**Практическое занятие № 50**

**Тема:** Нагрузочное тестирование

**Теоретические вопросы**

1. **Понятие нагрузочного тестирования, приложения, виртуального пользователя, итерации, нагрузки, производительности, масштабируемости приложения, профиля нагрузки, нагрузочной точки, теста производительности.**

Нагрузочное тестирование – это процесс оценки производительности системы или приложения путем увеличения нагрузки на него до точки, когда система перестает справляться с запросами или начинает демонстрировать нежелательное поведение. Целью такого тестирования является выявление проблем, связанных с производительностью, масштабируемостью и устойчивостью системы при реальных условиях использования.

Приложение (или прикладное программное обеспечение) – это программное обеспечение, разработанное для решения конкретной задачи или предоставления определенной функциональности конечным пользователям.

Виртуальный пользователь – это программное или аппаратное устройство, которое моделирует поведение реального пользователя в системе или приложении. Во время нагрузочного тестирования создаются множество виртуальных пользователей, чтобы оценить, как система реагирует на одновременные запросы от большого числа пользователей.

Итерация – это повторяющийся процесс или цикл разработки программного продукта, в котором выполняются определенные этапы работы над проектом с последующим анализом и корректировками.

Нагрузка – это количество запросов, генерируемых в систему за определенный период времени. Нагрузка может быть изменена во время тестирования, чтобы проверить, как система реагирует на разные уровни нагрузки.

Производительность системы – это способность системы эффективно обрабатывать запросы пользователей при заданных условиях нагрузки. Нагрузочное тестирование помогает оценить производительность системы и выявить ее ограничения.

Масштабируемость приложения относится к его способности эффективно масштабироваться (увеличивать или уменьшать ресурсы) для обработки большого количества запросов при изменяющейся нагрузке.

Профиль нагрузки – это описание типичного поведения пользователей системы или приложения, включая частоту запросов, тип запросов, длительность сеансов и т.д. Понимание профиля нагрузки помогает определить параметры для нагрузочного тестирования.

Нагрузочная точка – это уровень нагрузки, при котором система начинает проявлять признаки неэффективной работы или неспособности обрабатывать все поступающие запросы.

Тест производительности – это специфический вид нагрузочного тестирования, целью которого является измерение и оценка производительности системы в различных условиях нагрузки, например, при различных уровнях нагрузки или при изменении конфигурации системы.

1. **Цели нагрузочного тестирования.**

Нагрузочное тестирование имеет несколько целей, включая:

* определение производительности системы: одна из основных целей нагрузочного тестирования – это определение производительности системы. Тестирование позволяет оценить, как система реагирует на увеличение нагрузки, такое как количество пользователей или объем данных, и выявить ее ограничения и узкие места;
* оценка масштабируемости приложения: нагрузочное тестирование помогает оценить масштабируемость приложения, то есть его способность эффективно масштабироваться для обработки большого количества запросов или увеличения объема данных. Это позволяет оптимизировать архитектуру приложения и инфраструктуру для обеспечения роста и расширения;
* выявление узких мест и проблем производительности: нагрузочное тестирование позволяет выявить узкие места и проблемы производительности системы, такие как медленные запросы к базе данных, неэффективные алгоритмы обработки данных или недостаточные ресурсы сервера. Это помогает в идентификации проблем и их устранении до того, как они повлияют на пользователей в реальной среде;
* проверка стабильности системы под нагрузкой: нагрузочное тестирование также помогает проверить стабильность системы под нагрузкой. Это включает в себя оценку, насколько система способна поддерживать высокие нагрузки без сбоев, потери данных или нежелательного поведения;
* проверка соответствия требованиям производительности: нагрузочное тестирование также используется для проверки того, соответствует ли производительность системы требованиям, установленным для проекта. Это помогает гарантировать, что система работает в соответствии с ожиданиями заказчика и пользователя.

1. **Виды нагрузочного теста.**

Существует несколько видов нагрузочного тестирования, каждый из которых направлен на проверку определенных аспектов производительности и стабильности системы. Некоторые из основных видов нагрузочного тестирования включают:

* тест на нагрузку: в этом тесте системе подается максимально возможная нагрузка, чтобы проверить ее поведение и производительность при пиковых нагрузках. Целью является определение максимальной границы пропускной способности системы и выявление ее ограничений;
* тест на стресс: тест на стресс проводится для проверки стабильности системы под экстремальными условиями нагрузки или внешними воздействиями. В этом тесте система подвергается экстремальным нагрузкам или нештатным ситуациям, чтобы проверить, как она реагирует и восстанавливается после подобных ситуаций;
* тест на объем: этот вид тестирования направлен на оценку производительности системы при обработке больших объемов данных. Целью является определение уровня производительности и эффективности приложения при работе с большими объемами информации;
* тест на длительность нагрузки: в этом тесте системе подается постоянная нагрузка в течение длительного времени, чтобы проверить ее устойчивость и стабильность при продолжительной работе под нагрузкой. Этот тест также может помочь выявить утечки ресурсов или другие проблемы, возникающие со временем;
* тест на производительность: этот вид тестирования направлен на оценку производительности системы при определенных условиях нагрузки или сценариях использования. Целью является измерение времени ответа системы, пропускной способности и других параметров производительности в реальном времени;
* тест на масштабируемость: в этом тесте проверяется способность системы масштабироваться для обработки увеличенного числа пользователей или объема данных. Целью является определение предельного уровня масштабируемости системы и ее способности поддерживать рост пользовательской базы.

1. **Утилиты нагрузочного тестирования, их характеристики, достоинства и недостатки (не менее 10 примеров утилит).**

Существует множество утилит для нагрузочного тестирования, каждая из которых имеет свои особенности, достоинства и недостатки. Вот несколько примеров таких утилит:

1. Apache JMeter:

* характеристики: мощный инструмент с графическим интерфейсом для создания и выполнения различных видов нагрузочных тестов;
* достоинства: бесплатный, поддерживает разнообразные протоколы (HTTP, FTP, JDBC и др.), легко масштабируемый;
* недостатки: не всегда интуитивный интерфейс, требует времени для изучения.

1. Gatling:

* характеристики: основанный на Scala инструмент сценариев для тестирования производительности и нагрузки;
* достоинства: легко масштабируется, поддерживает кодирование сценариев на языке Scala, имеет отчеты в реальном времени;
* недостатки: требует знаний Scala для создания сложных сценариев, немногословен в использовании.

1. LoadRunner:

* характеристики: популярный коммерческий инструмент для нагрузочного тестирования от Micro Focus;
* достоинства: мощный и гибкий, поддерживает разнообразные протоколы, имеет обширные функциональные возможности;
* недостатки: высокая стоимость лицензии, тяжеловесный и требует значительных ресурсов.

1. Locust:

* характеристики: открытый и расширяемый инструмент для написания тестов на Python;
* достоинства: прост в использовании, легко масштабируется, имеет удобный интерфейс для создания сценариев;
* недостатки: ограниченная поддержка протоколов, меньше функциональных возможностей по сравнению с другими утилитами.

1. Tsung:

* характеристики: открытый и масштабируемый инструмент для нагрузочного тестирования, написанный на Erlang;
* достоинства: высокая производительность, поддерживает множество протоколов (HTTP, WebSockets, XMPP и др.), легко масштабируется;
* недостатки: сложность настройки, требует знания Erlang.

1. BlazeMeter:

* характеристики: облачный сервис для проведения нагрузочного тестирования, основанный на Apache JMeter;
* достоинства: легкость использования, масштабируемость, предоставляет аналитику и отчеты;
* недостатки: платный сервис, ограниченные возможности настройки.

1. Artillery:

* характеристики: инструмент для тестирования производительности и нагрузки, написанный на Node.js;
* достоинства: прост в использовании, поддерживает тестирование в реальном времени, легко расширяемый;
* недостатки: менее мощный по сравнению с некоторыми другими утилитами.

1. Apache Bench (ab):

* характеристики: утилита командной строки для тестирования HTTP-серверов;
* достоинства: прост в использовании, быстрый запуск тестов, доступен по умолчанию во многих системах;
* недостатки: ограниченный набор функций, не подходит для сложных сценариев.

1. Siege:

* характеристики: утилита командной строки для тестирования производительности веб-серверов;
* достоинства: прост в использовании, поддерживает тестирование нескольких URL одновременно, настраиваемая нагрузка;
* недостатки: ограниченные возможности аналитики и отчетности.

1. Neoload:

* характеристики: коммерческий инструмент для нагрузочного тестирования, предоставляющий различные возможности тестирования;
* достоинства: мощные аналитические инструменты, широкий спектр поддерживаемых технологий и протоколов;
* недостатки: высокая стоимость лицензии, сложность в использовании для новичков.

**Задание 1.**

Произвести тестирование 3 различных веб-сайтов со следующим планом тестирования: количество потоков – 100; количество циклов – 10; период наращивания – 100. В отчёт поместить скрины отчётов тестирования и сделать выводы.

Для нагрузочного тестирования с помощью утилиты Gatling мы выбрали сайты: <https://boilk.ru/>, <https://about.gitlab.com/>, <https://ru.pinterest.com/>.

