Выпускная квалификационная работа

Wait-free каталог баз данных в ClickHouse

Токмаков Александр Викторович, группа БПМИ165

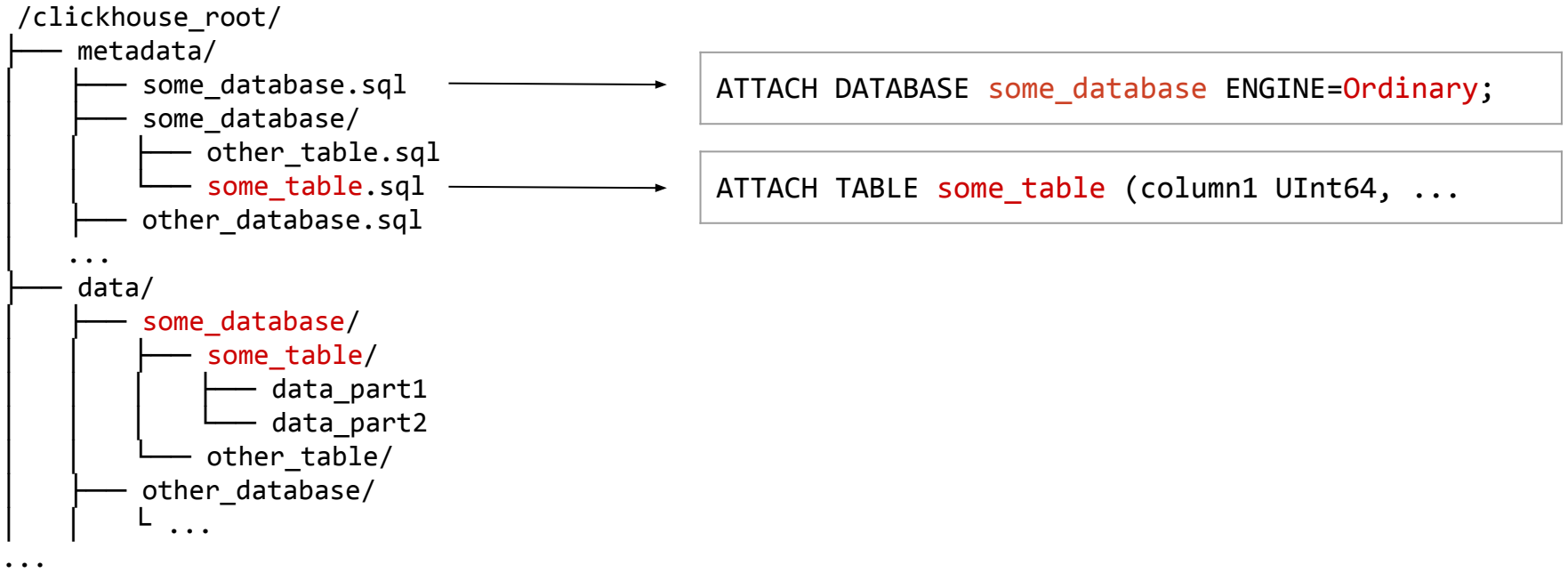
Научный руководитель: Миловидов Алексей Николаевич

Предметная область

- ClickHouse - аналитическая СУБД
- Каталог баз данных отвечает за хранение метаданных
- DDL-запросы требуют синхронизации

```
CREATE TABLE [IF NOT EXISTS] db.table …;
ATTACH TABLE [IF NOT EXISTS] db.table …;
DROP TABLE [IF EXISTS] db.table;
DETACH TABLE [IF EXISTS] db.table;
RENAME TABLE db.table1 TO db.table2;
```

Устройство каталога баз данных в ClickHouse



Синхронизация запросов

DDLGuard:

- С именем таблицы ассоциирован `std::mutex`
- DDL-запросы блокируют имя таблицы

Синхронизация запросов

DDLGuard:

- С именем таблицы ассоциирован `std::mutex`
- DDL-запросы блокируют имя таблицы

RWLock:

- Относится к объекту таблицы
- SELECT и INSERT блокируют таблицу на чтение
- DROP (DETACH) и RENAME блокируют таблицу на запись

