Netcracker
An NEC Company

# Чтение из Cassandra, внутреннее устройство и производительность

# О себе

- Дмитрий Константинов
- Системный архитектор и, по-прежнему, практикующий Java разработчик в компании Netcracker ☺

# О себе

- Дмитрий Константинов
- Системный архитектор и, по-прежнему, практикующий Java разработчик в компании Netcracker ☺
- Активно работаю с различными OpenSource технологиями, такими как Apache Cassandra, Zookeeper, Kafka, Hazelcast
- Профессиональные интересы:
  - распределенные системы
  - производительность
  - отказоустойчивость

# О чем вообще говорим

- Apache Cassandra 3.11.10
- Операция чтения
  - Общая схема
  - Детали внутреннего устройства
  - Производительность
  - Алгоритмы

# Наш опыт

- Несколько Cassandra кластеров в production
  - От 3 до 180 узлов в кластере
- Как open source версии, так и DSE
- Несколько datacenters
- Типы системы: как OLTP, так и offline batch
- ~ 5k - 200k запросов в секунду на кластер

# Почему знать внутренности важно/полезно

mechanical sympathy

# Почему знать внутренности важно/полезно

mechanical sympathy

- Трудозатраты на анализ проблем
- Возможность оптимизаций

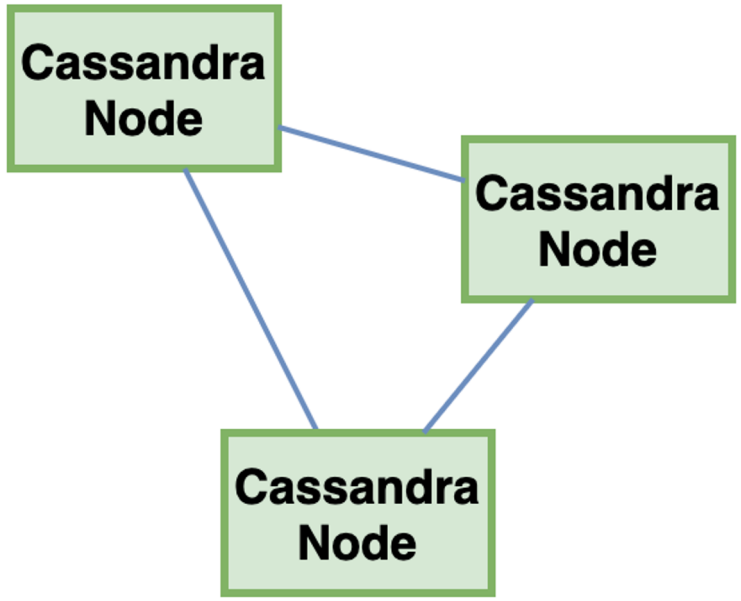
- Базовый тест

# Базовый тест

- Сделаем тест на производительность чтения
- Будем использовать его как базовый при сравнениях

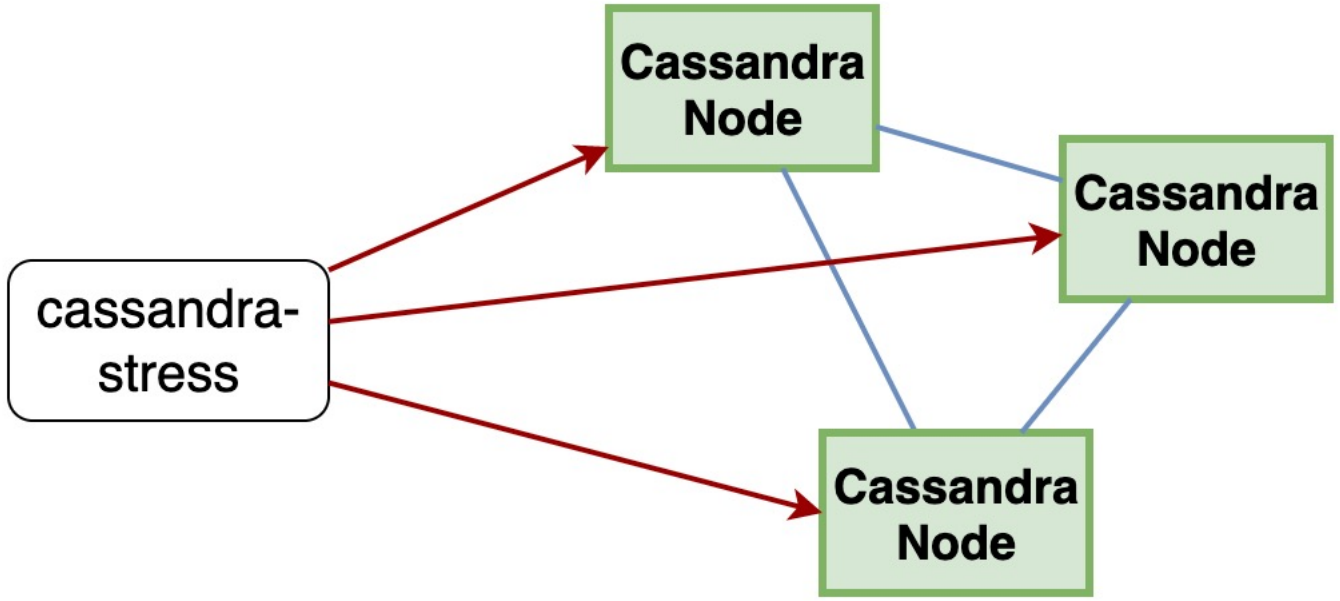
# Базовый тест – окружение

- 3 сервера, RF = 3, Consistency level = LOCAL_QUORUM
- 8 ядер на каждой ноде, 62 Gb RAM, сеть - 1Gbps, диск – SSD



# Базовый тест – окружение

- 3 сервера, RF = 3, Consistency level = LOCAL_QUORUM
- 8 ядер на каждой ноде, 62 Gb RAM, сеть - 1Gbps, диск – SSD
- cassandra-stress



# Базовый тест - данные

```
CREATE TABLE test_reads_table (
    part_key text, // 16 символов, 10,000,000
    clust_key text, // 16 символов, 10 в каждой партиции
    value1 text, // 16 символов
    value2 text, // 16 символов
    value3 text, // 16 символов
    value4 text, // 16 символов
    PRIMARY KEY ((part_key), clust_key));
```

# Базовый тест - данные

```
CREATE TABLE test_reads_table (
    part_key text, // 16 символов, 10,000,000
    clust_key text, // 16 символов, 10 в каждой партиции
    value1 text, // 16 символов
    value2 text, // 16 символов
    value3 text, // 16 символов
    value4 text, // 16 символов
    PRIMARY KEY ((part_key), clust_key));


SELECT * FROM test_reads_table
     WHERE part_key=? AND clust_key=? // равномерное распределение
```

# Базовый тест - результат

- Время ответа: ?
- Пропускная способность: ?

- Базовый тест
- Виды запросов на чтение

# Виды запросов на чтение

- Одна партиция
  - По полному первичному ключу
  - По списку clustering ключей
  - По диапазону clustering ключей
  - Вся партиция целиком

# Виды запросов – по полному ключу

| partition key | clustering key |
|---|---|
| USER_ID | SESSION_ID |
| Alice | 123 |
| | 456 |
| The Dodo | ... |
| Black Queen | ... |

**SELECT * FROM TABLE WHERE USER_ID='Alice' AND SESSION_ID='123'**

# Виды запросов – по диапазону clustering ключей

| partition key | clustering key |
|---|---|
| USER_ID | EVENT_TIME |
| Alice | 18.02.2021 |
| | 02.02.2021 |
| | 01.01.2020 |
| Black Queen | ... |

**SELECT * FROM TABLE WHERE USER_ID='Alice' AND EVENT_TIME >= 02.02.2021**

# Виды запросов на чтение

- Несколько партиций
  - По списку партиций
  - По диапазонам токенов
  - Вся таблица целиком

# Виды запросов – по диапазонам токенов

| partition key | clustering key |
|---|---|
| USER_ID | EVENT_TIME |
| Alice | ... |
| The Dodo | ... |
| The Hatter | ... |
| The Duchess | ... |
| The Mouse | ... |
| The Lory | ... |

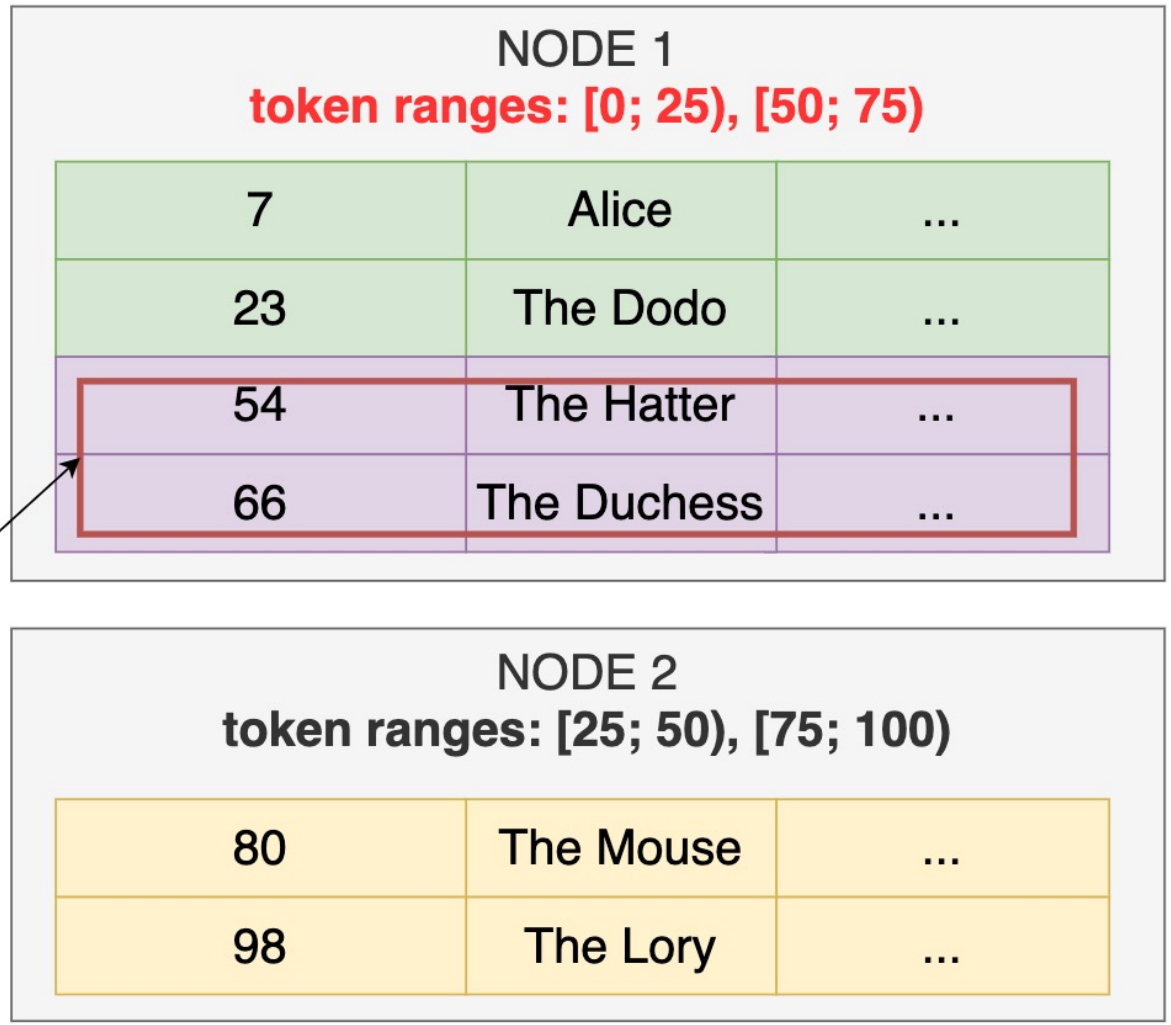
# Виды запросов – по диапазонам токенов

| hash(part_key) | partition key | clustering key |
|---|---|---|
| | USER_ID | EVENT_TIME |
| 7 | Alice | ... |
| 23 | The Dodo | ... |
| 54 | The Hatter | ... |
| 66 | The Duchess | ... |
| 80 | The Mouse | ... |
| 98 | The Lory | ... |

**SELECT * FROM TABLE WHERE WHERE token(USER_ID)>=50 AND token(USER_ID)<75**

# Виды запросов – по диапазонам токенов

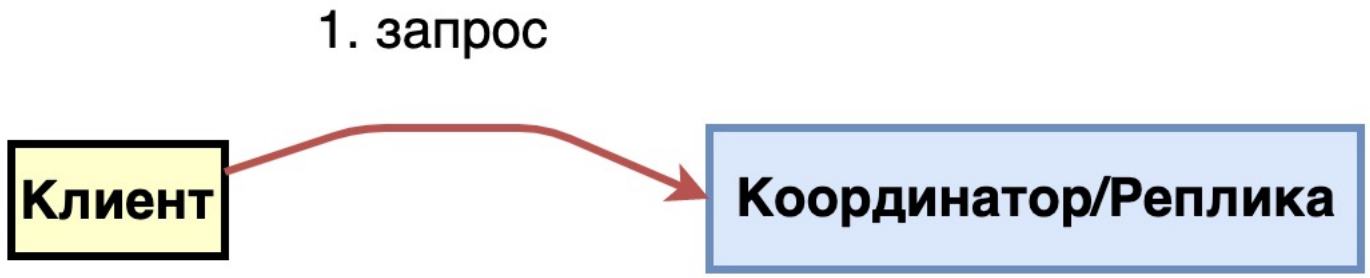
| hash(part_key) | partition key | clustering key |
|---|---|---|
| | USER_ID | EVENT_TIME |
| 7 | Alice | ... |
| 23 | The Dodo | ... |
| 54 | The Hatter | ... |
| 66 | The Duchess | ... |
| 80 | The Mouse | ... |
| 98 | The Lory | ... |




**SELECT * FROM TABLE WHERE WHERE token(USER_ID)>=50 AND token(USER_ID)<75**

- Базовый тест
- Чтение, уровень кластера

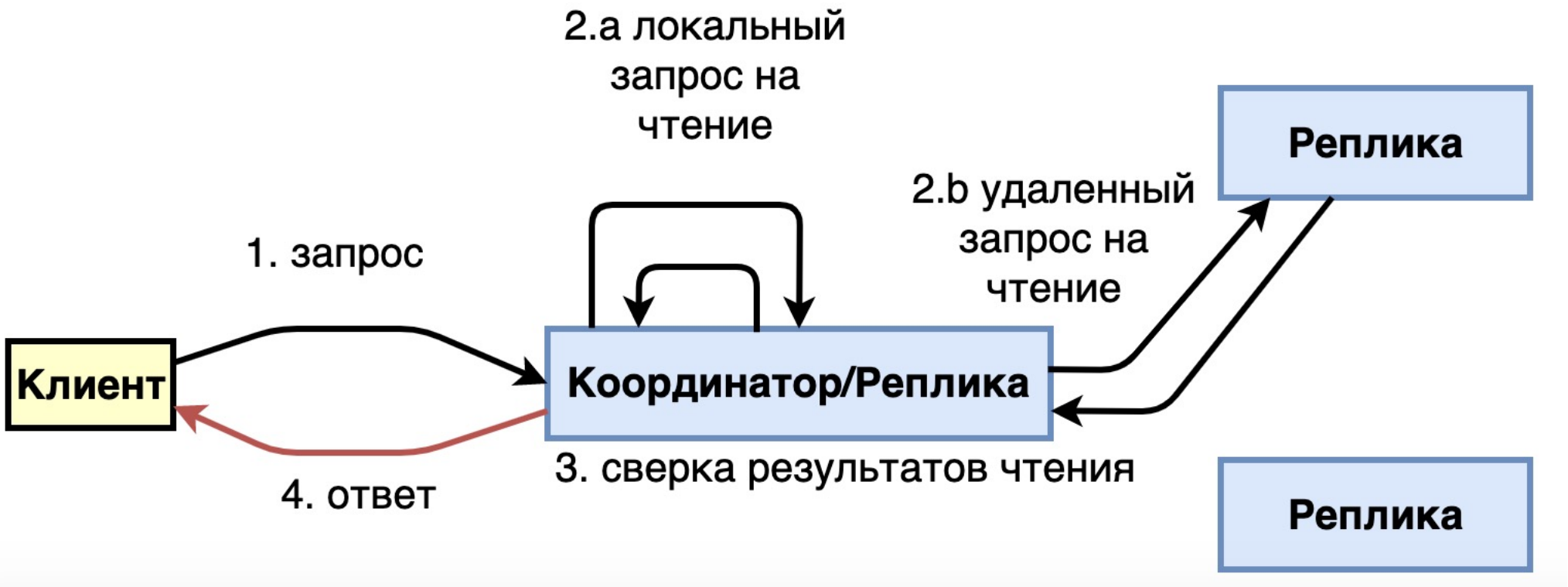
# Чтение – уровень кластера



# Чтение – уровень кластера

# Чтение – уровень кластера

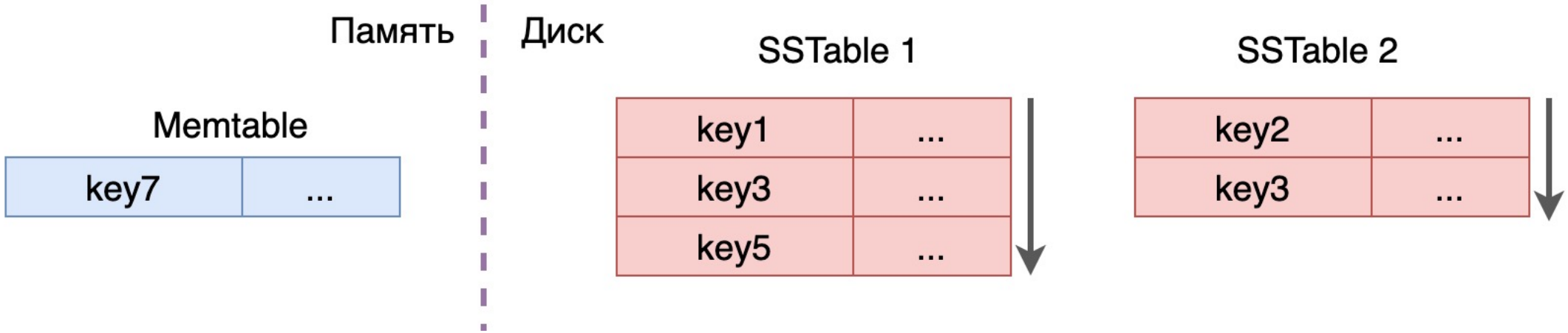
# Чтение – уровень кластера



- Базовый тест
- Запись, уровень кластера
- Чтение, уровень реплики

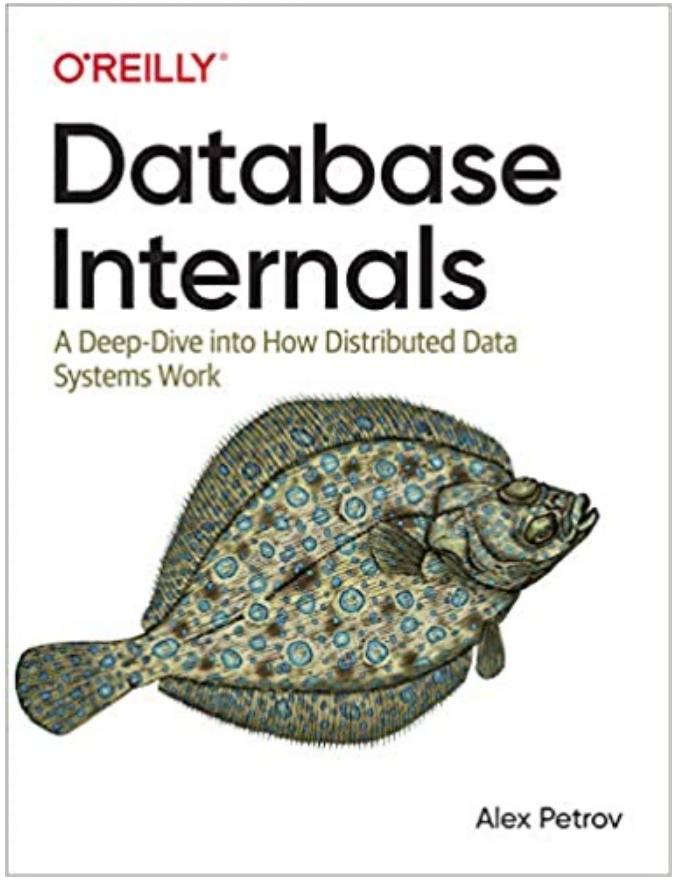
# Запись – что происходит внутри реплики

- LSM-tree (Log Structured Merge-tree)

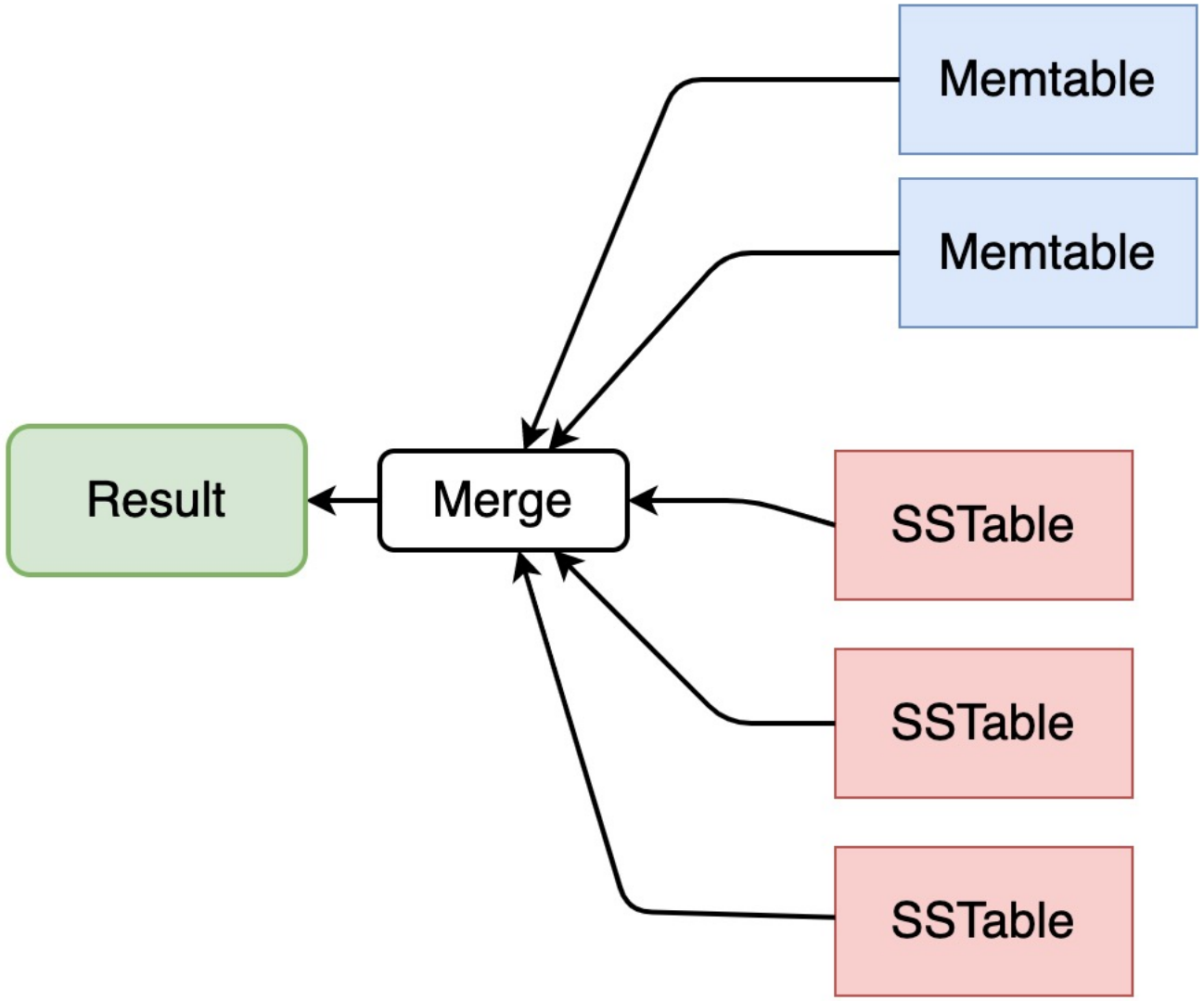


# Запись – что происходит внутри реплики

- Ссылки:
  - The Log-Structured Merge-Tree (LSM-Tree),  Patrick O'Neil, Edward Cheng, Dieter Gawlick, Elizabeth O'Neil; 1996
  - Database Internals: A Deep Dive into How Distributed Data Systems Work, Alex Petrov; 2019



# Чтение – LSM tree (log structured merge-tree)



- Базовый тест
- Запись, уровень кластера
- Запись, уровень реплики
- Базовый тест - результат

# E2E тест - результат

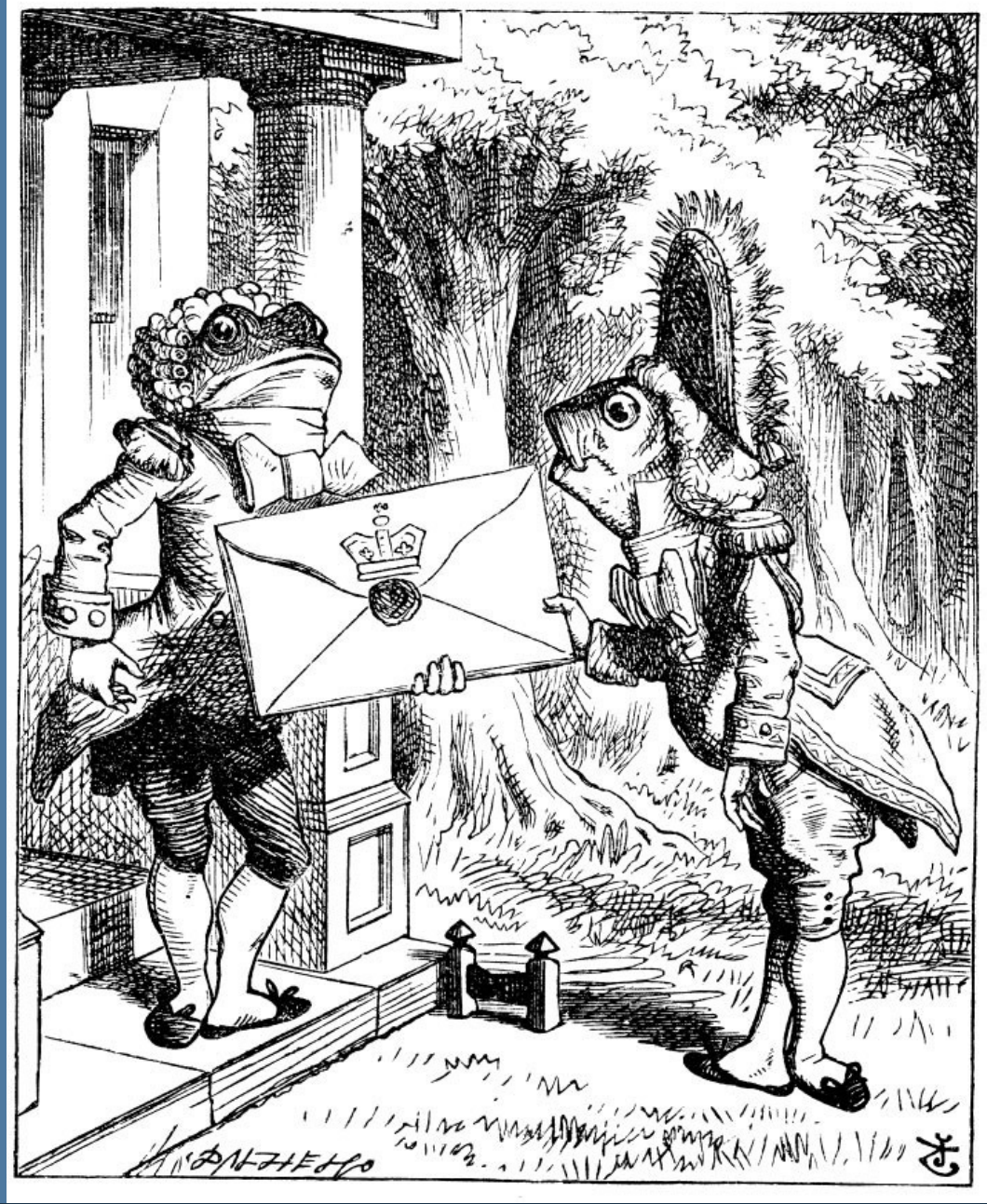
- Пропускная способность
  - 44K операций в секунду

# E2E тест - результат

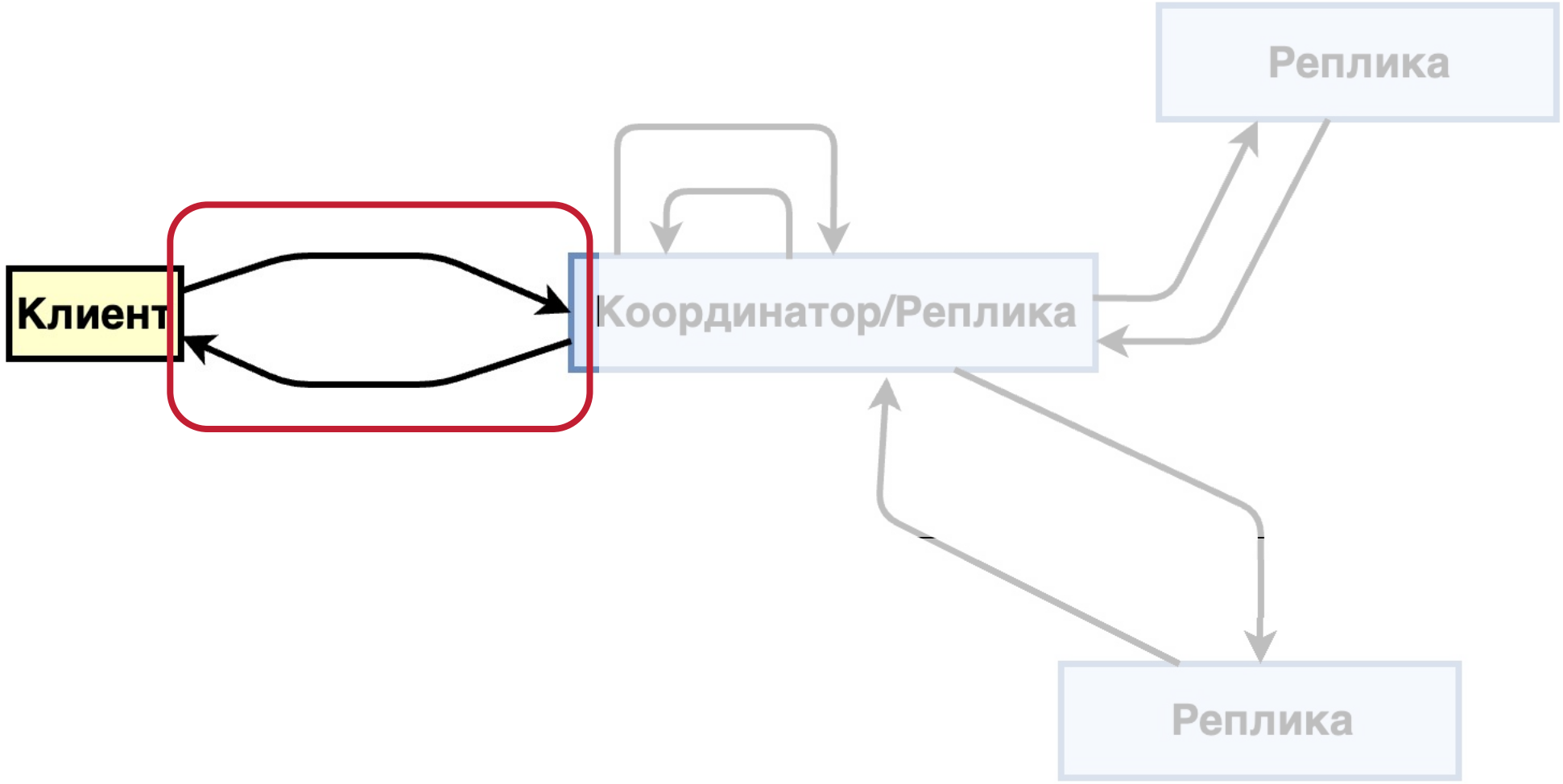
- Пропускная способность
  - 44K операций в секунду
- Время ответа для 25k TPS

```
Latency mean                :     3.8 ms
Latency median              :     1.3 ms
Latency 95th percentile     :    19.2 ms
Latency 99th percentile     :    42.0 ms
```

- Базовый тест
- Запись, уровень кластера
- Запись, уровень реплики
- Базовый тест - результат
- Клиент-серверное взаимодействие
  - Протокол



# Client-server взаимодействие



# Client-server взаимодействие

- Протокол:
  - CQL (Cassandra Query Language)
  - Бинарный TCP протокол

# Client-server взаимодействие

- Протокол:
  - CQL (Cassandra Query Language)
  - Бинарный TCP протокол
  - Асинхронный запрос/ответ (корреляция через stream id)

# Client-server взаимодействие

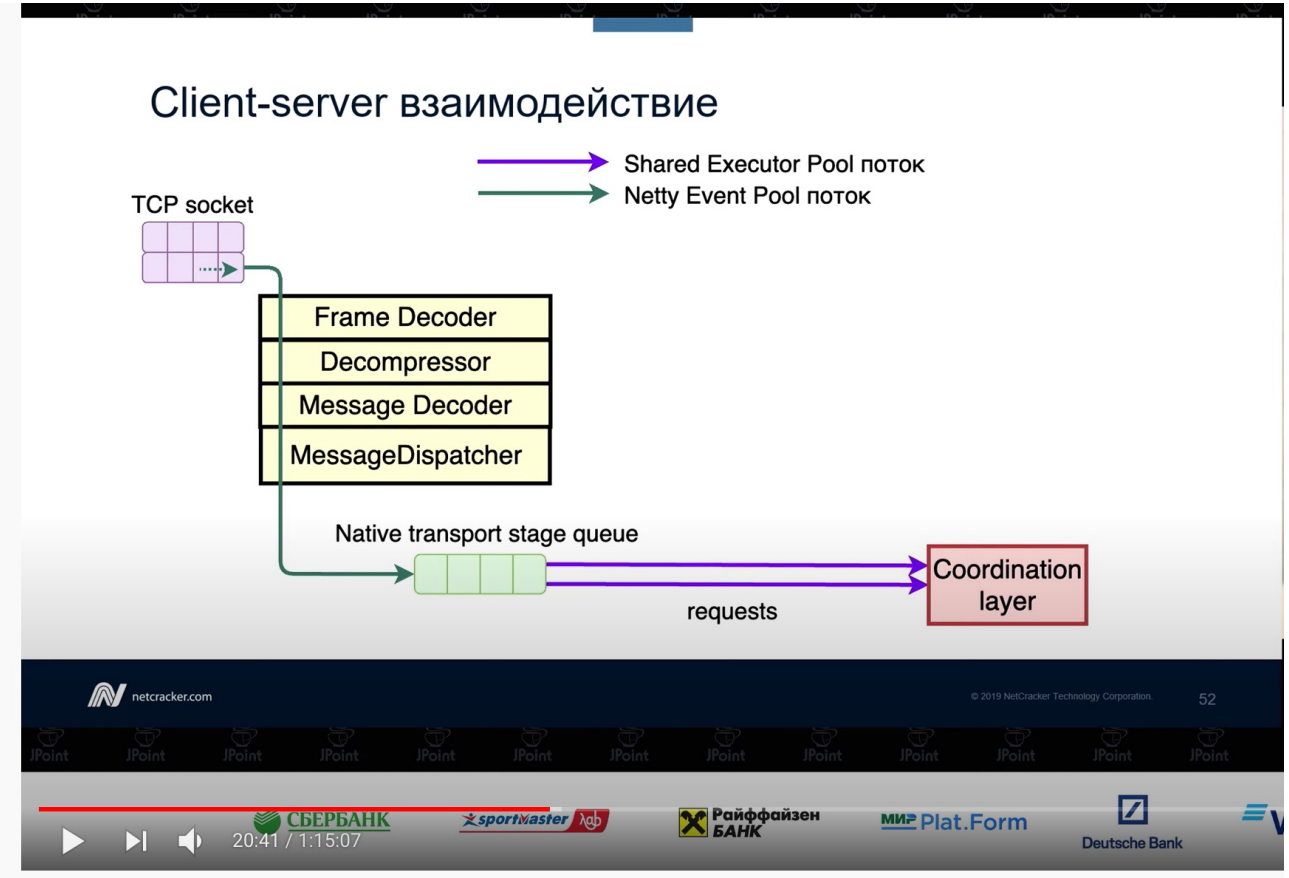
- Протокол:
  - CQL (Cassandra Query Language)
  - Бинарный TCP протокол
  - Асинхронный запрос/ответ (корреляция через stream id)
  - Поддерживаются prepared statements

# Client-server взаимодействие

- Протокол:
  - CQL (Cassandra Query Language)
  - Бинарный TCP протокол
  - Асинхронный запрос/ответ (корреляция через stream id)
  - Поддерживаются prepared statements
- Сервер:
  - Netty, epoll-based

# Client-server взаимодействие

Больше деталей – в докладе ”Запись в Cassandra — внутреннее устройство и производительность”, Jpoint 2020



- Базовый тест
- Запись, уровень кластера
- Запись, уровень реплики
- Базовый тест - результат
- Клиент-серверное взаимодействие
  - Протокол
  - TLS

# Client-server взаимодействие - TLS

- TLS поддерживается для client-server взаимодействия

# Client-server взаимодействие - TLS

- TLS поддерживается для client-server взаимодействия
- Сколько стоит включить TLS?

