



**УНИВЕРСИТЕТ ИТМО**

ФГАОУ ВО «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

---

**ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

**Лабораторная работа №4**

Нагрузочное и стресс-тестирование

---

Лабушев Тимофей

Группа Р3402

Санкт-Петербург

2020

## Задание

С помощью программного пакета *Apache JMeter* 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 следующей нагрузке:

1. максимальное количество параллельных сессий — 14;
2. средняя нагрузка, формируемая одной сессией — 20 запросов в минуту;
3. максимально допустимое время обработки запроса — 880 мс.

При выборе конфигурации принять следующие стоимости:

- Конфигурация 1 — \$2100
- Конфигурация 2 — \$3200
- Конфигурация 3 — \$3800

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

## Тестовой план JMeter для нагрузочного тестирования

Структура тестового плана:

- Thread Group (Number of Threads: 14, Loop Count: 50)
  - HTTP Request (`http://...&conf=1`)
    - Constant Throughput Timer (20 samples per minute)
- Thread Group (Number of Threads: 14, Loop Count: 50)
  - HTTP Request (`http://...&conf=2`)
    - Constant Throughput Timer (20 samples per minute)
- Thread Group (Number of Threads: 14, Loop Count: 50)
  - HTTP Request (`http://...&conf=2`)
    - Constant Throughput Timer (20 samples per minute)

Для запуска тестового плана использовалась следующая команда:

```
apache-jmeter-5.3/bin/jmeter -n -t load-test.jmx -l load-test.csv
```

Где `load-test.jmx` — файл конфигурации тестового плана, `load-test.csv` — файл с выходными значениями, в который записываются показатели каждого отправленного запроса.

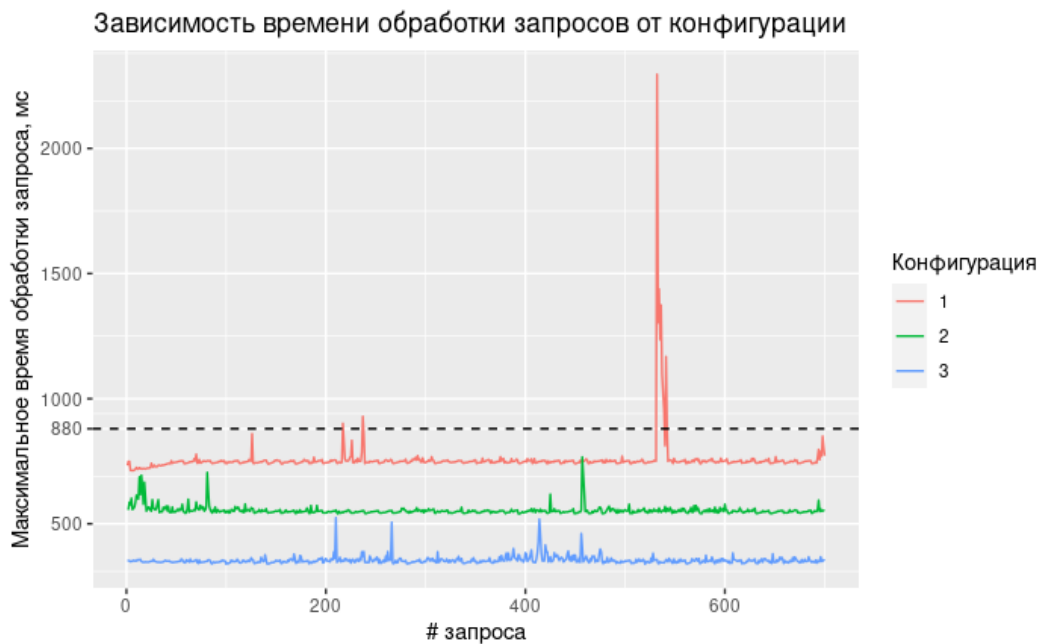
# Результаты нагрузочного тестирования

По собранным данным составим сравнительную таблицу конфигураций:

Конфигурация	Максимальное время обработки запроса, мс	Среднее время обработки запроса, мс	Стоимость
1	2296	757	\$2100
2	767	553	\$3200
3	524	354	\$3800

Максимальное время обработки запроса при использовании первой конфигурации в разы превышает требования к системе, однако среднее время обработки запроса им удовлетворяет.

Чтобы убедиться, что высокий показатель не был единичным случаем, построим график изменения времени обработки запросов на протяжении теста:



Можем увидеть, что при использовании первой конфигурации превышение максимально допустимого времени происходит многократно. Таким образом, **оптимальной является вторая конфигурация**, которая удовлетворяет требованиям и имеет меньшую стоимость по сравнению с третьей.

## Тестовой план JMeter для стресс-тестирования

Стресс-тестирование произведено для второй конфигурации, которая была выбрана как наиболее дешевая, при этом удовлетворяющая заданным требованиям.

