Apache Doris 3.0 云原生存算分离架构的设计与实现

周飞

飞轮科技云原生技术负责人 & Apache Doris Committer

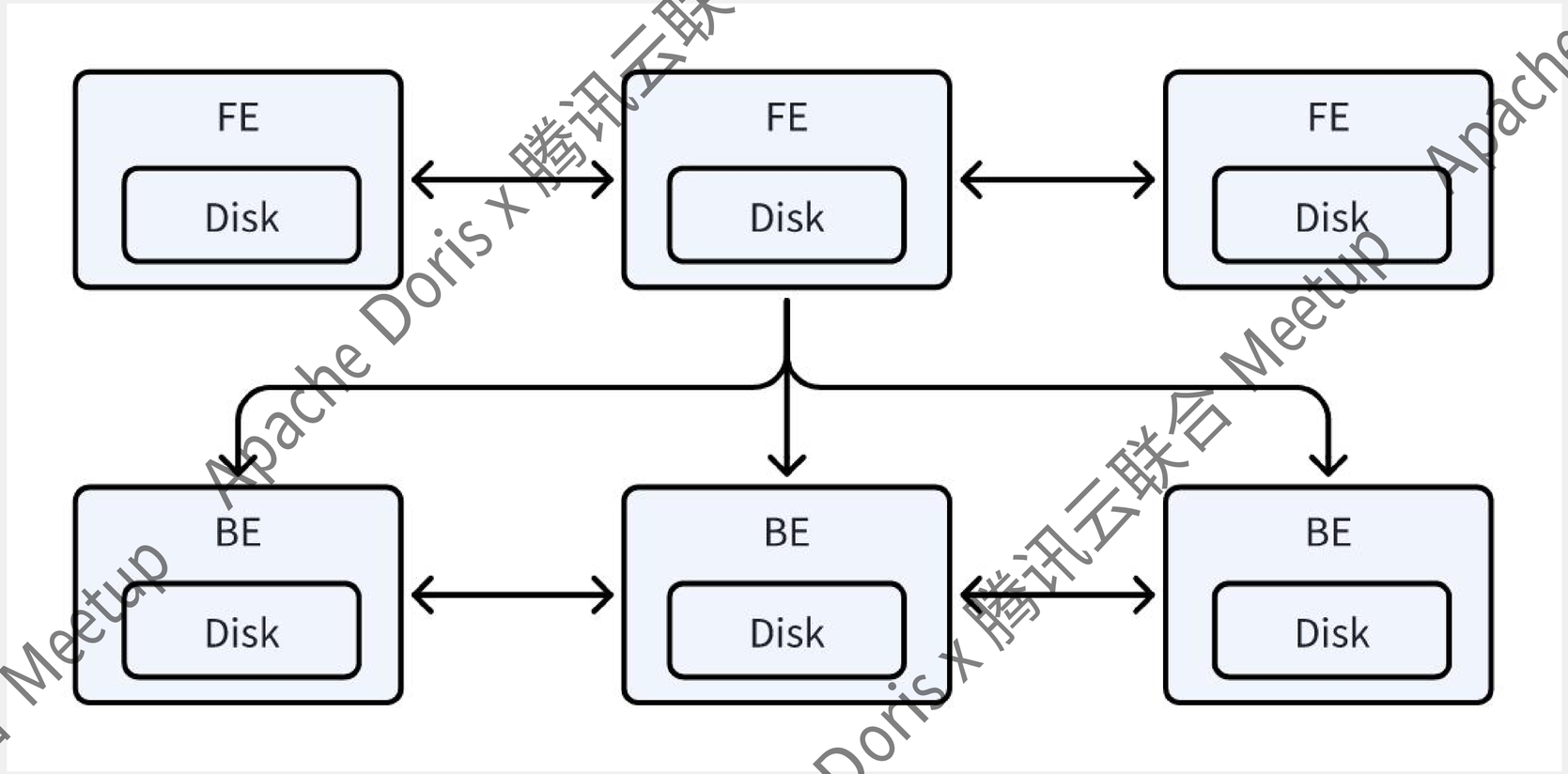
个人介绍

周飞

- 飞轮科技云原生技术负责人、Apache Doris committer
- 主导了 Apache Doris 存算分离设计与实现
- 有近 10 年云存储以及分布式数据库架构设计、研发与团队管理经验

