

Apache Doris在美团的统一 OLAP引擎实践

曾林西

美团 - 高级技术专家

曾林西 美团查询引擎团队负责人

- ▶ 14年加入美团数据组，深度参与美团Hadoop集群规模从百到十万级的架构演进
- ▶ 主要负责离线数仓生产、Adhoc查询、OLAP分析引擎与服务在美团业务场景的落地与演进

目录

1. 美团OLAP场景特点
2. 美团OLAP引擎选型与实践
3. 当前挑战与未来规划

1 美团OLAP场景特点

美团 OLAP 场景特点

时延敏感

大部分数据报表响应时延要求在3秒内，部分toB场景要求亚秒级响应

数据量大

千亿级别数据分析，传统RDBMS/MPP/SQL on Hadoop方案搞不定

业务丰富

交易、经营分析、用户、流量、广告、金融、财务、LBS、性能监控

	PM / 运营	BD / 骑手	B 端商户
可用性	99.9%	99.9%	99.99%
QPS	10~100	100~1000	1K~10K
TP99响应	< 3s	< 1s	< 500ms

业务场景	特点及挑战
经营分析	数百个分析指标、复杂关联查询
广告商家报表	日千万行，点查询，高QPS
应用性能分析	高写入吞吐和时效性、海量数据聚合

OLAP 场景举例：外卖经营分析

查看各组织节点下所有商户的上百项经营指标

特点：

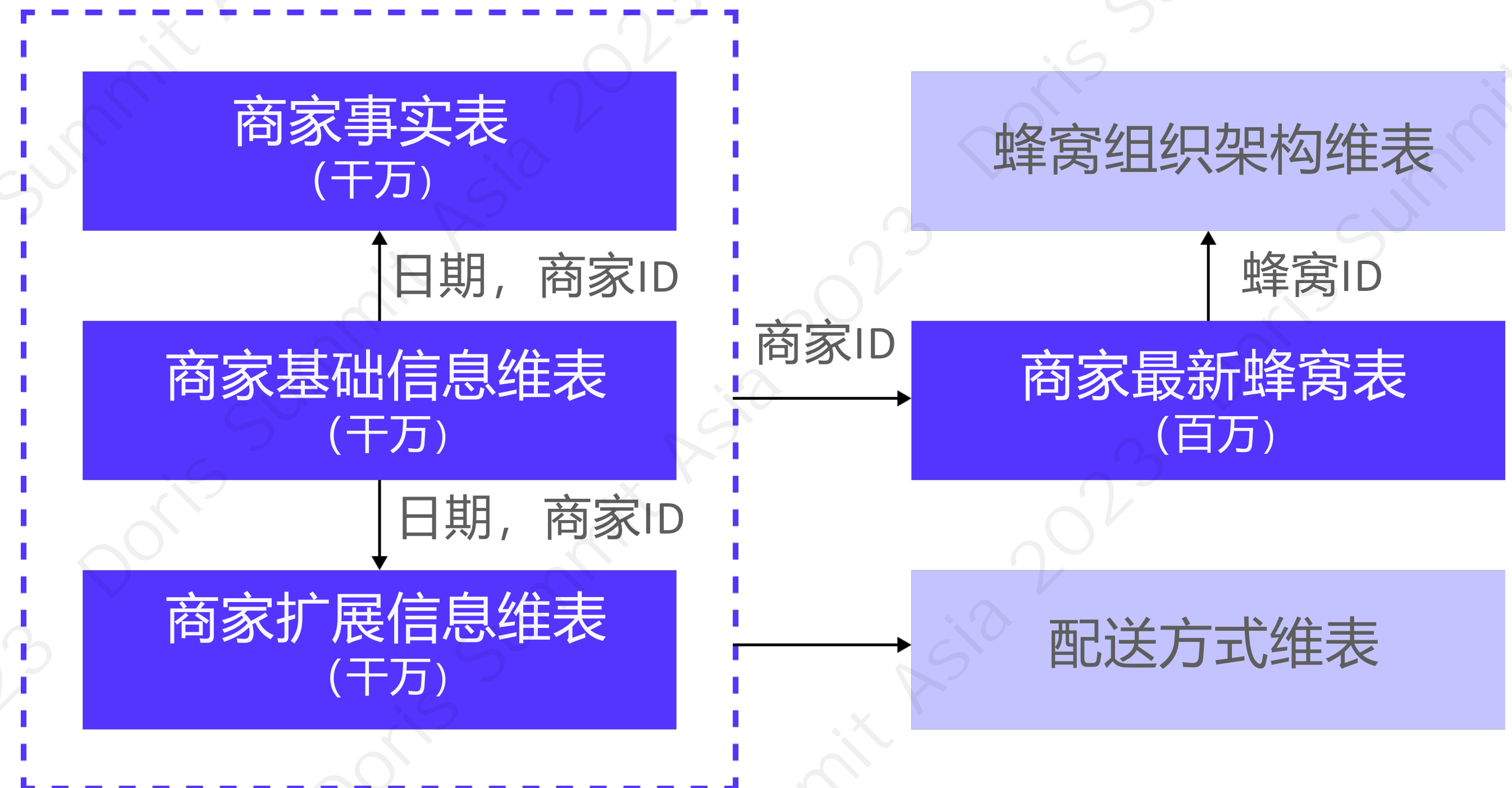
数据量大、变化维、复杂业务逻辑

技术难点：

多张亿级别大表现场实时关联聚合

关键技术：

MPP、列存储、ColocatedJoin



OLAP 场景举例：到店餐饮 BD 人效分析

查看各组织节点下BD的业绩指标

特点	技术难点	关键技术
<ul style="list-style-type: none">模型复杂：100个维度，60个SUM指标，20个去重指标单天百万行，单次查询需要分析半年数据	支持高效的 count distinct	预聚合 Bitmap去重指标、数据自动上卷到月粒度

