

Apache Doris 在招联金融 的多场景应用实践

严奕华

招联金融 数仓团队负责人



个人介绍



严奕华

招联金融 数仓团队负责人

负责招联金融大数据基础平台的架构及对应团队。

目录

01 背景介绍

02 招联金融数仓生态

03 应用场景与案例

04 未来展望

01 背景介绍

成立于 2015 年 3 月，是经中国银保监会批准、由招商银行和中国联通共同组建的持牌消费金融公司，注册地址位于深圳前海，注册资本 100 亿元。

打造两大 **纯线上** 消费金融产品体系



好期货

自主支付类：无担保，纯凭个人信用申请的个人消费信用贷款。获得额度后，客户可通过指定客户端，随时借款，资金实时到账。

系列产品：信用贷、白领贷、大期货、员工贷等。

信用付

受托支付或消费分期类：客户通过消费场景切入申请流程，先申请“信用付”额度，后在消费场景直接使用额度消费。

系列产品：运营商信用付、非运营商线下3C分期、信用花等。

01 背景介绍

数据分析

业务支撑

大数据平台研发

数据开发、数据可视化、数据治理...

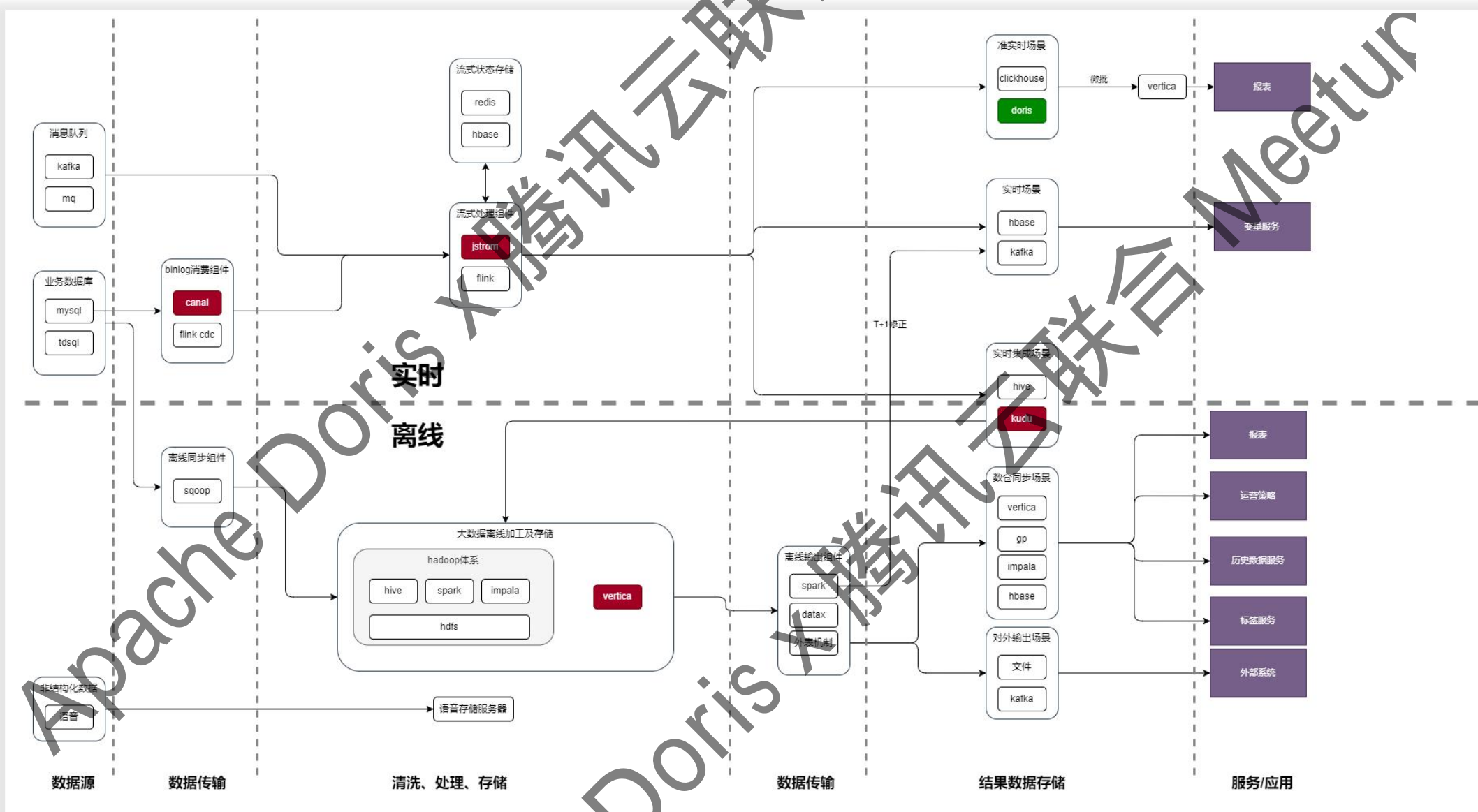
大数据基础组件支撑

Hadoop、Kafka、Flink、Spark、Vertica、ClickHouse、Hbase、Greenplum...

