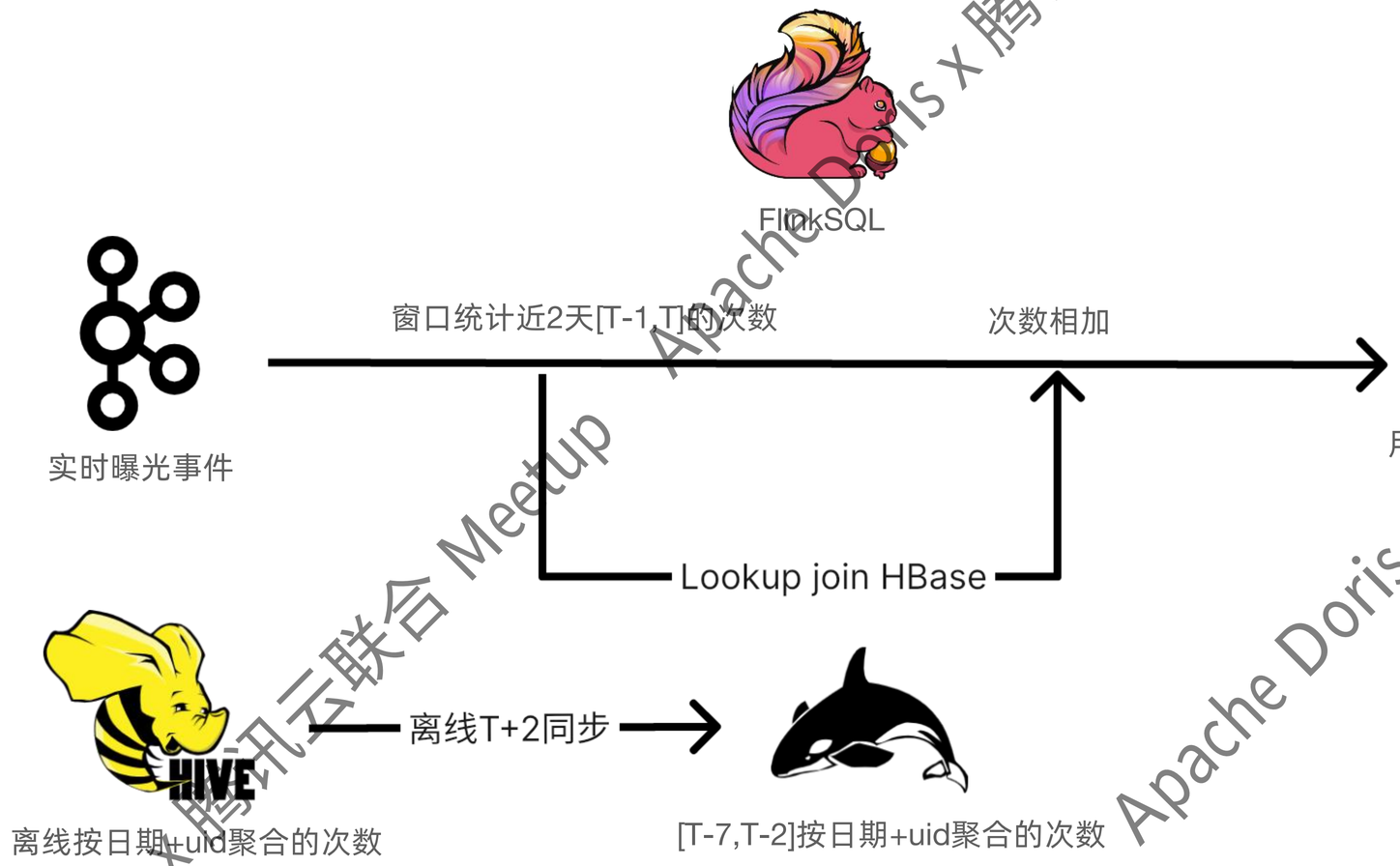
实时维表部分列更新



1. 延迟=Checkpoint时间
2. 逻辑简单清晰

数据开发-特征工程

使用 FlinkSQL 实时计算**每一个用户近 7 天累计曝光次数**，所需要的工作：



痛点

1.开发成本高

2.抗风险能力低

3.运维成本高

数据开发-特征工程

使用 Doris 实时计算**每一个用户近 7 天累计曝光次数**

优点

1.开发成本低

2.抗风险能力高

3.运维成本低

4.模型复用性高

前置

全链路都需要保证EOS语义

数据开发-智能运营

APP推送消息

推送效果分析

通道	计划触达用户数	下发成功用户数	到达用户数	点击用户数	完成首要目标
华为付费	100000	90000	50000	10000	5000
OPPO付费	100000	90000	50000	10000	5000
小米付费	100000	90000	50000	10000	5000
VIVO付费	100000	90000	50000	10000	5000
第三方	100000	90000	50000	10000	5000