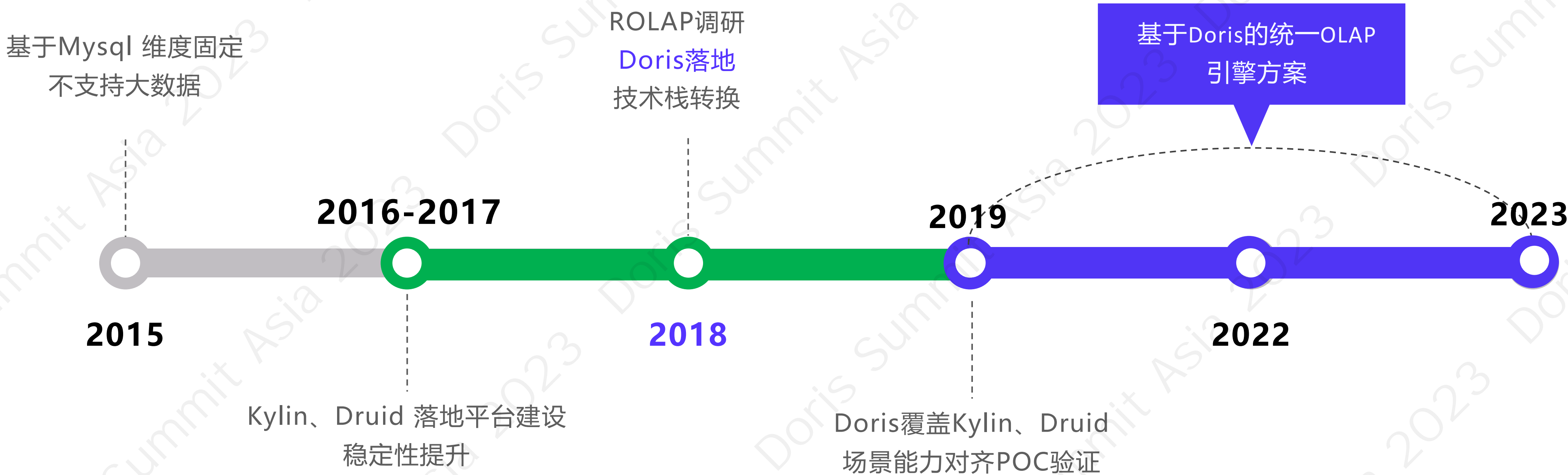
OLAP 场景举例：B 端商户报表

商家查看旗下门店的经营状况(广告投放效果/营业额/菜品销售量等)

