Реплицируемые базы данных в ClickHouse

Александр Токмаков,
разработчик ClickHouse

Движок баз данных Replicated

- Основан на Atomic
- Выполняет DDL запросы (почти) как ON CLUSTER
- Метаданные таблиц в ZooKeeper
- Подключение новых и восстановление отставших реплик
- Динамическая конфигурация кластера в ZooKeeper

DDL ON CLUSTER

```
<remote_servers>
    <test>
        <shard>
            <replica>
                <host>node1</host>
                <port>9000</port>
            </replica>
            <replica>
                <host>node2</host>
                <port>9000</port>
            </replica>
        </shard>
        <shard>
            <replica>
                <host>node3</host>
                <port>9000</port>
            </replica>
        </shard>
    </test>
</remote_servers>
```

DDL ON CLUSTER

```
<remote_servers>
    <test>
        <shard>
            <replica>
                <host>node1</host>
                <port>9000</port>
            </replica>
            <replica>
                <host>node2</host>
                <port>9000</port>
            </replica>
        </shard>
        <shard>
            <replica>
                <host>node3</host>
                <port>9000</port>
            </replica>
        </shard>
    </test>
</remote_servers>
```

```
node1 :) ALTER TABLE t ON CLUSTER test ADD COLUMN ...


┌host──┬status┬─error──────────────────────────────────── ...
│node1│    0 │
│node2│    0 │
│node3│    0 │
└─────┴──────┴────────────────────────────────────────── ...
```

DDL ON CLUSTER

```
<remote_servers>
    <test>
        <shard>
            <replica>
                <host>node1</host>
                <port>9000</port>
            </replica>
            <replica>
                <host>node2</host>
                <port>9000</port>
            </replica>
        </shard>
        <shard>
            <replica>
                <host>node3</host>
                <port>9000</port>
            </replica>
        </shard>
    </test>
</remote_servers>
```

```
node1 :) ALTER TABLE t ON CLUSTER test ADD COLUMN ...


┌host─┬status┬─error──────────────────────────────────── ...
│node1│    0 │
│node2│  517 │ Metadata on replica is not up to date ...
│node3│    0 │
└─────┴──────┴────────────────────────────────────────── ...

There was an error on [node2:9001]: Code: 517,
e.displayText() = DB::Exception: Metadata on replica is
not up to date with common metadata in Zookeeper. Cannot
alter


node2 :) ALTER TABLE t ADD COLUMN ...
```

DDL ON CLUSTER

```
<remote_servers>
    <test>
        <shard>
            <replica>
                <host>node1</host>
                <port>9000</port>
            </replica>
            <replica>
                <host>node2</host>
                <port>9000</port>
            </replica>
        </shard>
        <shard>
            <replica>
                <host>node3</host>
                <port>9000</port>
            </replica>
        </shard>
    </test>
</remote_servers>
```

+

```
<replica>
    <host>node4</host>
    <port>9000</port>
</replica>
```

Создание реплицируемой базы

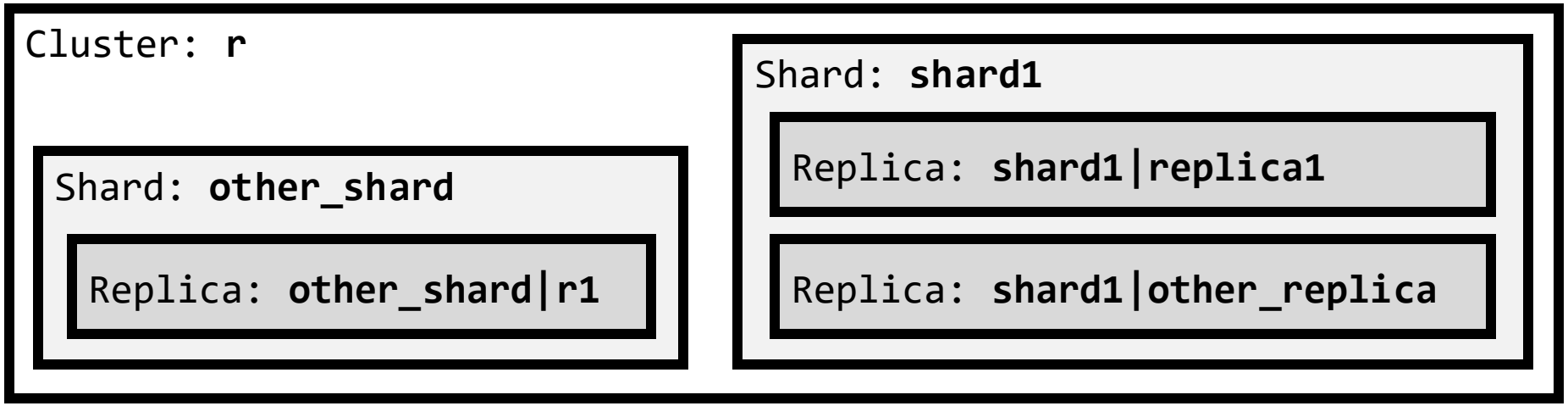
```
node1 :) CREATE DATABASE r ENGINE=Replicated('/some/path/r', 'shard1', 'replica1')
```

```
node2 :) CREATE DATABASE r ENGINE=Replicated('/some/path/r', 'shard1', 'other_replica')
```

```
node3 :) CREATE DATABASE r ENGINE=Replicated('/some/path/r', 'other_shard', '{replica}')
```