Синхронизация запросов

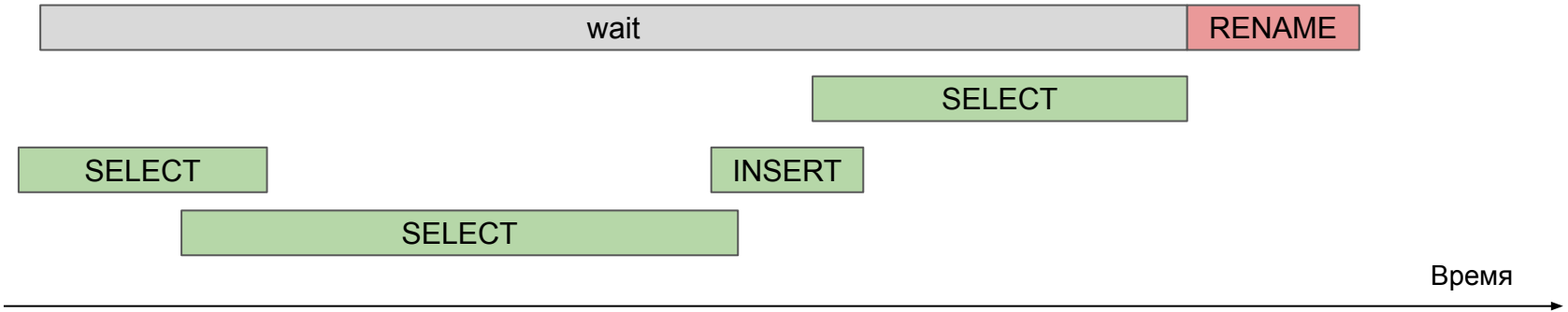
Специфическая реализация RWLock

- рекурсивный; блокируется запросом, а не потоком

Синхронизация запросов

Специфическая реализация RWLock

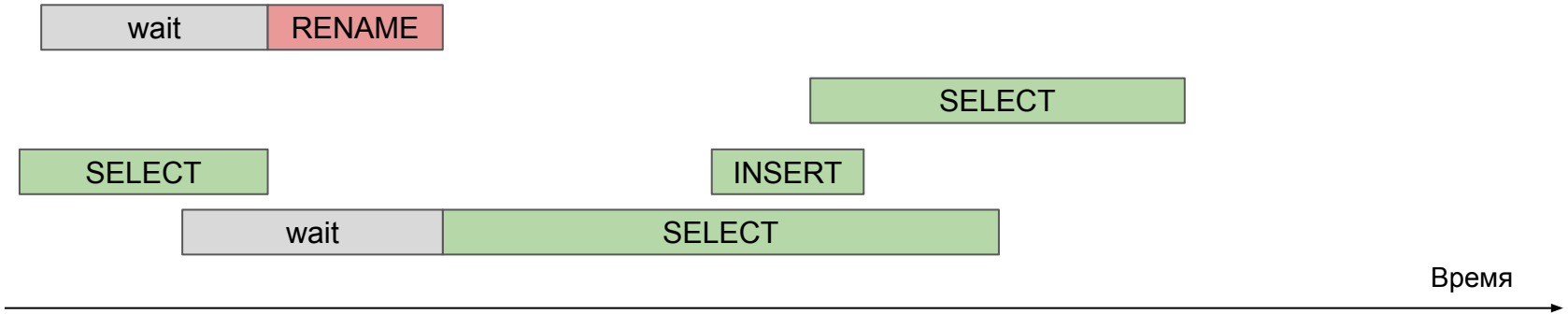
- рекурсивный; блокируется запросом, а не потоком



Синхронизация запросов

Специфическая реализация RWLock

- рекурсивный; блокируется запросом, а не потоком
- честный



Синхронизация запросов

Специфическая реализация RWLock

- рекурсивный; блокируется запросом, а не потоком
- честный
- механизм предотвращения взаимных блокировок

