Университет ИТМО

Факультет ПИиКТ

Тестирование программного обеспечения

Лабораторная работа №4

Вариант 264429

Выполнила: Наумова Н. А.

Группа P33022

Преподаватель: Исаев И. В.

Санкт-Петербург

2021 г.

**Задание**

С помощью программного пакета [Apache JMeter](http://jmeter.apache.org/" \t "_blank) провести нагрузочное и стресс-тестирование веб-приложения в соответствии с вариантом задания.

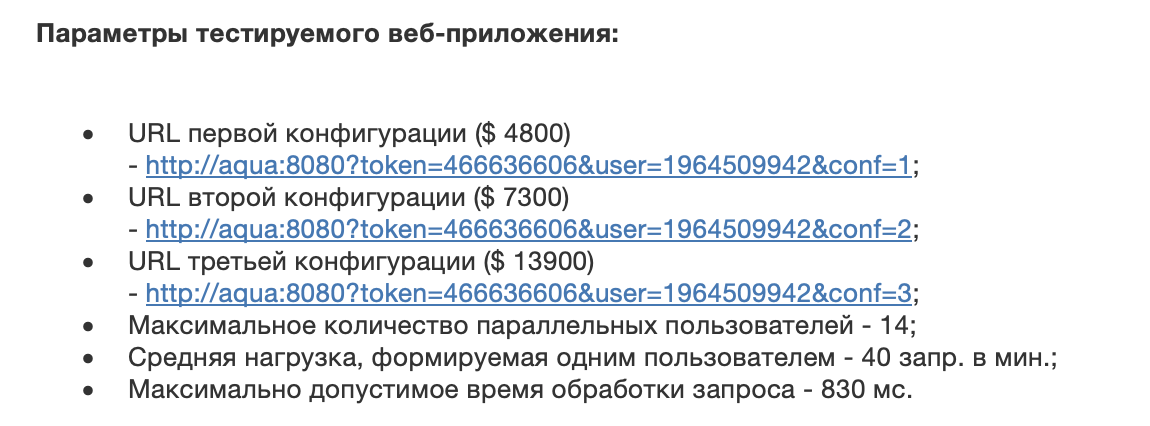
В ходе нагрузочного тестирования необходимо протестировать 3 конфигурации аппаратного обеспечения и выбрать среди них наиболее дешёвую, удовлетворяющую требованиям по максимальному времени отклика приложения при заданной нагрузке (в соответствии с вариантом).

В ходе стресс-тестирования необходимо определить, при какой нагрузке выбранная на предыдущем шаге конфигурация перестаёт удовлетворять требованиями 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.

**Webapp properties:**

* First hardware configuration ($ 4800) URL - [http://aqua:8080?token=466636606&user=1964509942&conf=1](https://se.ifmo.ru/appUrl);
* Second hardware configuration ($ 7300) URL - [http://aqua:8080?token=466636606&user=1964509942&conf=2](https://se.ifmo.ru/appUrl);
* Third hardware configuration ($ 13900) URL - [http://aqua:8080?token=466636606&user=1964509942&conf=3](https://se.ifmo.ru/appUrl);
* Maximum parallel sessions count - 14;
* Load average (requests per minute; per session) - 40;
* Maximum request processing timeout - 830 ms.

**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.

**Выполнение работы**

**Выбор оптимальной конфигурации**

**Скриншот структуры тестового планаA computer screen capture

Description automatically generated with medium confidence**

**Выставленные конфигурации**

**Graphical user interface, application

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

**// Чтобы получить график зависимости времени отклика от нагрузки, выставим время между запуском потоков (ramp up period), чтобы был график, а не одна точка на графике – не 0, а хотя бы 14 – так будет запускаться примерно 1 поток в секунду.**

