

# Webex 基于 Apache Doris 的 故障分析实践

赵发凯

Cisco Cloud Engineer



# 目录 / Agenda


01 背景介绍

02 架构变迁

03 业务实践

04 未来规划

# 背景介绍



故障分析

大数据平台

数据开发、数据治理、  
数据可视化...

大数据基础组件

Kafka、Spark、Flink、ELK...

- **指标采集**

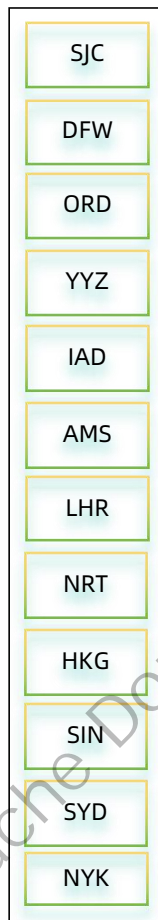
- 全球机房数据
- 跨云端服务

- **分析及故障排除**

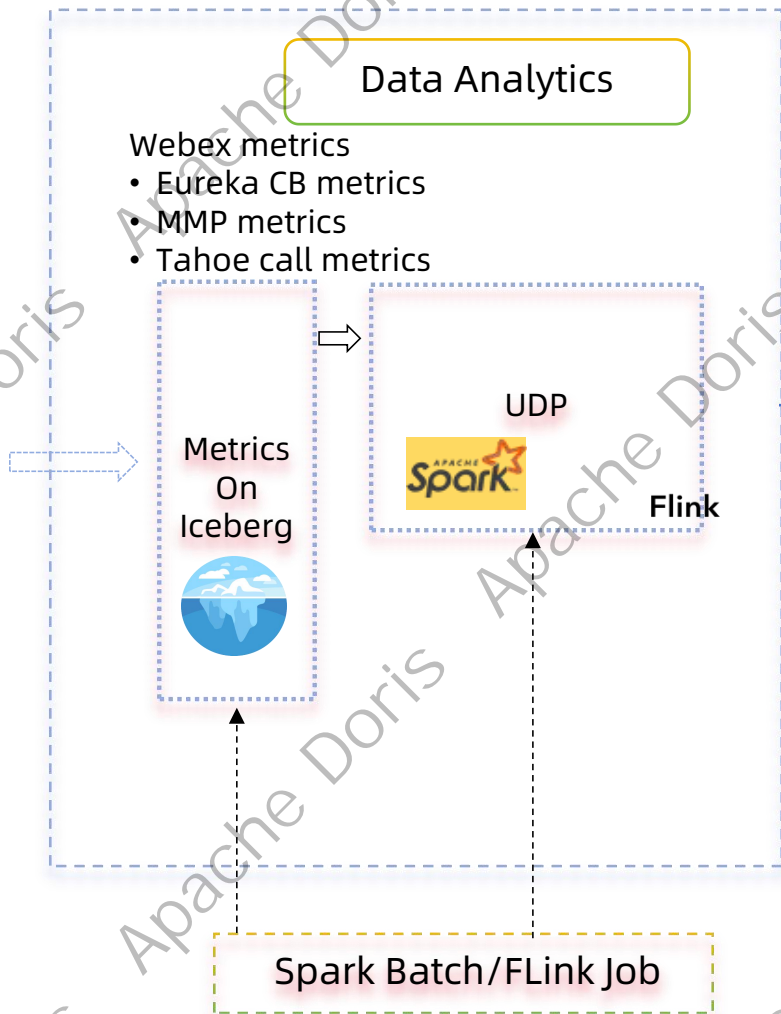
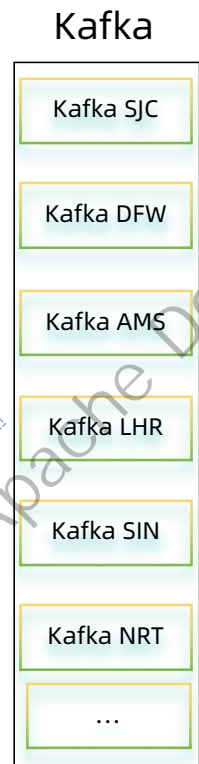
- Webex 会议
- Webex 用户
- Webex 数据中心
- Webex 呼叫中心
- Webex 联络中心

