

开放创新 构建和谐生态

Apache Hadoop Meetup 2021

北京
站

10月16日 9:30 - 18:00



字节跳动基于HUDI的实时数据湖 平台介绍

耿筱喻

字节跳动

大数据研发工程师

01

HUDI简介

02

应用场景

03

核心技术

04

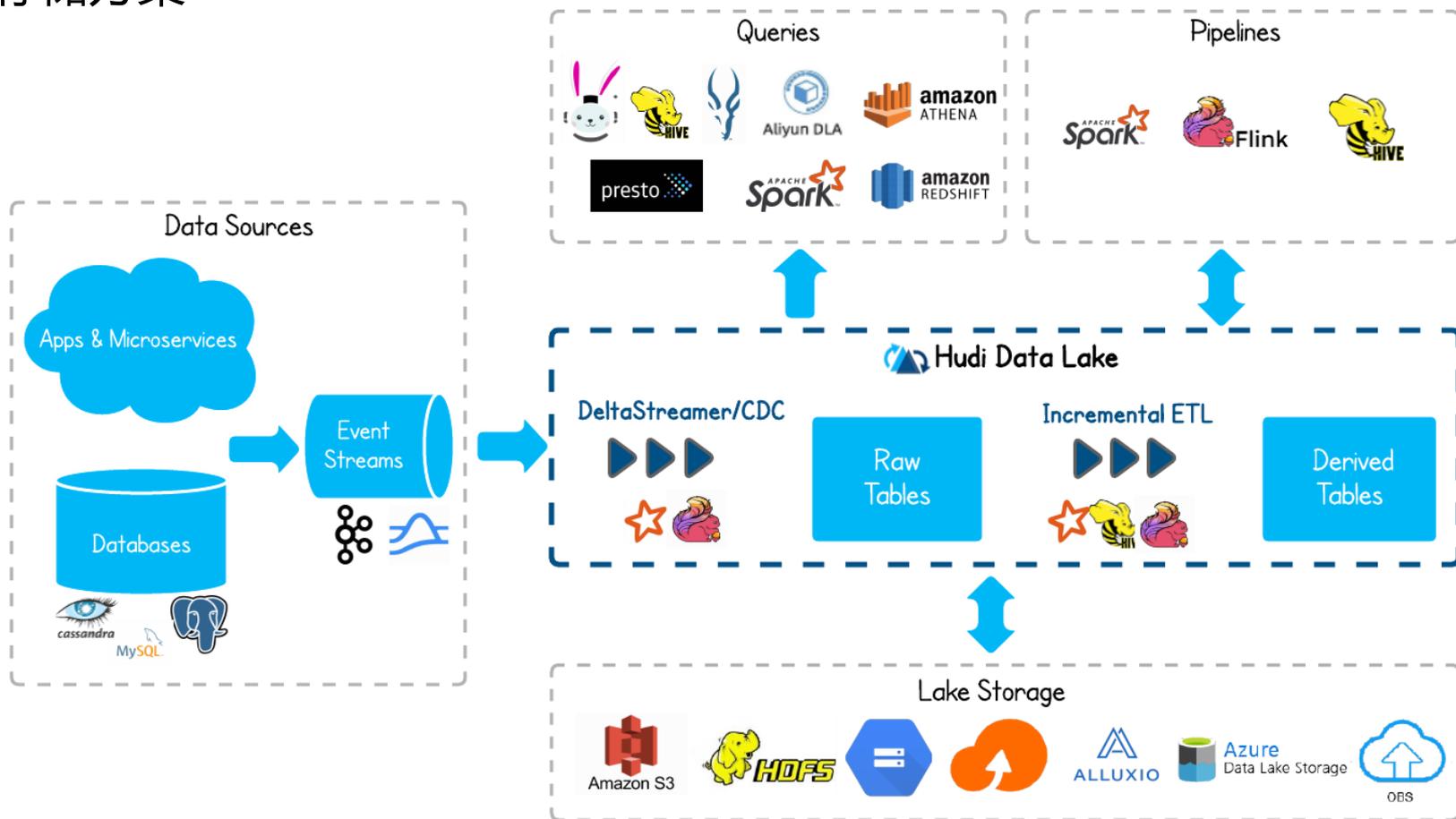
未来规划

01

HUDI简介

HUDI 简介

HUDI 作为一个流式数据湖平台，支持 ACID、支持增量更新与消费的存储方案



HUDI 基本概念

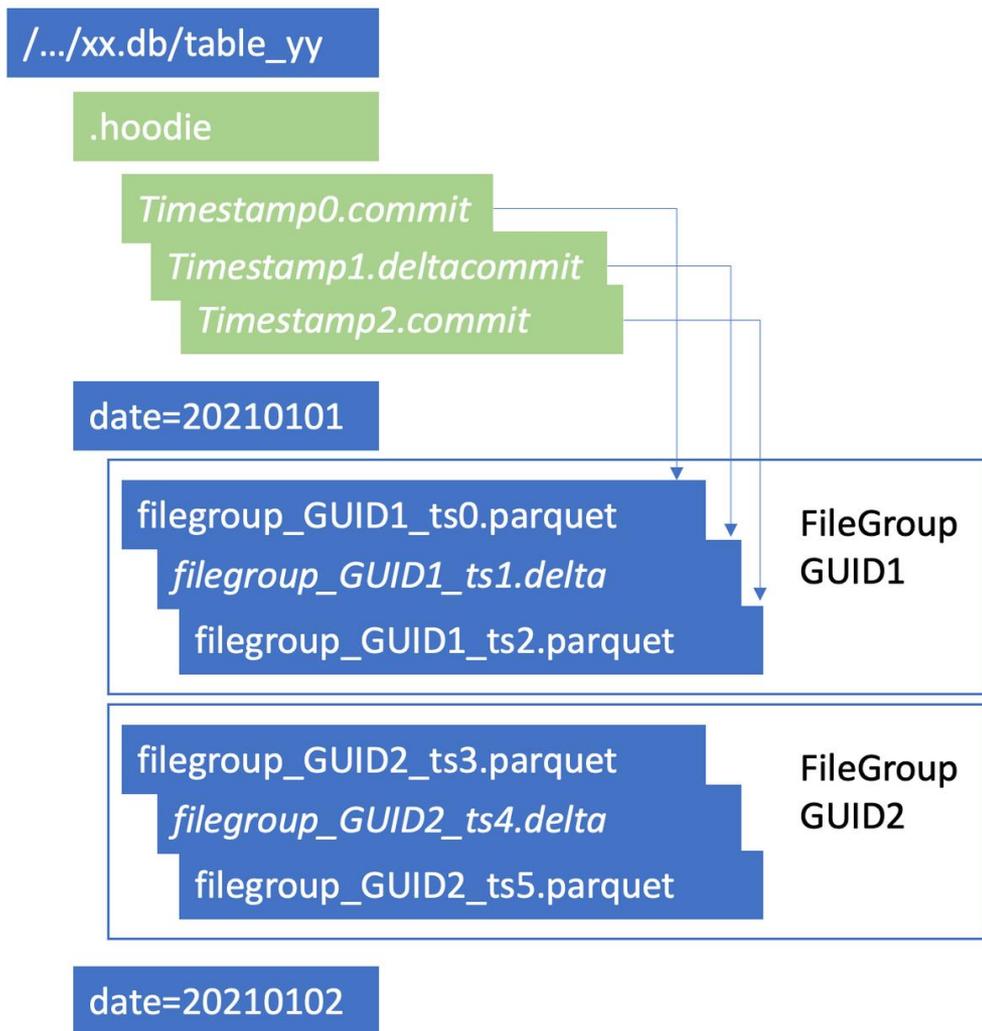
FileGroup

逻辑含义

- 单个分区内数据的水平切分单位
- 每个 Key 只存在于一个 File Group 中 (全局 or 分区)
- 逻辑上的文件概念, 有特定的 File ID

组成部分

- 所有归属于这个 File ID 下多个版本的文件 FileSlice 的集合
- Base File 为列存文件
- Log File 为行存文件





