

## 1. Описание выбранных методов тестирования

В рамках проекта не предусмотрено тестирование на наличие багов, т. к. основная цель проекта заключается в разработке архитектуры, удовлетворяющей требованиям по пропускной способности, доступности и гибкости масштабирования.

Требования:

- Заполненность БД с событиями пользователей - 10\_000\_000\_000;
- Количество пользователей в БД - 1\_000\_000;
- Количество событий на одного пользователя - 1000-10000;
- Размер страницы (количество извлекаемых событий за один запрос) - 100;
- Система должна обеспечивать пропускную способность на чтение 250 RPS;
- Система должна обеспечивать пропускную способность на запись 8000 RPS.

Для верификации выполнения требований тестирования лучше всего подходит нагрузочное тестирование.

Инструментарий для тестирования:

- Prometheus - используется для сбора метрик работы сервисов;
- Grafana - используется для визуализации собранных метрик сервисов;
- Yandex Tank - используется как фреймворк для нагрузочного тестирования;
- Phantom - используется как источник нагрузки в Yandex Tank;
- h2load - используется как альтернативный источник нагрузки для стресс тестов;
- Overload - используется для визуализации метрик с тестирования Yandex Tank.

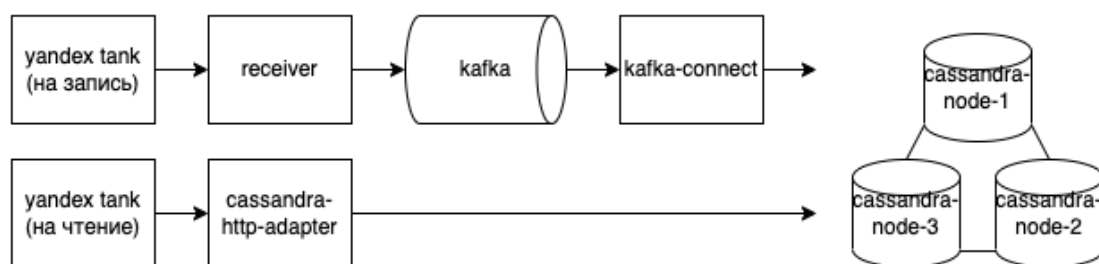
Нагрузочное тестирование проводили на 3 виртуальных дата-центрах (ВДЦ) в Mts Cloud. Каждый ВДЦ имеет следующие характеристики:

- Процессор: Xeon Gold 6254 (6 виртуальных ядер);
- Оперативная память: 24 ГБ;
- Дисковое пространство: 150 ГБ (SSD\_Ultra\_Plus\_Policy).

Распределение ресурсов в системе:

| Компонент     | RAM (gb) | vCPUs | SSD (gb) | Количество |
|---------------|----------|-------|----------|------------|
| Cassandra     | 10       | 2     | 132      | 3          |
| Kafka         | 1        | 1     | 16       | 1          |
| Kafka connect | 1        | 1     | 16       | 1          |
| Receiver      | 1        | 4     | 16       | 1          |

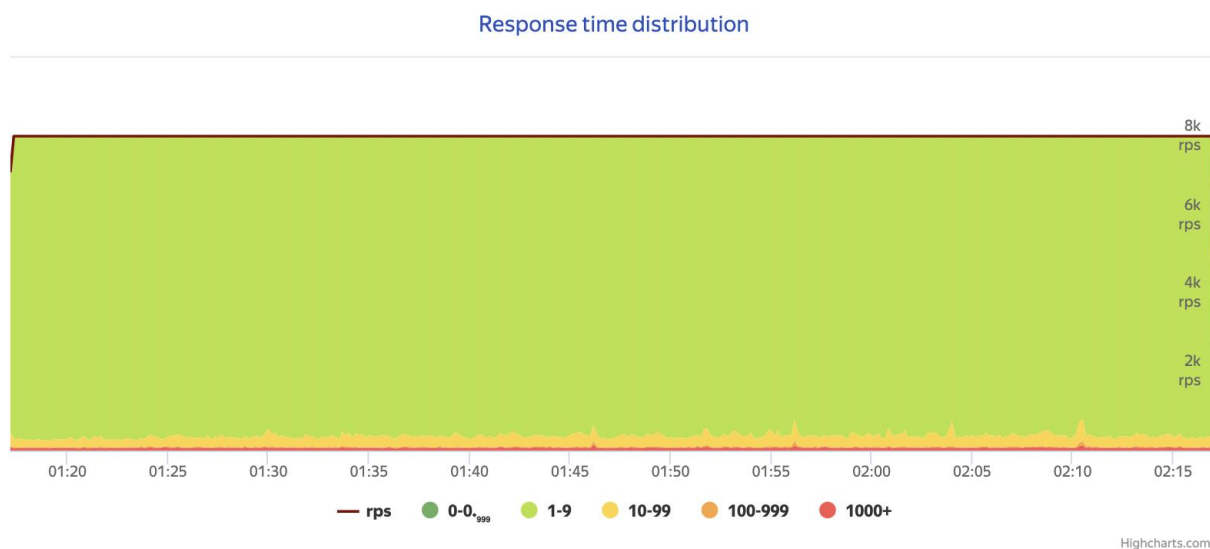
Тестируемая система:



