# **Тест №4.6. «Тест объема» API Ввода проводки**

# Параметры теста

* Каждый пользователь выполняет ввод проводки по API Create

## Время проведения теста

2025-09-25 (18:36:00 - 19:57:00)

## Ссылка на тест

https://graf.protod.moscow.alfaintra.net/d/b40939ab-c9b2-465a-abb1-4b479a2c2d4c/jmeter-dashboard-victoriametrics?orgId=1&from=2025-09-25T15:36:00.000Z&to=2025-09-25T16:57:00.000Z&timezone=browser&var-data\_source=P4169E866C3094E38&var-application=uc\_4\_6\_linux.jmx&var-transaction=$\_\_all&var-Filters=

## Команда для запуска

~/apache-jmeter-5.6.3/bin/jmeter -n -t ~/git\_absrnd/lt/uc/grpc/uc\_4\_6\_linux.jmx -Juser\_rpm=60

## План подачи нагрузки

* Заполнение БД тестовыми данными в общем на 1 терабайт (заполняем 2 таблицы: **/Root/asdb/remains/t\_balance** и **/Root/asdb/journals/t\_mainapp\_funds\_movement**)
* RumpUp 10 минут
* Выход на ступень со стабильной нагрузкой 100% профиля;
* Удержание нагрузки 60 минут;
* RumpDown 10 минут

## Конфигурация теста

Пейсинг - максимальное время выполнения 1 запроса пользователем

Таблица 1 – параметры теста

|  |  |
| --- | --- |
| **Название** | **Параметр** |
| Thread Group | Ultimate Thread Group |
| Количество выполняемых запросов в секунду 1 пользователем | 1 рпс |
| Пейсинг | 1 сек |
| Таймаут по запросу | 1 сек |
| SLA % ошибок | 5 % |
| SLA времени выполнения запроса | - |
| Прогретость БД (количество данных в 1 таблице) | 50 ГБ |
| Количество инстансов приложения | 3 |
| Параметры приложения | worker\_threads="8"  worker\_fs\_threads="2"  worker\_grpc\_threads="2"  monitor\_worker\_threads="1"  ydb\_max\_pool\_size="8"  ydb\_min\_pool\_size="4"  topic\_reader\_consumers\_number="10"  topic\_sender\_non\_part\_write\_sessions\_number="10" |
| Базовый профиль нагрузки | 50 запросов в секунду |

# Результаты

## Описание

Определена производительность приложения при наполнении дискового хранилища БД на 92% на нагрузке 100% от базового профиля во время выполнения запроса API ввода проводки.

* Наполнение БД не повлияло на производительность приложения
* Количество запросов в секунду сохранилось, составляет 50 рпс, количество ошибок до 2%, время ответа в среднем 883 мс.

## Таблица с результатами

Среднее значение ключевых показателей НТ на целевой ступени нагрузки

Таблица 2 – результаты теста

|  |  |  |  |
| --- | --- | --- | --- |
| **Интенсивность запросов** | **Время ответа** | **% количества ошибок** | **Пользователи** |
| <Таблица\_2\_1\_2\_Интенсивность\_запросов> | <Таблица\_2\_2\_2\_Время\_ответа> | <Таблица\_2\_3\_2\_Процент\_количества\_ошибок> | <Таблица\_2\_4\_2\_Пользователи> |

# Бизнес метрики (JMeter)

## Throughput

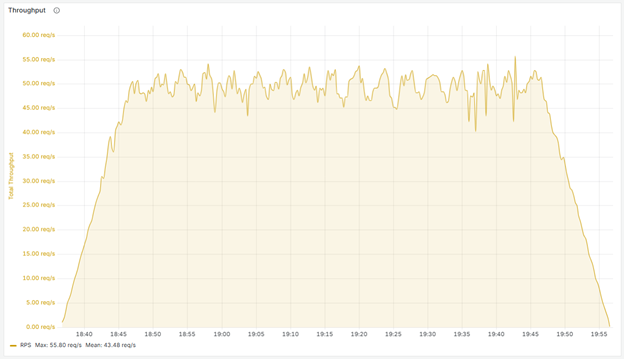


График интенсивности запросов в секунду.

