

无锡锡商银行基于Apache Doris 实时数仓的探索与实践

武基鹏

无锡锡商银行 大数据研发中心技术经理

目录

1. 公司与个人介绍
2. 离线数仓的现状与痛点
3. 实时数仓的选型与架构
4. 实时数仓的收益与案例
5. Doris 2.0探索与未来规划

1 公司与个人介绍

无锡锡商银行股份有限公司



无锡锡商银行是全国第十九家、江苏省第二家民营银行。注册资本人民币20亿元，由红豆集团有限公司等江苏省内多家优秀民营企业共同发起设立，于2020年4月16日正式开业。

无锡锡商银行坚持恪守信用、稳健经营的理念，坚定科技普惠、错位发展的方向，牢记普惠金融、八方共赢的使命，以移动金融和物联网等技术为手段，以服务中小微企业、服务科技创新、服务城乡居民为己任，让产品易知、服务易得、实惠易享，打造具有物联网特色的科技型银行。

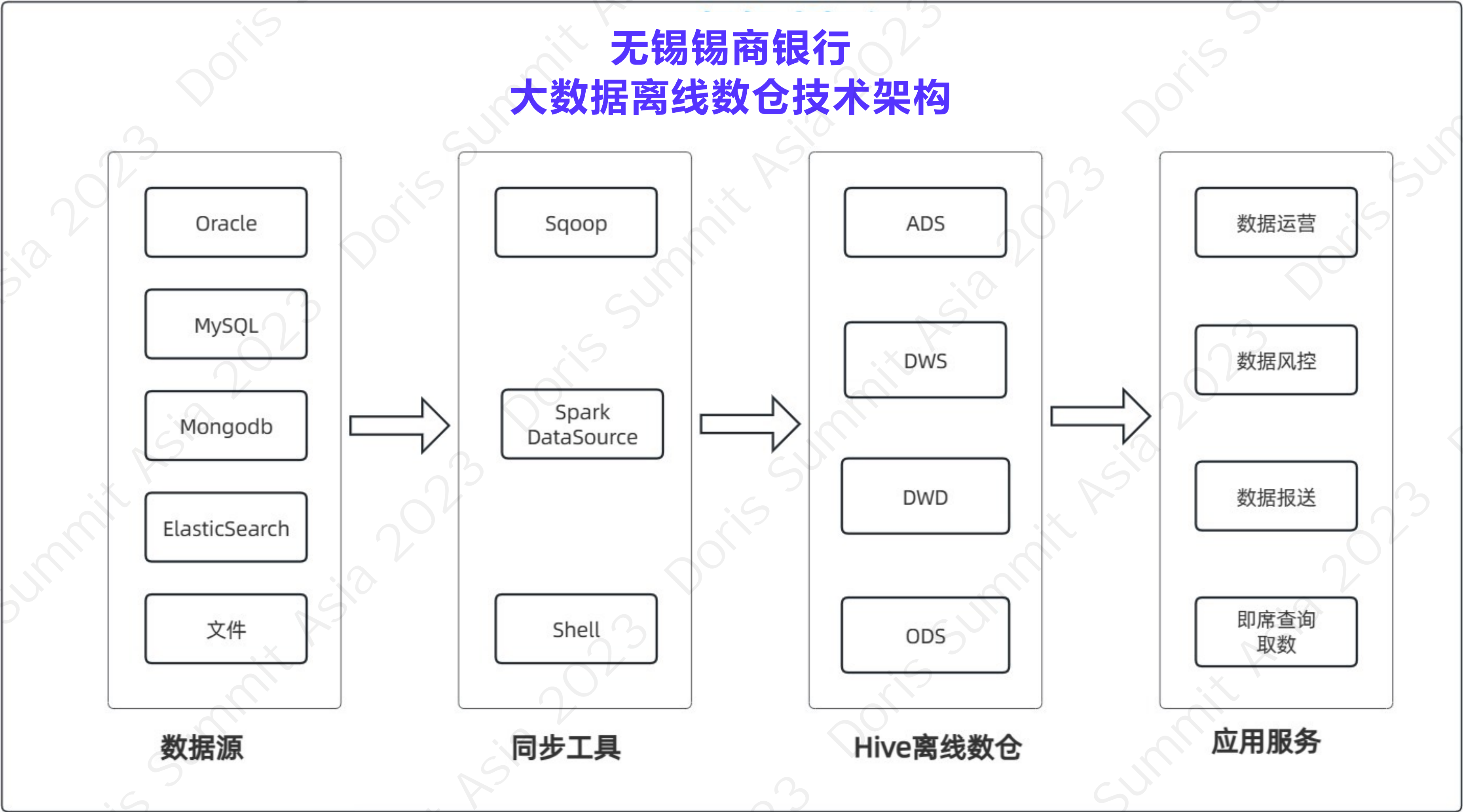
个人介绍

- 拥有10年+大数据相关经验；
- Apache HBase 技术社区委员会成员之一，同时也是第四届 HBase Meetup 演讲嘉宾；
- 现主要负责我行大数据平台的底层技术栈、离线&实时数仓的建设，保障其稳定性、安全性和高效性；
- 有较强的问题诊断和解决能力，能够快速处理遇到的问题和故障。

