

Apache Doris 在知乎 AB 实验平台的应用实践

张潇鹤 数据平台开发工程师

目录

01 知乎 AB 平台业务背景介绍

02 知乎 AB 实验平台架构演进历程

03 知乎 AB 在 Apache Doris 上的实践

04 未来展望

01

知乎 AB 实验平台业务背景介绍

知乎 - AB 实验平台介绍

- 知乎

高质量的在线问答社区

- AB 实验平台

AB 实验的主要目的在于降低风险和分析策略结果。其基本思想是从大盘中取出一小部分流量，随机地将用户分给对照组和实验组，通过收集、分析不同分组用户行为指标数据，再结合统计学方法得出实验结论。



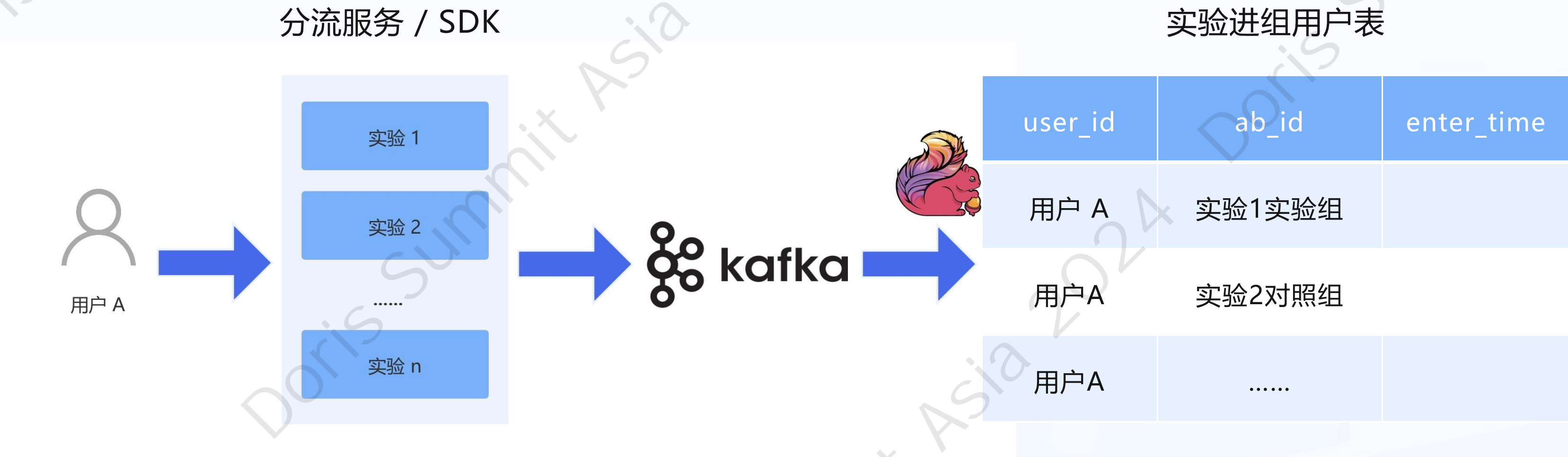
知乎 - AB 平台的基本背景介绍

- ◆ 知乎 AB 实验平台支撑知乎主站、盐言故事、知乎知学堂等多条业务线，每天平台运行上千个实验。
- ◆ 实验进组用户表日均数据量百亿级。
- ◆ 支持实验分析场景多样：支持基础计算类、留存类、LTN 类等 4000 多个指标分析，离群值剔除、多维度下钻等。