招联-数智平台部负责构建和迭代招联金融的数据中台。

本次分享将详细介绍 Doris 在招联业务场景落地的实践经验。

01 背景介绍



01 背景介绍

难点

- 架构复杂，技术组件繁多，部分组件闭源，运维困难；
- 资源利用率低，没法灵活扩容；
- 数据时效性与查询效率较低；

需求

- 统一组件标准，降低组件维护难度，对平台升级迭代可控；
- 弹性扩容、离在线混部，提升资源利用率，实现降本增效；
- 分析实时化，搭建实时数仓能力；
- 完整开放体系：云原生

目录

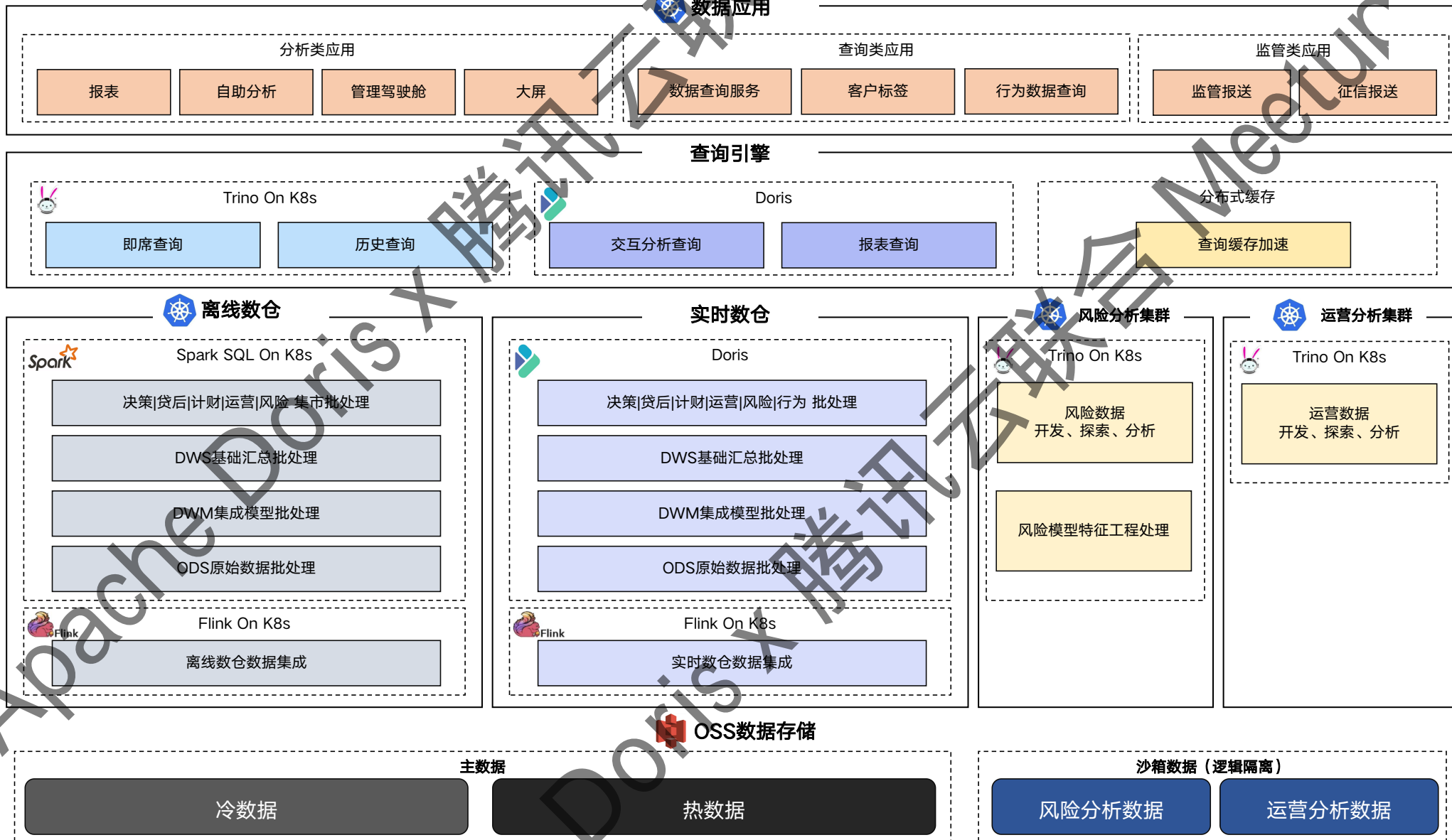
01 背景介绍

02 招联金融数仓生态

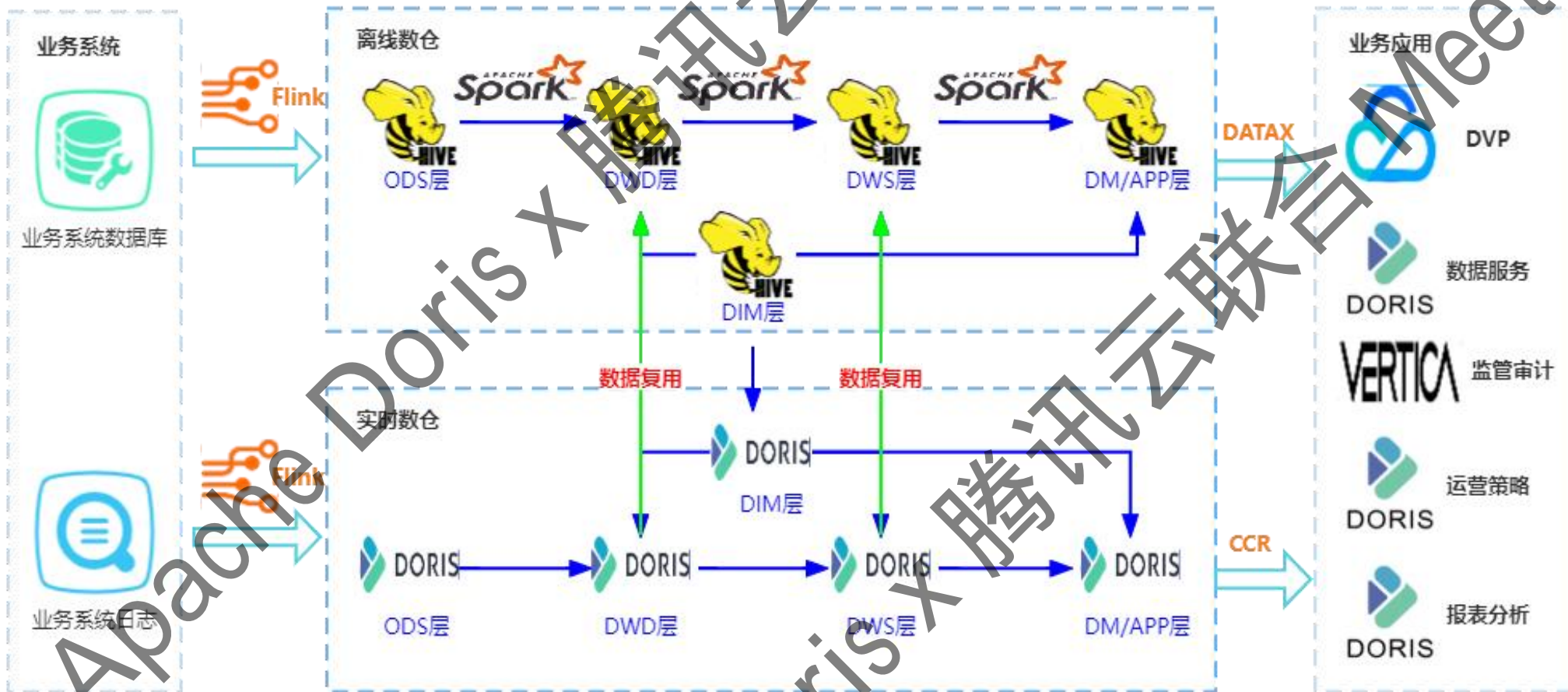