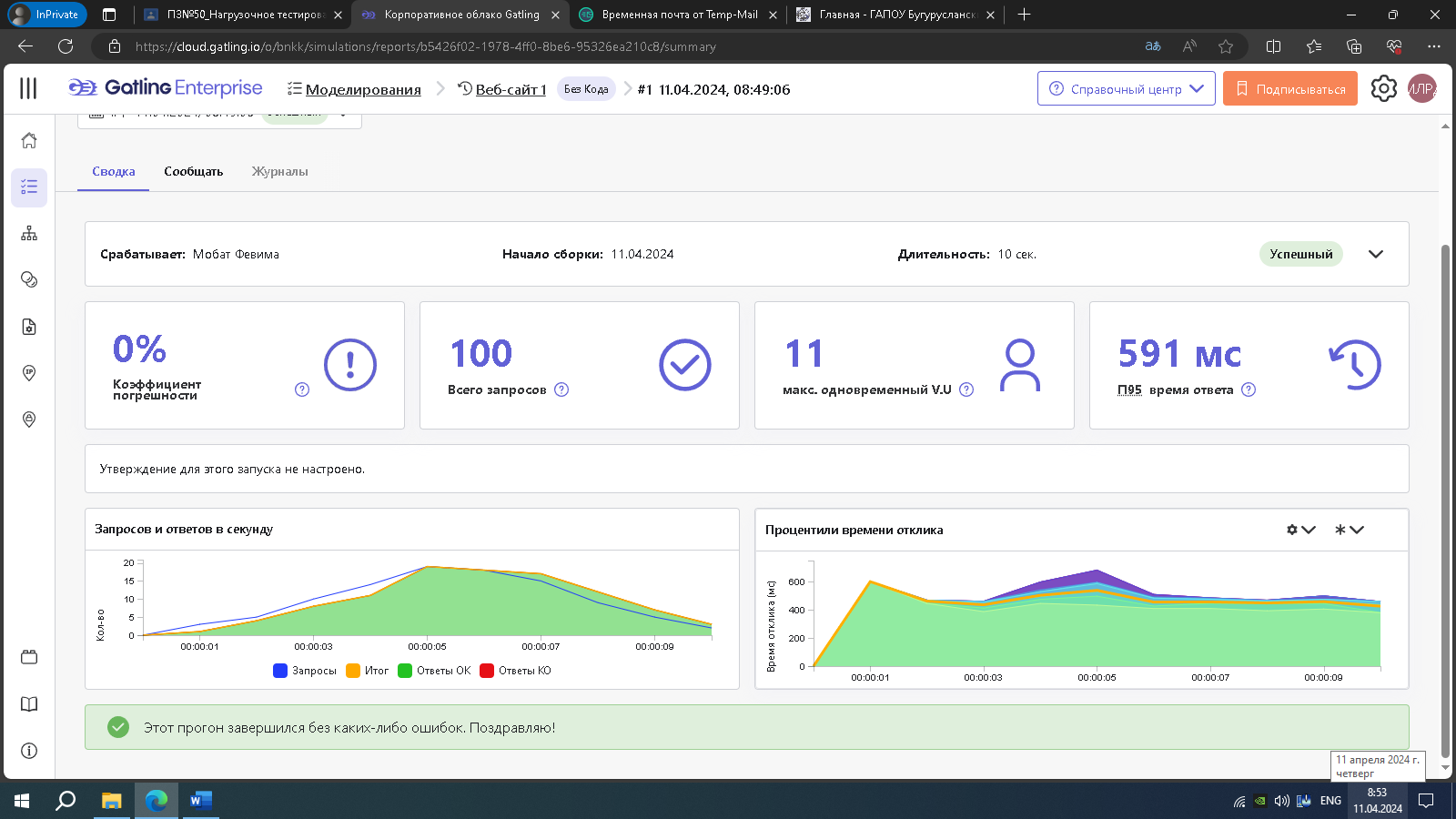


Рисунок 1 – Результат нагрузочного тестирования веб-сайта БНК

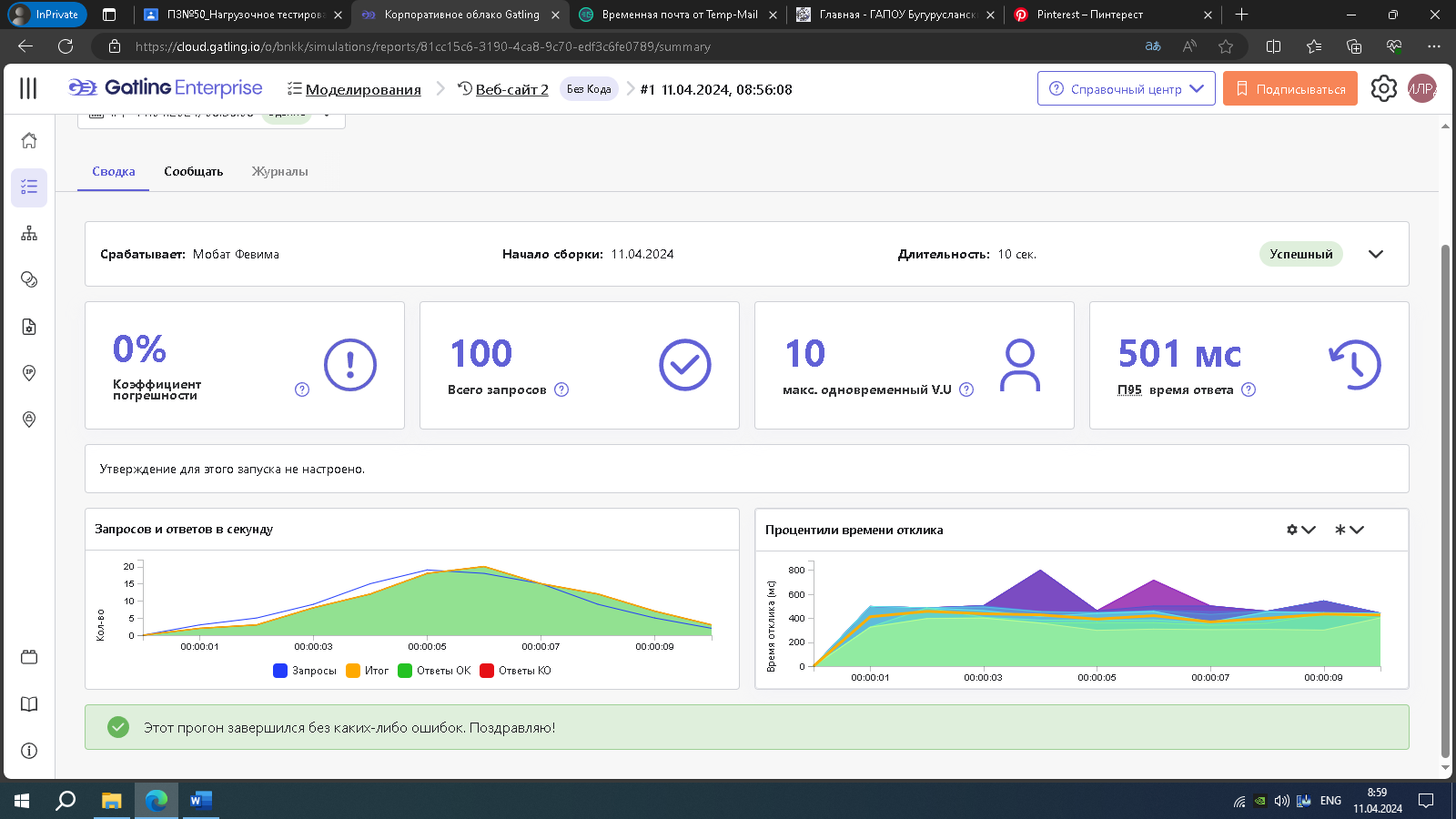


Рисунок 2 – Результат нагрузочного тестирования веб-сайта Pinterest

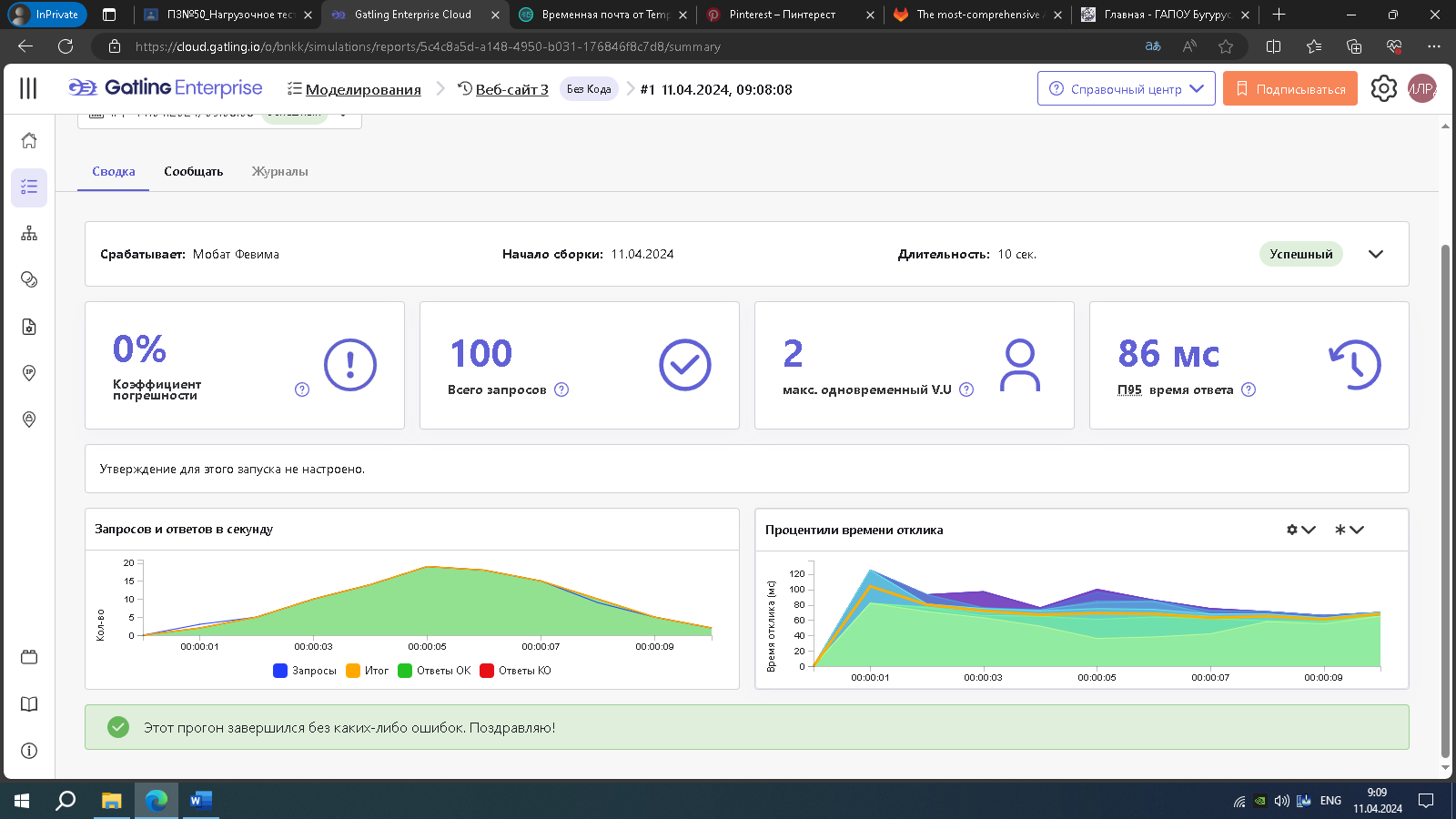


Рисунок 3 – Результат нагрузочного тестирования веб-сайта GitLab

Теперь проведем нагрузочное тестирование с помощью JMeter следующих