2 离线数仓的现状与痛点

离线仓库的现状

无锡锡商银行
大数据离线数仓技术架构



离线仓库的痛点

当前离线数据仓库架构对离线数据业务有很好的支撑，但是在面对实时场景，痛点也明显：

- ① 数据时效性 T+1，无当日最新实时数据；
- ② 数据查询效率不高，执行引擎主要是 Hive 及 Spark；
- ③ 底层技术栈复杂 (LDAP/Ranger/ZooKeeper/HDFS/YARN/Hive/Spark)，维护成本高；
- ④ 线上也有 HBase + Phoenix 实时存储与服务，其组件比较“重”、社区不活跃、其某些特性导致无法满足实时场景；

故目标是建设以 Apache Doris 为主体的实时数仓平台。

3 实时数仓的选型与架构

为什么会选择Doris

主要研究探索基于MPP数据库Apache Doris，构建实时数据仓库，赋能金融业务。

主要选择 Doris 理由：

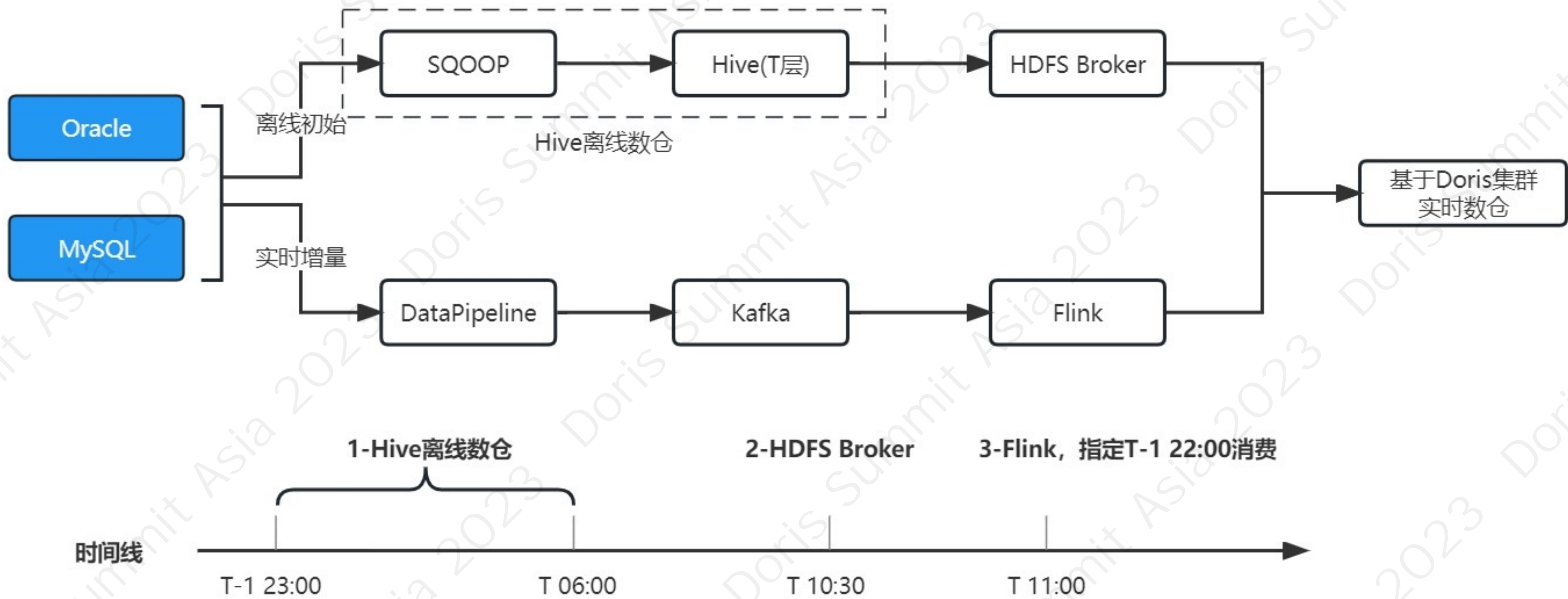
- ① 支持标准SQL语法；
- ② 支持秒级别的数据实时写入、更新、删除；
- ③ 轻量化表结构修改；
- ④ 数据写入小合并机制；
- ⑤ 单表及多表复杂查询，且高并发，能够在毫秒、秒级别返回结果；
- ⑥ 支持索引、物化视图等能力加速查询；

