**Отчет о тестировании производительности проекта WP**

# 1. Сведения о проекте

В рамках проекта WP разработан web-сайт с функциями создания постов и комментариев к постам. Служебная информация, посты и комментарии хранятся в базе данных типа MariaDB.

В проекте реализовано две хранимые процедуры для административного добавления комментариев: dorepeat\_v1 и dorepeat\_v2 в базу данных, а также вспомогательные административные процедуры для очищения базы данных и эмуляции работы посетителей сайта.

# 2. Цели тестирования

1. Определение производительности процедур dorepeat\_v1 и dorepeat\_v2.
2. Определение нагрузочной способности проекта по количеству комментариев.
3. Выдача рекомендаций по реализации процедуры добавления комментариев.

# 3. Создание окружения

1. Тестовый стенд: ПК на процессоре 12th Gen Intel(R) Core(TM) i5-1245U, ОЗУ 32 Гб, SSD 250 Гб, OS Windows 10 Pro x64 22H2.
2. Используемое ПО: Jmeter v5.6.3, Docker Desktop v4.45.0, Chrome v139.0.7258.155, DBeaver v25.1.5.

# 4. Предварительные условия

1. Запущен проект WP из группы контейнеров в Docker.
2. Открыто приложение DBeaver, подключена база данных проекта MariaDB.
3. В базе данных содержится 1 пост и 5 комментариев к нему.
4. Открыто приложение Jmeter.
5. В браузере открыта страница проекта.

# 5. План тестирования

## 5.1. Определение производительности процедуры **dorepeat\_v1**

Для проверки производительности процедуры **dorepeat\_v1 используется приложение Jmeter, для которой создан тестовый сценарий с запуском dorepeat\_v1. После запуска сценария на указанное время 3 мин результаты измерения времени отклика базы данных в виде графика фиксируются для дальнейшего анализа.**

## 5.2. Определение производительности процедуры **dorepeat\_v2**

Для проверки производительности процедуры **dorepeat\_v1 используется приложение Jmeter, для которой создан тестовый сценарий с запуском dorepeat\_v2. После запуска сценария на указанное время 3 мин результаты измерения времени отклика базы данных в виде графика фиксируются для дальнейшего анализа.**

## 5.3. Определение нагрузочной способности проекта по количеству комментариев

**Определение нагрузочной способности проекта по количеству комментариев состоит из следующих проверок: определение времени отклика базы данных при наличии 1 комментария, определение времени отклика базы данных при наличии 1 комментария, определение нагрузочной способности проекта по количеству комментариев.**

**Время отклика базы при различном количестве комментариев фиксируется для последующего анализа.**

**Проверка определения нагрузочной способности по количеству комментариев выполняется путем запуска одновременно двух сценариев, имитирующих создание в базе данных 100 комментариев в 1 секунду и чтения комментариев 1 раз в секунду. С помощью Jmeter измеряется время отклика базы данных и время web запроса. Время загрузки страницы измеряется расширением браузера Page load time. При достижении времени обновления** страницы браузера 20 сек тест останавливается.