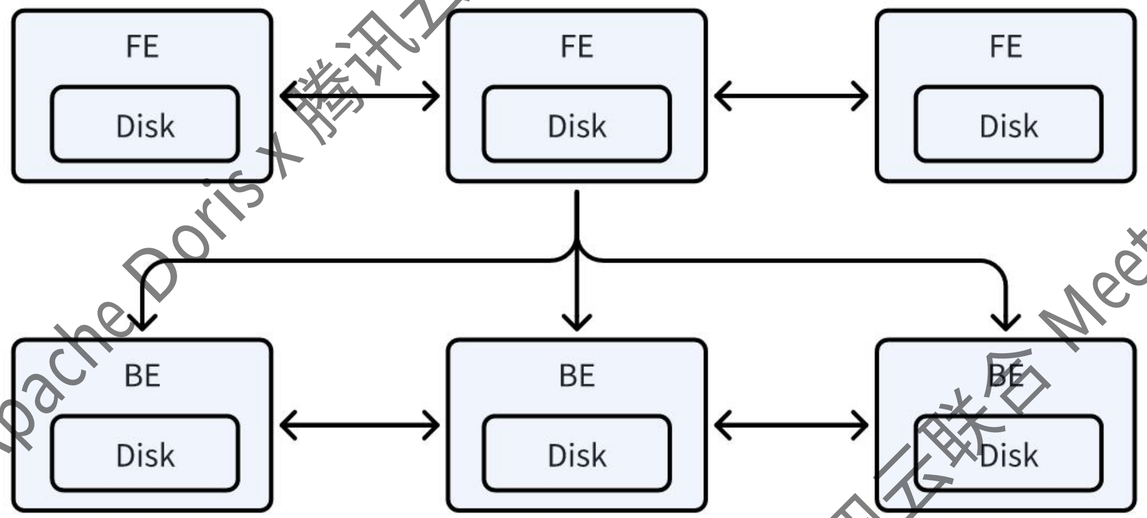
目录

Apache Doris 存算一体模式



在存算一体架构下，BE 节点上存储与计算紧密耦合，数据主要存储在 BE 节点上，多 BE 节点采用 MPP 分布式计算架构。

Apache Doris 存算一体模式



- **部署简单**：仅 FE 与 BE 进程，BE 和 FE 都可以单独扩容
- **稳定可靠**：不依赖共享存储系统
- **性能优异**：计算节点访问本地存储