Структура тестового плана:

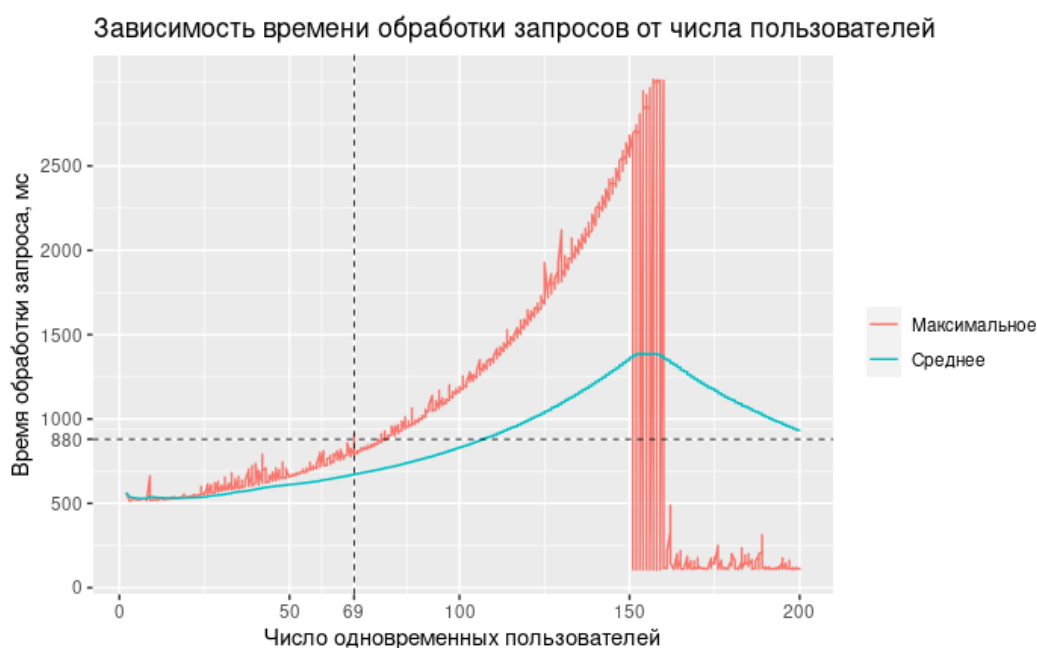
- Thread Group (Number of Threads: 200, Ramp-up period: 90 seconds)
  - HTTP Request (`http://...&conf=2`)
    - Constant Throughput Timer (20 samples per minute)

Тестовый план запускается следующей командой:

```
apache-jmeter-5.3/bin/jmeter -n -t load-test.jmx -l load-test.csv
```

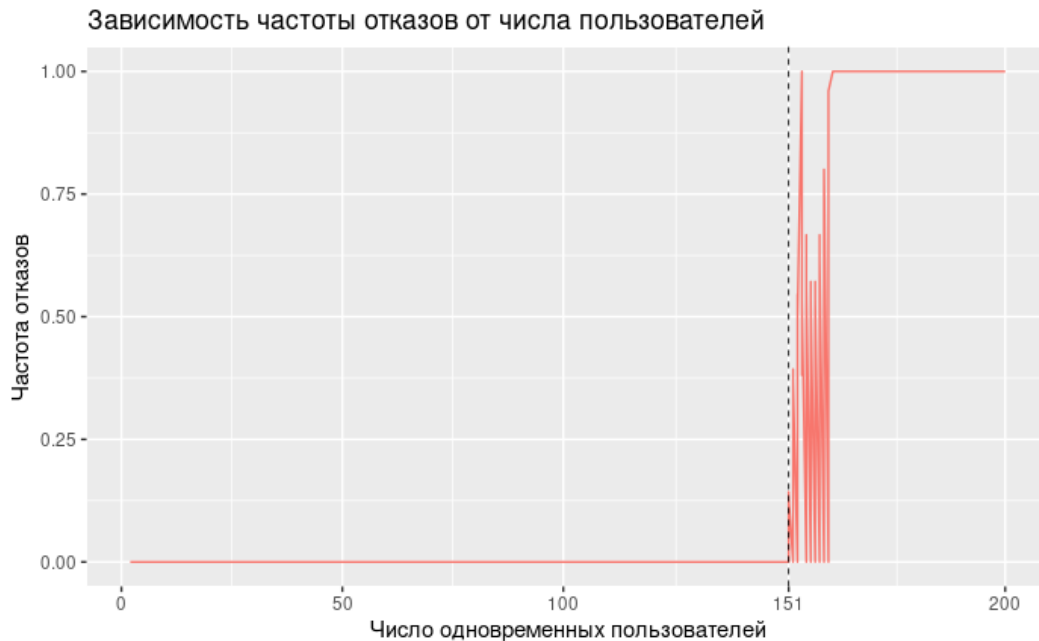
## Результаты стресс-тестирования

Построим график зависимости максимального и среднего времени обработки запросов от числа одновременных пользователей:



Начиная с **69** одновременных пользователей, система прекращает удовлетворять требованиям к максимальному времени обработки запроса. Среднее время обработки запроса продолжает оставаться ниже 880 мс в пределах 100 пользователей.

При достижении 150 пользователей наблюдается странный эффект: время обработки запросов резко падает. Объяснение подобному поведению можно найти, построив график частоты ошибок, возвращаемых сервером:



Начиная со 151 одновременного пользователя выбранная конфигурация не справляется с нагрузкой и начинает отказывать в обслуживании, возвращая ошибки HTTP 503.

## Исходный код

<https://github.com/timlathy/itmo-fourth-year/tree/master/Software-Testing-7th-Term/Lab4>

## Выводы

В ходе выполнения работы был рассмотрен инструмент для проведения нагрузочного тестирования *Apache JMeter*. Было произведено сравнение трех конфигураций на основании данных, собранных в ходе тестирования, и принято решение о выборе оптимальной из них.

Для выбранной конфигурации также было произведено стресс-тестирование, в ходе чего было определено максимальное количество одновременных пользователей, удовлетворяющих требованиям, а также число пользователей, при котором система начинает отказывать в обслуживании.