сайтов: <https://boilk.ru/>, <https://about.gitlab.com/>, <https://ru.pinterest.com/>.

Для начала тестирование необходимо выполнить следующие действия, которые изображены на рисунках 4-8:

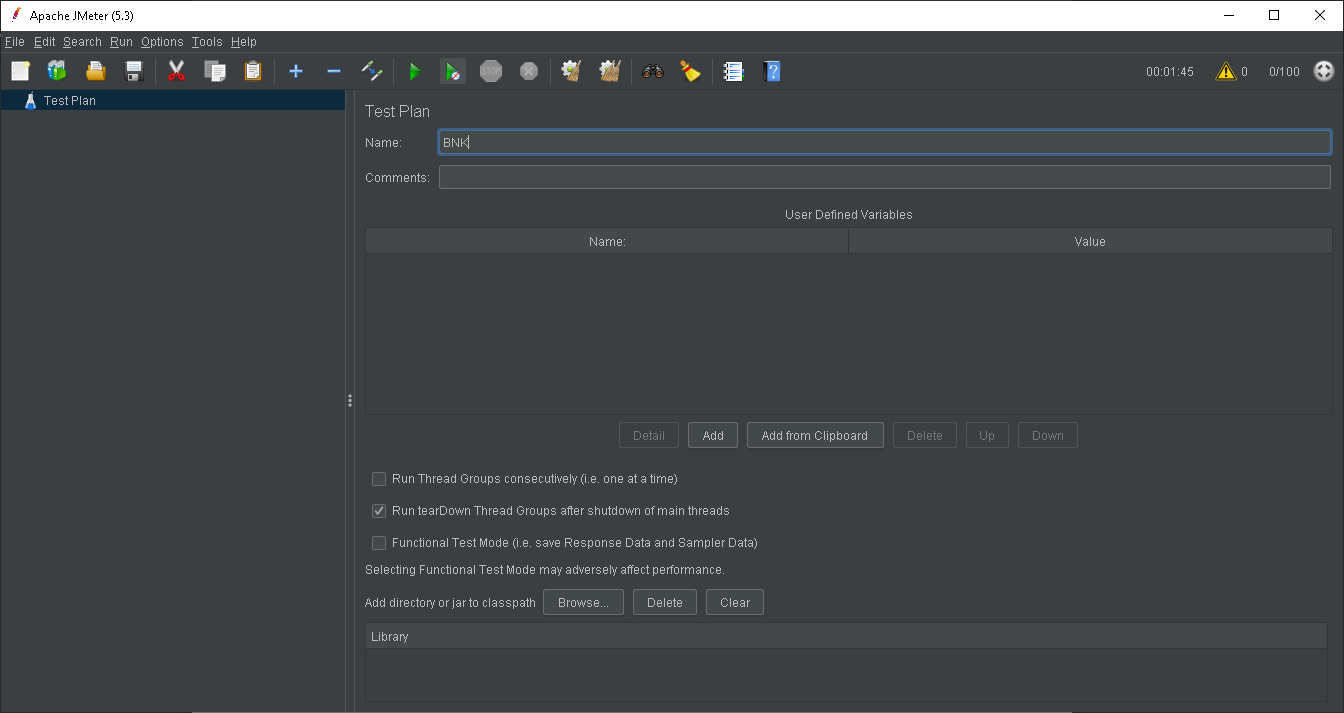


Рисунок 4 – Создание теста

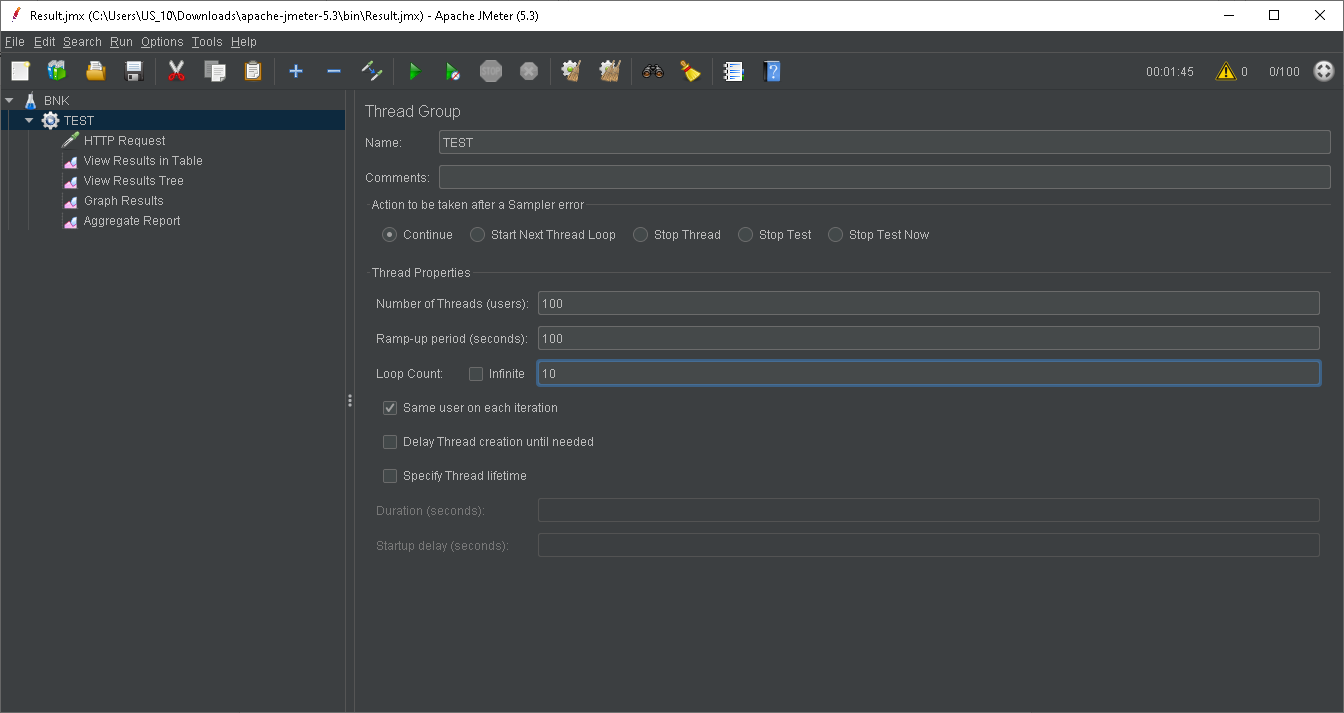


Рисунок 5 – Создаем группу, вводим количество пользователей, циклов и время продолжительности