# Client-server взаимодействие - TLS

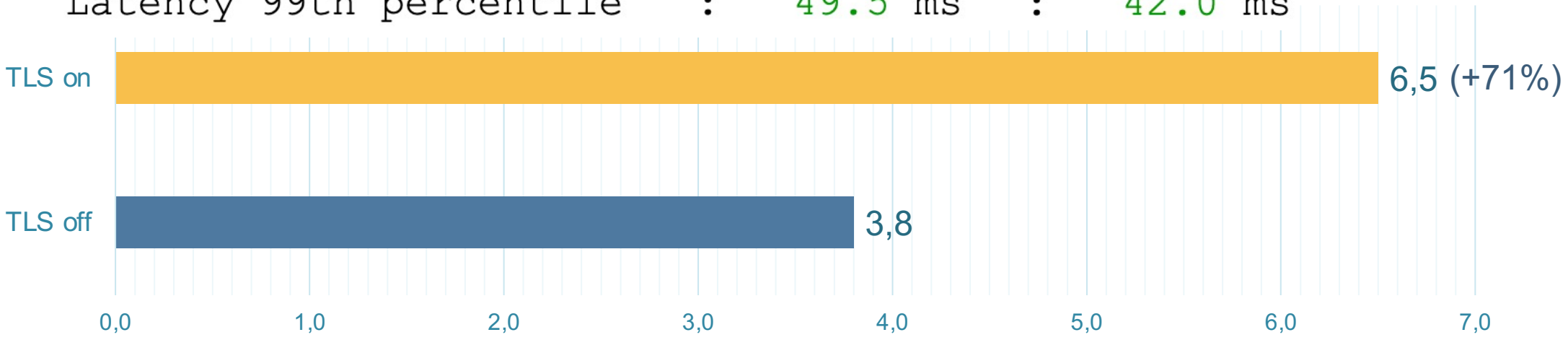
- TLS поддерживается для client-server взаимодействия
- Сколько стоит включить TLS?
- Тест: 25k чтений в секунду, сравним время ответа
- Варианты:
  - TLS выключен
  - TLS включен (TLS 1.2, TLS_RSA_WITH_AES_128_CBC_SHA)

# Client-server взаимодействие - TLS

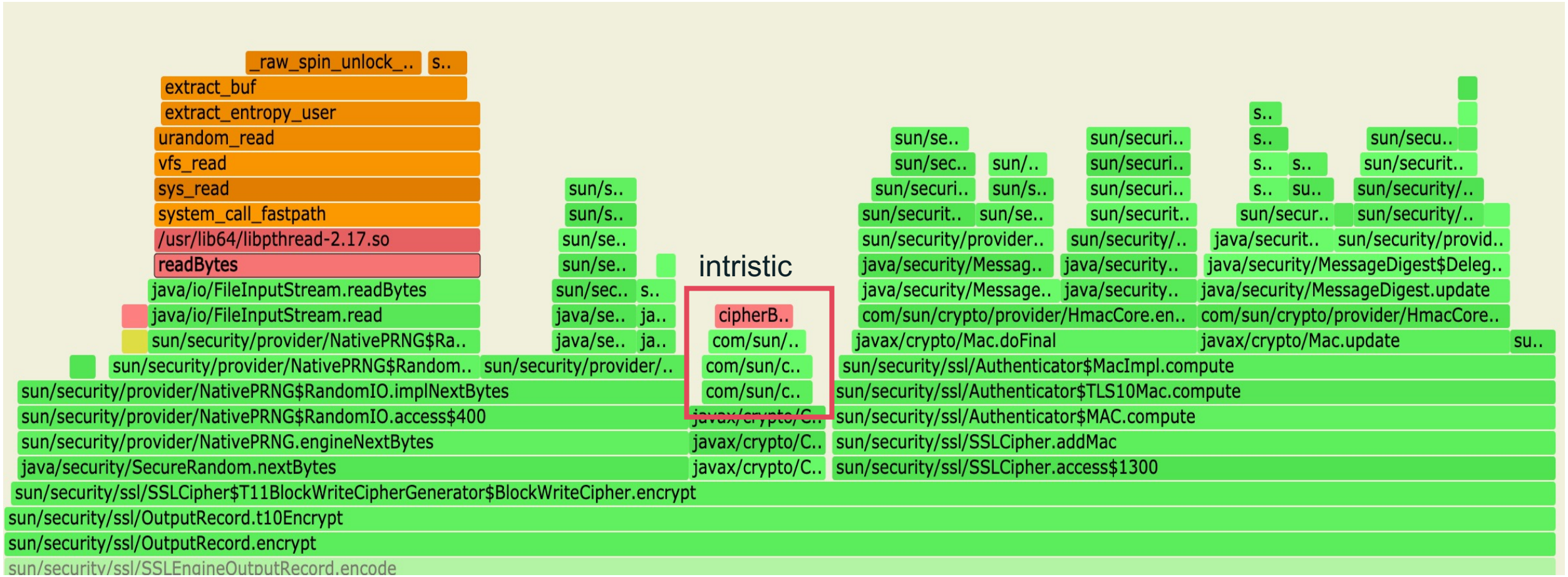
```
Latency mean                :     6.5 ms
Latency median              :     1.7 ms
Latency 95th percentile     :    26.0 ms
Latency 99th percentile     :    49.5 ms
```

# Client-server взаимодействие - TLS

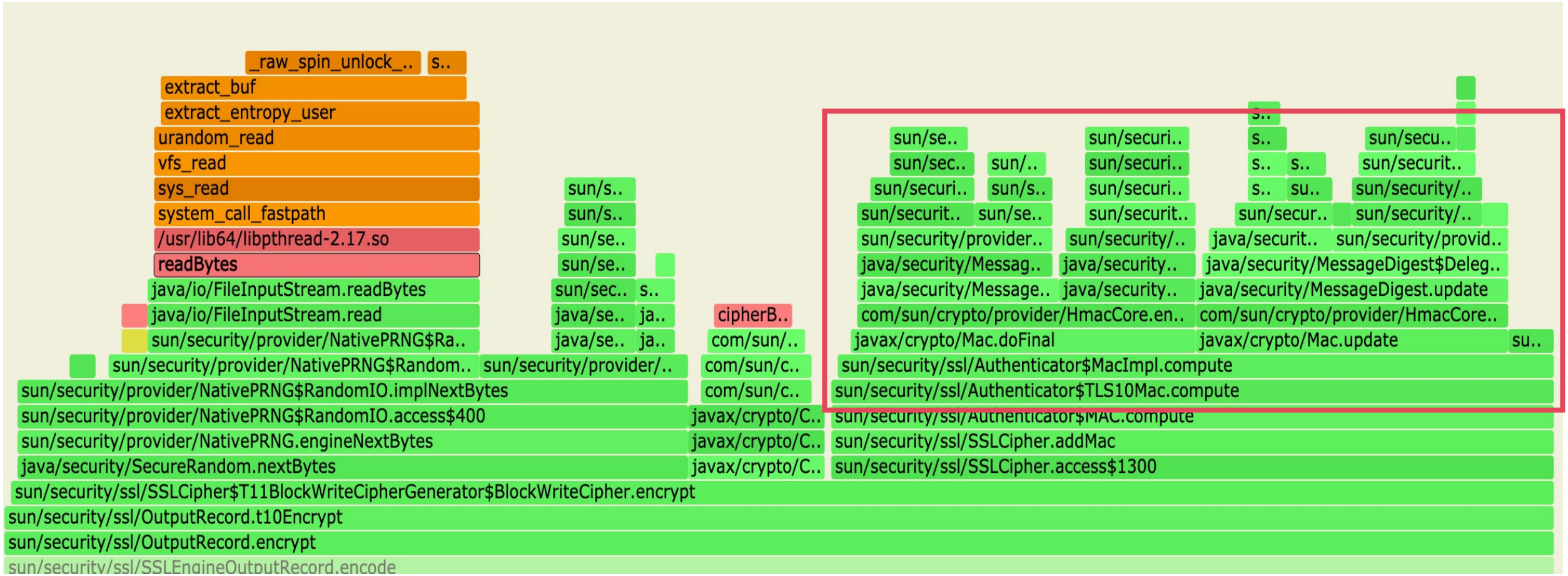
| | TLS on | TLS off |
|---|---|---|
| Latency mean | 6.5 ms | 3.8 ms |
| Latency median | 1.7 ms | 1.3 ms |
| Latency 95th percentile | 26.0 ms | 19.2 ms |
| Latency 99th percentile | 49.5 ms | 42.0 ms |



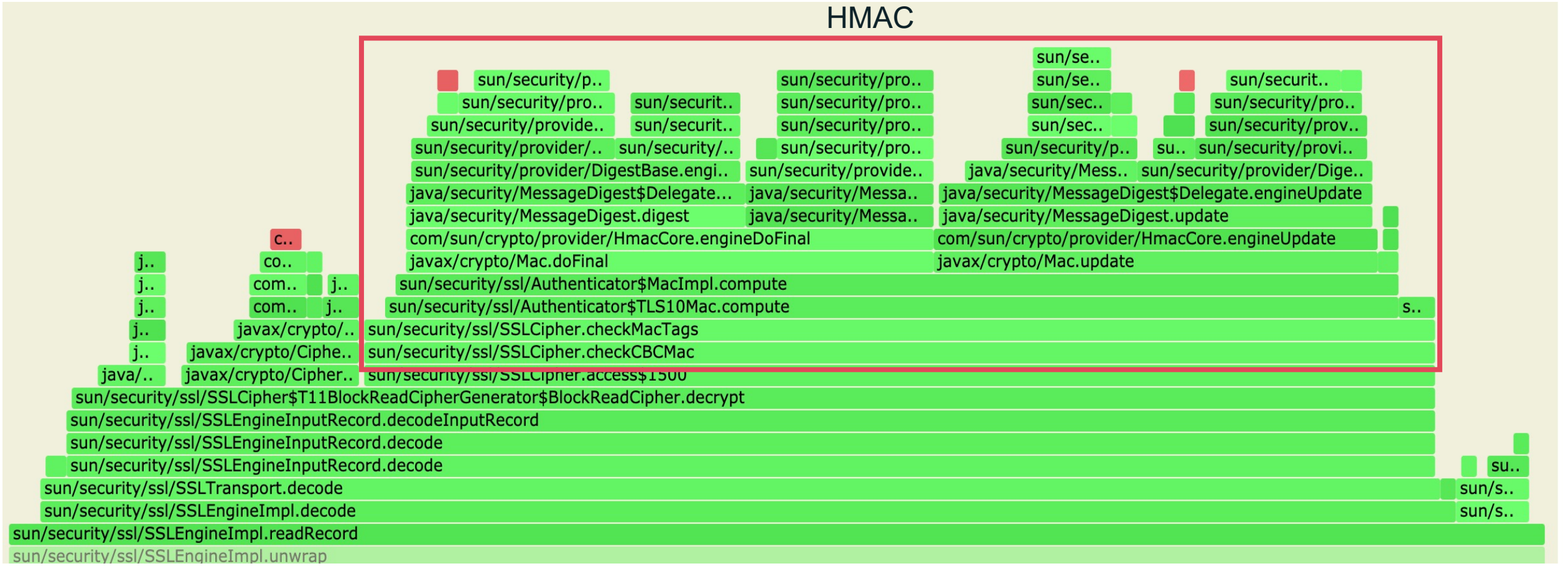

# Client-server взаимодействие – TLS, encrypt



# Client-server взаимодействие – TLS, encrypt



# Client-server взаимодействие – TLS, decrypt



# TLS - что можно улучшить

- Native SSL (Netty)
  - https://netty.io/wiki/forked-tomcat-native.html

# TLS - что можно улучшить

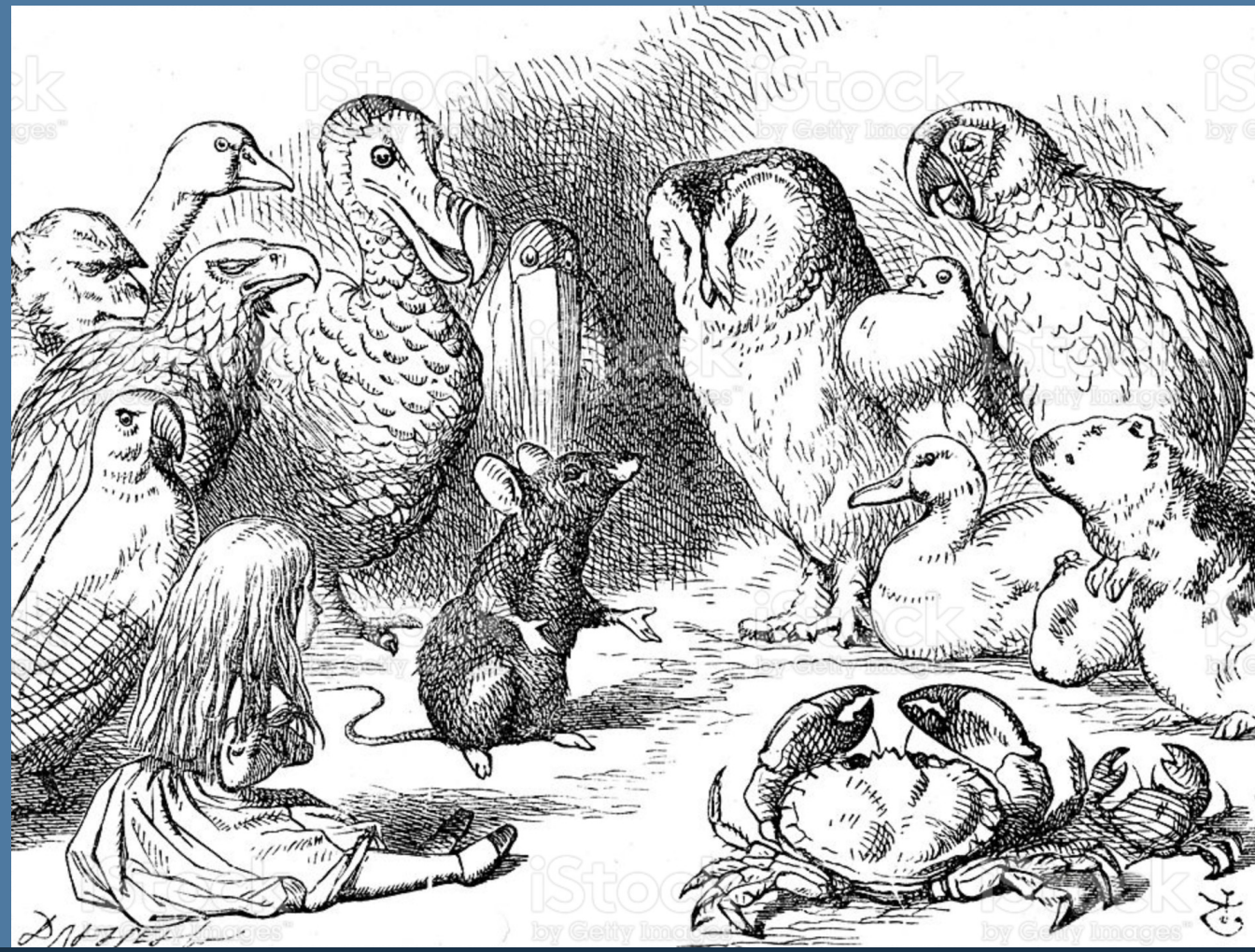
- Native SSL (Netty)
  - https://netty.io/wiki/forked-tomcat-native.html
  - https://docs.datastax.com/en/developer/java-driver/3.10/manual/ssl/#netty - драйвер
  - CASSANDRA-10735 (поддержка в Cassandra 4.0)

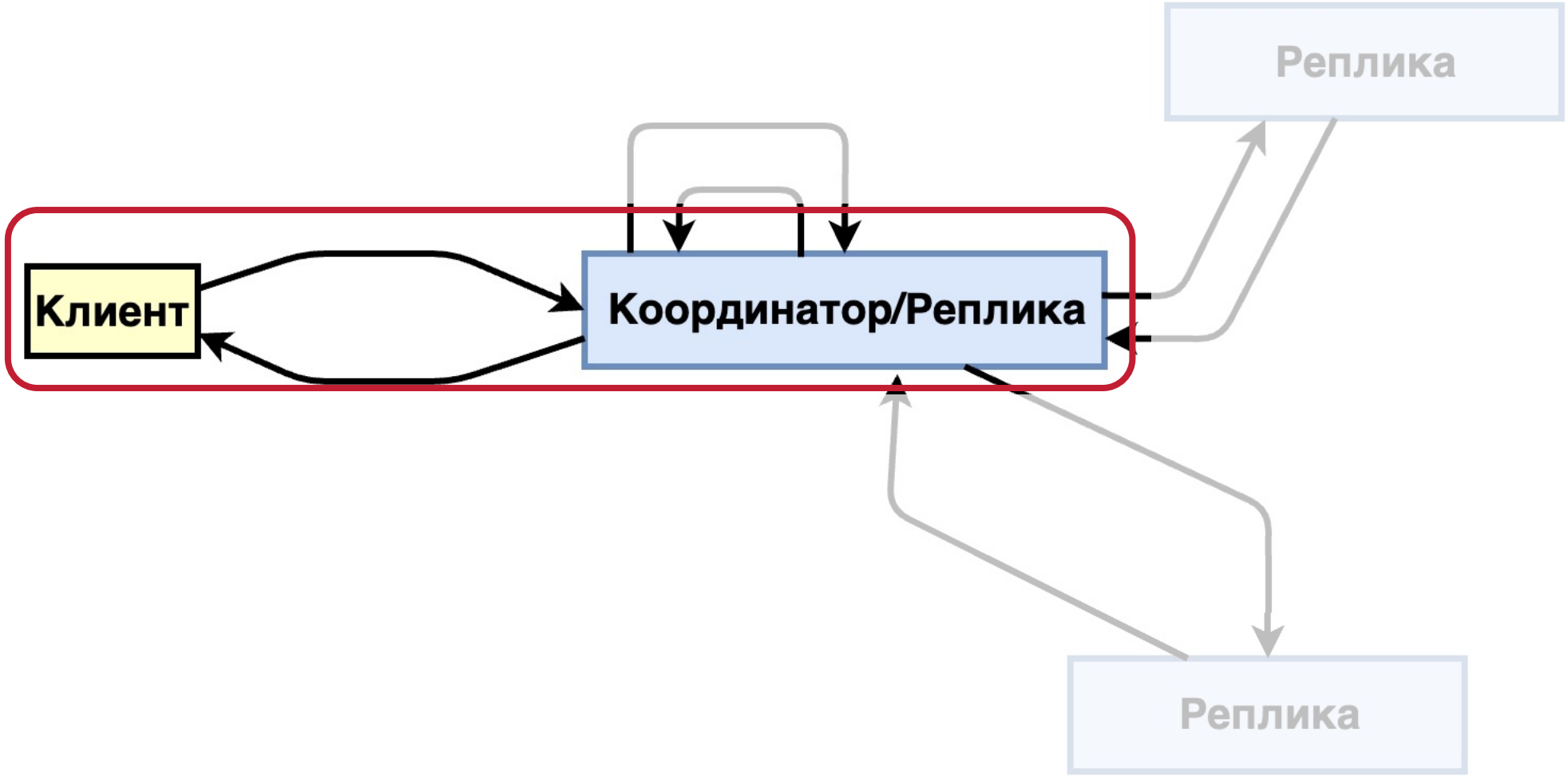
# TLS - что можно улучшить

- Native SSL (Netty)
  - https://netty.io/wiki/forked-tomcat-native.html
  - https://docs.datastax.com/en/developer/java-driver/3.10/manual/ssl/#netty - драйвер
  - CASSANDRA-10735 (поддержка в Cassandra 4.0)

- Intrinsic для SHA 1/2 - JDK-8150767 (JDK 9)

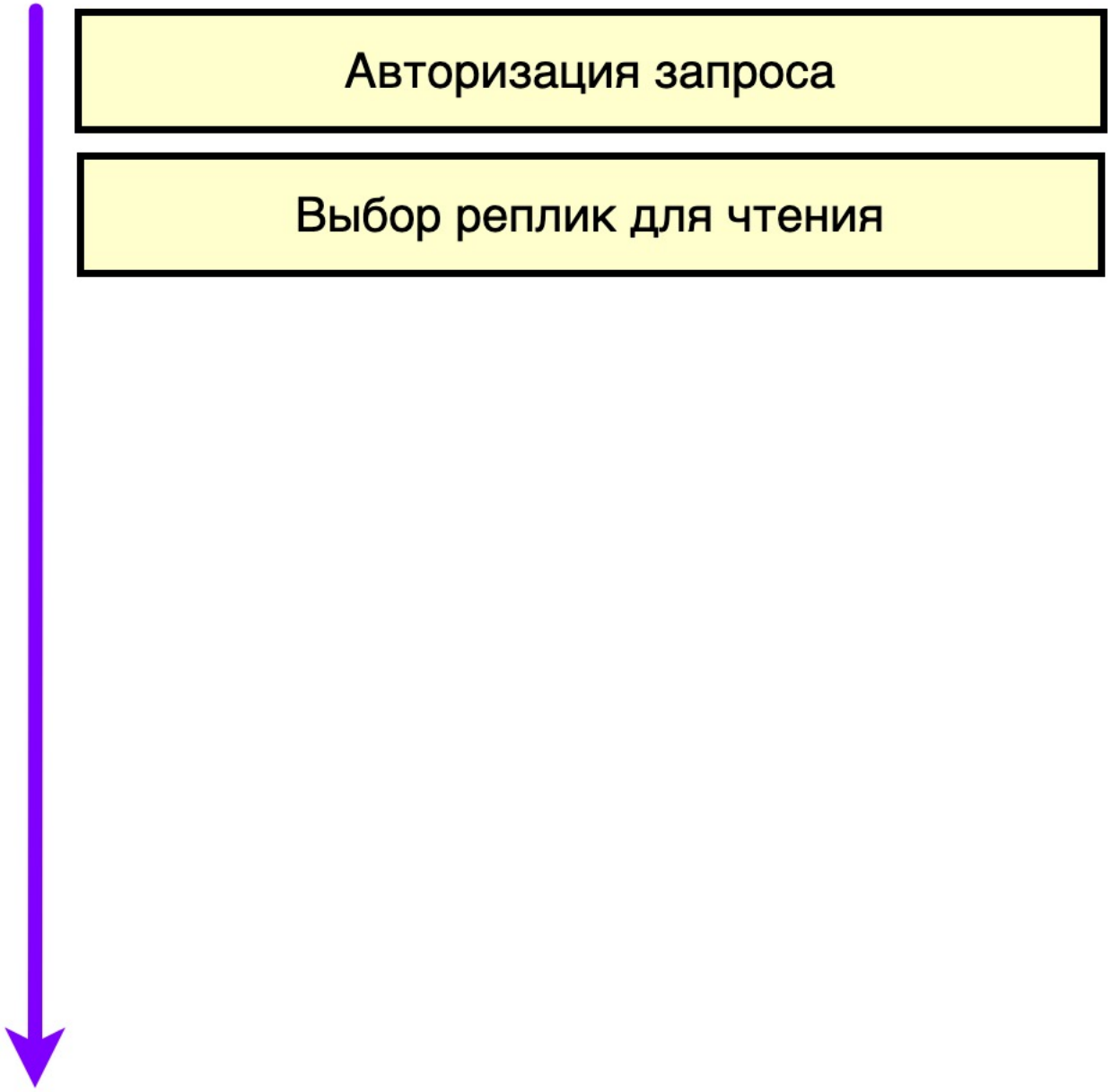
- Запись, уровень кластера
- Запись, уровень реплики
- Базовый тест
- Клиент-сервер
- Координация
  - Общая последовательность



