Минимальная поддержка транзакций для множества вставок/чтений

Кузнецов Максим Анатольевич
Руководитель ВКР: Миловидов Алексей Николаевич

ClickHouse

Столбцовая система управления базами данных (СУБД) для онлайн обработки аналитических запросов (OLAP).

Особенности:

- Параллельное выполнение на нескольких ядрах
- Распределенное выполнение на нескольких серверах
- Поддержка репликации данных

Транзакция

Группа последовательных операций с базой данных, которая представляет собой логическую единицу работы с данными.

Транзакция может быть выполнена либо целиком и успешно, либо не выполнена вообще.

Актуальность

ClickHouse популярная система, которая используется многими компаниями

Транзакции позволяют избавиться от несогласованности данных в аналитических отчетах

ACID

- **A**tomicity — Атомарность

  Транзакция не может быть зафиксирована в системе частично
- **C**onsistency — Согласованность

  Каждая успешная транзакция сохраняет согласованность данных
- **I**solation — Изолированность

  Параллельные транзакции не должны оказывать влияния на результат
- **D**urability — Стойкость

  Результаты успешно выполненной транзакции сохраняются в системе навсегда

Подходы к реализации транзакций

Свойство стойкости:

- Упреждающая журнализация (write-ahead logging)
- Теневой механизм (shadow paging)

Свойства атомарности и изолированности:

- Блокировки (locking)
- MVCC (multiversion concurrency control)

Обзор существующих решений

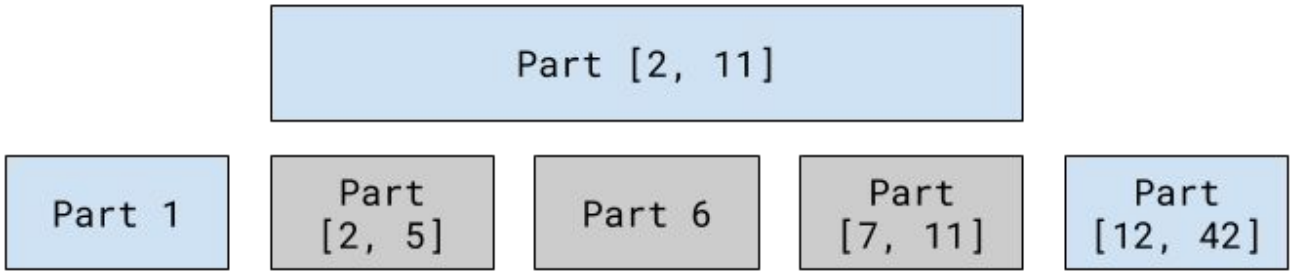
- Exasol
- Postgre
- MySQL
- MonetDB

MergeTree

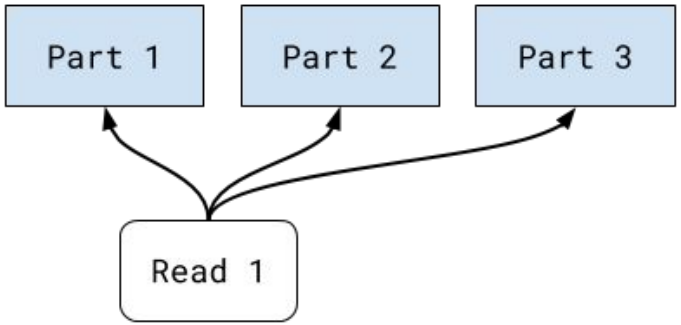
- Данные сортированы по первичному ключу
- Эффективны для большого количества чтений
- Эффективны для чтений из диапазона первичного ключа
- Эффективны для редких вставок