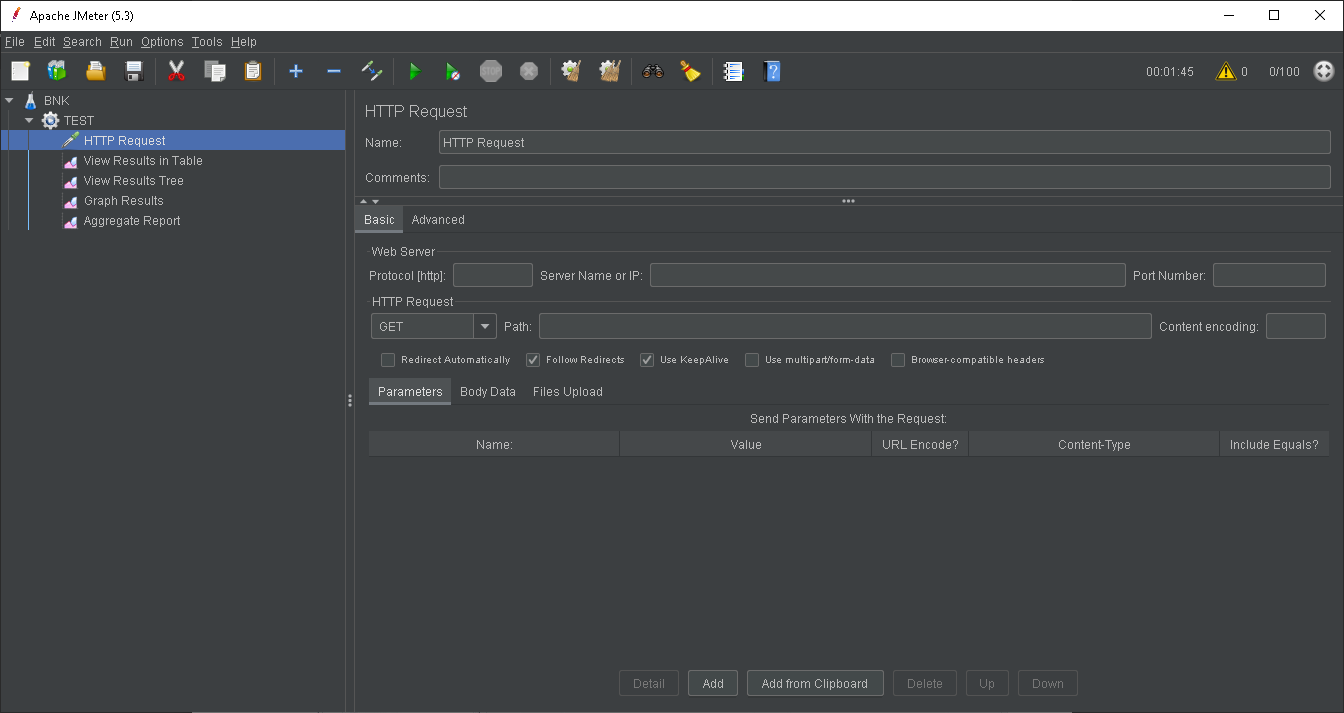


Рисунок 6 – Добавляем средства отображения результата

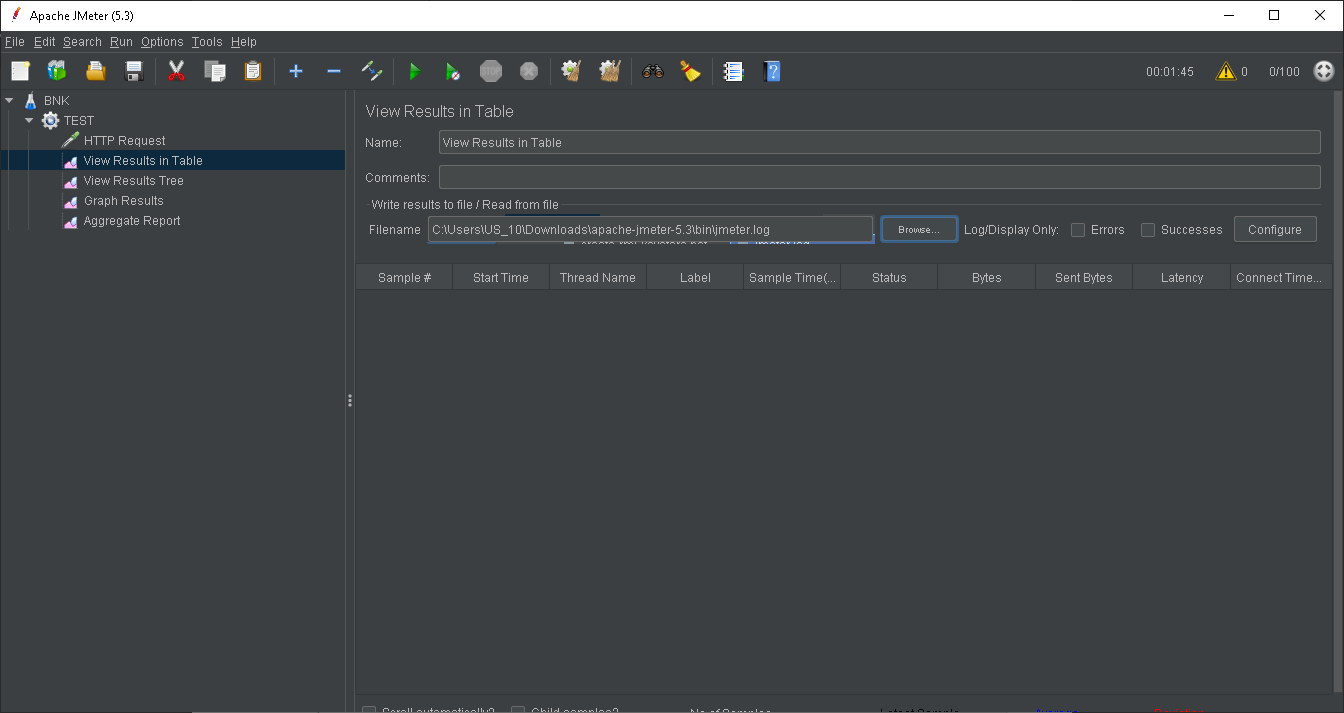


Рисунок 7 – Добавление место для сохранения результата

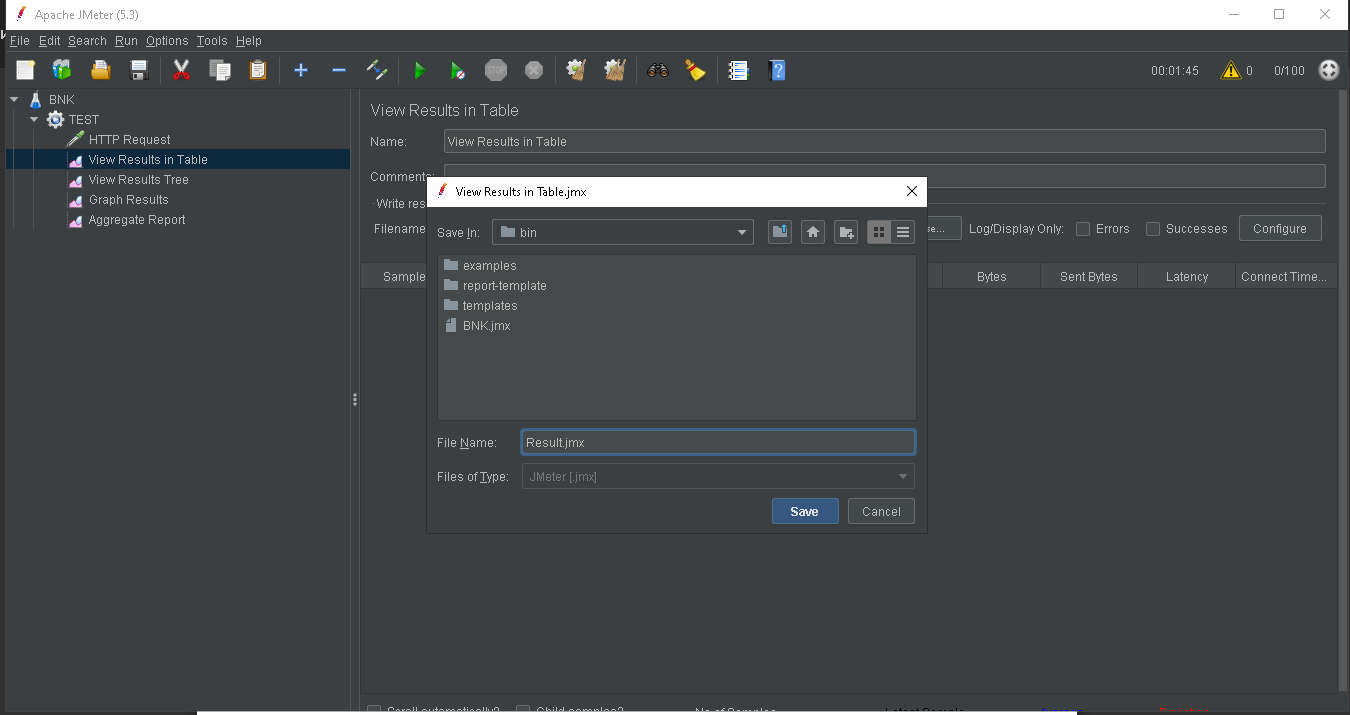


Рисунок 8 – Сохраняем файл

Тестирование сайта БНК показано на рисунках 9-12:

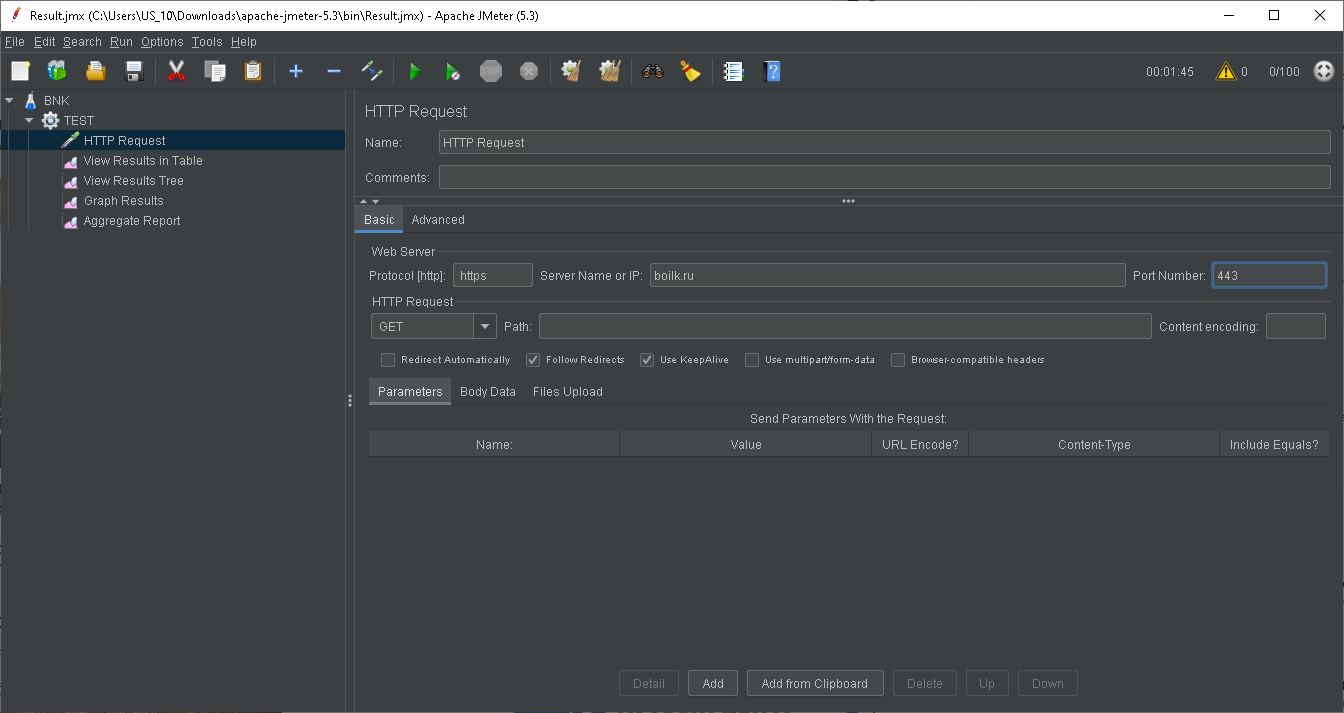


Рисунок 9 – Вводим данные сайта

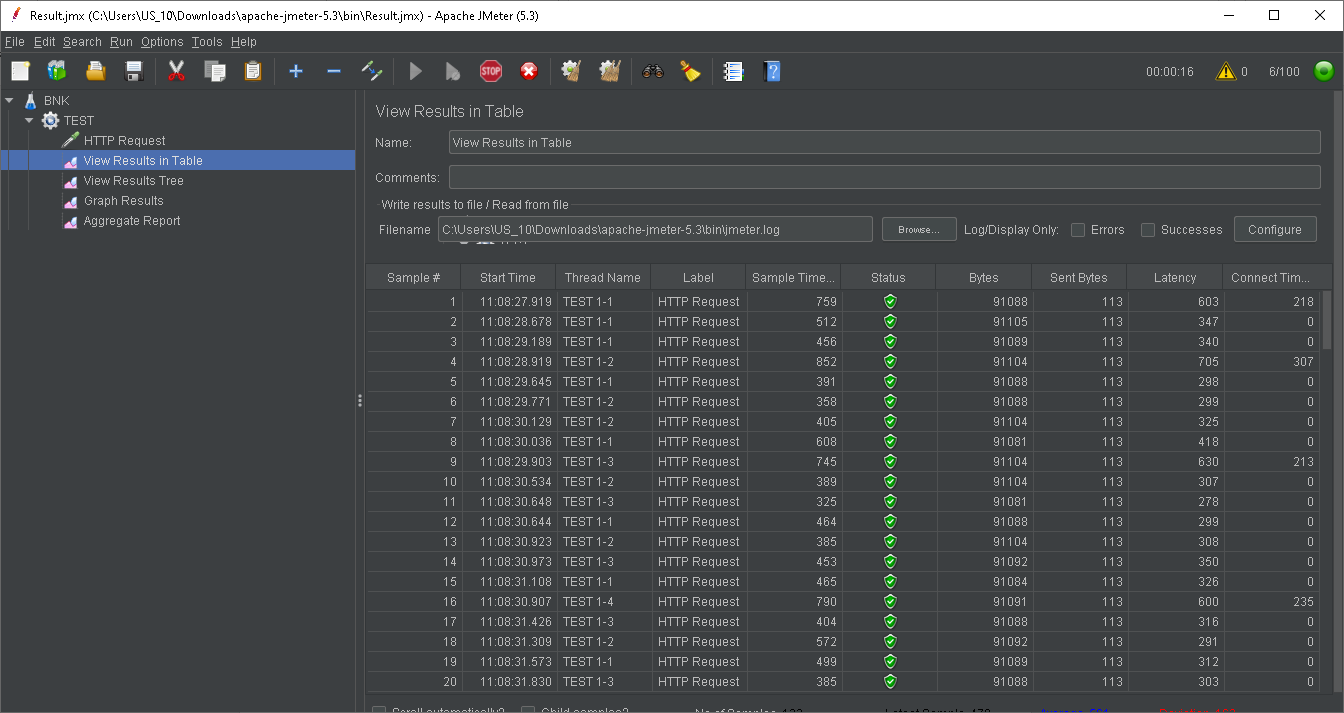


Рисунок 10 – Результат тестирования в виде таблицы

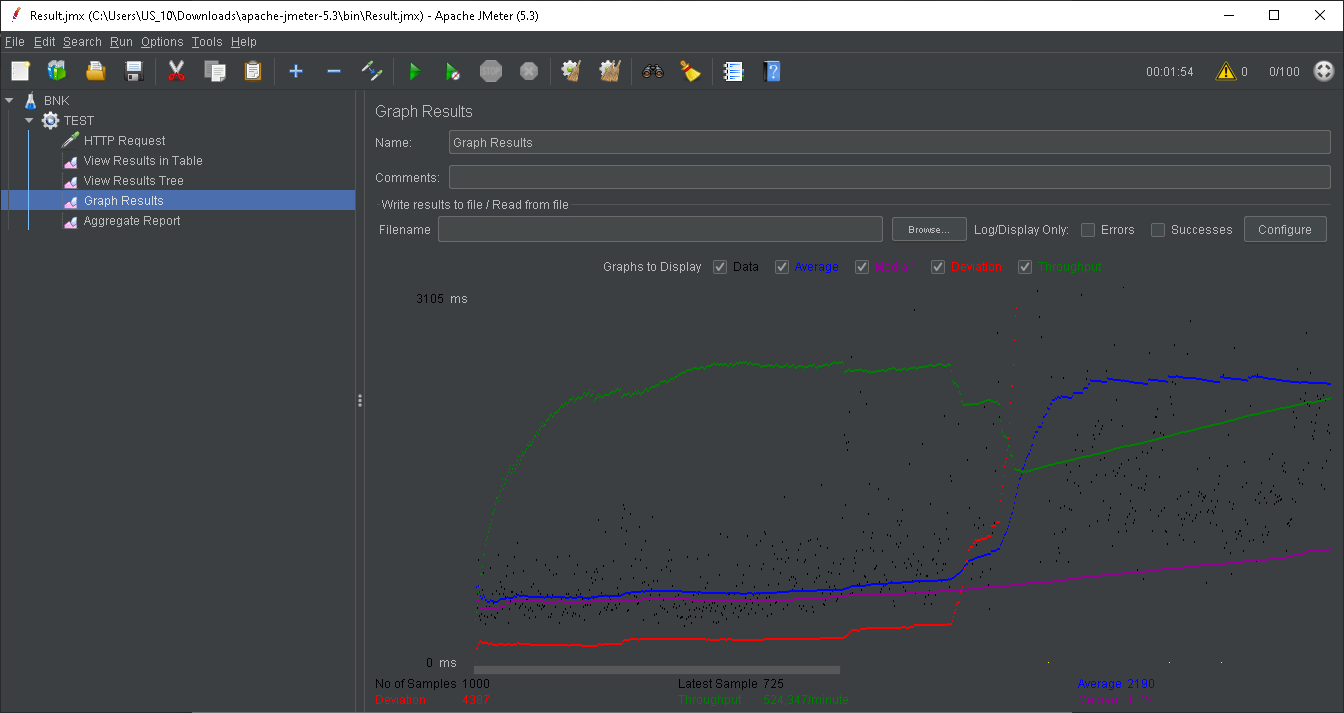


Рисунок 11 – Результат тестирования в виде графика

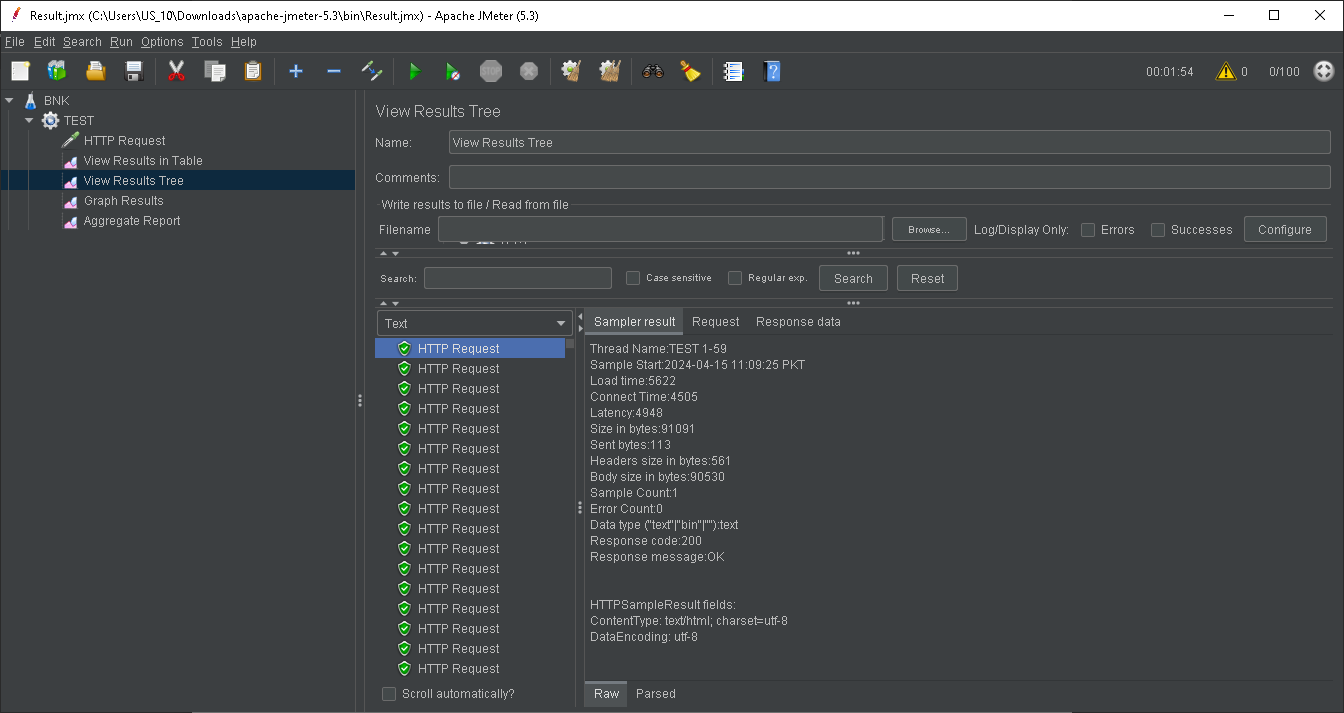


Рисунок 12 – Результат тестирования в виде дерева

Тестирование сайта GitLab показано на рисунках 13-16

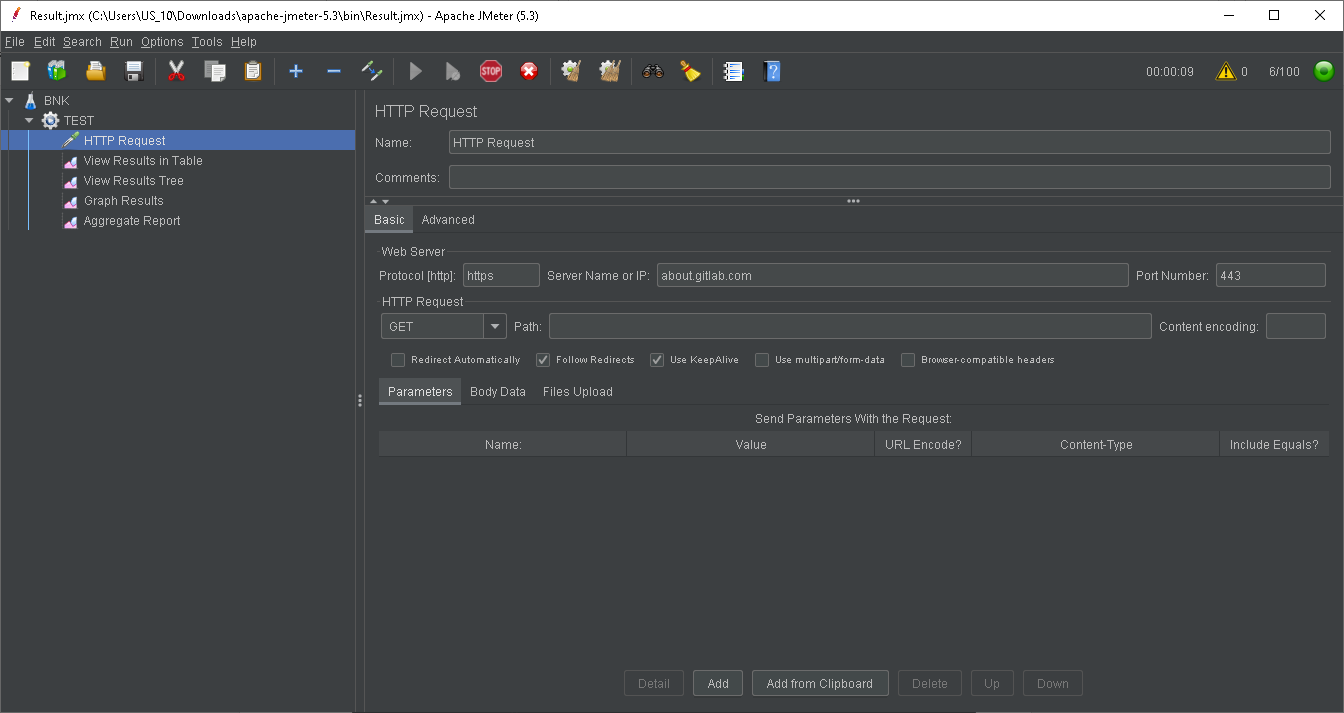


Рисунок 13 – Вводим данные сайта

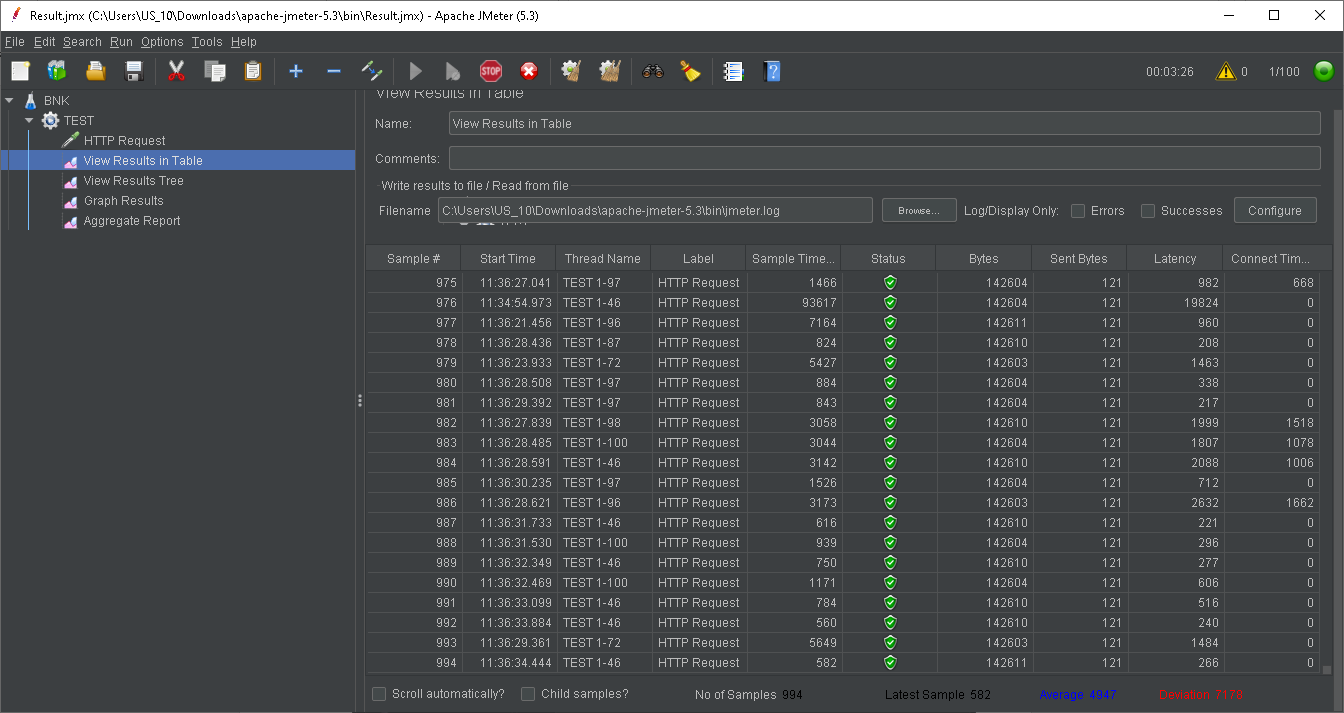


Рисунок 14 – Результат тестирования в виде таблицы

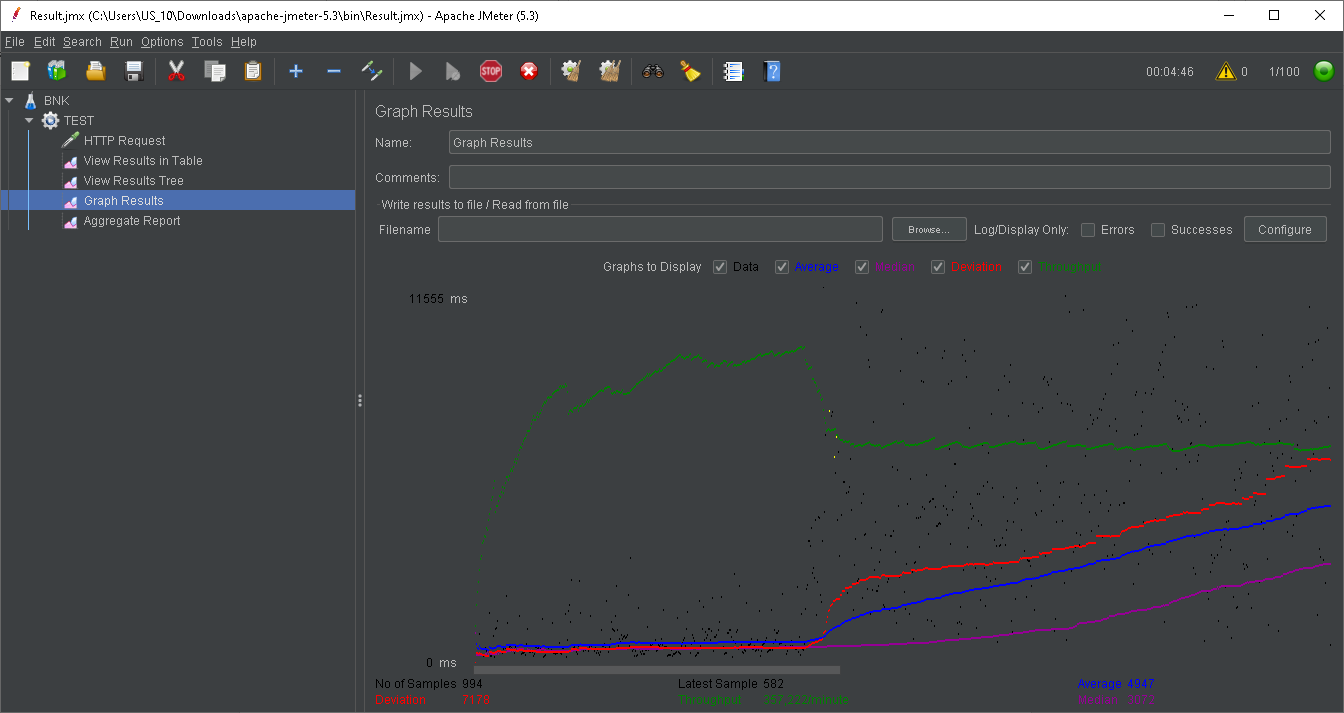


Рисунок 15 – Результат тестирования в виде графика

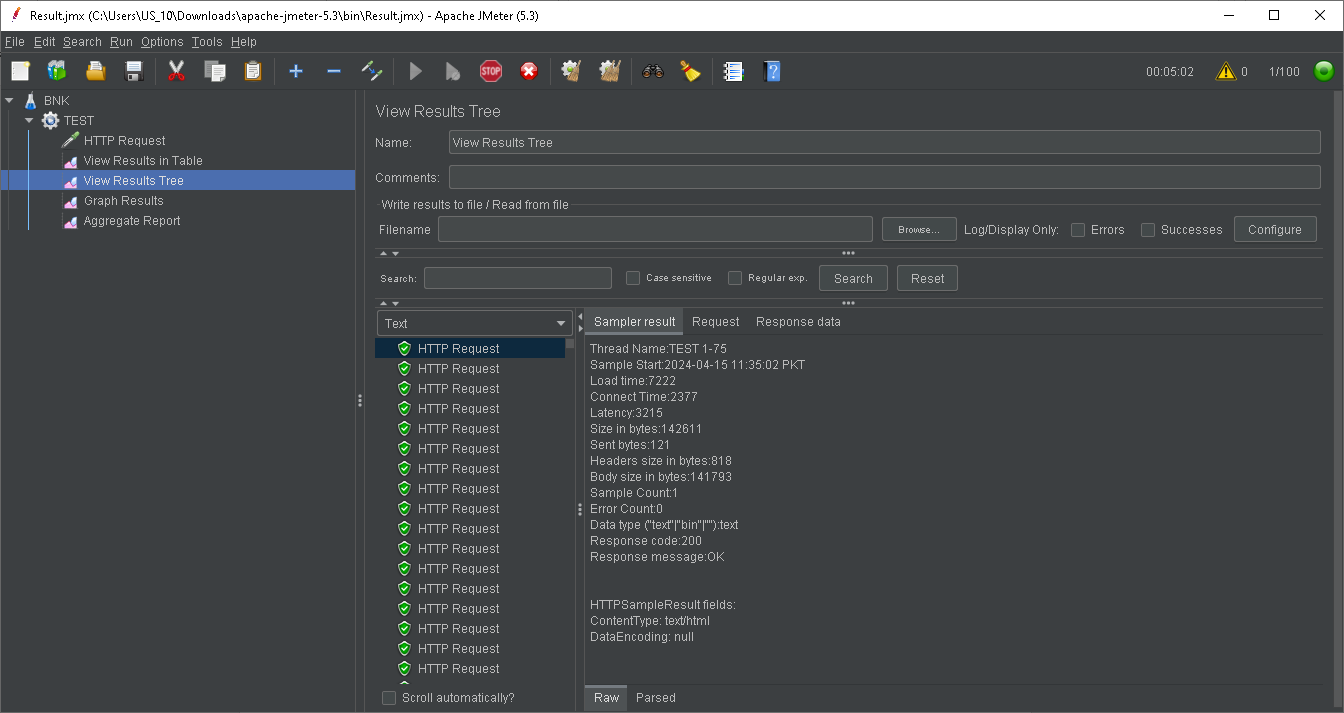


Рисунок 16 – Результат тестирования в виде дерева