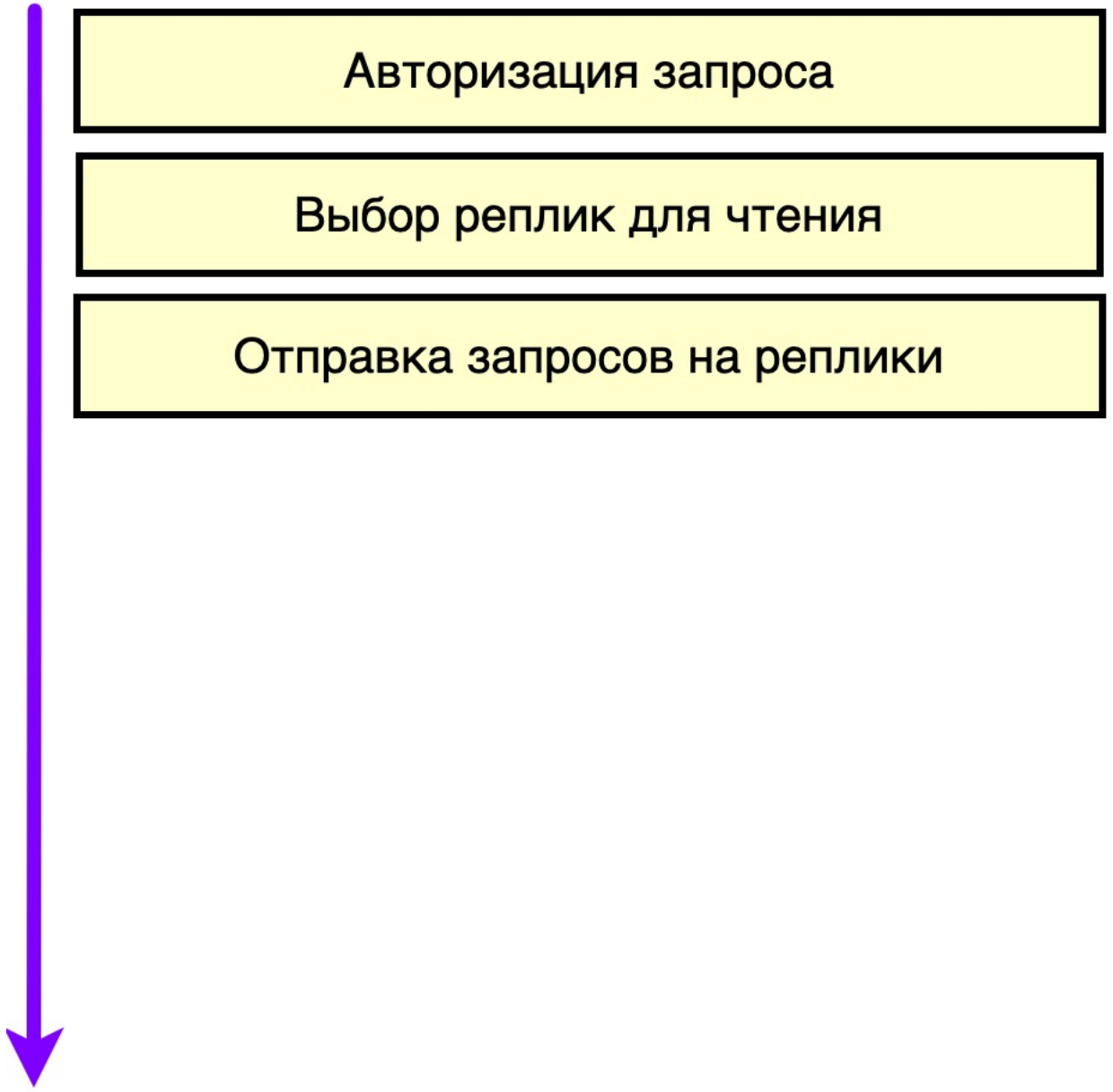


# Координация

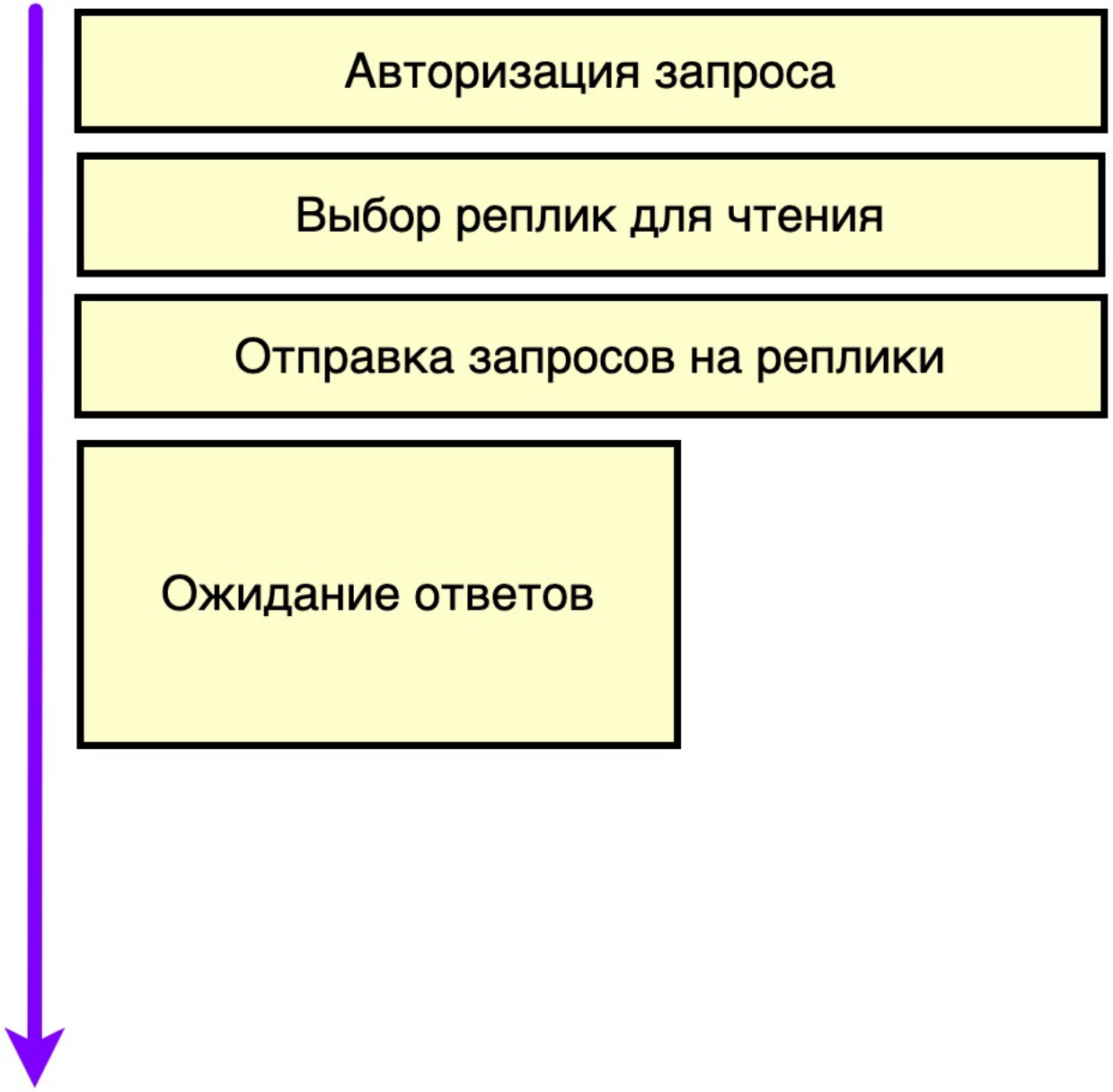
# Координация



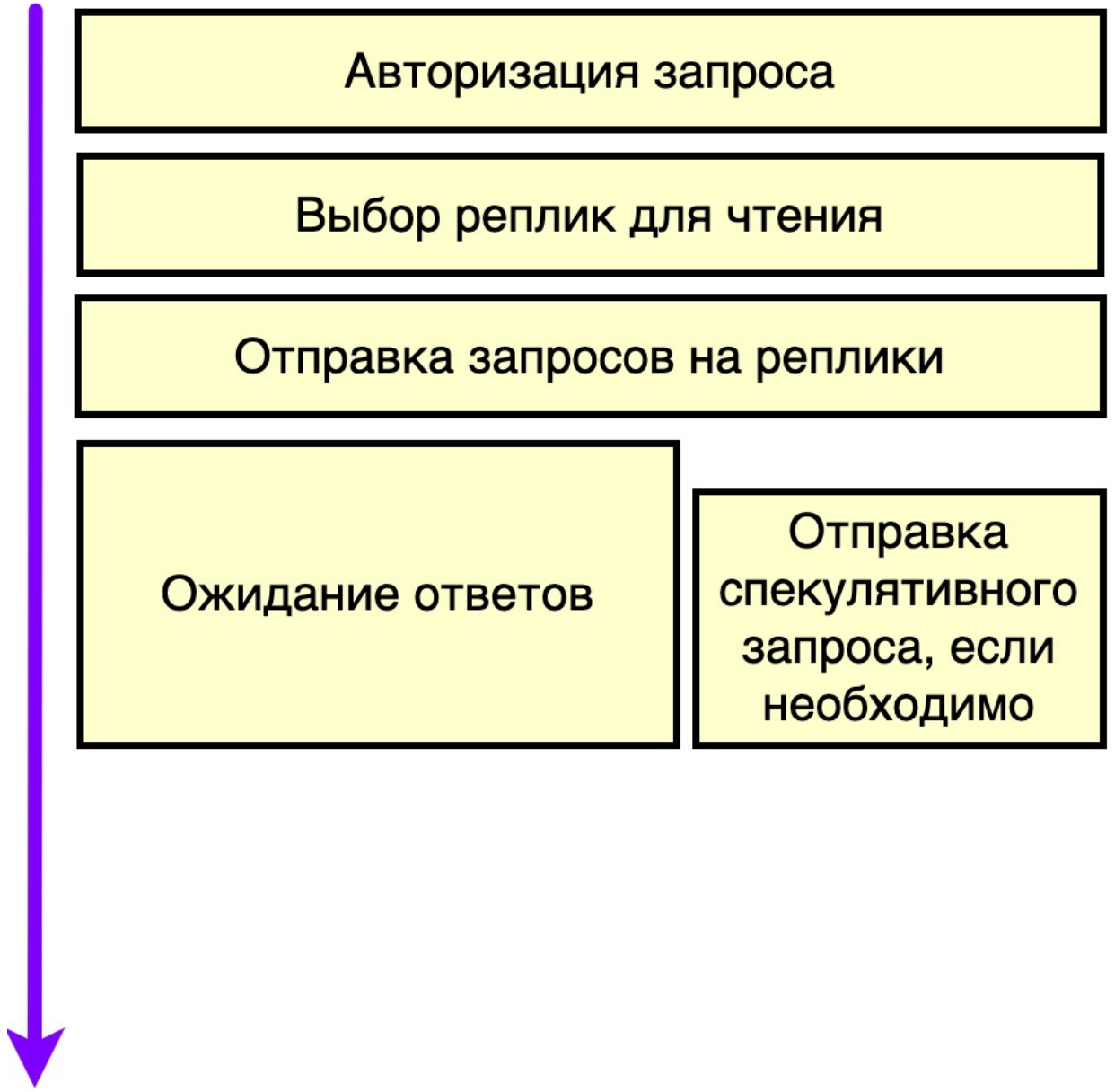
# Координация



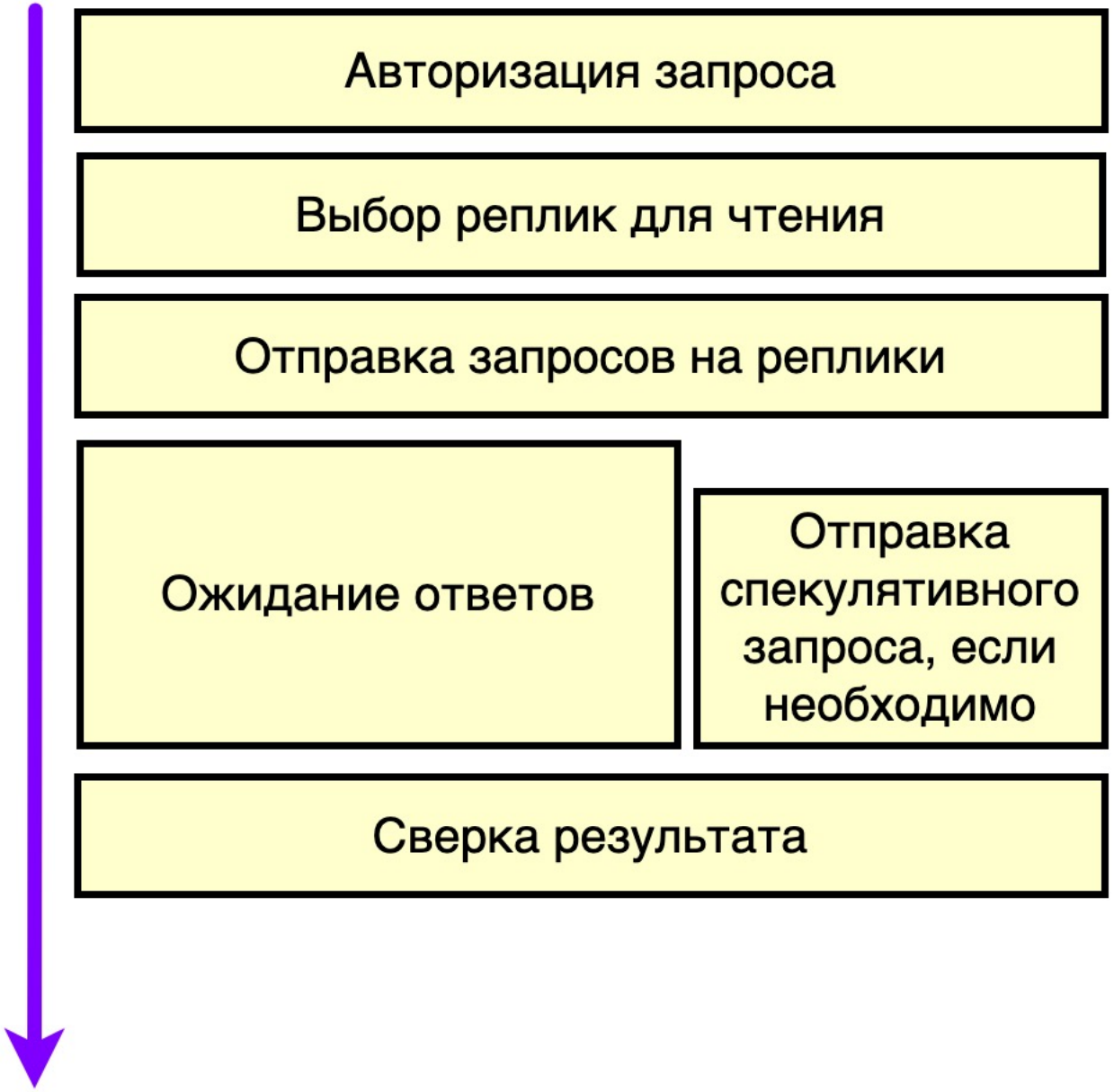
# Координация



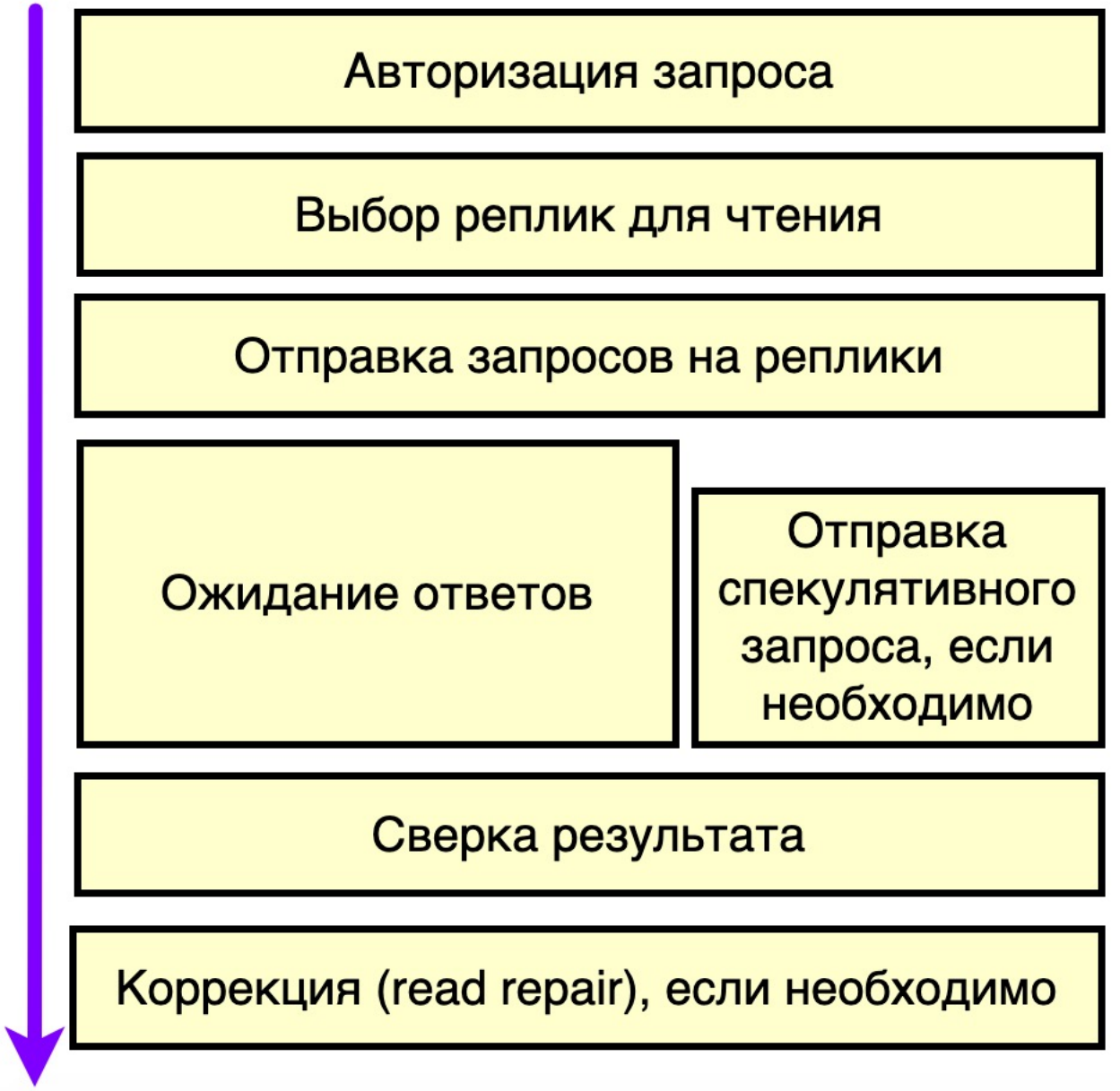
# Координация



# Координация

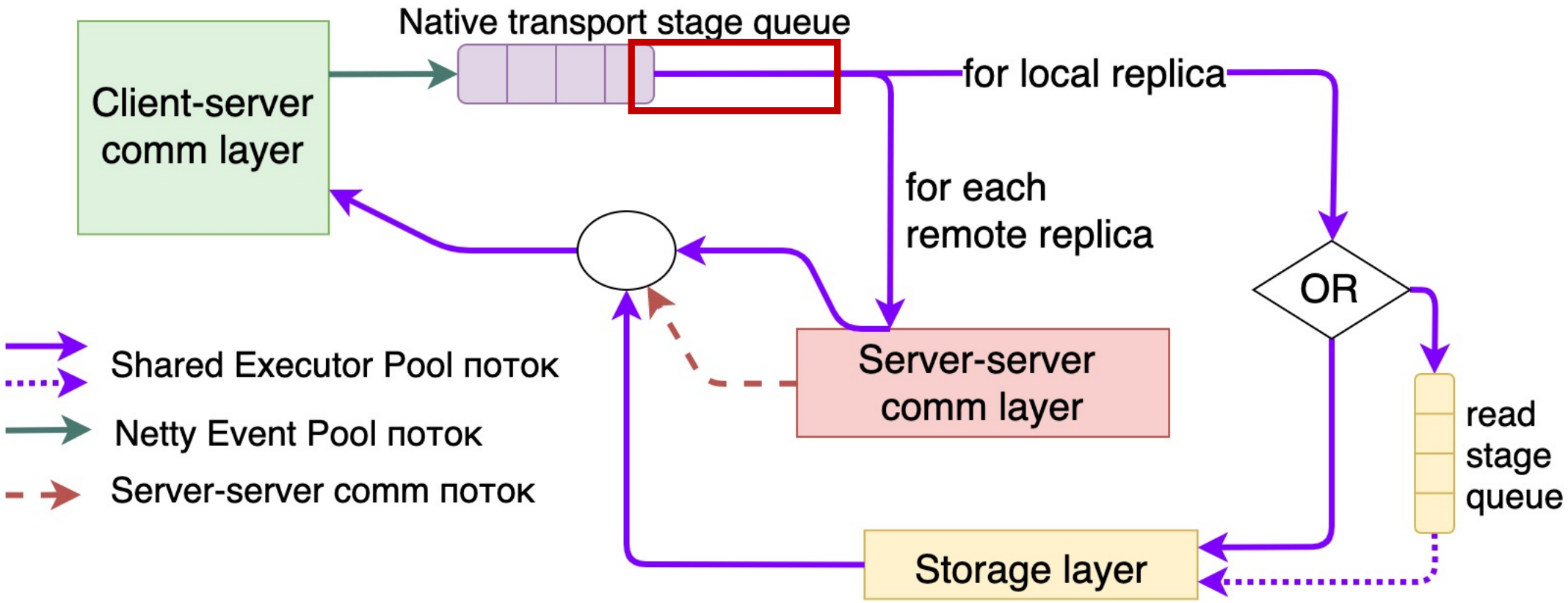


# Координация



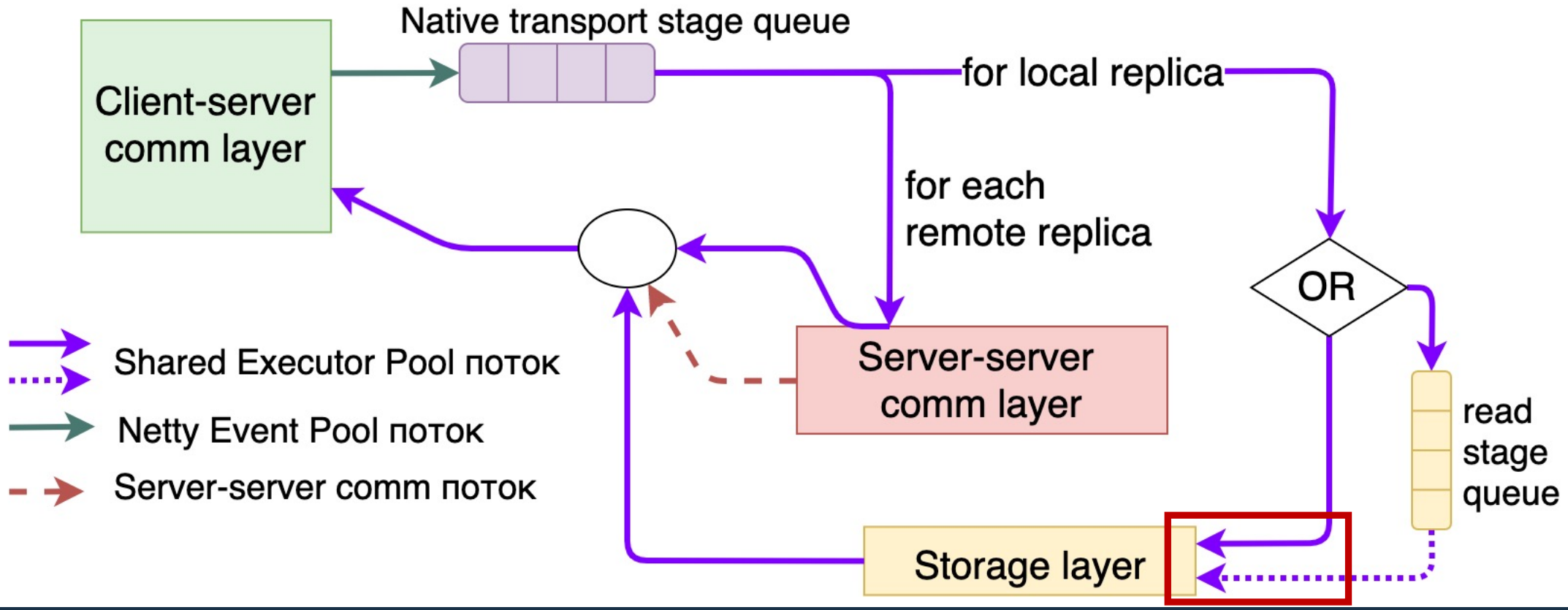
# Координация

- Потоки:
  - native_transport_max_threads: 128



# Координация

- Потоки:
  - native_transport_max_threads: 128
  - concurrent_reads: 32



- Запись, уровень кластера
- Запись, уровень реплики
- Базовый тест
- Клиент-сервер
- Координация
  - Общая последовательность
  - Авторизация

# Авторизация

| key | | value |
|---|---|---|
| User | Resource | Set<Permission> |
| Alice | data | ALTER/DROP/SELECT/MODIFY/AUTHORIZE |
| Alice | data/keyspace_name | ALTER/DROP/SELECT/MODIFY/AUTHORIZE/CREATE |
| Alice | data/keyspace_name/table_name | ALTER/DROP/SELECT/MODIFY/AUTHORIZE/CREATE |

Resources:
- Data
- Functions
- MBean (JMX)
- Roles

# Авторизация

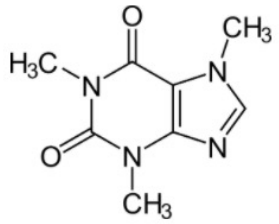
| key | | value |
|---|---|---|
| User | Resource | Set<Permission> |
| Alice | data | ALTER/DROP/SELECT/MODIFY/AUTHORIZE |
| Alice | data/keyspace_name | ALTER/DROP/SELECT/MODIFY/AUTHORIZE/CREATE |
| Alice | data/keyspace_name/table_name | ALTER/DROP/SELECT/MODIFY/AUTHORIZE/CREATE |

Resources:
- Data
- Functions
- MBean (JMX)
- Roles

- Cassandra 3.x – Guava Cache
- Cassandra 4.x – Caffeine

# https://github.com/ben-manes/caffeine

build passing | coverage 93% | maven central 3.0.1 | javadoc 3.0.1 | license apache | stack overflow caffeine



Caffeine is a high performance, near optimal caching library. For more details, see our user's guide and browse the API docs for the latest release.

## Cache

Caffeine provides an in-memory cache using a Google Guava inspired API. The improvements draw on our experience designing Guava's cache and ConcurrentLinkedHashMap.

```
LoadingCache<Key, Graph> graphs = Caffeine.newBuilder()
    .maximumSize(10_000)
    .expireAfterWrite(5, TimeUnit.MINUTES)
    .refreshAfterWrite(1, TimeUnit.MINUTES)
    .build(key -> createExpensiveGraph(key));
```

**Ben Manes**
ben-manes

Follow ···

981 followers · 1 following · ☆ 52

Mountain View, Ca
ben.manes@gmail.com
https://www.linkedin.com/in/benmanes

# Авторизация

| key | | value |
|---|---|---|
| User | Resource | Set<Permission> |
| Alice | data | ALTER/DROP/SELECT/MODIFY/AUTHORIZE |
| Alice | data/keyspace_name | ALTER/DROP/SELECT/MODIFY/AUTHORIZE/CREATE |
| Alice | data/keyspace_name/table_name | ALTER/DROP/SELECT/MODIFY/AUTHORIZE/CREATE |

Resources:
- Data
- Functions
- MBean (JMX)
- Roles

- Сброс кеша:
  - По таймауту (по умолчанию – 2 секунды)
  - JMX (org.apache.cassandra.auth.AuthCacheMBean#invalidate)

# Авторизация – накладные расходы, тест

```
CREATE ROLE test_user WITH LOGIN = true AND PASSWORD = 'password';
CREATE ROLE test_role WITH LOGIN = false;
```

# Авторизация – накладные расходы, тест

```
CREATE ROLE test_user WITH LOGIN = true AND PASSWORD = 'password';
CREATE ROLE test_role WITH LOGIN = false;
GRANT CREATE ON ALL KEYSPACES to test_role; // for stress tool only
GRANT SELECT ON KEYSPACE test_reads to test_role;
GRANT MODIFY ON KEYSPACE test_reads to test_role;
```

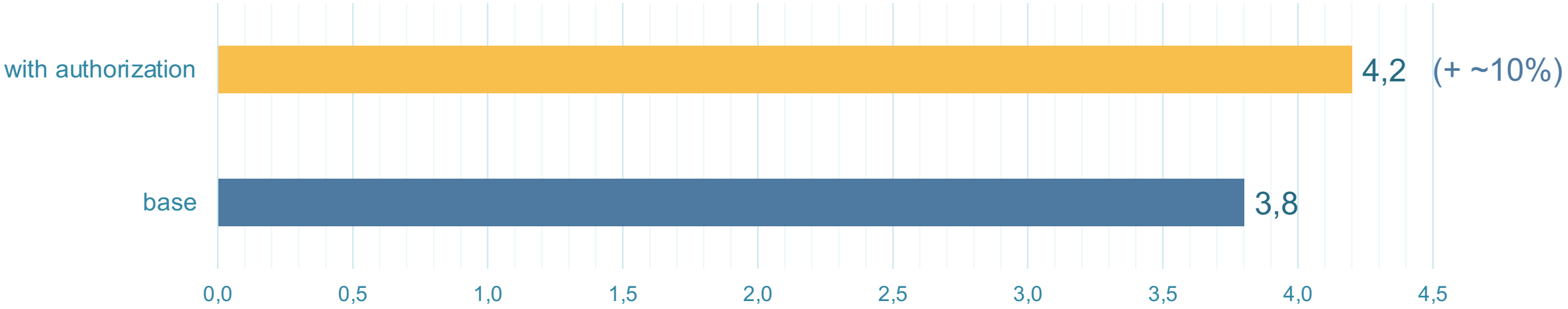
# Авторизация – накладные расходы, тест

```
CREATE ROLE test_user WITH LOGIN = true AND PASSWORD = 'password';
CREATE ROLE test_role WITH LOGIN = false;
GRANT CREATE ON ALL KEYSPACES to test_role; // for stress tool only
GRANT SELECT ON KEYSPACE test_reads to test_role;
GRANT MODIFY ON KEYSPACE test_reads to test_role;
GRANT test_role TO test_user;
```

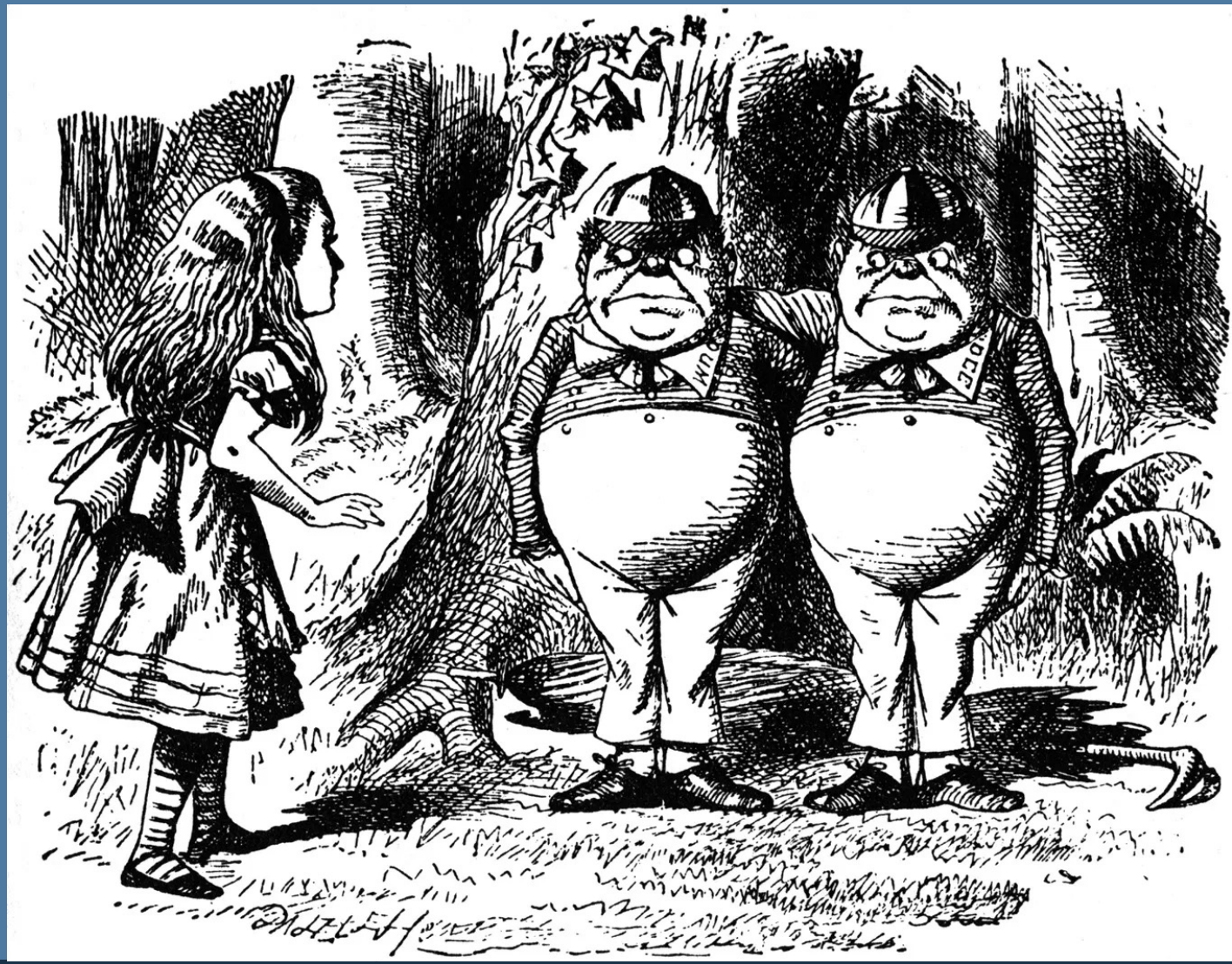
# Авторизация – накладные расходы, тест

- Результаты:

```
Latency mean                :     4.2 ms
Latency median              :     1.4 ms
Latency 95th percentile     :    20.1 ms
Latency 99th percentile     :    44.2 ms
```

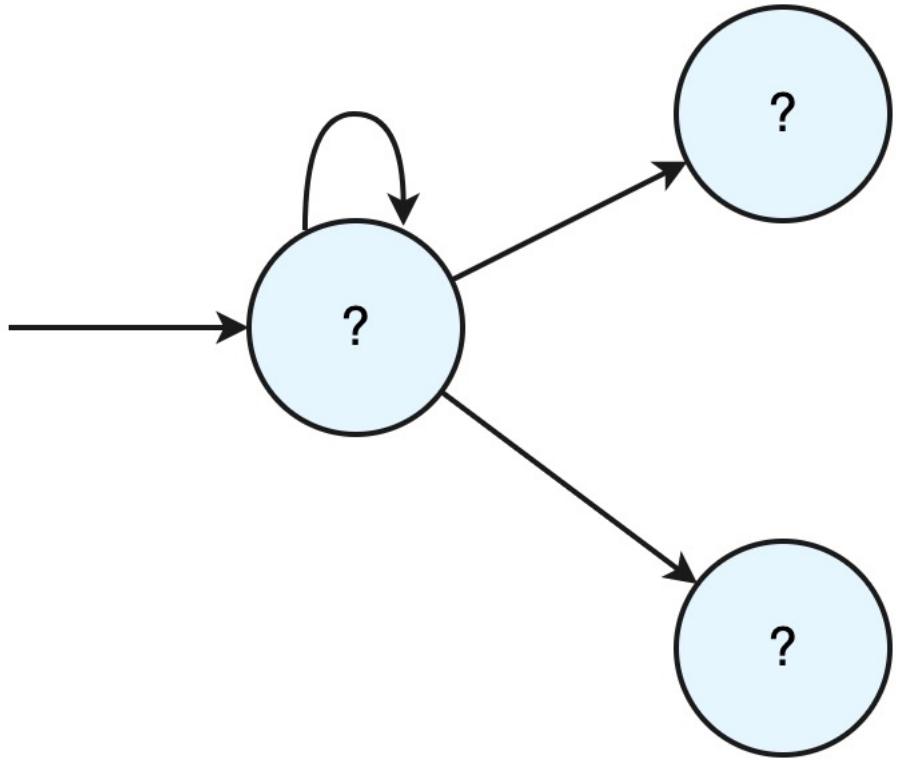


- Запись, уровень кластера
- Запись, уровень реплики
- Базовый тест
- Клиент-сервер
- Координация
  - Как устроена
  - Авторизация
  - Выбор реплик



# Координация –выбор реплик

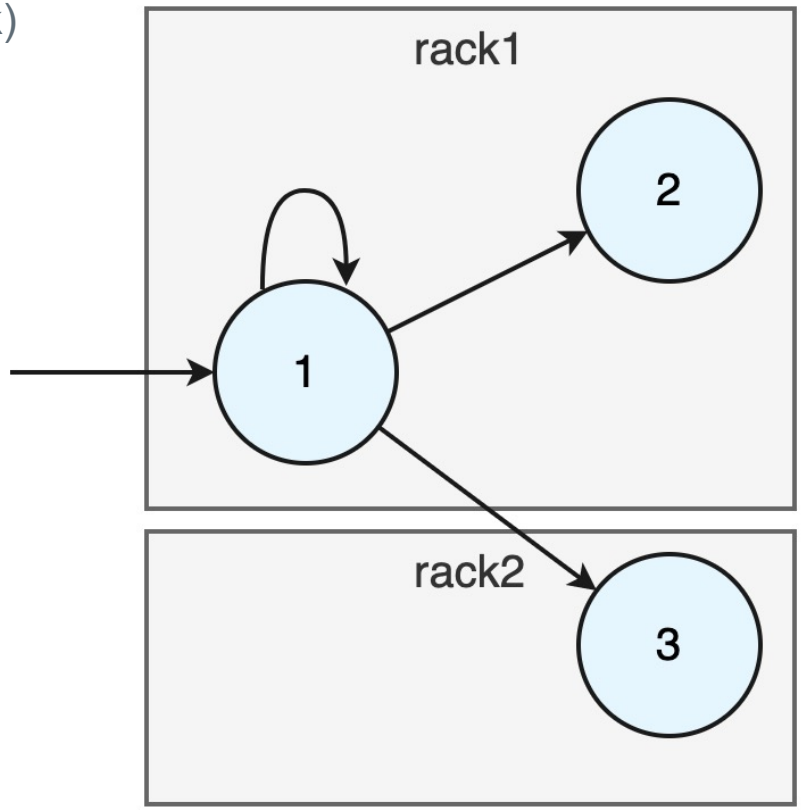
С каких реплик читать?



# Координация –выбор реплик

## Статический выбор

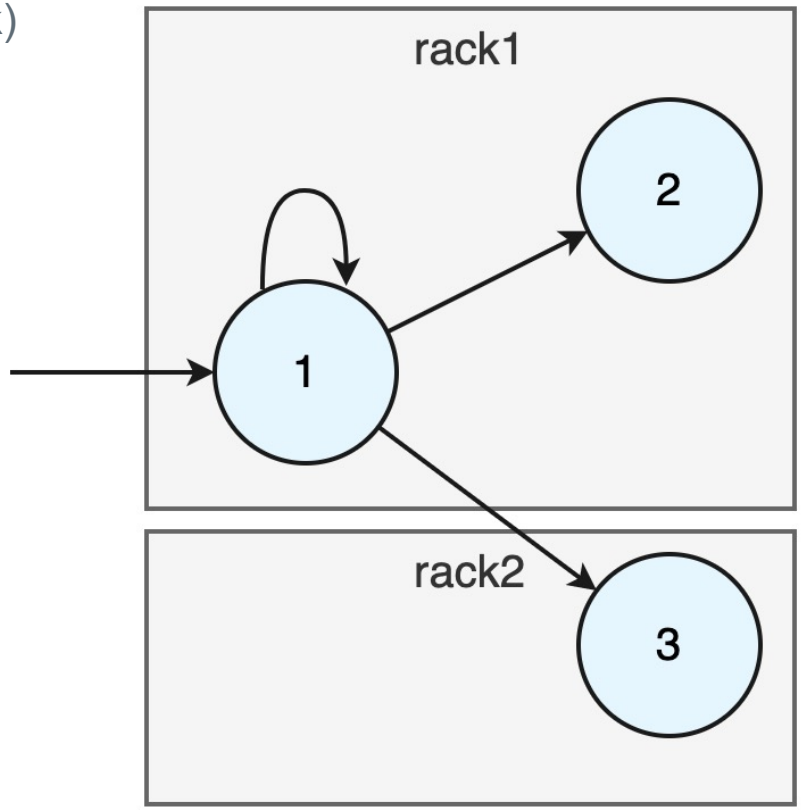
- Сортировка по степени близости к координатору
  - Текущая нода
  - В той же стойке (rack)
  - В том же датацентре

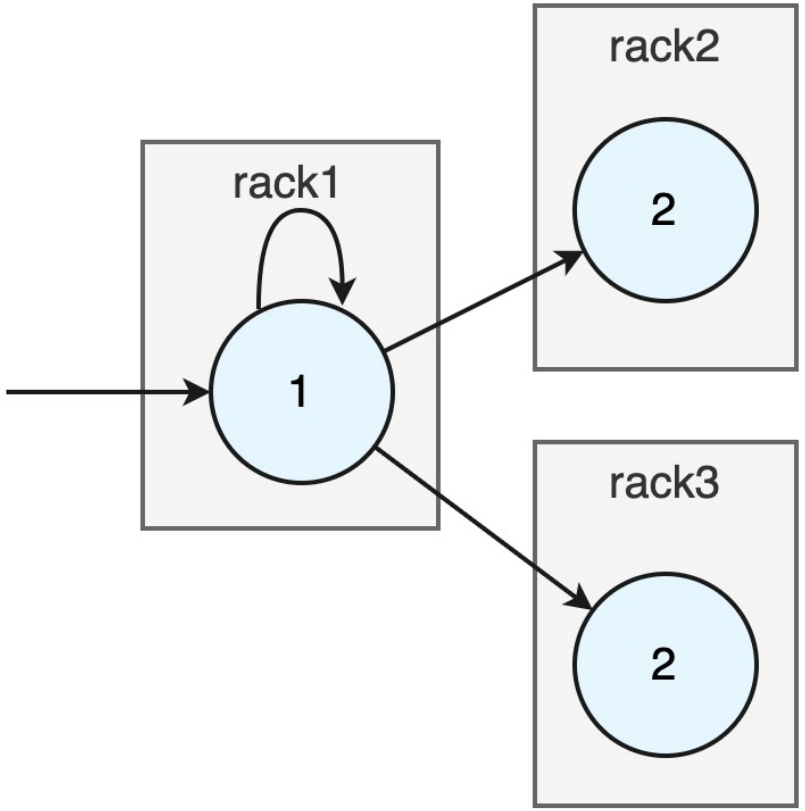


# Координация –выбор реплик

## Статический выбор

- Сортировка по степени близости к координатору
  - Текущая нода
  - В той же стойке (rack)
  - В том же датацентре



# Координация –выбор реплик

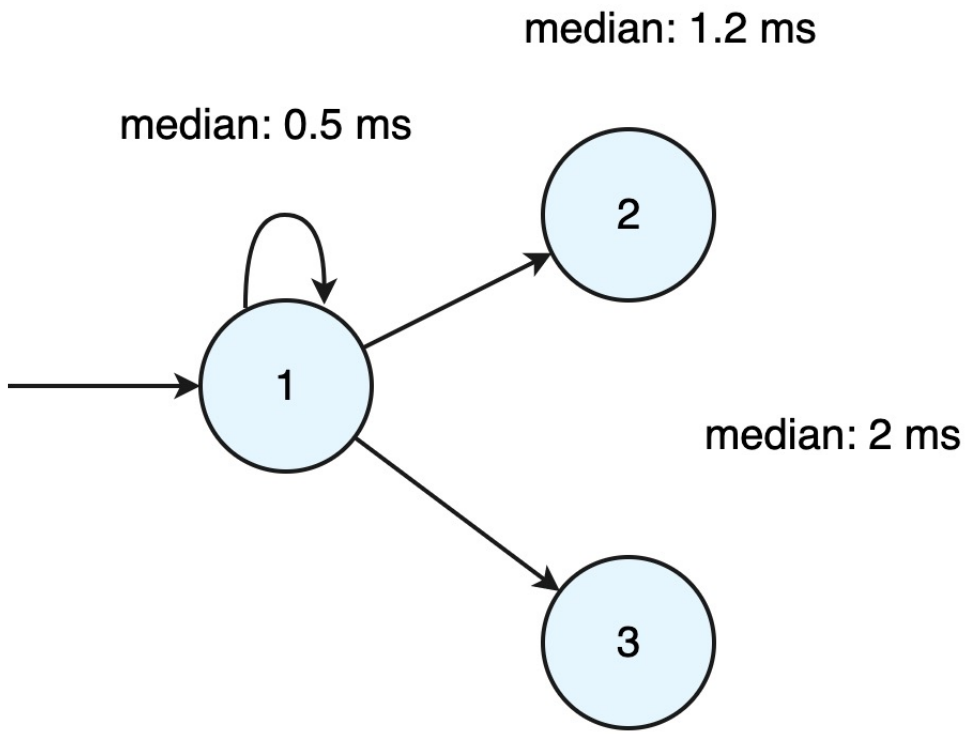
## Dynamic snitch

- Учет времени выполнения запросов на репликах
  - Метрика – медиана (exponential decay)
  - Сбрасывает периодически: dynamic_snitch_reset_interval_in_ms, 10 min

# Координация –выбор реплик

## Dynamic snitch

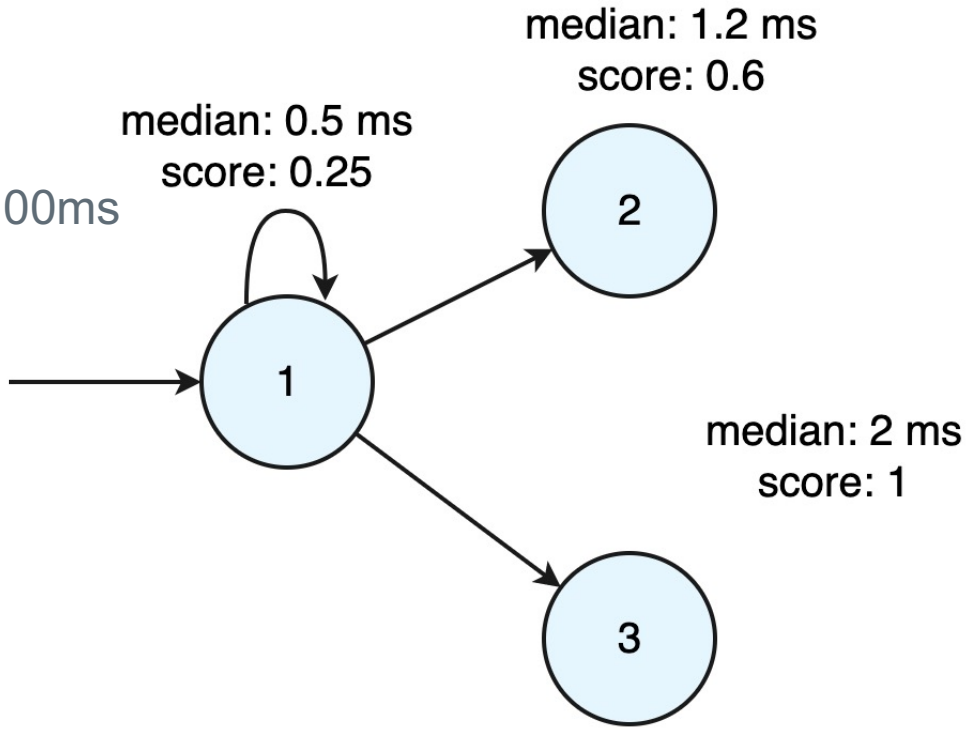
- Учет времени выполнения запросов на репликах
  - Метрика – медиана (exponential decay)
  - Сбрасывает периодически: dynamic_snitch_reset_interval_in_ms, 10 min



# Координация –выбор реплик

## Dynamic snitch

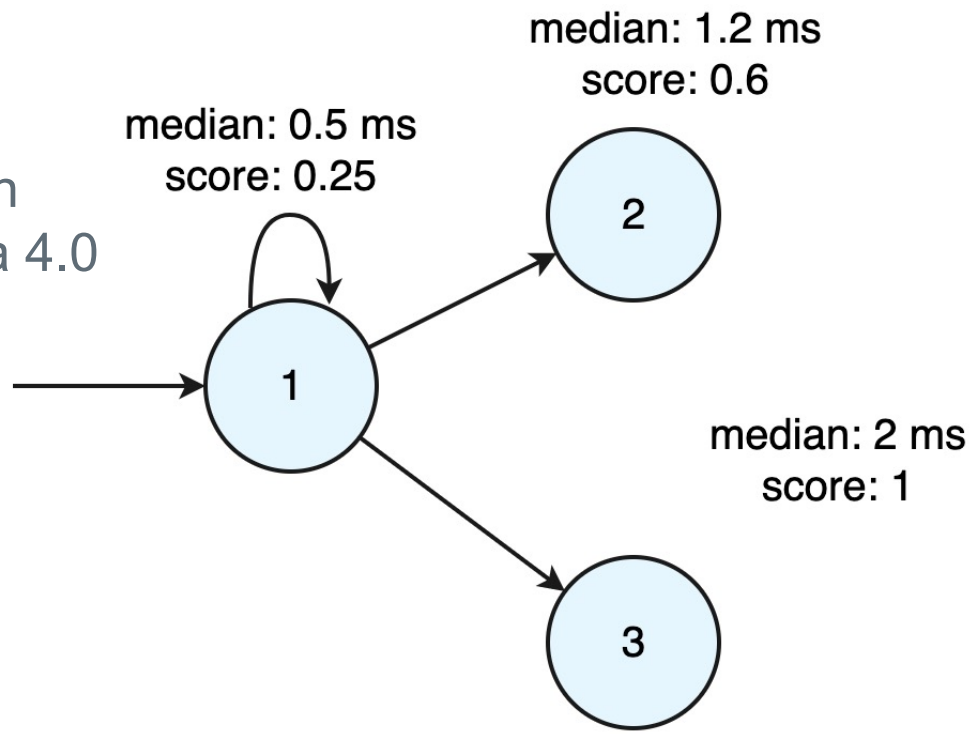
- Учет времени выполнения запросов на репликах
  - Метрика – медиана (exponential decay)
  - Сбрасывает периодически: dynamic_snitch_reset_interval_in_ms, 10 min
  - score(nodeX) = median(NodeX) / max(median)
  - Обновляется периодически dynamic_snitch_update_interval_in_ms, default = 100ms
  - Сортировка реплик по score