# 整体架构

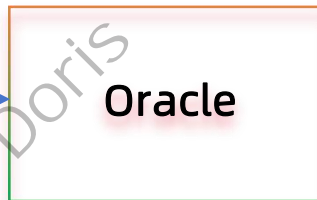
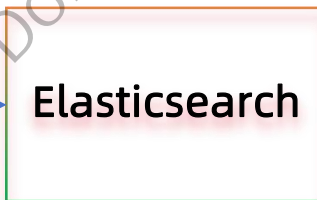
Metrics logs in  
Eureka/Tahoe/MMP/MQE



Logstash



- Webex metrics
- Eureka CB metrics
  - MMP metrics
  - Tahoe call metrics



# 痛点与需求

## 架构痛点

- 架构复杂，技术栈繁多
- 需求研发成本高
- 实时数据需求成本高
- 所需硬件资源多
- 查询效率低
- Apache Pinot 功能支持弱

## 目标需求

- 架构简单，易于使用、维护
- 统一技术栈，降低学习、使用成本
- 时效性高，支持实时数据导入
- 功能多，快速支持需求
- 查询效率、并发度高

# 目录 / Agenda

01 背景介绍

02 架构变迁

03 业务实践

04 未来规划

# 技术选型

关键指标	Apache Doris
查询性能	TPC、SSB 测试及业务场景测试均满足需求，1.1 之后支持向量化执行，支持查询优化器等，有更多的性能提升
时效性	支持 Routine Load、Stream Load，并且有事务支持，让实时导入数据更高效、安全
功能	<ul style="list-style-type: none"><li>• 支持 Aggregate、Unique、Duplicate 数据模型，让业务有更多选择空间。</li><li>• 并且支持物化视图，让数据聚合更轻松</li><li>• 2.0 支持倒排索引，让全文检索更容易</li><li>• 2.1 版本后支持 VARIANT 类型，对 JSON 操作更简单、高效</li></ul>
使用方式	兼容 MySQL 协议，学习成本基本为 0；接入和迁移成本低
运维部署	<ul style="list-style-type: none"><li>• 架构简单，不依赖第三方组件，扩缩容简单</li><li>• 数据自动平衡到各 BE，运维更省心</li><li>• 支持 zstd 等高效压缩方式，更节省存储资源</li></ul>
开源方式	Apache License 开源协议，安全性高、灵活性强
其它	文档、资料丰富；社区活跃度高

# 基于 Apache Doris 的架构

Metrics logs in Eureka/Tahoe/MMP/MQE

- SJC
- DFW
- ORD
- YYZ
- IAD
- AMS
- LHR
- NRT
- HKG
- SIN
- SYD
- NYK

Logstash

Kafka

- Kafka SJC
- Kafka DFW
- Kafka AMS
- Kafka LHR
- Kafka SIN
- Kafka NRT
- ...