知乎 - AB平台进组用户生成逻辑介绍

每天的实验进组用户数据量是 DAU 用户的数倍，一个用户携带的实验标签数量级介于数十 ~ 上百个不等。

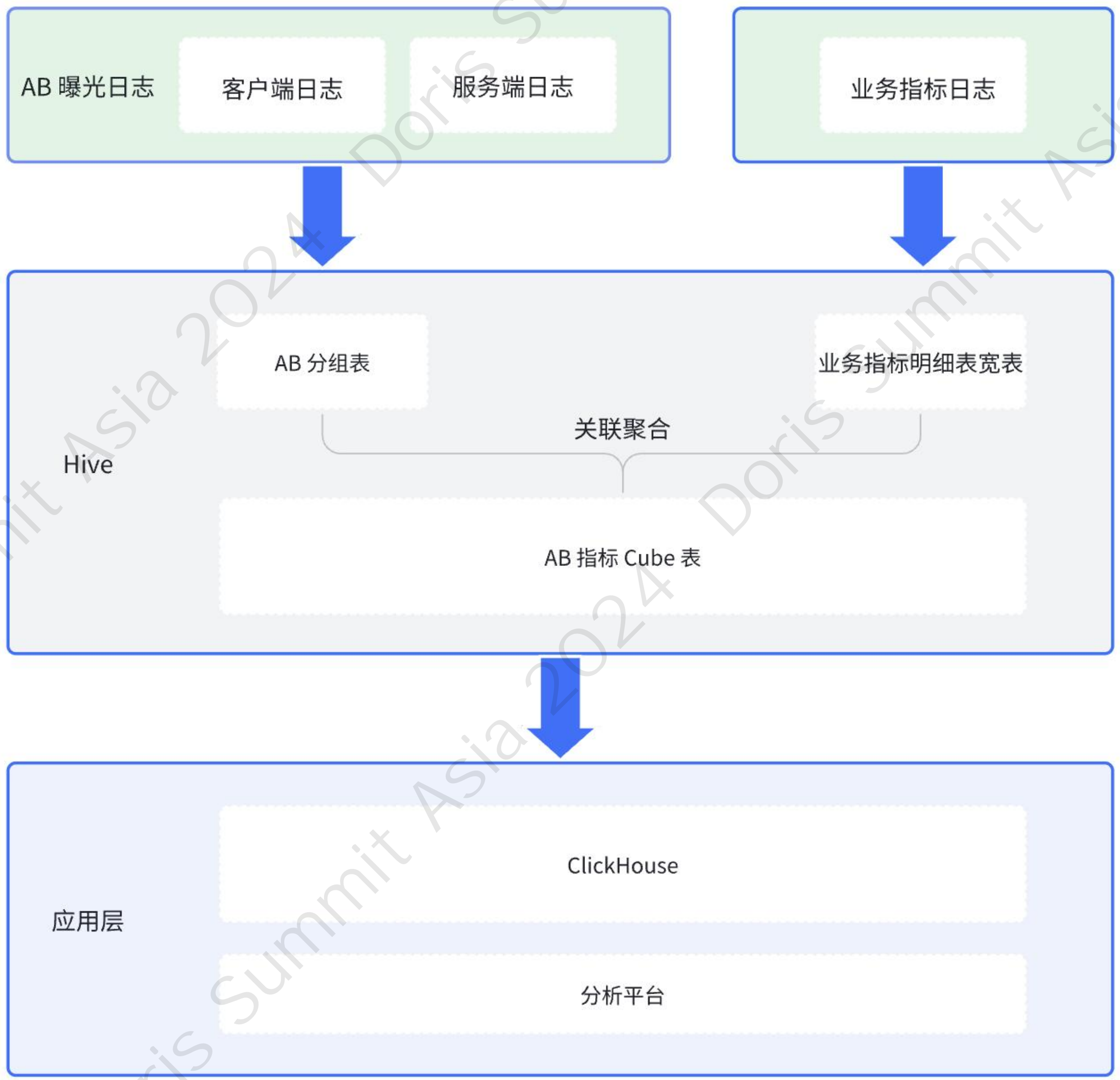
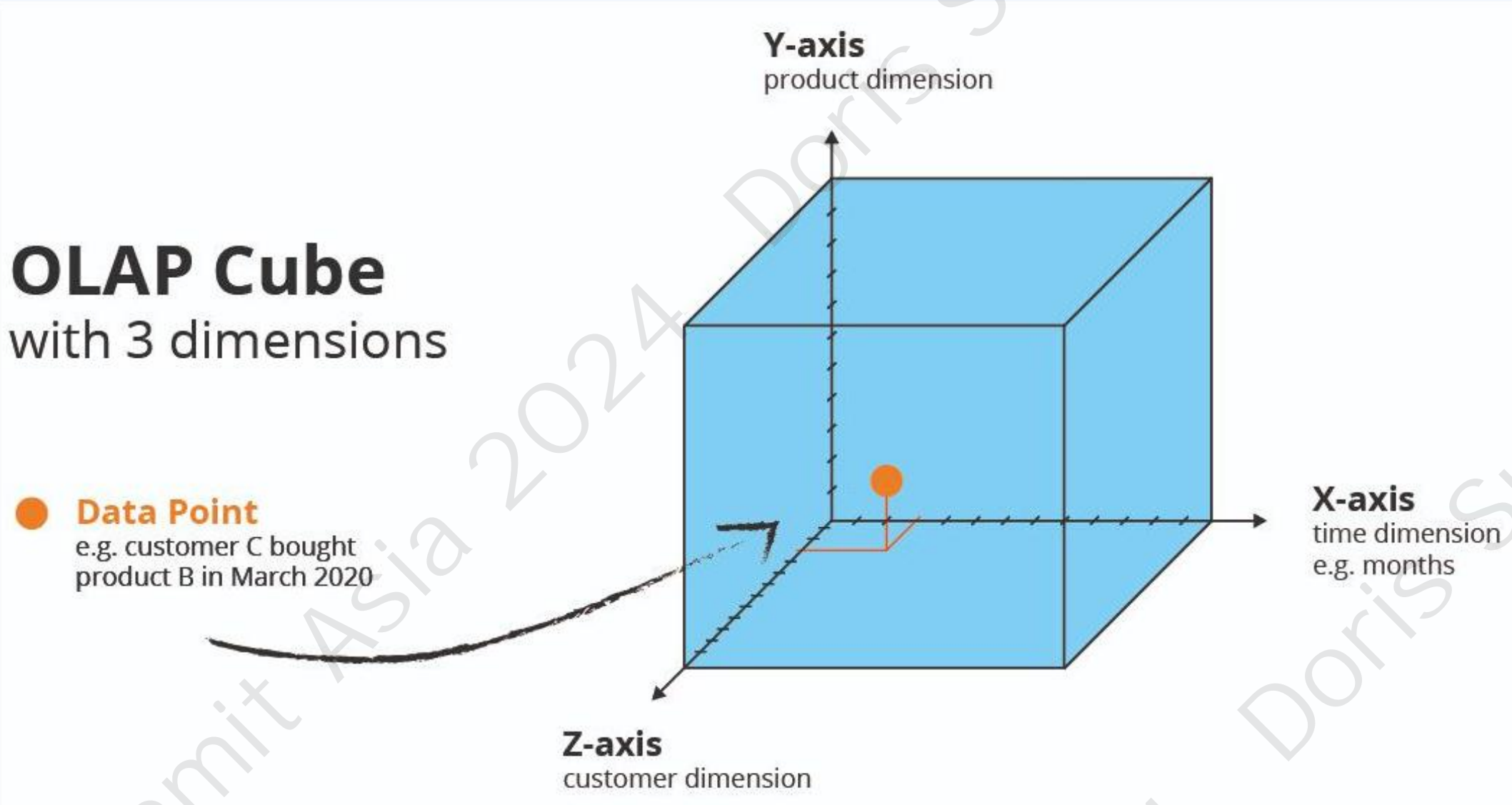