# Координация –выбор реплик

## Dynamic snitch

- Проблемы
  - Замедленная реакция
  - Склонен авто-колебаниям
  - CASSANDRA-6908 (Dynamic endpoint snitch destabilizes cluster under heavy load)
  - CASSANDRA-16285 (Change Dynamic Snitch Default Badness Threshold to 1.0), Cassandra 4.0



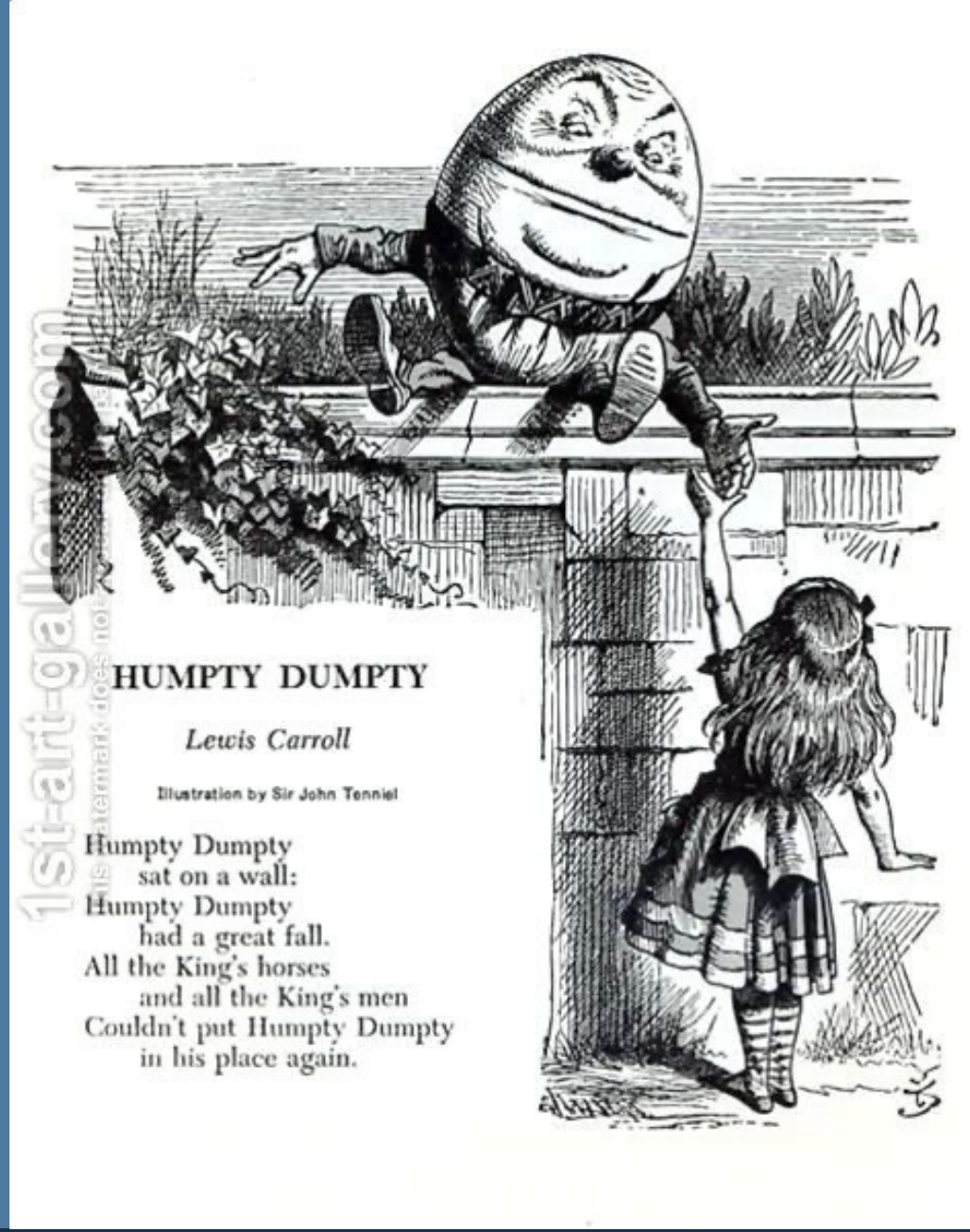
# Координация –выбор реплик

## Альтернативы – а что говорят исследователи?

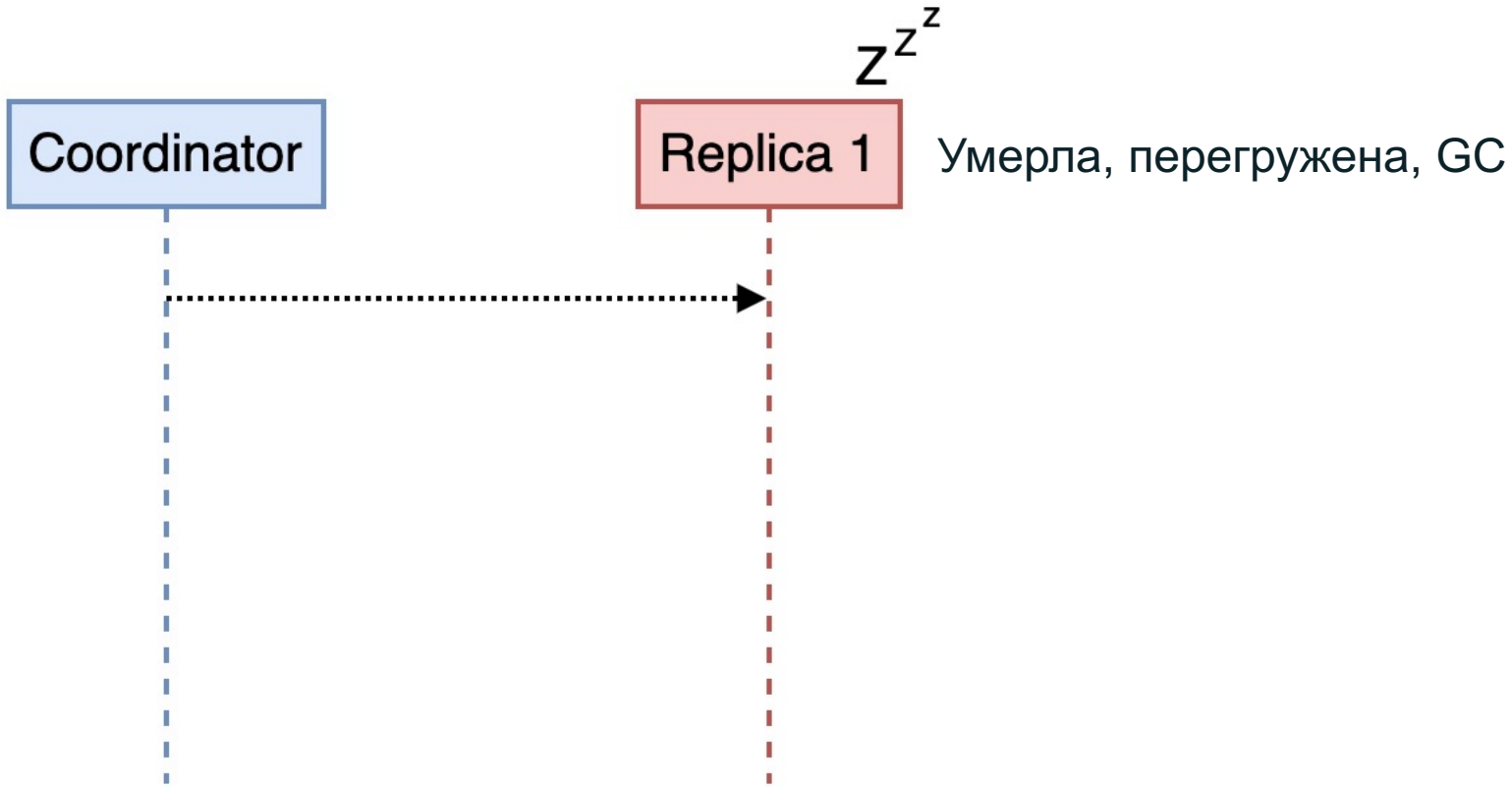
# Координация –выбор реплик

- https://www.researchgate.net/publication/343335132_A_Prediction_Based_Replica_Selection_Strategy_for_Reducing_Tail_Latency_in_Distributed_Systems
- https://www.usenix.org/system/files/conference/nsdi15/nsdi15-paper-suresh.pdf (C3: Cutting Tail Latency in Cloud Data Stores via Adaptive Replica Selection)
  http://www.diva-portal.org/smash/get/diva2:827372/FULLTEXT01 (Replica selection in Apache Cassandra)
  https://www.univie.ac.at/ct/stefan/nsdi15_slides_suresh.pdf
  https://www.elastic.co/blog/improving-response-latency-in-elasticsearch-with-adaptive-replica-selection
- https://digitalcommons.unl.edu/cgi/viewcontent.cgi?article=1184&context=computerscidiss (Reducing the Tail Latency of a Distributed NoSQL Database)
- https://hal.archives-ouvertes.fr/hal-01896686/document (Heron: Taming Tail Latencies in Key-Value Stores under Heterogeneous Workloads)
- https://tel.archives-ouvertes.fr/tel-02301338/document (Improving Performance Predictability in Cloud Data Stores)
- https://cris.maastrichtuniversity.nl/ws/portalfiles/portal/48876879/Jaiman_DAIS20.pdf (TailX: Scheduling Heterogeneous Multiget Queries to Improve Tail Latencies in Key-Value Stores)
- https://people.kth.se/~dejanko/documents/publications/rein-eurosys17.pdf (Rein: Taming Tail Latency in Key-Value Stores via Multiget Scheduling)
- etc.

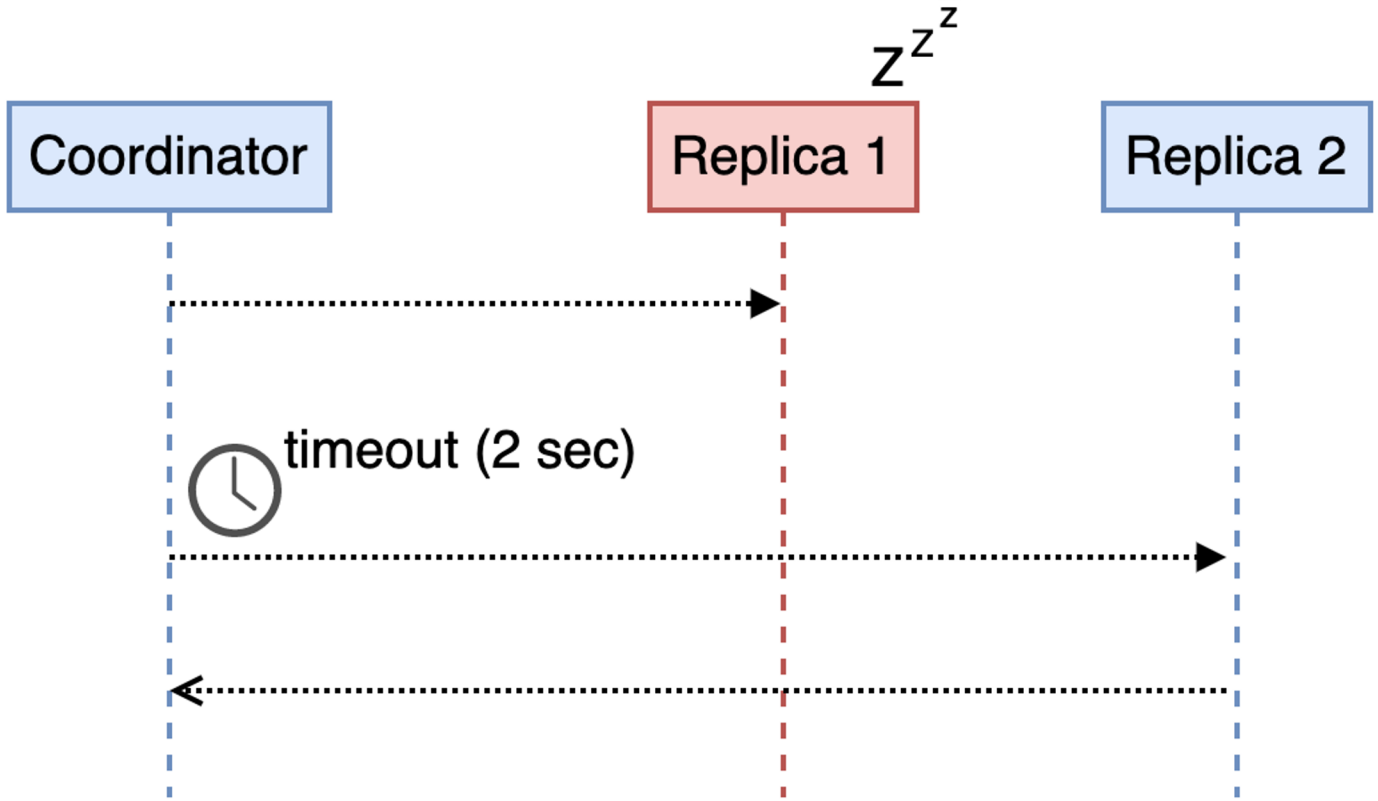
- Запись, уровень кластера
- Запись, уровень реплики
- Базовый тест
- Клиент-сервер
- Координация
  - Как устроена
  - Авторизация
  - Выбор реплик
  - Медленная реплика



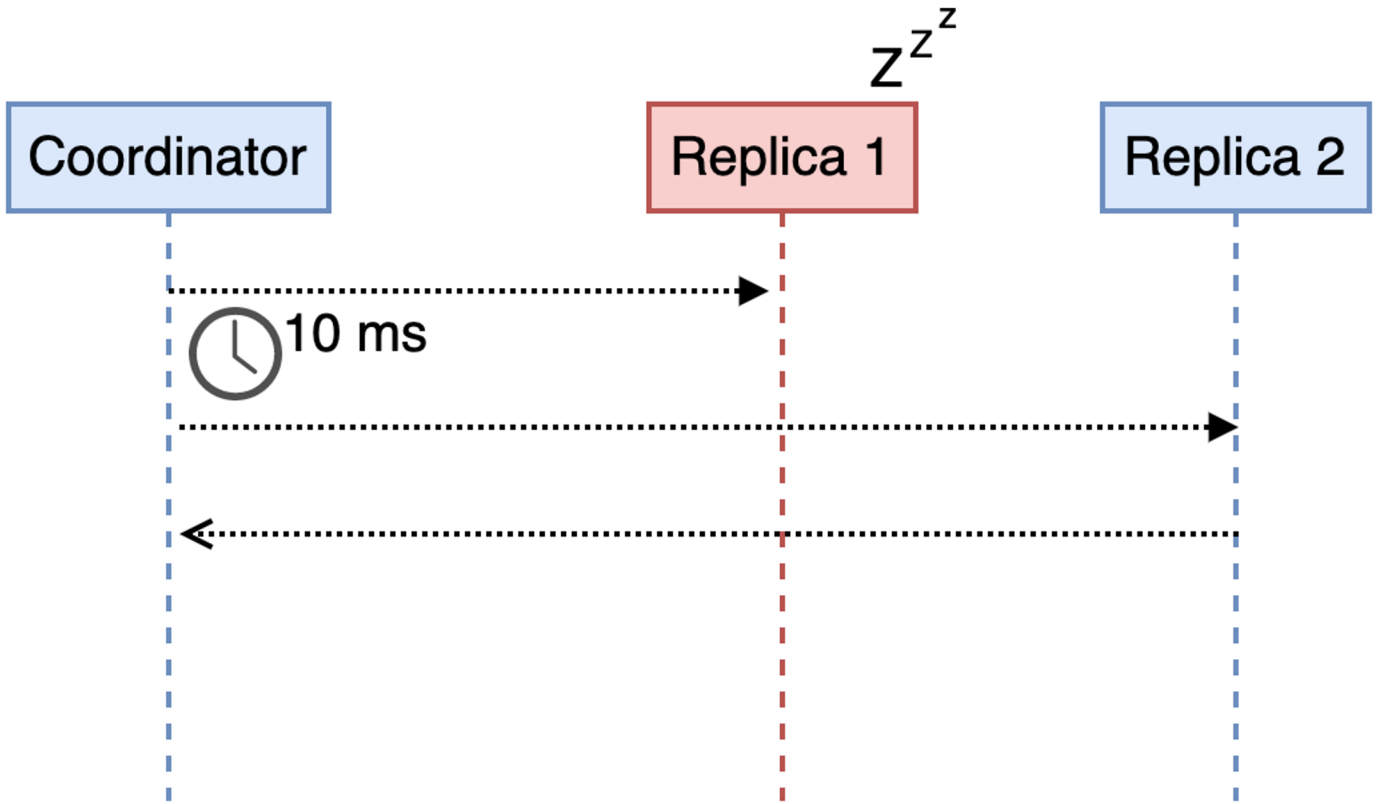
# Координация – а если реплика тупит



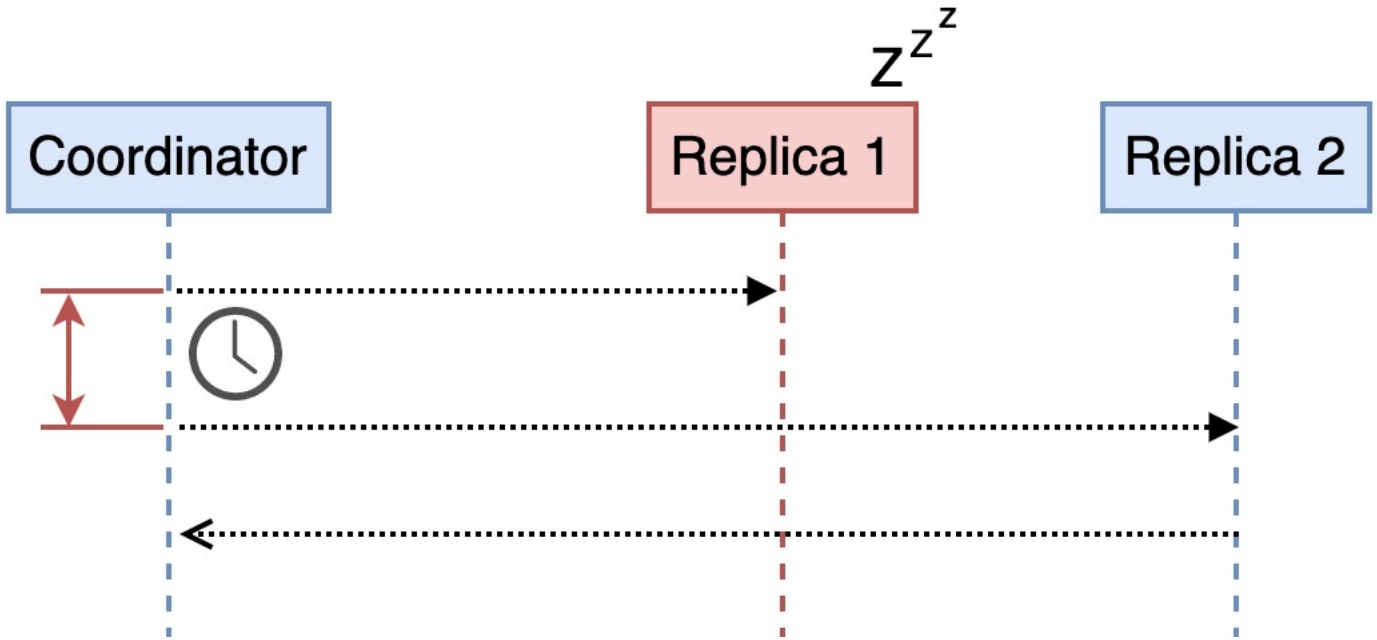
# Координация – повтор после таймаута?




# Координация – спекулятивный повтор

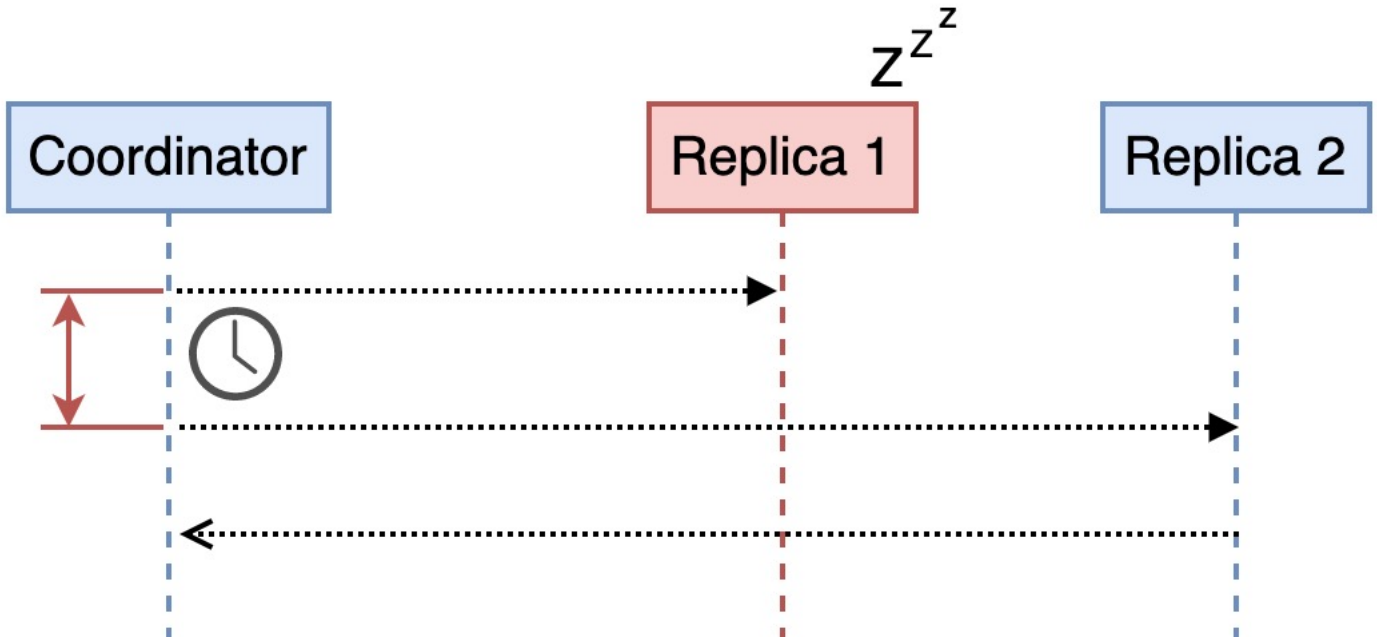



# Координация – спекулятивный повтор




- Настраивается на уровне таблицы (по умолчанию включено)
- Стратегии (через какой период времени делать спекулятивный повтор):
  - Percentile
  - Constant

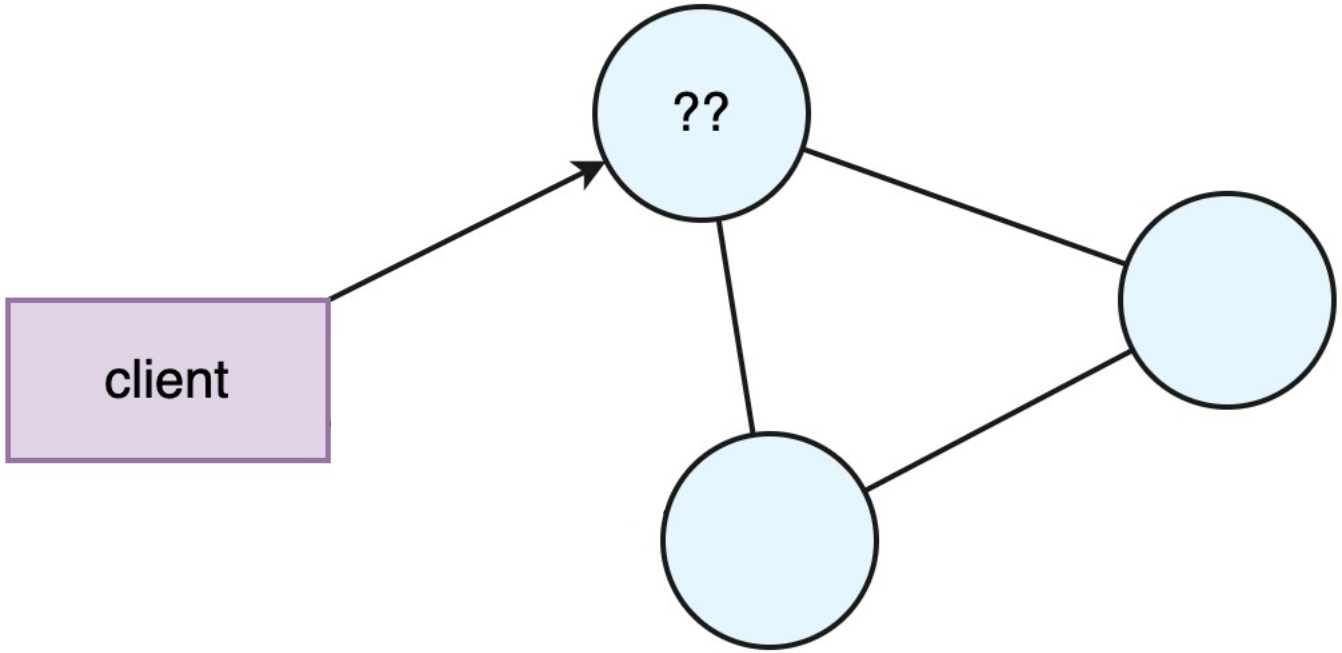
# Координация – спекулятивный повтор



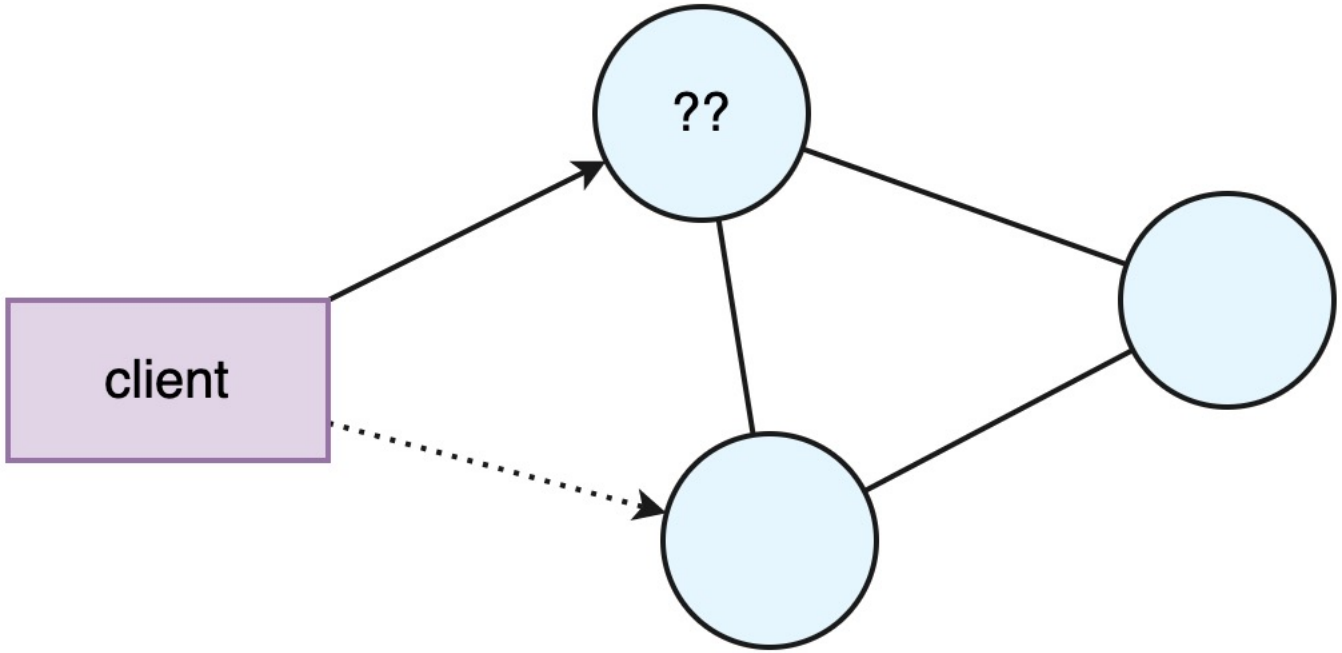

- Настраивается на уровне таблицы (по умолчанию включено)
- Стратегии (через какой период времени делать спекулятивный повтор):
  - Percentile
  - Constant

https://www.datastax.com/blog/rapid-read-protection-cassandra-202

# Координация – спекулятивный повтор для клиента

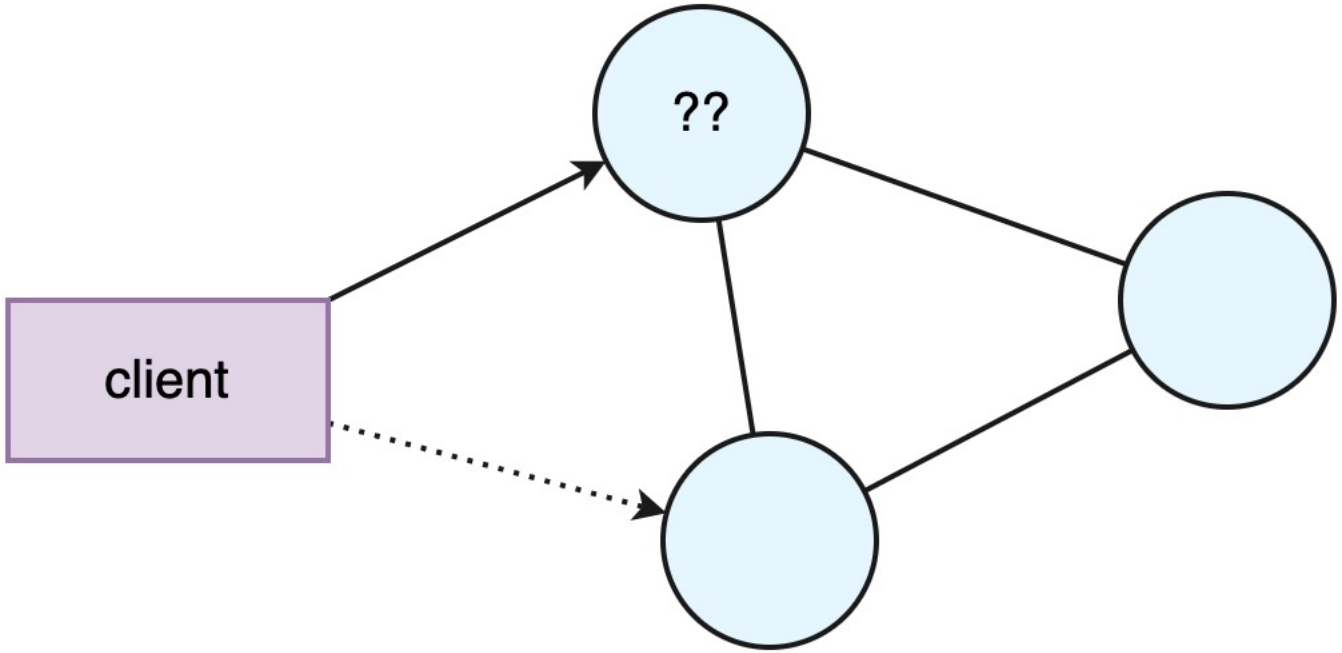


# Координация – спекулятивный повтор для клиента




- Аналогичная логика есть и на уровне драйвера
- Применима не только для чтений, но и для записи

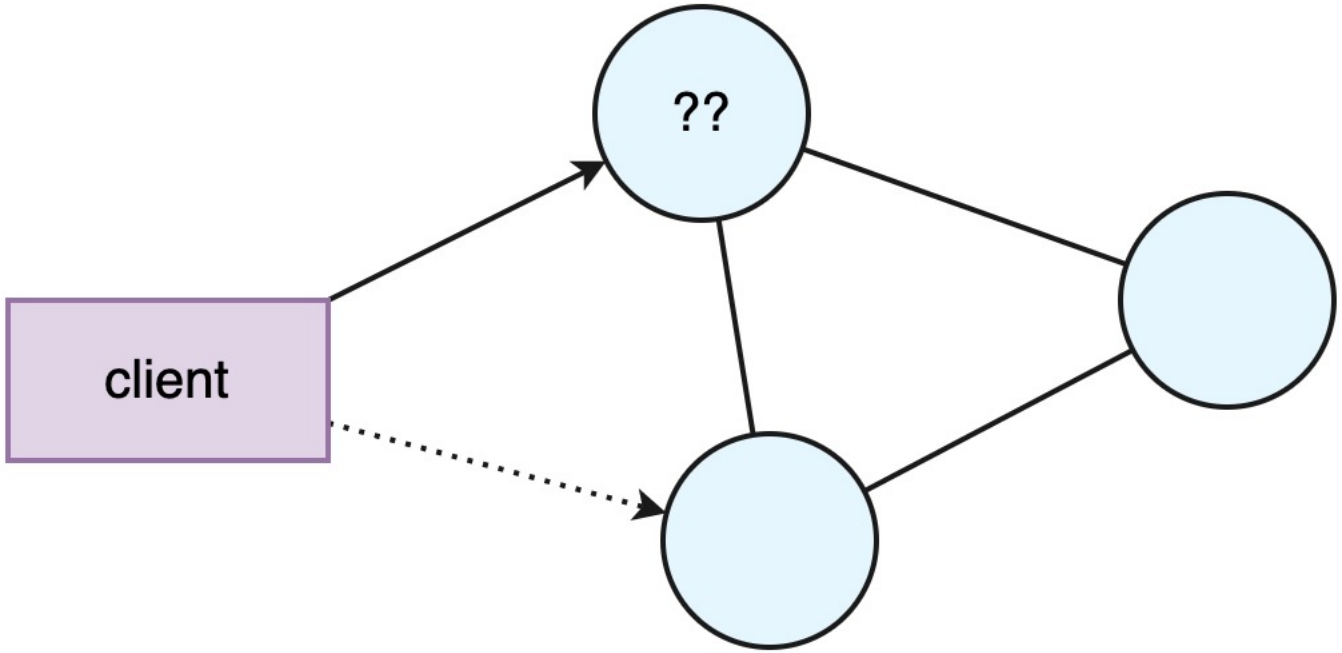
# Координация – спекулятивный повтор для клиента




- Аналогичная логика есть и на уровне драйвера
- Применима не только для чтений, но и для записи
- Нужно включить idempotency flag

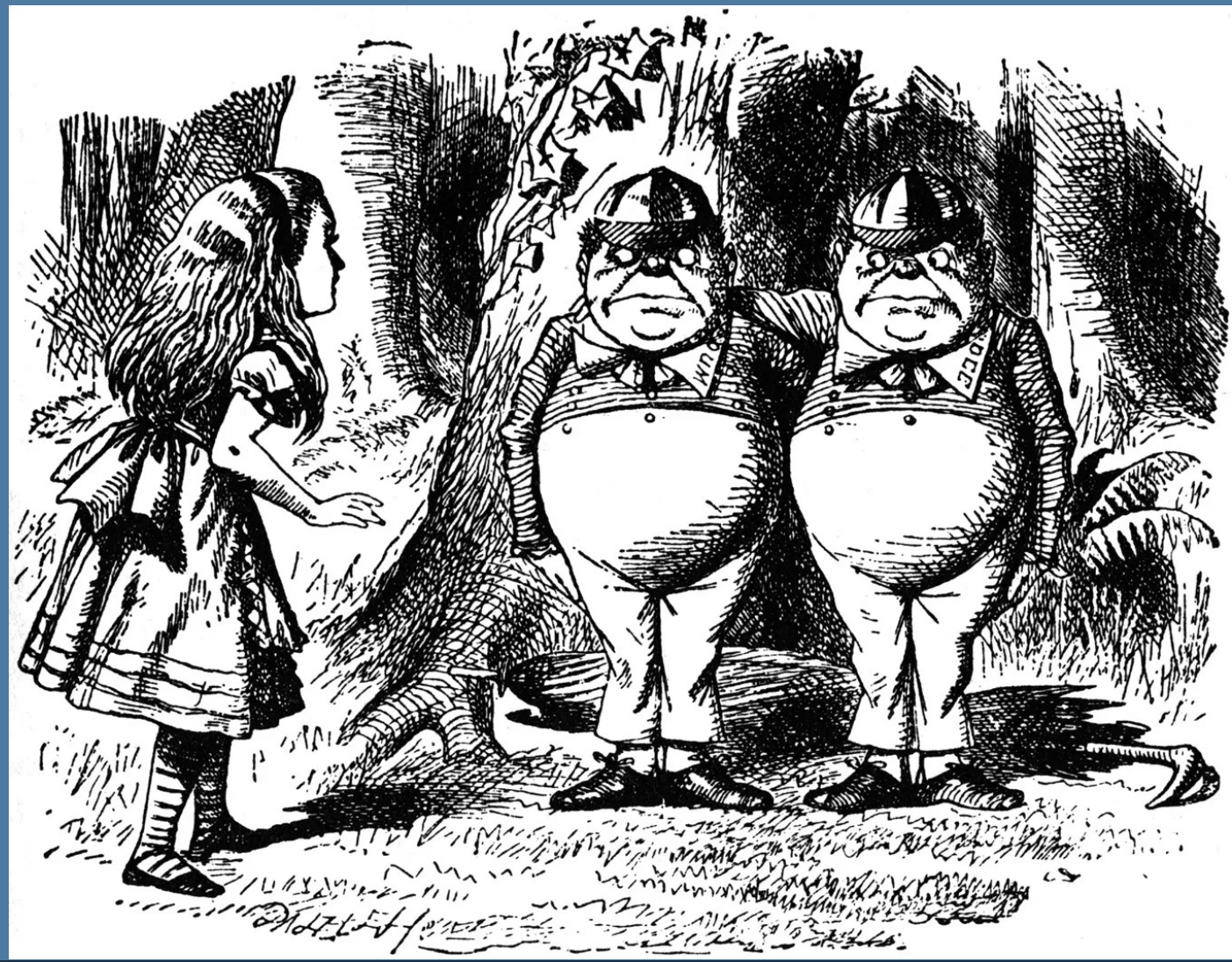
Олег Анастасьев:
«В любой непонятной ситуации выполняй идемпотентные операции»