特点	技术难点	使用成本高
数据总量大 点查询为主 时间跨度大	要求高并发 低延迟 4个9可用性	前缀索引 数据自动上卷到月粒度

2 美团 OLAP 引擎选型与实践

美团OLAP引擎选型与演进历程



先分治，再统一

先解决“时延敏感”和“数据量大”，再解决场景丰富

统一 OLAP 引擎：问题和挑战

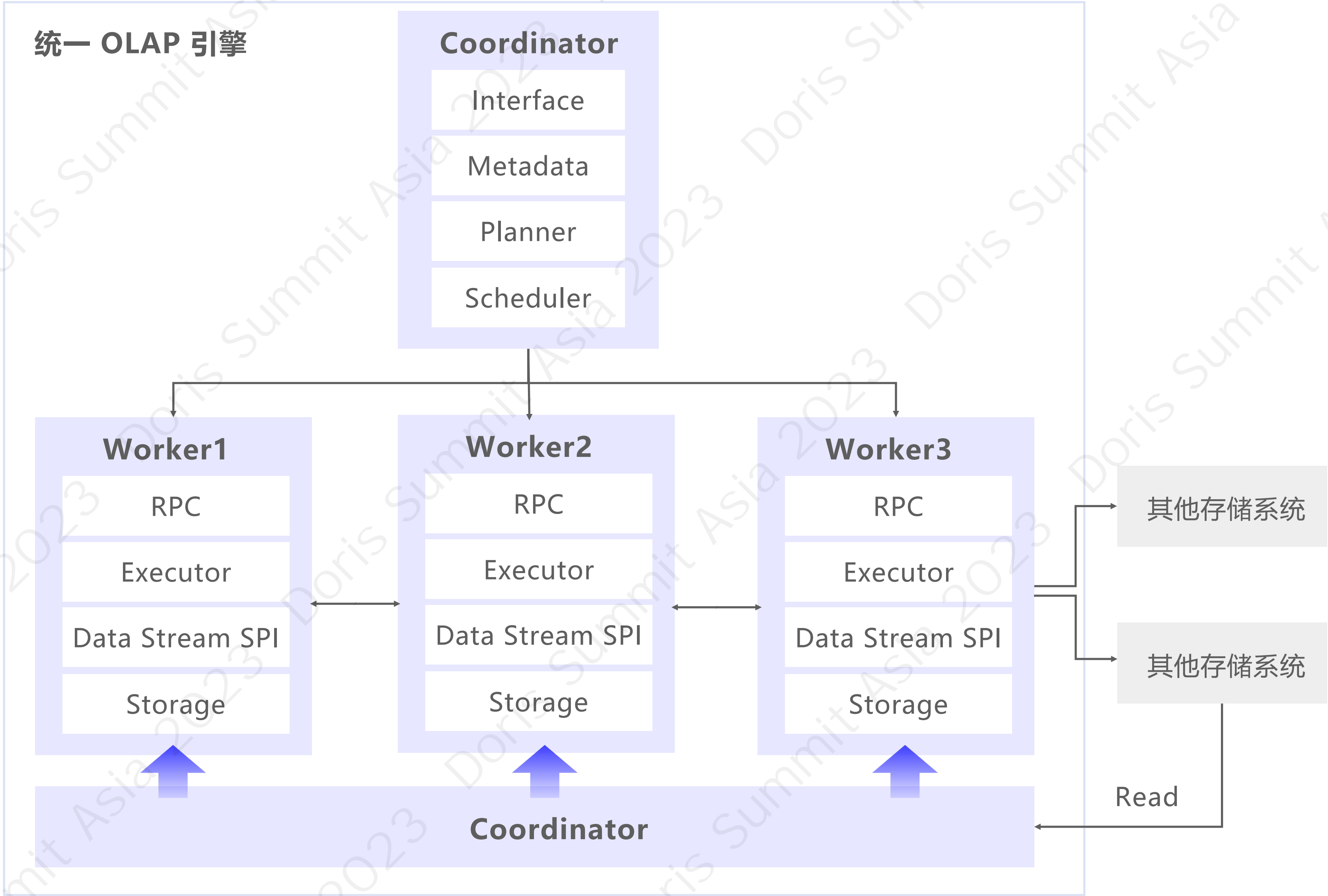
过去针对不同场景选择不同引擎的方式带来了“三高”问题

维护成本高	资源浪费高	使用成本高
重复建设，精力分散，迭代速度变慢，支持新场景更困难	业务数据需要存到多个引擎中，资源浪费	业务学习使用，数据治理成本增加