02

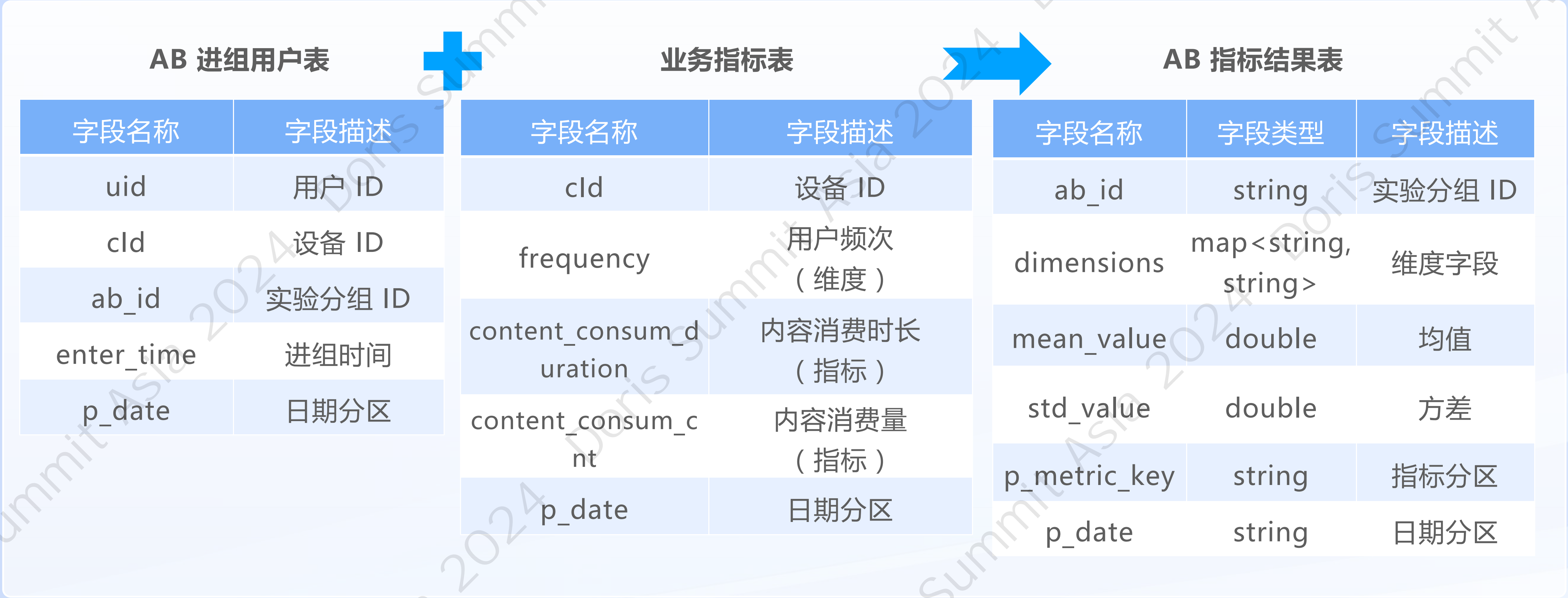
知乎 AB 实验平台 架构演进历程

知乎 - AB2.0 基于Clickhouse的平台架构

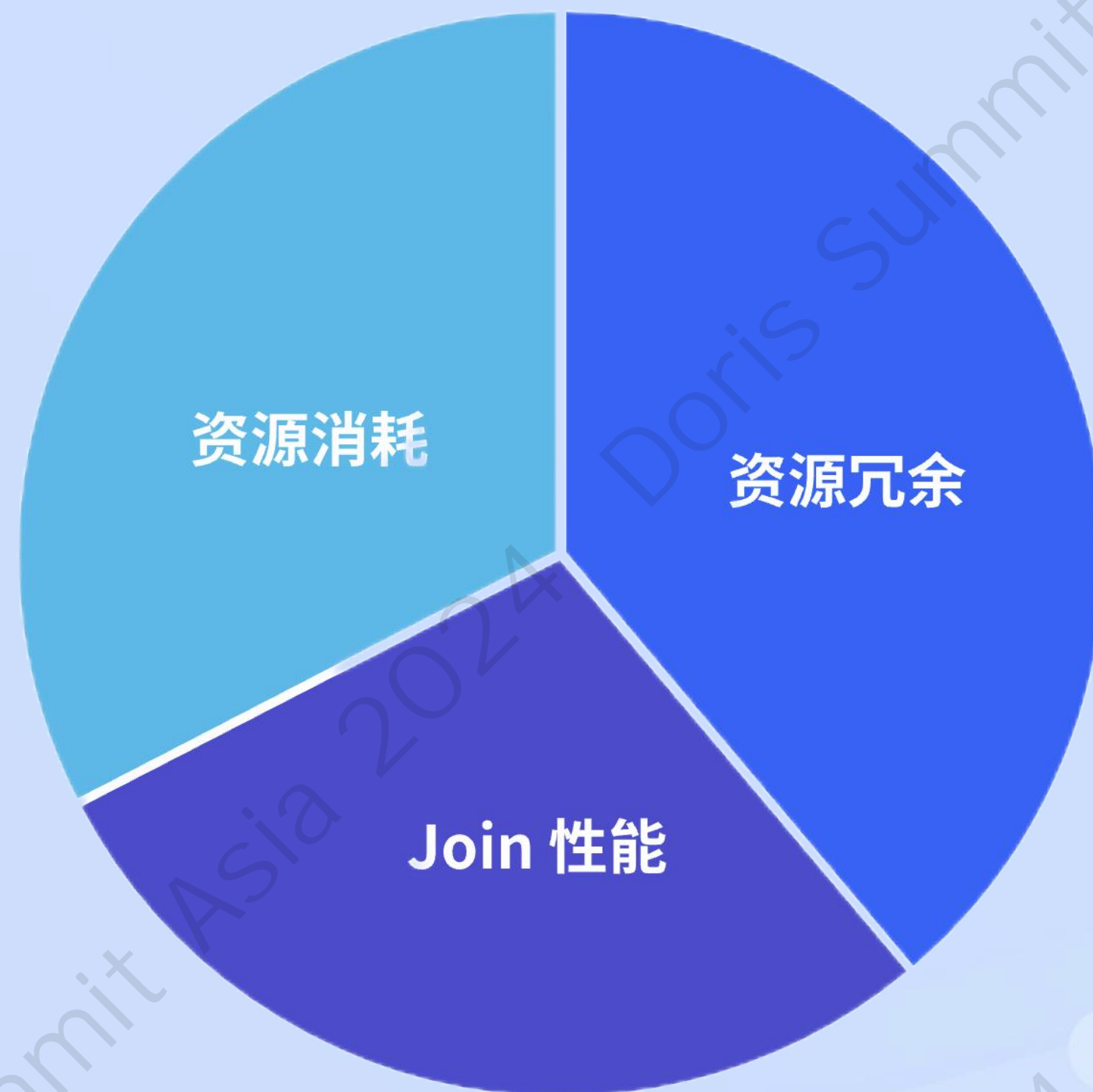
- ◆ AB 2.0 平台采用了预计算的方式，最终将预计算结果写入到 ClickHouse 中，充分利用了 ClickHouse 的单表查询能力。