<LLM\_Throughput>

## Virtual Users

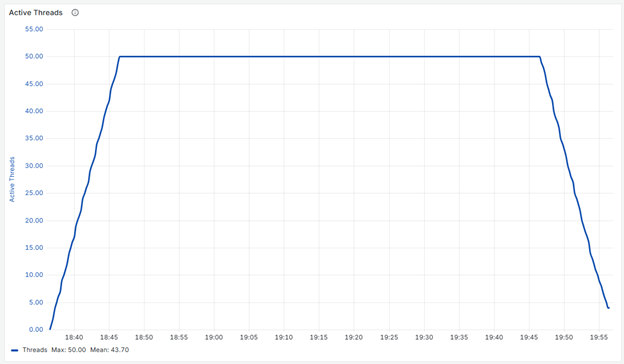


График выхода пользователей на рабочую нагрузку.

<LLM\_VirtualUsers>

## Response time (95th percentile)

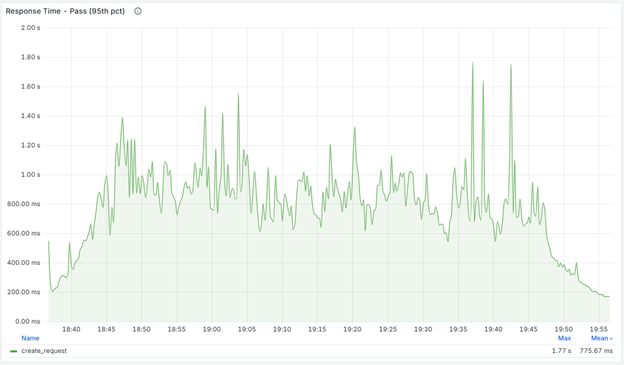


График времени ответа по успешным запросам.

<LLM\_ResponseTime>

## Percent errors

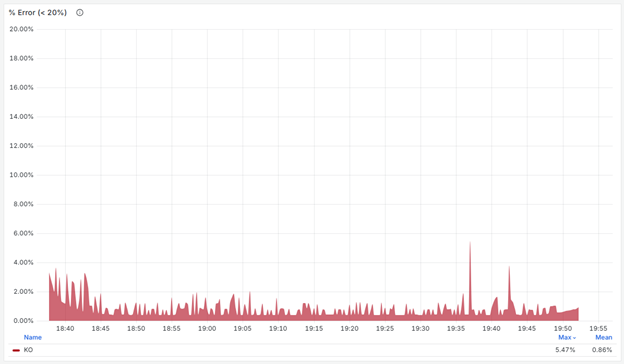


График с ошибками на протяжении всего теста в процентах (до 20 процентов).

<LLM\_PercentErrors>

## Count error in table

Ниже представлена таблица с количеством пяти главных ошибок за весь тест с сообщением:

Таблица 3 – Топ 5 ошибок за тест

|  |  |  |  |
| --- | --- | --- | --- |
| **Transaction** | **Code** | **Response Message** | **Sum ↓** |
| create\_request | 1001 | Success: false, Code: 1001, Status: RESULT\_STATUS\_REMAINS\_RESPONSE\_TIMEOUT, Description: Balance change response timeout. | 8833 |
| create\_request | 1000 | Success: false, Code: 1000, Status: RESULT\_STATUS\_SYSTEM\_ERROR, Description: YDB Commit failed: { <main>: Error: Operation aborted., code: 2011 } | 161 |
| create\_request | 1000 | Success: false, Code: 1000, Status: RESULT\_STATUS\_SYSTEM\_ERROR, Description: YDB Commit failed: [ { <main>: Error: Operation aborted., code: 2011 } { <main>: Error: Transaction locks invalidated. Tables: `/Root/asdb/journals/t\_mainapp\_funds\_movement`, code: 2001 } ] | 69 |
| create\_request | 1000 | Success: false, Code: 1000, Status: RESULT\_STATUS\_SYSTEM\_ERROR, Description: YDB Commit failed: [ { <main>: Error: Operation aborted., code: 2011 } { <main>: Error: Transaction locks invalidated. Tables: `/Root/asdb/journals/t\_mainapp\_funds\_movement/idx\_instance\_id\_and\_status/indexImplTable`, code: 2001 } ] | 58 |
| create\_request | 1000 | Success: false, Code: 1000, Status: RESULT\_STATUS\_SYSTEM\_ERROR, Description: SessionClosed { Status: NOT\_FOUND Issues: "<main>: Error: status is not ok: <main>: Error: ydb/core/kqp/session\_actor/kqp\_session\_actor.cpp:2964: Unable to navigate:path: 'Root/asdb/topics/q\_remains\_responses/8db12696-5b8f-403e-87e6-5dc4517bq5f1' status: LookupError , code: 500030 " } | 48 |