HUDI 表更新过程

Copy On Write 表

- 适用于离线批量更新场景
- 读取 File Group 中旧的 Base File，合并更新数据，生成新 Base File

Merge On Read 表

- 适用于实时高频更新场景
- 更新数据写入 File Group 的 Log 行存文件
- 读时 Merge，通过 Compaction 进行合并



HUDI 索引基本概念

索引用于定位更新数据所在的 File Group

- Bloom Filter Index

通过 Bloom Filter 判断 key 是否在已有的 File Group 中

- HBase Index

通过 HBase 记录 key 和文件的映射关系

- Bucket Index (RFC-29)

通过 key 的哈希值定位到 File Group



字节实时数据湖平台

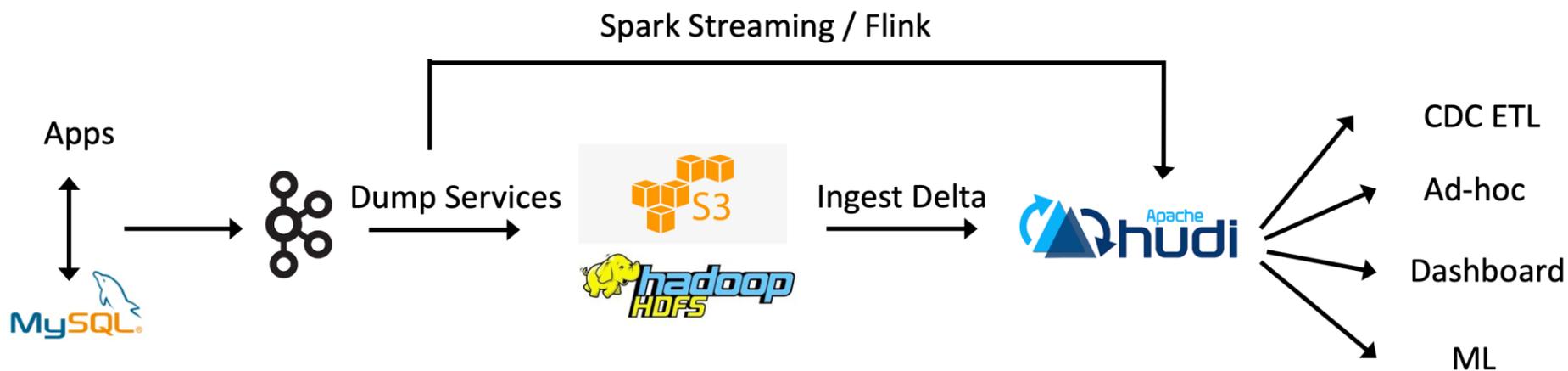
字节跳动基于 HUDI 通过秒级数据可见支持实时数仓，除了提供 HUDI 社区的所有功能外，还支持