# Координация – спекулятивный повтор для клиента

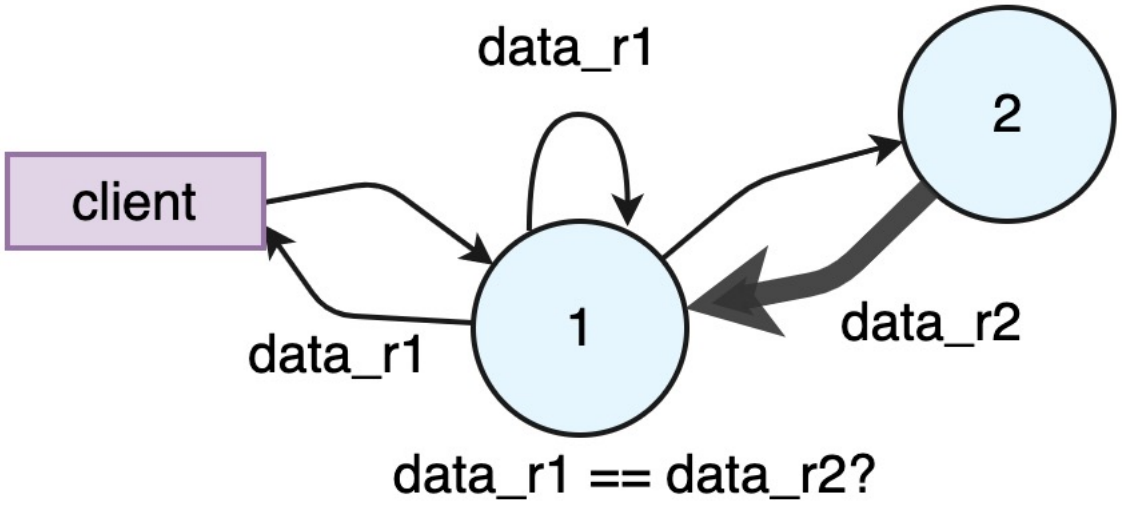



- Аналогичная логика есть и на уровне драйвера
- Применима не только для чтений, но и для записи
- Нужно включить idempotency flag
- Можно реализовать свою стратегию - через какое время отправлять speculative запрос
  - com.datastax.driver.core.policies.SpeculativeExecutionPolicy

- Запись, уровень кластера
- Запись, уровень реплики
- Базовый тест
- Клиент-сервер
- Координация
  - Как устроена
  - Авторизация
  - Выбор реплик
  - Спекулятивные повторы
  - Сверка ответов от реплик

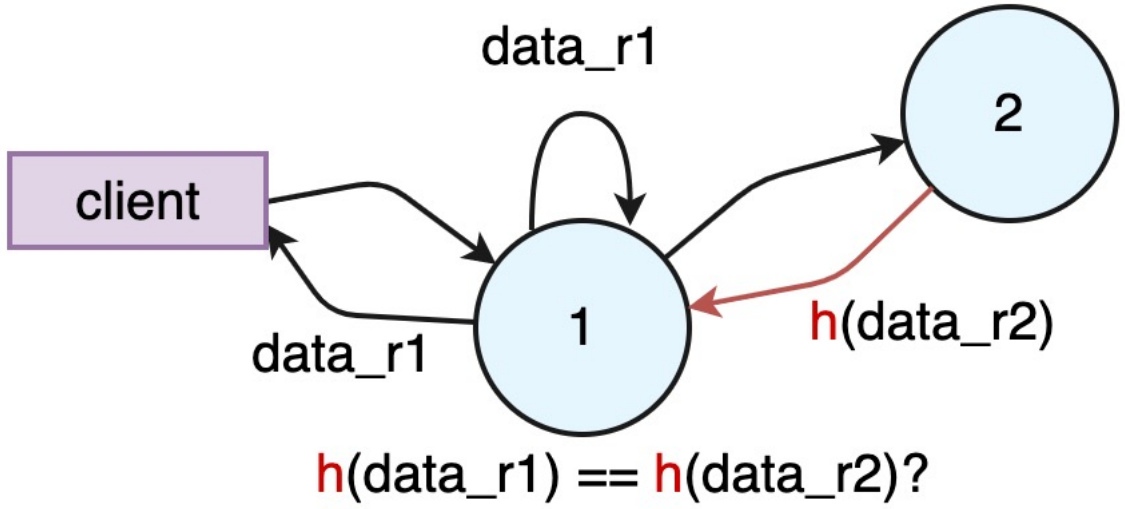


# Координация – сверка ответов от реплик




Data_r2 – сериализуем и передаем по сети весь результат выборки

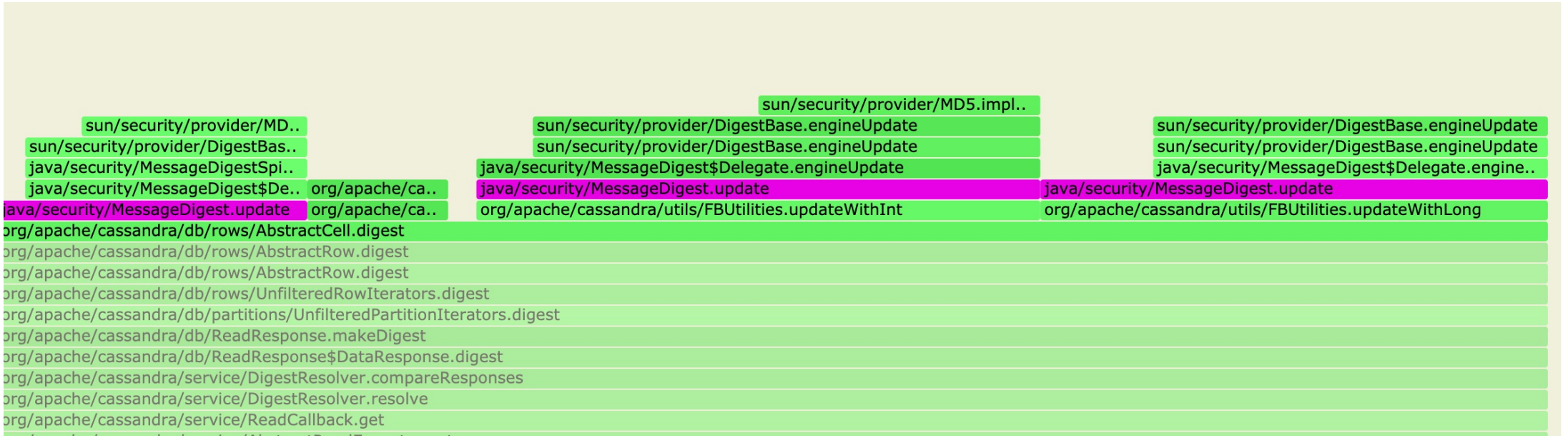
# Координация – сверка ответов от реплик




Data_r2 – сериализуем и передаем по сети только короткий хеш

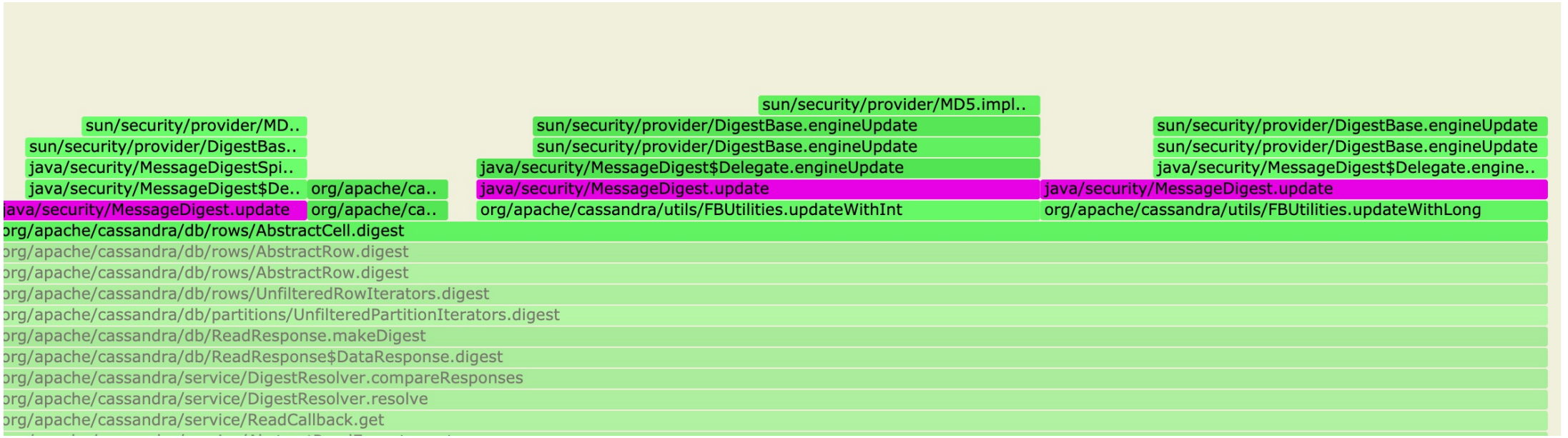
# Координация – сверка ответов от реплик

- MD5 ☹, 1.8% в CPU flamegraph



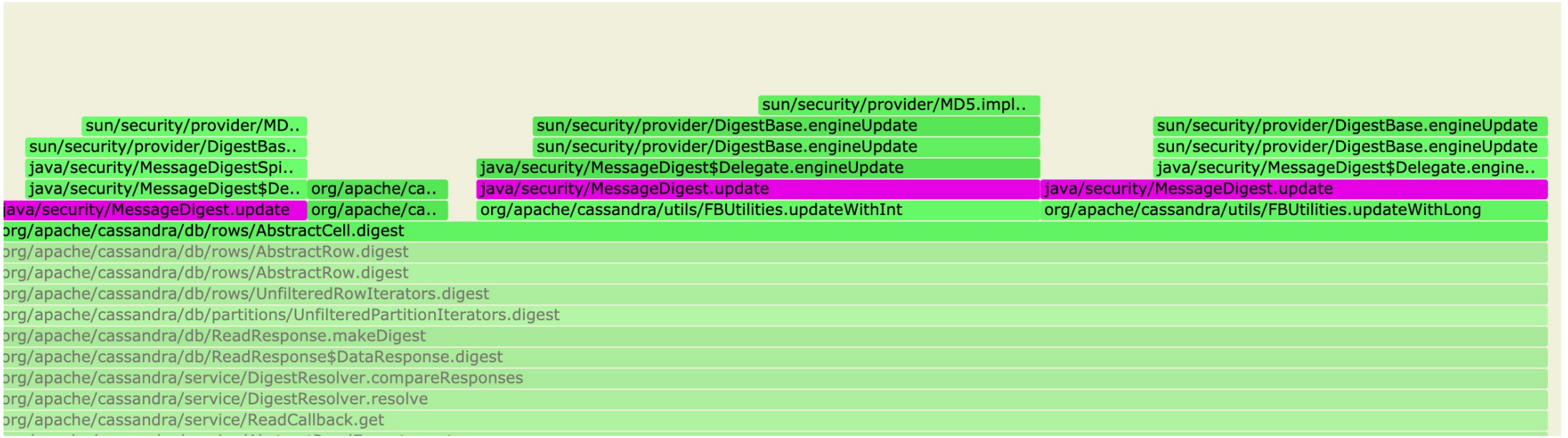
# Координация – сверка ответов от реплик

- MD5 ☹, 1.8% в CPU flamegraph
- CASSANDRA-13291 (Cassandra 4.0) – ввели слой абстракции (Guava Hasher)



# Координация – сверка ответов от реплик

- MD5 ☹, 1.8% в CPU flamegraph
- CASSANDRA-13291 (Cassandra 4.0) – ввели слой абстракции (Guava Hasher)
  - но реализация все равно MD5 ☹



https://github.com/OpenHFT/Zero-Allocation-Hashing

# Zero-Allocation Hashing

## Version



## Overview

This project provides a Java API for hashing any sequence of bytes in Java, including all kinds of primitive arrays, buffers, `CharSequence`s and more.

Written for Java 7+ under Apache 2.0 license.

The key difference compared to other similar projects, e.g. Guava hashing, is that this has no object allocation during the hash computation and does not use `ThreadLocal`.

# Координация – сверка ответов от реплик

- MD5 intrinsic ?

# Координация – сверка ответов от реплик

- MD5 intrinsic ?
- А в JDK 14 MD5 стал больше не обязателен к реализации согласно MissageDigest JavaDoc …

13/docs/api/java.base/java/security/MessageDigest.html

Every implementation of the Java platform is required to support the following standard `MessageDigest` algorithms:

- `MD5`
- `SHA-1`
- `SHA-256`

14/docs/api/java.base/java/security/MessageDigest.html

Every implementation of the Java platform is required to support the following standard `MessageDigest` algorithms:

- `SHA-1`
- `SHA-256`

# Координация – сверка ответов от реплик

- MD5 intrinsic ?
- А в JDK 14 MD5 стал больше не обязателен к реализации согласно MissageDigest JavaDoc …
- JDK-8250902 (JDK 16)

JDK / JDK-8250902

## Implement MD5 Intrinsics on x86

**Details**

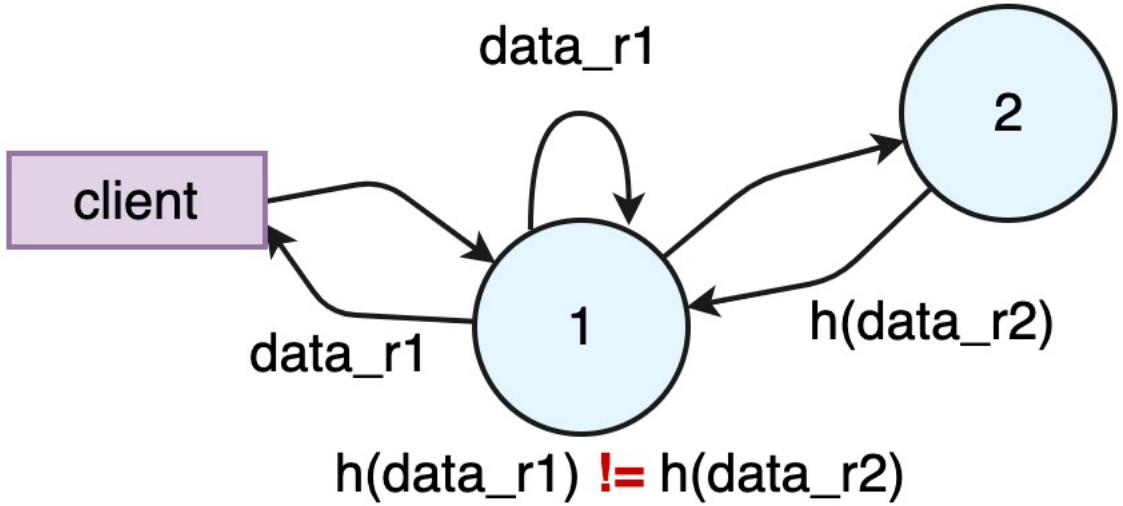
| | | | |
|---|---|---|---|
| Type: | Enhancement | Status: | RESOLVED |
| Priority: | P4 | Resolution: | Fixed |
| Affects Version/s: | 16 | Fix Version/s: | 16 |
| Component/s: | hotspot | | |
| Labels: | intrinsic | | |
| Subcomponent: | compiler | | |
| Resolved In Build: | b10 | | |
| CPU: | x86, x86_64 | | |
| OS: | generic | | |

**Backports**

| Issue | Fix Version | Assignee | Priority | Status | Resolution | Resolved In Build |
|---|---|---|---|---|---|---|
| JDK-8251319 | 11-pool | Ludovic Henry | P4 | Open | Unresolved | |

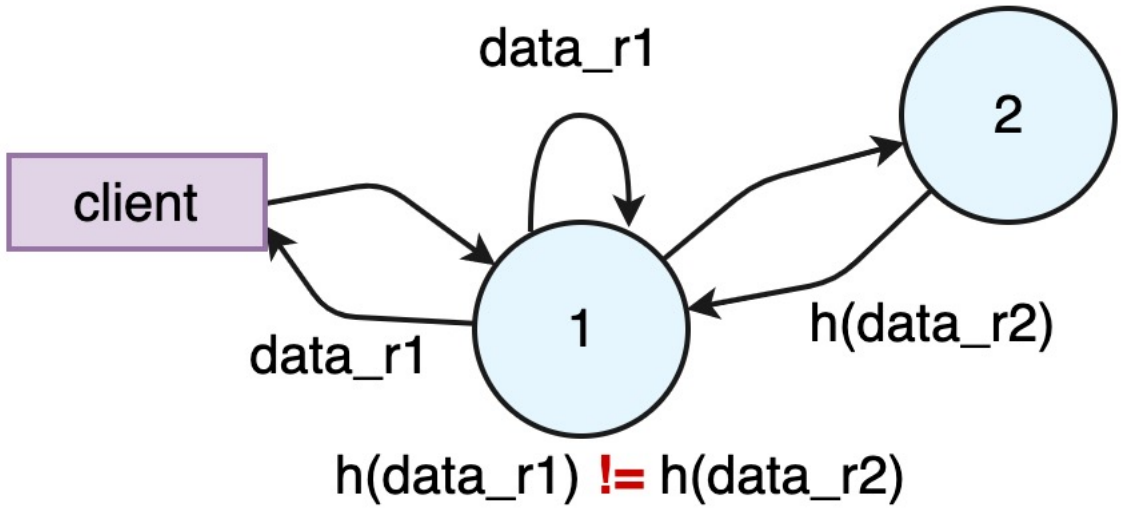
- Запись, уровень кластера
- Запись, уровень реплики
- Базовый тест
- Клиент-сервер
- Координация
  - Как устроена
  - Авторизация
  - Выбор реплик
  - Спекулятивные повторы
  - Сверка ответов от реплик
  - Read repair

# Координация – read repair



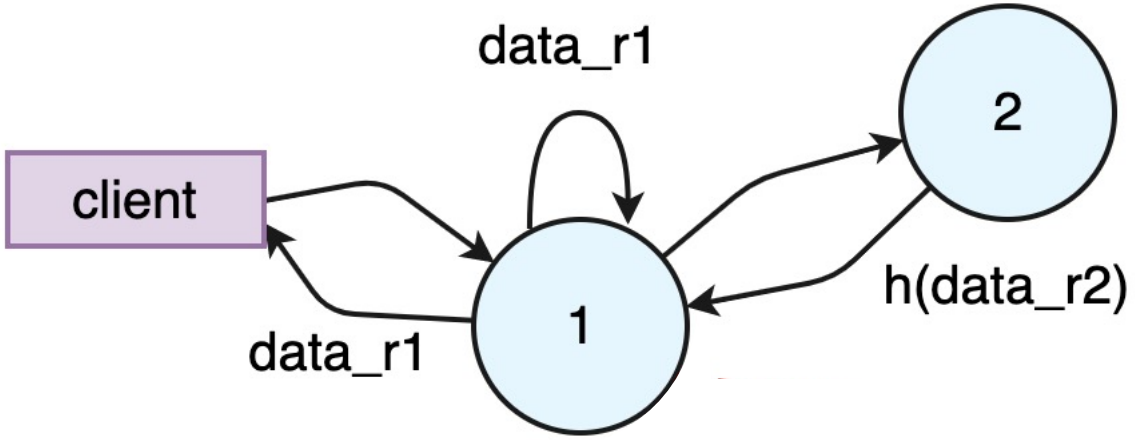

- Данные на репликах разошлись

# Координация – read repair



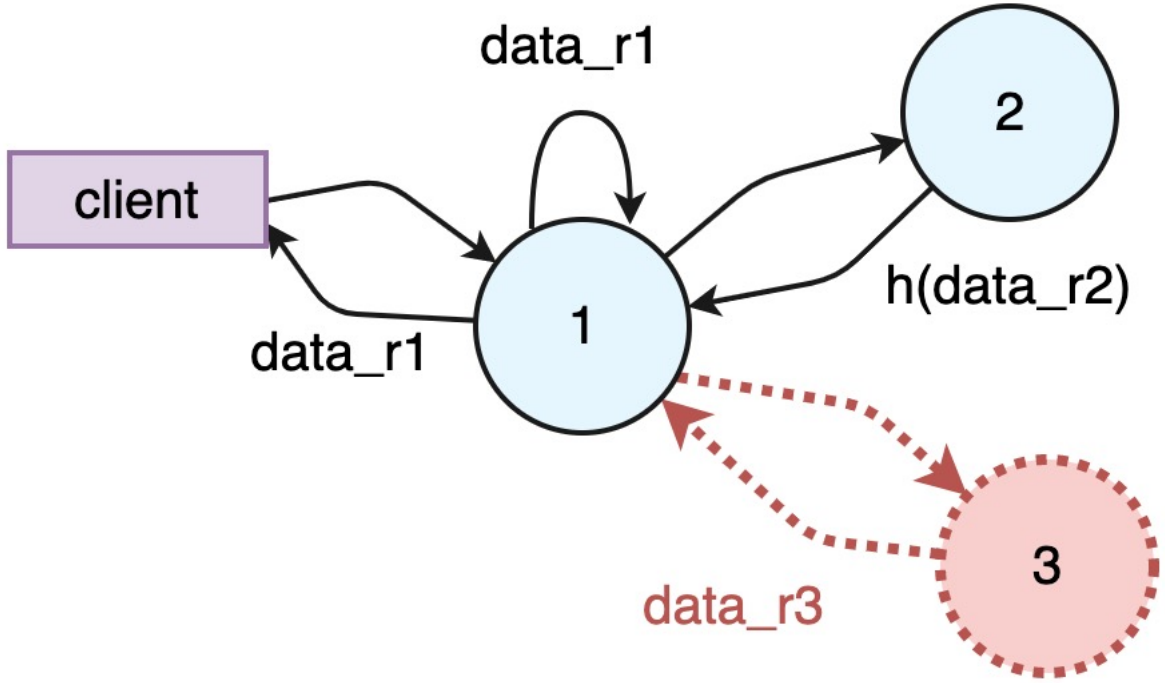

- Данные на репликах разошлись
- Придется прочитать и смержить данные с тех же реплик, а затем записать результат обратно – read repair
- Read repair – блокирующая операция, ответ клиенту отсылается только после ее завершения

# Координация – background read repair



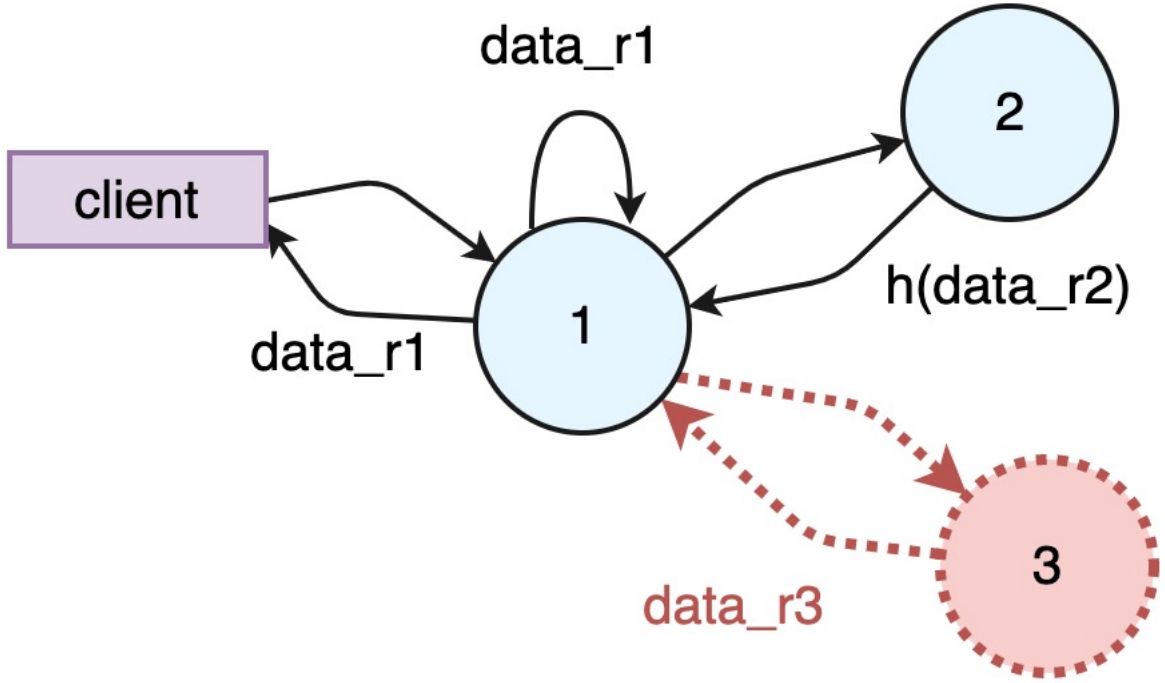

- dclocal_read_repair_chance (0.1) – вероятность того, что координатор отошлет запросы на дополнительные реплики для async read repair

# Координация – background read repair




- dclocal_read_repair_chance (0.1) – вероятность того, что координатор отошлет запросы на дополнительные реплики для async read repair
- Set dclocal_read_repair_chance = 0.0

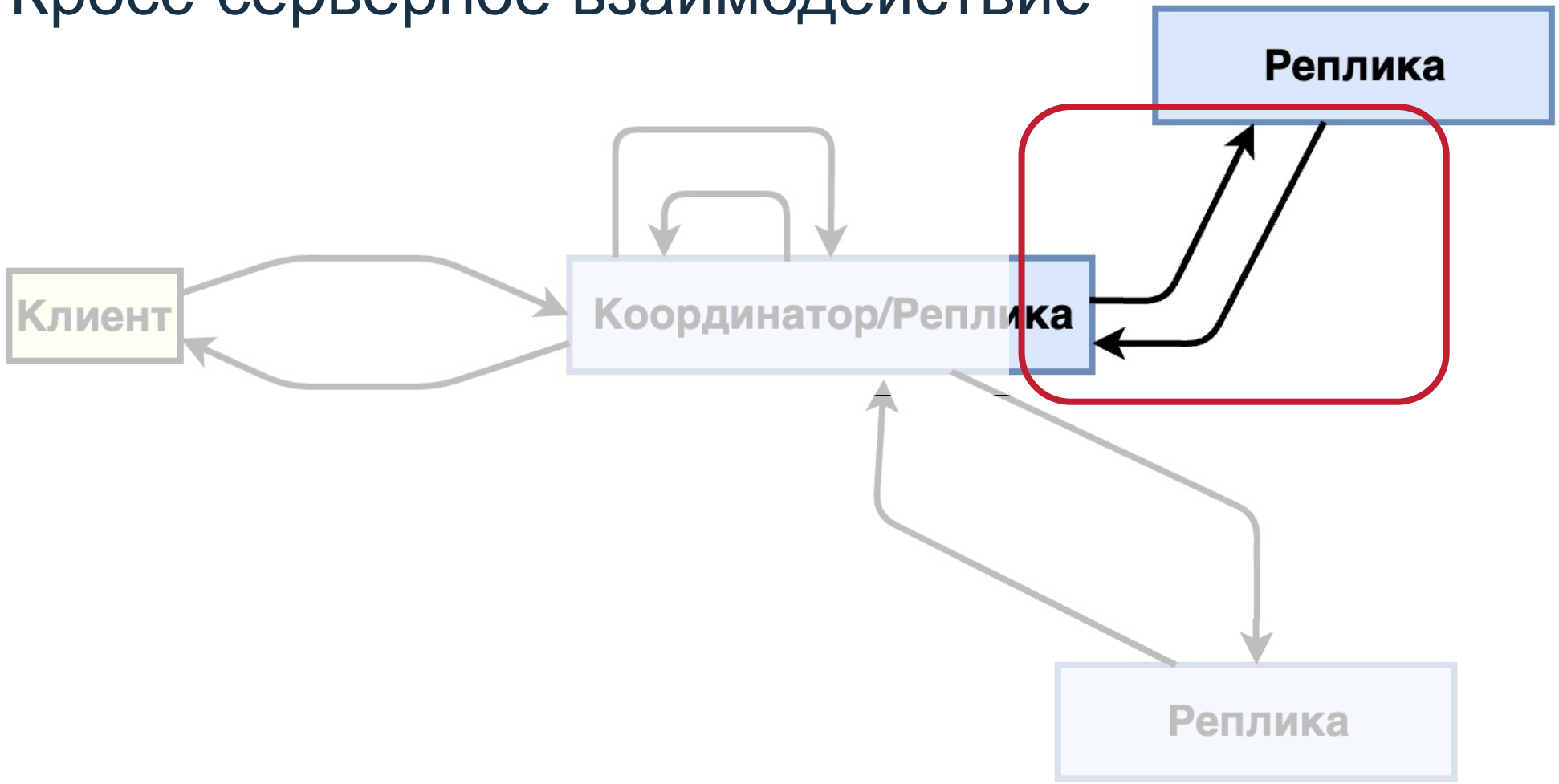
# Координация – background read repair




- dclocal_read_repair_chance (0.1) – вероятность того, что координатор отошлет запросы на дополнительные реплики для async read repair
- Set dclocal_read_repair_chance = 0.0
- CASSANDRA-13910 (Cassandra 4.0) – Remove read_repair_chance / dclocal_read_repair_chance
- https://thelastpickle.com/blog/2021/01/12/get_rid_of_repair_repair_chance.html

- Запись, уровень кластера
- Запись, уровень реплики
- Базовый тест
- Клиент-серверное взаимодействие
- Координация
- Кросс-серверное взаимодействие

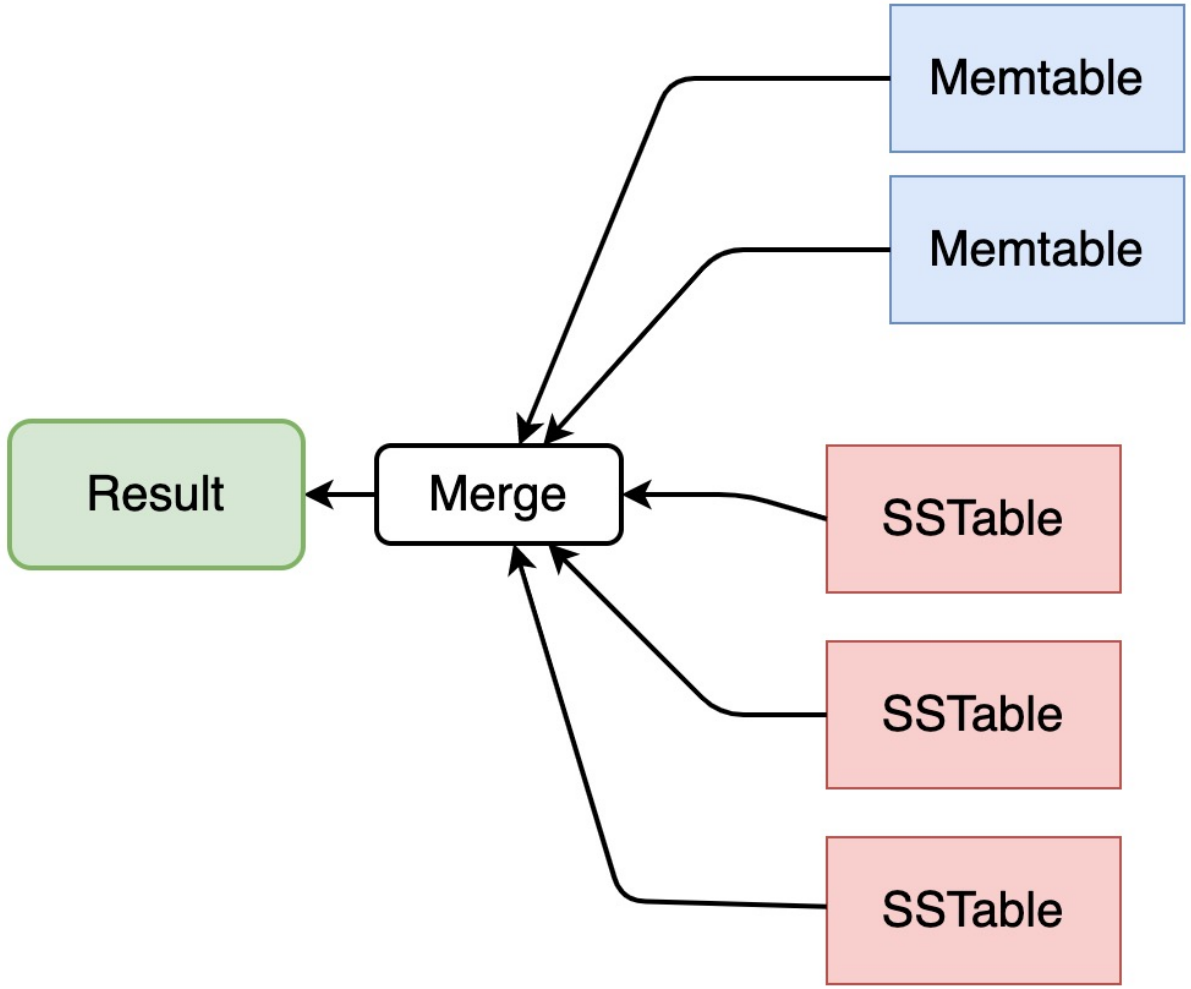
# Кросс-серверное взаимодействие




- Та же логика, что и для записей - ”Запись в Cassandra — внутреннее устройство и производительность”, Jpoint 2020

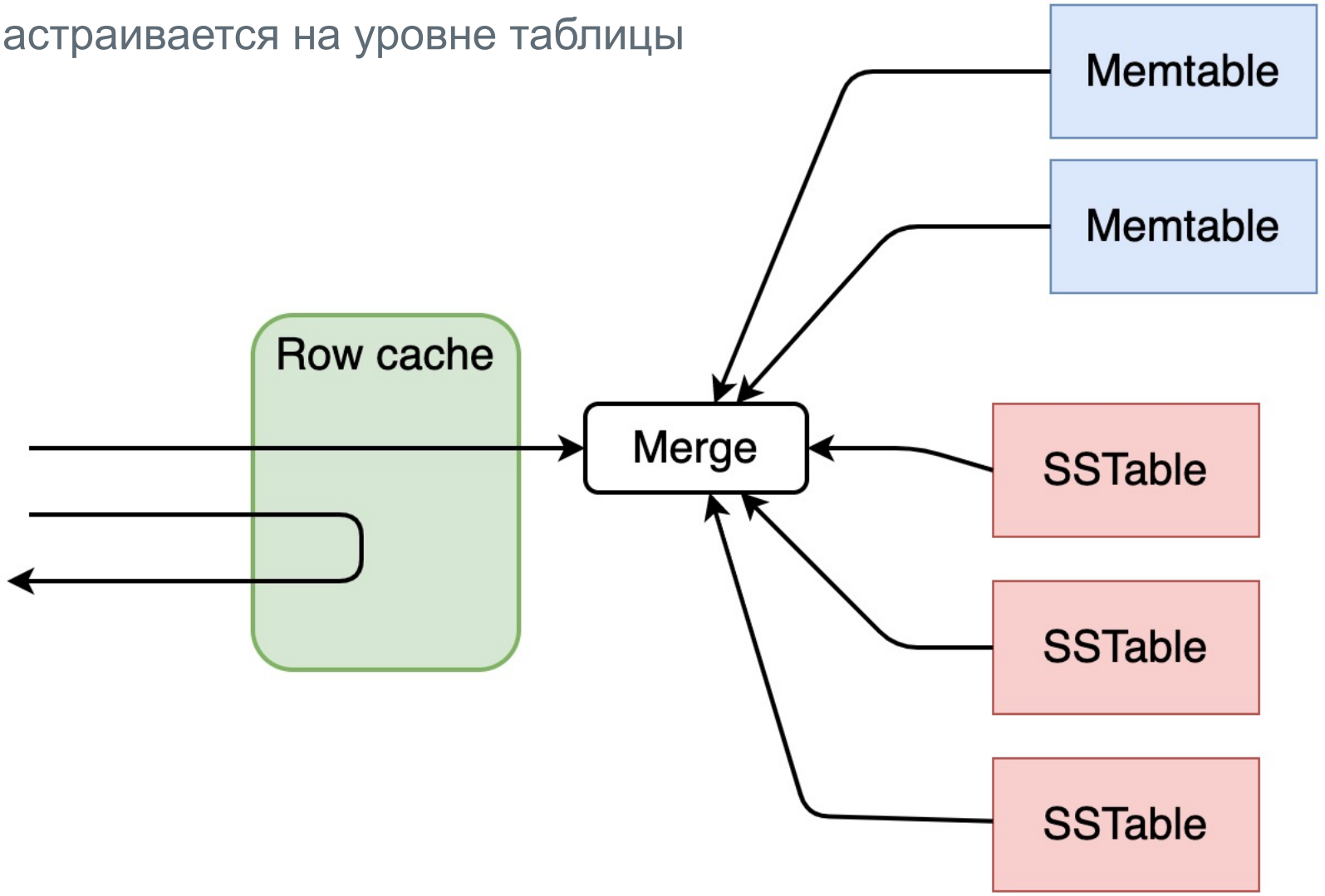
- Запись, уровень кластера
- Запись, уровень реплики
- Базовый тест
- Клиент-сервер
- Координация
- Хранилище
  - Row cache