目标：建设能满足各种业务场景的统一OLAP引擎，解决“三高”问题

统一 OLAP 引擎：架构设计

- MPP架构的计算引擎 (通用性)
- 内置存储格式和引擎 (高性能)
- 物化视图和多种索引 (多场景)



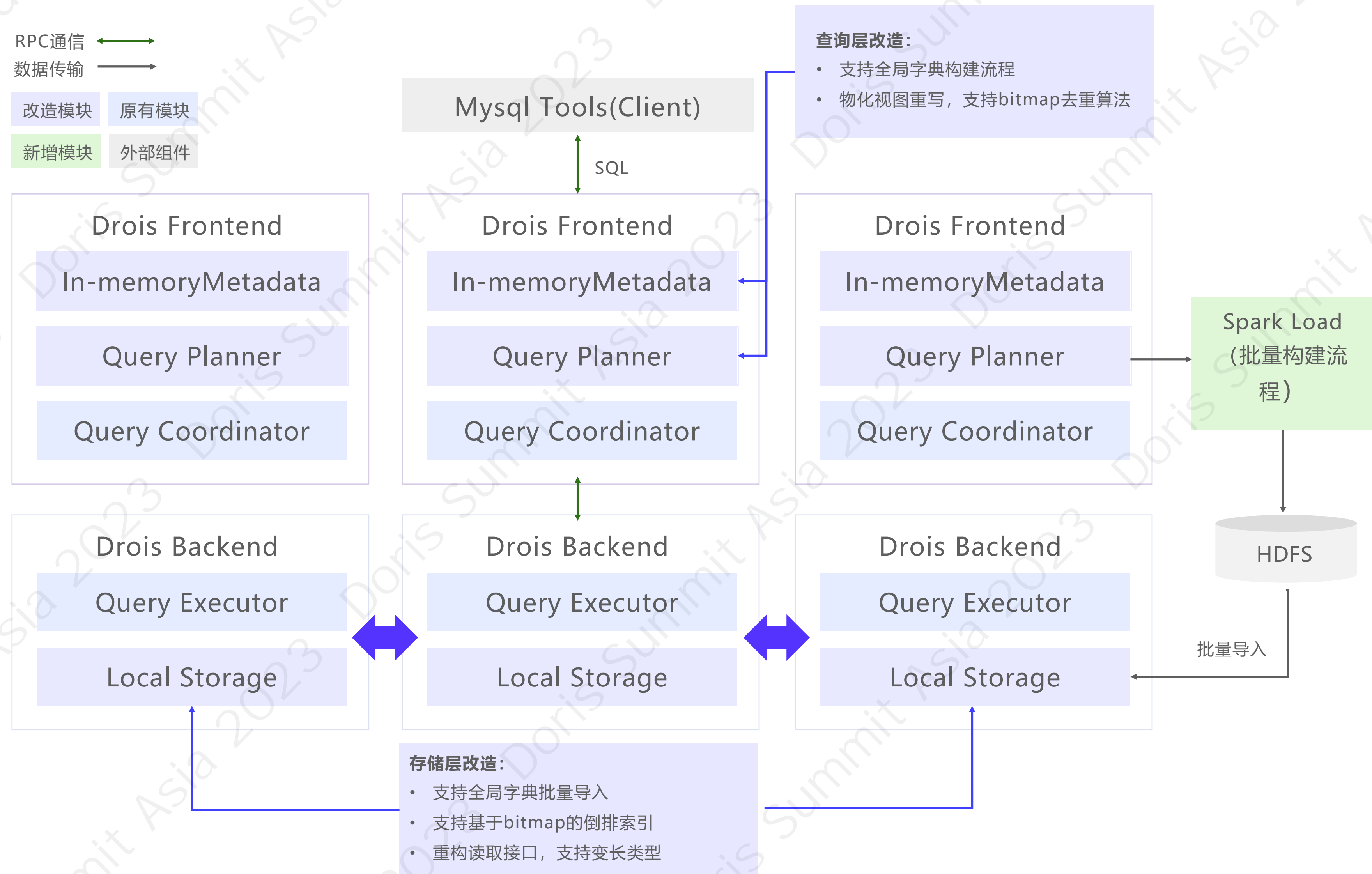
统一 OLAP 引擎：技术选型

基于Presto改造	基于Doris改造	自研
技术栈匹配度	项目落地周期	系统掌控力
代码可扩展性	业务迁移成本技	长期迭代效率项
项目落地周期	术栈匹配度代码	目落地周期
业务迁移成本	可扩展性	实施风险