数据开发-智能运营

推送事件明细

计划事件	用户ID	计划时间
下发事件	用户ID	下发时间
到达事件	用户ID	到达时间
点击事件	用户ID	点击时间
完成目标事件	用户ID	完成时间

微批调度

取增量事件数据

结果表设计

应用层_智能运营APP推送效果-Aggregate模型

字段名	聚合类型
推送计划ID	主键
厂商通道	主键
计划uid bitmap	BITMAP_UNION
下发uid bitmap	BITMAP_UNION
到达uid bitmap	BITMAP_UNION
点击uid bitmap	BITMAP_UNION
完成目标uid bitmap	BITMAP_UNION

数据治理-监控

腾讯云 TCHouse-D 提供完善的可视化监控页面

数据治理-监控

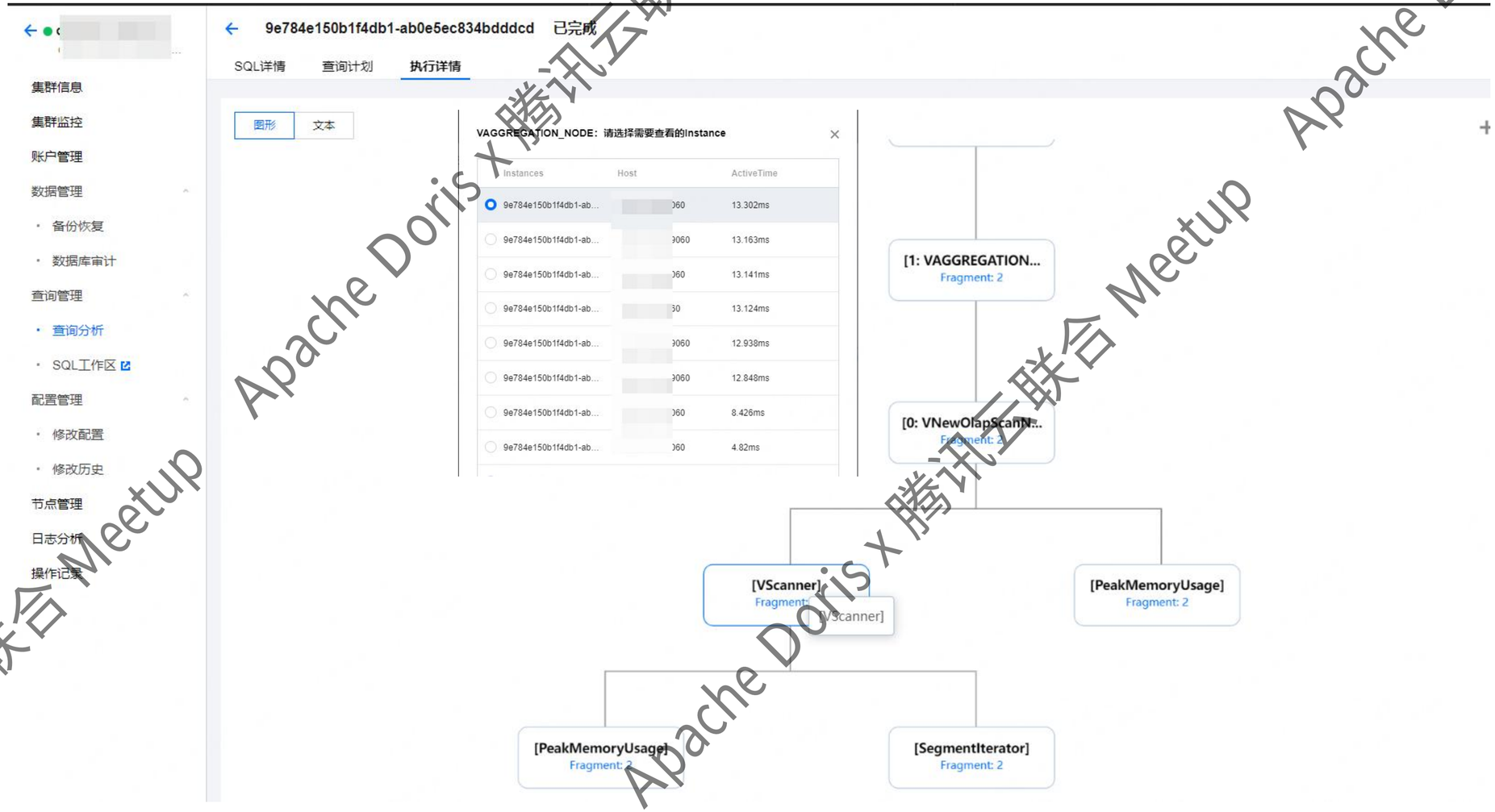
腾讯云 TCHouse-D 提供完善的可视化监控页面

数据治理-监控

腾讯云 TCHouse-D 提供完善的可视化监控页面

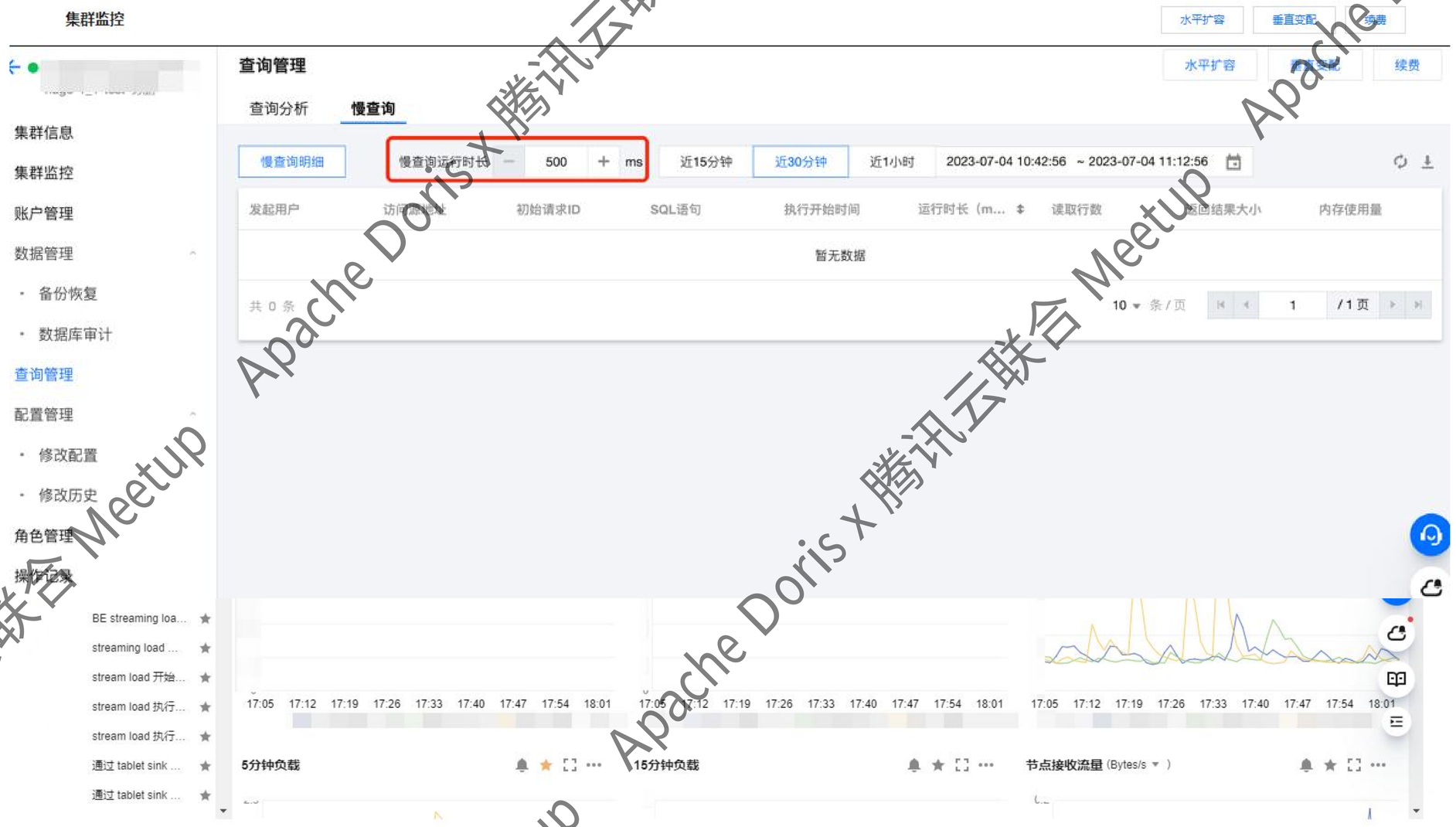
数据治理-监控

腾讯云 TCHouse-D 提供完善的可视化监控页面



数据治理-监控

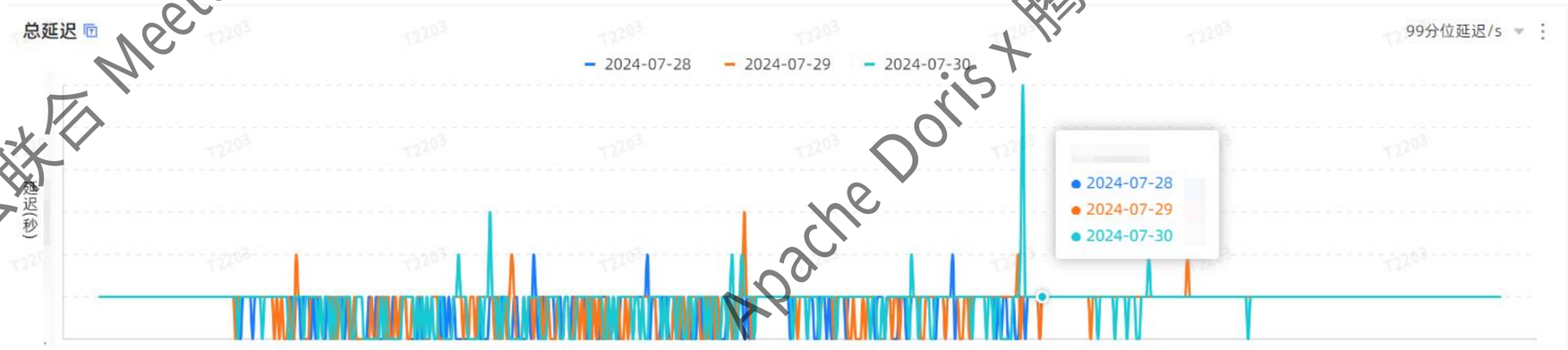
腾讯云 TCHouse-D 提供完善的可视化监控页面



数据治理-监控

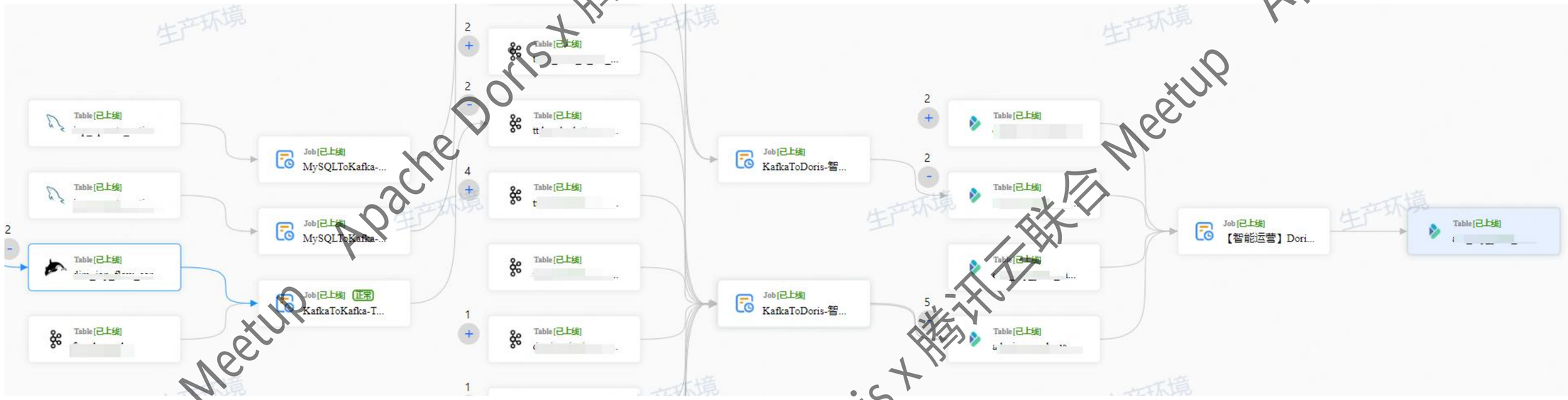
数据量波动明细