# Уровень хранения данных



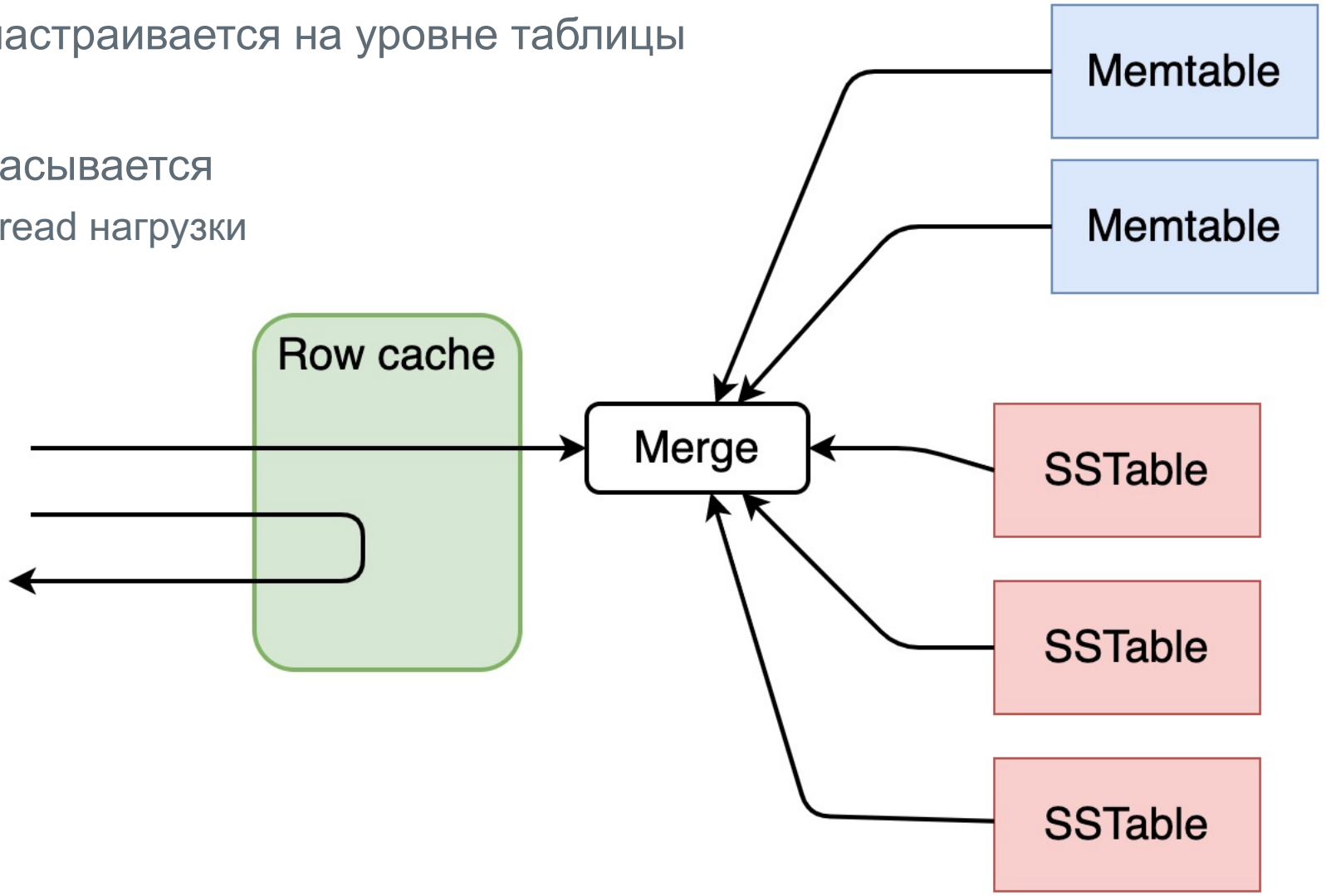
# Уровень хранения данных – row cache

- Выключен по умолчанию, настраивается на уровне таблицы



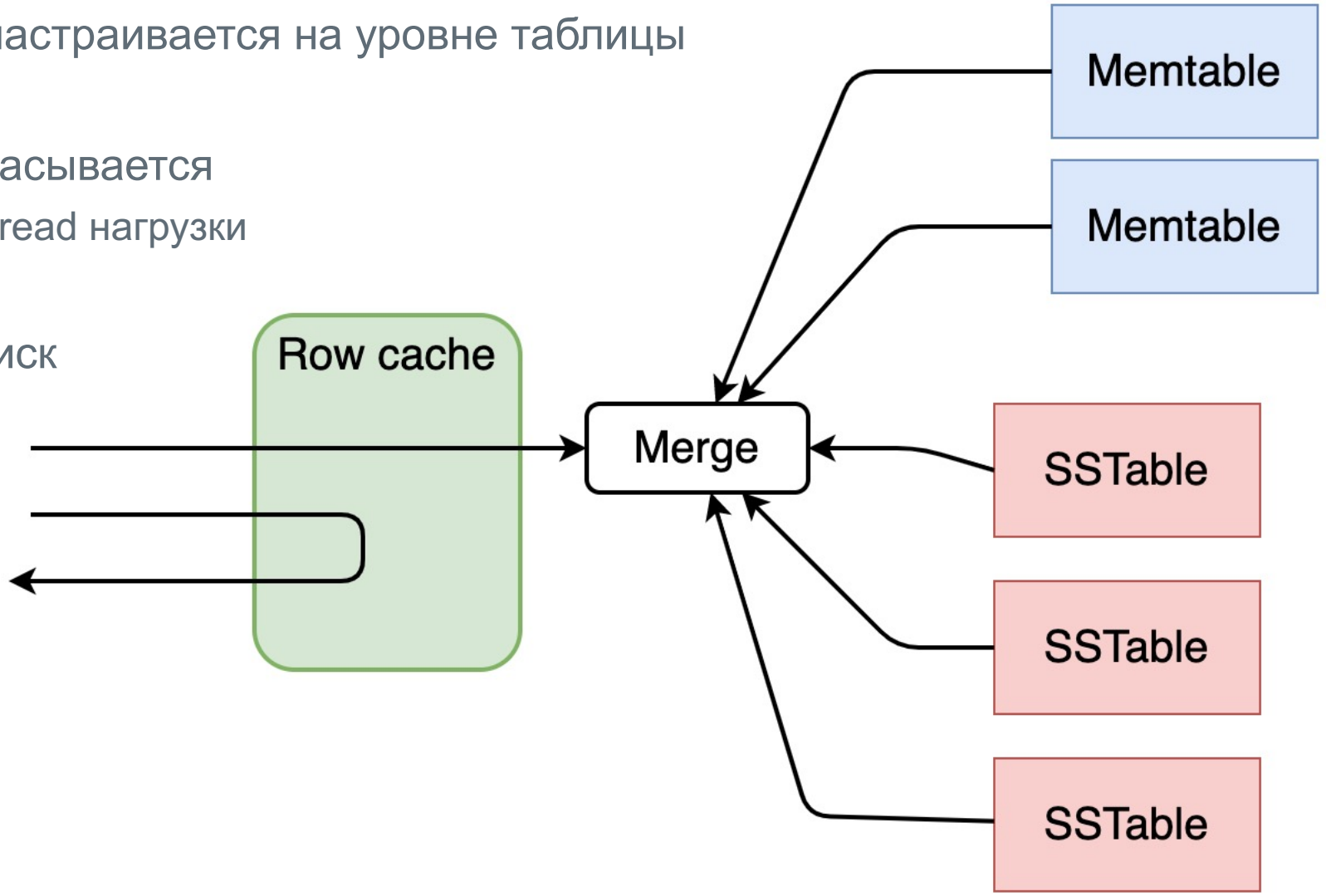
# Уровень хранения данных – row cache

- Выключен по умолчанию, настраивается на уровне таблицы
- Read-through
- При записи – partition выбрасывается
  - Полезен только для mostly-read нагрузки



# Уровень хранения данных – row cache

- Выключен по умолчанию, настраивается на уровне таблицы
- Read-through
- При записи – partition выбрасывается
  - Полезен только для mostly-read нагрузки
- Off-heap
- Может быть сохранен на диск (row_cache_save_period)



# Структура row cache

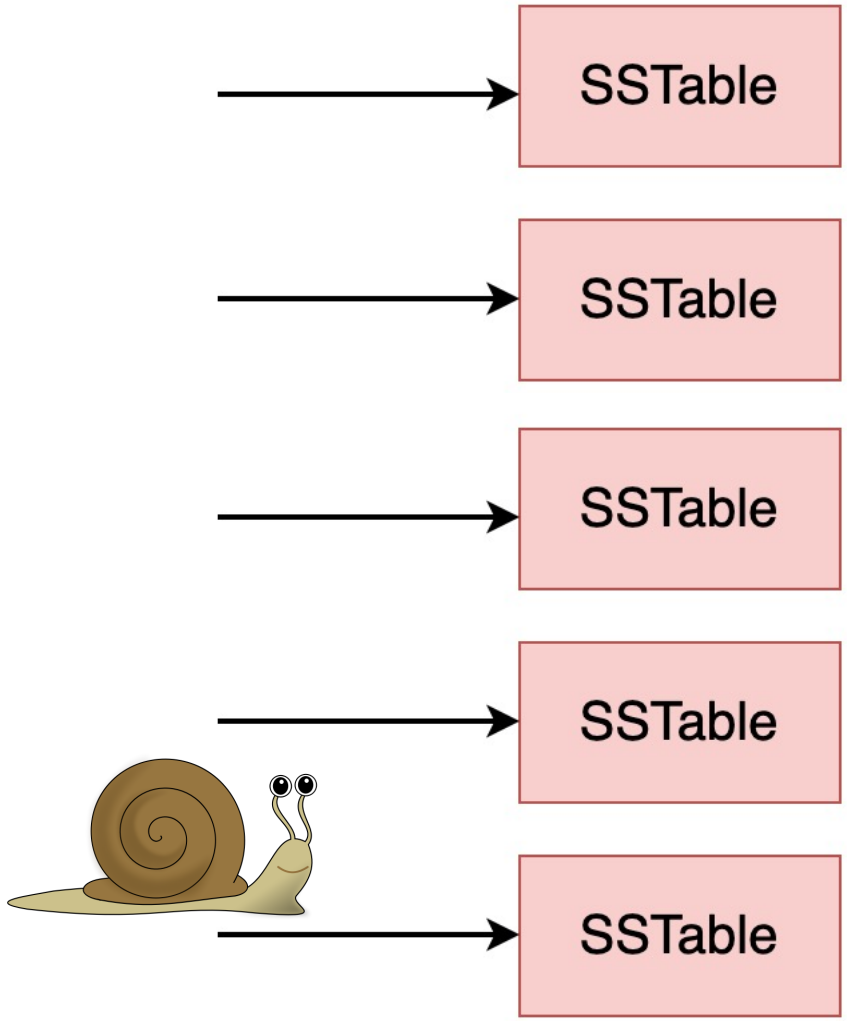
| row cache key | | row cache value | |
|---|---|---|---|
| <keyspace>.<table> | Partition key | Clustering key | Value |
| | user_id | event_time | message |
| space.messages | Алиса | 17:00:01 | куда мне отсюда идти? |
| | Чеширский кот | 17:00:05 | А куда ты хочешь попасть? |
| | Алиса | 17:00:09 | Мне все равно… |
| | Чеширский кот | 17:00:12 | Тогда все равно, куда и идти |

2 rows

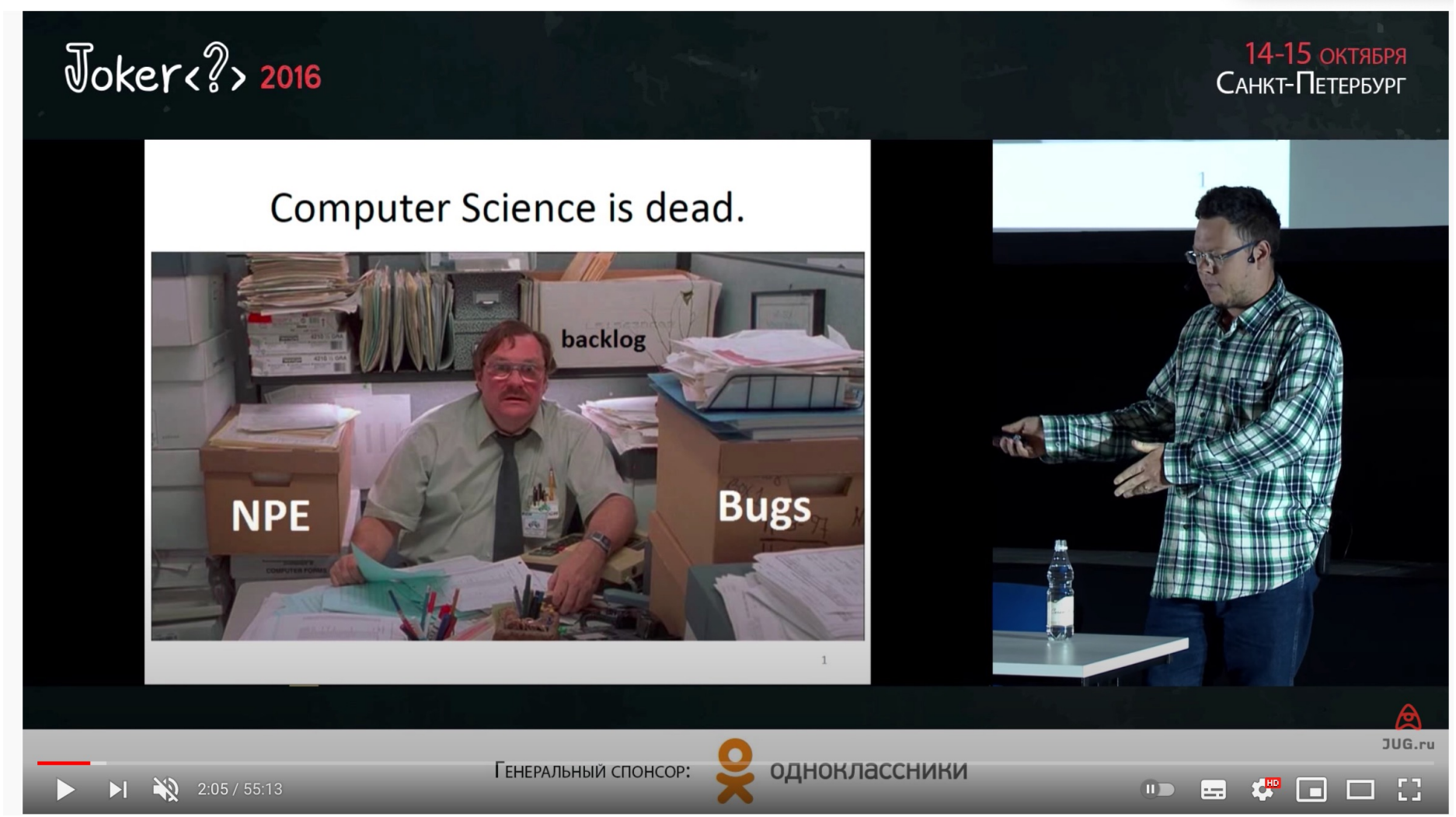
```
WITH caching = { 'rows_per_partition' : '2' };
```

- Запись, уровень кластера
- Запись, уровень реплики
- Базовый тест
- Клиент-сервер
- Координация
- Хранилище
  - Row cache
  - Какие SSTable читать?

# Читать все эти SSTables !?

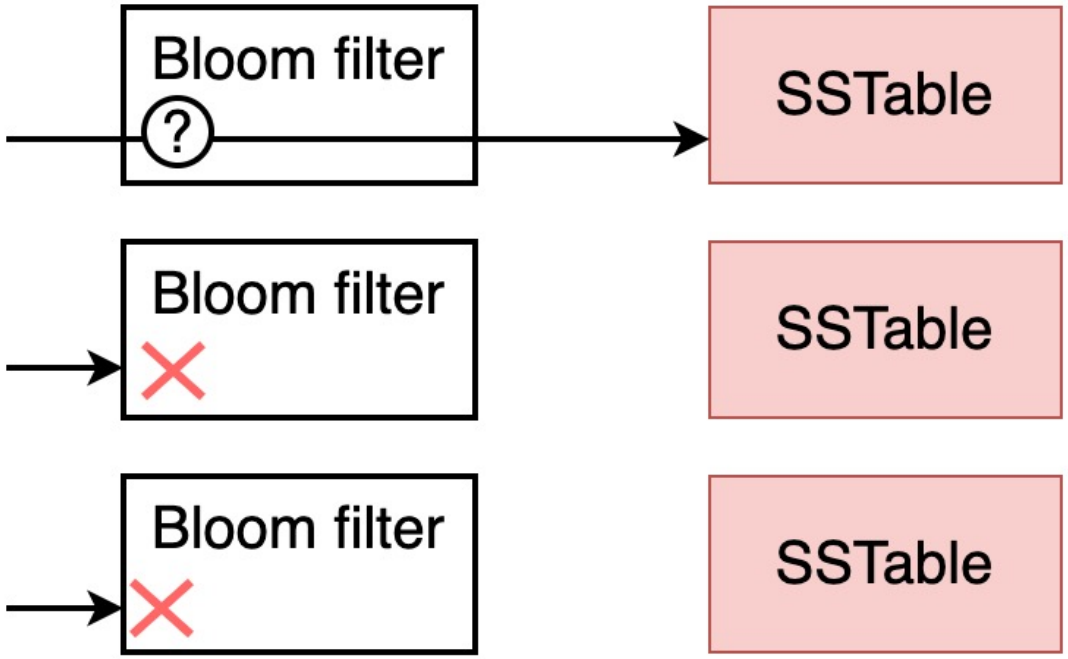


# Алексей Кудрявцев — Computer Science еще жива



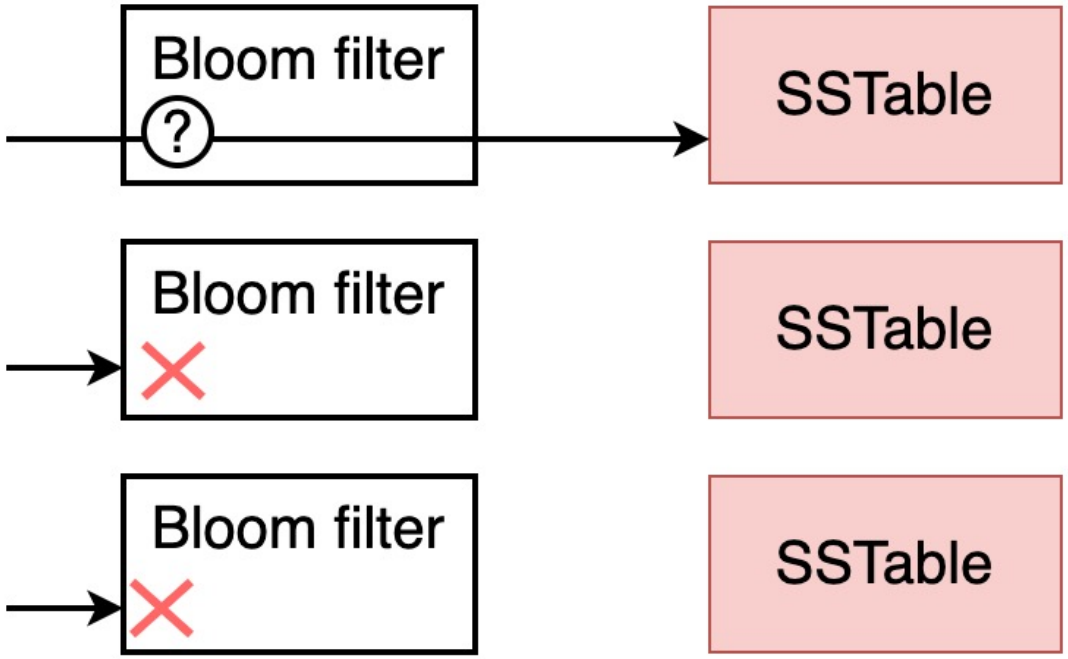
# Фильтр Блума

- Вероятностная структура данных
- Возвращаемое значение:
  - Может есть…
  - Точно нет!



# Фильтр Блума

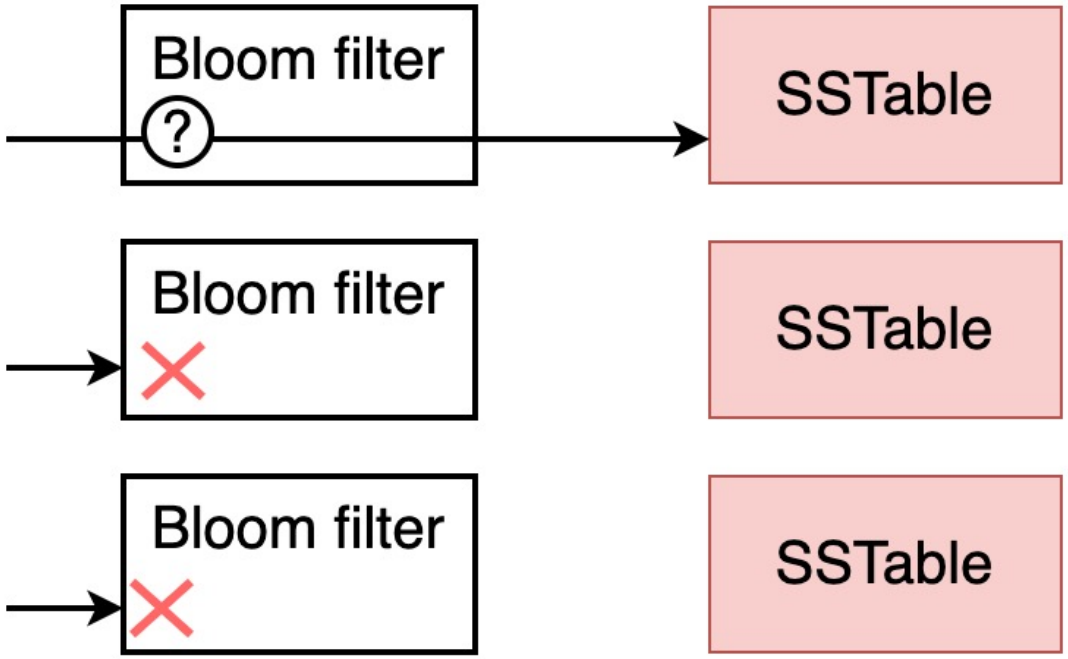
- Вероятностная структура данных
- Возвращаемое значение:
  - Может есть…
  - Точно нет!
- Позволяет быстро отсечь SSTables, в которых нет требуемых данных
- Отслеживает partition ключи



# Фильтр Блума

- Вероятностная структура данных
- Возвращаемое значение:
  - Может есть…
  - Точно нет!
- Позволяет быстро отсечь SSTables, в которых нет требуемых данных
- Отслеживает partition ключи
- Вероятность ошибки можно настраивать, чем больше памяти даем – тем меньше ошибка

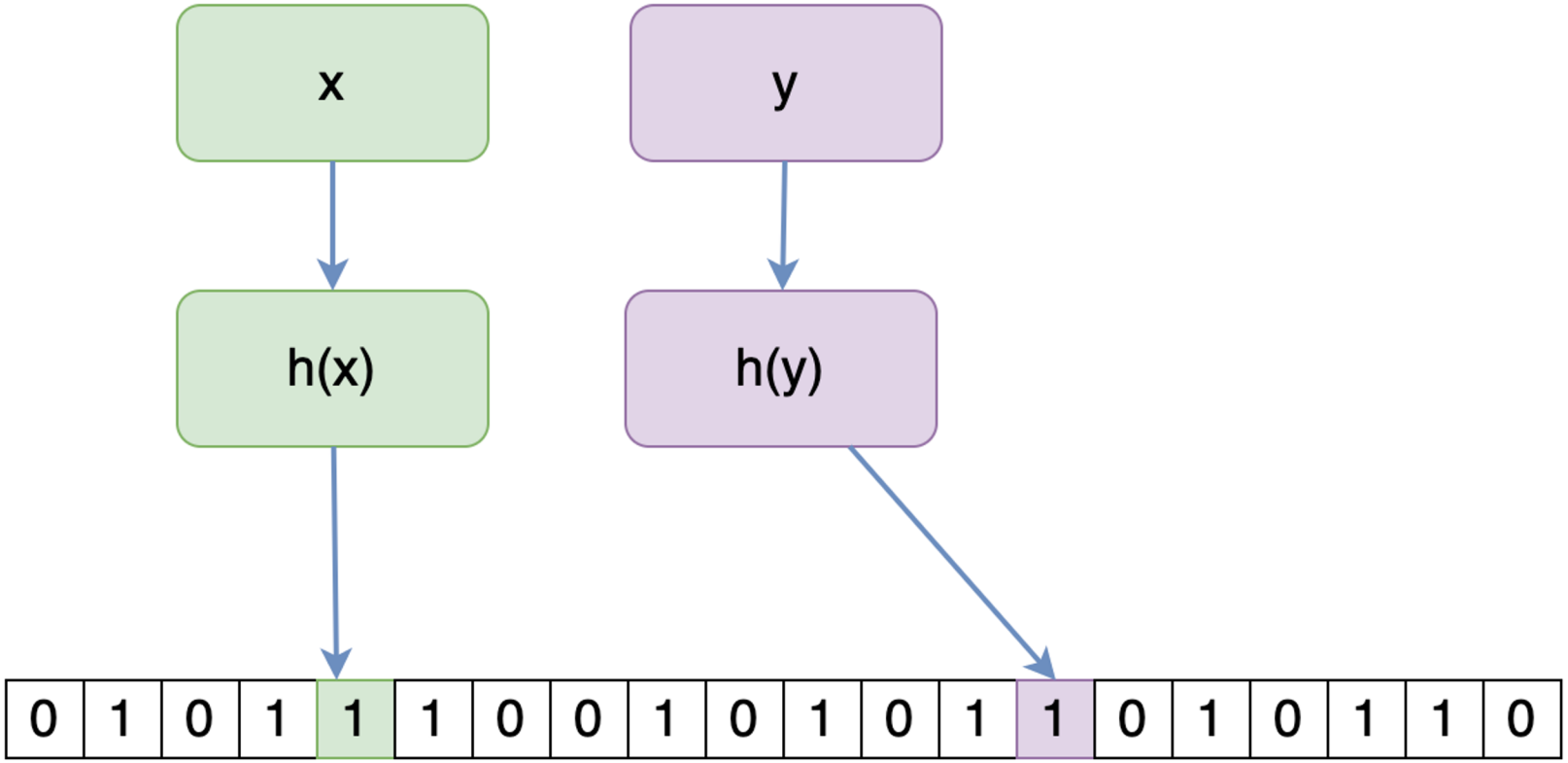
```
WITH bloom_filter_fp_chance=0.01
```



# Уровень хранения данных – фильтр Блума

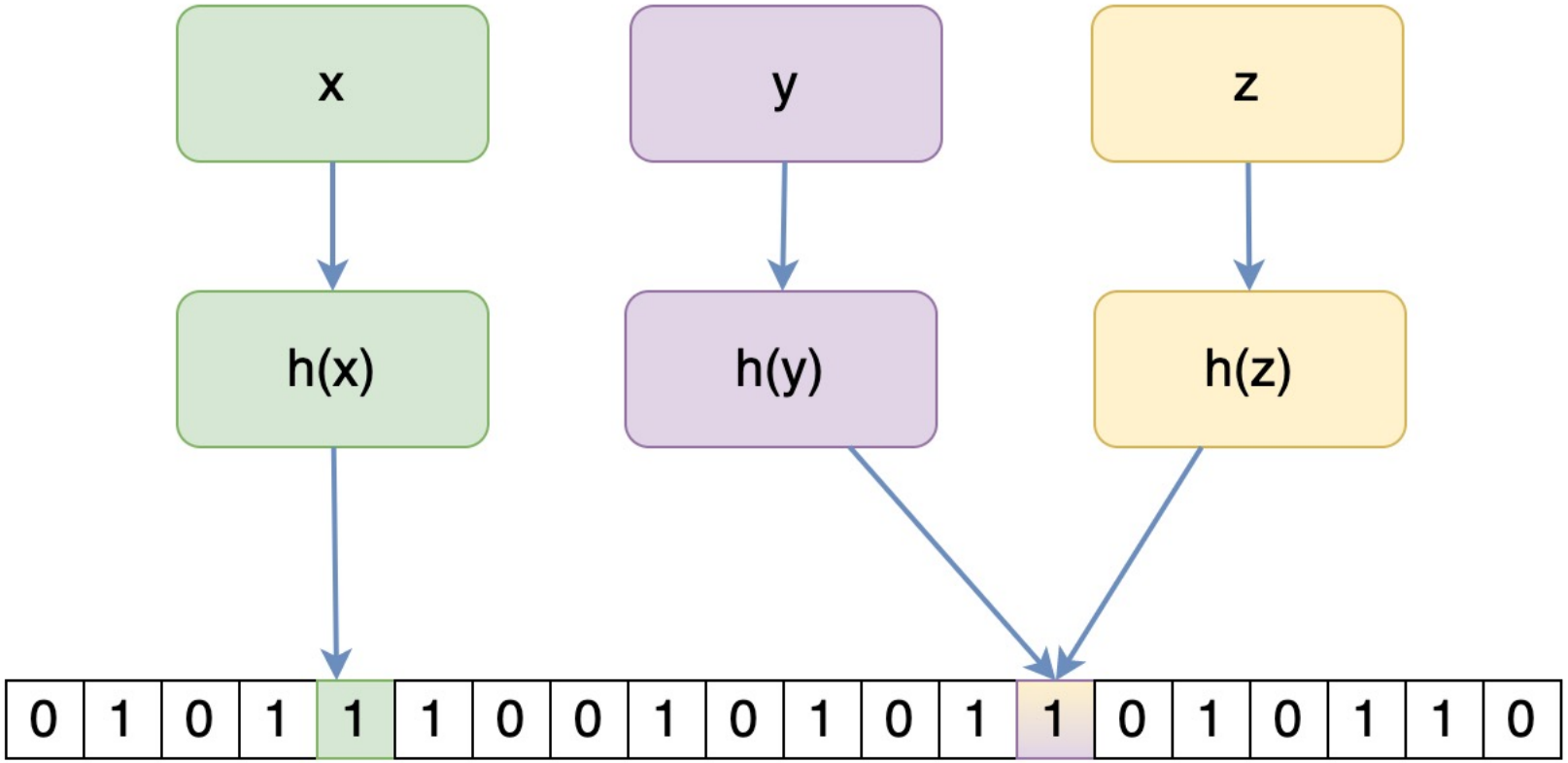

h(x):
1 => может есть
0 => точно нет

# Уровень хранения данных – фильтр Блума



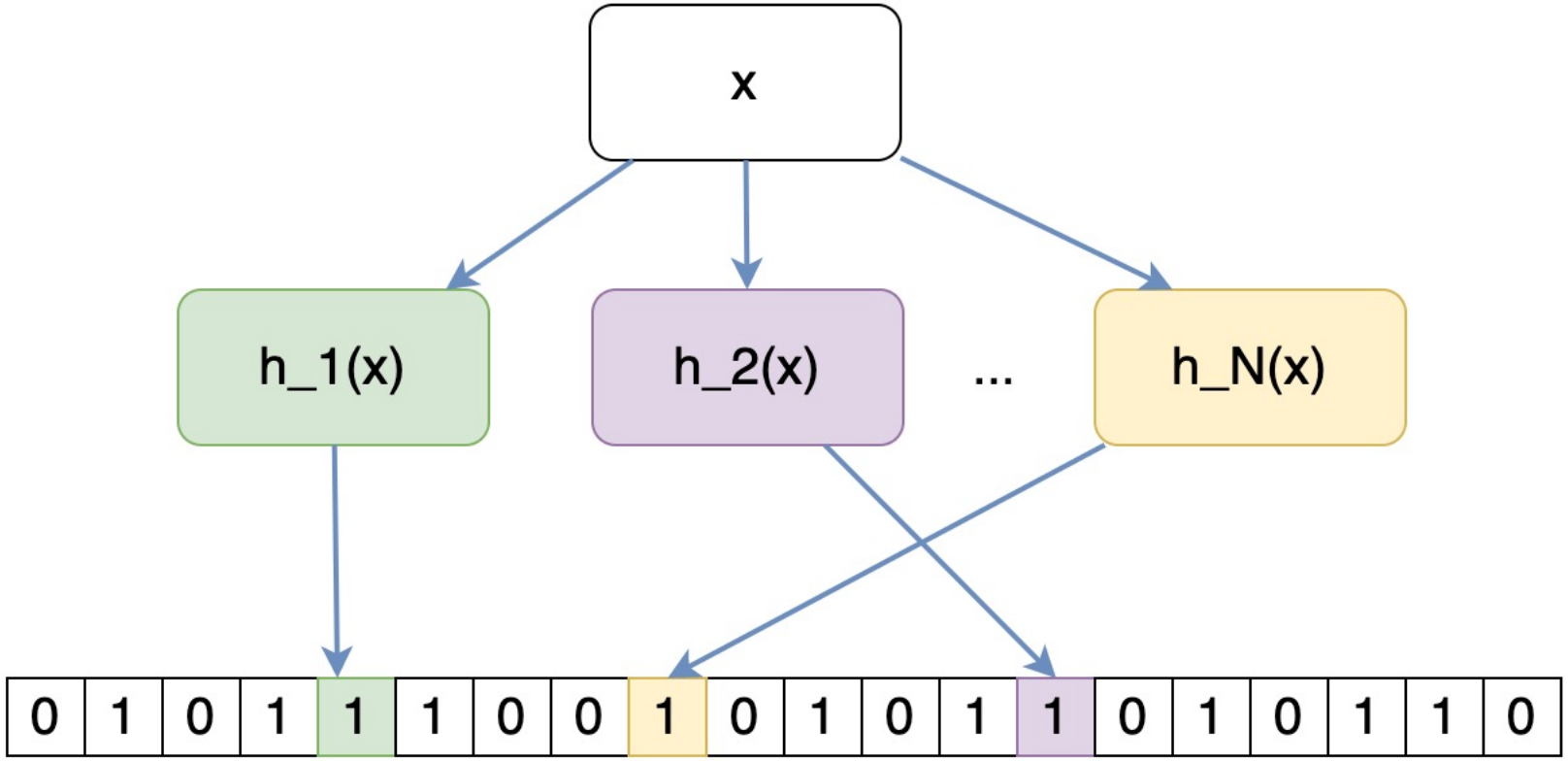

h(x):
1 => может есть
0 => точно нет

# Уровень хранения данных – фильтр Блума



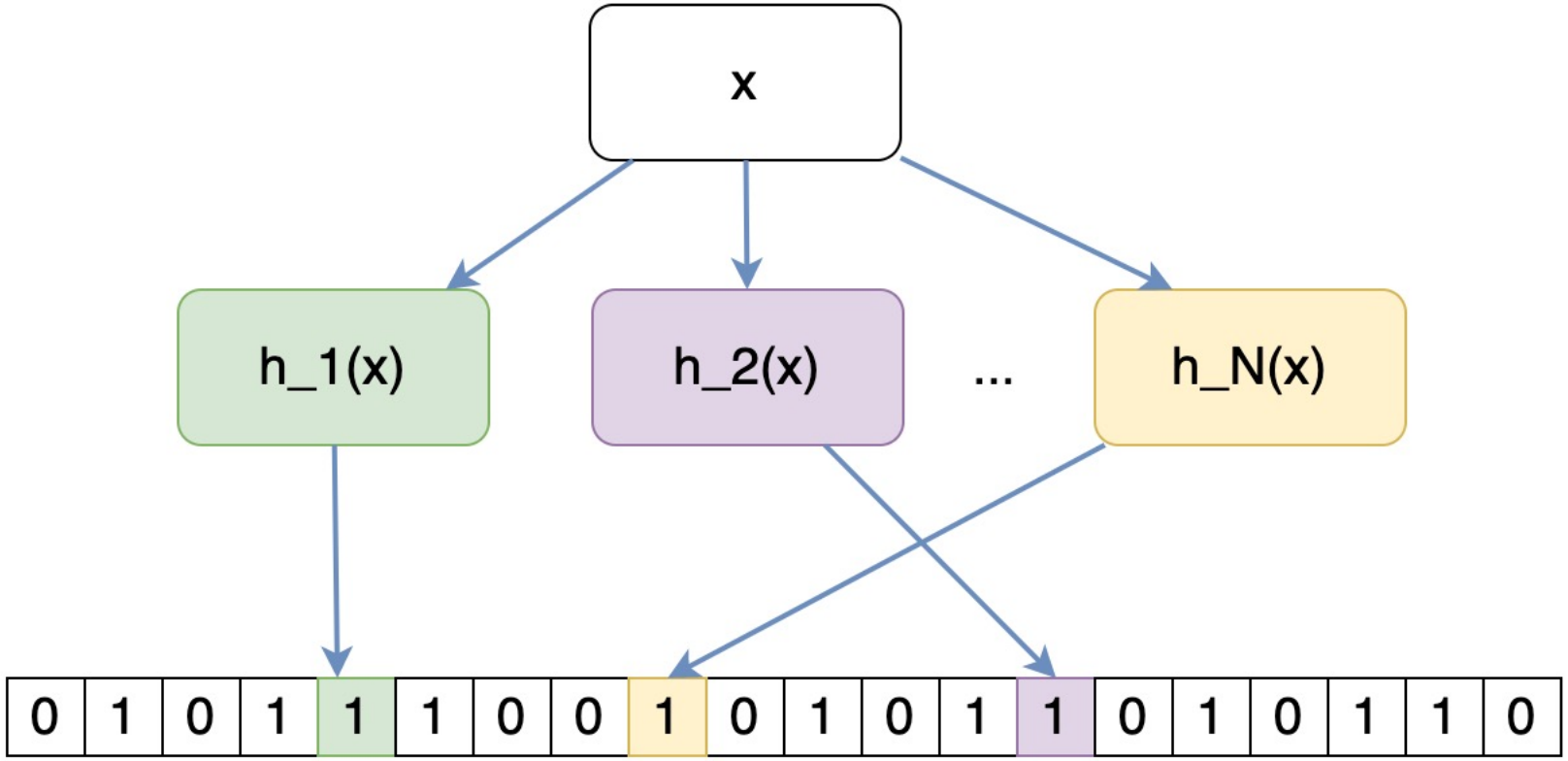

h(x):
1 => может есть
0 => точно нет

# Уровень хранения данных – фильтр Блума



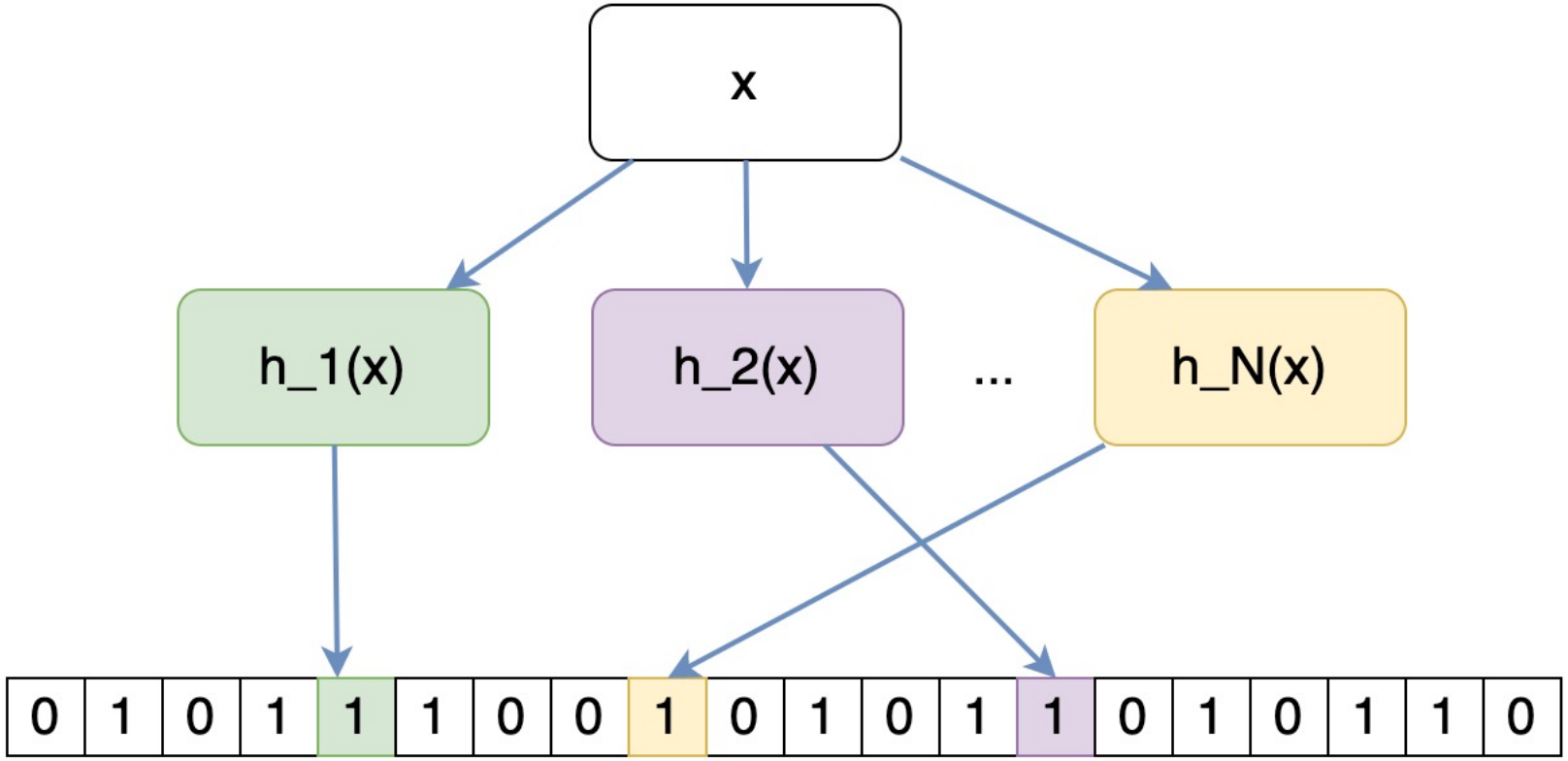

h_1(x) … h_N(x):
все 1 => может есть
есть 0 => точно нет

# Уровень хранения данных – фильтр Блума




MURMUR3(x) = MURMUR3_0(x) || MURMUR3_1(x)
h_i(x) = MURMUR3_1(x) + i * MURMUR3_0(x)