# 6. Протокол тестирования

Для определения целей тестирования выполнены следующие проверки.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Номер** | **Проверка** | **Шаг** | **Способ проверки** | **Ожидаемый результат** | **Результат испытаний** |
| 1 | Проверка производительности процедуры **dorepeat\_v1** | а | Открыть JMeter. Загрузить тестовый сценарий, файл wp\_db\_test.jmx | Jmeter запущен. Открыт тестовый сценарий wp\_db\_test | Успешно пройдено |
| б | Проверить настройки тестового сценария: Thread Group/writers/Number of Thread(users)=10 | Настройки соответствуют | Успешно пройдено |
| в | Запустить сценарий тестирования процедуры **dorepeat\_v1 на 3 мин.** | Сценарий запускается | Успешно пройдено |
| г | Сохранить результаты тестирования (скриншоты графиков) | Скриншоты сохранены | Успешно пройдено |
| д | Очистить базу данных, используя сценарий setup Thread Group | В базе данных содержится 1 пост и 5 комментариев | Успешно пройдено |
| 2 | Проверка производительности процедуры **dorepeat\_v2** | а | Открыть JMeter. Загрузить тестовый сценарий, файл wp\_db\_test.jmx | Jmeter запущен. Открыт тестовый сценарий wp\_db\_test | Успешно пройдено |
| б | Проверить настройки тестового сценария: Thread Group/writers/Number of Thread(users)=10 | Настройки соответствуют | Успешно пройдено |
| в | Запустить сценарий тестирования процедуры **dorepeat\_v2 на 3 мин.** | Сценарий запускается | Успешно пройдено |
| г | Сохранить результаты тестирования (скриншоты графиков) | Скриншоты сохранены | Успешно пройдено |
| д | Очистить базу данных, используя сценарий setup Thread Group | В базе данных содержится 1 пост и 5 комментариев | Успешно пройдено |
| 3 | Определение времени отклика базы данных при наличии 1 комментария |  | Открыть JMeter. Загрузить тестовый сценарий, файл wp\_db\_test.jmx | Jmeter запущен. Открыт тестовый сценарий wp\_db\_test | Успешно пройдено |
|  | Очистить базу данных, используя сценарий setup Thread Group/delete | В базе данных содержится 1 пост и 0 комментариев | Успешно пройдено |
|  | Добавить 1 комментари1 к посту через web интерфейс проекта | В базе данных содержится 1 пост и 1 комментарий | Успешно пройдено |
|  | Проверить настройки тестового сценария: Thread Group/readers/Number of Thread(users)=1 | Настройки соответствуют | Успешно пройдено |
|  | Запустить сценарий тестирования времени отклика Thread Group/readers **на 2 мин.** | Сценарий запускается | Успешно пройдено |
| г | Сохранить результаты тестирования (скриншоты графиков) | Скриншоты сохранены | Успешно пройдено |
| 4 | Определение времени отклика базы данных при наличии 1000000 комментария |  | Открыть JMeter. Загрузить тестовый сценарий, файл wp\_db\_test.jmx | Jmeter запущен. Открыт тестовый сценарий wp\_db\_test | Успешно пройдено |
|  | Проверить настройки тестового сценария: Thread Group /writers/ Number of Thread(users)=10, setUp Thread Group/ Loop Count=10 000 | Настройки соответствуют | Успешно пройдено |
|  | Подготовить базу данных (внести 1 млн комментариев), используя сценарий writers/v1 | В базе данных содержится 1 пост и 1000 005 комментариев | Успешно пройдено |
|  | Запустить сценарий тестирования времени отклика Thread Group/readers **на 2 мин.** | Сценарий запускается | Успешно пройдено |
|  | Сохранить результаты тестирования (скриншоты графиков) | Скриншоты сохранены | Успешно пройдено |
|  | Проверить настройки тестового сценария: Thread Group /setUp Thread Group/ Number of Thread(users)=1, setUp Thread Group/ Loop Count=1 | Настройки соответствуют | Успешно пройдено |
|  | Очистить базу данных, используя сценарий setup Thread Group | В базе данных содержится 1 пост и 5 комментариев | Успешно пройдено |
| 5 | Определение нагрузочной способности проекта по количеству комментариев |  | Открыть JMeter. Загрузить тестовый сценарий, файл wp\_db\_test.jmx | Jmeter запущен. Открыт тестовый сценарий wp\_db\_test | Успешно пройдено |
|  | Проверить настройки тестового сценария: Thread Group /writers/ Number of Thread(users)=1, Loop Count=infinite;  Thread Group /writers/v1/Query=call dorepeat\_v1(100); | Настройки соответствуют | Успешно пройдено |
|  | Запустить одновременно тестовые сценарии v1, Test\_web | Сценарии запускаются | Успешно пройдено |
|  | Контролировать 1 раз в сек время обновления страницы браузера на странице проекта с помощью расширения Page load time. При достижении времени обновления 20 сек остановить выполнения теста в Jmeter | Заданное время обновления достигнуто | Успешно пройдено |
|  | Зафиксировать время загрузки страницы проекта WP в браузере (скриншот) | Скриншоты сохранены | Успешно пройдено |
|  | Сохранить результаты тестирования (скриншоты графиков) Jmeter | Скриншоты сохранены | Успешно пройдено |
|  | Зафиксировать состояние базы данных (количество комментариев) в Dbeaver | Скриншоты сохранены | Успешно пройдено |
|  | Проверить настройки тестового сценария: Thread Group /setUp Thread Group/ Number of Thread(users)=1, setUp Thread Group/ Loop Count=1 | Настройки соответствуют | Успешно пройдено |
|  | Очистить базу данных, используя сценарий setup Thread Group | В базе данных содержится 1 пост и 5 комментариев | Успешно пройдено |