- 基于数据湖的元数据管理系统
- 行列级别的并发更新
- Bucket Index，基于哈希的索引方式
- Append的模式

02

应用场景

Classic Hudi Pipeline



推荐场景

- 需要对表格存储做高效的OLAP查询
- 低成本批量添加特征列





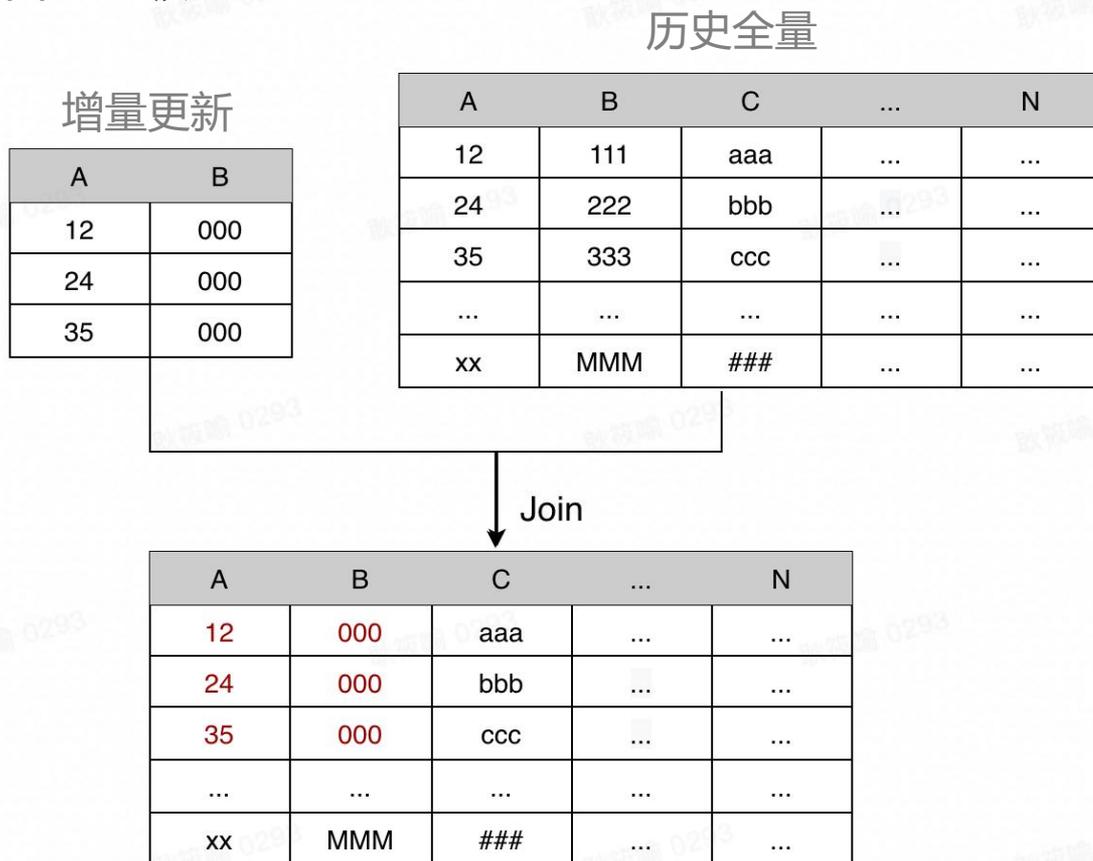
推荐场景

挑战

- 百 GB/s 的高吞吐近实时写入
- Schema 复杂，列数可到万级别，存在大量复杂类型
- 百万亿行数据的部分列的低成本更新
- 支持并发 Update

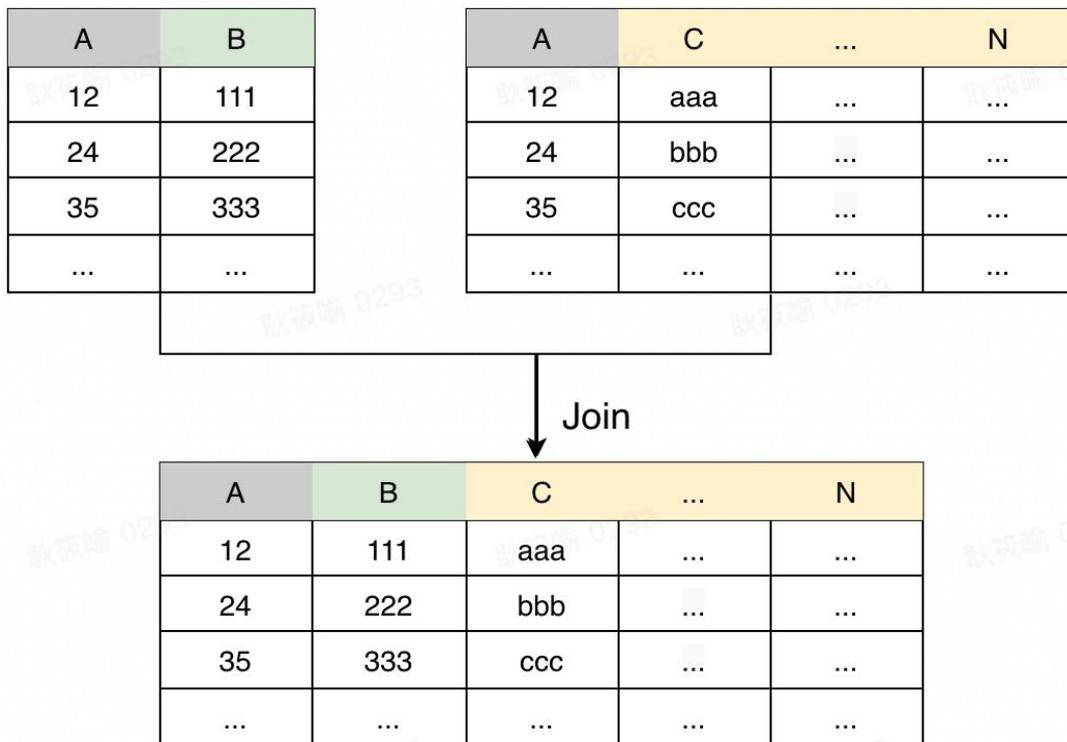
数仓场景