### Анализ таблицы JMeter - CountErrorInTable  
  
#### \*\*1. Есть ли деградация?\*\*  
\*\*Да\*\*  
  
#### \*\*2. Короткое обоснование с временными интервалами и процентными изменениями\*\*  
- В таблице видны ошибки с различными кодами и описаниями:  
 - \*\*Code: 1001\*\* (RESULT\_STATUS\_REMAINS\_RESPONSE\_TIMEOUT): Ошибка таймаута ответа (8833 ошибки).  
 - \*\*Code: 1000\*\* (RESULT\_STATUS\_SYSTEM\_ERROR): Системные ошибки, включая:  
 - Ошибки коммита транзакций (YDB Commit failed, код 2011).  
 - Недействительные блокировки транзакций (Transaction locks invalidated, код 2001).  
 - Ошибки сессии (SessionClosed, код 500030).  
- Эти ошибки указывают на системные проблемы, связанные с транзакциями, блокировками и доступом к данным.  
- Хотя конкретные временные интервалы не указаны в данных, наличие множественных ошибок указывает на системную деградацию.  
  
#### \*\*3. Степень серьёзности\*\*  
\*\*Критическая\*\*  
  
#### \*\*4. Уровень уверенности (0–100%)\*\*  
\*\*90%\*\*  
  
#### \*\*5. 3–5 смежных графиков для проверки (и почему)\*\*  
1. \*\*Latency P95/P99\*\*:  
 - Почему: Проверить, не растет ли время ответа для критических запросов, что может быть причиной таймаутов.  
2. \*\*Error Rate\*\*:  
 - Почему: Подтвердить, что ошибка в таблице не является изолированной и действительно влияет на общую работоспособность системы.  
3. \*\*CPU и GC на DB-нодах\*\*:  
 - Почему: Проверить, не перегружены ли базовые компоненты системы, что может привести к транзакционным ошибкам.  
4. \*\*Queue Length\*\*:  
 - Почему: Проверить, не возникла ли очередь запросов, что может быть причиной таймаутов.  
5. \*\*Downstream Service Availability\*\*:  
 - Почему: Убедиться, что внешние сервисы доступны и работают корректно, так как проблемы с ними могут быть причиной системных ошибок.  
  
#### \*\*6. Первые шаги для расследования\*\*  
1. \*\*Анализ логов\*\*:  
 - Собрать и проанализировать логи на время возникновения ошибок для выявления корреляций.  
2. \*\*Проверка доступности сервисов\*\*:  
 - Убедиться, что все внешние сервисы доступны и работают корректно.  
3. \*\*Мониторинг базы данных\*\*:  
 - Проверить состояние базы данных (блокировки, транзакции, GC, CPU).  
4. \*\*Проверка конфигурации\*\*:  
 - Убедиться, что конфигурация системы соответствует текущей нагрузке.  
5. \*\*Репликация и резервное копирование\*\*:  
 - Проверить состояние репликации и резервного копирования для выявления возможных проблем с доступностью данных.  
  
### \*\*Краткий вывод\*\*  
Система демонстрирует признаки критической деградации, вызванной системными ошибками, таймаутами и проблемами с транзакциями. Необходимо незамедлительно провести расследование и исправить выявленные проблемы.

## Error by message

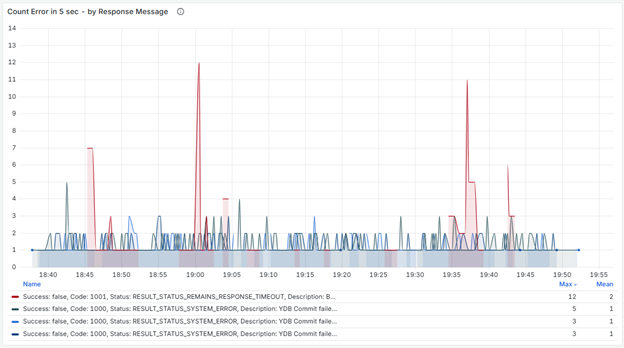
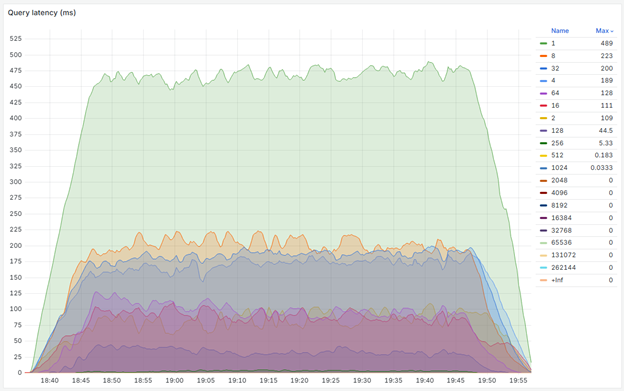


График с количеством ошибок (за период агрегации 5 секунд) в разрезе времени.