# 7. Результаты тестирования

7.1 Результаты выполнения процедуры dorepeat\_v1

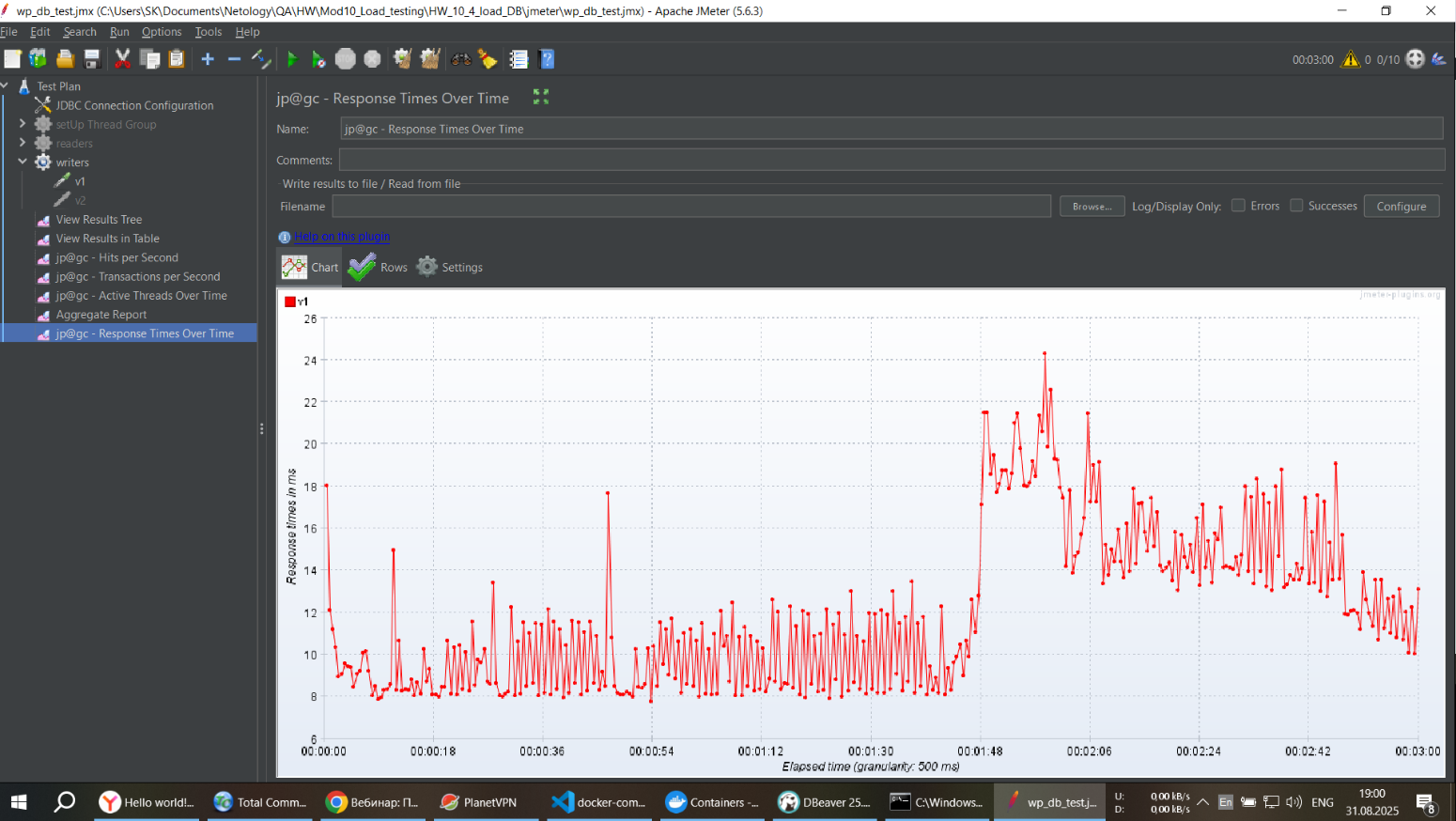


Рисунок 1 – Время отклика базы данных при работе dorepeat\_v1

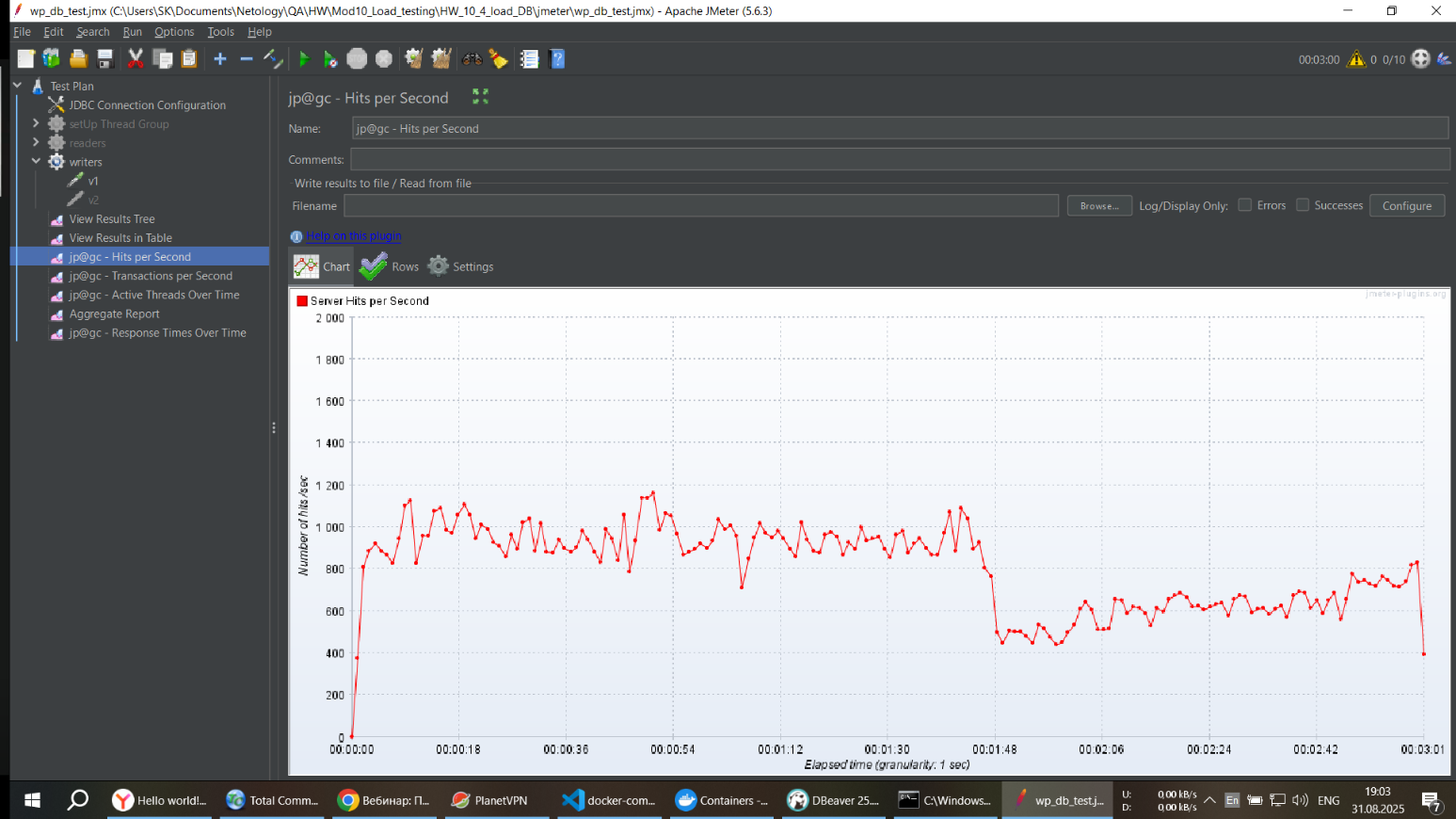