03 应用场景与案例

04 未来展望

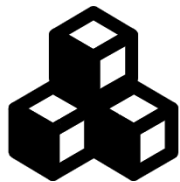
02 数仓生态



02 数仓生态

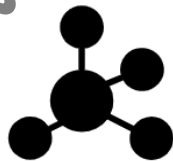


02 数仓生态：Doris 集群规模



5+

当前总集群数



100+

当前集群节点数



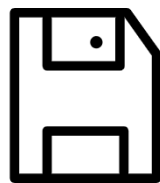
40+

项目数



8W+

某集群峰值QPS



280T+

最大集群数据量



2.5TB+

最大日增数据量

02 数仓生态：Doris 核心场景



实时数仓

- 准实时标签



数据运营

- 标签服务
- 客群筛选
- 交互运营



业务报表

- 高管驾驶舱
- 业务监控报表
- 多维分析报表



模型算法

- 模型训练
- 策略旁路

目录

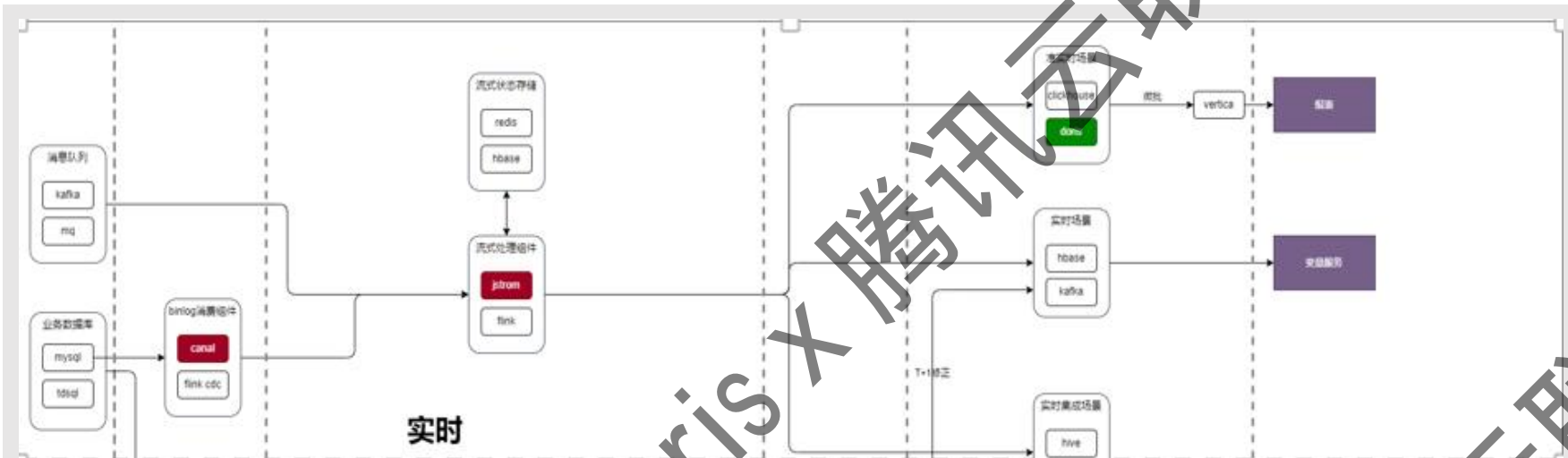
01 背景介绍

02 招联金融数仓生态

03 应用场景与案例

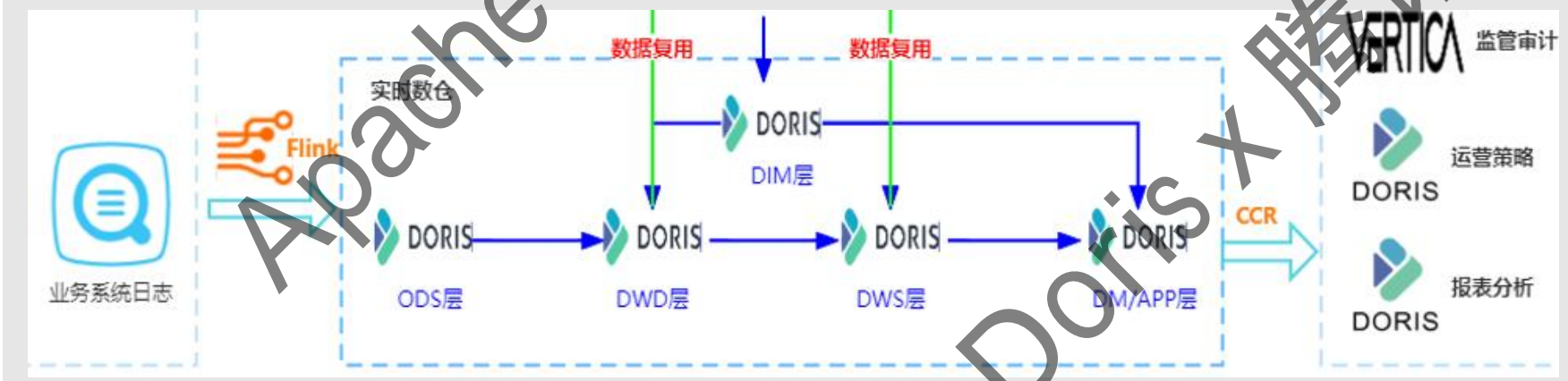
04 未来展望

03 应用场景与案例：实时数仓场景（微批）



优点

- 架构简单：调度引擎+计算引擎（Doris），相比原有的Flink架构，降低技术复杂度的同时，成本降低约10%
- 学习成本低：离线数仓无缝迁移，开发成本大大降低
- 运维简单