知乎 - AB 2.0 平台实验数据加工流程



知乎 - AB 2.0 架构的核心痛点



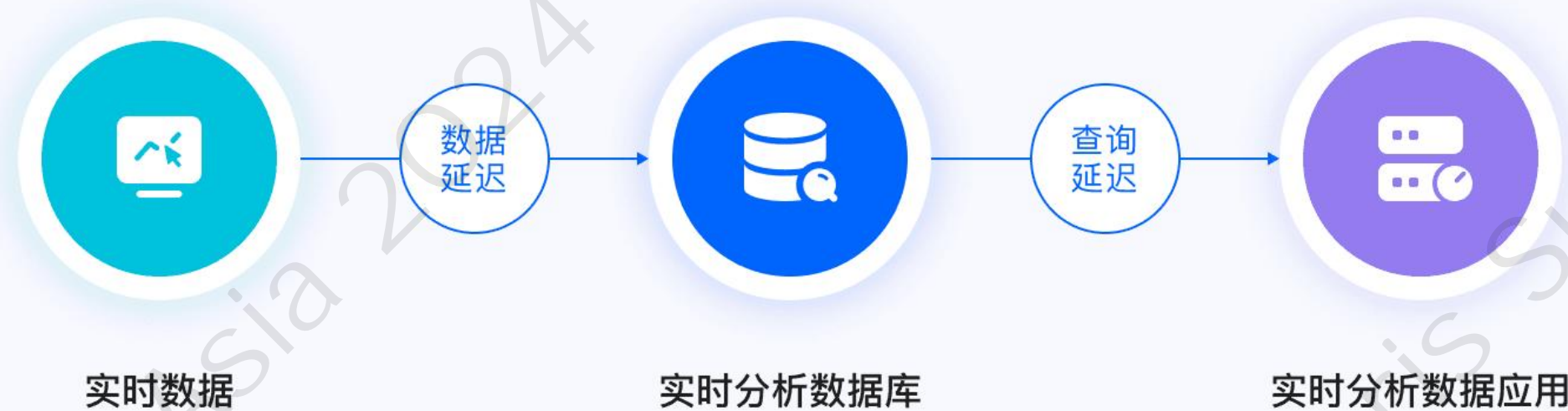
老架构的三大痛点

- ◆ **痛点1**：预计资源消耗大，影响集群其他任务，只能对做 Cube 剪枝
- ◆ **痛点2**：ClickHouse 对多表或大表 Join 支持有限，很多业务场景无法满足
- ◆ **痛点3**：已有指标数据无法复用，造成资源重复消耗

