УДК 004.052

АРХИТЕКТУРА СИСТЕМЫ ВЕРСИОННОГО КЕШИРОВАНИЯ МЕТАДАННЫХ ДЛЯ APACHE ICEBERG И DELTA LAKE

Лазарев Д. С. (ИТМО)
Научный руководитель – доцент Королёва Ю. А.
(ИТМО)

Введение. В lakehouse-архитектурах метаданные являются критическим узлом: вычислительные движки (Spark/Trino/Presto) обращаются к ним при планировании запросов, выборе партиций и файлов, а также при выполнении time travel. В Apache Iceberg [1] и Delta Lake [2] метаданные версионны (snapshot id / table version), но их чтение становится дорогим из-за цепочек манифестов (Iceberg) и транзакционного лога с checkpoint-ами (Delta). Это приводит к росту latency на стадии компиляции и планирования запросов, а также к увеличению нагрузки на объектное хранилище (LIST/HEAD/GET метафайлов). Целью работы стало проектирование архитектуры системы, которая ускоряет доступ к метаданным Iceberg/Delta за счёт версионного кеширования и событийной инвалидации, сохраняя корректность и воспроизводимость чтения.

Основная часть. Архитектура решения строится вокруг следующих компонентов и принципов.

Источники и нормализация метаданных. Система подключается к Iceberg Catalog/REST/Hive-metastore как «указателю» на текущее состояние и к Delta _delta_log как источнику версий. Через адаптеры метаданные приводятся к единому «план-ориентированному снимку»: схема, текущая версия, список файлов данных/партиций и минимальная статистика для pruning.

Версионные ключи кеша. Кэш хранит метаданные не «в целом для таблицы», а для конкретной версии: iceberg:{table}:{snapshotId} и delta:{table}:{version}/{checkpoint}. Это позволяет возвращать строго согласованное состояние и безопасно обслуживать time travel и конкурентные обновления.

Многоуровневое хранение (L1/L2/L3). L1 — in-memory горячие таблицы/последние версии для минимальной задержки. L2 — распределённый кэш (Redis/KeyDB) для шаринга между экземплярами и failover. L3 — персистентность (WAL/снапшоты состояния кэша) для быстрого восстановления после рестартов и защиты от cold start.

Инвалидация и обновление: события + TTL. Основной режим — событийная инвалидация: при commit новой версии таблицы кэш переводится на новую запись версии. TTL используется как страховка, если событие потерялось. Дополнительно применяется stale-while-revalidate: клиент получает последнее корректное значение сразу, а обновление выполняется в фоне [2].

Планировщик подзагрузки и защита источников. Для популярных таблиц система выполняет prefetch последних снапшотов/manifest list/checkpoint и применяет дедупликацию запросов (single-flight), чтобы при всплеске нагрузки не создавать лавину обращений к каталогу и объектному хранилищу [3].

API-слой и интеграция. Сервис предоставляет REST/gRPC API: получить метаданные таблицы на версии N, получить список файлов для фильтра/партиций, проверить актуальную версию. Интеграция реализована как прокси-слой между query engine и источниками метаданных, а также как библиотека-клиент для движков.

Наблюдаемость и эксплуатация. Собираются метрики cache hit/miss, latency по операциям (catalog/read manifests/read log), скорость появления новых версий, размер кэша, частота инвалидаций. Для HA используется несколько инстансов сервиса и репликация L2-кэша [4].

Выводы. Спроектированная архитектура сочетает преимущества lakehouse-подхода (версионирование и воспроизводимость состояния) с производительностью

инфраструктурного кэша. Версионные ключи устранили риск «случайно устаревших» метаданных, а событийная инвалидация перенесла обновления с модели TTL на модель «по факту появления новой версии». Многоуровневое хранение снизило latency для горячих таблиц и обеспечило устойчивость при рестартах. В результате достигнуто сокращение времени планирования запросов и количества обращений к объектному хранилищу за метафайлами без нарушения согласованности Iceberg/Delta.

Список использованных источников:

1. Apache Iceberg Table Spec [Электронный ресурс] // Apache Software Foundation. URL: https://iceberg.apache.org/spec/ (дата обращения: 13.02.2026).
2. Armbrust M. et al. Delta Lake: High-Performance ACID Table Storage over Cloud Object Stores // Proceedings of the VLDB Endowment. – 2020. – Vol. 13, No. 12. – P. 3411–3424. URL: https://www.vldb.org/pvldb/vol13/p3411-armbrust.pdf (дата обращения: 13.02.2026).
3. Armbrust M. et al. Lakehouse: A New Generation of Open Platforms that Unify Data Warehousing and Advanced Analytics // Proceedings of CIDR 2021. URL: https://www.cidrdb.org/cidr2021/papers/cidr2021_paper17.pdf (дата обращения: 13.01.2026).
4. Kleppmann M. Designing Data-Intensive Applications. – O’Reilly Media, 2017. – 616 p.