为什么会选择Doris

其中最主要研究内容：

- ① 基于 Oracle、MySQL 的实时表的表结构批量创建；
- ② 基于 Hive 离线数仓，使用 HDFS Broker 方式高效初始化历史数据；
- ③ 基于 Flink-Doris-Connector，构建实时数仓的数据链路，保证实时数据不丢、不重复；

实时仓库的技术架构



① 以HDFS Broker方式初始化表历史数据，使用Hive的T-1天历史数据；

② 以Flink-Doris-Connector方式实时解析T-1 22:00的Kafka数据到Doris集群，会存在部分数据重复，

由于Doris Uniq模型表支持幂等性，故不会出现丢失数据、重复数据。

实时仓库的技术架构考量

- ① 是否有必要建设所有库表；
- ② 为什么要选择基于离线数仓，做实时数仓的表历史数据初始化；
- ③ 而不选择直接业务库全量抽取再无缝切换为实时增量抽取；
- ④ 实时抽取 Oracle、MySQL 是否从主库或从库抽取；
- ⑤ 为什么需要构建 Kafka 层；
- ⑥ 实时链路是否会发生数据乱序，如业务产生是 insert、update1、update2、delete，而消费入库是 update2、insert、delete、update1；

4 实时数仓的收益与案例

实时数仓的收益



即席查询

取数
BI实时报表
当日数据分析



联邦查询

取数
历史数据分析



营销类数据服务

千人千面项目
资产净增活动
艺术家盲盒活动
AI智能客服



风控类数据服务

交易反欺诈
信贷反欺诈
关联历史订单
常规风控特征变量

实时数仓的收益

实时表及加工表，已接入

263_张

数据服务接口，已配置

149_个

平均SQL执行时效

1.5_s

数据服务接口，8月QPS

355_w

实时数仓的故障案例—初始化错列

Hive表结构

```
create table if not exists ods.verified (  
  ,base_acct_no      string  comment '账号'  
  ,client_no        string  comment '客户号'  
  ,document_id      string  comment '身份证号'  
  ,client_name      string  comment '客户名'  
  ,phone_no         string  comment '手机号'  
  ,tran_date        string  comment '交易时间'  
  ,verified_result   string  comment '验证情况'  
)  
comment '个人验证记录表'  
partitioned by(data_date string comment '数据日期')  
row format delimited fields terminated by '\t' lines terminated by '\n'  
stored as orcfile;
```

Doris表结构

```
create table if not exists ods.verified (  
  ,document_id      varchar(100) comment '身份证号'  
  ,base_acct_no     varchar(100) comment '账号'  
  ,client_no        varchar(100) comment '客户号'  
  ,client_name      varchar(100) comment '客户名'  
  ,phone_no         varchar(100) comment '手机号'  
  ,tran_date        datetime   comment '交易时间'  
  ,verified_result   varchar(100) comment '验证情况'  
)  
unique key (document_id)  
comment "个人验证记录表"  
distributed by hash(id) buckets 16  
PROPERTIES(  
  "light_schema_change" = "true" ,  
  "enable_unique_key_merge_on_write" = "true" );
```

实时数仓的故障案例—初始化错列

HDFS BROKER (导致列数据错位)

```
LOAD LABEL ods.verified_init2023091901
(
  DATA INFILE( "hdfs://...../tdata.db/verified/data_date=2023-09-19/*")
  INTO TABLE verified
  COLUMNS TERMINATED BY "\t"
  (document_id,base_acct_no,client_no,client_name,phone_no,tran_date,verified_result)
  SET(
    document_id=document_id,
    base_acct_no=base_acct_no,
    client_no=client_no,
    client_name=client_name,
    phone_no=phone_no,
    tran_date=tran_date,
    verified_result=verified_result
  )
)
WITH HDFS ("fs.defaultFS"="hdfs://..... ","hadoop.username"= "xxx")
PROPERTIES ("timeout"="3600","max_filter_ratio"="0");
```

- SET之前： 字段名称，是和doris一致，就会导致base_acct_no、 base_acct_no、 client_no 三个字段都错位，正确是hdfs表字段顺序；
- SET之内： 左侧是doris表字段， 右侧是hive表字段名称， 形成一一映射；