Создание реплицируемой базы

```
node1 :) CREATE DATABASE r ENGINE=Replicated('/some/path/r', 'shard1', 'replica1')
node2 :) CREATE DATABASE r ENGINE=Replicated('/some/path/r', 'shard1', 'other_replica')
node3 :) CREATE DATABASE r ENGINE=Replicated('/some/path/r', 'other_shard', '{replica}')
```



DDL запросы

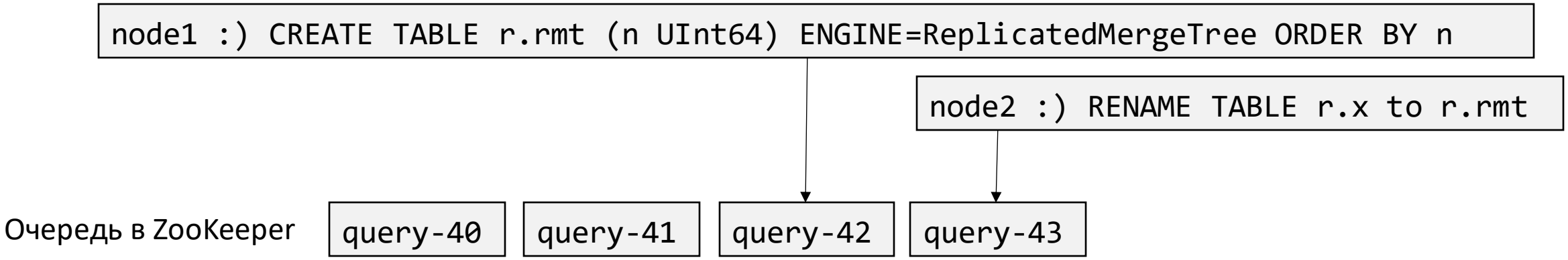
```
node1 :) CREATE DATABASE r ENGINE=Replicated('/some/path/r', 'shard1', 'replica1')
node2 :) CREATE DATABASE r ENGINE=Replicated('/some/path/r', 'shard1', 'other_replica')
node3 :) CREATE DATABASE r ENGINE=Replicated('/some/path/r', 'other_shard', '{replica}')
```

```
node1 :) CREATE TABLE r.rmt (n UInt64) ENGINE=ReplicatedMergeTree ORDER BY n

┌─host──────────────────┬─status─┬─error─┬─num_hosts_remaining─┬─num_hosts_active─┐
│ shard1|replica1       │      0 │       │                   2 │                0 │
│ shard1|other_replica  │      0 │       │                   1 │                0 │
│ other_shard|r1        │      0 │       │                   0 │                0 │
└───────────────────────┴────────┴───────┴─────────────────────┴──────────────────┘
```