Webex metrics

- Eureka CB metrics
- MMP metrics
- Tahoe call metrics

Data Analytics

Metrics On Iceberg



UDP



Flink

Kafka

Spark Batch/FLink Job

Analytics Service

REST APIs

MATS Portal



DORIS

Data Visualization

- Meeting
- MMP
- Telephony
- CMR



- Executive
- AVOPS
- Engineering
- TAC/CSM
- CSRE



# 集群规模

4

集群总数  
各机房做冗余备份

30+

总节点数  
独立部署 FE/BE

100000+

在线服务  
平均每日查询总数量

40TB+

存储总量  
按照业务设置不同的 retention

5TB+

通过 Flink 实时  
导入的日增数据量

# 目录 / Agenda

01 背景介绍

02 架构变迁

03 业务实践

04 未来规划

# 业务需要解决问题

## 数据导入

- 时效性
- 稳定性
- 高效性

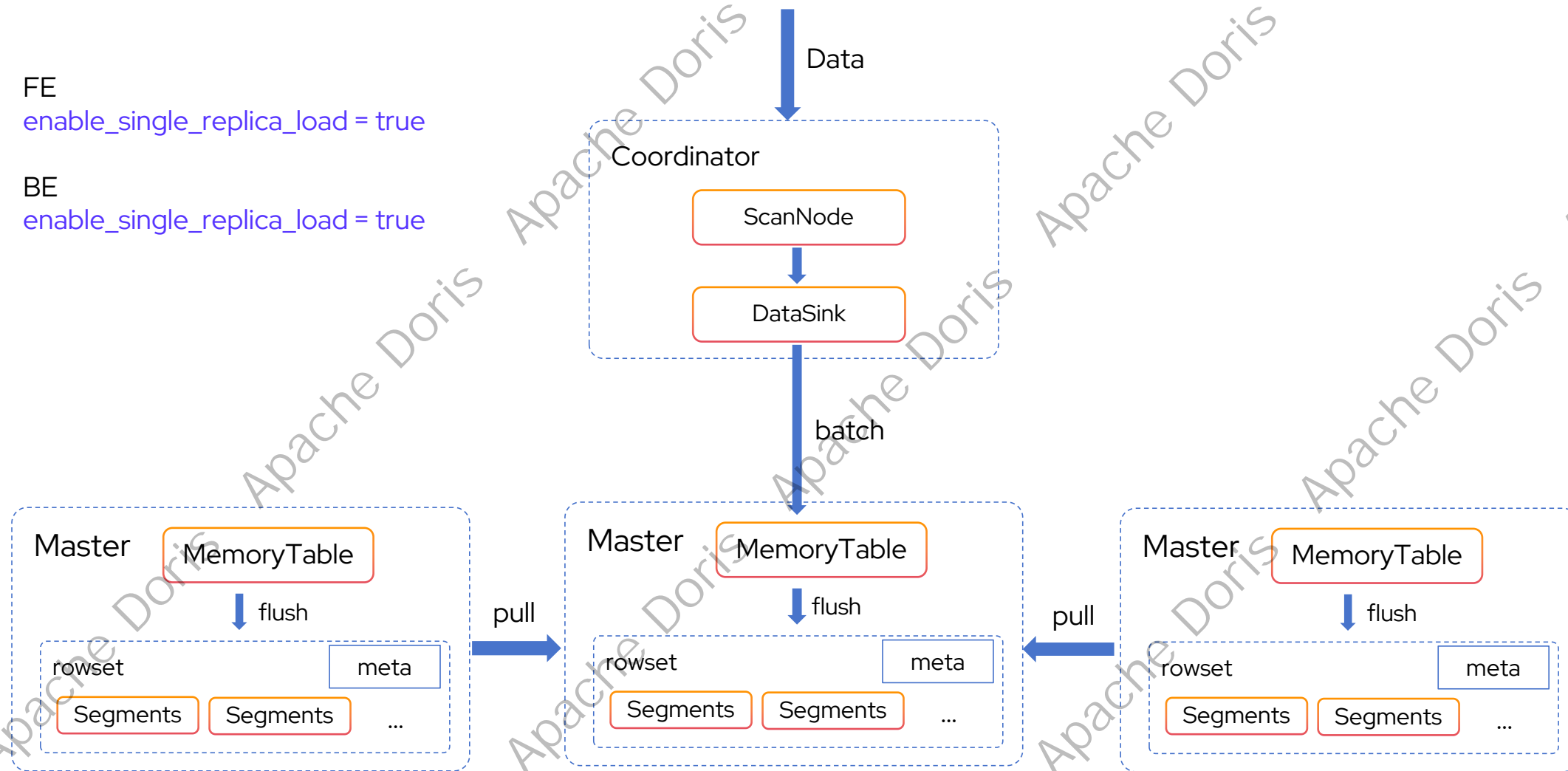
## 查询性能

- 毫秒级查询结果