统一 OLAP 引擎 POC 验证

核心思路

- 选型改造 Doris
- 复用社区基于 MPP 的灵活查询能力，Mysql 接口，增量更新等能力
- 存储层预聚合，倒排索引等优化手段“多模态共存”
- 保留通用计算性能，特殊场景能力对齐 Kylin/Druid**



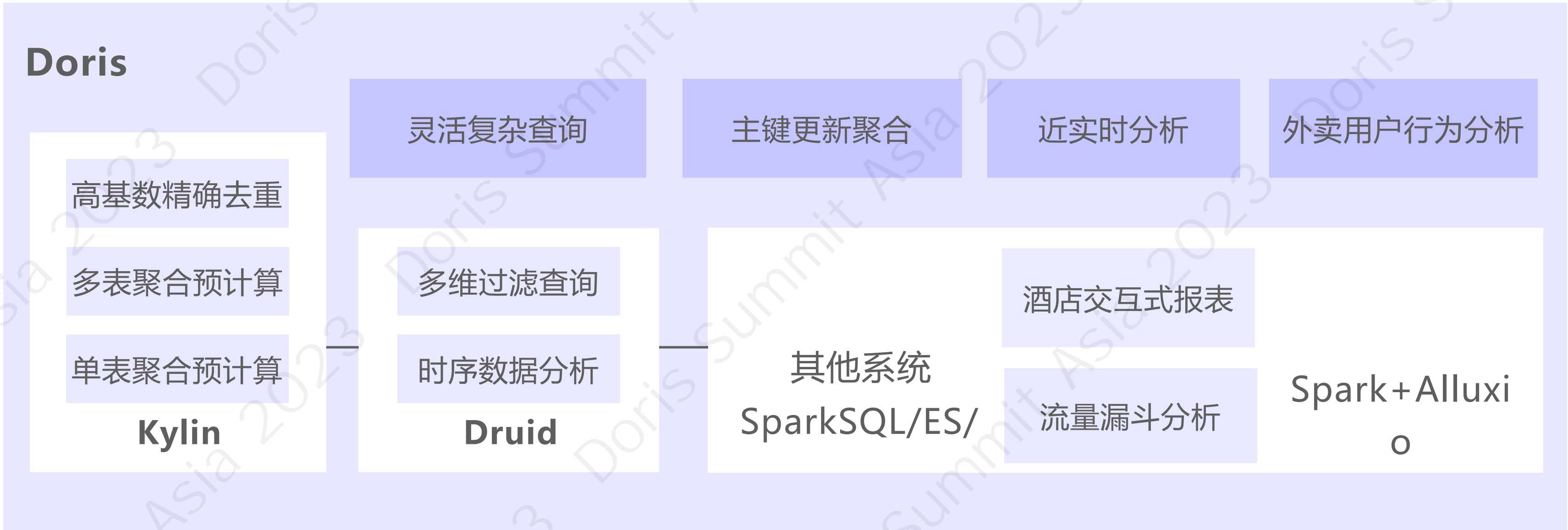
统一 OLAP 引擎 POC 验证

核心思路

- 选型改造 Doris
- 复用社区基于 MPP 的灵活查询能力，Mysql 接口，增量更新等能力
- 存储层预聚合，倒排索引等优化手段 “多模态共存”
- 保留通用计算性能，特殊场景能力对齐 Kylin/Druid



存量场景覆盖 ↓



基于Doris的统一OLAP引擎实践

软件架构层优化：开源社区深度合作引擎内核优化+ 内部平台产品建设提升易用性

- **建模优化**：基于历史查询的智能物化、表模型优化建议
- **SQL优化**：CBO 查询优化器
- **执行层**：向量化执行、Pipeline执行框架、Colocate join 优化
- **存储层**：Rollup、Bitmap 正交分桶、倒排索引
- **工具链**：回归测试框架、性能 Profile、大查询自动检测与治理、集群健康监控

新硬件应用（调研和摸索阶段）：

- **算力**：GPU 加速查询执行器，目前代表性 GPU 数据库系统大部分是学术项目，使用 GPU 加速查询主要面临数据传输（PCIe 带宽有限）、显存管理、算子 GPU 化实现方面的挑战
- **IO**：通过 NVM/SSD 实现多级存储，加速对热点数据的访问