Хранение данных

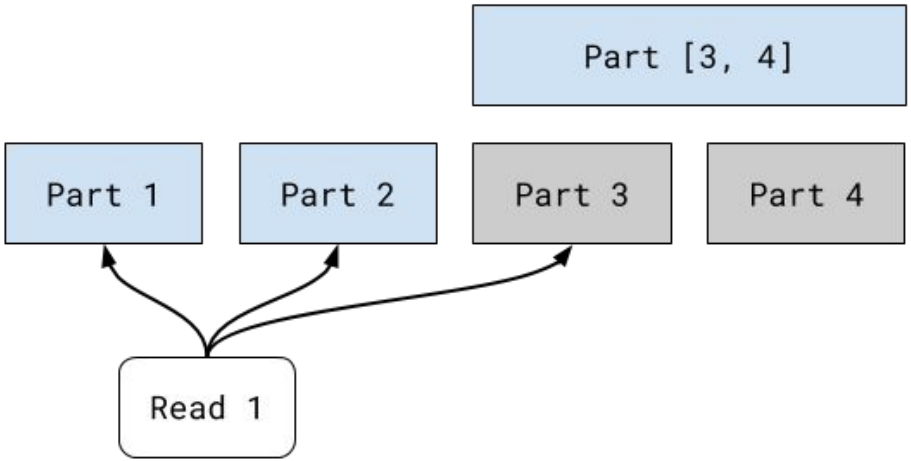
Слияние данных



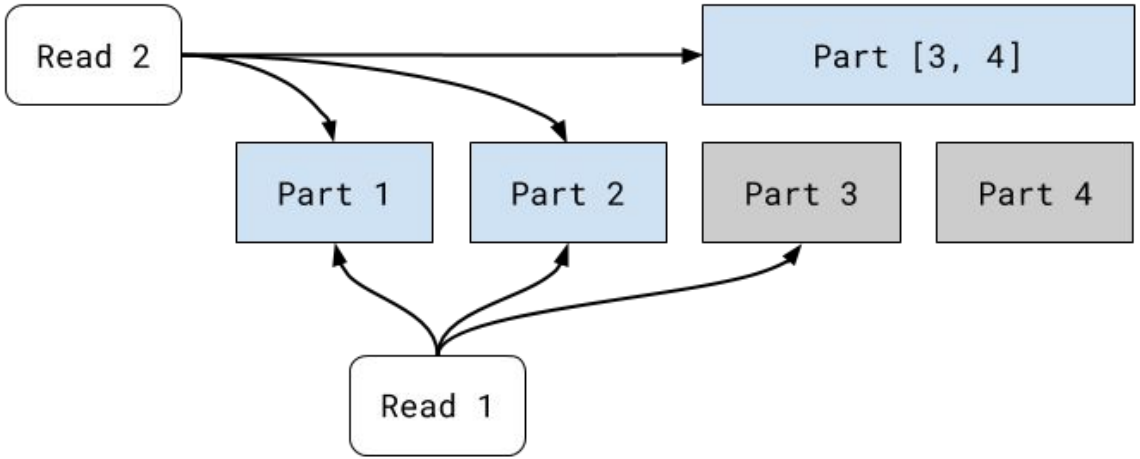
Чтение данных



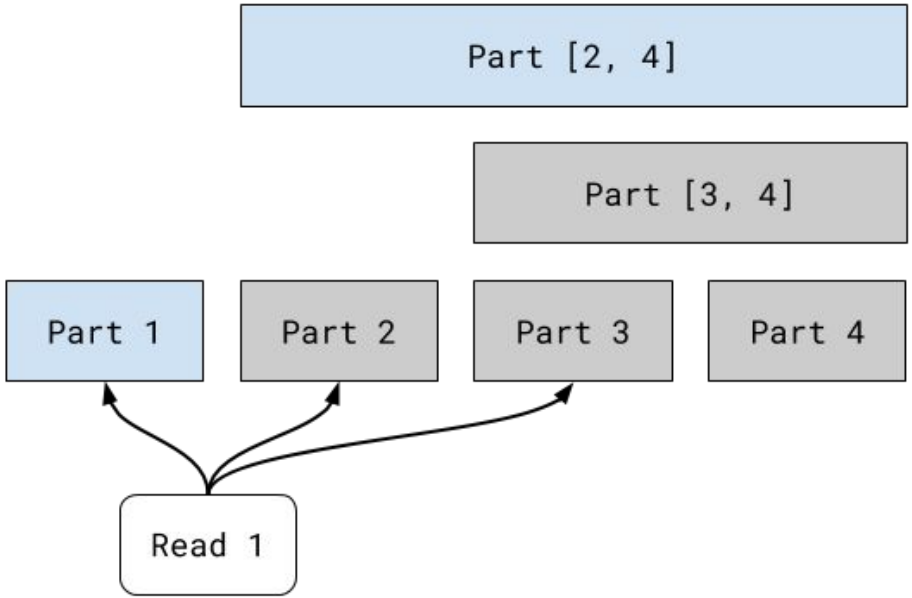
Пример



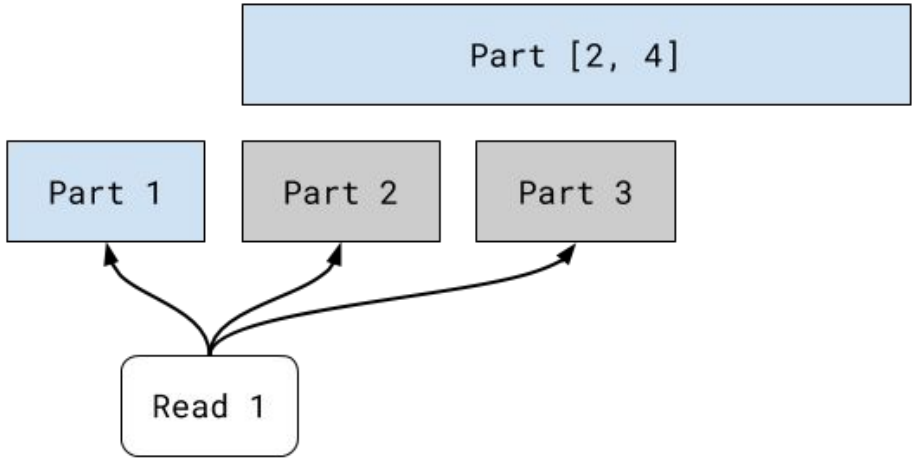
Пример



Пример



Пример

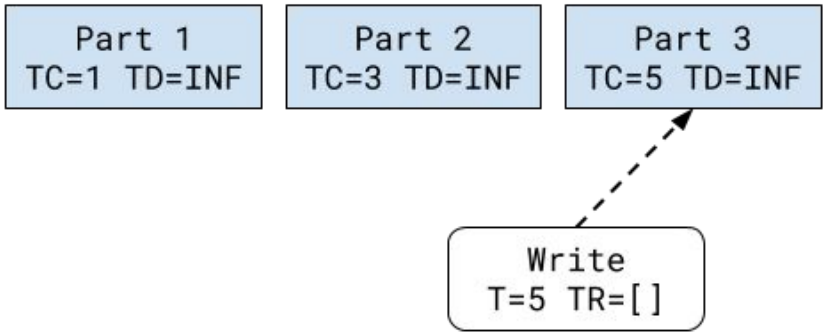


Пример

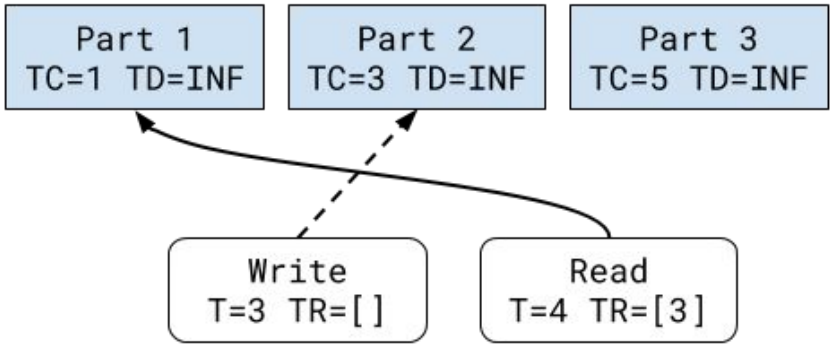