## Тестирование постоянной нагрузки

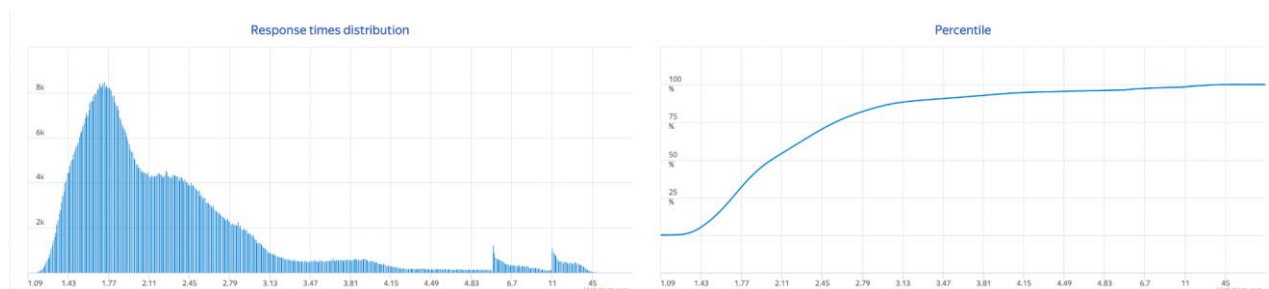
В этом тесте одновременно проводилась запись событий с постоянной скоростью 8000 RPS и чтение из кассандры с постоянной скоростью 250 RPS. Тест запускался на 1 час.

График нагрузки ресивера (RPS) от времени:



Отсутствие пустот на графике показывает, что сервис справляется с нагрузкой и не выдаёт timeout. То, что график почти полностью заполнен зеленым значит, что большая часть ответов приходит за время от 1 мс до 9 мс.

График распределение времени ответа ресивера:



По графикам видно, что большинство запросов выполнялись в пределах 3-5 мс.

## Метрики сервиса ресивера:



По графикам работы сервиса ресивера можно убедиться в записи событий со скоростью 8000 RPS. Также видно, что ресивер тратит на обработку одного запроса не более 1 мс, а в среднем 0.1 мс.

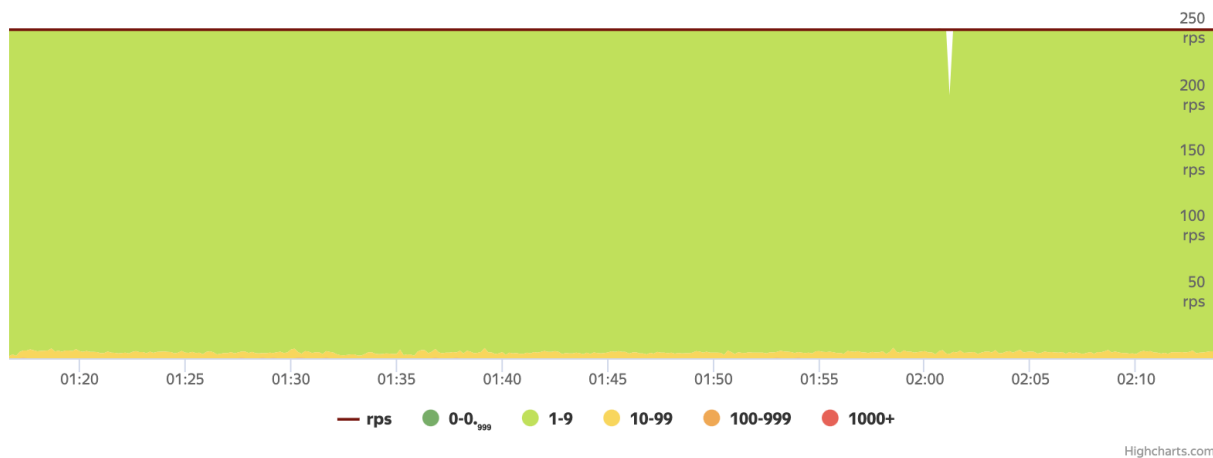
## Метрики сервиса kafka connect:



Верхний левый график подтверждает поток записи в кассандру на уровне 8000 RPS.