<LLM\_Error\_by\_message>

# Метрики БД

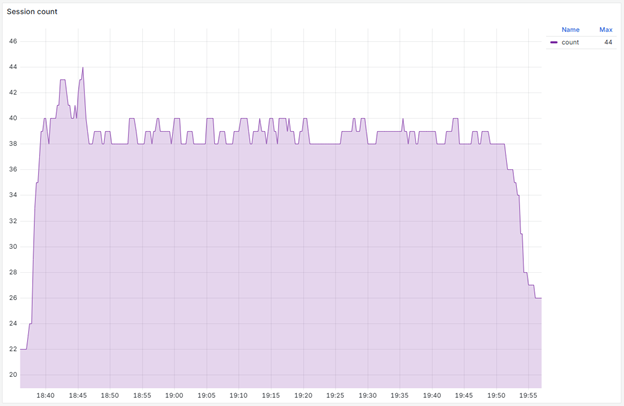
## Query latency (ms)



Время выполнения запроса от момента получения запроса YDB до возврата результата.

<LLM\_QueryLatency>

## Session count



Количество сессий БД на протяжении всего теста.

<LLM\_SessionCount>

## Requests

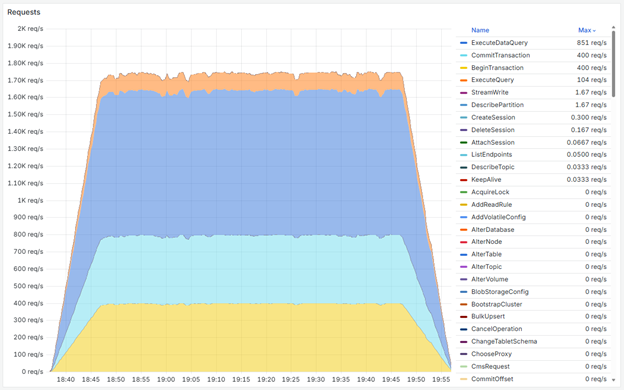


График количества выполненных запросов в секунду.

<LLM\_Requests>

## Errors type

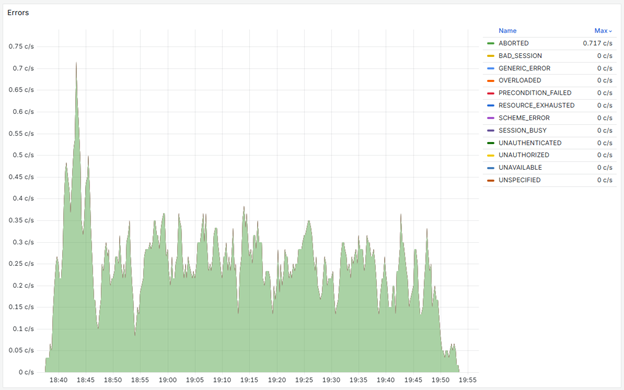


График с интенсивностью неуспешных запросов в секунду в разрезе типа ошибки.