知乎 - 新架构选型的二大核心目标

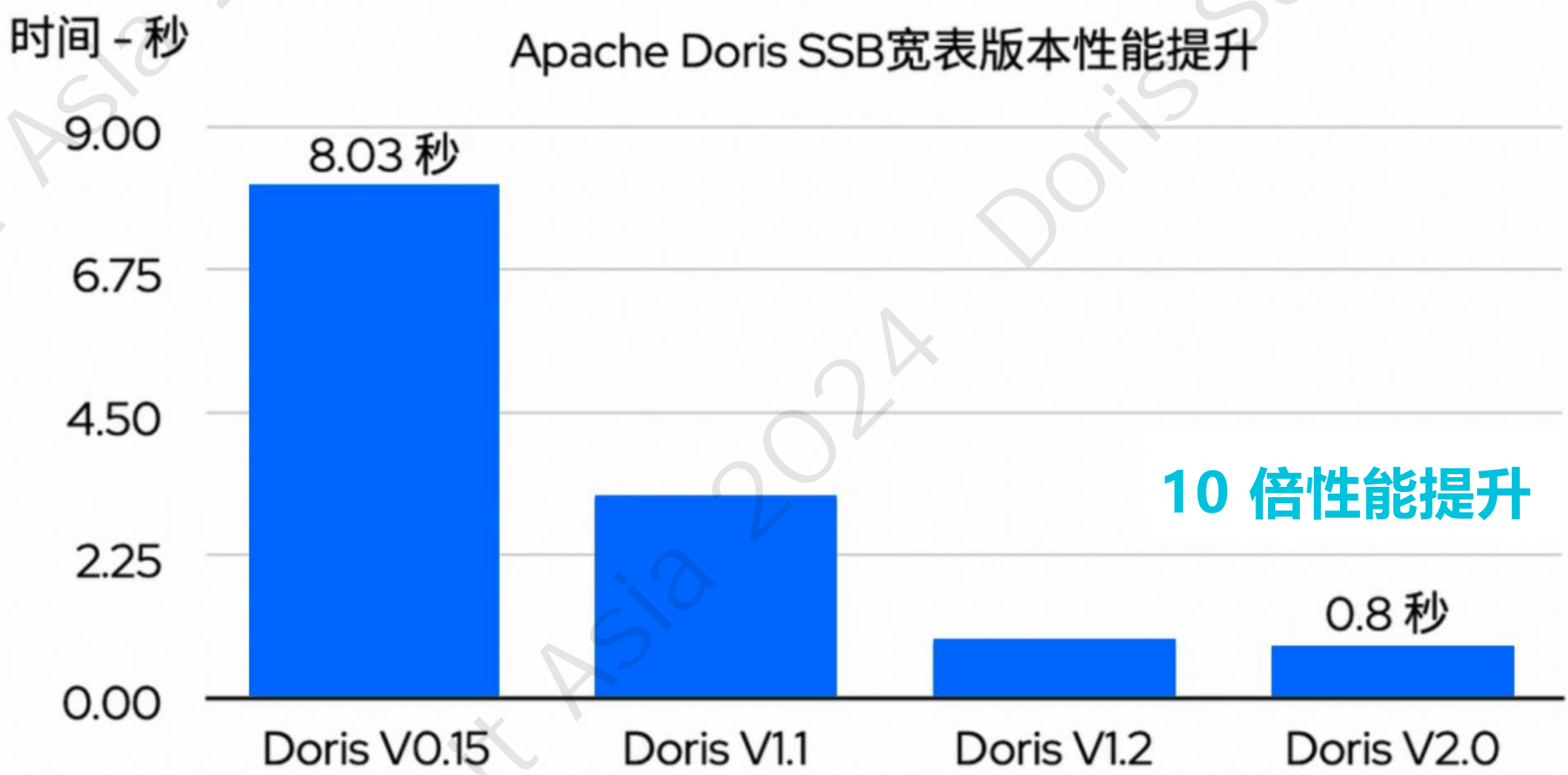
◆ 新的选型产品在海量数据同步和多表关联查询性能两个场景下同时满足业务诉求

数据同步



- 需要支持每日百亿级数据量写入，支持事务导入，数据精准写入。

查询性能

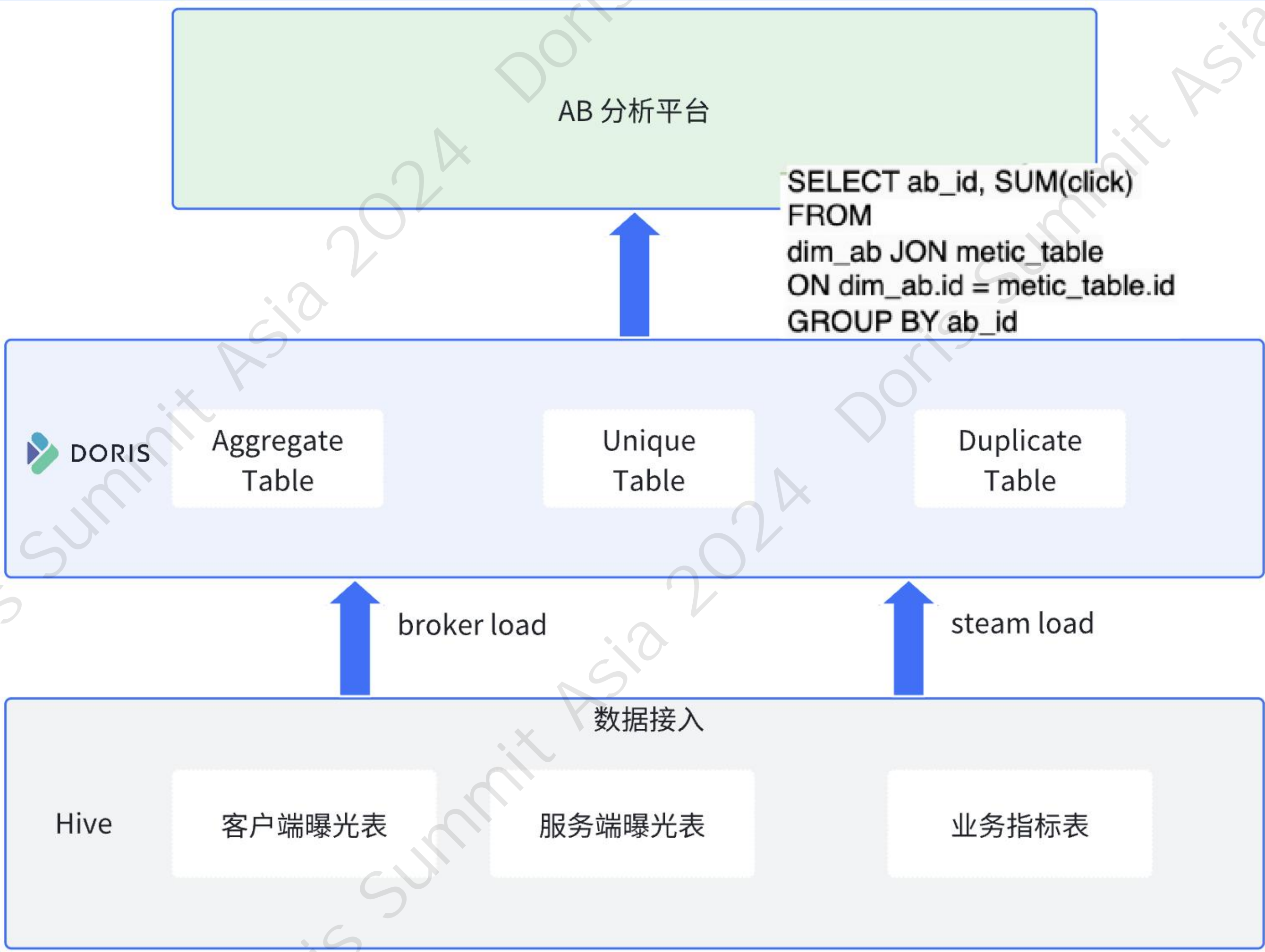


- 新的 OLAP 引擎需要满足 AB 进组用户表和业务指标表进行即席关联查询，部分业务场景需要 3-4 张表的 Join。

知乎 - 基于 Apache Doris 的 AB 数据流

切换到 Doris 后数据流程清晰简单：

- Hive 中只做 AB 进组用户的落表和指标数据的加工，不再与 AB 业务耦合
- 通过 Broker Load / Steam Load 进行 Doris 的数据写入
- AB 平台直接对接 Doris 进行即席数据查询



知乎 - 新旧 AB 平台功能与收益对比

- Hadoop 资源消耗大幅降低
- Doris 在多个大表 Join 下性能表现优秀，新版 AB 实现了相关性分析、归因分析、指标离群值剔除等实验功能
- 业务可根据已有数据进行指标口径自定义，缩短数据加工链路

对比维度	基于 ClickHouse 的旧架构	基于 Doris 的新平台架构
维度下钻能力支持	每个实验最多支持 3 个维度	不限制维度下钻个数，支持任意维度组合
指标离群值剔除	需要手动耦合到 Hive 作业中，很难支持	算法计算产出指标每天的阈值，自动生效
指标归因分析	不支持	支持指标下钻拆解，快速帮助业务定位变动原因
留存分析	根据留存范围预先计算	支持灵活配置，一张数据表可计算出次日/三日/七日留存数据
指标引入	完全依赖数据开发进行加工	支持基于现有指标，在 Doris 进行二次加工