OLAP 引擎优化举例：Colocate Join

问题

外卖经营分析大表关联查询执行超时

原因

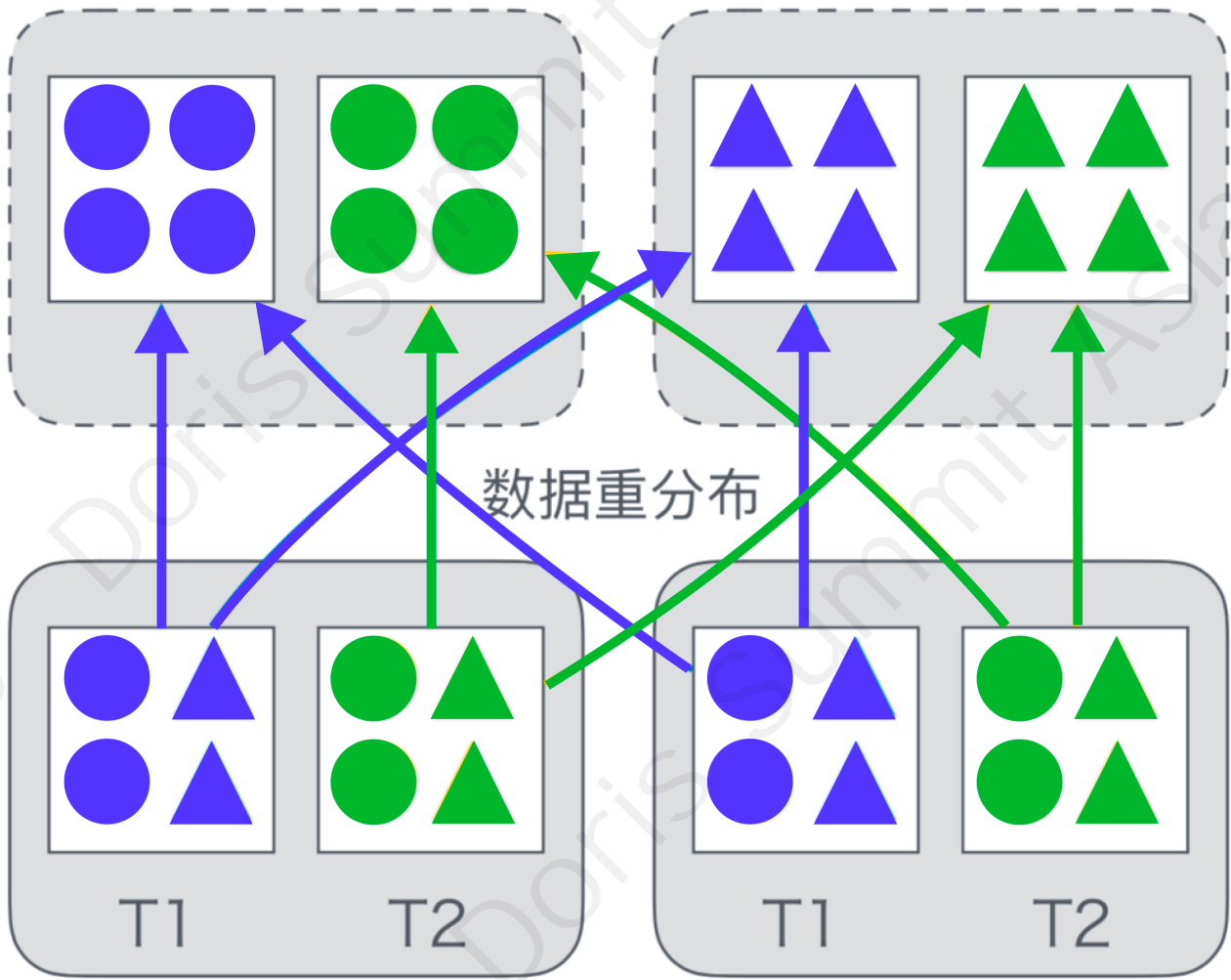
Shuffle Join的数据重分布开销大

- 哈希计算
- 序列号/反序列化开销
- RPC开销