为什么需要存算分离

低成本与资源弹性

- 计算和存储解绑，单独扩缩容
- 计算资源波谷波峰，灵活弹性
- 数据存储冷热效应明显

负载隔离

- 读写任务分离
- 更彻底的业务隔离，解决不同业务间的相互影响以及资源抢占问题

数据共享

- 单一数据面向不同的分析负载使用
- 数据快速移动、快速备份恢复
- Single Source of Truth

云基础设施的成熟

- 云上基础设施逐步完善，提供可靠的共享存储
- 完全按量付费，灵活可控

目录

设计出发点 - 性价比与架构稳定性

如何降低成本

- 引入对象存储节省冷数据资源
- 增加弹性计算能力，按需使用计算资源

如何迭代

- 绝大多数用户已采取存算一体架构
- 升级过程中需要保证对已有架构的兼容

设计目标

负载隔离

读写分离

业务隔离

内部负载隔离

低成本

存储成本大幅下降

计算和存储可以独立弹性

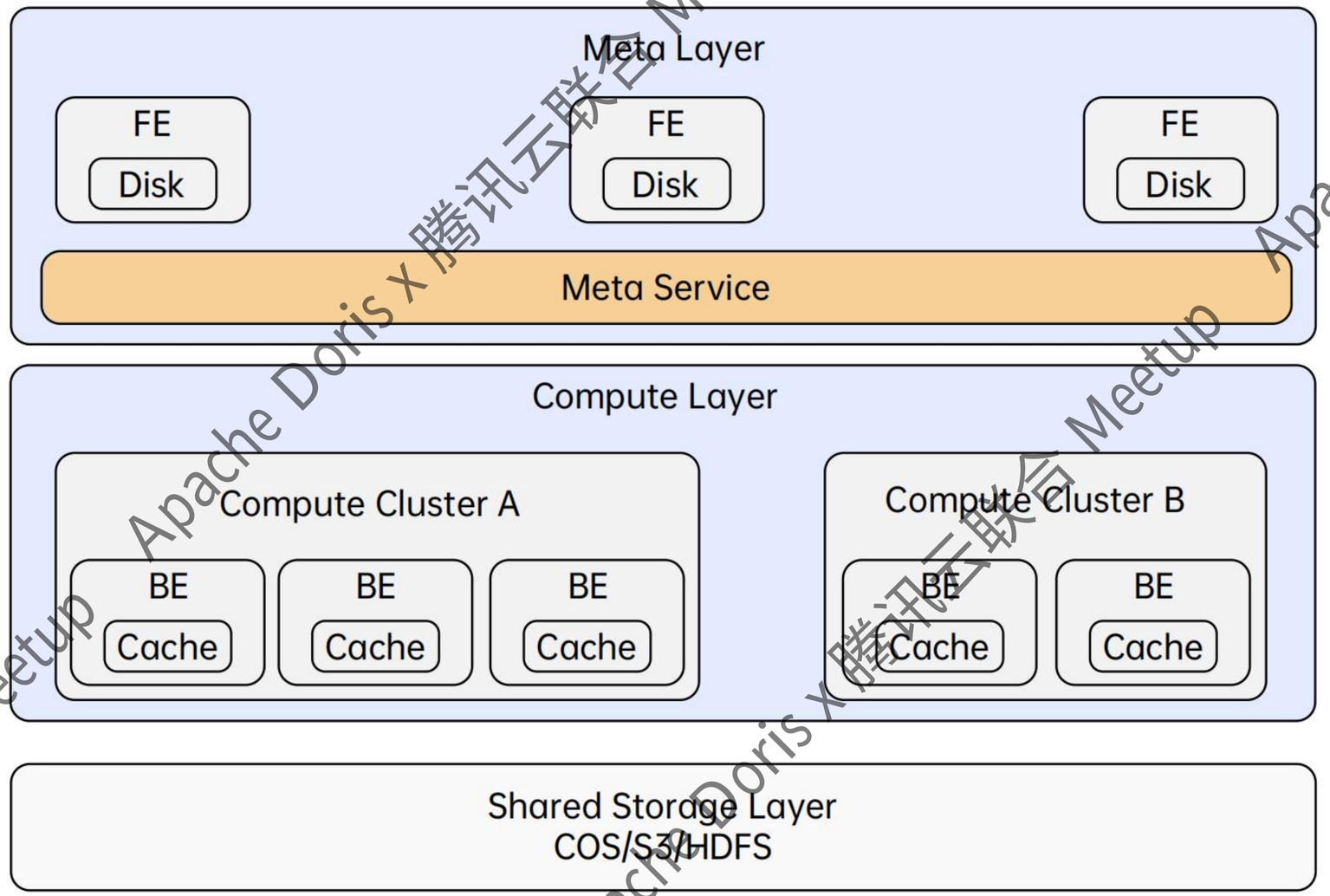
使用业务的波峰波谷调整计算资源

数据共享

统一元数据服务

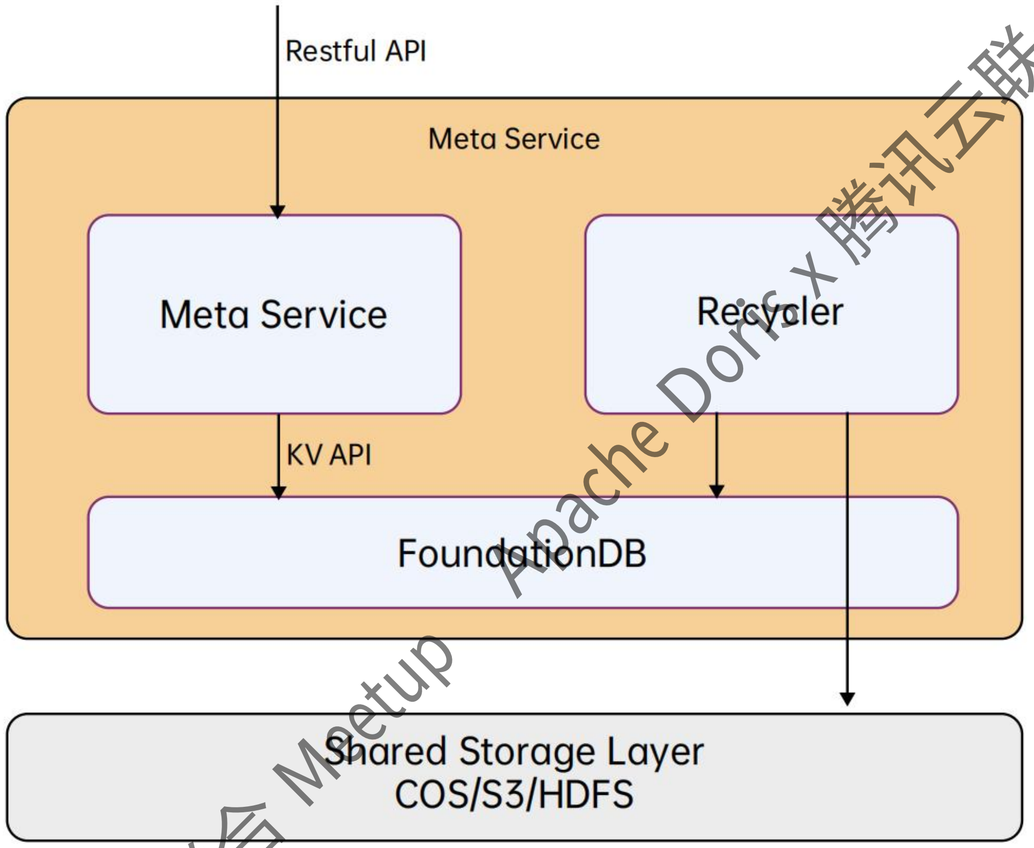
服务统一存储

存算分离整体架构



目录

元数据服务层



- 统一语义层：Restful API
- 高性能分布式 KV 存储：FoundationDB