- 对历史全量数据进行部分行、列的更新
- 单表数据量百 PB 级



近实时数仓场景

- 多数据源实时导入，支持列拼接
- 单表数据规模几十 TB 量级

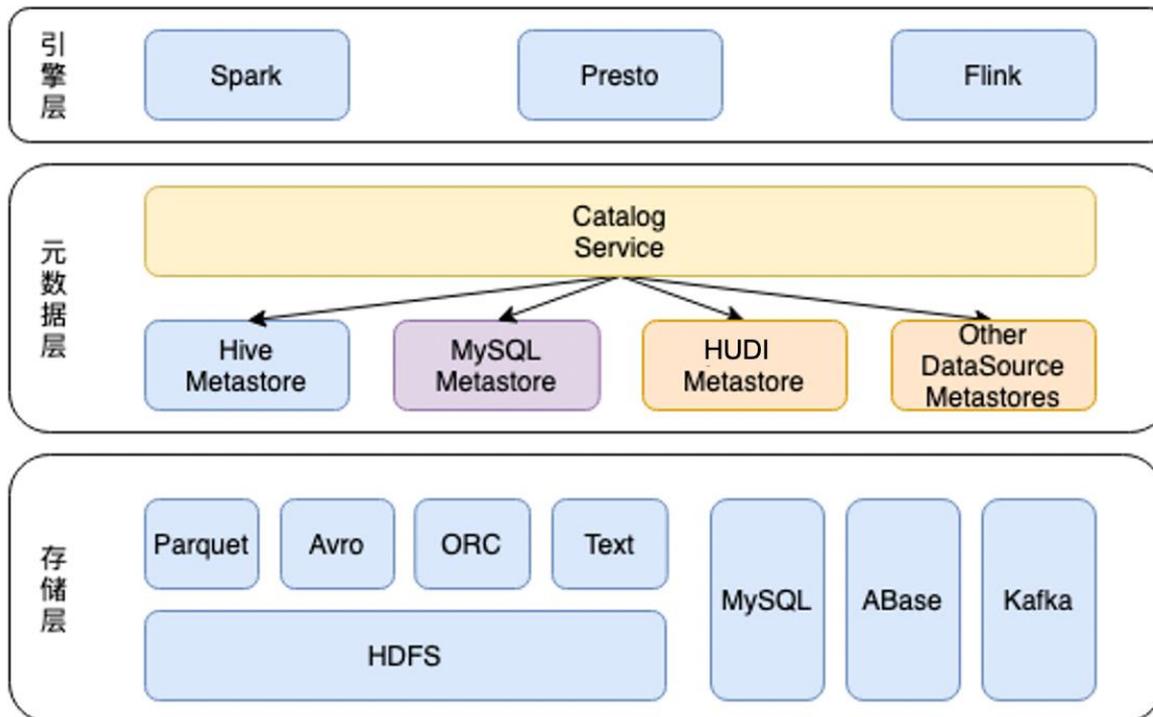


03

核心技术

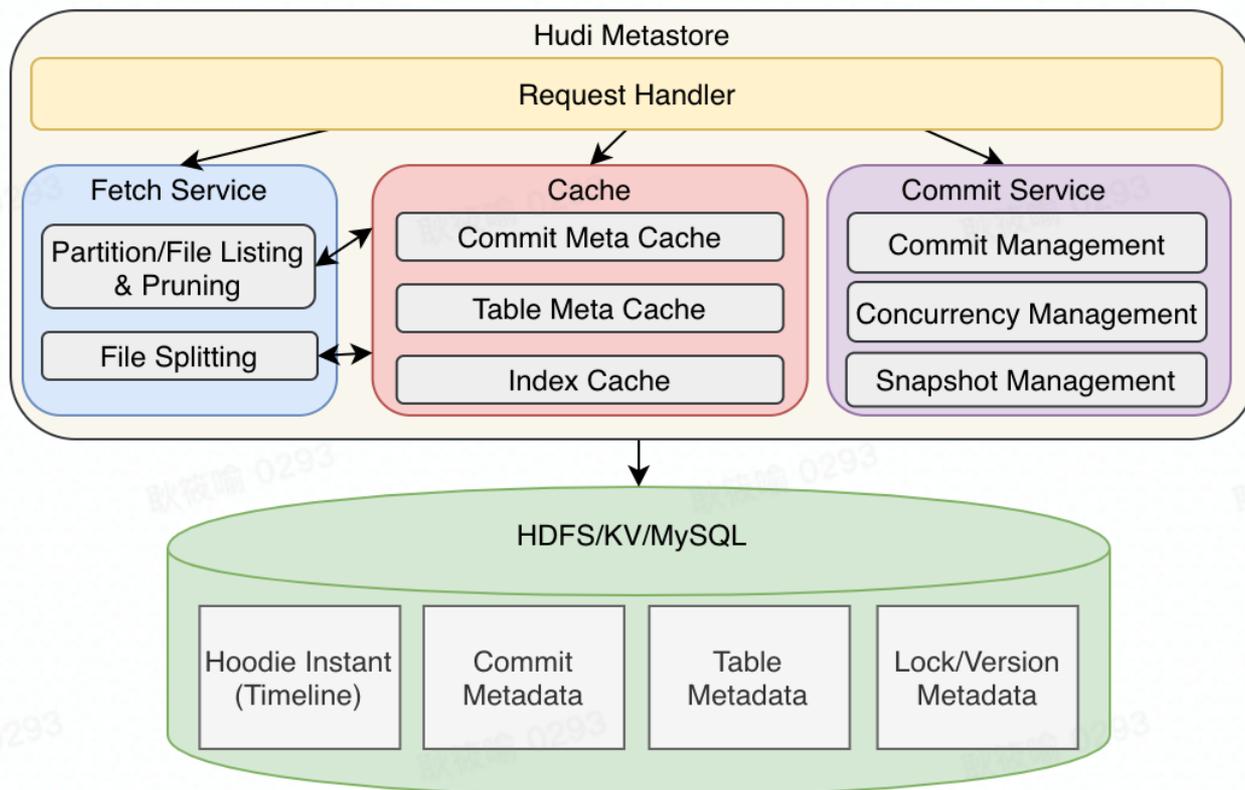
湖仓一体元数据服务

- 统一的元数据视图，与 Hive Metastore 完全兼容
- 无缝对接多个计算引擎 Spark / Presto / Flink
- 跨源查询分析能力，直接查询 MySQL, Abase(kv), Kafka
- 为数据湖定制的Metastore，支持高效数据更新