Алгоритм транзакций

- Каждый запрос имеет свой timestamp - монотонно возрастающее число
- Каждая часть теперь имеет 2 новых поля:
  - timestamp_created - timestamp операции, которая создала эту часть
  - timestamp_deleted - timestamp операции, которая удалила эту часть
- Каждый запрос дополнительно имеет еще два поля:
  - min_timestamp - минимальный timestamp среди всех выполняющихся в данный момент операций
  - timestamps_running - массив timestamp’ов всех операций, которые выполняются на момент получения массива

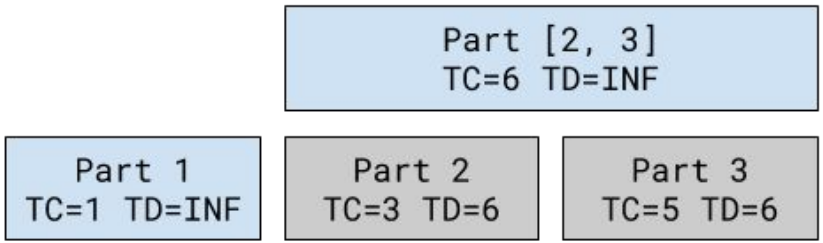
Запись



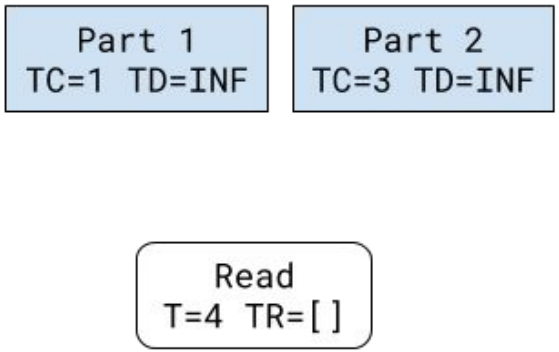
Чтение


Part 1
TC=1 TD=INF
Part 2
TC=3 TD=INF
Part 3
TC=5 TD=INF
Write
T=3 TR=[]
Read
T=4 TR=[3]