- 正向垃圾回收：Recycler

数据存储层

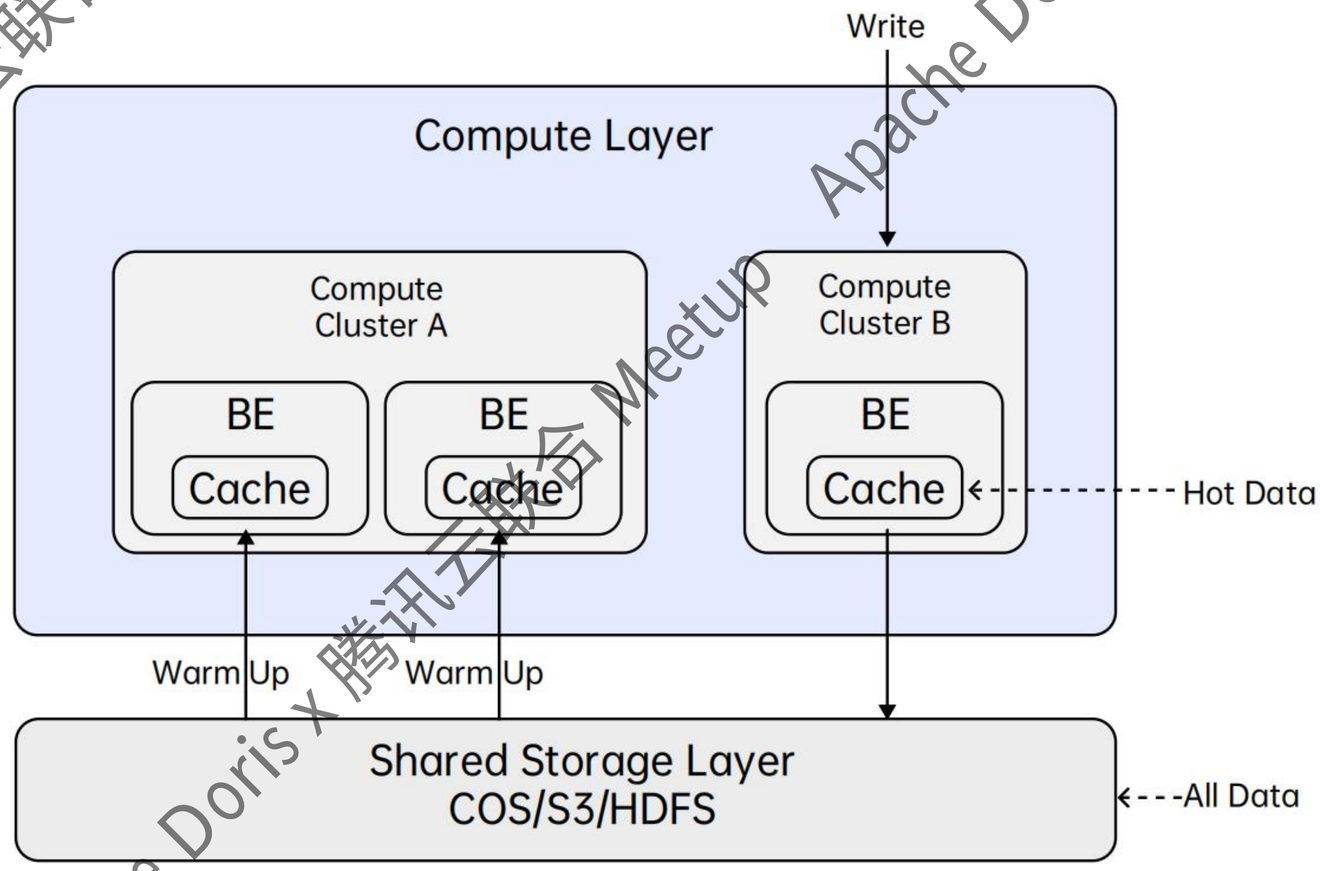
成本最高降低 90%

- **存算一体**
- 全量数据 * 3 * 块存储价格
- **存算分离**
- 热数据 * 1 * 块存储价格 + 全量数据 * 对象存储价格

- 最高可以节省 90% 以上

灵活的 Cache 管理

- 多优先级淘汰策略：LRU、TTL
- 缓存主动预热：table 粒度、cluster 粒度、导入



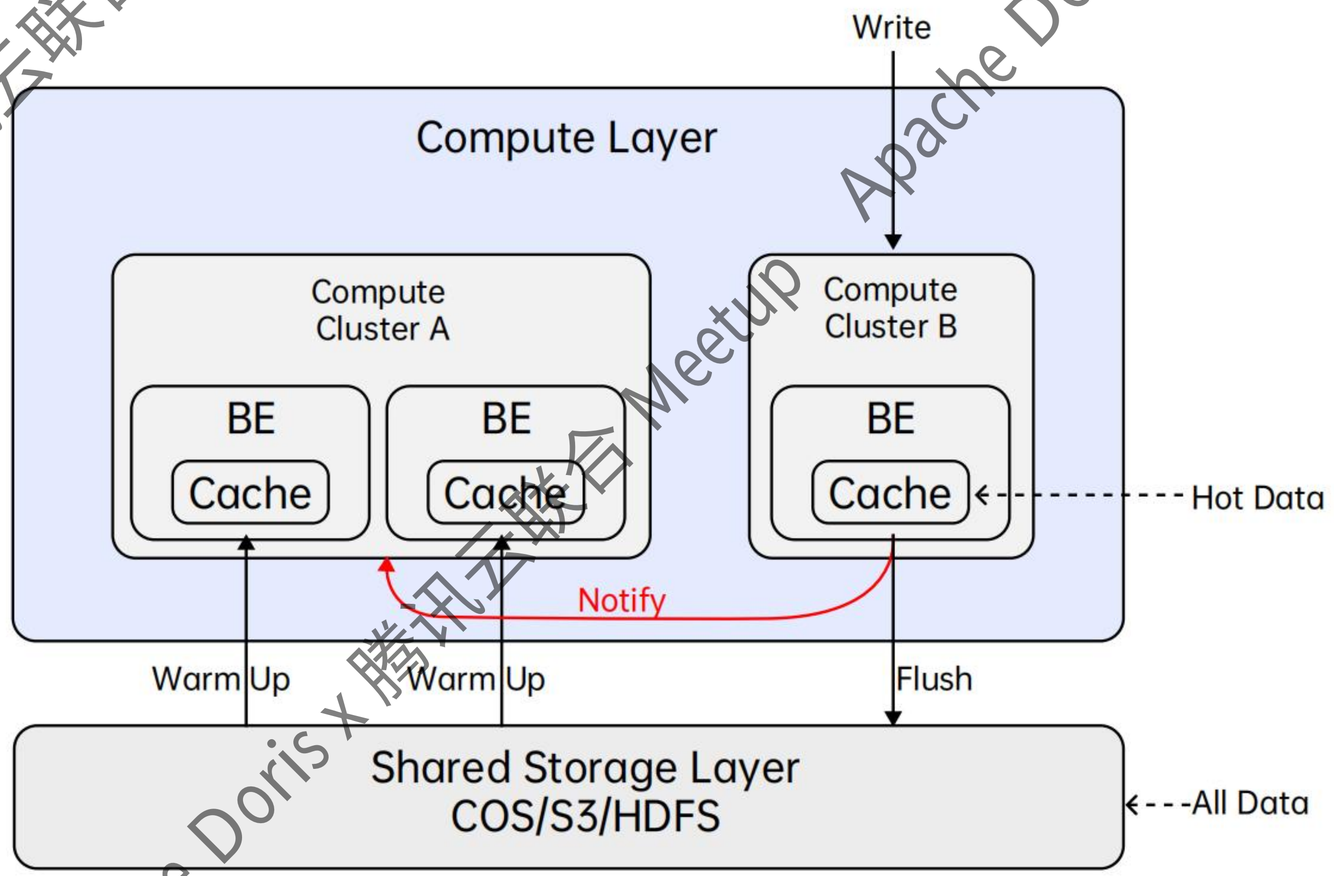
