

基于 SelectDB 的实时数仓平台构建

目录

01 基于 CDP 的旧实时数仓

02 面临的困难与挑战

03 基于 SelectDB 的解决方案

04 未来展望

01

基于 CDP 的旧实时数仓

实时计算：持续、低时延、事件触发的计算作业

实时数据

- 实时计算面对的数据是实时且流式的，数据按照时间发生顺序被实时计算程序订阅和消费

持续计算

- 触发源为无界流式数据，一旦有新的流数据进入实时计算系统，它就立刻发起并进行一次计算任务，因此整个过程是持续进行的

实时数据集成

- 流处理的计算结果可以同流式数据一样，持续的写入目的数据存储

维度	批量计算	实时计算
处理对象	有界的静态数据集	无界的动态数据流
处理时延	高：小时到分钟级	低：秒级到毫秒级
吞吐量	大	中到小
容错机制	相对简单 数据不变可以重复计算	复杂 中间状态需要保证一致性
触发机制	定时任务	持续运行

系统部分大数据组件



02

面临的困难与挑战

数据具有一个重要特性，即其价值会随着时间的推移逐步降低

所以如何高效地完成海量数据的清洗、转换以及计算分析等环节，为业务部门提供最具价值的时效性数据，以支持快速决策和响应，成为当前面临的重大挑战；

- 除了要求平台要有高效的 T+1 的离线处理能力，
- 还要具有日中的实时数据处理能力；

Kudu 实时数据交互

- 原有 Kudu 实时数据的写入能力有限，包括其扩展性；
- 数据获取方式有限，一般需结合 Impala 进行 SQL 查询；
- 数据管理方式有限，需结合 HMS 进行元数据管理；
- 数据权限控制能力有限，需结合 Ranger 进行权限控制；
- 数据表组织形式有限，无法提供针对不同场景的表存储优化；

.....

原有架构的局限性

报表查询

Impala + Kudu 作为大数据报表分析的底层查询引擎来完成业务分析诉求，在应用初期，业务查询数据量、查询复杂度以及查询时延要求都不高，该技术架构可满足业务要求；

随着公司业务的蓬勃发展、各个业务线需要查询的指标复杂程度逐步提升，该架构逐渐展露出局限性，难以继续满足当下业务方的查询诉求。

- 权限控制、资源细粒度的控制、本身产品的局限性等，都无法满足。

运维成本

CDP 集群的维护极为复杂，成本居高不下并且稳定性欠佳，运维压力不小。

- SelectDB 数据库核心架构简洁，只设有 FE（FrontEnd）和 BE（Backend）
- 两种角色，不依赖于其他外部系统。
- 集群支持线性扩展，节点异常退出后可快速拉起恢复服务，运维简洁高效。
- 公司运维支持团队以及技术团队的响应也十分的及时且高效。

03

基于 SelectDB 的解决方案

使用方式

- 采用 MySQL 协议，高度兼容 MySQL 语法，支持标准 SQL，用户可以通过各类客户端工具来访问 SelectDB，并支持与 BI 工具的无缝对接。

存储模型

- 支持 aggregate key 模型、unique key 模型和 duplicate key 模型。丰富的数据模型为多样化的应用场景提供了便捷。
- 将表中的数据切分成多个 tablet，数据能够自动进行副本修复和均衡。
- 采用列式存储，按列进行数据的编码压缩和读取，能够实现极高的压缩比。