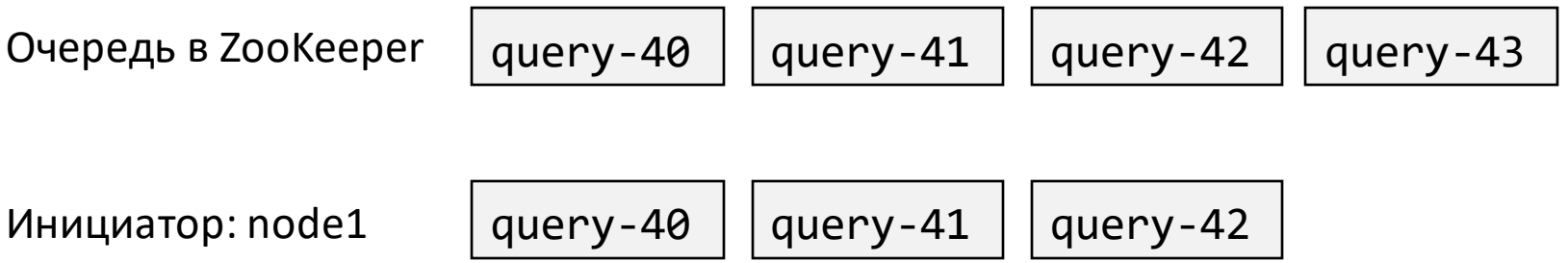
DDL запросы

- Запрос получает номер в очереди репликации



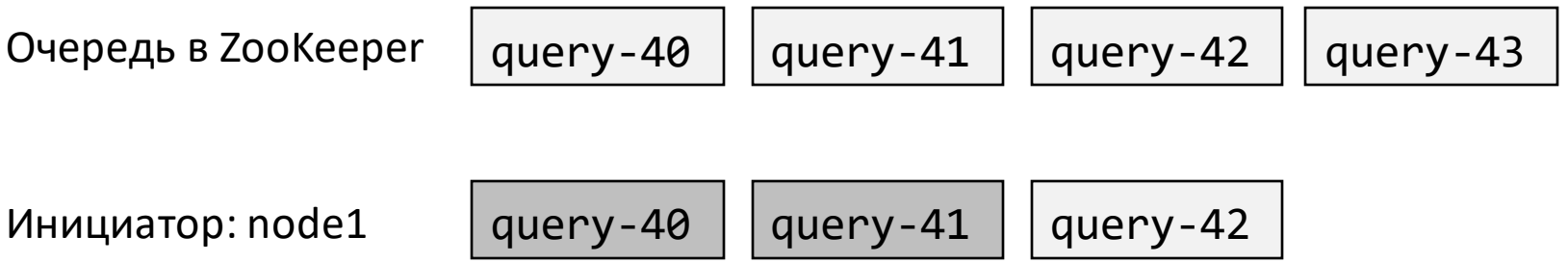
DDL запросы

- Запрос получает номер в очереди репликации
- Инициатор дожидается выполнения очереди до этого номера



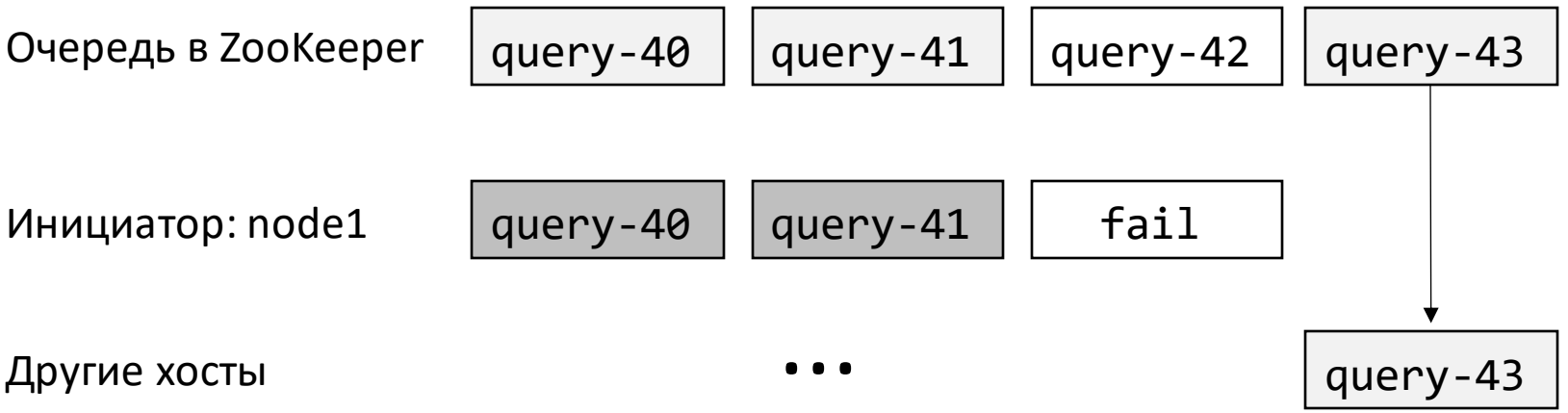
DDL запросы

- Запрос получает номер в очереди репликации
- Инициатор дожидается выполнения очереди до этого номера
- Инициатор пытается выполнить запрос