P99延迟波动明细



数据治理-血缘

全链路血缘治理

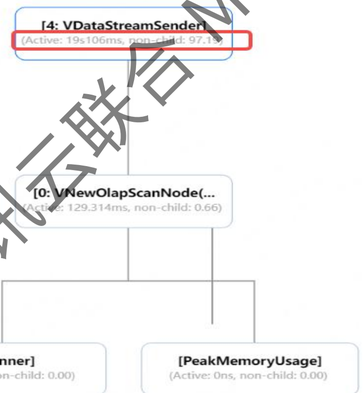


开发调优

优化1：取 Top1 建议用 max_by、min_by 代替 row_number 窗口排序

使用row_number

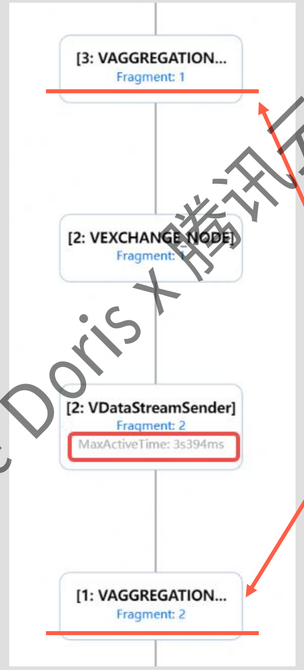
```
select
    *
from
    (
        select
            user_id
            ,event_time
            ,platform
            ,ip
            ,device_id
            ,row_number() over(partition by user_id order by event_time) as rn
        from
            event_table
        where
            data_date = '2024-07-23'
    ) t
where
    rn = 1
```




```
[4: VDataStreamSender]
(Active: 19s106ms, non-child: 97.19)
  - counters:
      - BlocksSent: 6.723K (6723)
      - BrpcSendTime: 4s51ms