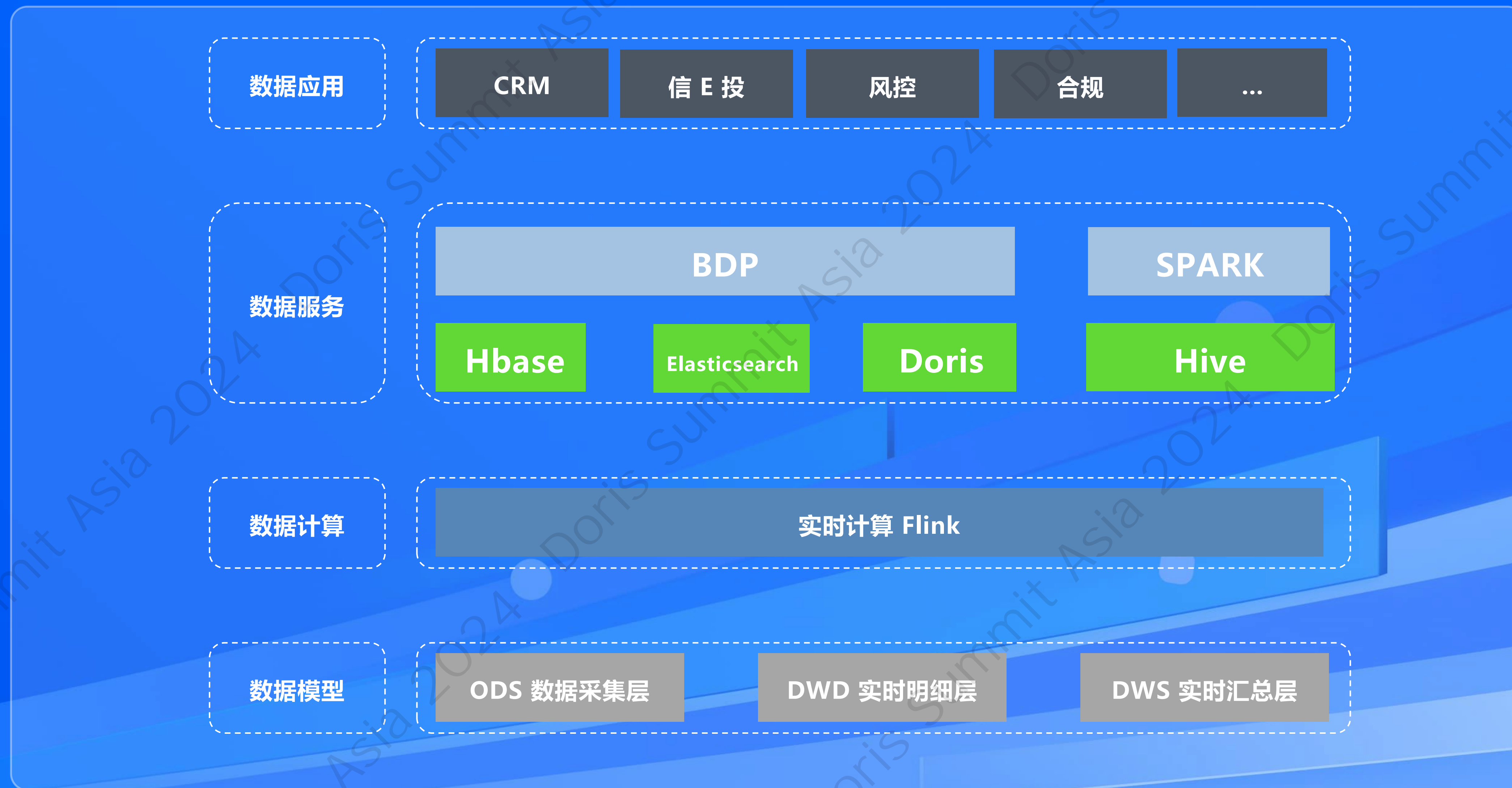
数据导入事务保障

- SelectDB 通过两阶段提交保证多表导入的原子性，通过 label 机制保证数据导入不丢不重。
- 在并发方面，SelectDB 通过 MVCC 技术机制实现海量数据高并发控制。
- 查询引擎采用 MPP 的计算模型，节点间和节点内都并行执行，也支持多个大表的分布式 Shuffle Join，从而能够更好应对复杂查询。
- 支持强一致物化视图和异步物化视图。
- 具备库表行列四级权限控制来实现数据的安全合规。