- 高并发

# 数据导入-单副本

FE  
enable\_single\_replica\_load = true

BE  
enable\_single\_replica\_load = true



# 数据导入-Compaction 优化

`max_tablet_version_num=2000`

增加 tablet 版本数，防止因为 Compaction 执行速度不够导致导入数据失败

`compaction_task_num_per_disk=4`

增加单磁盘 Compaction 频率提升效率

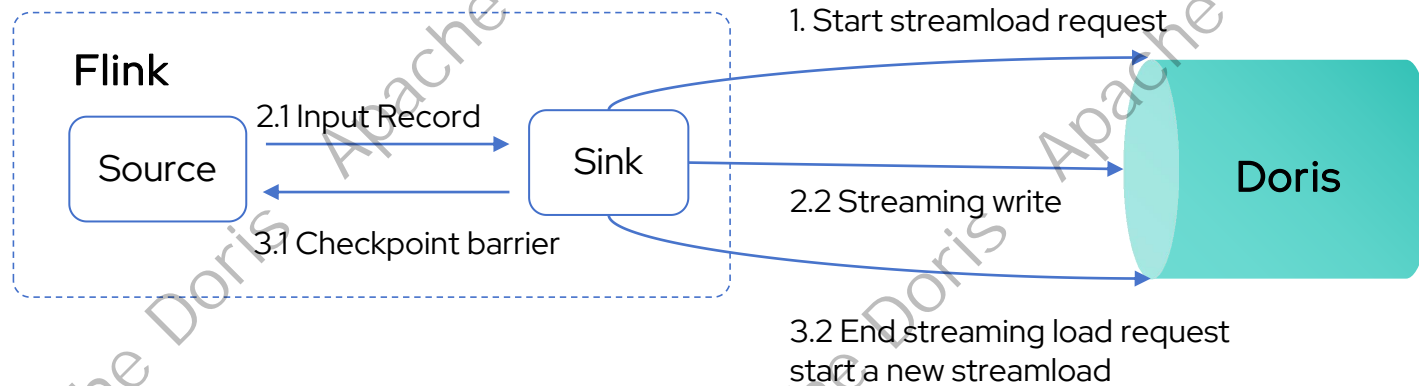
`dacompaction_task_num_per_fast_disk=16`

增加 Compaction 总线程数

`enable_vertical_compaction = true`

降低内存开销、提升 Compaction 的执行速度

# 数据导入-Flink优化



1. 增加 **streaming\_load\_json\_max\_mb** 单次 batch 的 json 数据大小提升导入效率

2. 减小 **cumulative\_size\_based\_promotion\_min\_size\_mbytes**，尽量减少单次数据导入量，提升数据时效性