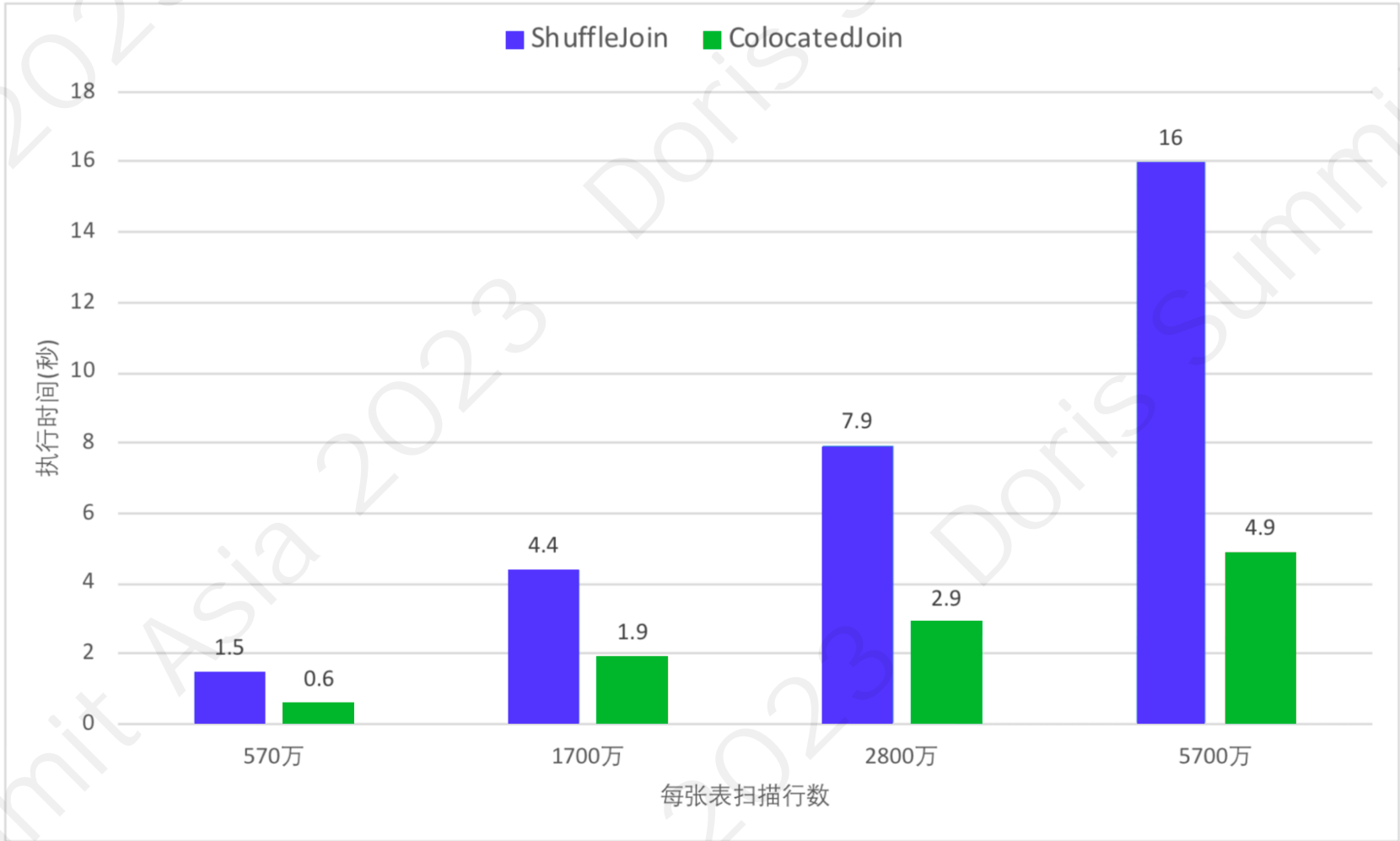
方案

Colocated Join

- 多表预先根据关联字段同分布存储

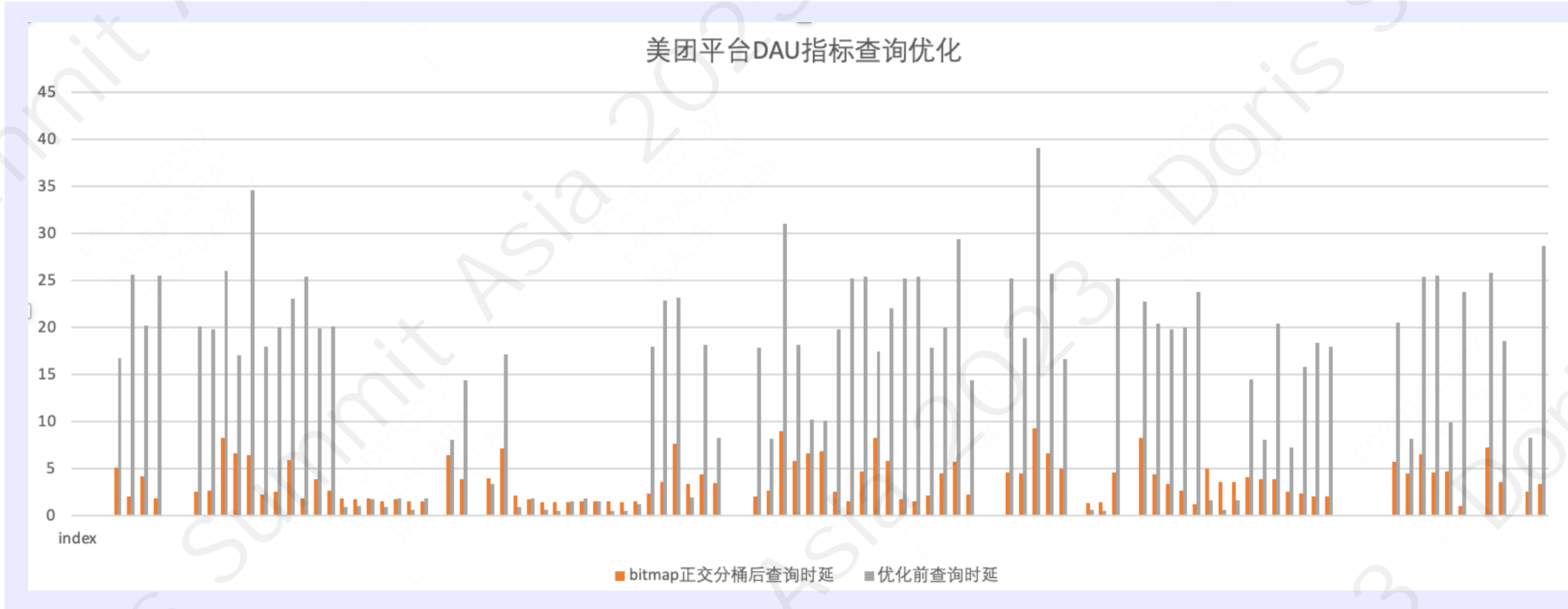
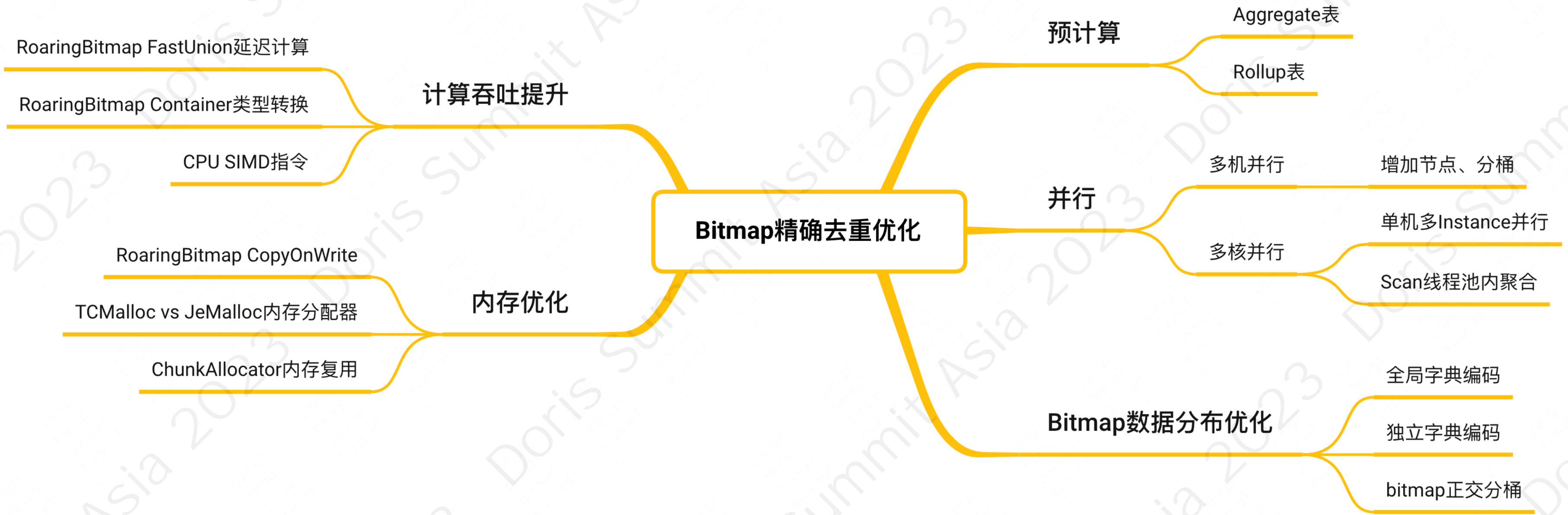