<LLM\_ErrorsType>

## Dropped requests

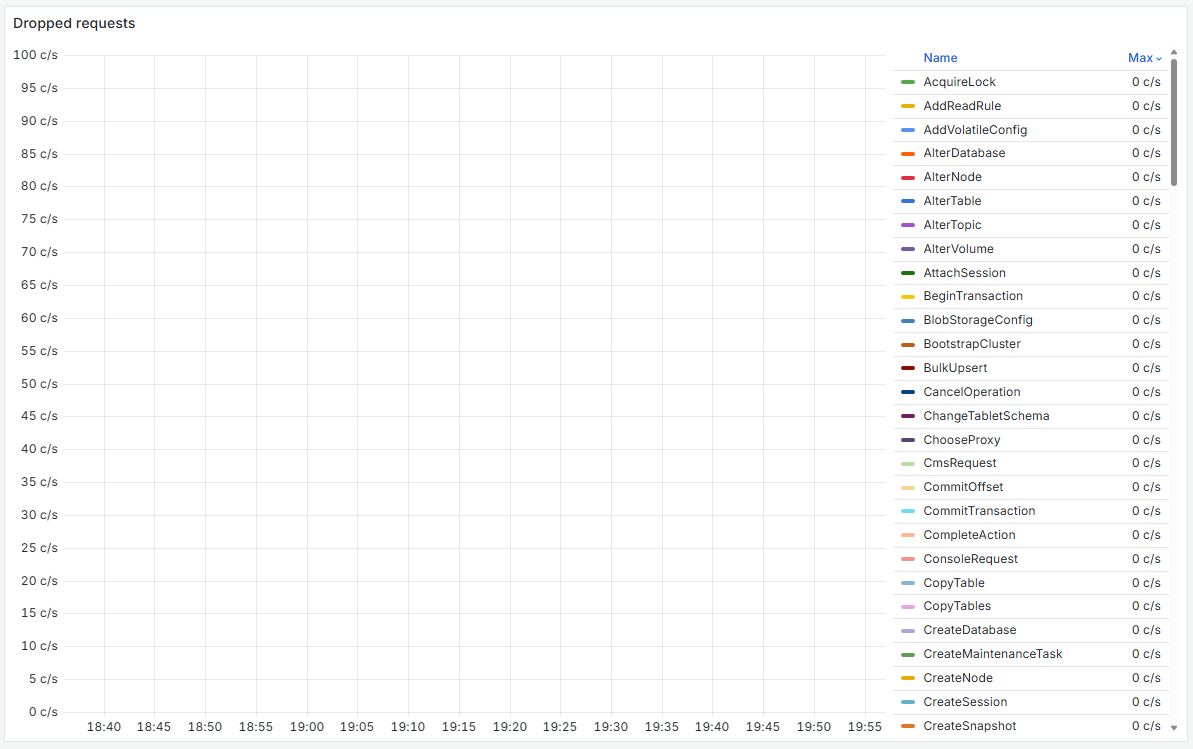


График с отброшенными запросами БД.

<LLM\_DroppedRequests>

## Dropped response

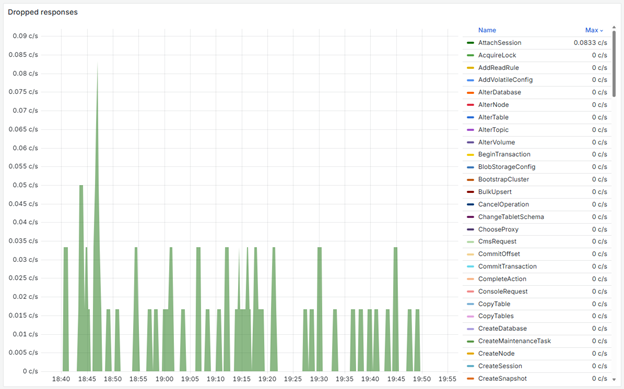


График с отброшенными ответами БД.

<LLM\_DroppedResponse>

## Requests in flight

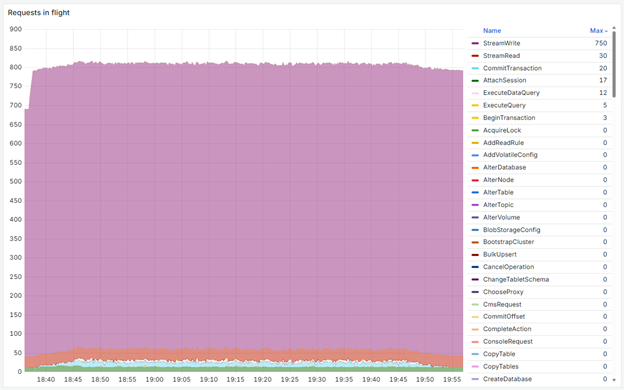
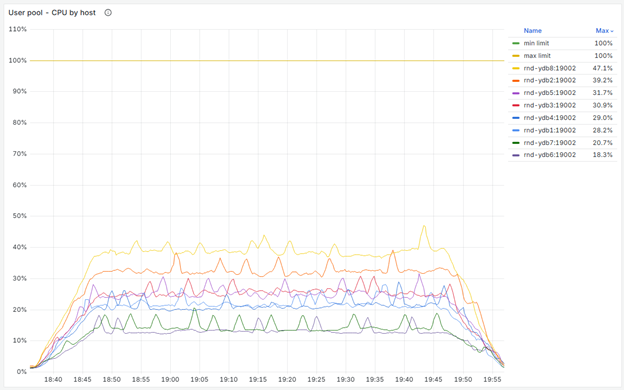


График количества запросов в обработке БД на протяжении всего теста.

