

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО  
ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ  
НАПРАВЛЕНИЕ ПРОГРАММНАЯ ИНЖЕНЕРИЯ  
ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА СИСТЕМНОЕ И ПРИКЛАДНОЕ  
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ  
СПЕЦИАЛИЗАЦИЯ СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3**  
**курса «Тестирование программного обеспечения»**  
**по теме: «Нагрузочное тестирование»**  
**Вариант по номеру ИСУ**

Выполнил студент:  
Тюрин Иван Николаевич  
группа: Р33102

Преподаватель:  
Клименков С.В.,  
Харитонов А.Е.

Санкт-Петербург, 2024 г.

# Содержание

<b>Лабораторная работа № 3. Нагрузочное тестирование</b>	<b>2</b>
1. Задание варианта по номеру ИСУ . . . . .	2
2. Выполнение задания . . . . .	4
3. Нагрузочное тестирование . . . . .	4
1. Конфигурирование . . . . .	4
2. Запуск тестов . . . . .	6
3. Результаты тестирования . . . . .	7
4. Стресс-тестирование . . . . .	8
1. Конфигурирование . . . . .	8
2. Запуск тестов . . . . .	8
3. Результаты тестирования . . . . .	8
5. Вывод . . . . .	9

# Лабораторная работа № 3

## Нагрузочное тестирование

### 1. Задание варианта по номеру ИСУ

, , ,

#### Лабораторная работа #4

С помощью программного пакета <https://jmeter.apache.org/> провести нагрузочное и стресс-тестирование веб-приложения в соответствии с вариантом задания.

В ходе нагрузочного тестирования необходимо протестировать 3 конфигурации аппаратного обеспечения и выбрать среди них наиболее дешёвую, удовлетворяющую требованиям по максимальному времени отклика приложения при заданной нагрузке (в соответствии с вариантом).

В ходе стресс-тестирования необходимо определить, при какой нагрузке выбранная на предыдущем шаге конфигурация перестает удовлетворять требованиям по максимальному времени отклика. Для этого необходимо построить график зависимости времени отклика приложения от нагрузки.

Приложение для тестирования доступно только во внутренней сети кафедры.

Если запрос содержит некорректные параметры, сервер возвращает HTTP 403.

Если приложение не справляется с нагрузкой, сервер возвращает HTTP 503.

Параметры тестируемого веб-приложения:

- URL первой конфигурации (\$ 3800) -  
<http://stload.se.ifmo.ru:8080?token=492521201&user=2129818211&config=1;>
- URL второй конфигурации (\$ 4100) -  
<http://stload.se.ifmo.ru:8080?token=492521201&user=2129818211&config=2;>
- URL третьей конфигурации (\$ 7200) -  
<http://stload.se.ifmo.ru:8080?token=492521201&user=2129818211&config=3;>
- Максимальное количество параллельных пользователей - 14;

- Средняя нагрузка, формируемая одним пользователем - 20 запр. в мин.;
- Максимально допустимое время обработки запроса - 690 мс.

Отчёт по работе должен содержать:

1. Текст задания.
2. Описание конфигурации JMeter для нагрузочного тестирования.
3. Графики пропускной способности приложения, полученные в ходе нагрузочного тестирования.
4. Выводы по выбранной конфигурации аппаратного обеспечения.
5. Описание конфигурации JMeter для стресс-тестирования.
6. График изменения времени отклика от нагрузки для выбранной конфигурации, полученный в ходе стресс-тестирования системы.
7. Выводы по работе.

Вопросы к защите лабораторной работы:

1. Тестирование системы целиком - системное тестирование
2. Тестирование возможностей, стабильности, отказоустойчивости, совместимости
3. Тестирование производительности - CARAT
4. Альфа и Бета тестирование. Приемочное тестирование
5. Нагрузочное тестирование - виды, цели и решаемые задачи.
6. Принципы реализации нагрузочного тестирования ПО.
7. Инструменты для реализации нагрузочного тестирования.
8. Apache JMeter - архитектура, поддерживаемые протоколы, особенности конфигурации.
9. Стресс-тестирование - основные понятия, виды стресс-сценариев.
10. Стресс-тестирование ПО. Виды стресс-тестов ПО. Тестирование ёмкости.

, , ,

## 2. Выполнение задания

Для получения доступа к серверу нужно использовать кафедральный VPN или пробросить порты через ssh:

```
1 ssh -N -L 8080:stload.se.ifmo.ru:8080 username@helios.se.ifmo.ru -p 2222
```

здесь опция `-N` говорит не выполнять дополнительных команд, а лишь построить подключение.