Тестирование сайта Pinterest показано на рисунках 17-20

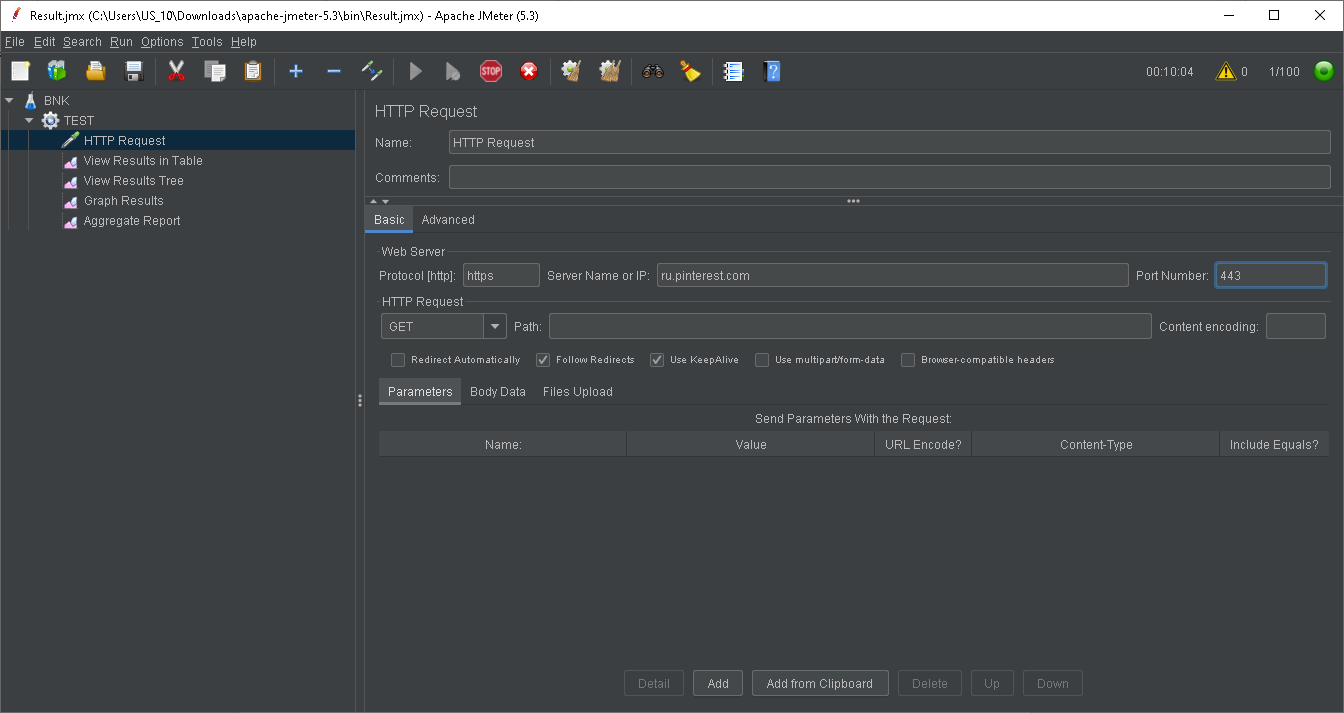


Рисунок 17 – Вводим данные сайта

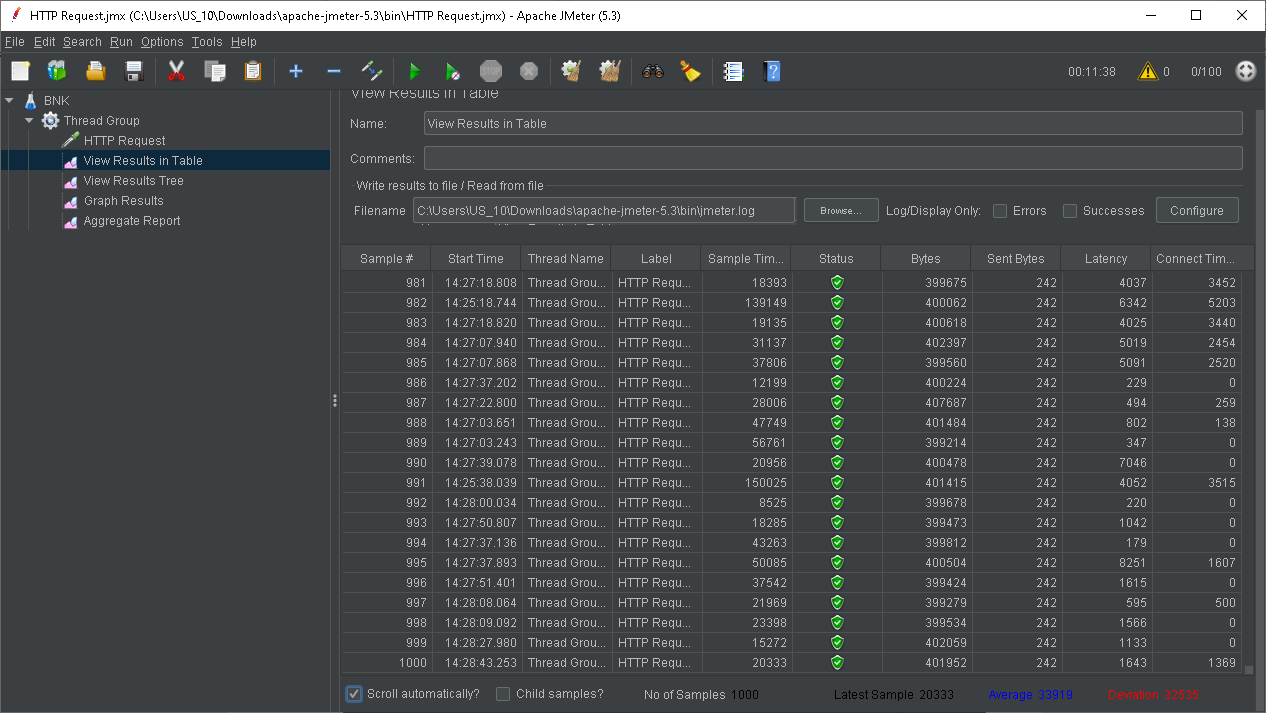


Рисунок 18 – Результат тестирования в виде таблицы

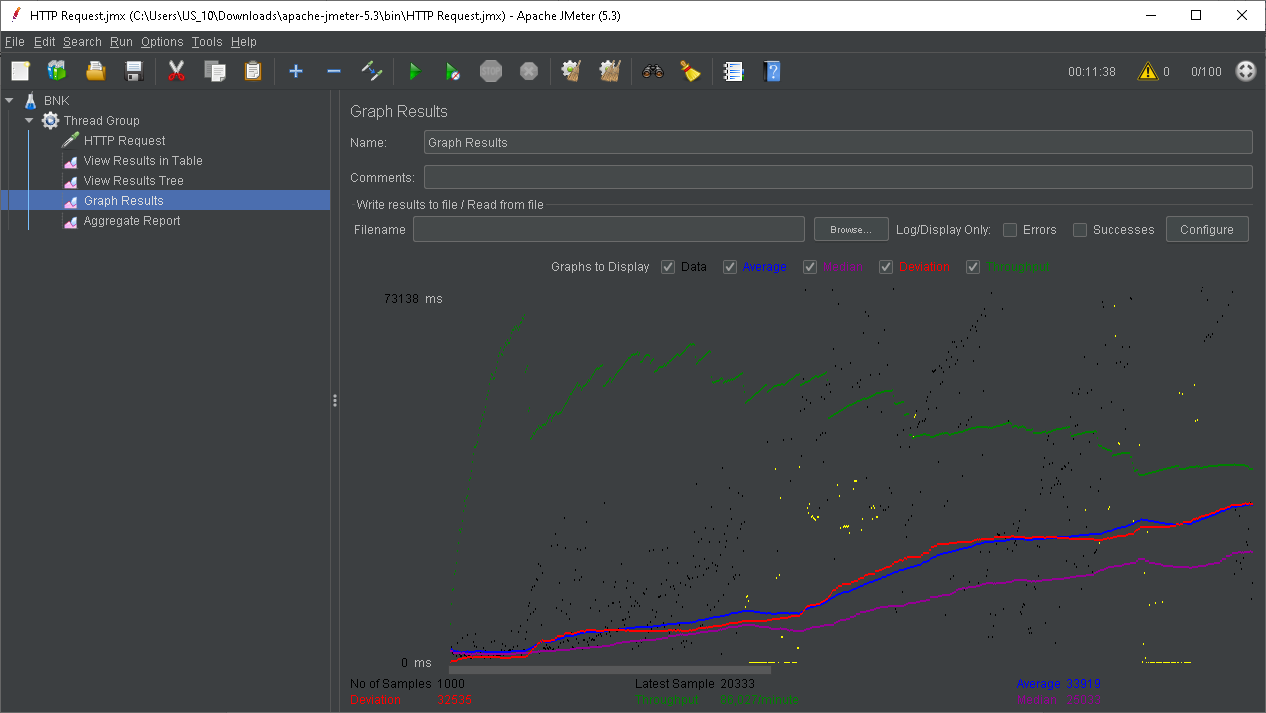


Рисунок 19 – Результат тестирования в виде графика

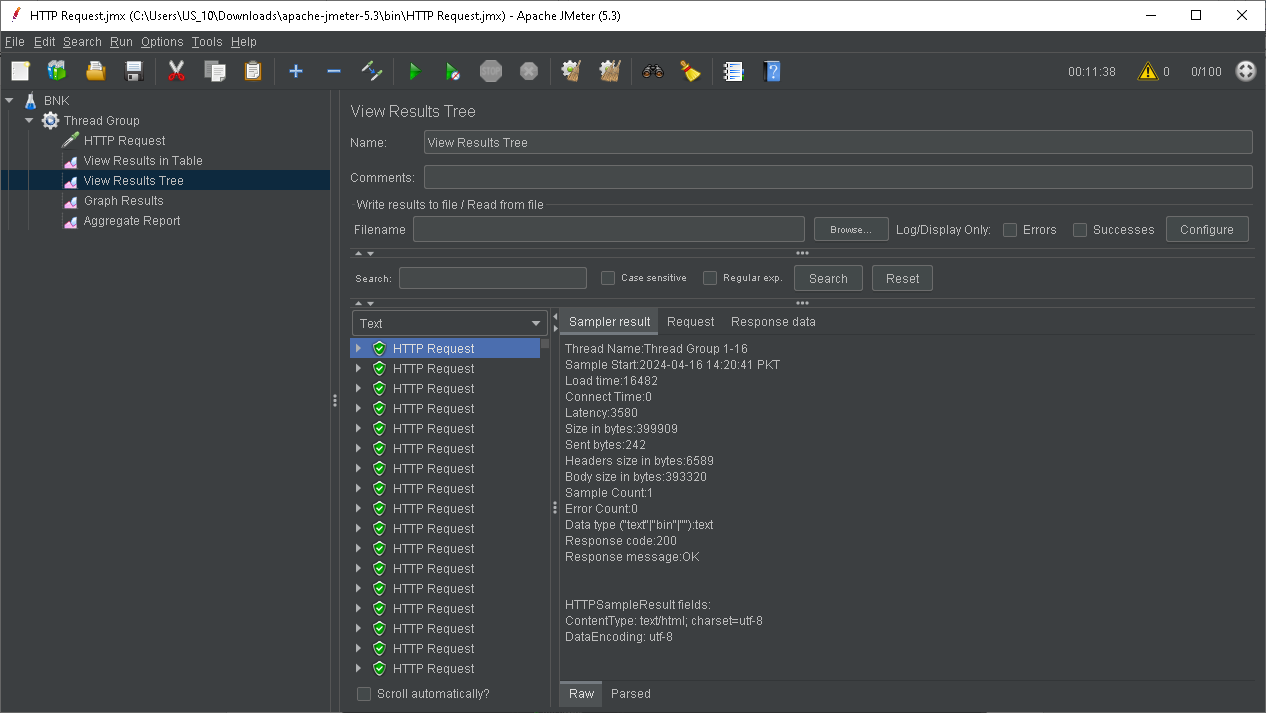


Рисунок 20 – Результат тестирования в виде дерева

В результате проведенного тестирования трех различных веб-сайтов с использованием плана тестирования, включающего 100 потоков, 10 циклов и период наращивания в 100, можно сделать следующие выводы:

Производительность веб-сайтов:

Веб-сайт БНК продемонстрировал стабильную производительность на протяжении всего теста, с незначительным увеличением времени ответа со временем.