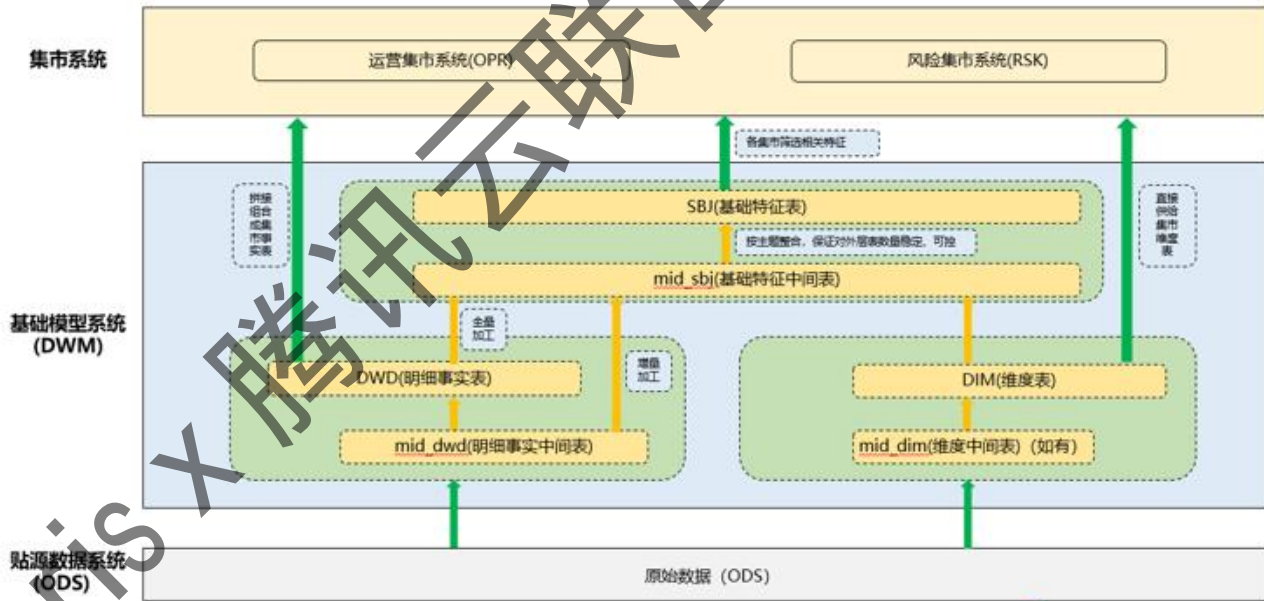
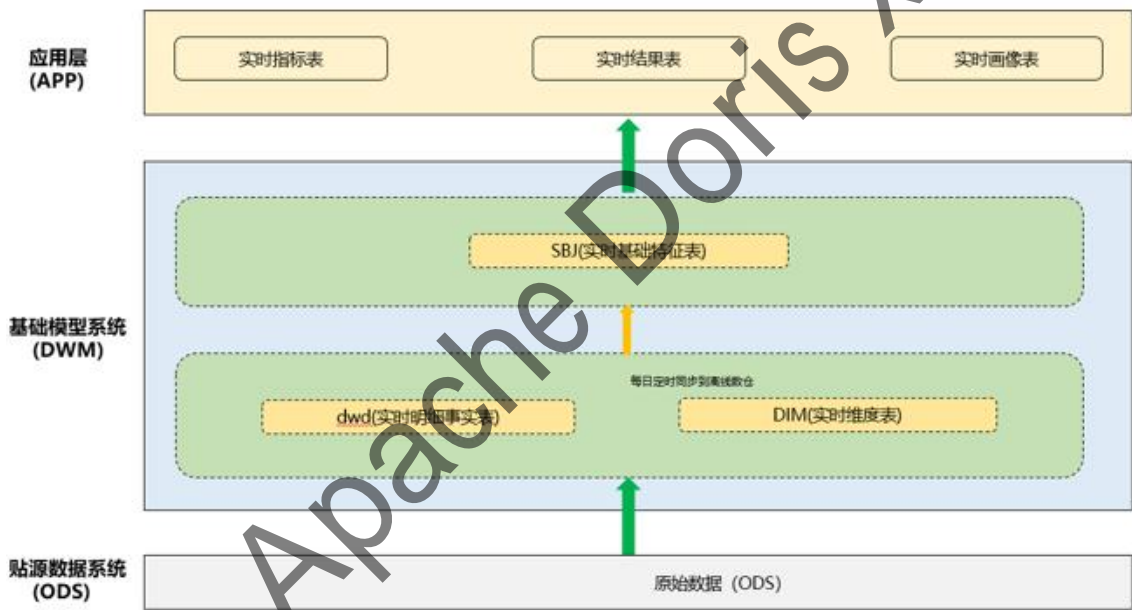
03 应用场景与案例：实时数仓场景（微批）