实时数仓的故障案例—同一天离线&实时投产丢数据



■ 丢数据：Flink是从16:00消费入库Doris，会丢失14-16点数据；

■ 丢数据：Flink是从18:00消费入库Doris，会丢失16-18点数据；

■ 不丢数据：需从14-16点消费入库Doris，就不会出现丢失数据；



实时数仓的场景案例—交易流水表保留7天数据

需求：

交易流水表数据量较大，下游实时查询效率较低，评估需求，需要提供一张只保留7天数据的表。

Doris表：

Uniq模型，id是主键，建表语句如下：

```
CREATE TABLE IF NOT EXISTS ods.tran_records(
  id          varchar(420)    COMMENT '序号',
  tran_date   datetime        COMMENT '交易日期',
  tran_type   varchar(240)    COMMENT '交易类型',
  tran_amount varchar(240)    COMMENT '交易金额'
)
UNIQUE KEY(id)
```


实时数仓的场景案例—交易流水表保留7天数据

方案一：SQL删除历史

- 通过计算tran_date和当前日期差 > 7来删除数据：

```
DELETE FROM  
ods.tran_records WHERE  
DATEDIFF(to_date(NOW()),  
to_date(tran_date)) > 7;
```

- 不对列名做函数计算，直接删除某一天数据：

```
DELETE FROM ods.tran_records  
WHERE tran_date = '2023-09-15  
00:00:00';
```



SQL 错误 [1105] [HY000]: errCode = 2, detailMessage = Left expr of binary predicate should be column name



SQL 错误 [1105] [HY000]: errCode = 2, detailMessage = errCode = 2, detailMessage = Column[tran_date] is not key column or storage model is not duplicate or column type is float or double.

实时数仓的场景案例—交易流水表保留7天数据

方案二：多主键（业务日期字段）

- 创建动态分区表自动删除超过7天的数据;
- 原业务表只有单主键 id，现在要根据 tran_date 进行分区则必须将 tran_date 创建为主键，那么该动态分区表的主键则为 id+tran_date;
- 存在 id 的数据进行 tran_date 跨天更新场景，则会有多条相同 id 不同 tran_date 的数据，不符合原业务场景。

```
CREATE TABLE IF NOT EXISTS ods.tran_records(  
  id          varchar(420)      COMMENT '序号',  
  tran_date   datetime         COMMENT '交易日期',  
  tran_type   varchar(240)     COMMENT '交易类型',  
  tran_amount varchar(240)     COMMENT '交易金额'  
)  
  
  UNIQUE KEY(id,tran_date)  
  COMMENT '交易流水表最近7天数据'  
  PARTITION BY RANGE(tran_date)()  
  distributed by hash(seq_no) buckets 64  
  PROPERTIES  
  (  
    "light_schema_change" = "true",  
    "enable_unique_key_merge_on_write" = "true",  
    "dynamic_partition.enable" = "true",  
    "dynamic_partition.time_unit" = "DAY",  
    "dynamic_partition.start" = "-7",  
    "dynamic_partition.end" = "3",  
    "dynamic_partition.prefix" = "p",  
    "dynamic_partition.buckets" = "32",  
    "dynamic_partition.replication_num" = "3"  
  );
```

实时数仓的场景案例—交易流水表保留7天数据

方案三：多主键（系统日期字段）

- 创建动态分区表自动删除超过 7 天的数据;
- id 的数据进行 tran_date 跨天更新，回表查询出 insert 操作时 partition_date 对应的值，拼接 update 的 json 入库。

```
CREATE TABLE IF NOT EXISTS ods.tran_records(  
  partition_date  date NULL COMMENT '分区日期',  
  id              varchar(420)    COMMENT '序号',  
  tran_date       datetime        COMMENT '交易日期',  
  tran_type       varchar(240)    COMMENT '交易类型',  
  tran_amount     varchar(240)    COMMENT '交易金额'  
)  
  
UNIQUE KEY(partition_date,id)  
COMMENT '交易流水表最近7天数据'  
PARTITION BY RANGE(partition_date)()  
distributed by hash(seq_no) buckets 64  
PROPERTIES  
(  
  "light_schema_change" = "true",  
  "enable_unique_key_merge_on_write" = "true",  
  "dynamic_partition.enable" = "true",  
  "dynamic_partition.time_unit" = "DAY",  
  "dynamic_partition.start" = "-7",  
  "dynamic_partition.end" = "3",  
  "dynamic_partition.prefix" = "p",  
  "dynamic_partition.buckets" = "32",  
  "dynamic_partition.replication_num" = "3"  
)
```