涉及的组件



系统架构



数据模型



Apache Doris 实时数据湖

有了 Flink 实时数仓，为什么还要做实时数据湖？

开发门槛

- Flink 开发上手门槛较高
- 开发周期较长

计算复杂度

- Flink 对复杂分析的支持不足

运维成本

- Flink 任务运维成本较高

实时性要求

- 部分应用场景对实时性要求到分钟级即可
- 不需要持续计算

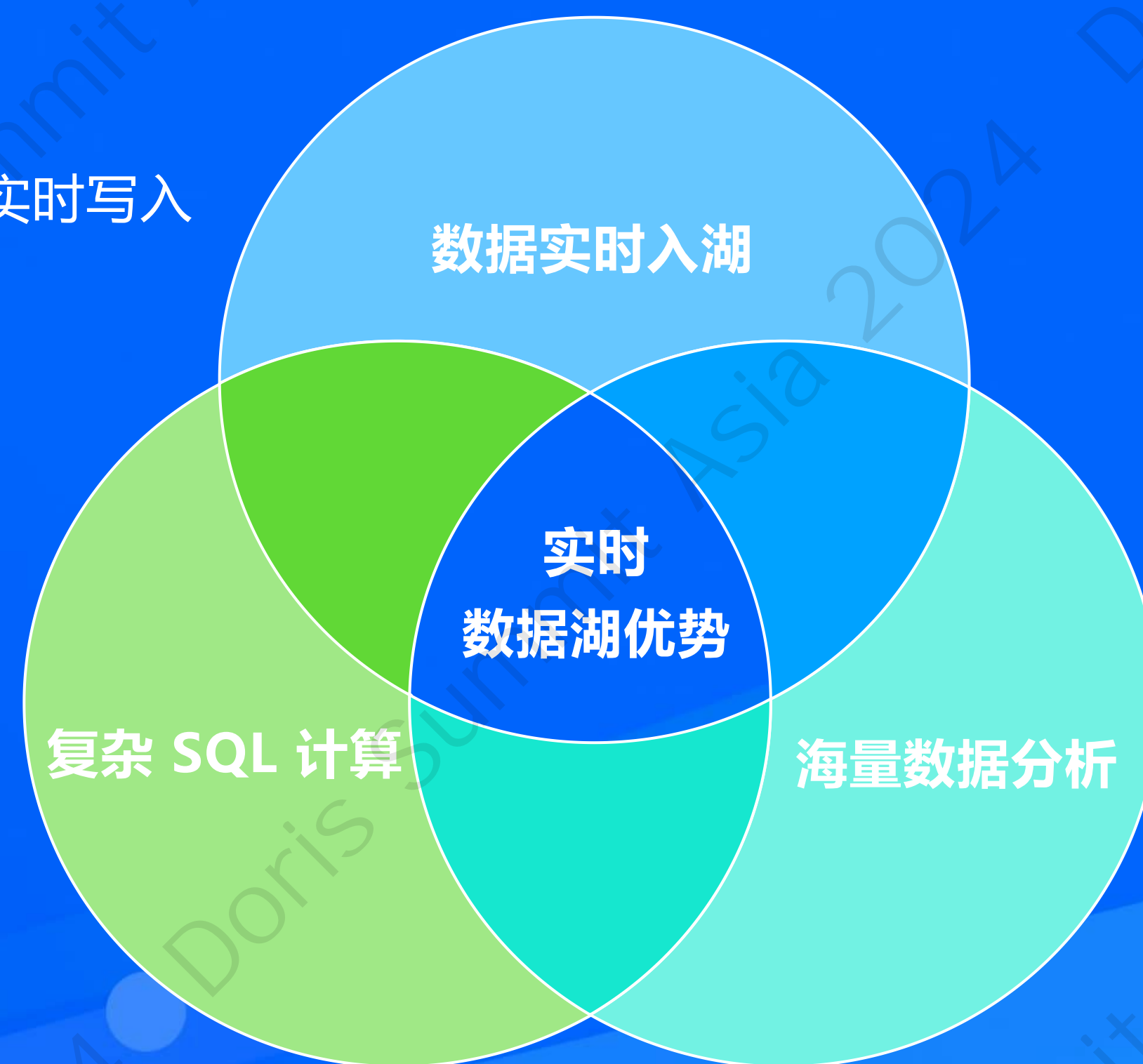


基于 Apache Doris 实时数据湖系统架构



基于 Apache Doris 实时数据湖的优势

- 业务系统数据通过 Flink 实时写入



- 支持复杂的OLAP分析

- 汇集各个系统数据
- 分担交易系统数据库查询压力

Apache Doris 报表加速性能表现

报表查询 加速

- 亚秒级响应时间即可返回海量数据下的查询结果

复杂分析

- 支持高吞吐的复杂分析场景
- TPC-DS 与其余测试产品相比，性能很优秀

实时数据