      - BrpcSendTime.Wait: 1s914ms
      - BytesSent: 609.28 MB
```

使用min_by

```
select
    user_id
    ,min(event_time) as event_time
    ,min_by(platform, event_time) as platform
    ,min_by(ip, event_time) as ip
    ,min_by(device_id, event_time) as device_id
from
    event_table
where
    data_date = '2024-07-23'
group by
    user_id
```



两阶段提交

```
[2: VDataStreamSender]
(Active: 24.66ms, non-child: 0.70)
  - counters:
      - BlocksSent: 42
      - BrpcSendTime: 13.262ms
      - BrpcSendTime.Wait: 416.577us
      - BytesSent: 8.66 MB
```

开发调优

优化2：表设计优化

索引

- 一行数据的前 36 个字节 作为这行数据的前缀索引。当遇到 VARCHAR 类型时，前缀索引会直接截断
  - ✓ 查询、关联使用频率高的字段放前面，例如user_id
  - ✓ varchar类型尽量放bigint后面

自动分桶

- 建表时设置自动分桶
  - ✓ 初始分桶根据数量划分
  - ✓ 后续分桶数会根据最多前 7 个分区数据量
  - ✓ 事前事后均无需人工评估桶数

```
CREATE TABLE if not exists dwd_table (
    `data_date`          date          comment '数据日期'
   -- 常用查询字段放前面，成为前缀索引
   ,`user_id`            bigint        comment '用户UID'
   -- 前缀索引遇varchar会截断，尽量放在bigint类型后面
   ,`device_id`          varchar(128) comment '设备ID'
)
engine=olap
unique key(`data_date`, `user_id`, `device_id`)
comment "明细层_事件表"
partition by range(`data_date`)()
-- 自动分桶，无需根据量级和存储手动设置
distributed by hash(`data_date`,`user_id`, `device_id`) buckets auto
```

目录

未来展望

存算分离架构

- 使用更为廉价的存储介质以降低成本,数据湖场景下更灵活地弹性部署;

湖仓一体

- 引入数据湖相关技术，探索基于Doris的湖仓一体应用;

资源隔离

- 应用更加灵活的多租户资源隔离方案;

Thanks !