5 Doris 2.0 探索与未来规划

Doris2.0探索-点查

目前我行有 2 套 HBase 集群，主要是提供点查服务，支持营销及风控场景。

在使用过程中，会出现一些问题：

- ① Master/Regionserver 节点异常退出；
- ② Region 进入 RIT 状态；

```
@Override  
protected void handleException(Throwable t) {  
    LOG.warn("Fatal error occurred while closing region {}, aborting."  
    getServer().abort(reason: "Failed to close region " + encodedName +  
    }  
}
```

Doris2.0探索-点查QPS

开启高并发点查: "store_row_column" = "true "

- 服务器3节点, 配置: 8C
- FE内存: 10G
- BE内存: 10G

5千万数据单表查询

2.5万

5千万数据多表读写

2万

大SQL查询

2.5万

多表实时读写

2.5万

实时数仓-未来使用规划

高并发点查服务

计划性迁移点查服务，下线 hbase 集群

自动分桶

线上版本 1.2.1，目前是手动计算

Pipeline执行引擎

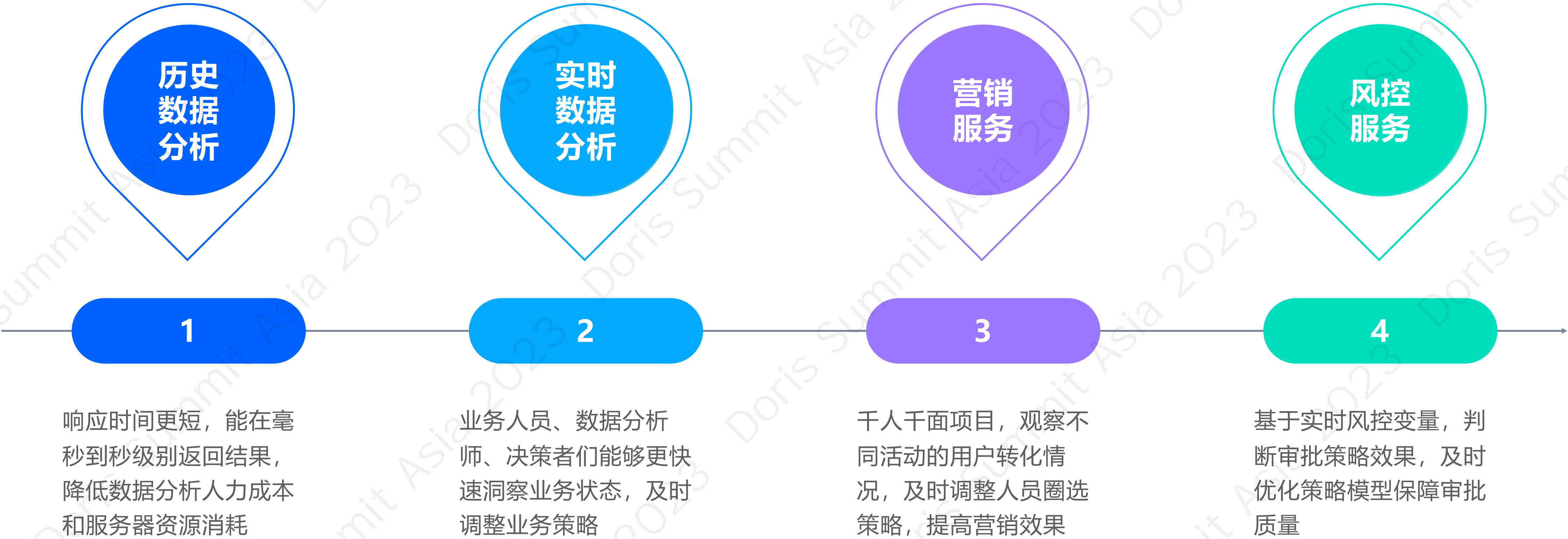
当前是基于火山模型设计，无法充分利用多核计算能力

双集群模式

主要是实时数仓，支撑各种场景，对高可用性要求较高，尤其数据接口服务



实时数仓-未来更好的服务业务





获取更多社区动态与最佳实践

Apache Doris 官方平台:

- Apache Doris 官网: doris.apache.org
- Apache Doris GitHub: github.com/apache/doris/

获取更多峰会资料:

- Doris Summit 峰会官网: doris-summit.org.cn
- Doris Summit 峰会回放: <https://space.bilibili.com/1196172099/channel/collectiondetail?sid=1824324>