Для каждого тестового плана инкрементально составляется конфигурация для тестирования: добавляется Thread Group, в нее добавляется HTTP Request, Assertions, Timers и прочее. Для удобства конфигурирования в тестовый план добавляется User Defined Variables с указанием всех нужных параметров теста, см. табл. 1.1. Удобнее всего конфигурацию составлять в графическом приложении JMeter, но вообще формат `.jmx` является стандартом и можно использовать другие инструменты.

Name	Value	Description
THREAD_GROUP_RUMPUP_PERIOD	120	
THREAD_GROUT_LOOP_COUNT	20	
MAX_NUM_PARALLEL_USERS	14	Максимальное количество параллельных пользователей
AVG_USER_LOAD_RPM	20	Средняя нагрузка, формируемая одним пользователем
MAX_REQ_PROCESSING_TIME_MS	690	Максимально допустимое время обработки запроса
SERVER_NAME	localhost	Порт проброшен на localhost
SERVER_PORT	8080	Номер проброшенного порта
GET_PARAM_TOKEN	492521201	Значение токена из GET-запроса
GET_PARAM_USER	2129818211	Значение идентификатора пользователя из GET-запроса

Таблица 1.1: Пример содержания User Defined Variables

Так как в задании нет требований к времени увеличения нагрузки и количества итераций, то значения для параметров `TREAD_GROUP_RUMPUP_PERIOD` и `TREAD_GROUT_LOOP_COUNT` устанавливаются на усмотрение тестировщика. Так как они влияют на время прохождения тестов, то для отладки использовались значения поменьше, порядка 5–10, а для проведения полноценного тестирования можно установить и больше, порядка 10–100.

## 3. Нагрузочное тестирование

### 3. 1. Конфигурирование

Для нагрузочного тестирования в тестовом плане получается 3 Thread Group: по одной для каждой из 3 конфигураций, содержимое этих групп в тестовом плане в сущности отличается лишь номером конфигурации в запросе.

В каждой конфигурации Tread Group используется такой набор функциональных нод:

- HTTP Request (Thread Group > Add > Sampler) – отправка HTTP-запроса с параметрами на указанный URL (`GET_PARAM_TOKEN`, `GET_PARAM_USER`),
- Duration Assertion (Thread Group > Add > Assertions) – ограничитель на время выполнения запроса (`MAX_REQ_PROCESSING_TIME_MS`),
- Constant Throughput Timer (Thread Group > Add > Timer) – ограничение количества запросов в минуту (`AVG_USER_LOAD_RPM`), рассчитанное для всех потоков в группе.

Для отладки конфигурации удобно добавить вывод результатов в таблицу и на график:

- View Results in Table (Thread Group > Add > Listener),
- Graph Results (Thread Group > Add > Listener).

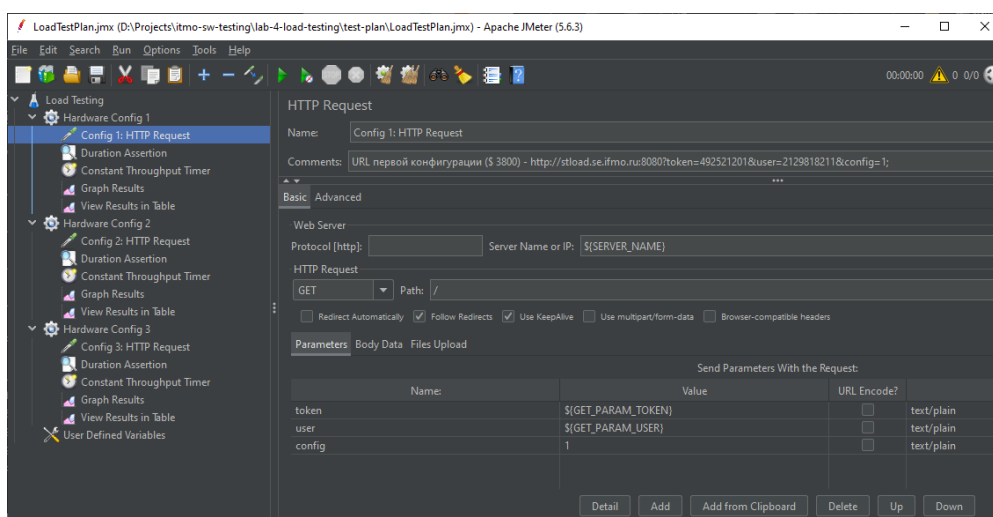


Рис. 1.1: Скриншот графического приложения JMeter для демонстрации настройки семлера