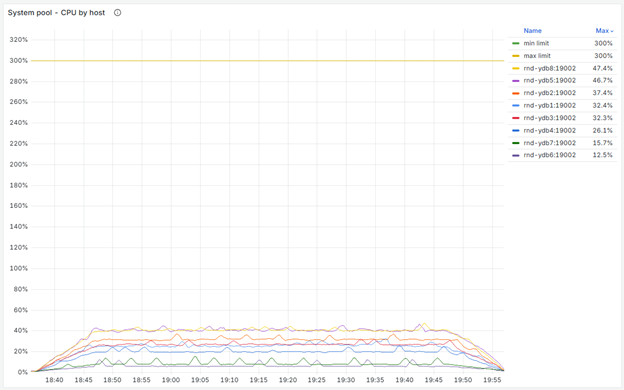
<LLM\_RequestsInFlight>

## User pool - CPU by host

 Потребление CPU пулом для пользовательских операций.

<LLM\_UserPool>

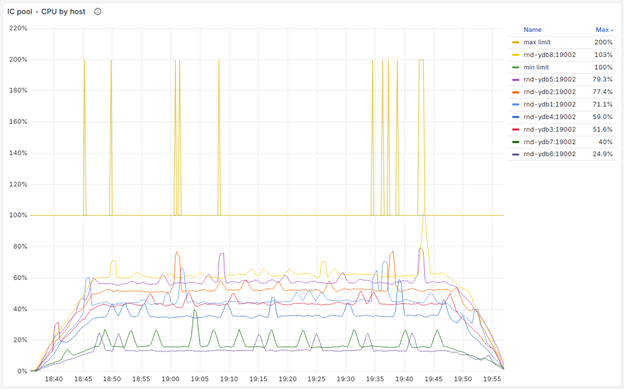
## System pool - CPU by host



Потребление CPU пулом для системных задач.

<LLM\_SystemPool>

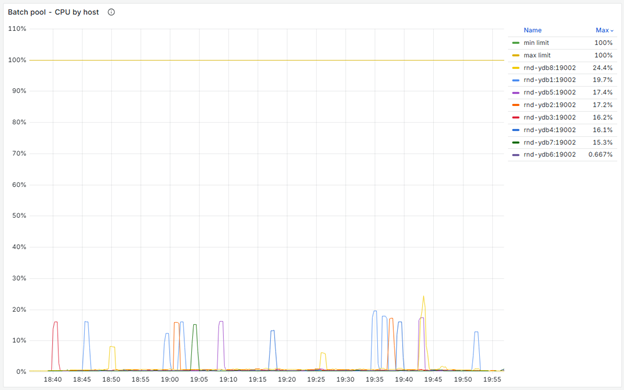
## IC pool - CPU by host



Потребление CPU пулом межсетевой коммуникации (Interconnect Pool) между узлами БД.