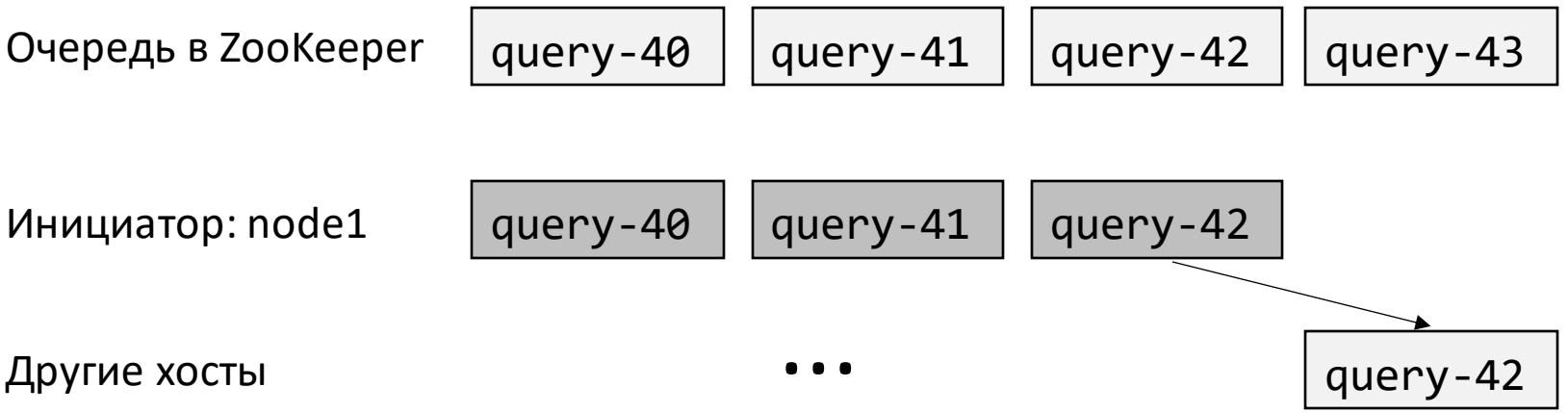
DDL запросы

- Запрос получает номер в очереди репликации
- Инициатор дожидается выполнения очереди до этого номера
- Инициатор пытается выполнить запрос



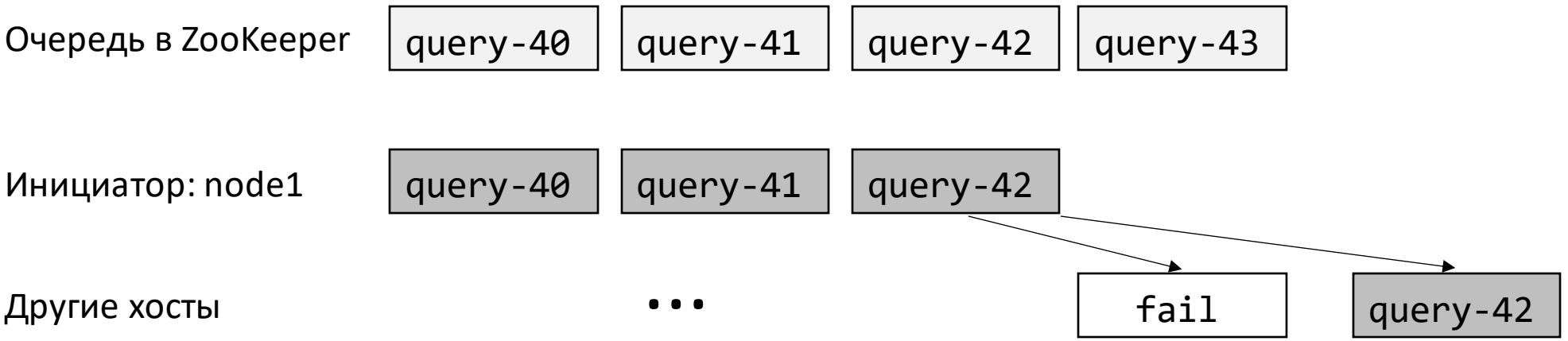
DDL запросы

- Запрос получает номер в очереди репликации
- Инициатор дожидается выполнения очереди до этого номера
- Инициатор пытается выполнить запрос
- Остальные хосты выполняют запрос