- 秒级的数据实时更新（主键表）与追加

Apache Doris 报表加速落地场景

1

证金客户标签类数据分析

2

资管报表数据查询

3

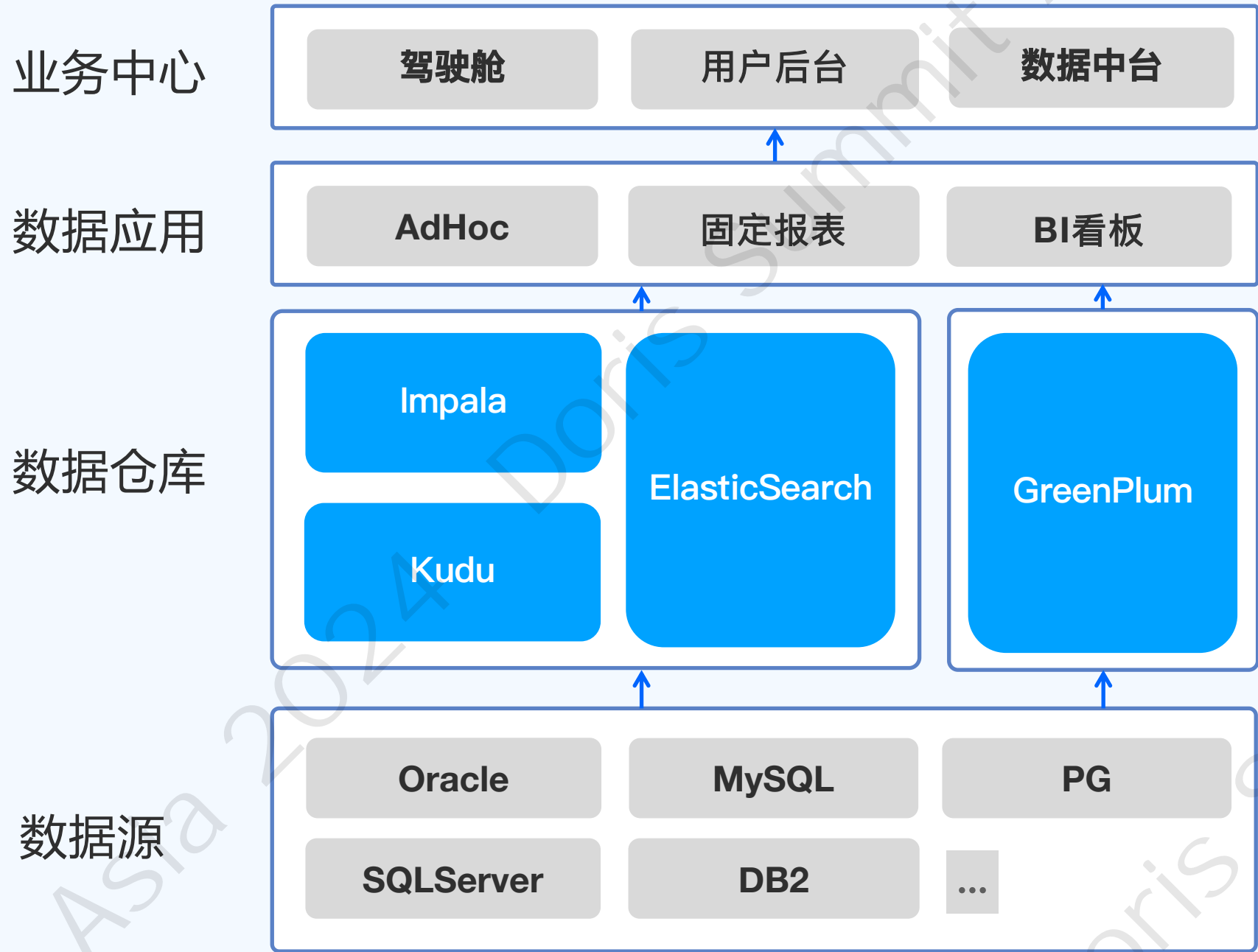
固收报表数据查询

4

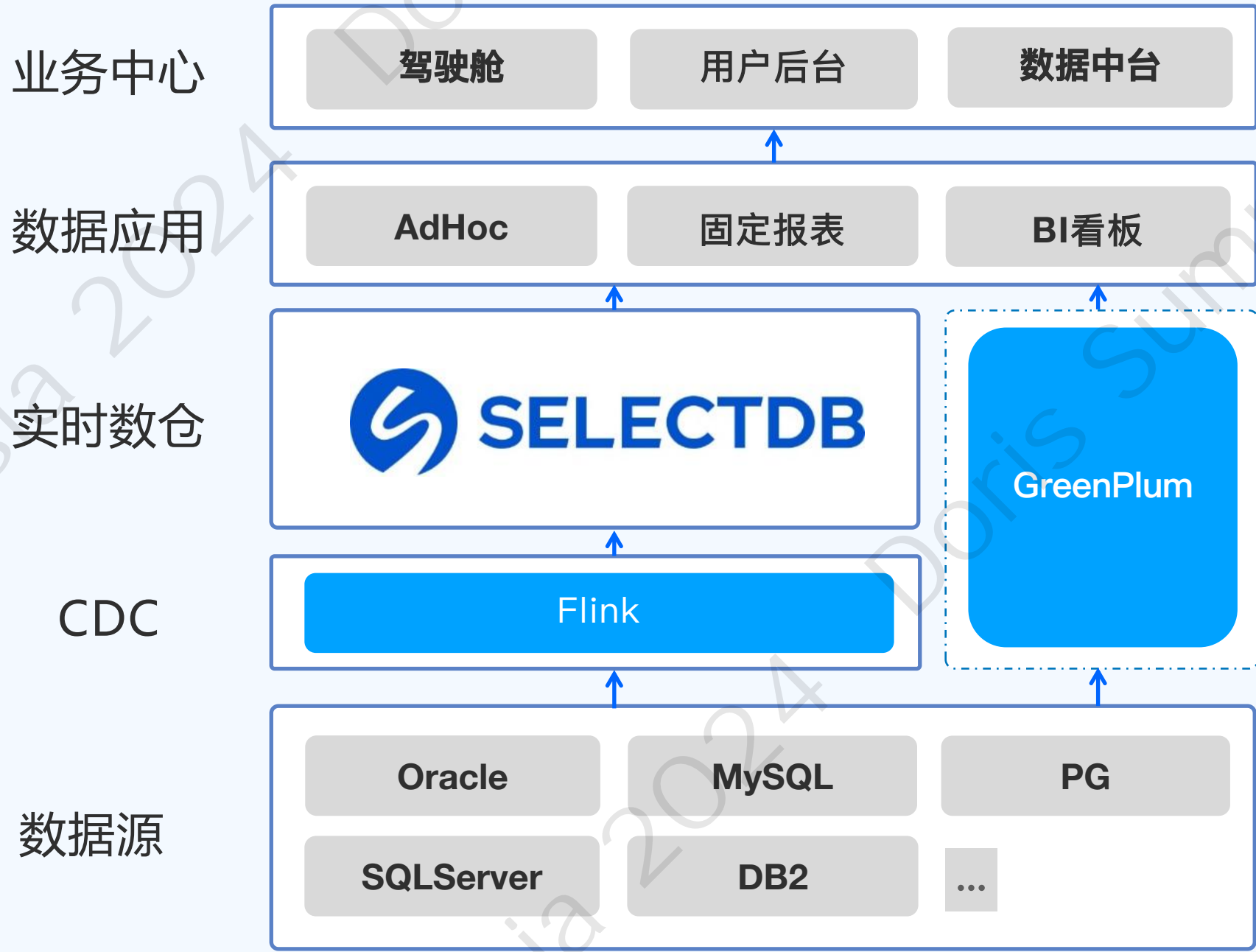
财富委经营数据实时看板

实时数据仓库改造演进

改造前整体架构



一期改造后整体架构



业务场景

- 典型的数据中台架构，即离线数仓+诸多数据应用组建而成的数据应用分析平台
- 主要对内部数据平台及公司内部业务提供数据支持

痛点诉求

- 数据时效性低，绝大部分场景还是 T+1
- 查询时延高，单个 Query 查询基本都在分钟级以上
- 架构维护成本高，不同组件之间的适配较难

一期改造收益

- 数据时效性提升至分钟级，部分业务秒级
- 单个查询时延在毫秒级到秒级
- 架构简单，维护成本大幅度下降，一个单一组件完成了 Impala+Kudu+ES 的替换

SelectDB 带来的提升

- 查询性能，相较 Impala + Kudu 的架构，实现了数十倍的查询加速；
- 运维成本，运维金钱、时间、人员成本都显著降低。
- 功能拓展，SelectDB 不仅具备原有架构的所有常规查询能力，还新增了联邦查询、倒排查询等功能。
- 时效性，SelectDB 满足了秒级实时查询的时效性要求，确保了数据分析的及时性。

整体而言：

- 使用 SelectDB 替换原有架构后，SelectDB 提供了秒级实时的数据分析查询能力，提升了业务决策的效率与精准度，使公司能够迅速响应市场变化，把握先机。
- 在平台方面，SelectDB 简化了技术架构的复杂度，降低了运维成本。
- 相较于原集群，综合成本降低了 50%。

04

未来展望

- 简单易用的架构设计、技术栈的统一管理
- 完全满足现行需求以及未来可能的多样化需求
- Apache 基金会顶级项目
- 充足的英文文档
- 强悍的计算能力与便捷的部署使用能力
- 基于开源的生态融合能力
- 团队研发能力和服务支持力度

Thanks for Watching !