Рисунок 2 – Количество транзакций в секунду базы данных при работе dorepeat\_v1

7.2 Результаты выполнения процедуры dorepeat\_v2



Рисунок 3 – Время отклика базы данных при работе dorepeat\_v2

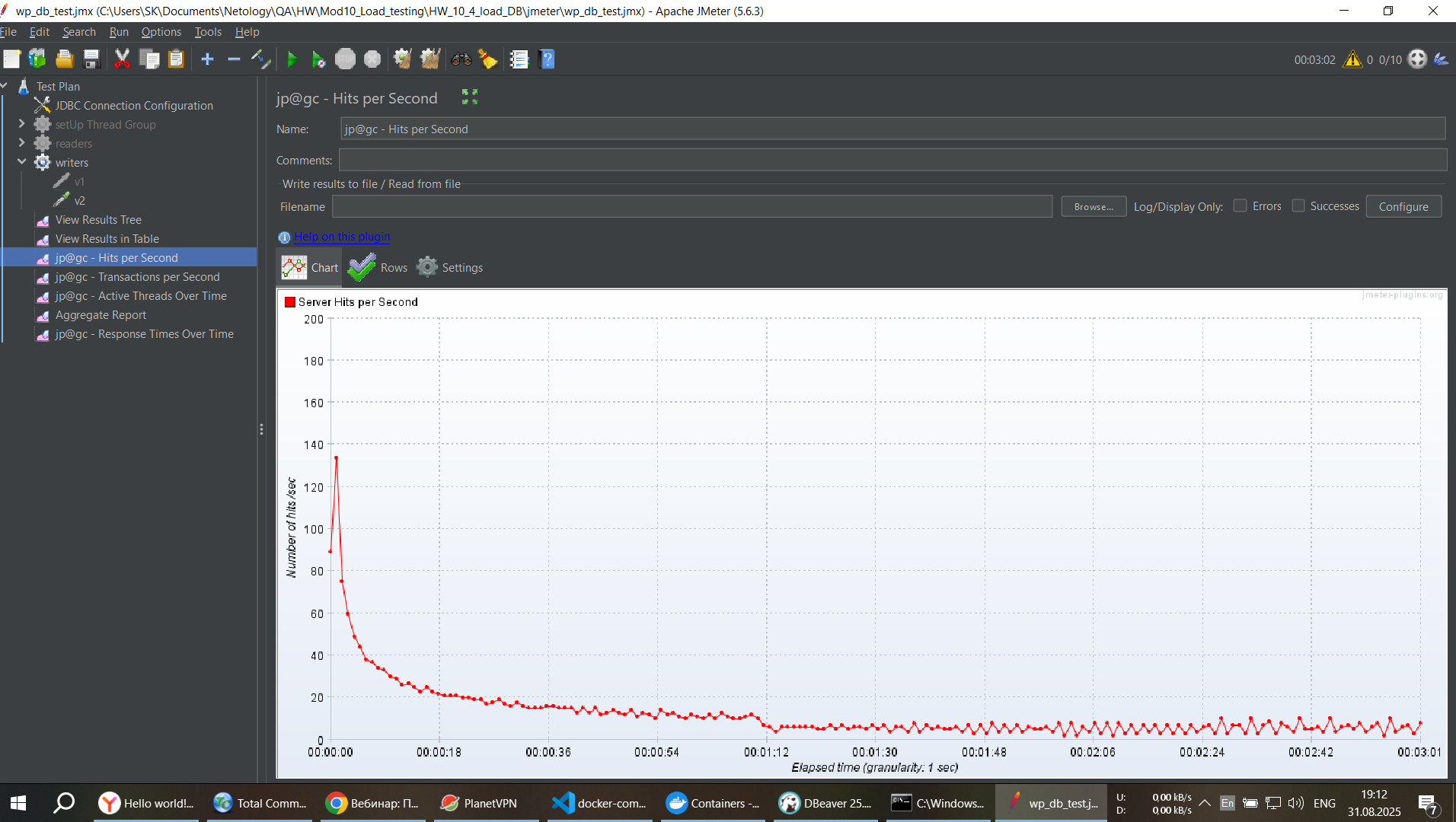