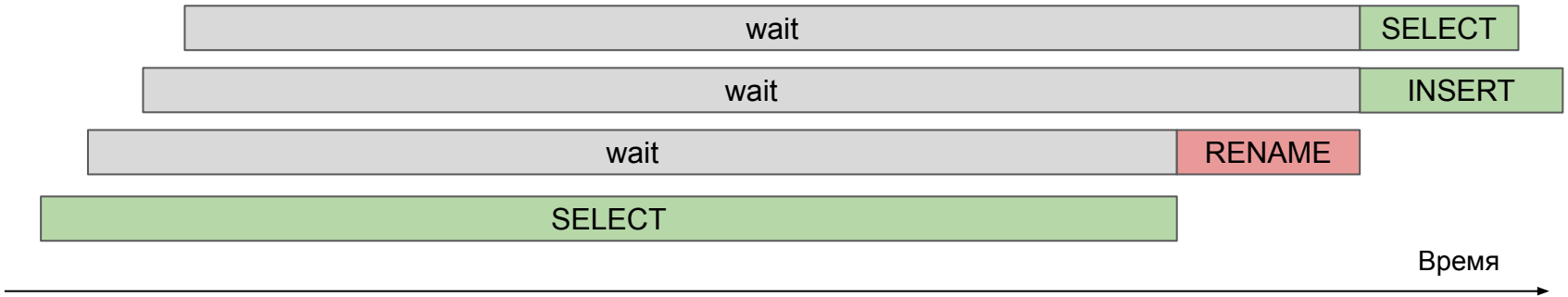
Синхронизация запросов

Специфическая реализация RWLock

- рекурсивный; блокируется запросом, а не потоком
- честный
- механизм предотвращения взаимных блокировок
- много ложноположительных срабатываний

Синхронизация запросов

Иногда честности блокировки недостаточно:



Недостатки существующего подхода

- `DB::Exception: Possible deadlock avoided. Client should retry.`
- Возможно длительное ожидание при выполнении DROP и RENAME
- Неатомарное переименование таблицы

Задачи работы

Реализовать движок баз данных Atomic, который:

- Может заменить движок Ordinary
- Не использует механизм табличных RWLock
- Выполняет DROP и RENAME не допуская длительного ожидания

Общее описание решения

В базе данных с движком Atomic:

- Таблицы имеют постоянный UUID
- Имена отображаются на идентификаторы
- Подсчёт ссылок на таблицы
- Фоновое удаление неиспользуемых таблиц

Изменения в структуре директорий

```
/clickhouse_root/
├── metadata/
│   ├── some_database.sql  ──►  ATTACH DATABASE some_database ENGINE=Atomic;
│   ├── some_database/
│   │   ├── other_table.sql
│   │   └── some_table.sql  ──►  ATTACH TABLE _ UUID '8e73547a-52fe-4a97-a911-e94e3653d946'
│   │                            (column1 UInt64, ...
│   ├── other_database.sql
│   ...
├── store/
│   ├── 8e7/
│   │   ├── 8e73547a-52fe-4a97-a911-e94e3653d946/
│   │   │   ├── data_part1
│   │   │   └── data_part2
│   │   ...
├── metadata_dropped/
│   ├── 99385ebe-e41f-4c38-af70-4d687166ef3e.sql
...
```

```
RENAME TABLE db.old_name TO db.new_name;
```

- Атомарно переименовывает файл метаданных
- Данные не перемещаются
- Нет длительных блокировок
- Устойчиво к `kill -9`

Множественное переименование

Частый случай использования:

```
RENAME TABLE table TO table_old, table_new TO table;
```

Альтернатива:

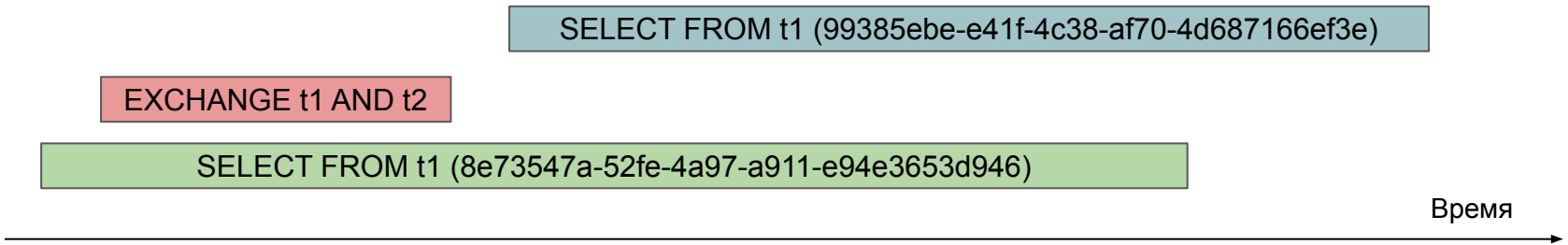
```
EXCHANGE TABLES table AND table_new;
```

Атомарно обменивает файлы метаданных (`renameat2`) и обновляет структуры данных в памяти