Веб-сайт Git Lab показал некоторые признаки деградации производительности по мере увеличения нагрузки, проявляющиеся в увеличении времени ответа.

Веб-сайт Pinterest испытал серьезные проблемы с производительностью при высокой нагрузке, что привело к значительному увеличению времени ответа и к отказам.

Стабильность и надежность:

Веб-сайт БНК проявил стабильность и надежность, не испытывая существенных сбоев или отказов во время тестирования.

Веб-сайт Git Lab показал некоторые признаки нестабильности при высокой нагрузке, возможно, из-за неэффективного управления ресурсами или недостаточной масштабируемости.

Веб-сайт Pinterest демонстрировал критическую нестабильность, что может свидетельствовать о неэффективной архитектуре приложения или недостаточной оптимизации.

Необходимость оптимизации и масштабирования:

Веб-сайтам Git Lab и Pinterest требуется оптимизация и, возможно, масштабирование для обеспечения стабильной производительности при высокой нагрузке.

Веб-сайт БНК также может потребовать оптимизации, особенно если предполагается дальнейшее увеличение нагрузки.

Рекомендации:

Для веб-сайтов Git Lab и Pinterest рекомендуется проведение дополнительных тестов и анализа производительности с целью выявления проблемных участков и принятия соответствующих мер.

Для всех трех веб-сайтов рекомендуется регулярное мониторинг производительности и проведение нагрузочного тестирования после внесения изменений в архитектуру или код.