Рисунок 4 – Количество транзакций в секунду базы данных при работе dorepeat\_v2

7.3 Определение нагрузочной способности проекта по количеству комментариев

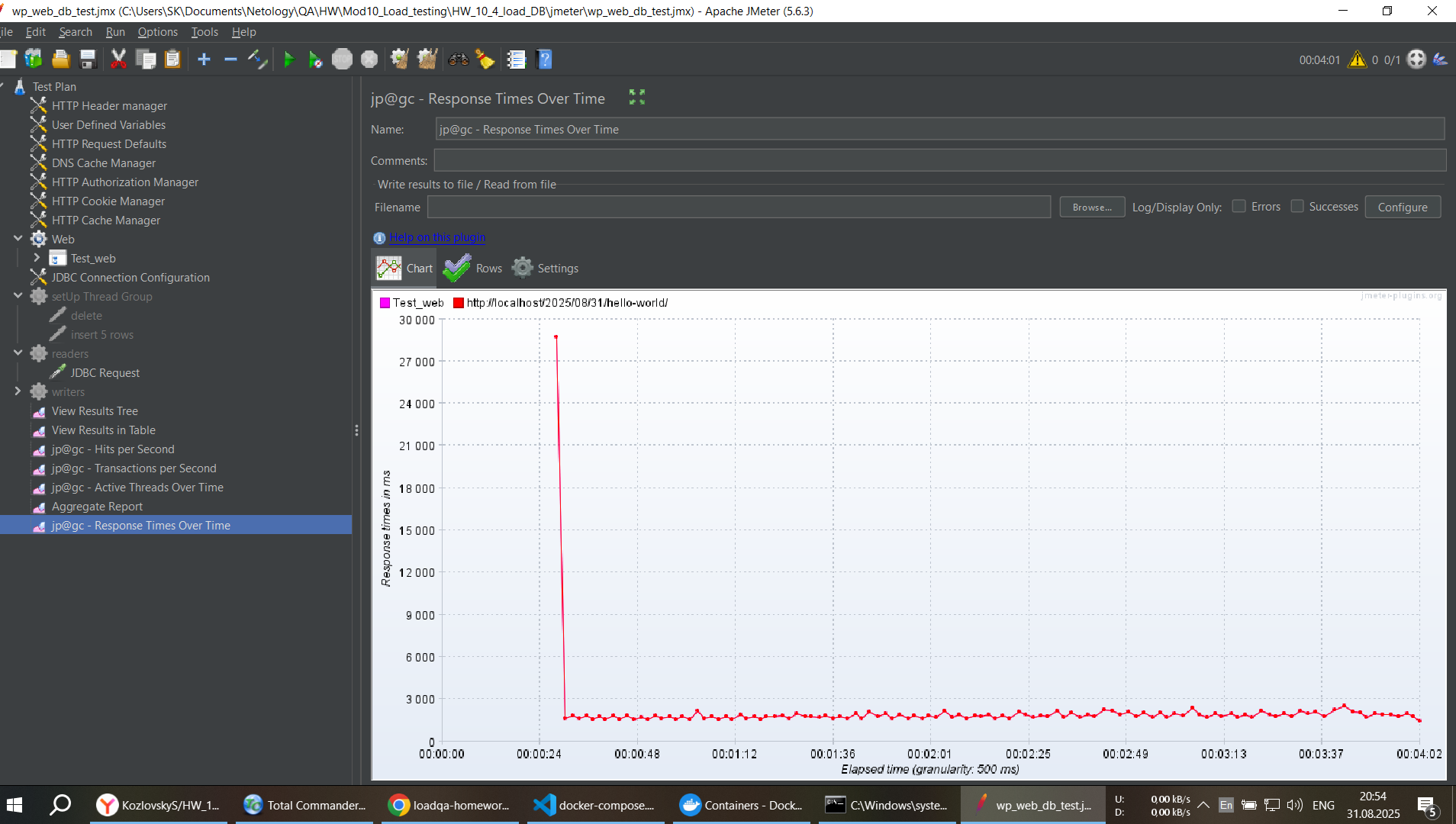


Рисунок 5 – Время отклика web запроса при количестве комментариев равно 1 000 005