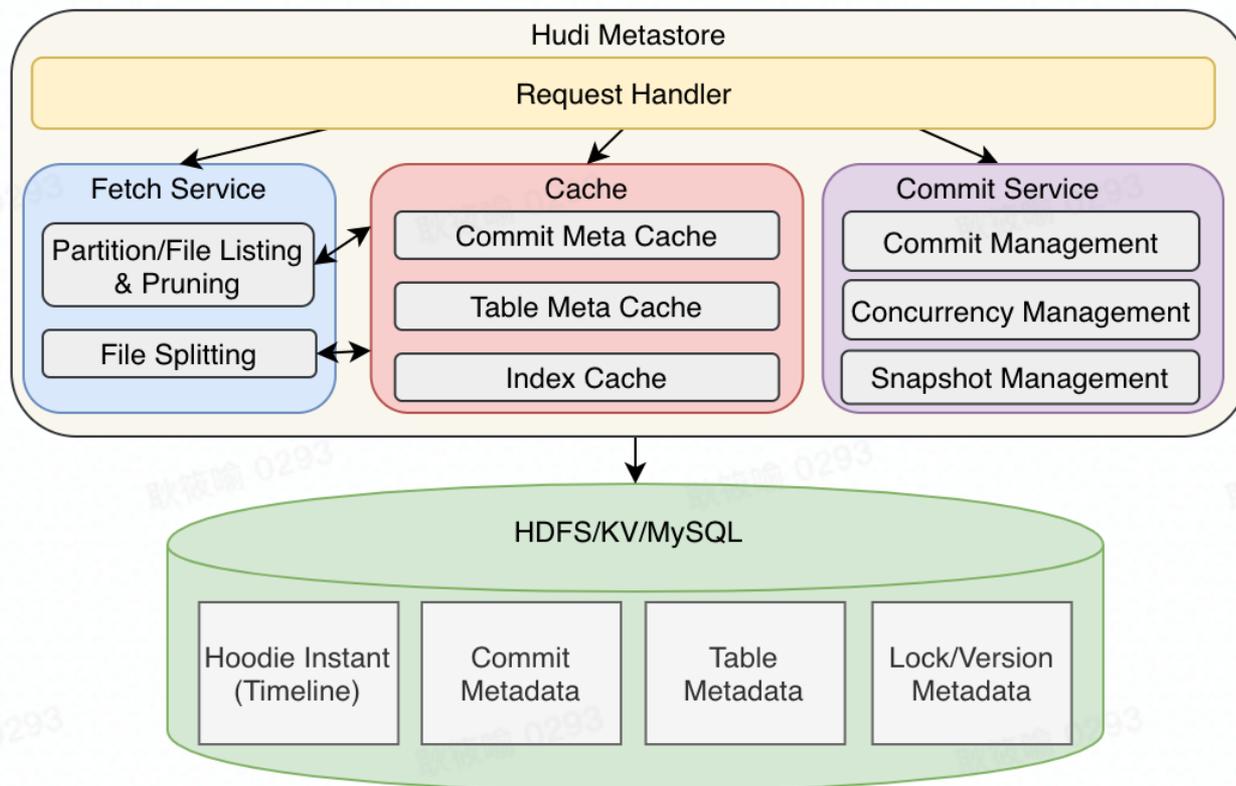
Hudi Metastore (HUDI RFC-36)

- 支持 Commit 形式的元数据管理，并支持并发更新
- 对最新元数据的 Snapshot 进行持久化，并支持高效查询
- 提供分区裁剪功能



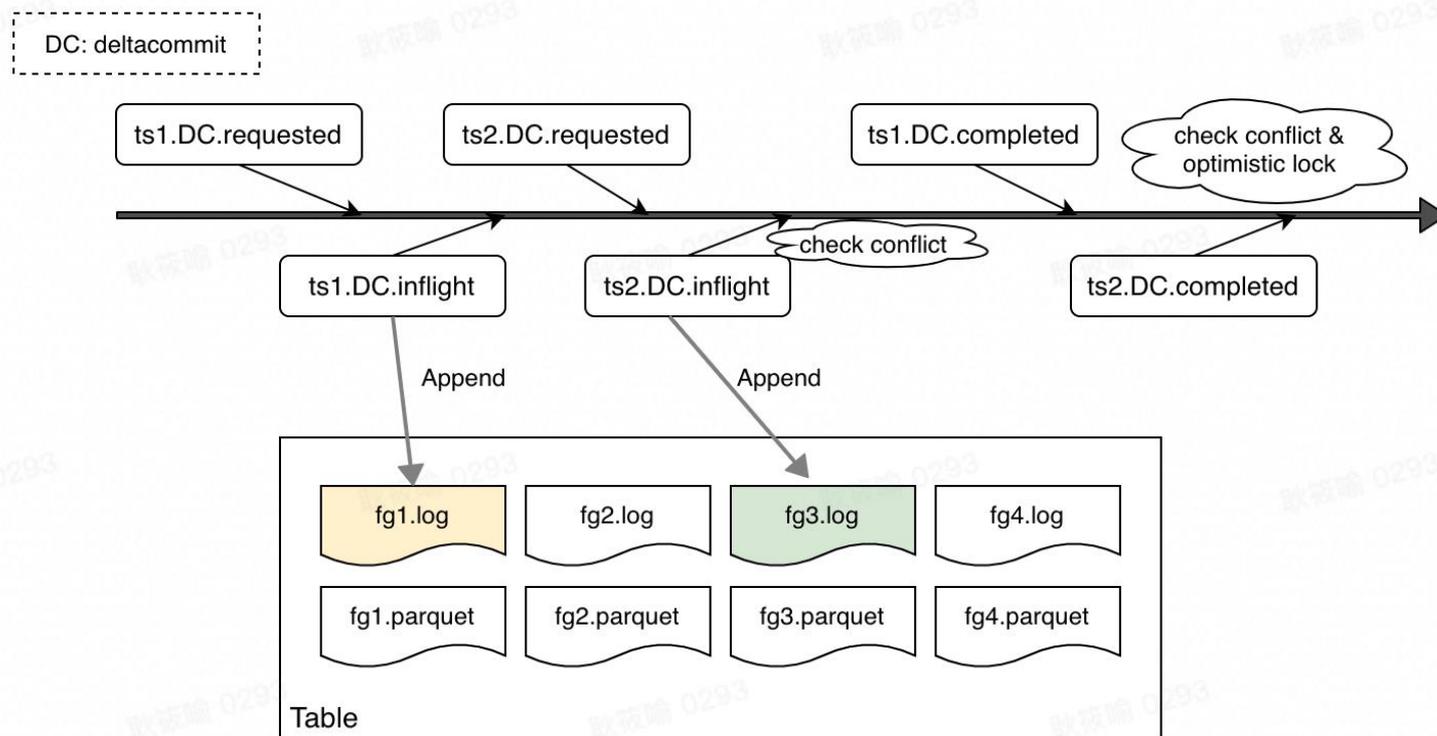
Hudi Metastore (HUDI RFC-36)