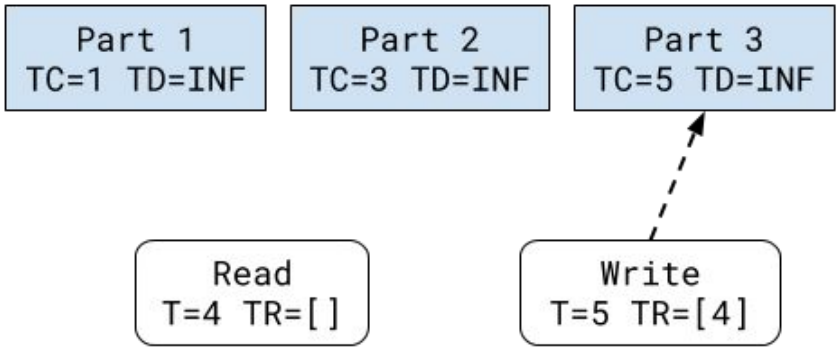
Слияние



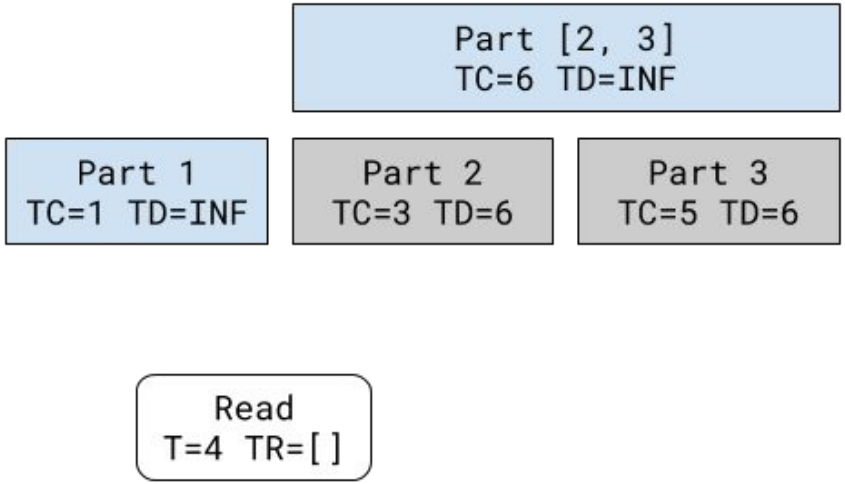
Пример



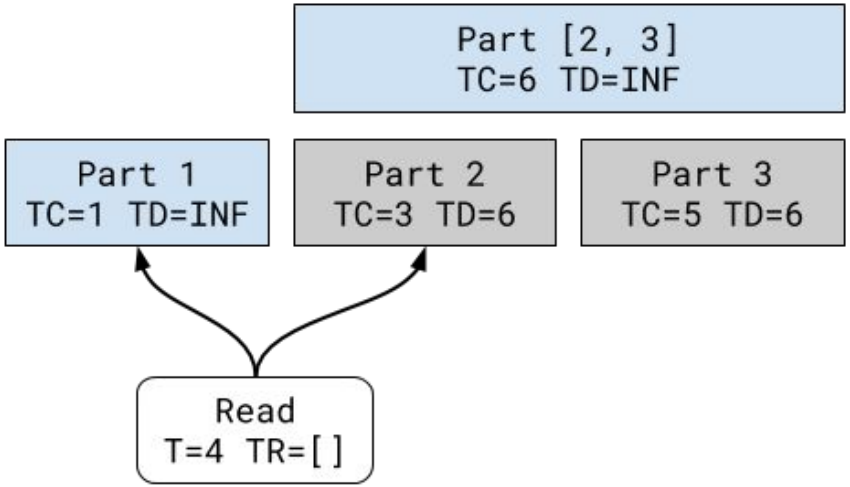
Пример


Part 1
TC=1 TD=INF
Part 2
TC=3 TD=INF
Part 3
TC=5 TD=INF
Read
T=4 TR=[]
Write
T=5 TR=[4]

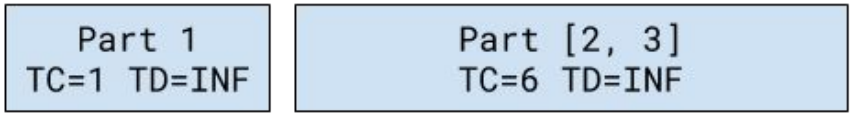
Пример


Part [2, 3]
TC=6 TD=INF
Part 1
TC=1 TD=INF
Part 2
TC=3 TD=6
Part 3
TC=5 TD=6
Read
T=4 TR=[]

Пример


Part [2, 3]
TC=6 TD=INF
Part 1
TC=1 TD=INF
Part 2
TC=3 TD=6
Part 3
TC=5 TD=6
Read
T=4 TR=[]

Пример



ReplicatedMergeTree

- ZooKeeper
- Последовательные ноды (sequential node) для получения timestamp’ов
- Преждевременное журналирование с использованием существующей ноды “/logs/” или новой ноды “/transactions/”

Результаты

- Рассмотрены существующие решения
- Разработан алгоритм, который:
  - Легко встраивается в существующую архитектуру
  - Эффективен
  - Удовлетворяет требованиям ACID