04

知乎 AB 在 Doris 上的实践

知乎 - 表结构设计基本准则

```
CREATE TABLE `table_test` (  
  `is_valid_svip` boolean NULL COMMENT "是否有效会员",  
  `is_new_vip` int(11) NULL COMMENT "是否是新会员「1新会员0老会员」",  
  `platform` varchar(36) NULL COMMENT "平台",  
  `device_brand` varchar(256) NULL COMMENT "设备品牌",  
  `member_id` bigint(20) COMMENT '用户 id',  
  `frequency_label` varchar(32) NULL COMMENT "频次",  
  `p_date` date NOT NULL COMMENT "yyyy-MM-dd") ENGINE=OLAP  
DUPLICATE KEY(`is_valid_svip`, `is_new_vip`, `platform`)  
COMMENT "ab用户路由指标表 (最近N天)"  
PARTITION BY RANGE(`p_date`)(  
DISTRIBUTED BY HASH(`member_id`) BUCKETS 10  
PROPERTIES (  
  "colocate_with" = "ab_member",  
  "replication_allocation" = "tag.location.default: 3",  
  "dynamic_partition.enable" = "true",  
  "dynamic_partition.time_unit" = "DAY",  
  "dynamic_partition.time_zone" = "Asia/Shanghai",  
  "dynamic_partition.start" = "-90",  
  "dynamic_partition.end" = "3",  
  "dynamic_partition.prefix" = "p",  
  "dynamic_partition.replication_allocation" = "tag.location.default: 3",  
  "dynamic_partition.buckets" = "10",  
  "dynamic_partition.create_history_partition" = "true",  
  "dynamic_partition.history_partition_num" = "90",  
  "dynamic_partition.hot_partition_num" = "0",  
  "dynamic_partition.reserved_history_periods" = "NULL",  
  "in_memory" = "false",  
  "storage_format" = "V2",  
  "compression" = "ZSTD"  
);
```

- 利用好 Doris 默认提供的前缀索引
- zstd 压缩方式
- 合理的 bucket 数量
- 指定 Group , 查询优先命中 Colocate Join
- 排序列

知乎 - 进组用户表设计

- 使用业务字段 exp_name 进行物理分区
- Spark 直接按照分区生成对应 parquet 文件，进行 broker load
- 使用 AGG 模型，部分列更新，提升查询和写入效率

```
CREATE TABLE `dim_ab_id_member` (  
  `member_id` varchar(32) NOT NULL COMMENT "用户member_id",  
  `exp_name` varchar(32) NOT NULL COMMENT "实验key",  
  `enter_date` date MIN NULL DEFAULT "2300-01-01" COMMENT "用户初次进组的日期",  
  `bucket` int(11) REPLACE NOT NULL COMMENT "bucket值",  
  `ab_id` varchar(32) REPLACE NULL DEFAULT "" COMMENT "实验ID: 实验key+组",  
  `active_date` date MAX NULL DEFAULT "2300-01-01" COMMENT "用户最新激活实验的日期",  
  `active_date_bitmap` bitmap BITMAP_UNION NULL COMMENT "用户进组明细"  
) ENGINE=OLAP  
AGGREGATE KEY(`member_id`, `exp_name`)  
COMMENT "abclient曝光表"  
PARTITION BY LIST(`exp_name`)  
DISTRIBUTED BY HASH(`member_id`) BUCKETS 30  
PROPERTIES (  
  "replication_allocation" = "tag.location.default: 3",  
  "colocate_with" = "ab_30_member",  
  "in_memory" = "false",  
  "storage_format" = "V2",  
  "compression" = "ZSTD"  
);
```

知乎 - 精准进组用户分析 (bitmap)

在精准分析业务中，会对用户进组/出组时间的滑动窗口查询，有两种实现方式：

- 记录每日明细数据，通过明细数据进行过滤
- 将用户的进组日期写到 Bitmap 里，通过 bitmap_and_count 函数与筛选日期做交叉查询

```
mysql> select exp_name,ab_id, bitmap_to_string(active_date_bitmap) from
+-----+-----+-----+
| exp_name | ab_id | bitmap_to_string('active_date_bitmap') |
+-----+-----+-----+
|          |      | 20241122,20241123,20241124,20241126 |
+-----+-----+-----+
```

```
SELECT ab_id, count(1)
FROM doris_table_test
WHERE
exp_name='xxx'
AND
bitmap_and_count(bitmap_from_string("20241121,20241122,20241123,20241124,20241125,20241126,20241127"),
active_date_bitmap)>0
GROUP BY ab_id;
```


知乎 - 指标查询逻辑

- 指标拆解，按需查询
- 归属同一张表的指标查询 SQL 合并
- 缓存加速

推荐页千展负反馈数_M P2 实验指标 ?

详情

变更记录

业务信息

类型: 计算指标 ?