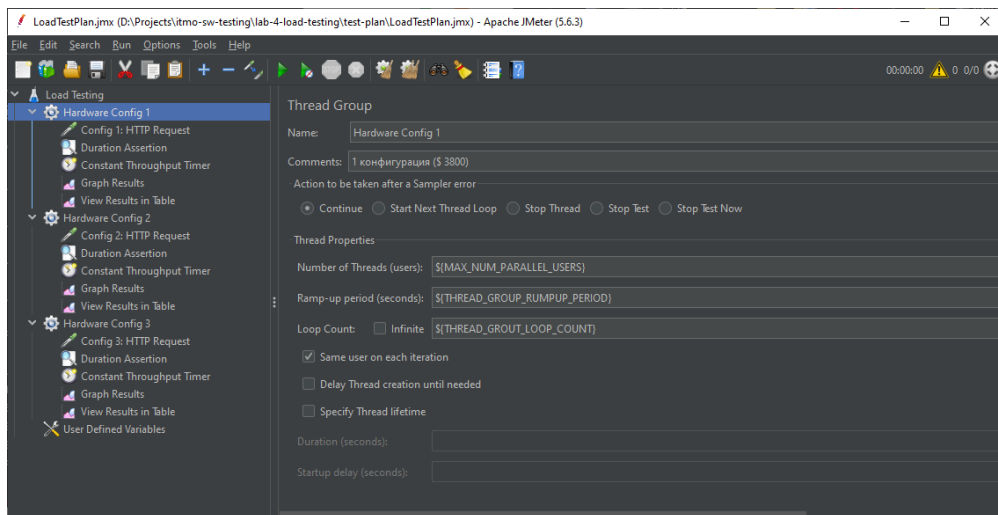


Рис. 1.2: Скриншот графического приложения JMeter для демонстрации настройки Thread Group

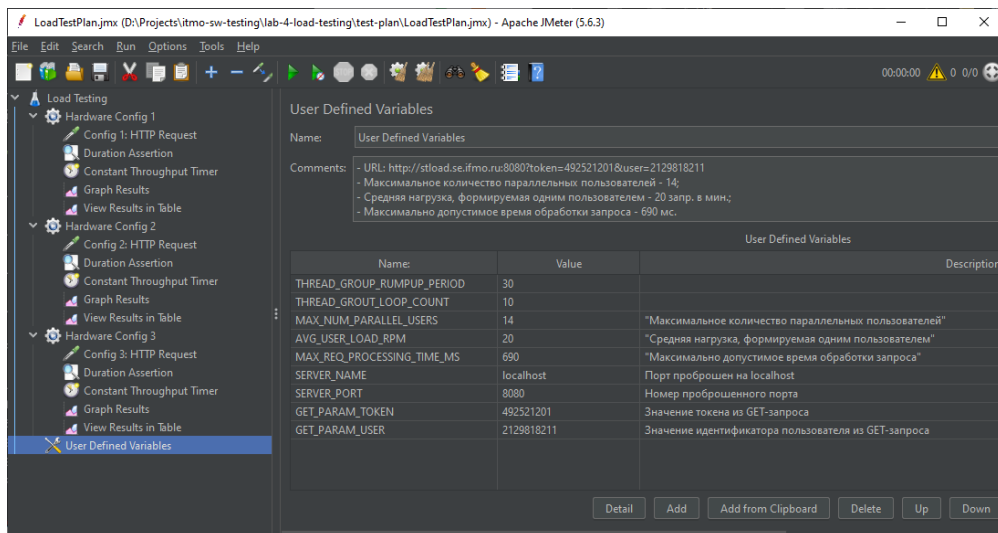


Рис. 1.3: Скриншот графического приложения JMeter для демонстрации настройки пользовательских переменных

## 3. 2. Запуск тестов

Запуск тестирования, как рекомендуется в документации, производится из консоли, без графического интерфейса с указанием флагов для сохранения результатов:

```
1 mkdir load-test
2 jmeter.bat -n -t test-plan/LoadTestPlan.jmx -l load-test/results.log -e -o
  load-test/report
```

### 3. 3. Результаты тестирования

Еще на этапе отладки стало ясно, что первая и вторая конфигурации не укладываются во временные ограничения, их нормальное время ответа было 750 и 1000 мс, что больше требуемых 690 мс.