# Уровень хранения данных – фильтр Блума

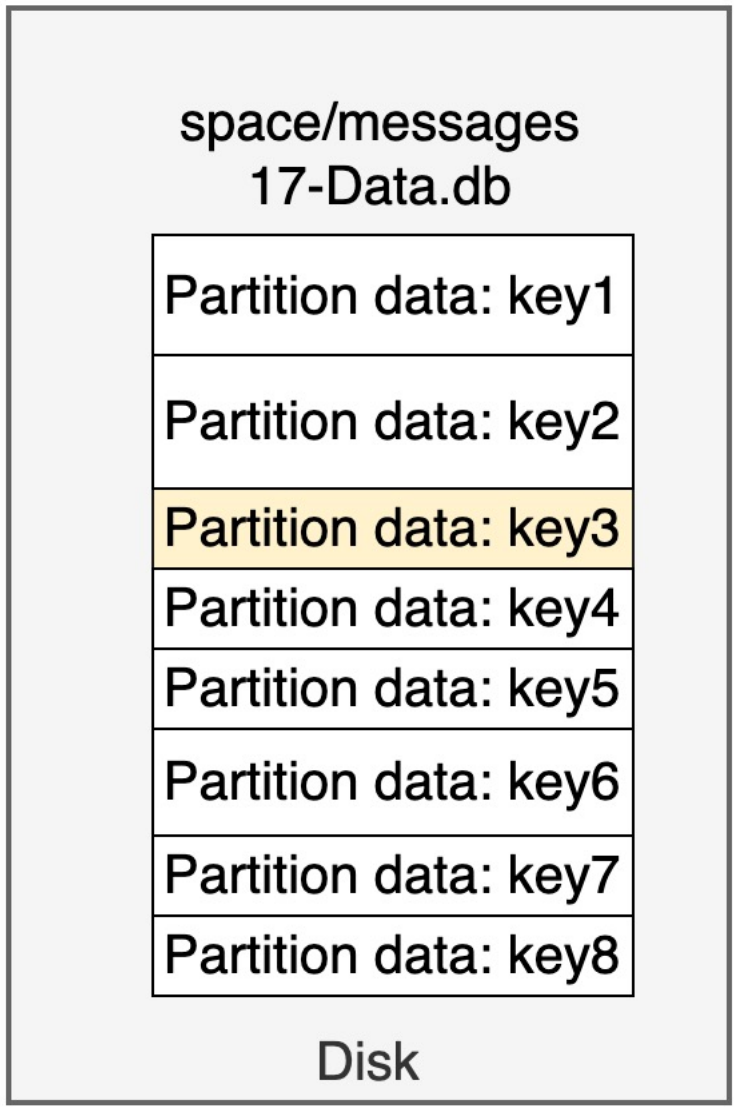



MURMUR3(x) = MURMUR3_0(x) || MURMUR3_1(x)
h_i(x) = MURMUR3_1(x) + i * MURMUR3_0(x)

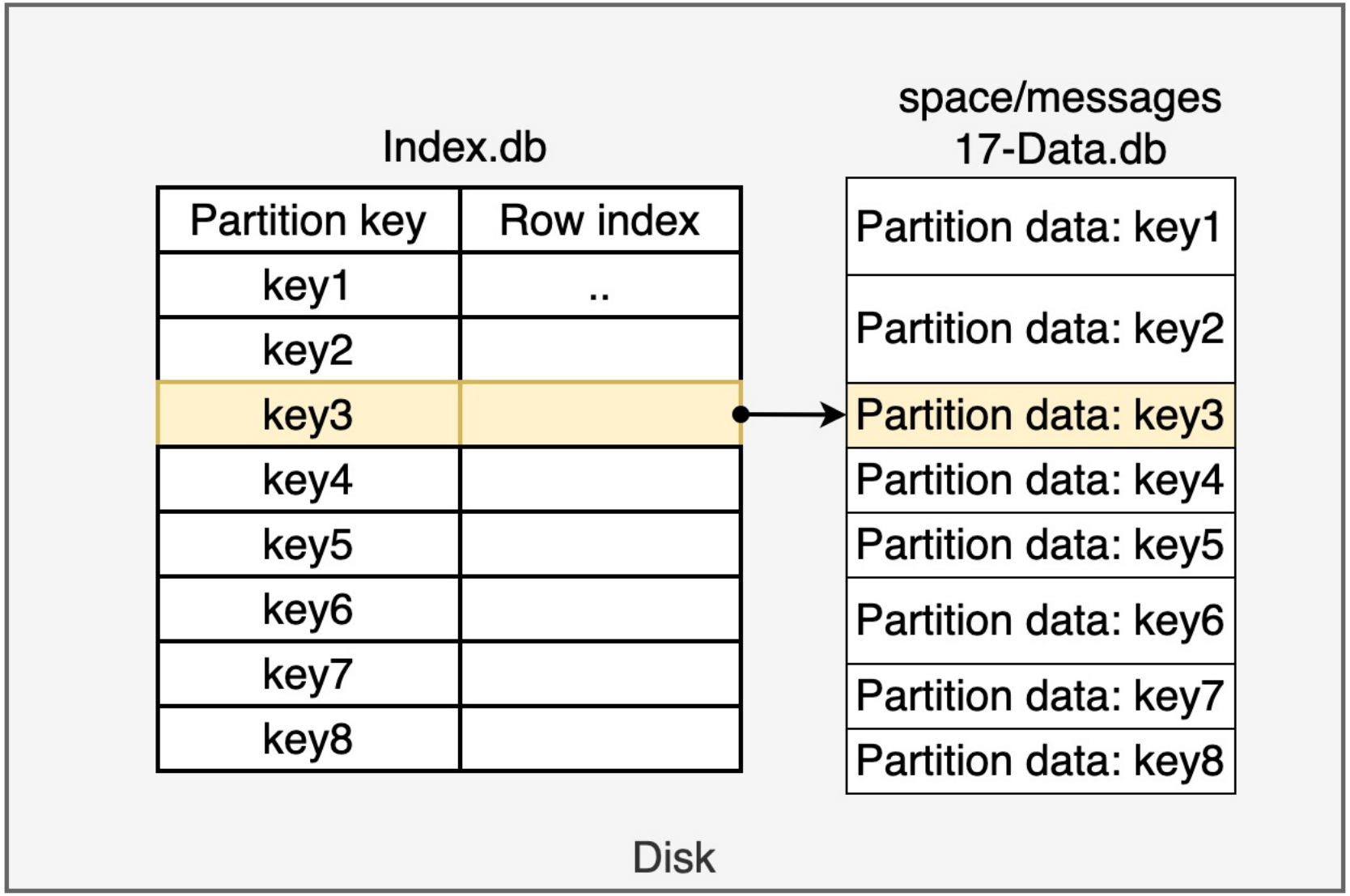
https://www.eecs.harvard.edu/~michaelm/postscripts/tr-02-05.pdf - почему так можно

- Запись, уровень кластера
- Запись, уровень реплики
- Базовый тест
- Клиент-серверное взаимодействие
- Координация
- Хранилище
  - Row cache
  - Bloom filter
  - Поиск в SSTable

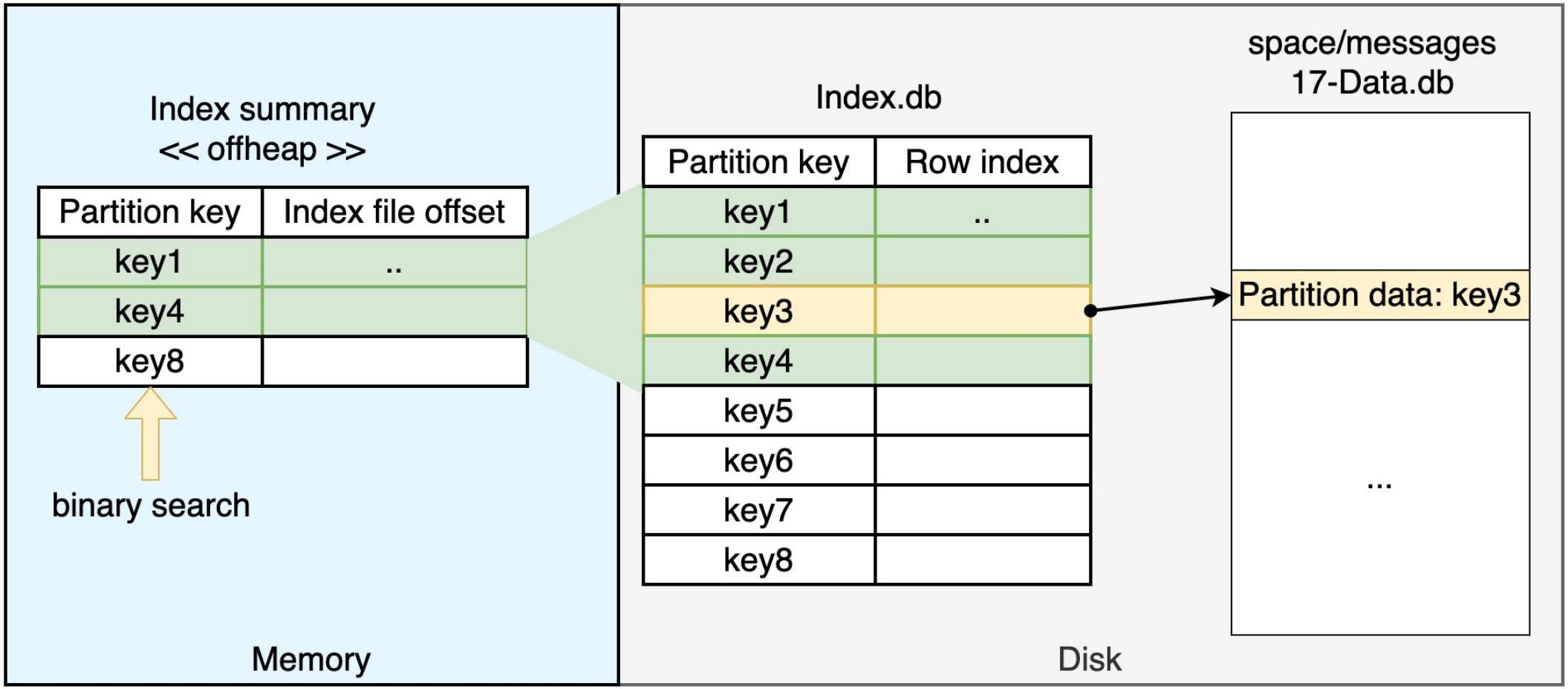
# Уровень хранения данных – поиск в SSTable



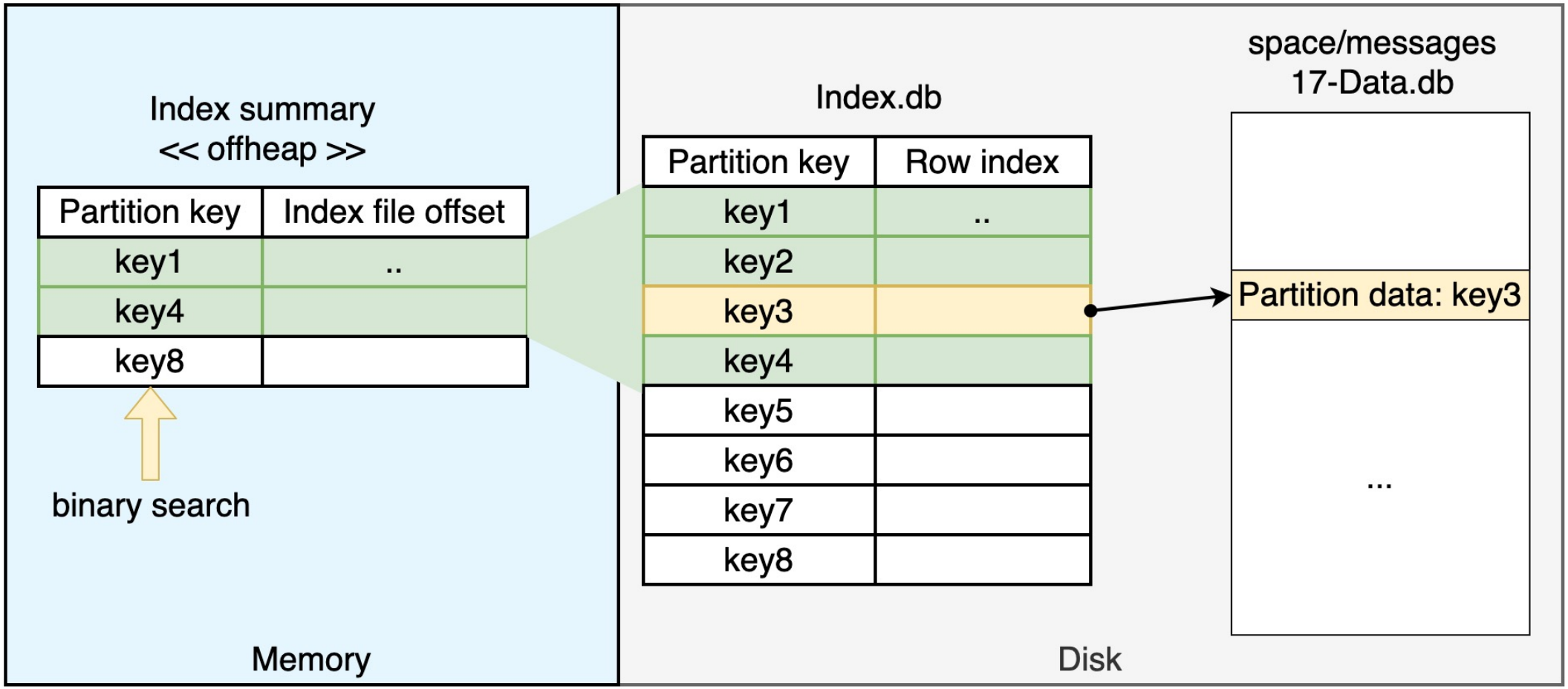
# Уровень хранения данных – поиск в SSTable



# Уровень хранения данных – поиск в SSTable

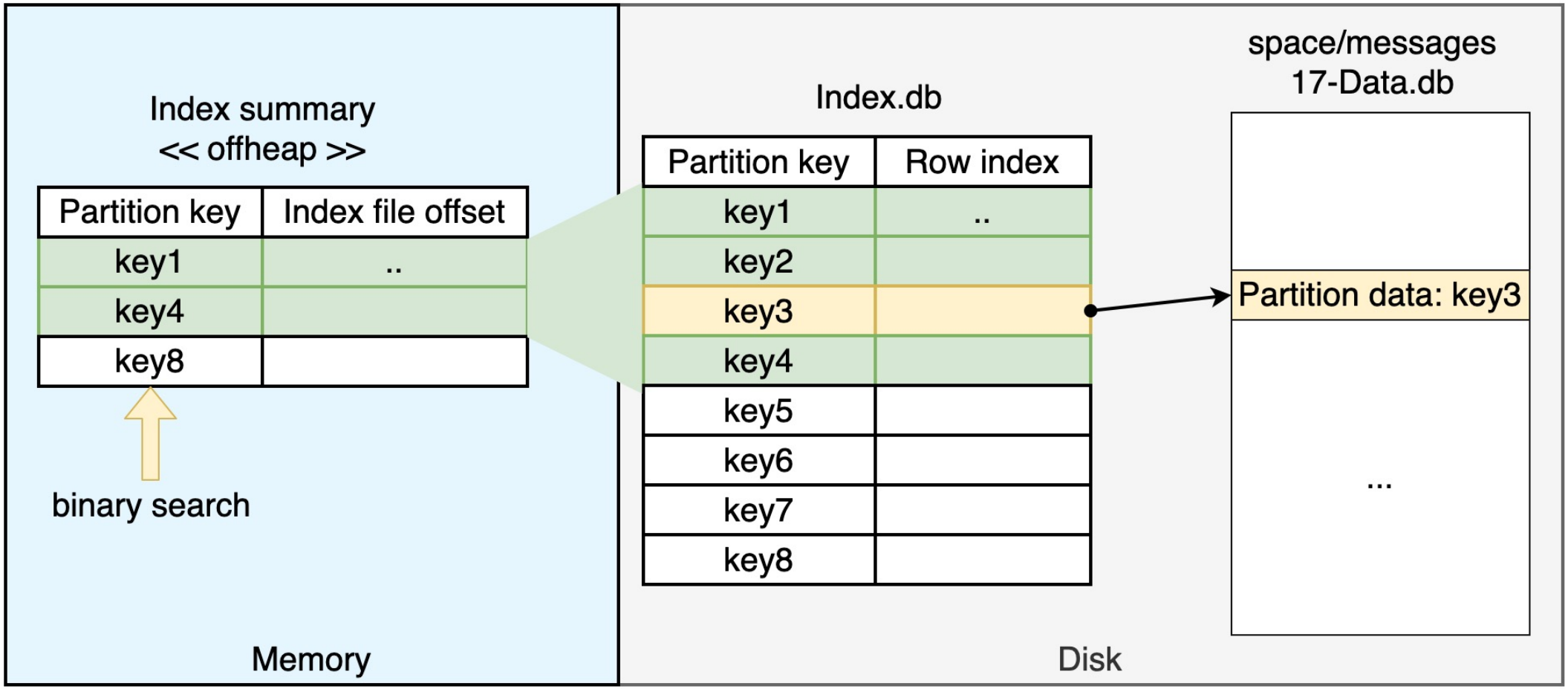


# Уровень хранения данных – поиск в SSTable



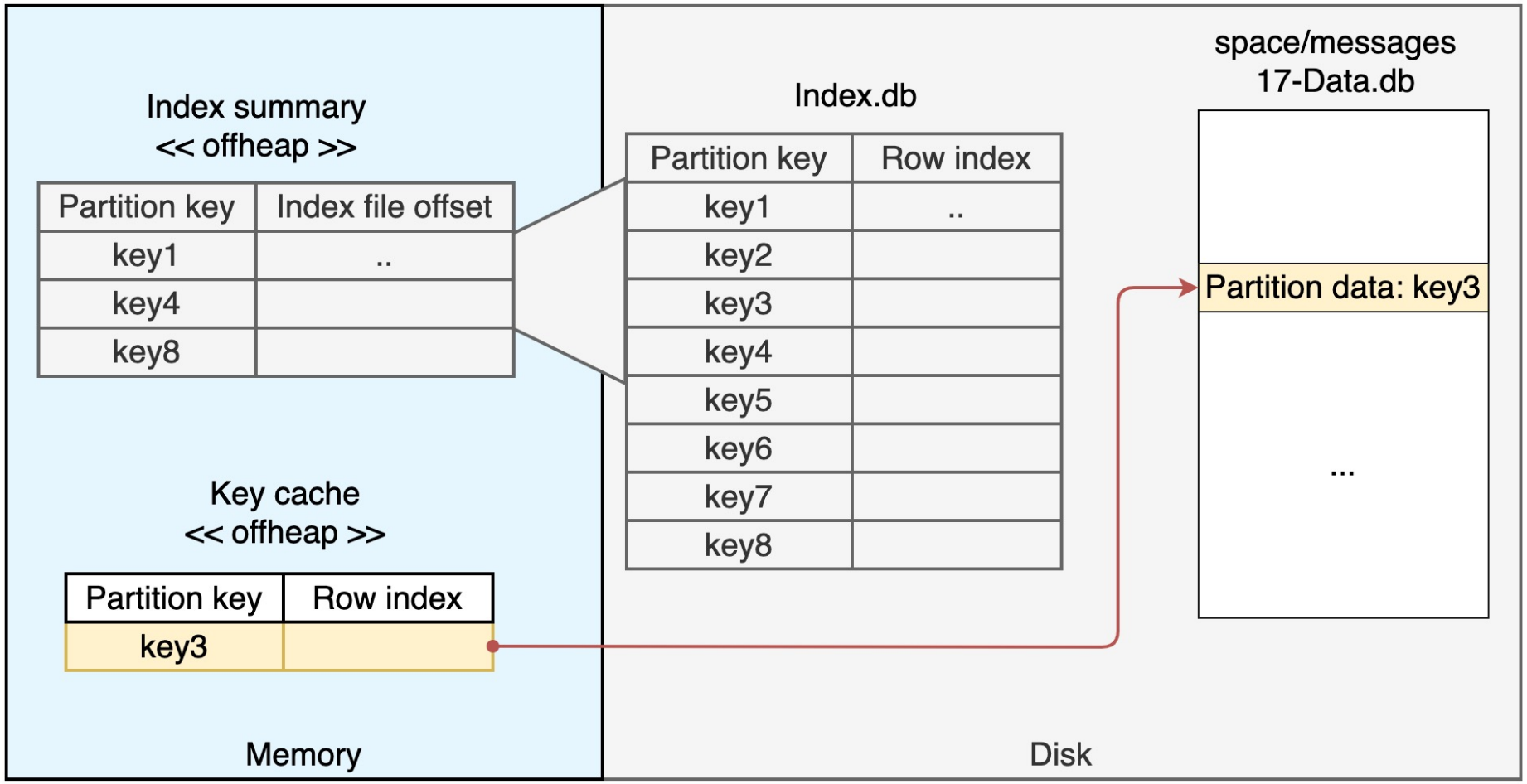

- Настройки уровня таблицы: min_index_interval = 128 / max_index_interval = 2048

# Уровень хранения данных – поиск в SSTable

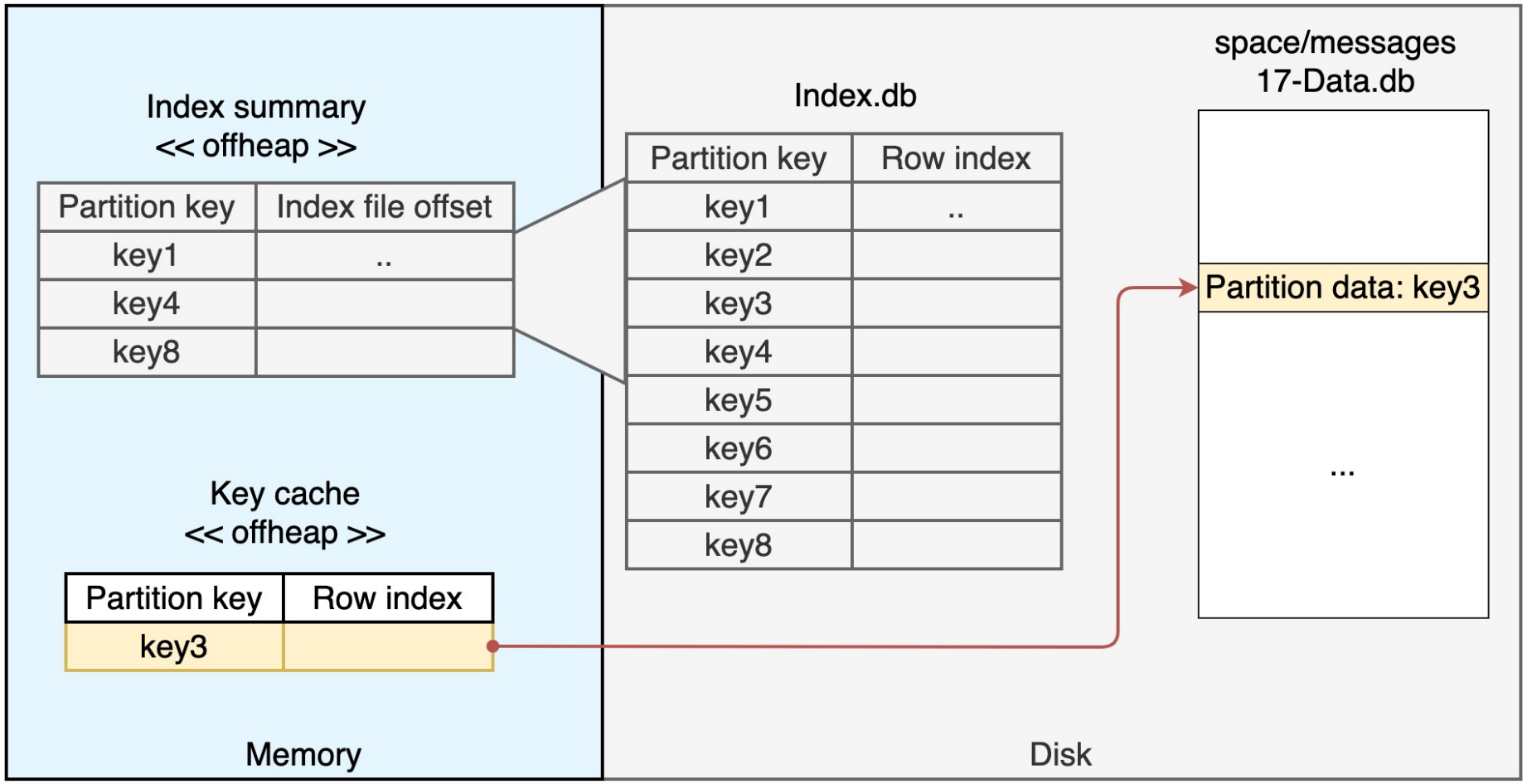



- Настройки уровня таблицы: min_index_interval = 128 / max_index_interval = 2048
- Cassandra.yaml: index_summary_capacity_in_mb (default: 5% of heap size)

# Уровень хранения данных – поиск в SSTable

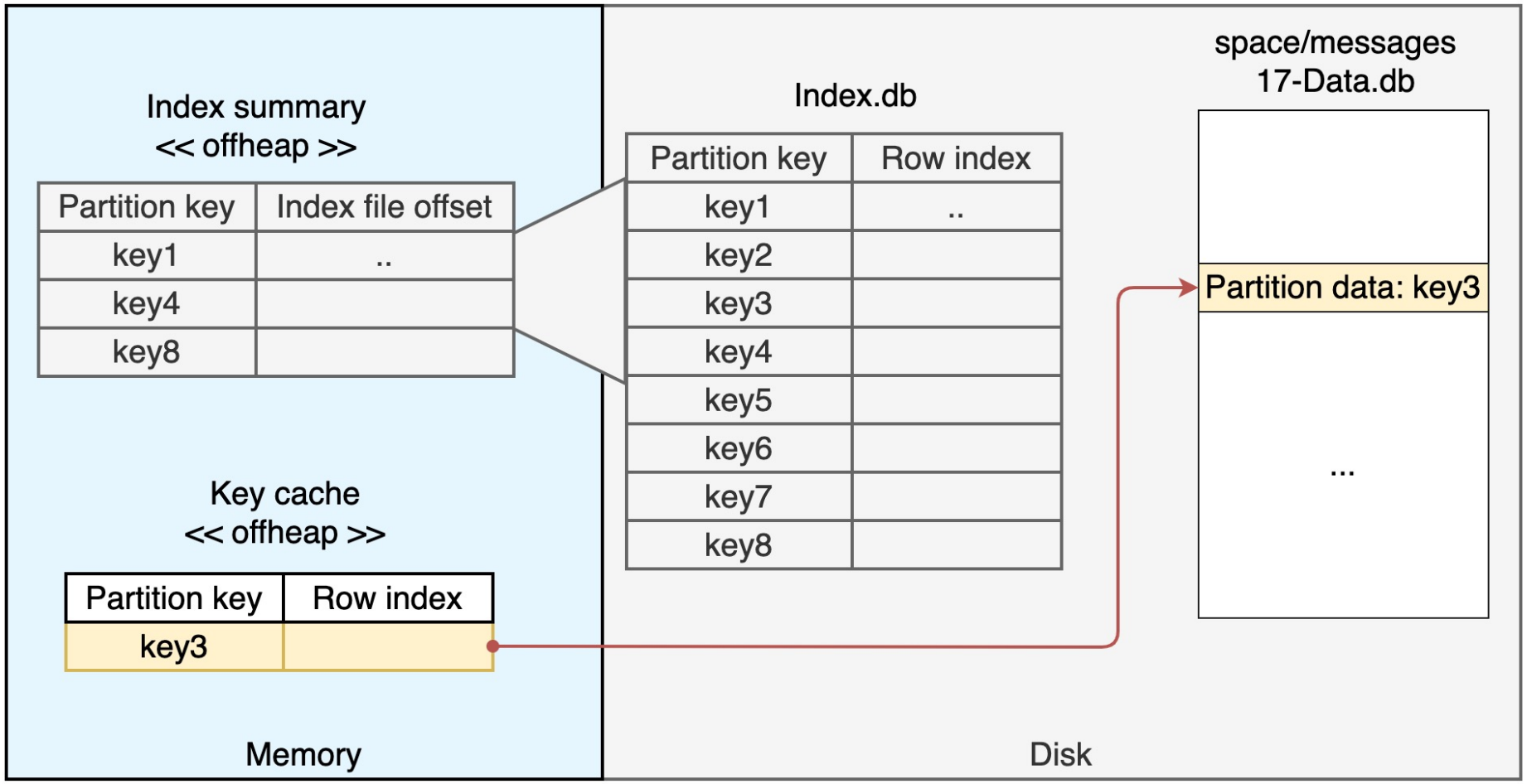


# Уровень хранения данных – поиск в SSTable




- Настройки уровня таблицы: caching = {'keys': 'ALL | NONE'}

# Уровень хранения данных – поиск в SSTable

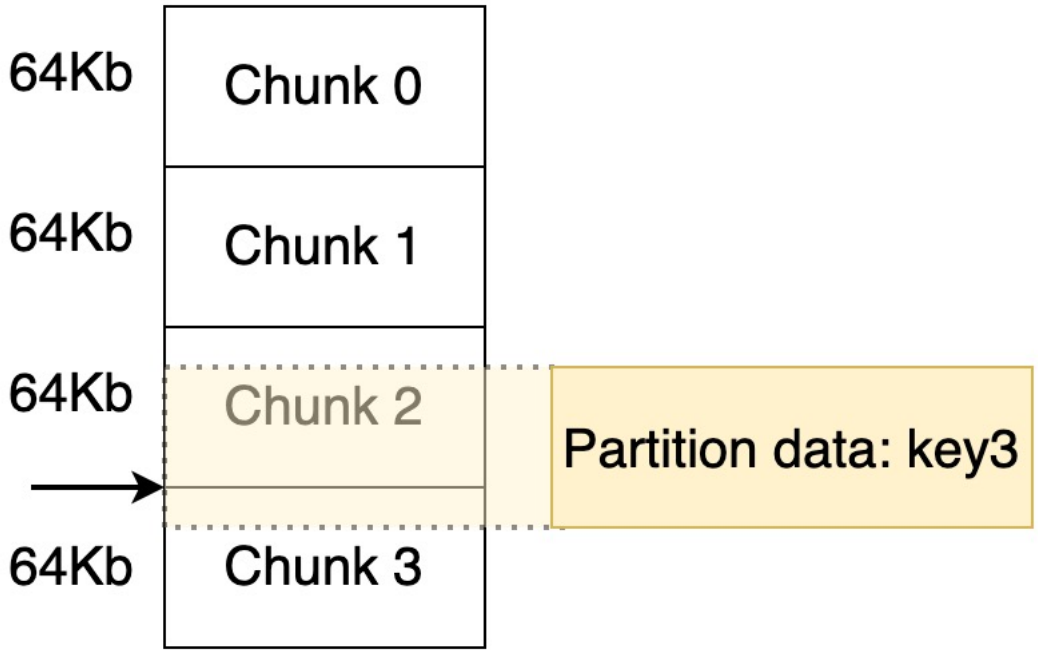



- Настройки уровня таблицы: caching = {'keys': ‘ALL | NONE’}
- Cassandra.yaml: key_cache_size_in_mb ( default: min(5% of Heap (in MB), 100MB) )

- Запись, уровень кластера
- Запись, уровень реплики
- Базовый тест
- Клиент-серверное взаимодействие
- Координация
- Хранилище
  - Row cache
  - Bloom filter
  - Поиск по SSTable
  - Сжатие