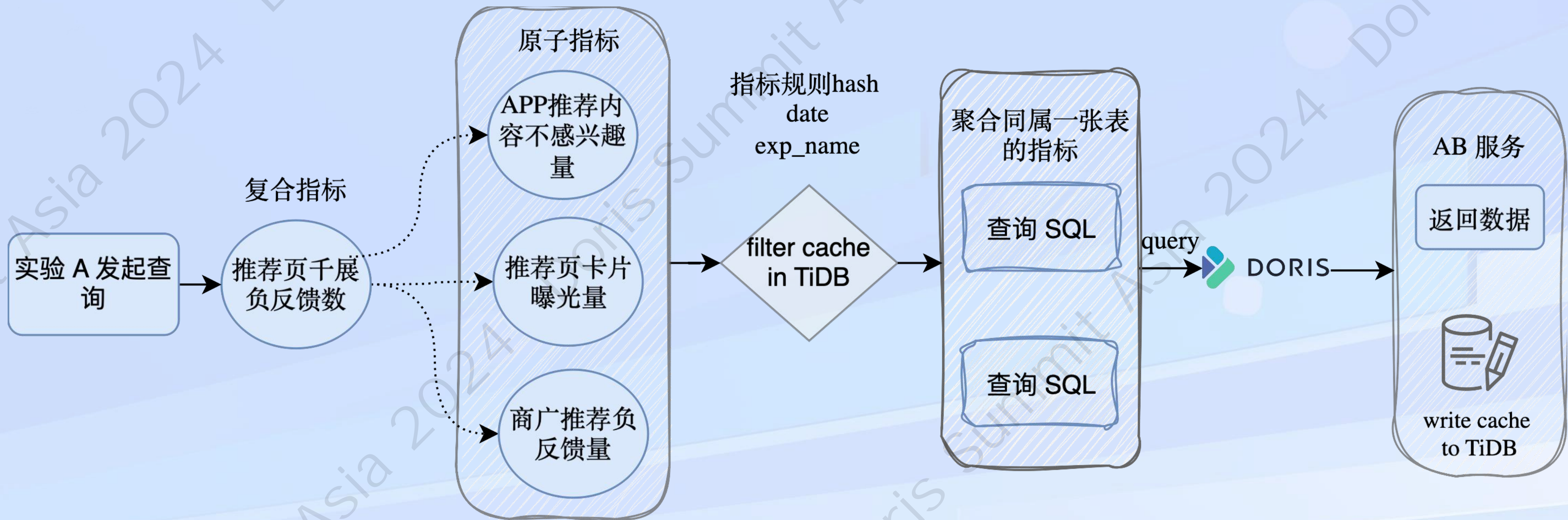
指标名称: 推荐页千展负反馈数_M

口径: 推荐页内容千次曝光带来的负反馈量

关联维度: --

别名: --

计算公式: $(\text{APP端推荐场景内容不感兴趣量}_M + \text{商广推荐负反馈量}_M) * 1000 / \text{推荐页APP端卡片曝光量}_M$