DDL запросы

- Запрос получает номер в очереди репликации
- Инициатор дожидается выполнения очереди до этого номера
- Инициатор пытается выполнить запрос
- Остальные хосты выполняют запрос
- Запрос либо вернёт ошибку на инициаторе, либо выполнится на всех хостах



Кластер

```
node1 :) CREATE DATABASE r ENGINE=Replicated('/some/path/r', 'shard1', 'replica1')
node2 :) CREATE DATABASE r ENGINE=Replicated('/some/path/r', 'shard1', 'other_replica')
node3 :) CREATE DATABASE r ENGINE=Replicated('/some/path/r', 'other_shard', '{replica}')
```

```
node1 :) SELECT cluster, shard_num, replica_num, host_name, host_address, port, is_local
FROM system.clusters WHERE cluster='r'


┌─cluster─┬─shard_num─┬─replica_num─┬─host_name─┬─host_address─┬─port─┬─is_local─┐
│ r       │         1 │           1 │ node3     │ 127.0.0.1    │ 9002 │        0 │
│ r       │         2 │           1 │ node2     │ 127.0.0.1    │ 9001 │        0 │
│ r       │         2 │           2 │ node1     │ 127.0.0.1    │ 9000 │        1 │
└─────────┴───────────┴─────────────┴───────────┴──────────────┴──────┴──────────┘
```

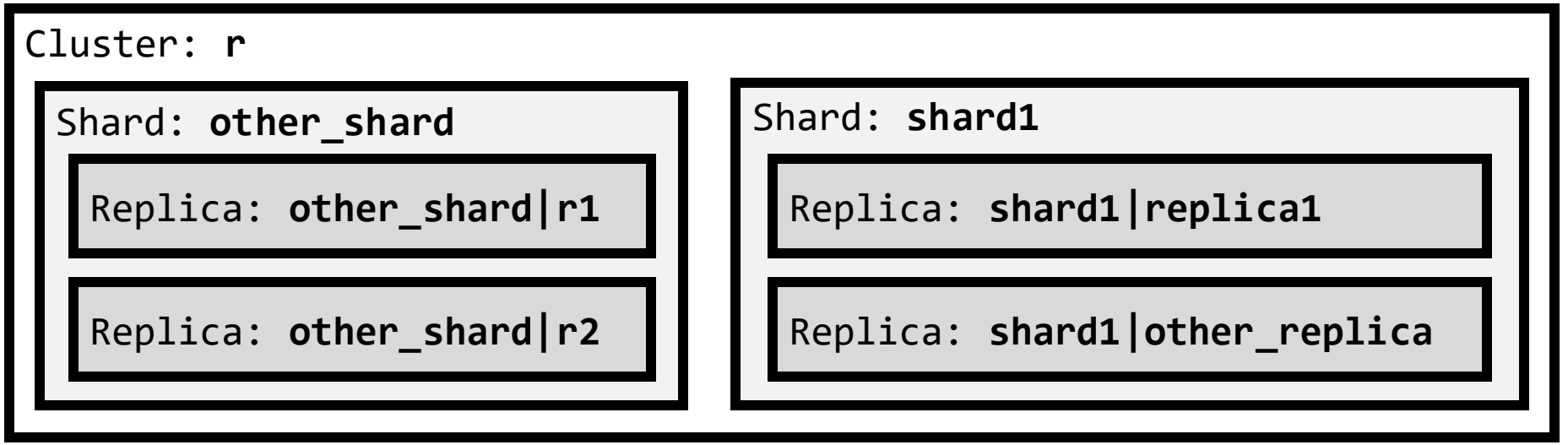
Кластер

```
node2 :) CREATE TABLE r.d (n UInt64) ENGINE=Distributed('r', 'r', 'rmt', n % 2)
node3 :) INSERT INTO r.d SELECT * FROM numbers(10)
node1 :) SELECT materialize(hostName()) AS host, groupArray(n) FROM r.d GROUP BY host

┌─host─┬─groupArray(n)─┐
│ node1 │ [1,3,5,7,9]   │
│ node3 │ [0,2,4,6,8]   │
└──────┴───────────────┘
```

Добавление реплики

```
node4 :) CREATE DATABASE r ENGINE=Replicated('/some/path/r', 'other_shard', 'r2')
```