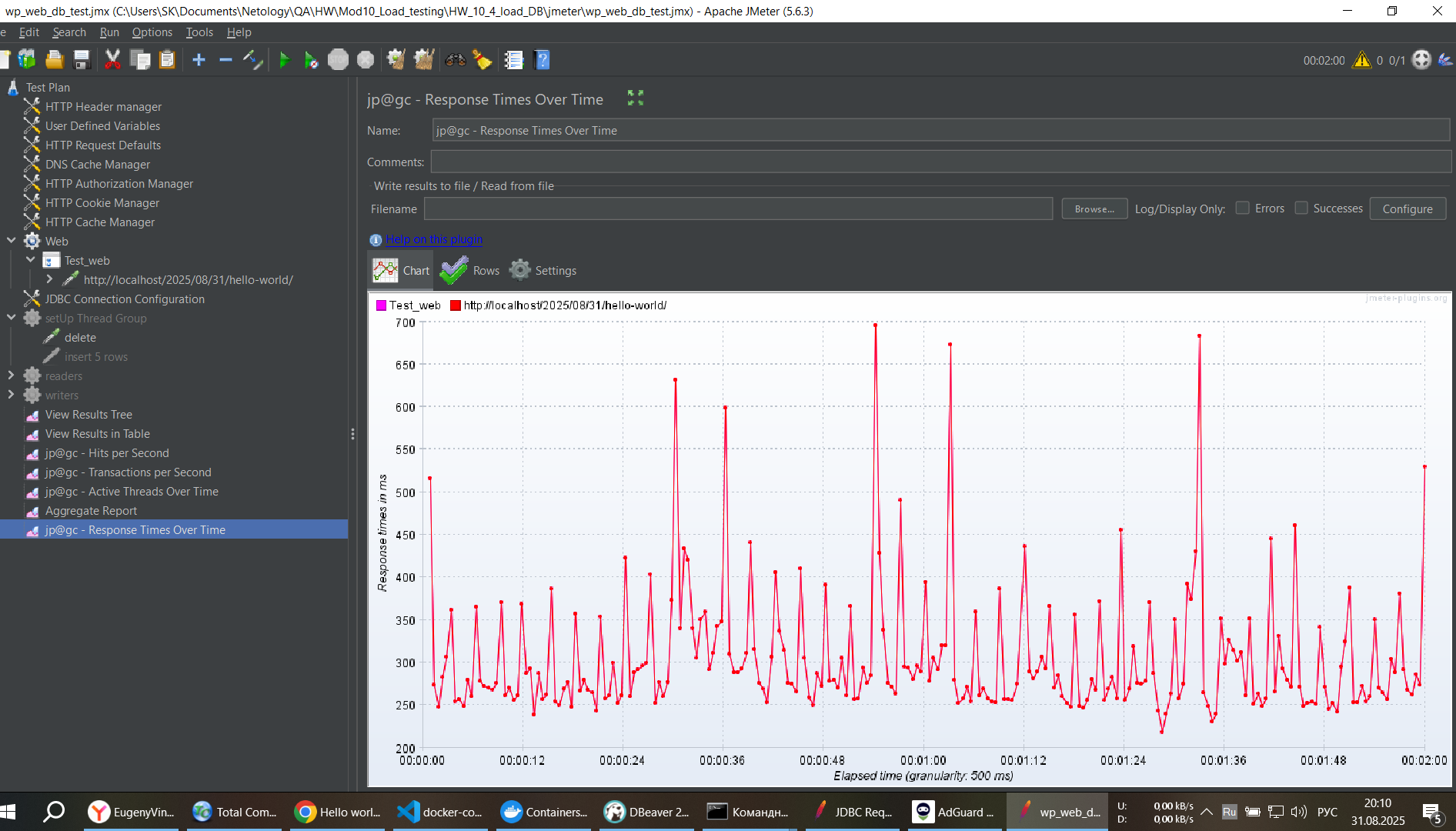


Рисунок 6 – Время отклика web запроса при количестве комментариев равно 1

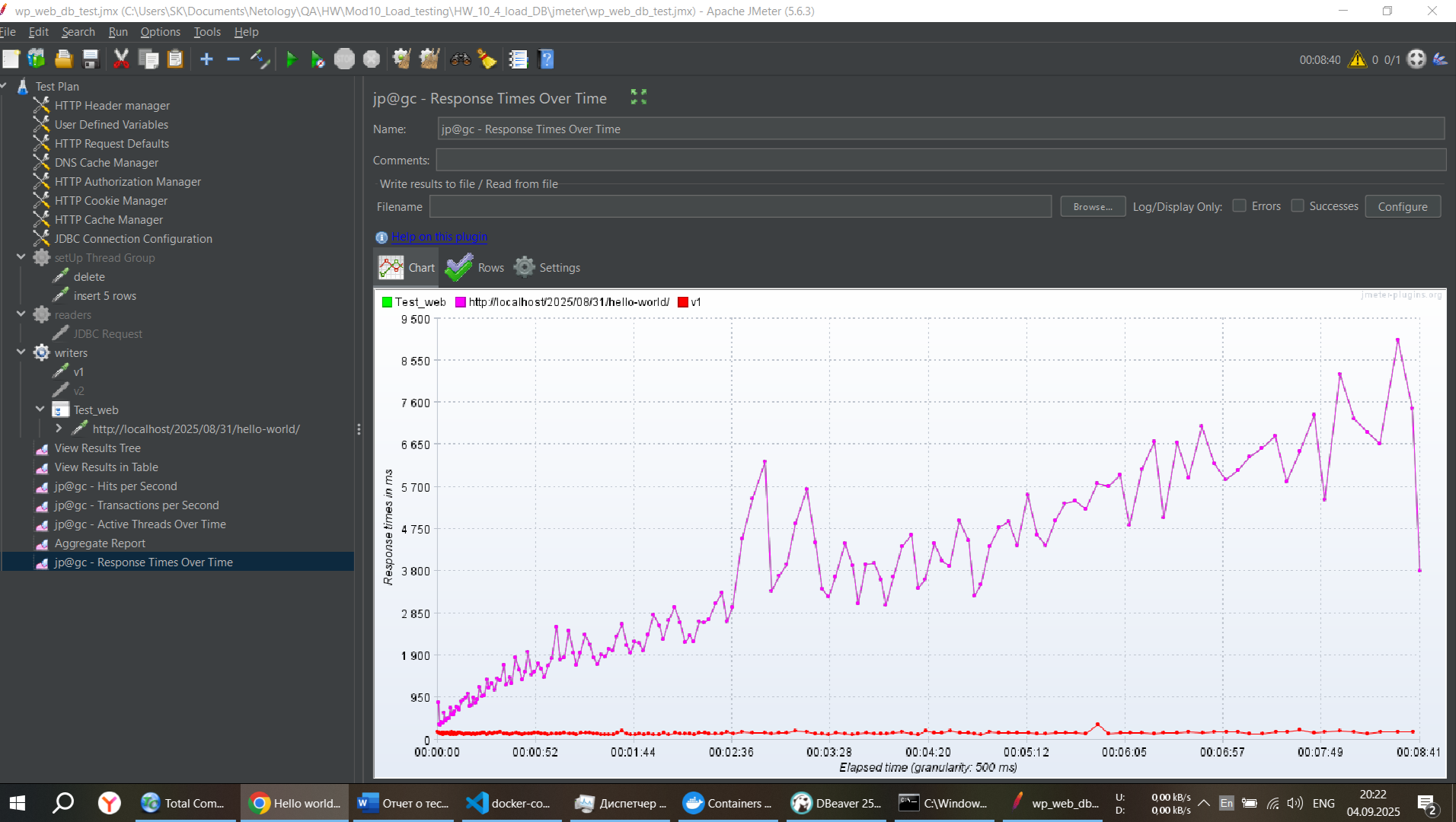


Рисунок 7 – Время отклика web запросов при добавлении 100 комментариев в секунду

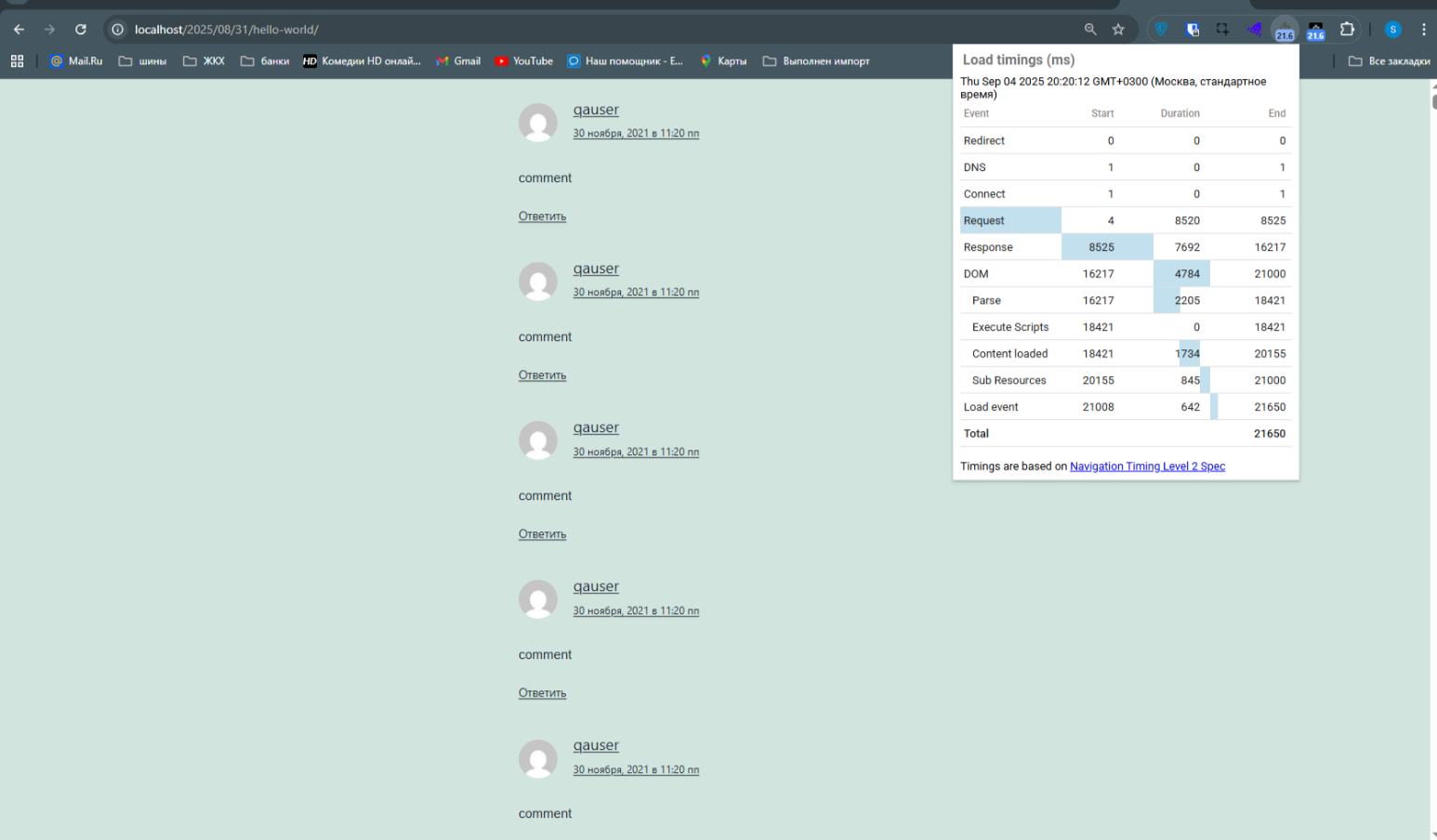


Рисунок 8 – Определение времени обновления страницы браузера

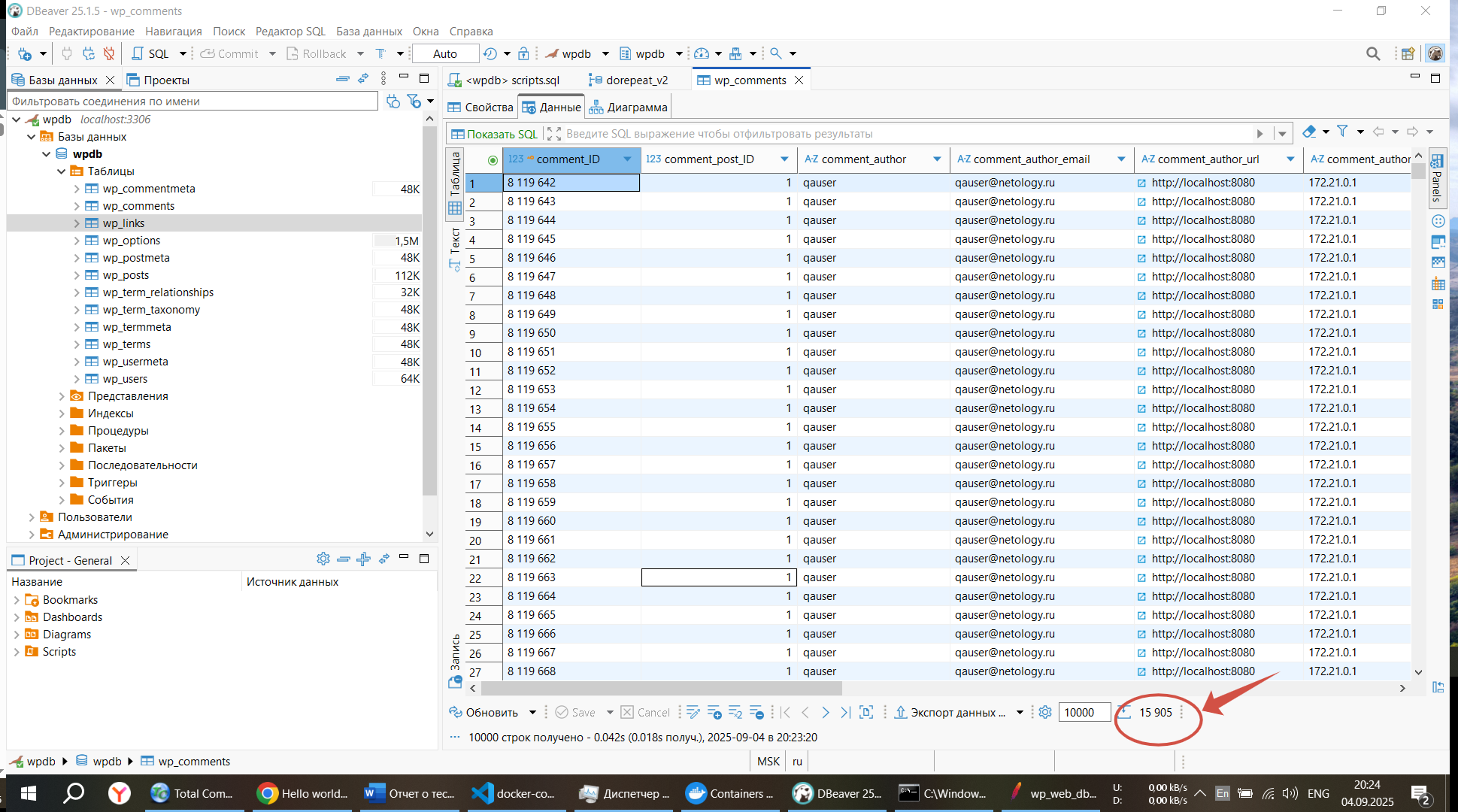


Рисунок 9 – Состояние базы данных проекта при достижении времени обновления страницы более 20 сек

# 8. Выводы

8.1. **Время отклика БД в процессе выполнения** процедуры **dorepeat\_v1 находится в пределах в пределах 15мс (за исключением нескольких пиков),и мало зависит от количества комментариев.**

**Количество транзакций в секунду при выполнении** процедуры **dorepeat\_v1 находится в пределах 800-100.**

**8.2. Время отклика БД в процессе выполнения** процедуры **dorepeat\_v2 изначально больше, чем в dorepeat\_v2, имеет выраженную зависимость от количества комментариев, при 20000 комментариев достигает значения около 2 сек.**

**Количество транзакций в секунду при выполнении** процедуры **dorepeat\_v2 уменьшается с 80 до 10 при наполнении базы данных до 20000 комментариев.**

**8.3.** **П**роцедура **dorepeat\_v1 является более производительной, чем процедура dorepeat\_v2.**

**8.4. Рекомендуется в проекте использовать dorepeat\_v2, как более нетребовательную к количеству записей в БД и превосходящую по быстродействию.**

**8.5 Минимальное время отклика web запроса (при 1 комментарии) составляет около 300 мс, увеличиваясь до 1,5 сек при 1 млн записей комментариев.**

**8.6 При одновременном добавлении 100 комментариев за 1 сек, время обновления страницы браузера достигает 20 сек при количестве комментариев около 15000, при этом время отклика web запроса составляет около 8 сек.**