**Graphical user interface, application

Description automatically generated**

**Graphical user interface, application

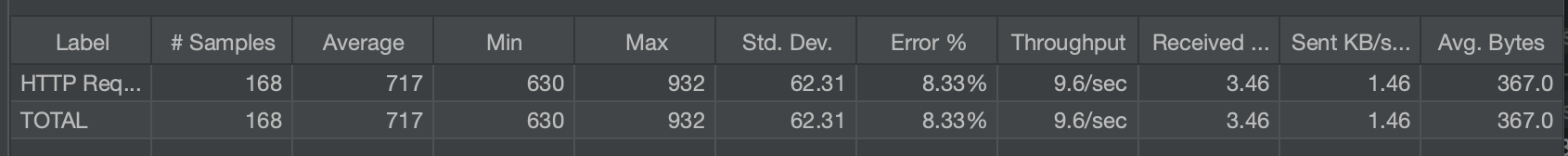
Description automatically generated**

**Graphical user interface, application

Description automatically generated**

/Users/mmmlpmsw/Downloads/apache-jmeter-5.4.1/bin/jmeter.sh -g /Users/mmmlpmsw/Documents/sem/testing/software\_testing\_lab3/aaaa.csv -o /Users/mmmlpmsw/Documents/sem/testing/software\_testing\_lab3/current

**Первая конфигурация (упала практически сразу)**

****

**График зависимости времени отклика от нагрузки**

**Chart, line chart

Description automatically generated**

**Вторая конфигурация (упала достаточно быстро)**

**Graphical user interface

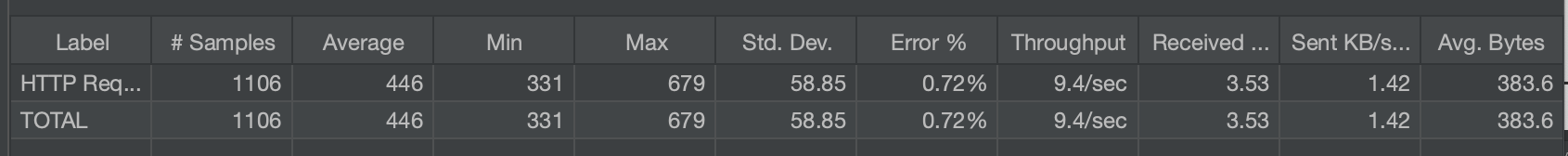
Description automatically generated**

**График зависимости времени отклика от нагрузки**

**Chart, line chart

Description automatically generated**

**Третья конфигурация (результаты спустя 2 минуты после запуска)**

****

**График зависимости времени отклика от нагрузки**

**Chart, line chart

Description automatically generated**

**Поскольку, согласно варианту, максимально допустимое время обработки запроса равно 830 мс, нам подойдет только третья конфигурация.**

**Стресс-тестирование выбранной конфигурации**

**Поменяем некоторые параметры конфигурации:**

* **убираем Duration Assertion, так как сейчас ловим 503, а не не\_проход\_запроса\_по\_времени**
* **менять Stop Test на Continue, потому что сервер может рандомно и случайно не ответить одному из клиентов и это не 503**
* **меняем количество пользователей на 1000, Ramp Up Period – на 4000 – так один поток будет запускаться примерно один раз в 4 секунды**

Выбранная на предыдущем шаге конфигурация перестаёт удовлетворять требованиям по максимальному времени отклика при 133 пользователях.

График зависимости времени отклика приложения от нагрузки:

Chart, line chart

Description automatically generated

Вывод

Проделав данную работу, я познакомилась с программой Apache JMeter и научилась выполнять нагрузочное и стресс-тестирование с помощью данной программы. А еще я узнала, что в Best Practices тестирования умные товарищи говорят, что не надо использовать Apache JMeter.

A picture containing text, clipart

Description automatically generatedDiagram

Description automatically generated

A picture containing text, clipart

Description automatically generated

A picture containing text

Description automatically generated

A cat wearing a crown

Description automatically generated