Множественное переименование

Альтернатива:

```
EXCHANGE TABLES table AND table_new;
```

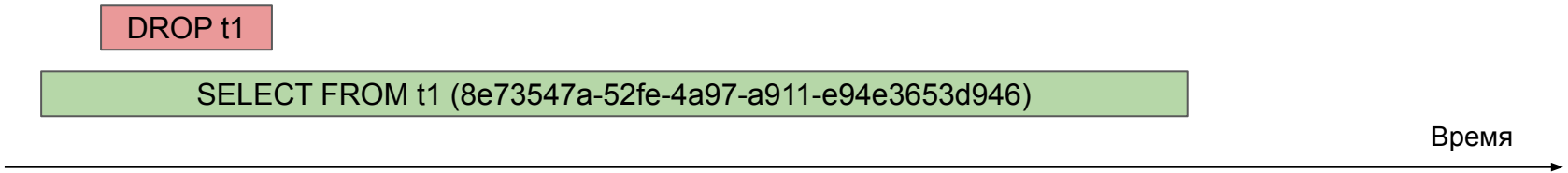


```
DROP TABLE db.table_name;
```

- Атомарно перемещает файл метаданных в `metadata_dropped/`
- Отвязывает UUID от имени таблицы

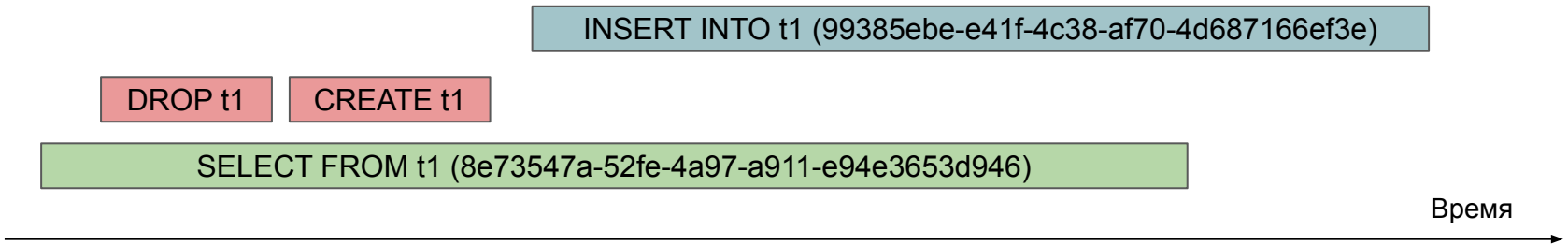
```
DROP TABLE db.table_name;
```

- Атомарно перемещает файл метаданных в `metadata_dropped/`
- Отвязывает UUID от имени таблицы
- Данные не удаляются
- Запросы продолжают выполняться



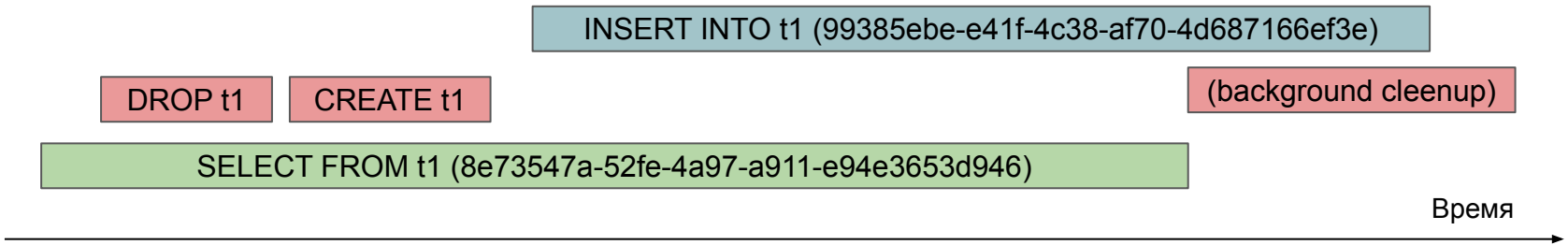
```
DROP TABLE db.table_name;
```

- Атомарно перемещает файл метаданных в `metadata_dropped/`
- Отвязывает UUID от имени таблицы
- Данные не удаляются
- Запросы продолжают выполняться
- Новая таблица может быть создана сразу



```
DROP TABLE db.table_name;
```

- Атомарно перемещает файл метаданных в `metadata_dropped/`
- Отвязывает UUID от имени таблицы
- Данные не удаляются
- Запросы продолжают выполняться
- Новая таблица может быть создана сразу
- Фоновый поток удаляет неиспользуемые данные



Результаты

- В движке Atomic устранены существенные недостатки движка Ordinary
- Доступно в ClickHouse с версии 20.4 (в экспериментальном режиме)

Вопросы