与离线数仓相比，同样采用分层分域的架构，但层级更少（由5层减为3层）

实时数仓架构

VS

离线数仓架构



03 应用场景与案例：实时数仓场景（微批）



关键点

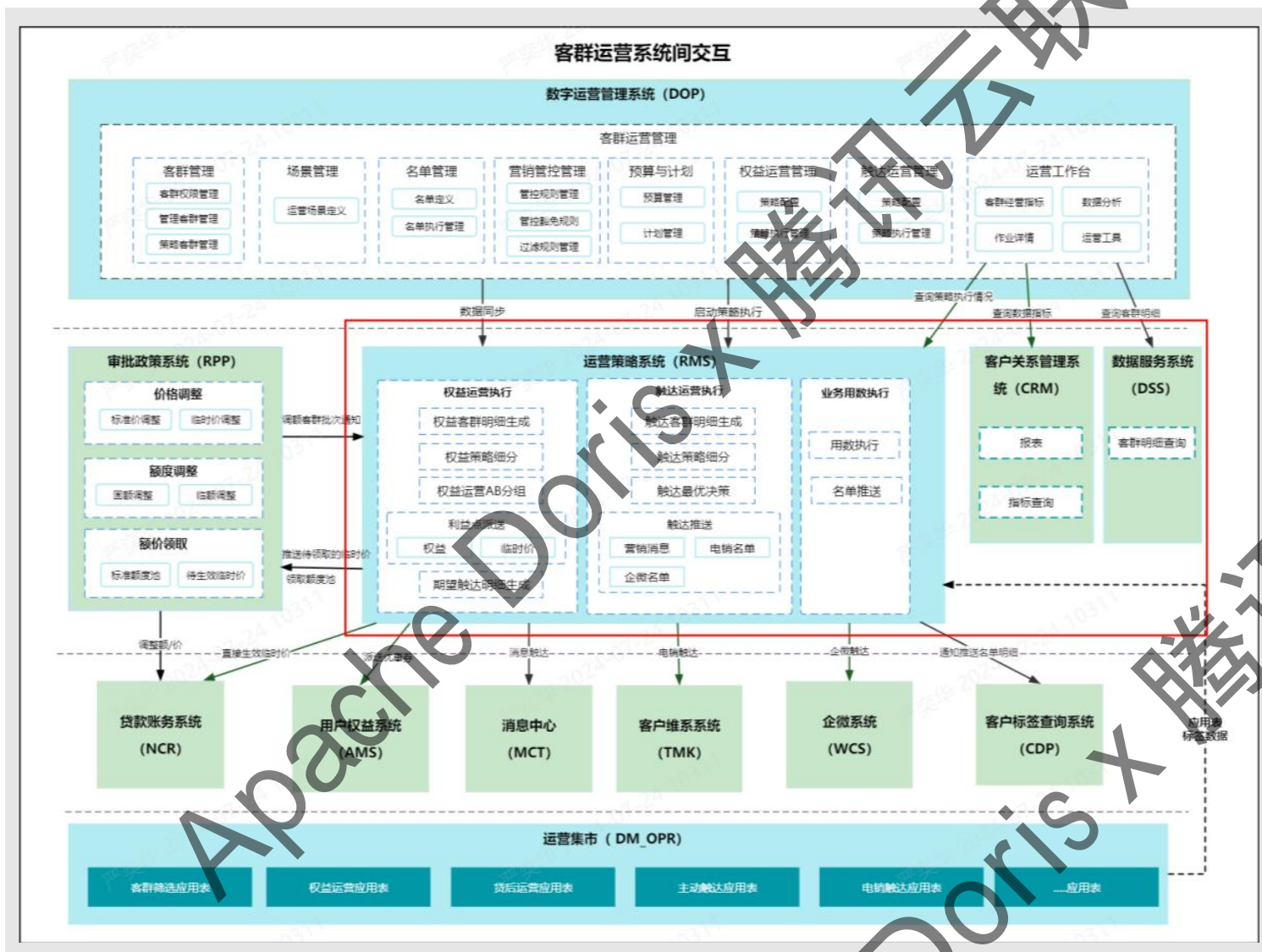
- watermark 机制

调度引擎实现了 Flink 中的 watermark 机制来容忍数据的晚到

- 任务串行

多批次串行，避免数据乱序导致数据不准确

03 应用场景与案例：数据运营场景



精简架构

- 统一存储和计算引擎为 Doris，剔除旧的 Spark+Impala+Vertica 等方案