基于Shuffle Join的实现



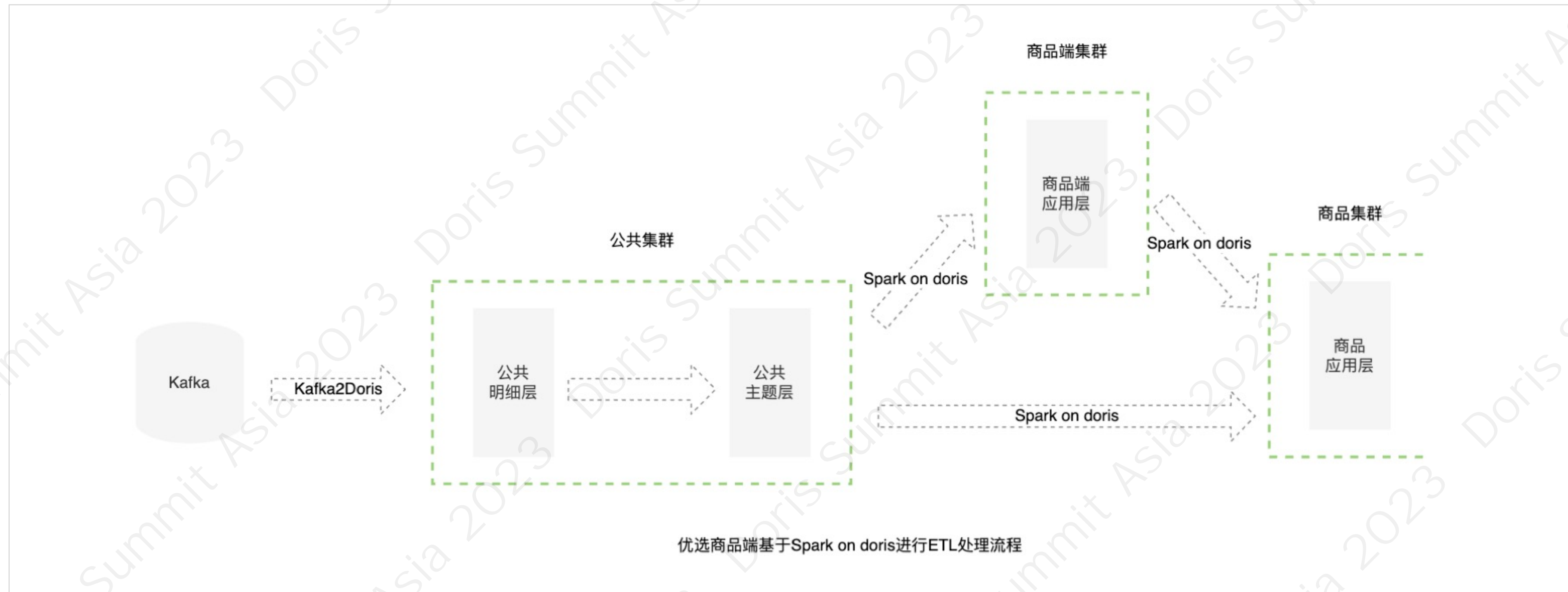
Join性能平均提升3倍，满足外卖商家报表需求

OLAP 引擎优化举例：Bitmap 精确去重



SQL百亿数据量级亿级别基数精确去重
指标计算平均性能提升4~5倍，单表查询10s内

OLAP 引擎优化举例：Spark on Doris



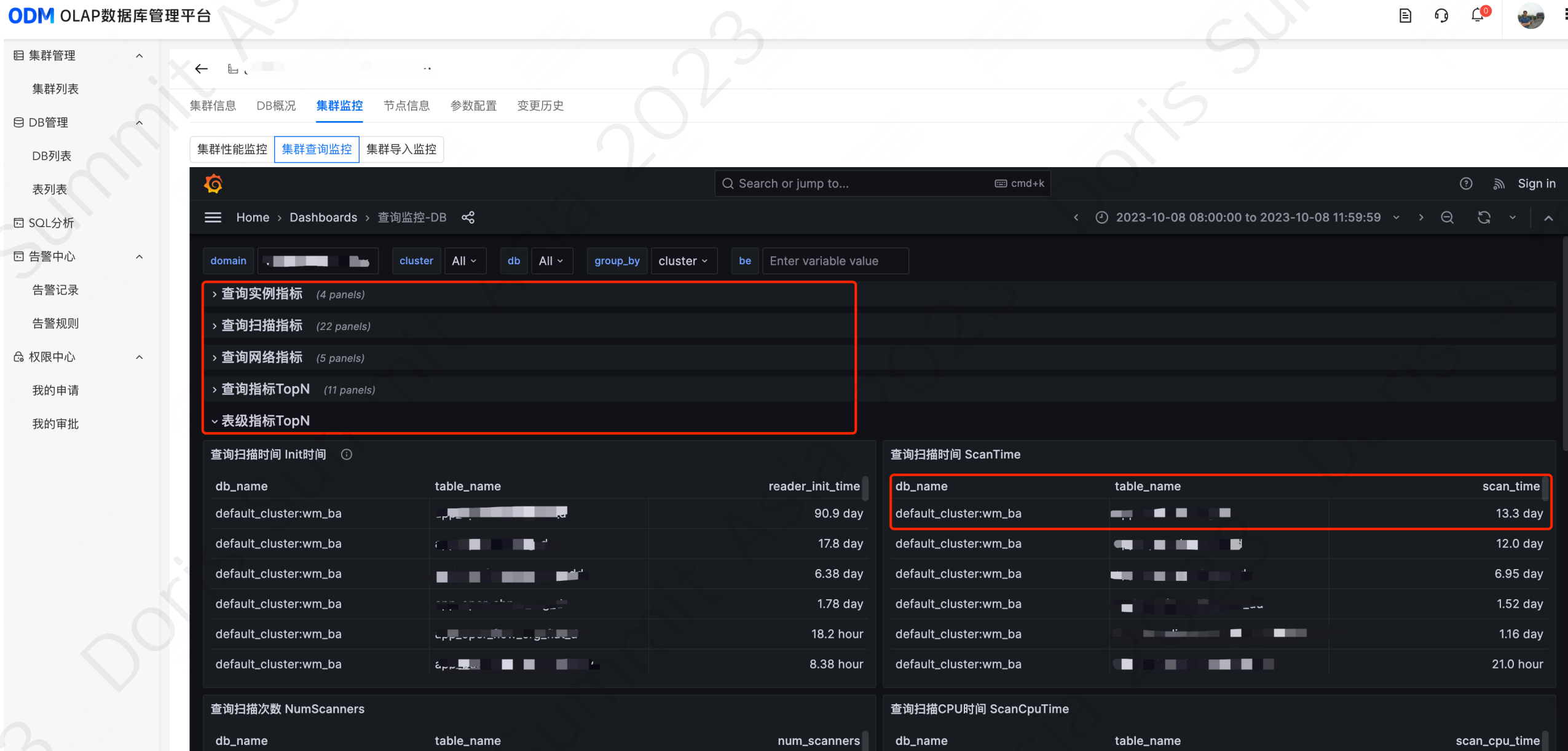
支持跨集群读写能力，避免了重复计算逻辑，减少了计算和存储资源

将复杂的数据加工逻辑放在Spark上执行，大大降低 Doris 的计算压力

对外提供的是SQL接口，原来的 Doris2Doris数据加工任务可以无缝迁移

OLAP 工具建设举例：大查询监控与治理

- 集群关键信息埋点收集
- 按查询实例、表、库粒度聚合



- 异常查询监控告警
- 打通大查询的监控与分析链路



OLAP 工具建设举例：查询性能可视化分析

[illegible]

3 当前挑战与未来规划

当前问题与挑战

资源效率：存算一体架构在需要资源伸缩场景面临的挑战

- 业务高低峰：节假日促销活动、用户在不同时间段的分析频率等因素带来的明显波峰和低谷
- 计算/存储资源扩容：计算资源不够需要扩容时，连带存储资源的扩容，磁盘空间使用率低

读写稳定性：从离线分析到支持在线决策，业务对可用性要求越来越高

- FE可扩展性：集群规模带来的元数据膨胀，FE单点瓶颈以及故障恢复时长影响
- 混合负载下的隔离：同一套分析系统里既要支持数据导入、加工处理，又要支持高并发低时延查询

查询性能：从偏固定报表查询场景往 Adhoc 查询场景拓展

- 场景复杂：既有千亿级带精确去重指标的流量分析、也有多张大表现场关联查询的经营分析
- 使用灵活：支持用户自助指标维度拖拽分析

未来规划

1

存算分离架构

2

弹性扩缩容

3

CBO 查询优化器

4

Pipeline 执行框架

5

智能物化视图构建



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>