<LLM\_ICPool>

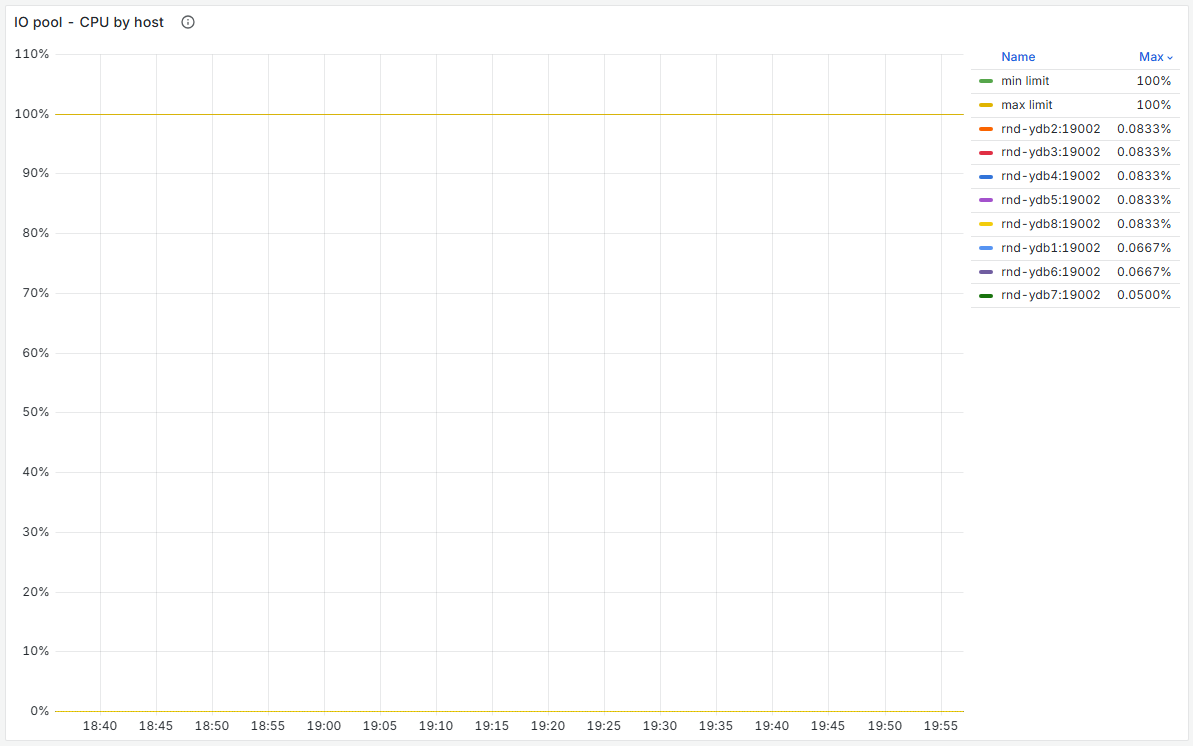
## Batch pool - CPU by host



Потребление CPU пулом пакетной обработки.

<LLM\_BatchPool>

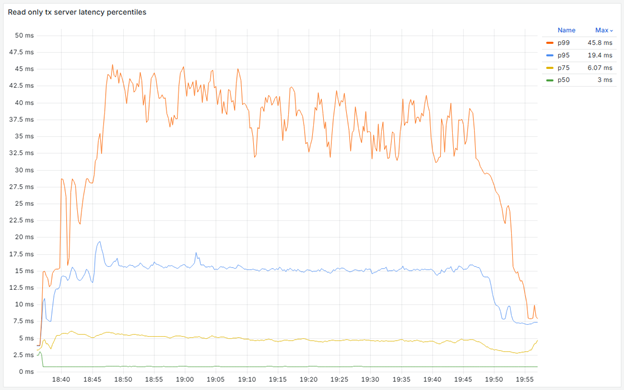
## IO pool - CPU by host



Потребление CPU пулом управления блокирующими операциями ввода-вывода при операциях чтения-записи.

<LLM\_IOPool>

## Read only tx server latency percentiles

 График с задержкой для чтения на протяжении всего теста.

<LLM\_TxReadOnly>

## Write only tx server latency percentiles

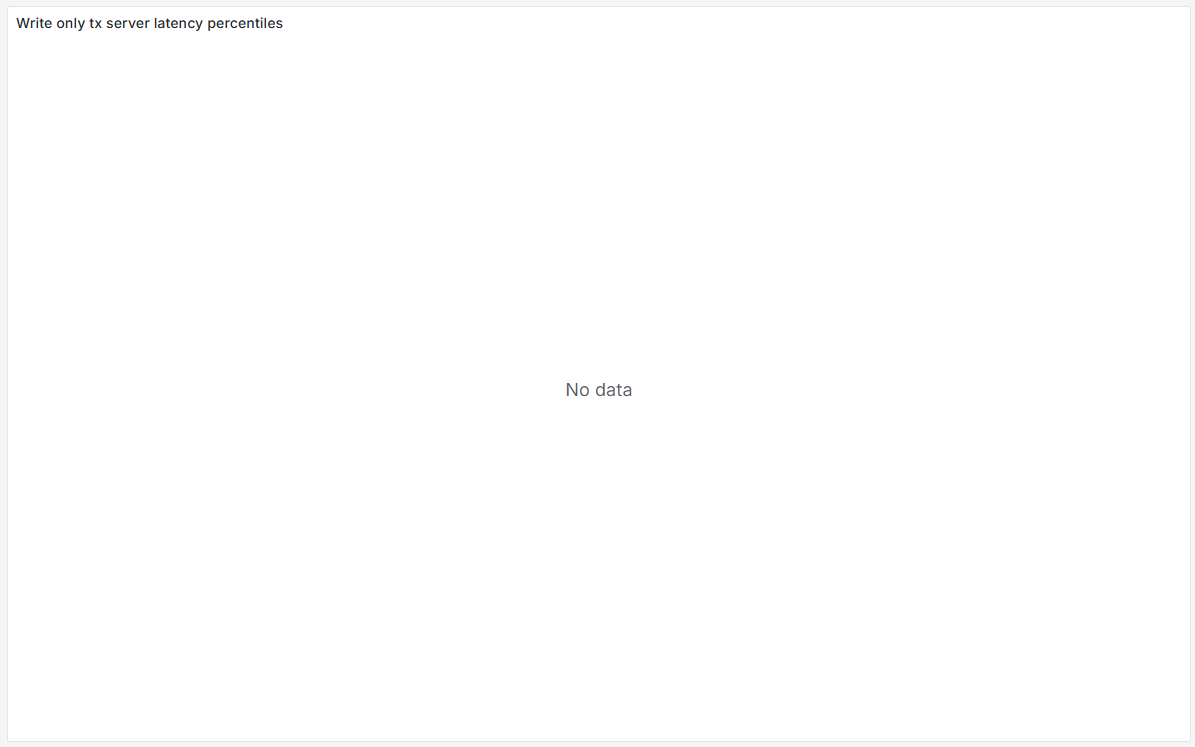


График с задержкой записи на протяжении всего теста.