数据计算层-数据导入

数据导入流程

1. 数据进入协调者 BE
2. 数据分发到多个 BE
3. 数据写入 Cache
4. 数据写入 S3
5. 读写分离 Cluster 预热 Cache

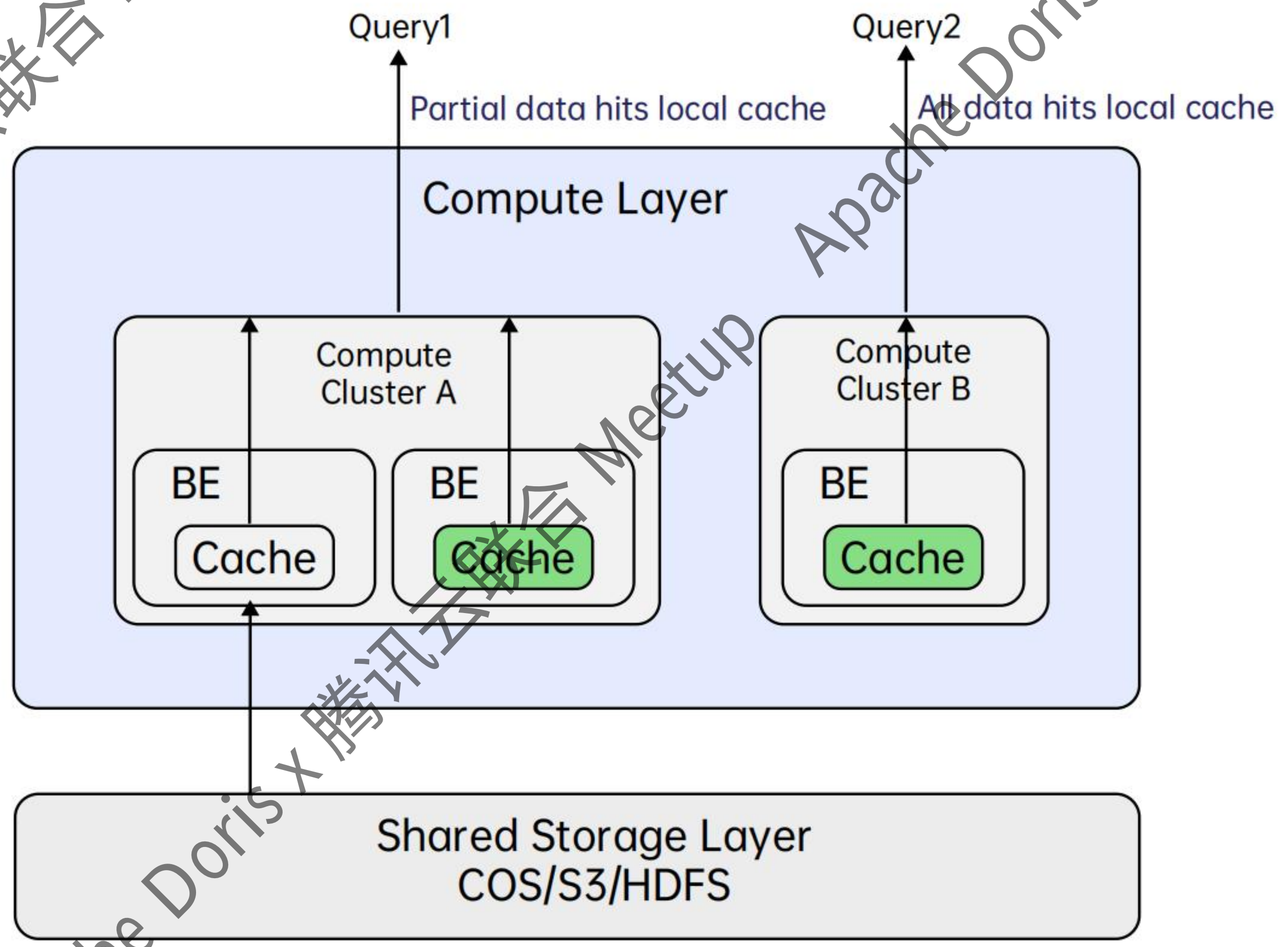
数据导入效率更高

- 只需处理单副本数据
- 数据和 BE 没有固定的关系
- 没有 Publish 阶段，写入流程更短

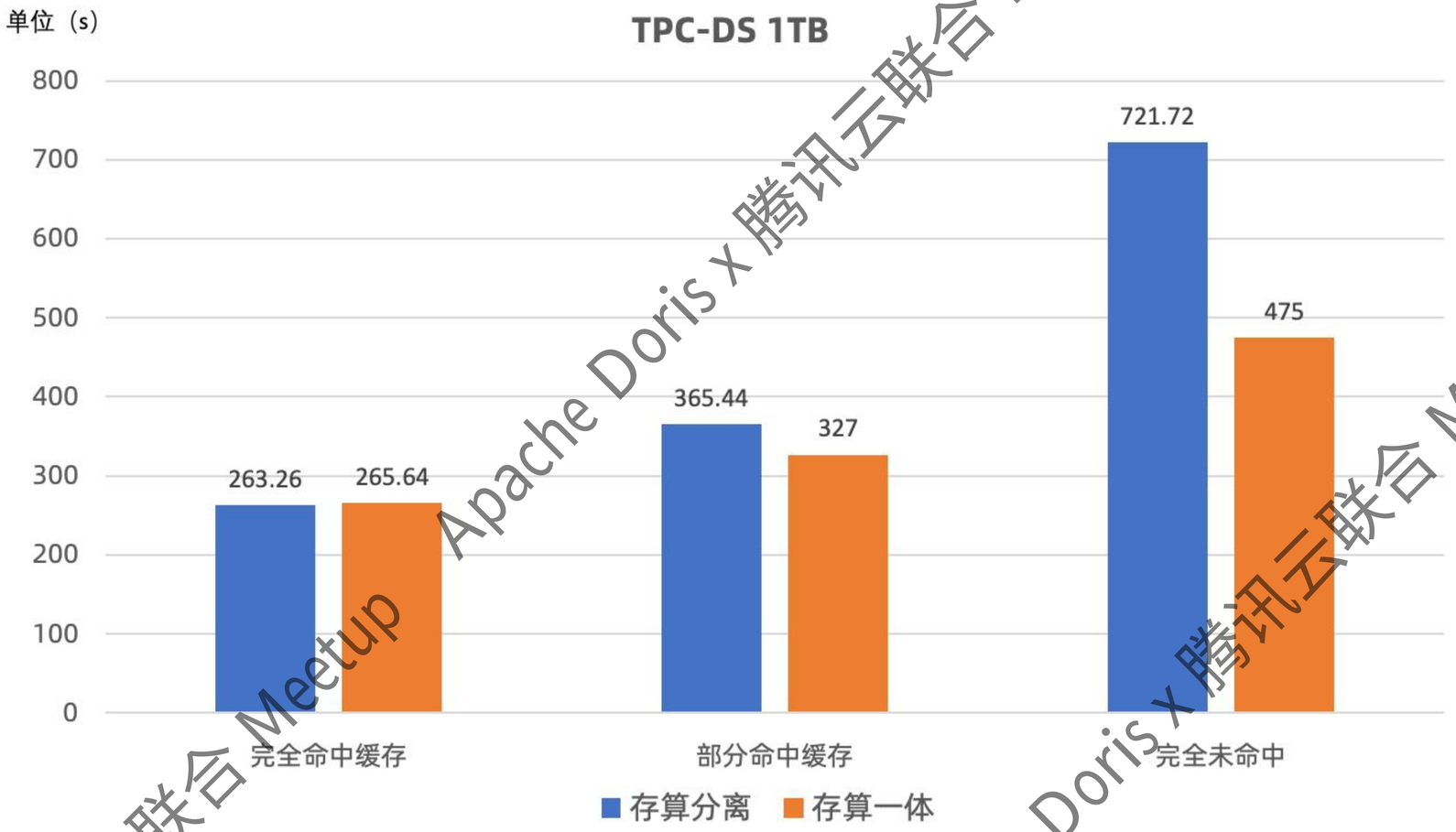


数据计算层-数据查询

- 多个 Computer Cluster 独立
- 不命中 Cache 时，从 S3 读数据