Добавление реплики

```
node4 :) CREATE DATABASE r ENGINE=Replicated('/some/path/r', 'other_shard', 'r2')

node1 :) SELECT cluster, shard_num, replica_num, host_name, host_address, port, is_local
  FROM system.clusters WHERE cluster='r'


┌─cluster─┬─shard_num─┬─replica_num─┬─host_name─┬─host_address─┬─port─┬─is_local─┐
│ r       │         1 │           1 │ node3     │ 127.0.0.1    │ 9002 │        0 │
│ r       │         1 │           2 │ node4     │ 127.0.0.1    │ 9003 │        0 │
│ r       │         2 │           1 │ node2     │ 127.0.0.1    │ 9001 │        0 │
│ r       │         2 │           2 │ node1     │ 127.0.0.1    │ 9000 │        1 │
└─────────┴───────────┴─────────────┴───────────┴──────────────┴──────┴──────────┘
```

Добавление реплики

```
node4 :) CREATE DATABASE r ENGINE=Replicated('/some/path/r', 'other_shard', 'r2')
node2 :) SELECT materialize(hostName()) AS host, groupArray(n) FROM r.d GROUP BY host


┌─host─┬─groupArray(n)─┐
│ node2 │ [1,3,5,7,9]   │
│ node4 │ [0,2,4,6,8]   │
└──────┴───────────────┘
```

Восстановление реплики

- Для каждой таблицы сравниваются метаданные
- Словари и таблицы без данных пересоздаются
- Нереплицируемые таблицы с данными перемещаются в другую базу, создаются новые пустые таблицы
- Для реплицируемых таблиц обновляются метаданные

Восстановление реплики

r.table1

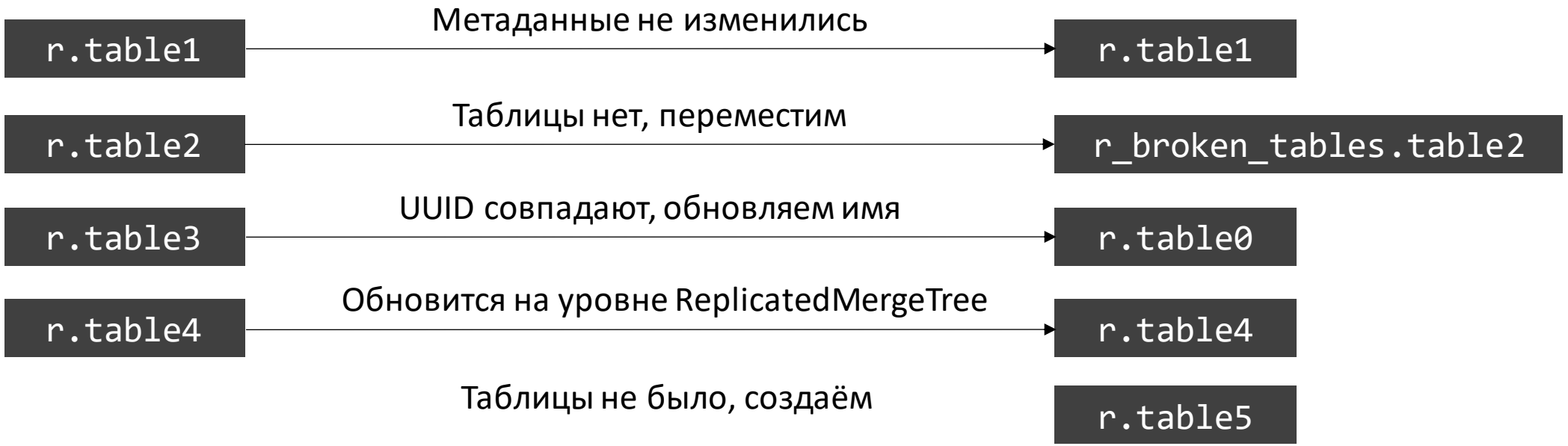
r.table2

r.table3

r.table4

```
:) DROP TABLE table2
:) RENAME TABLE table3 TO table0
:) ALTER TABLE table4 ADD COLUMN ...
:) CREATE TABLE table5 ...
```

Восстановление реплики




```
:) DROP TABLE table2
:) RENAME TABLE table3 TO table0
:) ALTER TABLE table4 ADD COLUMN ...
:) CREATE TABLE table5 ...
```

Спасибо за внимание!

Вопросы?

Александр Токмаков,
разработчик ClickHouse

Email: avtokmakov@yandex-team.ru
GitHub: tavplubix