- 数据传输用 CCR，实现读写分离

03 应用场景与案例：数据运营场景



性能提升

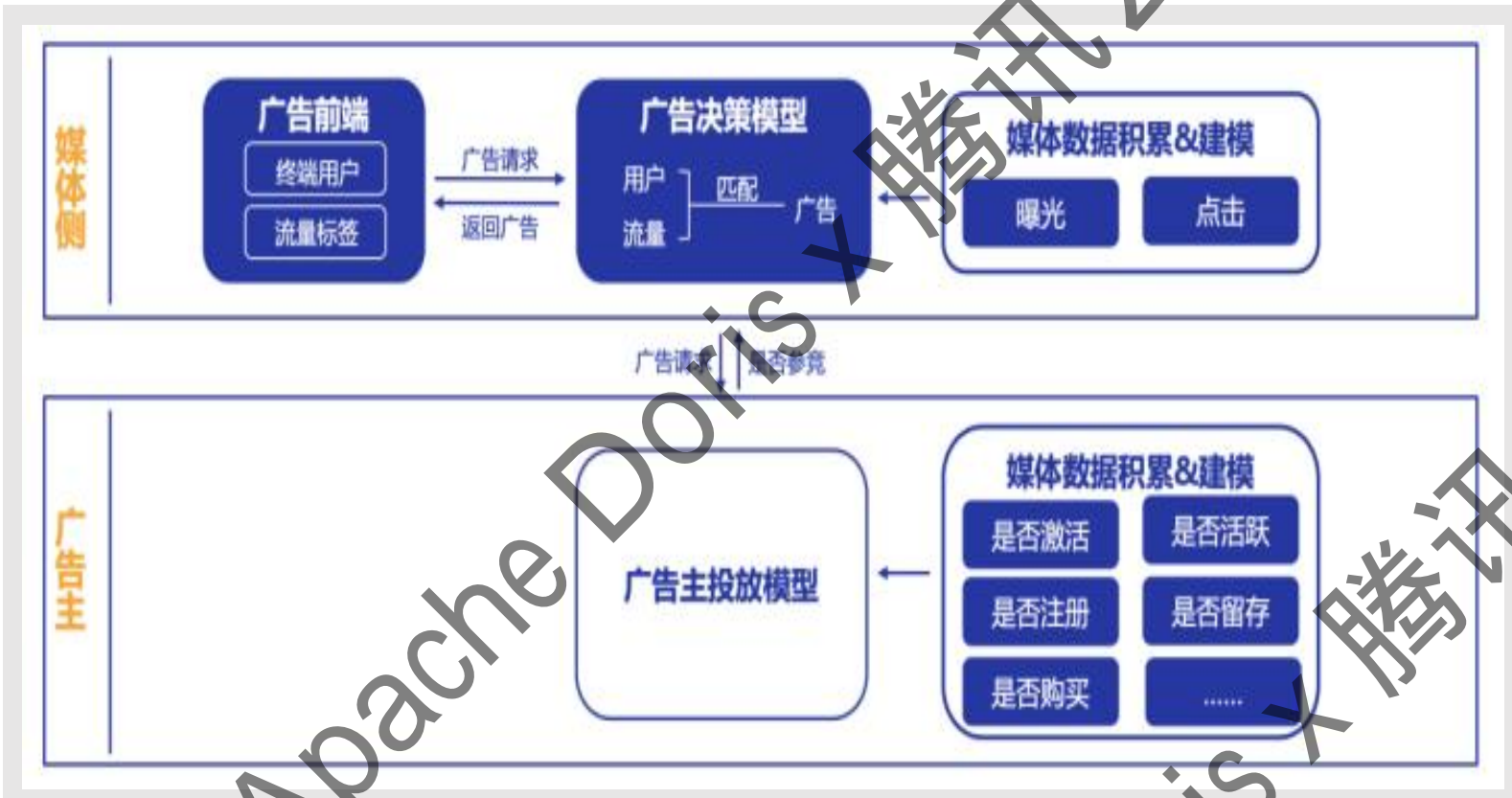
多张标签表关联，生成数据量两亿四千多万

- Vertica: 耗时 30-60min
- Doris: 耗时 5-10min

6倍



03 应用场景与案例：高频点查场景（RTA）



要求点

- QPS 峰值: 10W/S
- 单次接口耗时: 不超过 60ms
(抛开网络耗时和程序处理耗时, 数据查询耗时控制在 15ms)
- 单次数据更新: 20 亿+