В результате тестирования JMeter предоставляет статистику по прохождению тестов в виде HTML-страницы. Как и ожидалось, две трети всех запросов завершились с ошибкой, см. 1.4.

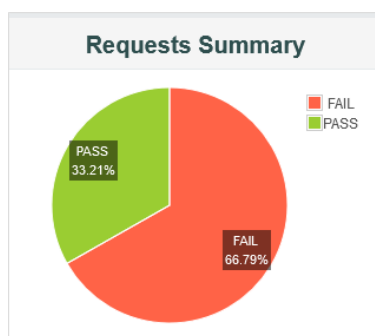


Рис. 1.4: Request Summary по выполненным тестам

На этом этапе нас не интересует зависимость времени ответа от количества запросов, нам больше интересно распределение времени ответа для разных конфигураций. Как можно видеть на рис. 1.5, среднее время ответа для 1 и 2 конфигураций выходит за рамки ограничений и находится в пределах 700–800 и 1100–1200 мс соответственно. Среднее время ответа 3 конфигурации получилось 300–400 мс.

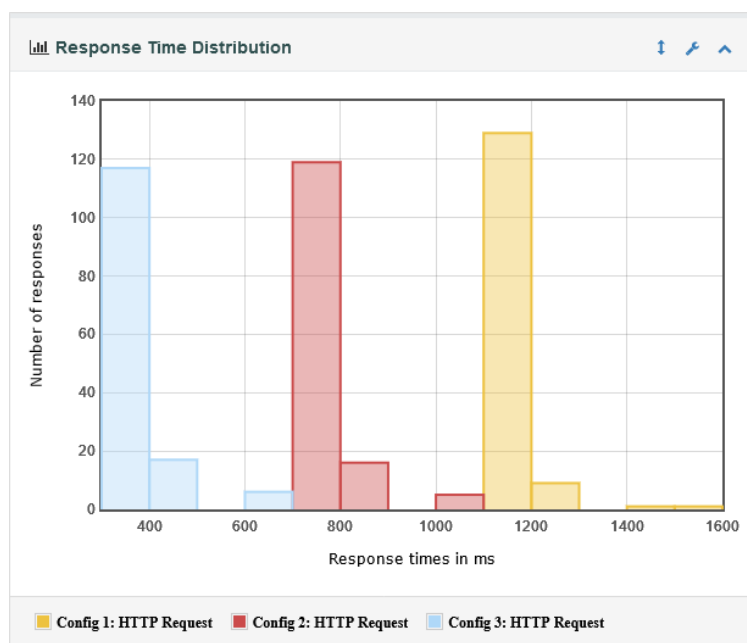


Рис. 1.5: Response Time Distribution диаграмма для нагрузочного тестирования



## 4. Стресс-тестирование

Для дальнейшего тестирования выбирается 3 конфигурация оборудования, т.к. она единственная может отвечать поставленным ограничениям.

### 4. 1. Конфигурирование

В качестве стресс-нагрузки будет использоваться такая же конфигурация как и ранее, но с большим числом параллельно работающих пользователей и дополнительным ограничением в виде Response Assertion (Thread Group > Add > Assertionas) — ограничитель на Response Code равный 200.

### 4. 2. Запуск тестов

Тесты запускаются так же без графического интерфейса из командной строки:

```
1 mkdir stress-test
2 jmeter.bat -n -t test-plan/StressTestPlan.jmx -l stress-test/results.log -
  e -o stress-test/report
```

### 4. 3. Результаты тестирования

По графикам из отчетов по тестированию можно понять, что повышением нагрузки на сервер он начинает отвечать медленнее. Но как можно видеть, время ответа в очень редких случаях выходит за границу допустимых значений.