- 底层存储可插拔
- 轻量且易于扩展
- Hive Metastore 兼容



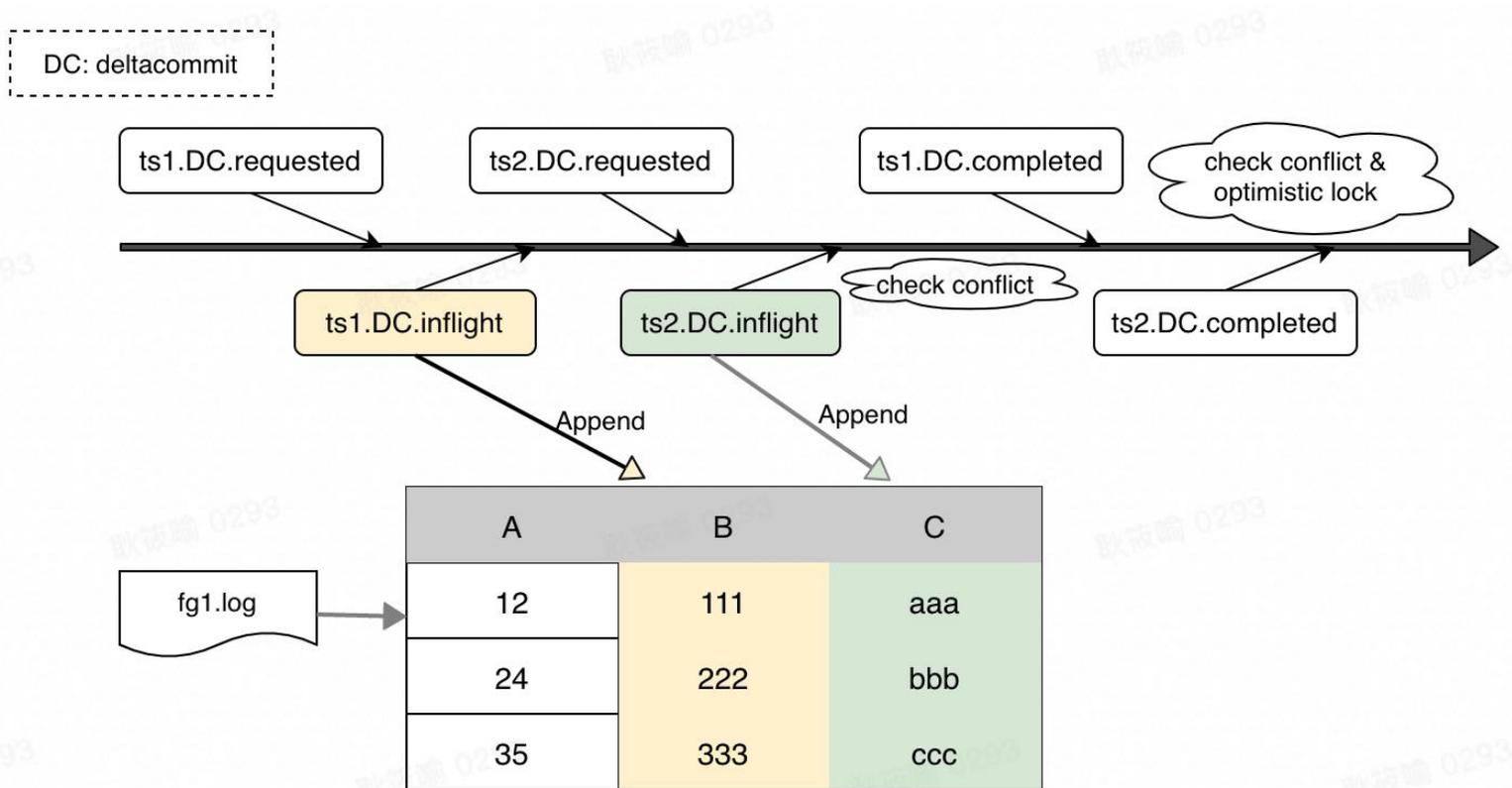
行列级并发写入

- 基于乐观锁的Timeline
- 灵活的行列冲突检查策略



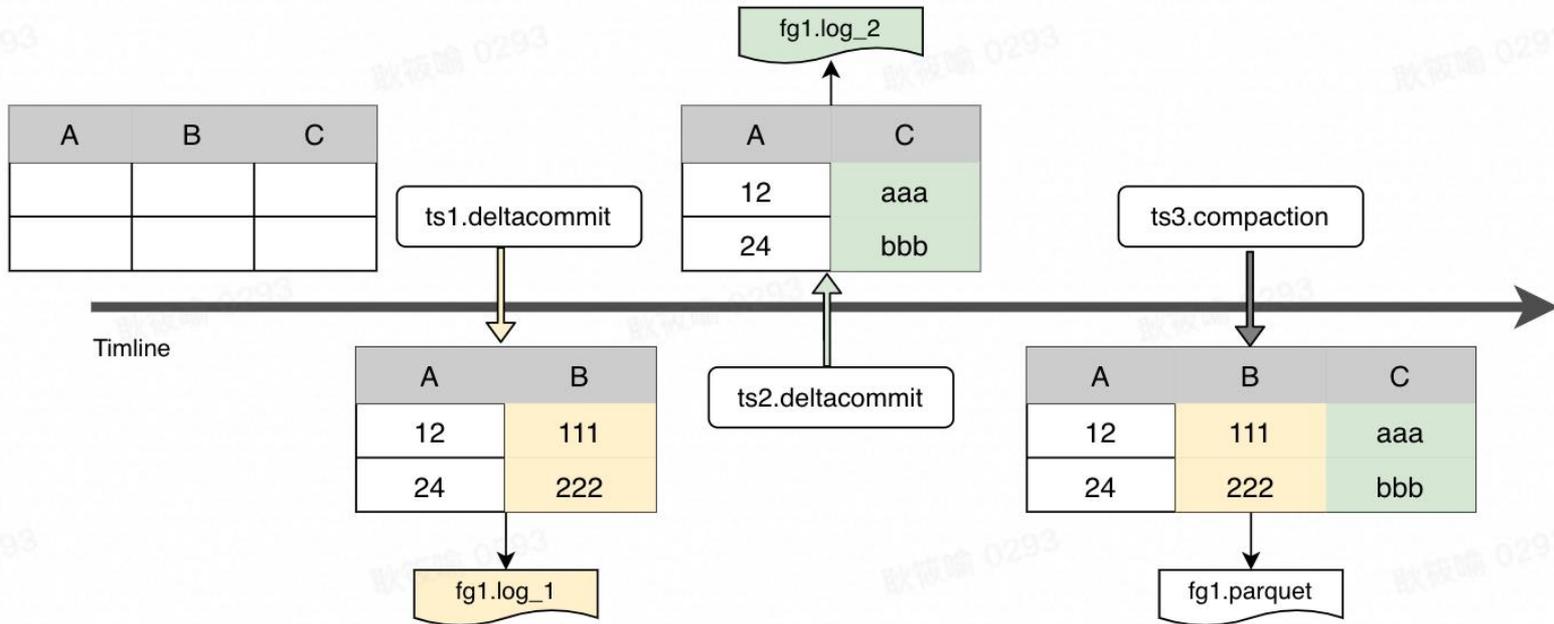
行列级并发写入