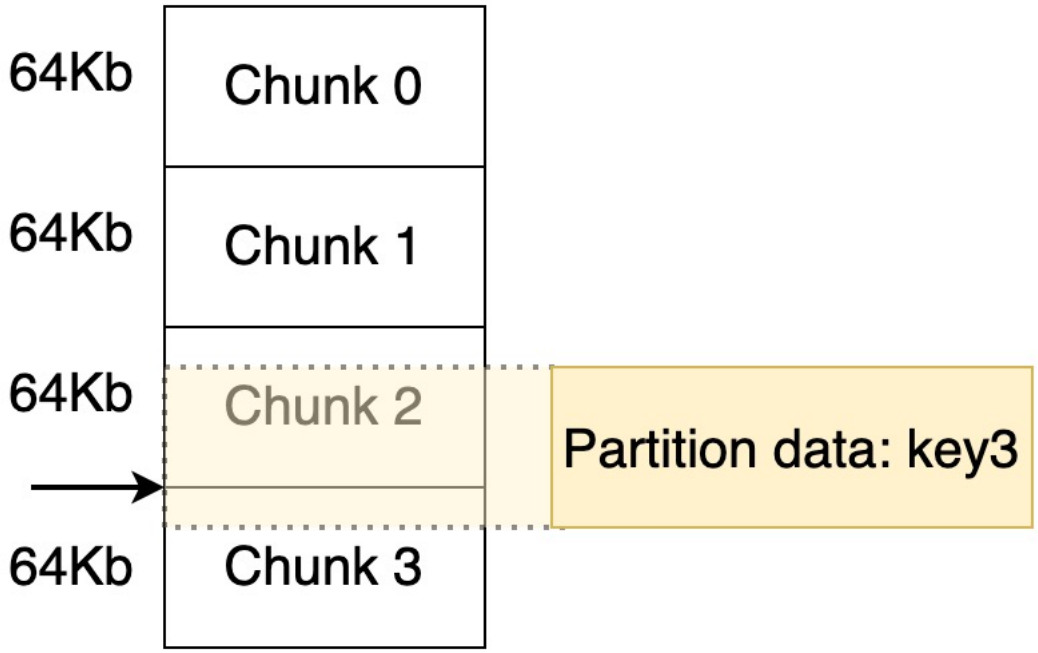
# Уровень хранения данных – сжатие

- SSTables могут быть сжаты
- По умолчанию режим сжатия ВКЛючен

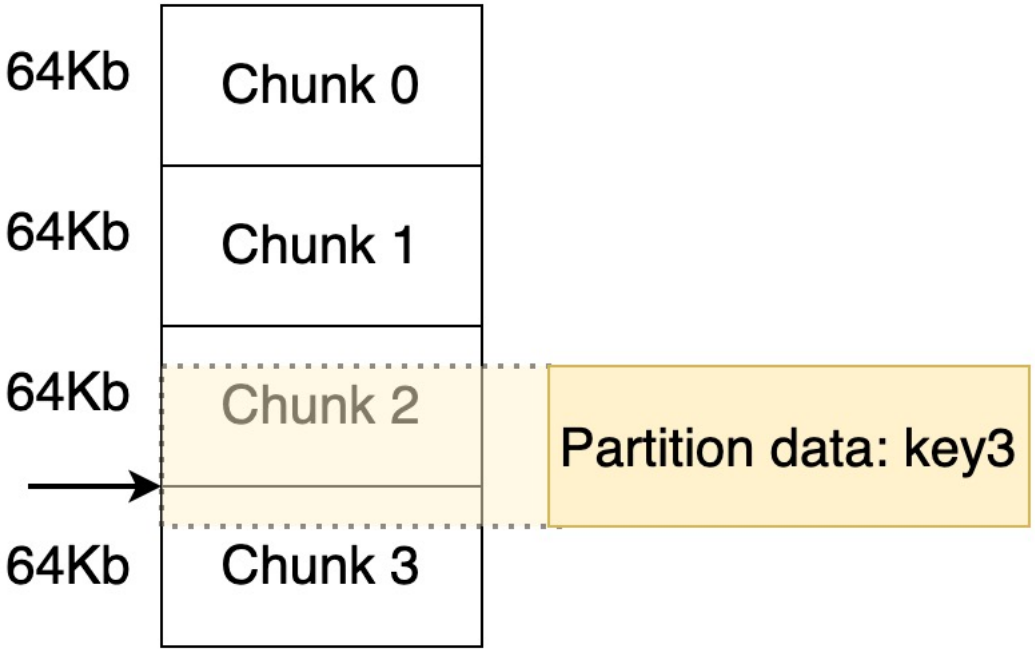


# Уровень хранения данных – сжатие

- SSTables могут быть сжаты
- По умолчанию режим сжатия включен
- Алгоритмы:
  - LZ4 (default)
  - Deflate → Zstd
  - Snappy → LZ4
  - Zstd (Cassandra 4.0)

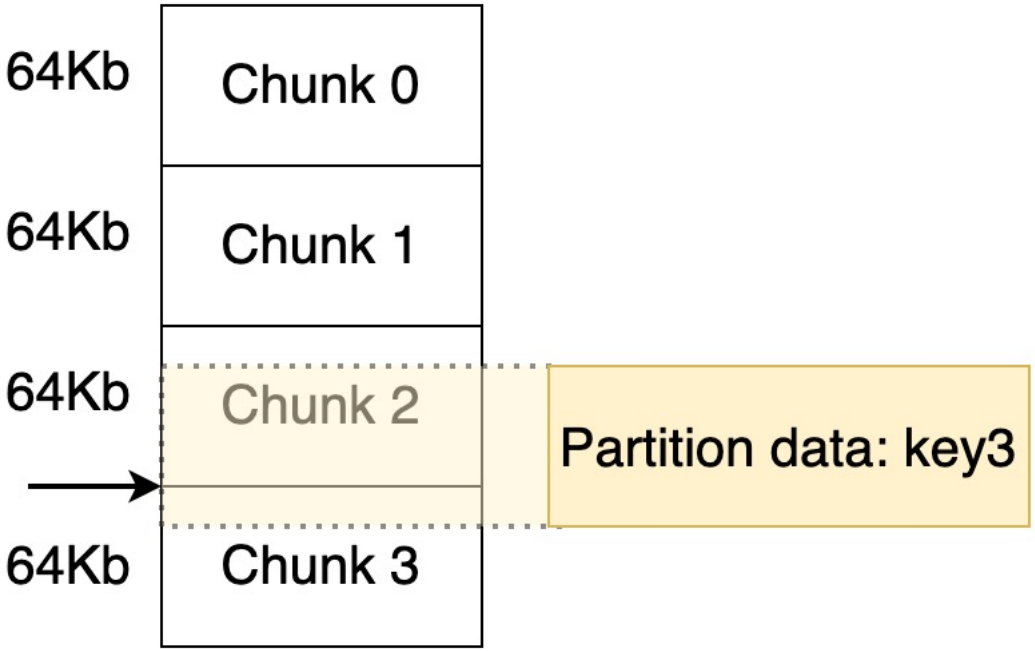


# Уровень хранения данных – сжатие



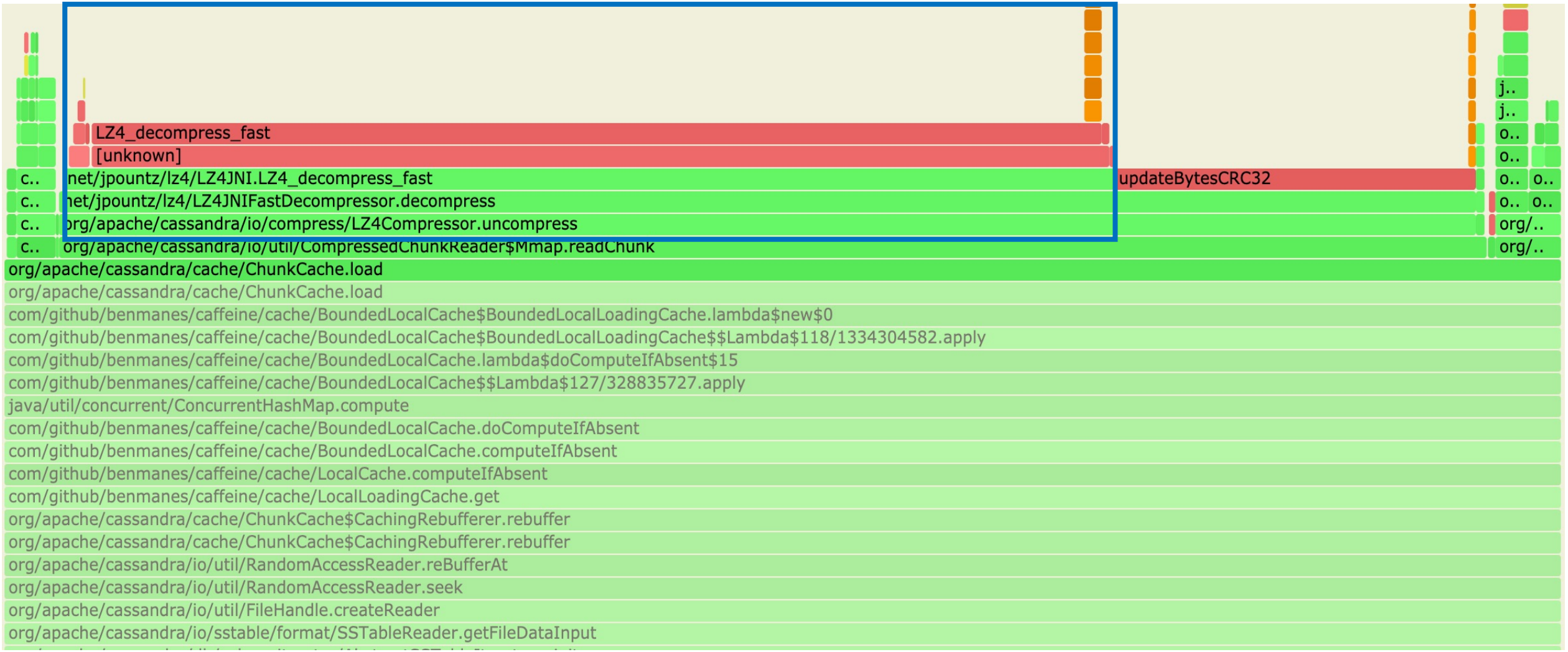

- Настройка на уровне таблицы compression = {'chunk_length_in_kb': '64', 'class': '...LZ4Compressor'}
- Данные сжимаются блоками фиксированной длины (chunks)

# Уровень хранения данных – сжатие

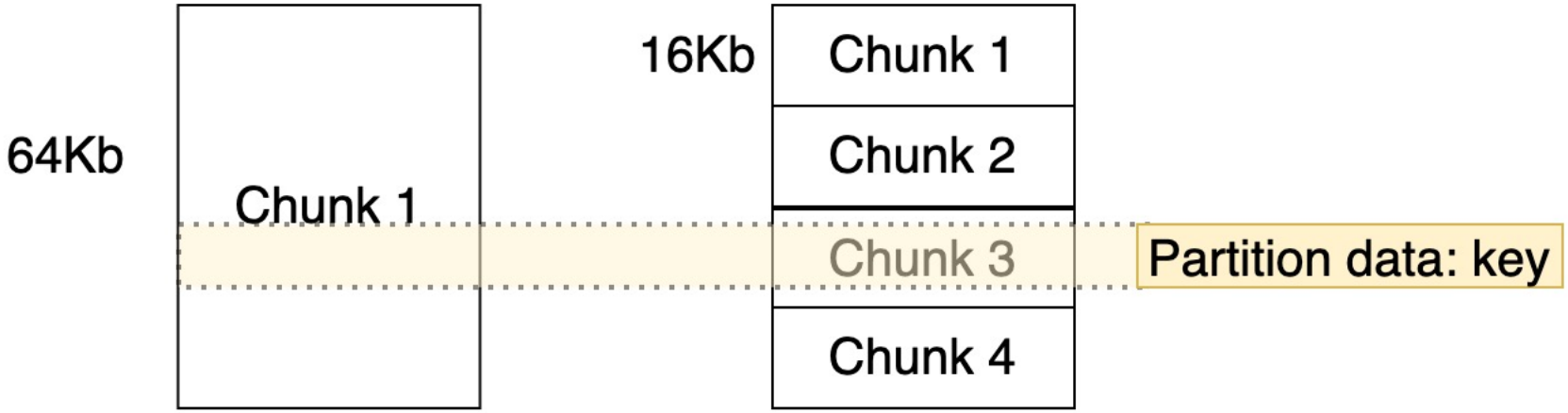



- Настройка на уровне таблицы compression = {'chunk_length_in_kb': '64', 'class': '...LZ4Compressor'}
- Данные сжимаются блоками фиксированной длины (chunks)
- chunk_length_kb - размер блока: compression efficiency vs read overhead
- Default: 64Kb

# Уровень хранения данных – сжатие



# Уровень хранения данных – сжатие




- Default: 64Kb → 16Kb (Cassandra 4.0, CASSANDRA-13241)
- https://thelastpickle.com/blog/2018/08/08/compression_performance.html

# https://github.com/lz4/lz4/releases

## LZ4 v1.9.0

**Cyan4973** released this on 16 Apr 2019

**Warning** : this version has a known bug in the decompression function which makes it read a few bytes beyond input limit. Upgrade to v1.9.1 is recommended.

LZ4 v1.9.0 is a performance focused release, also offering minor API updates.

| Version | v1.8.3 | v1.9.0 | Improvement |
|---|---|---|---|
| enwik8 | 4090 MB/s | 4560 MB/s | +12% |
| calgary.tar | 4320 MB/s | 4860 MB/s | +13% |
| silesia.tar | 4210 MB/s | 4970 MB/s | +18% |

# https://github.com/lz4/lz4/releases

## LZ4 v1.9.0

**Cyan4973** released this on 16 Apr 2019

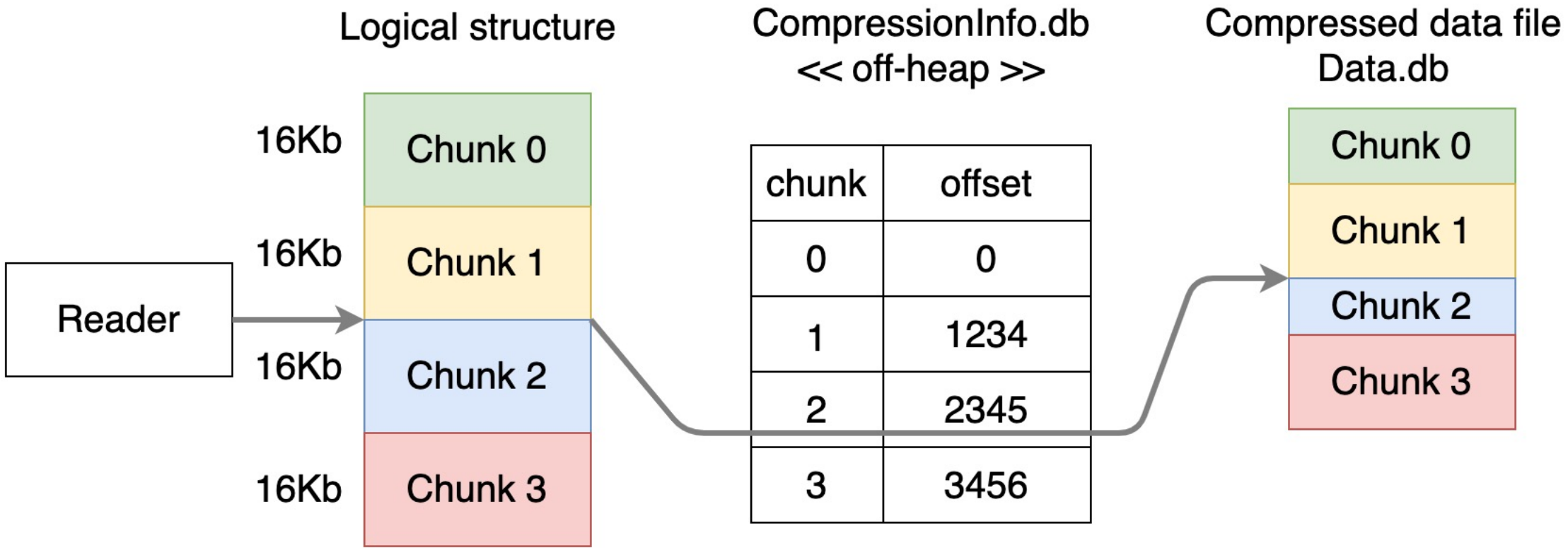
**Warning** : this version has a known bug in the decompression function which makes it read a few bytes beyond input limit. Upgrade to v1.9.1 is recommended.

LZ4 v1.9.0 is a performance focused release, also offering minor API updates.

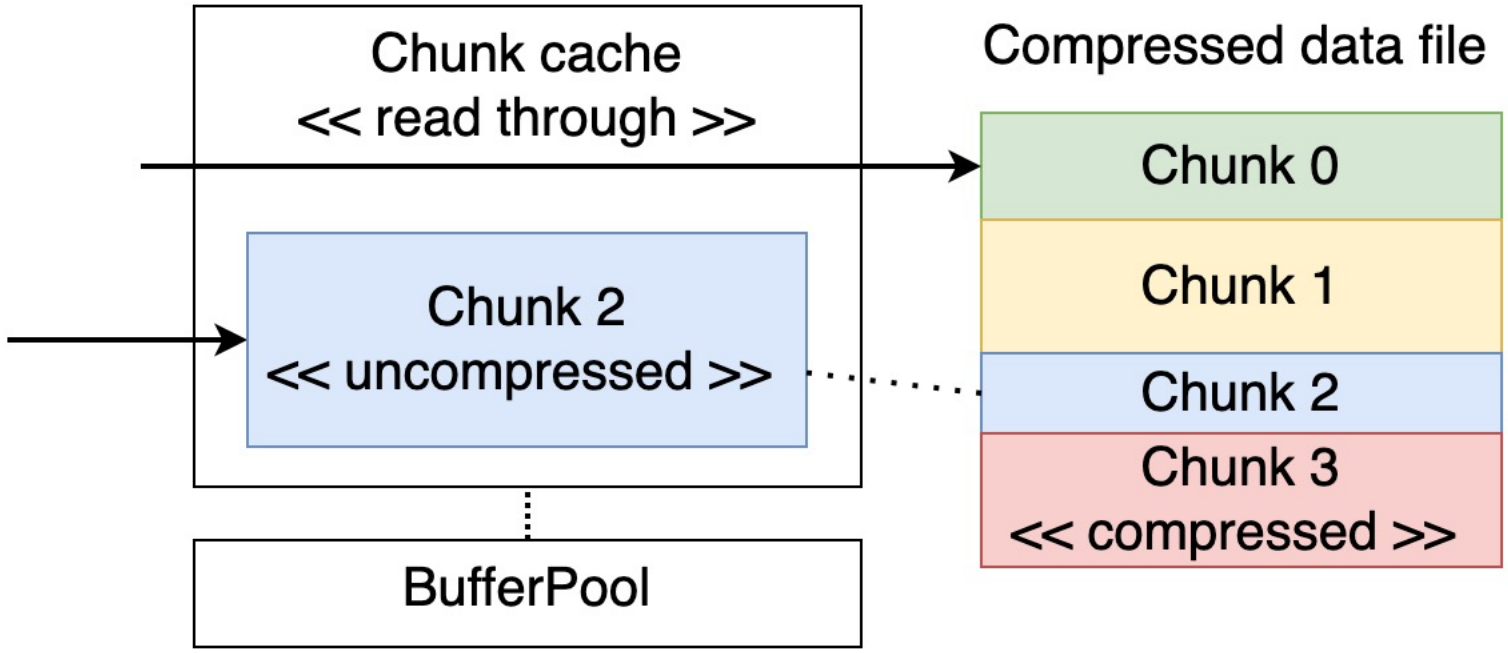
| Version | v1.8.3 | v1.9.0 | Improvement |
|---|---|---|---|
| enwik8 | 4090 MB/s | 4560 MB/s | +12% |
| calgary.tar | 4320 MB/s | 4860 MB/s | +13% |
| silesia.tar | 4210 MB/s | 4970 MB/s | +18% |

- lz4-java 1.3.0 → 1.7.1 (based on LZ4 1.9.2)
- CASSANDRA-15941 (open) Upgrade to lz4-java 1.7.2 when released (due to Centos 6 support)

# Уровень хранения данных – compression info

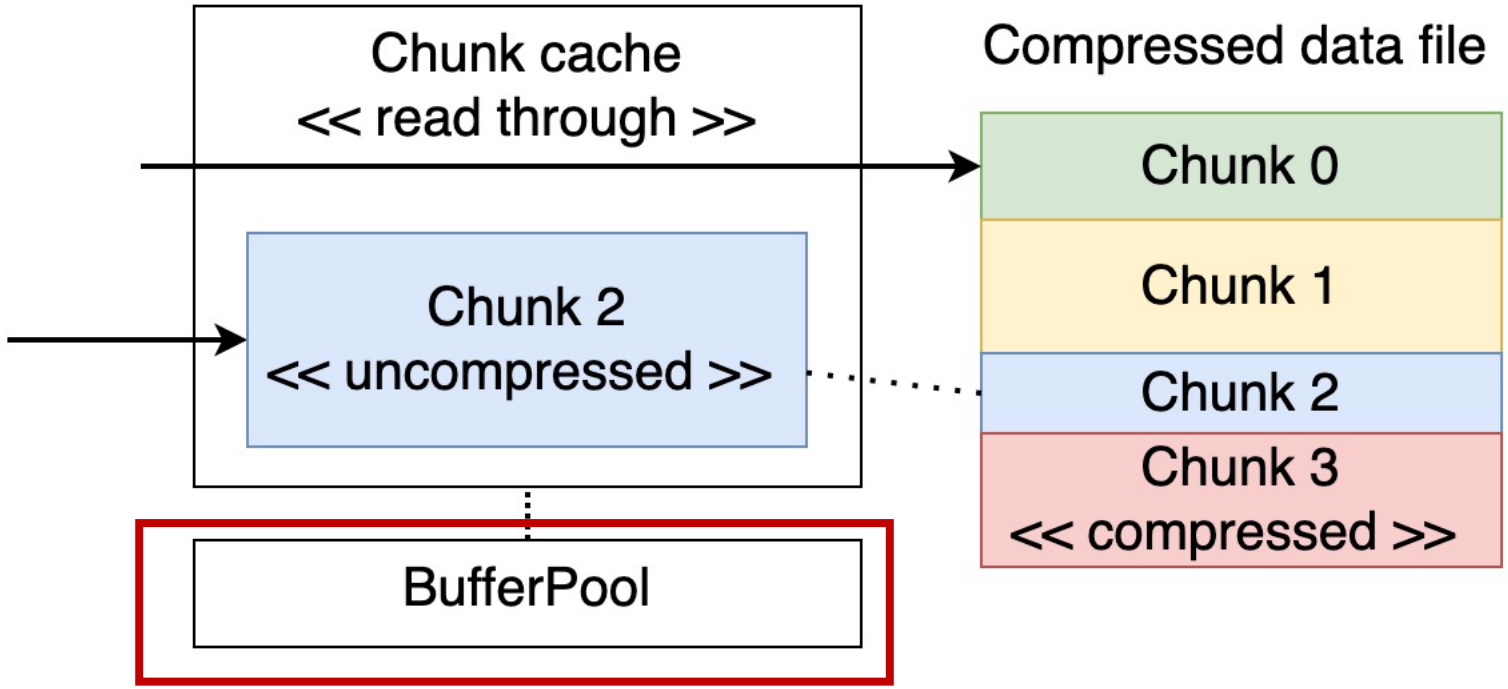


# Уровень хранения данных – chunk cache




- Caffeine
- Cassandra.yaml: file_cache_size_in_mb (default: min(512Mb, 25% of heap size) )

# Уровень хранения данных – chunk cache




- Caffeine
- Cassandra.yaml: file_cache_size_in_mb (default: min(512Mb, 25% of heap size) )
- CASSANDRA-15229 (Cassandra 4.0, Segregate Network and Chunk Cache BufferPools and Recirculate Partially Freed Chunks)

- Запись, уровень кластера
- Запись, уровень реплики
- Базовый тест
- Клиент-серверное взаимодействие
- Координация
- Хранилище
  - Row cache
  - Bloom filter
  - Поиск по SSTable
  - Сжатие
  - Чтение с диска

# Уровень хранения данных – чтение с диска

- disk_access_mode в cassandra.yaml:
  - mmap

```
ByteBuffer buffer = channel.map(FileChannel.MapMode.READ_ONLY, pos, size);
```

# Уровень хранения данных – чтение с диска

- disk_access_mode в cassandra.yaml:
  - mmap

```
ByteBuffer buffer = channel.map(FileChannel.MapMode.READ_ONLY, pos, size);
```

  - standard

```
FileChannel.open(file.toPath(), StandardOpenOption.READ);
channel.read(buffer, chunk.offset);
```

# Уровень хранения данных – чтение с диска

- disk_access_mode в cassandra.yaml:
  - mmap

```
ByteBuffer buffer = channel.map(FileChannel.MapMode.READ_ONLY, pos, size);
```

  - standard

```
FileChannel.open(file.toPath(), StandardOpenOption.READ);
channel.read(buffer, chunk.offset);
```

  - mmap_index_only

    Index – mmap, data – standard

# Уровень хранения данных – чтение с диска

- disk_access_mode в cassandra.yaml:
  - mmap

```
ByteBuffer buffer = channel.map(FileChannel.MapMode.READ_ONLY, pos, size);
```

  - standard

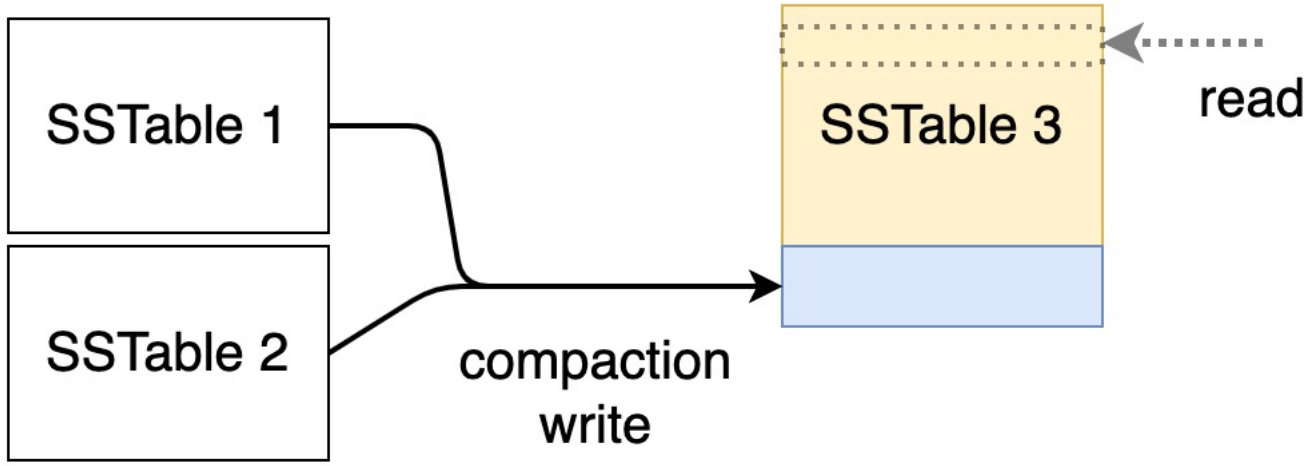
```
FileChannel.open(file.toPath(), StandardOpenOption.READ);
channel.read(buffer, chunk.offset);
```

  - mmap_index_only

    Index – mmap, data – standard
  - auto (default)
    - 64-bit OS => mmap
    - 32-bit OS => standard

# Уровень хранения данных – чтение с диска

sstable_preemptive_open_interval_in_mb - compaction еще пишет файл, а мы уже читаем (способ сгладить page cache churn)



# Уровень хранения данных – чтение с диска

- disk_optimization_strategy
  - ssd (default): иногда +4096 bytes к чтению
  - spinning:  всегда +4096 bytes к чтению
  - **Имеет смысл только для standard + disabled compression режима**
- CASSANDRA-14466 (open) – O_DIRECT?

- Запись, уровень кластера
- Запись, уровень реплики
- Базовый тест
- Клиент-серверное взаимодействие
- Координация
- Хранилище
- GC

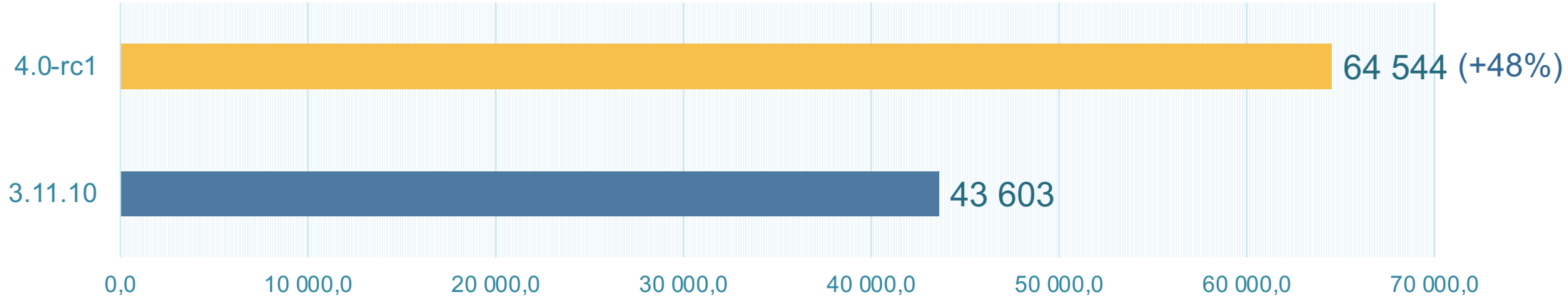
# Чтение и GC

- Shenandoah GC, ZGC
- https://thelastpickle.com/blog/2020/06/29/cassandra_4-0_garbage_collectors_performance_benchmarks.html
- https://www.scylladb.com/2020/10/06/c-scylla-in-battle-royale-against-javas-zgc-shenandoah-g1-cassandra/
- https://medium.com/outbrain-engineering/leveraging-shenandoah-to-cut-cassandras-tail-latency-d2a11725e4d8

- Запись, уровень кластера
- Запись, уровень реплики
- Базовый тест
- Клиент-серверное взаимодействие
- Координация
- Хранилище
- GC
- Cassandra 3.11.10 vs Cassandra 4.0

# Cassandra 4.0 vs 3.11

- Cassandra 4.0-rc1
- Пропускная способность



# Cassandra 4 vs 3.11

## Время ответа, фиксированная нагрузка: 25k per sec

| | 3.11.10 | 4.0.rc1 |
|---|---|---|
| Latency mean : | 3.8 ms | 2.4 ms |
| Latency median : | 1.3 ms | 1.1 ms |
| Latency 95th percentile : | 19.2 ms | 4.8 ms |
| Latency 99th percentile : | 42.0 ms | 23.3 ms |

Эпилог

# Итого

# Итого

Теперь вы знаете, куда и как “ударить” Cassandra

# Итого

- Пробуйте Cassandra 4.0-RC
- Пробуйте свежие JDK (MD5, SHA-1/2)

# Итого

- Пробуйте Cassandra 4.0-RC
- Пробуйте свежие JDK (MD5, SHA-1/2)
- Используйте профайлер для поиска узких мест
- CS можно применять на практике (Фильтр Блума, выбор реплики)

# Итого

- Пробуйте Cassandra 4.0-RC
- Пробуйте свежие JDK (MD5, SHA-1/2)
- Используйте профайлер для поиска узких мест
- CS можно применять на практике (Фильтр Блума, выбор реплики)
- В Java на любую тему найдется хорошая библиотека (Caffeine, LZ4)

# Thank You

Приложение

# Дополнительные слайды

# Метрики базы, избранное

- Встроенные метрики уровня базы - **RED** (Rate, Errors, Duration)
  - Ошибки
    - org.apache.cassandra.metrics:type=ClientRequest,scope=*,name=Timeouts
    - org.apache.cassandra.metrics:type=ClientRequest,scope=*,name=Failures
    - org.apache.cassandra.metrics:type=ClientRequest,scope=*,name=Unavailables
    - 
    - org.apache.cassandra.metrics:type=DroppedMessage,scope=*,name=Dropped
    - org.apache.cassandra.metrics:type=Storage,name=Exceptions

# Метрики базы, избранное

- Встроенные метрики уровня базы - **RED** (Rate, Errors, Duration)
  - Rate – производная от Count + Latency (среднее, персентили)
    - org.apache.cassandra.metrics:type=ClientRequest,scope=*,name=Latency
    - org.apache.cassandra.metrics:keyspace=*,name=ReadLatency,scope=*,type=Table
    - org.apache.cassandra.metrics:keyspace=*,name=CoordinatorReadLatency,scope=*,type=Table

# Метрики базы, избранное

- Пулы потоков
  - ...:type=ThreadPools,name=ActiveTasks,path=internal,scope=*
  - ...:type=ThreadPools,name=PendingTasks,path=internal,scope=*
  - ...:type=ThreadPools,name=TotalBlockedTasks,path=internal,scope=*

- Количество SSTables, читаемых во время read операции
  - …:keyspace=*,name=SSTablesPerReadHistogram,scope=*,type=Table
- Tombstones
  - …:keyspace=*,name=TombstoneScannedHistogram,scope=*,type=Table

# Трассировка запросов

- Трассировка % запросов
- nodetool settraceprobability 0.0001
- Больше деталей: https://www.datastax.com/dev/blog/tracing-in-cassandra-1-2

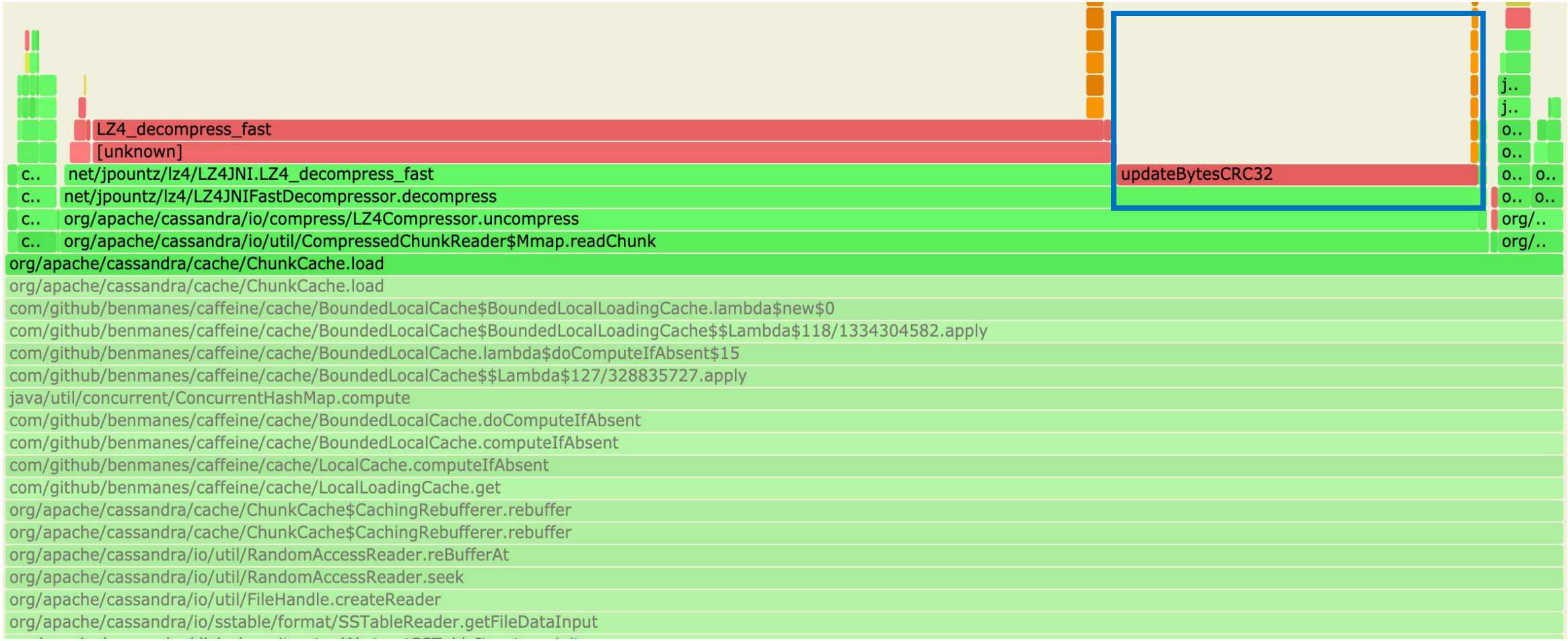
# Трассировка запросов - пример

- SELECT запрос, consistency_level = LOCAL_ONE

```
Tracing session: 6930c380-ddea-11e9-97f8-952ee9e78380

                 activity                                       | timestamp       |source | elapsed |
-----------------------------------------------------------------+-----------------+-------+---------+
                                    Execute CQL3 query | 06:10:46.840000 | ...49 |       0 |
                  Parsing query <skipped> ; [CoreThread-1] | 06:10:46.840001 | ...49 |     121 |
                       Preparing statement [CoreThread-1] | 06:10:46.840001 | ...49 |     231 |
             Reading data from [/10.101.18.49] [CoreThread-1] | 06:10:46.840001 | ...49 |     362 |
Executing single-partition query on test_table [CoreThread-4] | 06:10:46.840002 | ...49 |     691 |
                   Acquiring sstable references [CoreThread-4] | 06:10:46.840003 | ...49 |     725 |
     Merged data from memtables and 0 sstables [CoreThread-4] | 06:10:46.841000 | ...49 |     763 |
        Read 1 live rows and 0 tombstone cells [CoreThread-4] | 06:10:46.841000 | ...49 |     782 |
                                  Request complete | 06:10:46.840918 | ...49 |     918 |
```

# CRC




- Настройка на уровне таблицы, crc_check_chance = 1.0
- CASSANDRA-16360 (open) - CRC32C

# mmap issue

- https://support.datastax.com/hc/en-us/articles/360027838911
- https://support.datastax.com/hc/en-us/articles/360022747391-FAQ-Use-of-disk-access-mode-in-DSE-5-1-and-earlier

  “In cases where there are lots of random reads, and the set of SSTables being heavily read is larger than the available memory, the affected nodes will have a high number of page faults. In some cases, the affected servers run out of memory and the Linux oom-killer terminates DSE.”

- CASSANDRA-15531