- 命中时，从本地 Cache 读数据
- 弹性资源大幅降低成本



查询性能对比

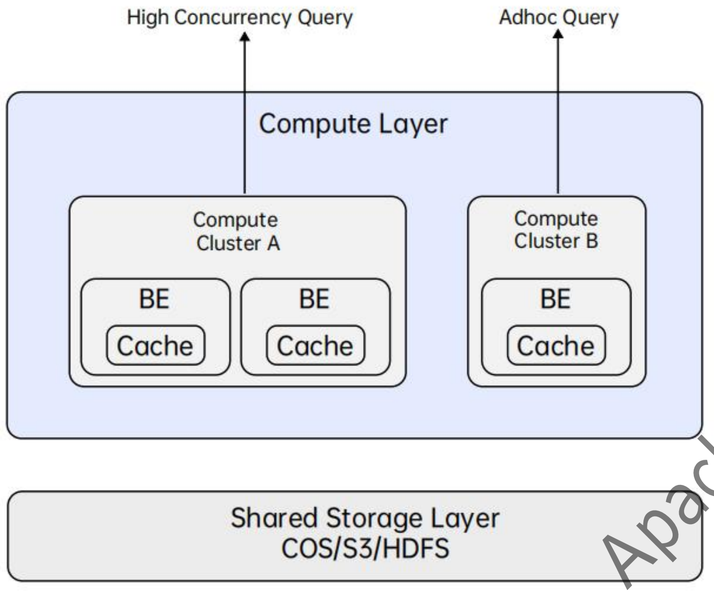


缓存层级

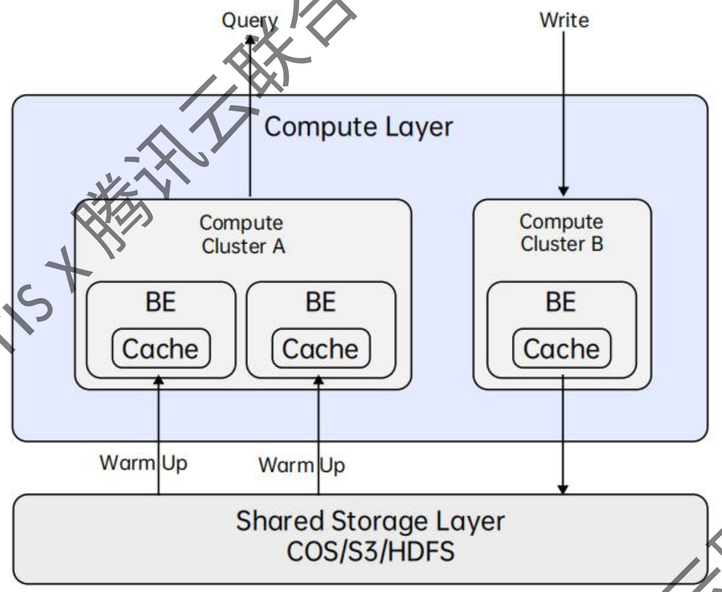
- Remote Storage
- Local File Cache
- System Page Cache
- Doris Page Cache

完全命中缓存时查询性能完全持平，部分命中缓存时有 10% 的性能损耗、随测试进行数据逐渐加载进缓存，性能随之提升；极端情况下（完全未命中任何缓存）性能损耗约 30%。