03 应用场景与案例：高频点查场景（RTA）

```
CREATE TABLE IF NOT EXISTS tmp.rta_index
(
  device_no VARCHAR(33) NOT NULL COMMENT "设备号",
  seq_01_tag DOUBLE COMMENT "指标1",
  seq_02_tag DOUBLE COMMENT "指标2",
  seq_03_tag DOUBLE COMMENT "指标3",
  seq_04_tag DOUBLE COMMENT "指标4",
  seq_05_tag DOUBLE COMMENT "指标5",
  seq_06_tag DOUBLE COMMENT "指标6",
  seq_07_tag DOUBLE COMMENT "指标7",
  seq_08_tag DOUBLE COMMENT "指标8",
  seq_09_tag DOUBLE COMMENT "指标9",
  seq_10_tag DOUBLE COMMENT "指标10"
)
UNIQUE KEY('device_no')
DISTRIBUTED BY HASH('device_no') BUCKETS 64
PROPERTIES (
  "replication_allocation" = "tag.location.default: 3",
  "enable_unique_key_merge_on_write" = "true",
  "store_row_column" = "true",
  "light_schema_change" = "true",
  "disable_auto_compaction" = "false",
  "storage_medium"="SSD"
);
```

角色	数量	CPU	内存	磁盘
FE	4	11C	64G	500G HDD
BE	7	11C	64G	1T SSD
NG	2	2C	4G	200G HDD

降本
增效

60亿+ 数据	成本
Redis (1T内存)	4.6万/月
Doris	1.4万/月

03 应用场景与案例：数据传输场景（CCR）

● 为啥要用CCR

CCR（Cross Cluster Replication）即跨集群数据复制，能够在库/表级别将源集群的数据变更同步到目标集群，可用于提升在线服务的数据可用性、隔离在离线负载、建设两地三中心等。

● CCR性能

数据量（单位万）	大小（含3副本）	耗时（包含建表、建立调度任务、同步）
100	934.236M	54s
1000	21.897G	192s
10000	219.994G	1559s

要求点

增量（单位万）	插入前存量数据（单位万）	原表插入时间	目标表延迟
100	10000	9.2s	< 3s
1000	10100	49.5s	< 9s
1000	11100	44.9s	< 9s
10000	12100	471.8s	< 54s

● 注意事项

- ◆ 源表禁用 buckets auto;
- ◆ 源表禁用 insert overwrite table/partition; truncate table /partition ;
- ◆ 因为表Id, 分区Id 重置, CCR 作业有可能会出异常源表找不到 table 和分区的问题, 且不可修复;
- ◆ 同步数据的账户 user 需要 admin 权限, 权限有些过大

03 经验分享

- **Mysql Load: Unexpected exception:bad packet sequnce**

副本写入过于频繁，当节点有问题时，导致副本找不到导致问题

通过调整tablet_rowset_stale_sweep_time_sec参数，降低出现此类问题的概率

- **CCR 超时: (TRollbackTxnResult_({Status:TStatus({StatusCode:OK
ErrorMsgs:[{}]} MasterAdd ress:<nil>}))**

网络波动存在丢包导致 RPC 超时，确保网络稳定

升级 CCR 版本至支持设置 RPC 超时时间的 2.1.4 版本

- **Create table as 语法导致的 slot 一系列问题**

2.0 版本在处理 create table as 语句时，用的是旧执行优化器，旧执行优化器因为列字段裁剪，普遍存在 slot 相关问题。

2.1 以上的版本，slot 相关问题才完全解决；

规避手段：先创建临时表 xxx，执行 set enable_nereids_dml = 'true'。

目录

01 背景介绍

02 招联金融数仓生态

03 应用场景与案例

04 未来展望

04 未来展望

一

云上实时数仓

- 精简云上架构

Flink/Jstorm+Redis/HBase /ClickHouse 优化成

Flink+MPP数据库(Doris)的流批一体架构，实时+跑批同一套 SQL 代码

- 实时集成
- 实时加工

二

容器化

- 对计算组件容器化
- 敏捷、高效的部署和管理
- 统一生产、开发、测试等应用环境

三

存算分离

- 引入对象存储，降低存储成本
- 应用集群、大数据集群资源弹性伸缩

四

湖仓一体

- 统一开发管理工具，满足多源异构数据的存储和分析需求，统一数据访问接口，提升异构数据访问效率
- 丰富数据管理能力、提升数据质量，完备的工具支撑数据建模和交互式分析

- **存算分离架构**：跟随平台整体架构的演进路线，支撑更灵活的弹性部署，降低运维成本
- **数据湖分析**：利用 Doris 特性加速数据湖上的查询效率
- **管理工具进阶**

Thanks !

Apache Doris x 腾讯云联合 Meetur

Apache Doris x 腾讯云联合 Meetur