- 基于乐观锁的Timeline
- 灵活的行列冲突检查策略



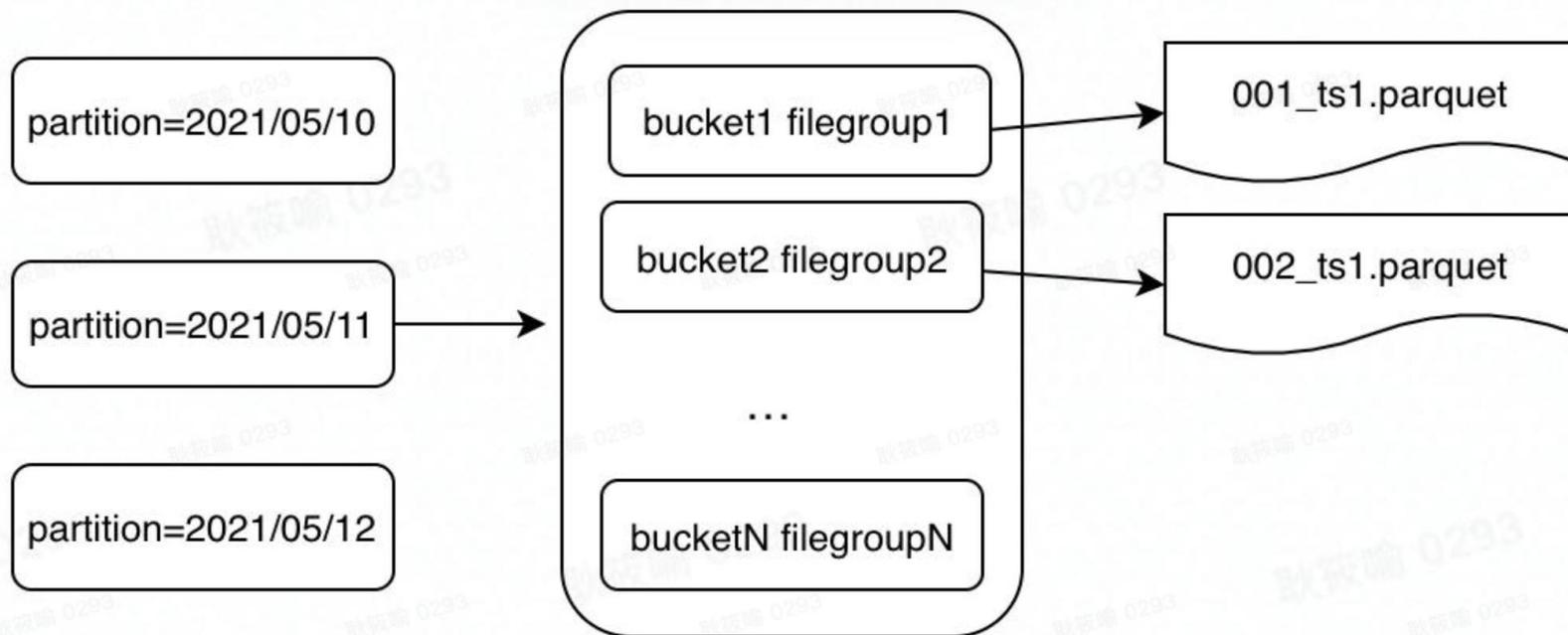
行列级并发写入

- 基于乐观锁的Timeline
- 灵活的行列冲突检查策略



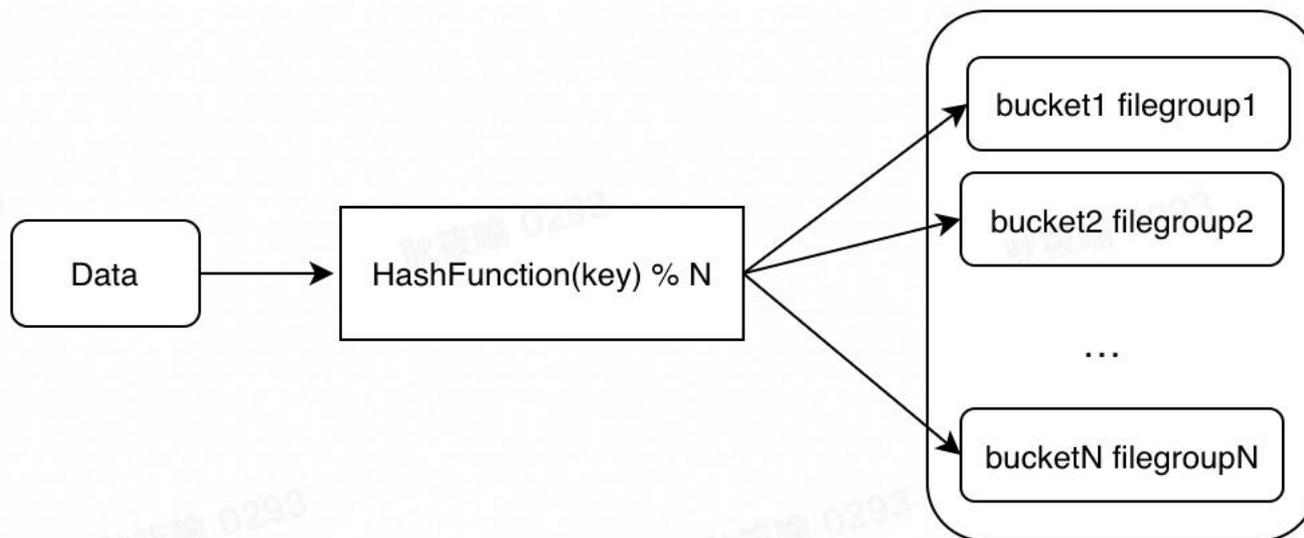
Bucket Index (HUDI RFC-29)