<LLM\_TxWriteOnly>

## Read/Write tx server latency percentiles

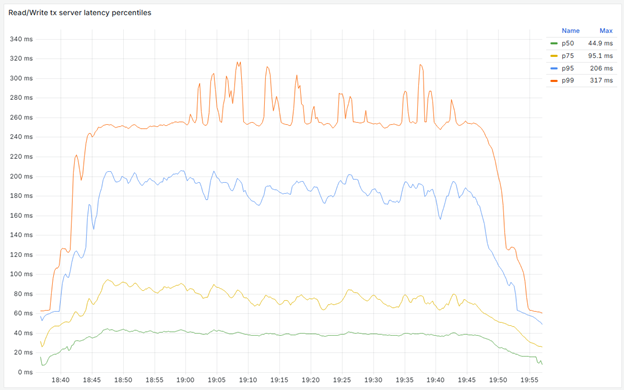
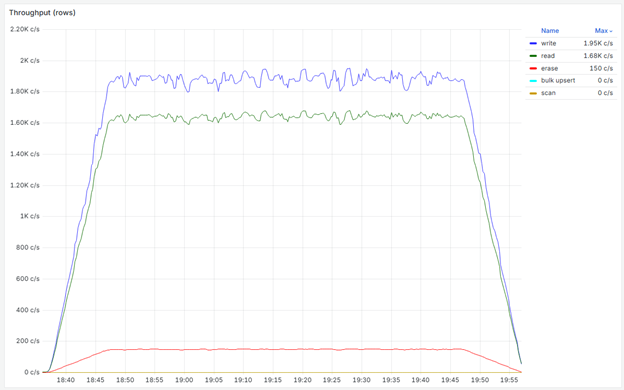


График с задержкой для чтения/записи на протяжении всего теста.

<LLM\_Error\_by\_message>

## DataShard. Throughput (rows).



Количество строк, обрабатываемых в секунду.

<LLM\_DataShardThroughput>

## Shard distribution by workload

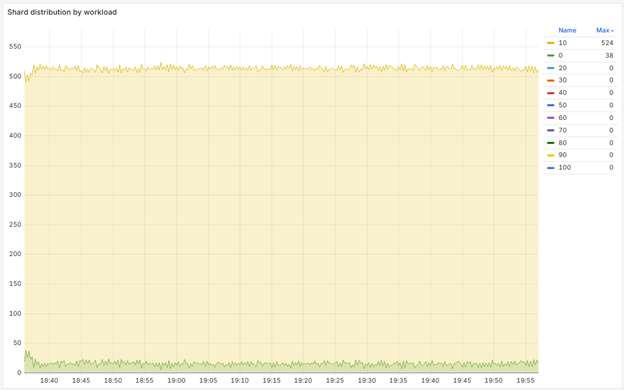


График распределения нагрузки между шардами.

<LLM\_ShardDistribution>

## Memory by host



График утилизации памяти на каждом хосте БД.

<LLM\_Memory>

## Disk data

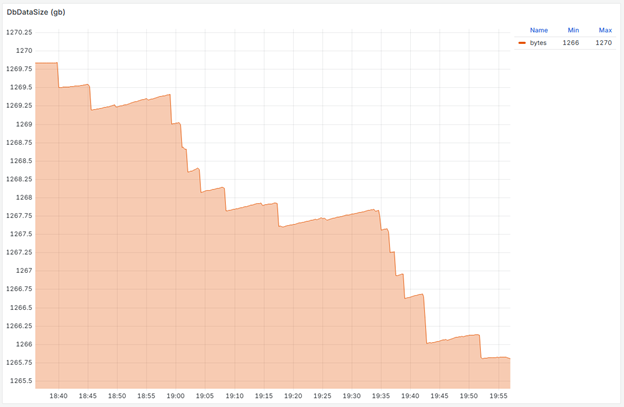


График показывает динамику использования дискового пространства БД.

<LLM\_Disk>