知乎 - 数据导入优化

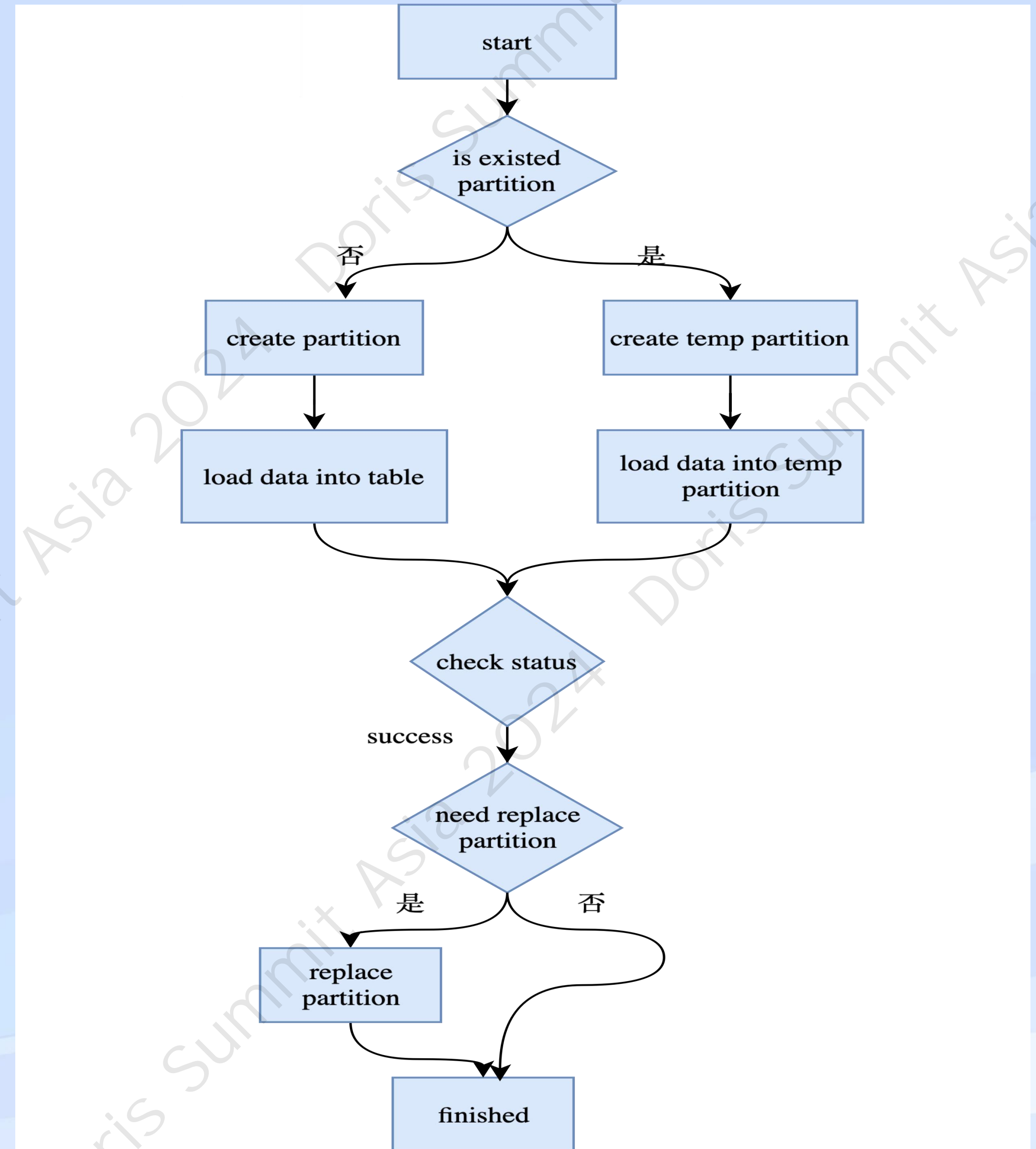
- 参数调整：

desired_max_waiting_jobs=200

async_pending_load_task_pool_size=15

async_loading_load_task_pool_size=15

- 通过临时分区实现数据的原子导入



知乎 - Join调优

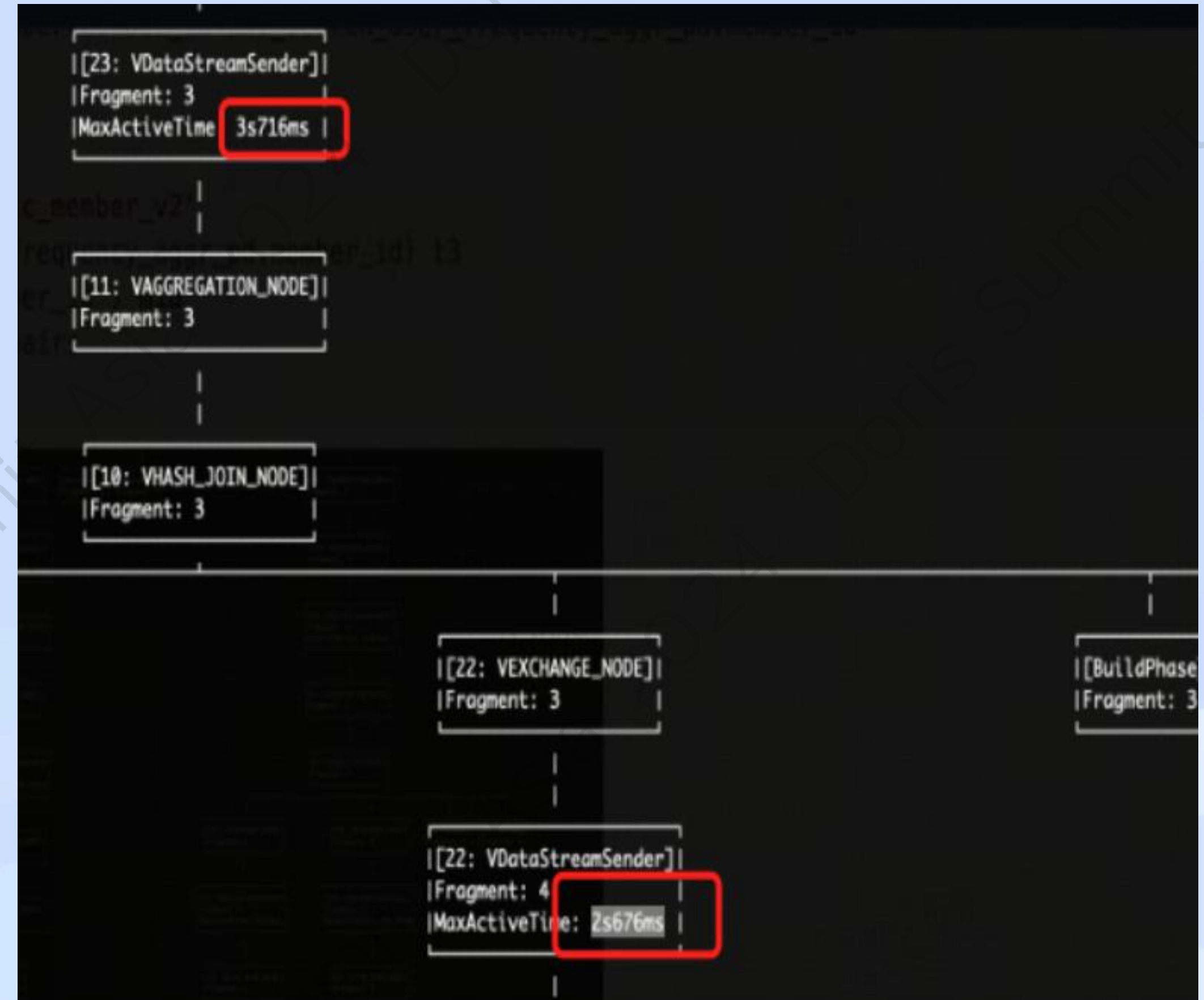
- 当不确定两张表的大小时，可以使用 sql hint，让 Doris 自己去决定采用 Join Reorder set
`enable_cost_based_join_reorder = true`
- 在一些情况下，Shuffle Join 的方式代价更低，可以显示指定 shuffle 方式，让 Doris 强制进行 Shuffle Join：

```
SELECT a.* FROM a
INNER JOIN [shuffle] b
on a.user_id = b.user_id
```

Shuffle方式	网络开销	物理算子	条件限制
Broadcast	$N * T(R)$	Hash Join/Nest Loop Join	无，并且 Broadcast 是将最小表广播到大表的各个节点，当小表数据量大的超过 BE 设置的内存 limit 时，Join 会失败。
Shuffle	$T(S) + T(R)$	Hash Join	无
Bucket Shuffle	$T(R)$	Hash Join	Join 条件中存在左表的分布式列，且左表执行时为单分区。
Colocate	0	Hash Join	Join 条件中存在左表的分布式列，且左表同属于一个 Clocate Group。

知乎 - Profiling

```
SELECT ab_id,  
       SUM(ltn) AS ln,  
       COUNT(1) AS ld  
FROM  
  (SELECT t1.ab_id,  
         t1.member_id,  
         t3.ltn  
   FROM  
     (SELECT ab_table.*  
      FROM  
        (SELECT ab_id,  
               ab_table.member_id  
         FROM table_a  
         WHERE exp_name='metric_member_v2') ab_table ) t1 INNER JOIN[shuffle]  
        (SELECT member_id  
         FROM table_b  
         WHERE p_date='2024-05-03'  
         GROUP BY member_id) t2  
        ON t1.member_id=t2.member_id INNER JOIN[shuffle]  
        (SELECT table_b.member_id,  
               COUNT(DISTINCT p_date) AS ltn  
         FROM table_b INNER JOIN[shuffle] table_a  
         ON table_a.member_id=table_b.member_id  
         WHERE p_date  
           BETWEEN '2024-05-03'  
           AND '2024-05-09'  
           AND exp_name='metric_member_v2'  
         GROUP BY table_b.member_id) t3  
        ON t1.member_id=t3.member_id ) mid  
   GROUP BY ab_id
```



知乎 - Apache Doris 调优后收益

- 1. 查询耗时降低：每周查询总耗时，在统计执行的 SQL 数量略有增加的情况下，SQL 执行总耗时降低了 **35%**。
- 2. 查询性能提升：Doris 查询 P99 的用时，由 **8s+** 降低到了 **3s+**，降幅 **60%**。

before_sql_count (bigint)	before_cost_secor (double)
1105029	1871010.4
after_sql_count (bigint)	after_cost_second (double)
1144276	1216660.54



04

未来展望

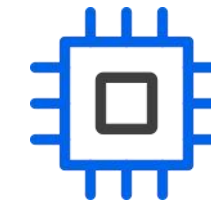
未来展望



数据冷热存储



更多字段支持



Local Shuffle 优化

Thanks for Watching !