1. 缩短 Flink **checkpoint interval** = 30s，减少因为大量数据导入 compaction 压力

2. 开启 **2pc**，提升导入数据的安全性

# 数据查询-建表经验

## 善于利用前缀索引

- 前缀索引无需显示定义
- 建表时自动取表的 36 字节作为前缀索引
- 建表字段顺序很关键

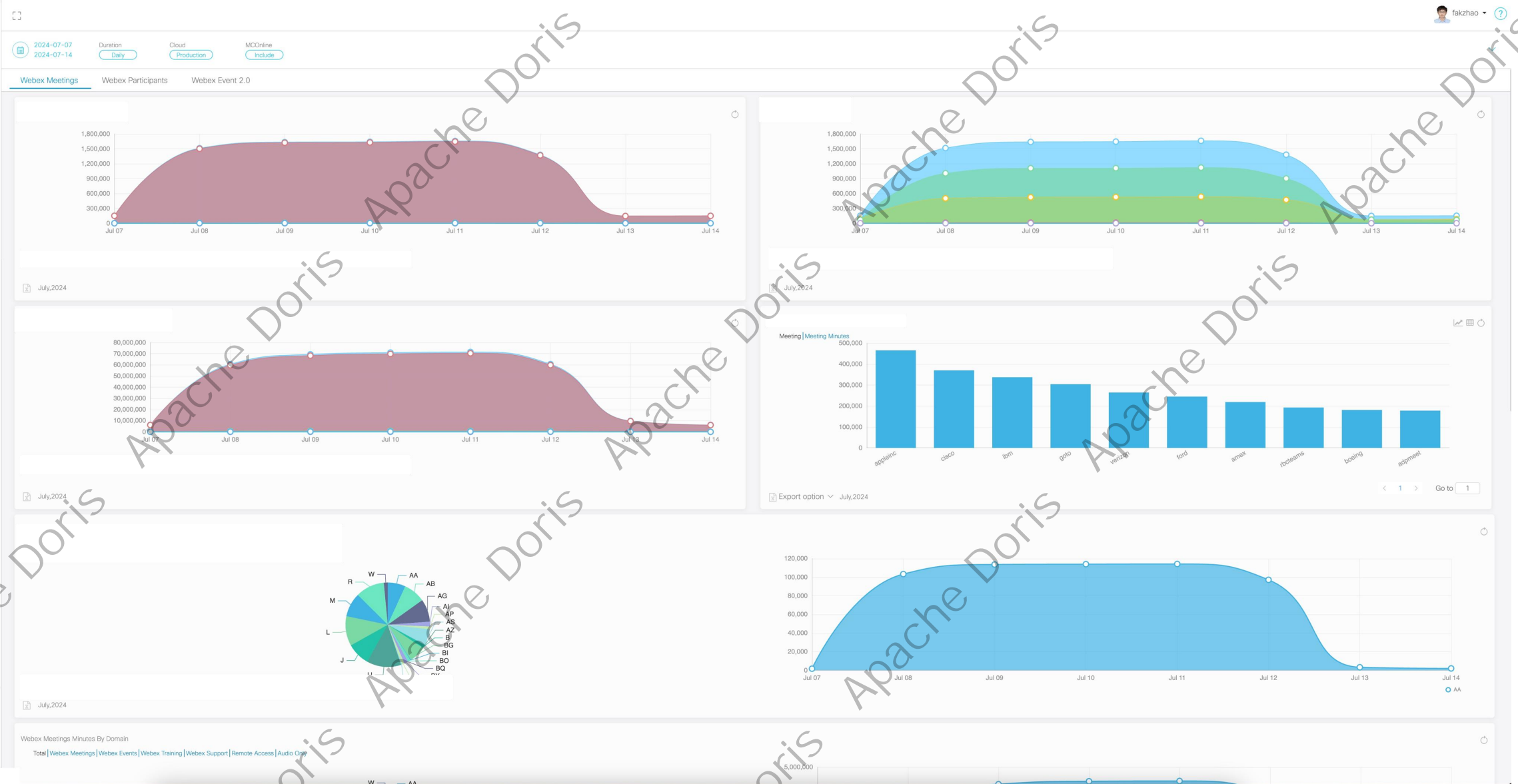
## 索引选择

- 给等值查询热点列加上 BloomFilter 索引
- LIKE 查询字段加 N-Gram 索引

## 分桶列选择

- **查询吞吐** 和 **查询并发** 之间的一种权衡
- 如果选择多个分桶列，则数据分布更均匀。如果一个查询条件不包含所有分桶列的等值条件，那么该查询会触发所有分桶同时扫描，这样查询的吞吐会增加，单个查询的延迟随之降低。这个方式适合大吞吐低并发的查询场景
- 如果仅选择一个或少数分桶列，则对应的点查询可以仅触发一个分桶扫描。此时，当多个点查询并发时，这些查询有较大的概率分别触发不同的分桶扫描，各个查询之间的 IO 影响较小（尤其当不同桶分布在不同磁盘上时），所以这种方式适合高并发的点查询场景