通过 key 的哈希值定位到 File Group, 提升导入实时性



Bucket Index (HUDI RFC-29)

通过 key 的哈希值定位到 File Group，提升导入实时性

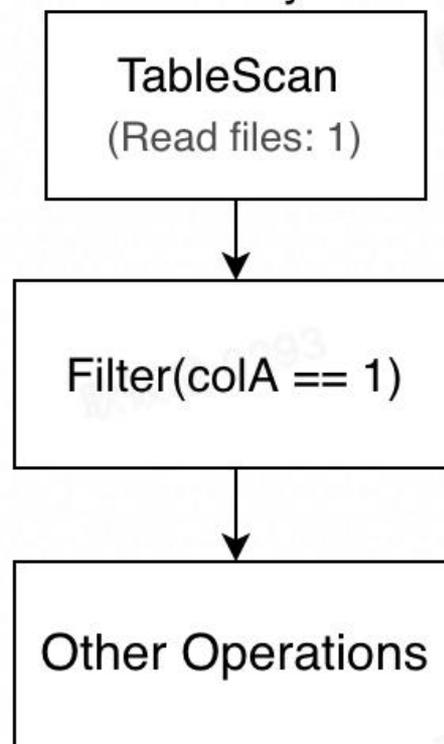


Bucket Index (HUDI RFC-29)

利用 Bucket 分布做查询优化

- **Bucket Pruning**
- Bucket Join

Table 1 (8 buckets)
bucketed by colA



Bucket Index (HUDI RFC-29)

利用 Bucket 分布做查询优化

- Bucket Pruning
- **Bucket Join**

Table 1
bucketed by colA

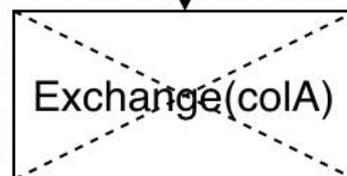
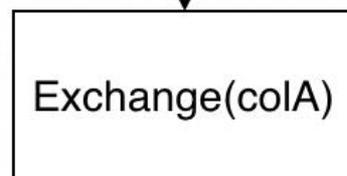
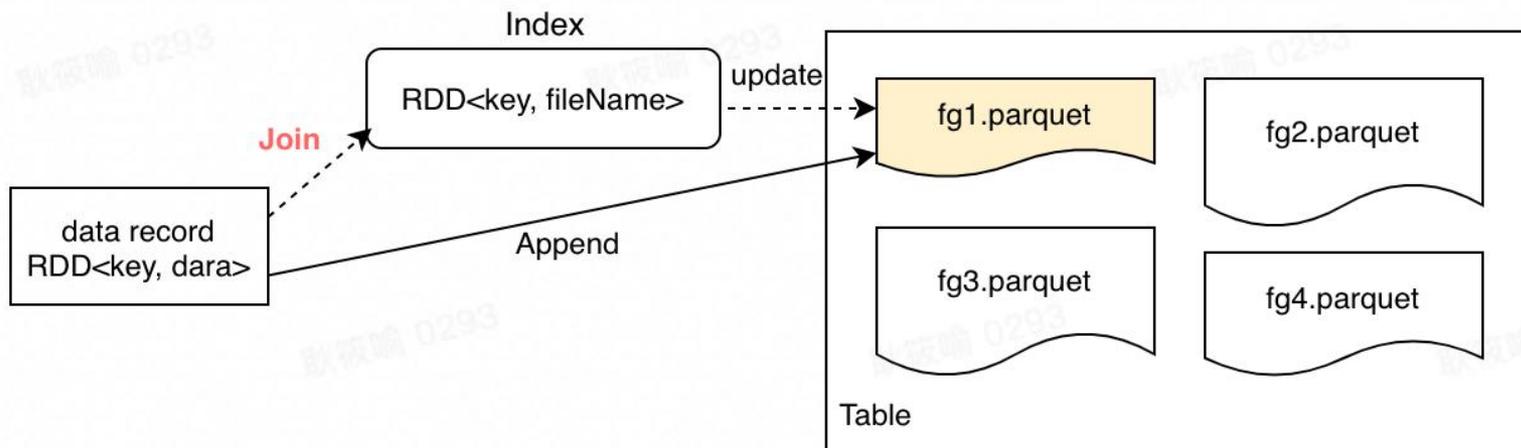


Table 2
with non-bucket



Append 模式支持

- 无需指定主键、比较列
- 支持日志场景
- Non Index



04

未来规划

未来规划

- 支持部分列更新下的完整 Bin Log 消费
- 可扩展哈希索引
- 存储服务化，数据秒级可见
- 基于Merge Tree的文件分布

谢谢



字节跳动数据引擎团队持续招人～

The background features a dark blue gradient with two large, sweeping, glowing arcs. The left arc is primarily blue with some orange highlights, while the right arc is primarily orange with some blue highlights. Both arcs are composed of numerous thin, parallel lines that create a sense of motion and depth. In the lower portion of the image, there is a faint, scattered pattern of binary code (0s and 1s) in a light blue color.

THANK YOU