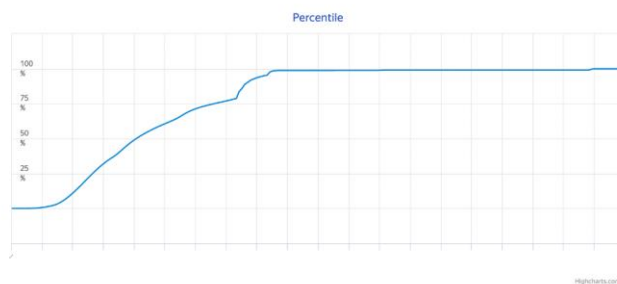
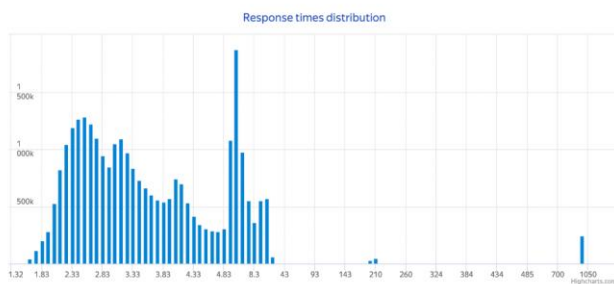
## График нагрузки кассандры на чтение (RPS) от времени:

Response time distribution



По графику видно, что кассандра стабильно отдаёт 250 RPS на чтение, наибольшая доля запросов обрабатывается за 9 мс.

## Распределение времен ответов запросов на чтение из кассандры:



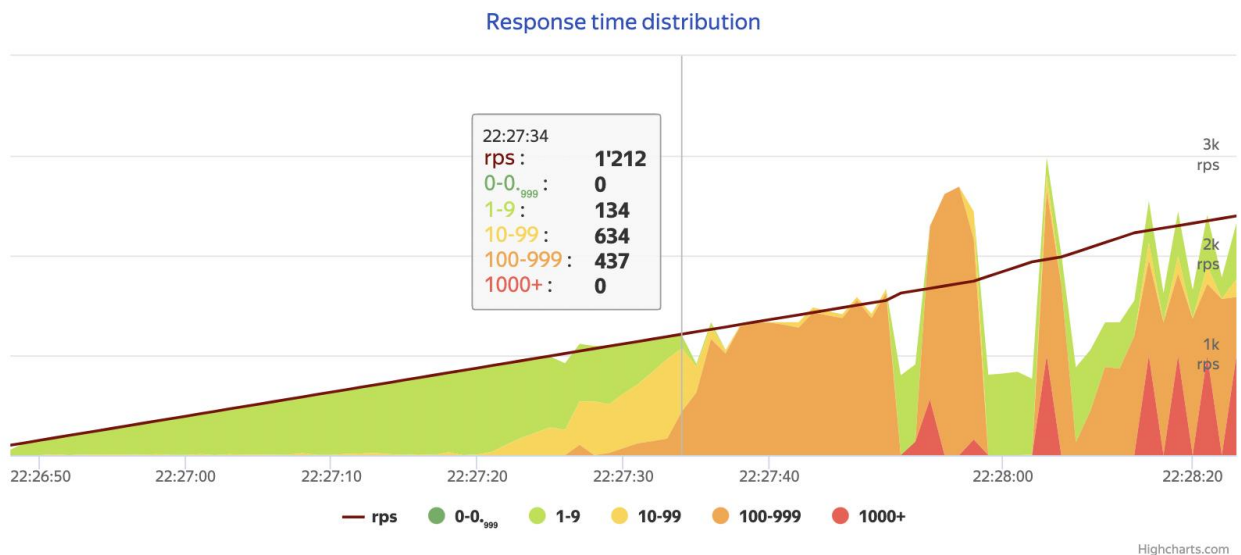
## Метрики http-адаптера для кассандры:



## Стресс тест чтения

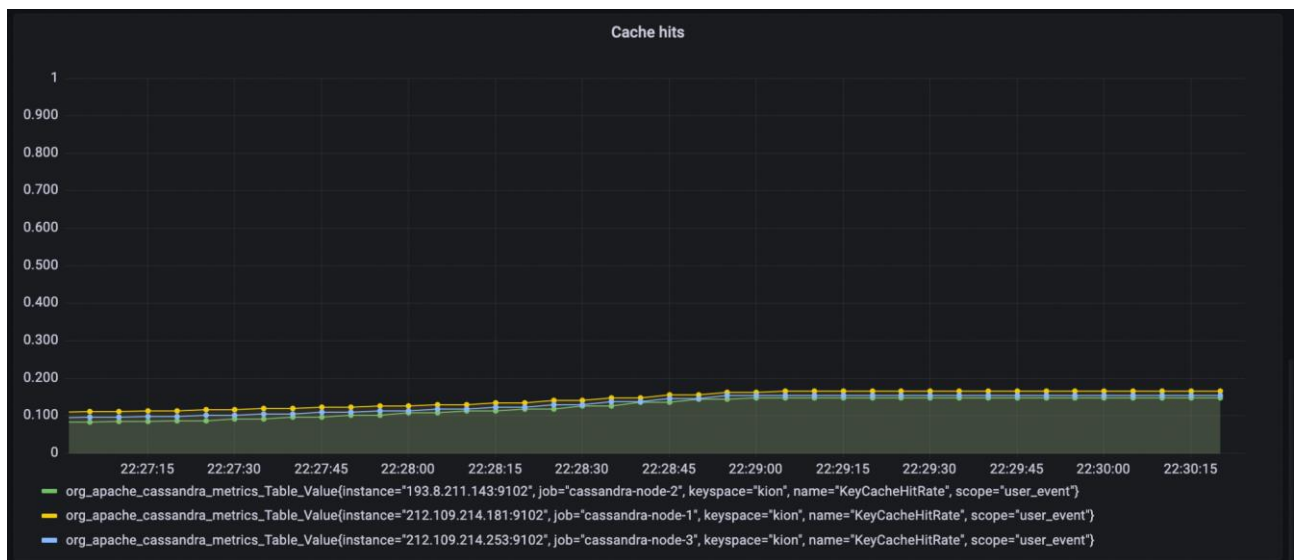
Попытка найти предельную пропускную способность кассандры на чтение при размере страницы в 100 записей.

**График нагрузки кассандры (RPS) от времени:**



Кассандра перестает справляться примерно при скорости 1200 RPS.

**Доля попаданий ключей в кеш кассандры:**



По нашим наблюдениям key cache кассандры довольно значительно влияет на скорость чтения. В этом тесте кеш не успел заполниться значительно, поэтому скорее всего предельная пропускная способность может быть немного выше в постоянно нагруженной системе.

## Стресс тест записи

Стресс тест записи проводился с помощью инструмента h2load. Тестирование проводилось не в vdc во владивостоке, а из Московской области.

### Метрики ресивер сервиса:



Максимальный выдаваемый RPS в районе 12000.

### Метрики сервиса kafka connect:



Максимальная скорость записи в кассандру так же в районе 12000 RPS.

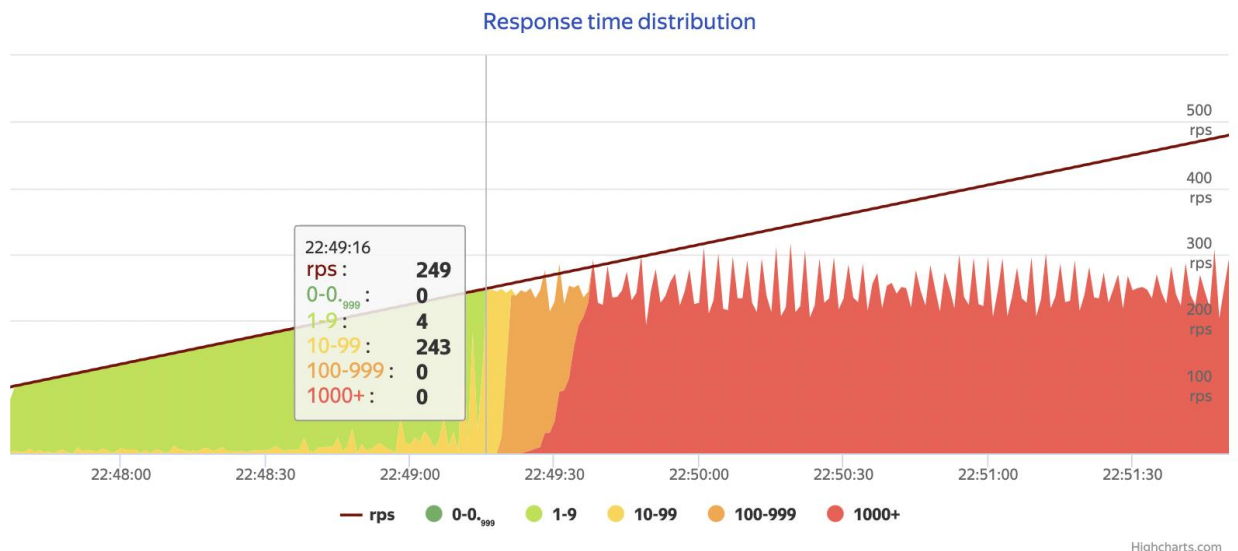
## Тест чтения при предельном размере страницы

Опытным путем было выяснено, что кластер кассандры выдерживает нагрузку 250 RPS при размере страницы < 1000 записей. При размере страницы 800 записей по графикам время ответа кассандры не деградирует, кластер стабильно выдает 250 RPS.