# 效果展示-物化视图







# 效果展示-日志检索

The screenshot displays the Apache Doris web interface for log search. The breadcrumb navigation shows 'Home > Explore'. The current workspace is 'WAP - Loki'. The 'Label filters' section contains three filters: 'db = mats\_doris', 'table = internal.logs.mats\_guest\_logs', and 'timeColumn = dt'. Below the filters, the first step of the query is shown: `{db="mats_doris", table="internal.logs.mats_guest_logs", timeColumn="dt"}` with the description 'Fetch all log lines matching label filters.' The second step is a query operation: `<expr> |= ``` with the description 'Return log lines that contain string ``'. At the bottom, the final query is displayed: `{db="mats_doris", table="internal.logs.mats_guest_logs", timeColumn="dt"} |= ```. The 'Options' section at the bottom indicates 'Type: Range' and 'Line limit: 1000'.

Home > Explore

Outline WAP - Loki

A (WAP - Loki)

Kick start your query Label browser Explain query

Label filters

db = mats\_doris table = internal.logs.mats\_guest\_logs timeColumn = dt

1 `{db="mats_doris", table="internal.logs.mats_guest_logs", timeColumn="dt"}`  
Fetch all log lines matching label filters.

Line contains  + Operations

2 `<expr> |= ```  
Return log lines that contain string ``.

`{db="mats_doris", table="internal.logs.mats_guest_logs", timeColumn="dt"} |= ```

> Options Type: Range Line limit: 1000

# 整体收益

关键指标	收益
查询性能	在线服务 Query 性能提升 <b>40%</b> (对比 Apache Pinot)
时效性	Dashboard 数据由原来 T+1 或 H+n 提升到 <b>秒级</b> 延时 (对比 Oracle 存储)
存储空间	开启zstd压缩后, 存储空间节省近 <b>80%</b> (对比 Apache Pinot)
机器资源	机器规模从原来的 64 台缩减到现在的 14 台,节省近 <b>85%</b> (单机房对比 Apache Pinot、Elasticsearch)
开发成本	由之前需要开发多个 Job, 读不同表业务代码中聚合相比, 现在只需要一个 Job 一个 SQL; 开发效率提升 <b>100%</b>

# 技术选型

关键指标	收益
查询性能	在线服务 Query 性能提升 <b>40%</b> (对比 Apache Pinot)
时效性	Dashboard 数据由原来 T+1 或 H+n 提升到 <b>秒级</b> 延时 (对比 Oracle 存储)
存储空间	开启zstd压缩后, 存储空间节省近 <b>80%</b> (对比 Apache Pinot)
机器资源	机器规模从原来的 64 台缩减到现在的 14 台,节省近 <b>85%</b> (单机房对比Apache Pinot、Elasticsearch)
开发成本	由之前需要开发多个 Job, 读不同表业务代码中聚合相比, 现在只需要一个 Job 一个 SQL; 开发效率提升 <b>100%</b>

# 目录 / Agenda

01 背景介绍

02 架构变迁

03 业务实践

04 未来规划

# 未来规划

## 存算分离架构

- 引入更为廉价的存储介质降低成本
- 在数据湖场景下更灵活的弹性部署
- 更好的实现读写分离、多租户功能

## 向量存储

- 探索 Doris 在 LLM RAG 中的应用场景
- 结合倒排索引探索 LLM 中多路召回场景

## 数据湖分析

- 利用 Doris 的 Catalog 加速数据湖上的查询

**Thanks !**