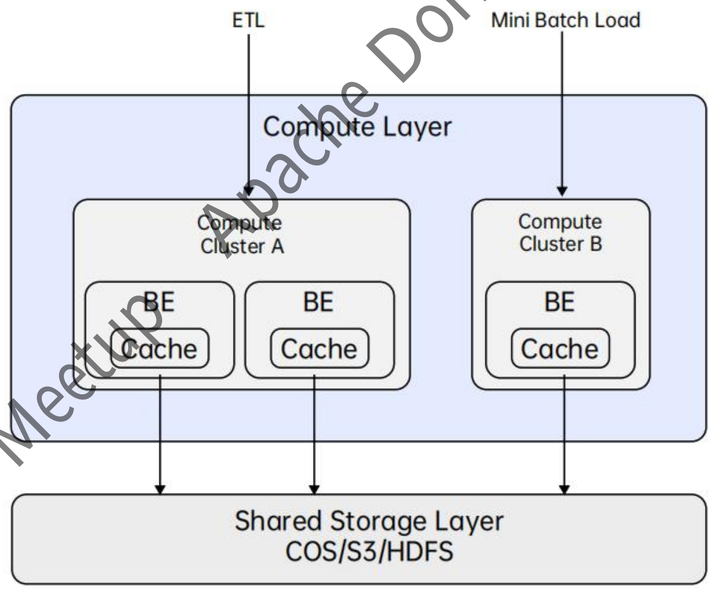
最佳实践



- **读读隔离**
- 高优查询和普通查询
- 高并发点查和即席分析



- **读写隔离**
- 实时同步
- 自动预热



- **写写隔离**
- 高频导入和ETL

目录

未来思考和规划

- **更精简的部署：**模块融合、自动化部署协助
- **更丰富的缓存策略：**更细粒度、更灵活的配置
- **更丰富的功能：**备份恢复、数据同步、数据共享
- **更高的隔离性：**compaction、schema change、导入完全隔离

Thanks !