### Графики нагрузки кассандры (RPS) от времени:



По графикам видно, что время ответов не деградирует и таймаутов также не происходит. При размере страницы 1000 записей начинаются таймауты после 250 RPS.



### Вывод

По результатам тестирования на выделенных ресурсах сервис выдерживает 8000 RPS на запись и 250 RPS на чтение. В течение 1 часа теста под постоянной нагрузкой никаких просадок и других проблем не нашли.

Есть небольшой запас на чтение и запись. Лучшая зафиксированная скорость чтения была в районе 1200 RPS (запас 950 RPS), лучшая скорость записи в районе 12\_000 RPS (запас 4\_000 RPS).

На пропускную способность запросов на чтение сильно влияет размер отдаваемой страницы. При размере страницы в 100 записей сервис способен обрабатывать поток 1200 RPS. Предельный размер страницы, при котором сервис может обрабатывать поток на чтение в 250 RPS - в районе 800 записей.

Максимальная скорость записи напрямую в кластер кассандры была в районе 60\_000 записей в секунду. Во время тестирования запись проходила через цепочку receiver → kafka → kafka connector → cassandra. Для экономии ресурсов пришлось создать по одному экземпляру каждого компонента в этой цепочке, поэтому скорость записи сильно меньше предельной (12\_000 RPS против 60\_000 RPS). Пропускную способность записи можно



увеличивать до предельной добавляя экземпляров в этой цепочке. Предельную пропускную способность записи можно увеличивать, расширяя кластер кассандры.

## 2. Приёмочное тестирование (заказчик проверяет соответствие ТЗ реализованному функционалу)

Заказчик подтверждает полученные результаты:

| № п/п  | Наименование работ в соответствии с ТЗ   | Статус<br>(статус заполняет заказчик):<br>выполнено/частично выполнено/не выполнено |
|--|--|---|
| 1.   | Придумать и описать механизм взаимодействия между клиентом и receiver сервисом | <u>ДА</u> / НЕТ / ЧАСТИЧНО  |
| 2.   | Разработать механизм взаимодействия receiver сервиса и хранилища               | <u>ДА</u> / НЕТ / ЧАСТИЧНО  |
| 3.   | Разработать механизм сбора метрик  | <u>ДА</u> / НЕТ / ЧАСТИЧНО  |
| 4.   | Выбрать веб сервер   | <u>ДА</u> / НЕТ / ЧАСТИЧНО  |
| 5.   | Выбрать брокера сообщений  | <u>ДА</u> / НЕТ / ЧАСТИЧНО  |
| 6.   | Выбрать хранилище  | <u>ДА</u> / НЕТ / ЧАСТИЧНО  |
| 7.   | Разработка receiver сервиса, конфигурация jvm                                  | <u>ДА</u> / НЕТ / ЧАСТИЧНО  |
| 8.   | Конфигурация кафки   | <u>ДА</u> / НЕТ / ЧАСТИЧНО  |
| 9.   | Конфигурация clickhouse  | <u>ДА</u> / НЕТ / ЧАСТИЧНО  |
| 10.  | Конфигурация cassandra   | <u>ДА</u> / НЕТ / ЧАСТИЧНО  |
| 11.  | Добавить экспорт метрик из сервисов  | <u>ДА</u> / НЕТ / ЧАСТИЧНО  |
| 12.  | Развертывание компонентов в ВДЦ  | <u>ДА</u> / НЕТ / ЧАСТИЧНО  |
| 13.  | Провести нагрузочное тестирование  | <u>ДА</u> / НЕТ / ЧАСТИЧНО  |
| Перечень НЕ выполненных работ<br>(заполняет заказчик): |  | 1.<br>2.<br>3.  |

Комментарии заказчика к результатам: \_\_\_\_\_

\_\_\_\_\_ Студентами выполнены все поставленные задачи, при нагрузочном тестировании проблем не обнаружено.

Заказчик

Личная подпись

Ответственный по проекту

Личная подпись

Дата:

25 марта 2022