Для лучшего моделирования повышения нагрузки имеет смысл использовать плагин для JMeter, предоставляющий Stepping Thread Group (<https://jmeter-plugins.org/wiki/SteppingThreadGroup/>)

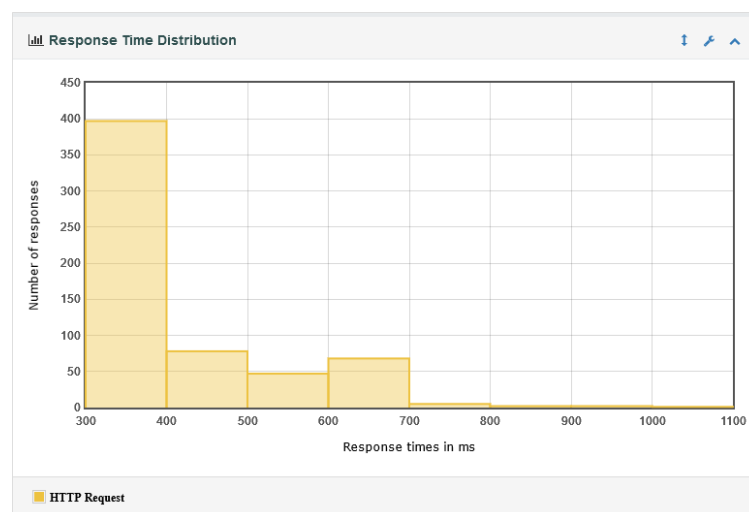


Рис. 1.6: Response Time Distribution диаграмма для стресс-тестирования

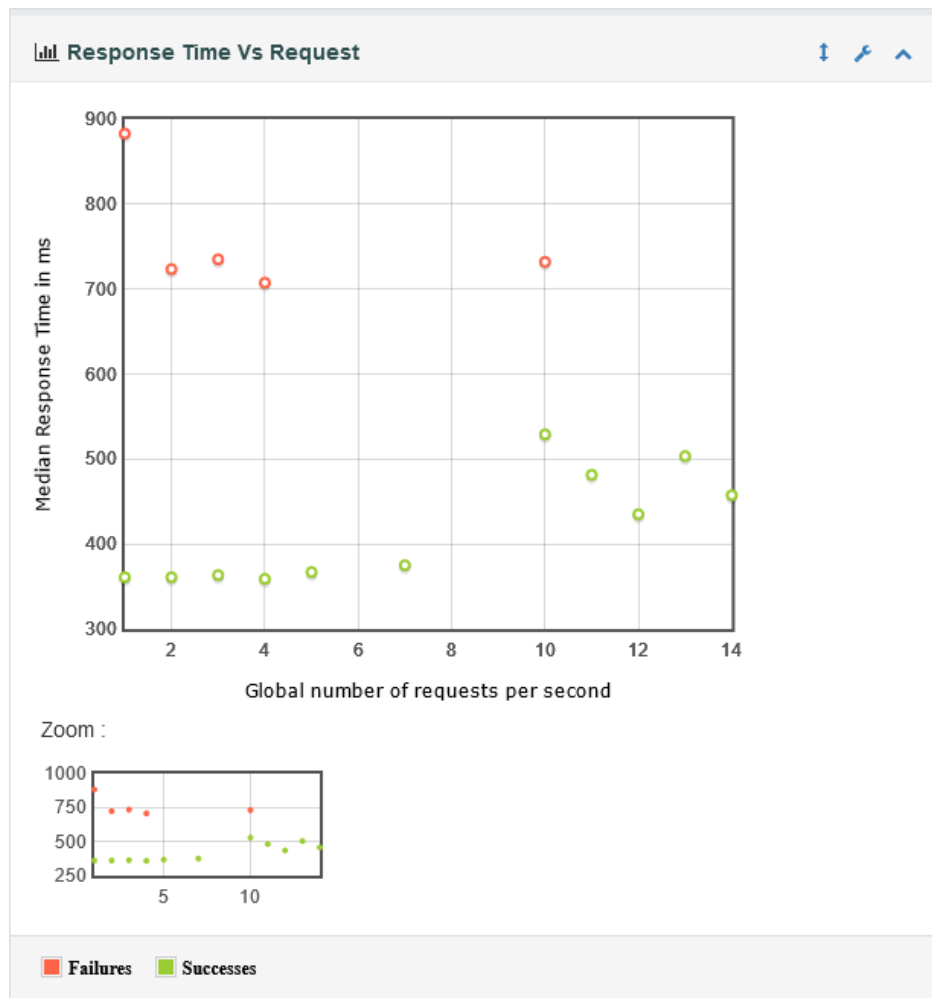


Рис. 1.7: Response Time vs Requests график для стресс-тестирования

## 5. Вывод

В соответствии с заданием было проведено нагрузочное и стресс тестирование выданных по варианту сервисов. Изучен процесс конфигурации тестовых планов с помощью JMeter и их выполнение.

Все исходные коды можно найти в репозитории: <https://github.com/e1turin/itmo-sw-testing/> в модуле lab-4-load-testing.