

Университет ИТМО

Факультет программной инженерии и
компьютерной техники

Направление подготовки 09.03.04 Программная
инженерия

Дисциплина «Тестирование программного
обеспечения»

Отчет

По лабораторной работе №4

Выполнили:
Нуцалханов Н. Г.
Грибов М. О.
Р33101

Преподаватель:
Машина Е. А.

Санкт-Петербург, 2024 г.

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 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.

Параметры тестируемого веб-приложения

- Максимальное количество параллельных пользователей - 9;
- Средняя нагрузка, формируемая одним пользователем - 40 запр. в мин.;
- Максимально допустимое время обработки запроса - 840 мс.

Выполнение

Проброс хоста перед выполнением:

```
ssh -N -L 8080:stload.se.ifmo.ru:8080 s335059@se.ifmo.ru -p 2222
```

Конфигурация для нагрузочного тестирования

Thread Group

Здесь установлены значение “Number of Threads” в 9, что равно максимальному количеству одновременных пользователей по условию:

Name: Thread Group

Comments:

Action to be taken after a Sampler error

☒ Continue ☐ Start Next Thread Loop ☐ Stop Thread ☐ Stop Test ☐ Stop Test Now

Thread Properties

Number of Threads (users): 9

Ramp-up period (seconds):

Loop Count: ☐ Infinite 40

☒ Same user on each iteration

☐ Delay Thread creation until needed

☐ Specify Thread lifetime

Duration (seconds): 30

Startup delay (seconds):

Http Request

Непосредственно запрос к ресурсу. Разные конфигурации тестируются благодаря смене URL параметра config:

Name: HTTP Request

Comments:

Basic Advanced

Web Server

Protocol [http]: Server Name or IP: localhost Port Number: 8080

HTTP Request

GET Path: / Content encoding:

☐ Redirect Automatically ☒ Follow Redirects ☒ Use KeepAlive ☐ Use multipart/form-data ☐ Browser-compatible headers

Parameters Body Data Files Upload

Send Parameters With the Request:

Name:	Value	URL Encode?	Content-Type	Include Equals?
token	492521234	<input checked="" type="checkbox"/>	text/plain	<input checked="" type="checkbox"/>
user	2109714991	<input checked="" type="checkbox"/>	text/plain	<input checked="" type="checkbox"/>
config	2	<input checked="" type="checkbox"/>	text/plain	<input checked="" type="checkbox"/>

Ограничения и проверки

Это ограничение устанавливает максимальное количество запросов за минуту. Установлено значение в 40, как того требует условие:

Constant Throughput Timer

Name: Constant Throughput Timer

Comments:

Delay before each affected sampler

Target throughput (in samples per minute): 40.0

Calculate Throughput based on: this thread only

Проверка на то, что время обработки запроса не больше, чем 840 мс:

Duration Assertion

Name: Duration Assertion

Comments:

Apply to:

☐ Main sample and sub-samples ☒ Main sample only ☐ Sub-samples only

Duration to Assert

Duration in milliseconds: 840

Проверяем, что код ответа ОК:

Response Assertion

Name: Response Assertion

Comments:

Apply to:

☐ Main sample and sub-samples ☒ Main sample only ☐ Sub-samples only ☐ JMeter Variable Name to use

Field to Test

☐ Text Response ☒ Response Code ☐ Response Message ☐ Response Headers

☐ Request Headers ☐ URL Sampled ☐ Document (text) ☐ Ignore Status

☐ Request Data

Pattern Matching Rules

☐ Contains ☐ Matches ☒ Equals ☐ Substring ☐ Not ☐ Or

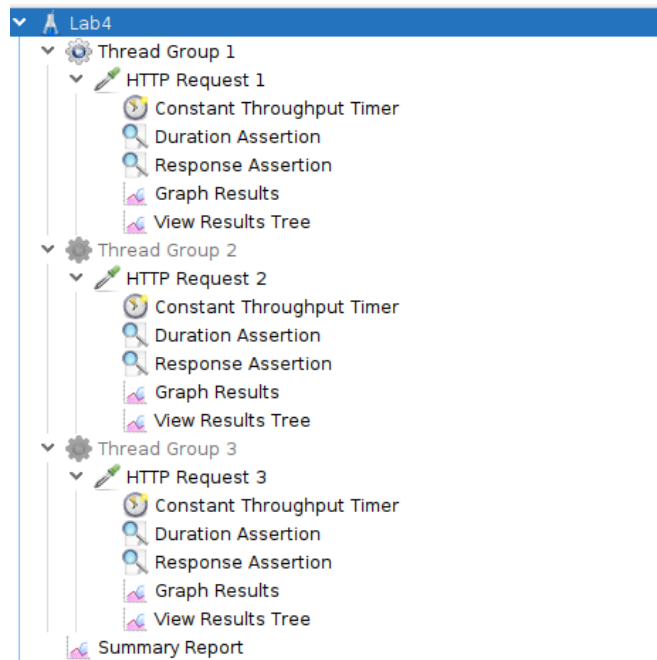
Patterns to Test

Patterns to Test

1	200
---	-----

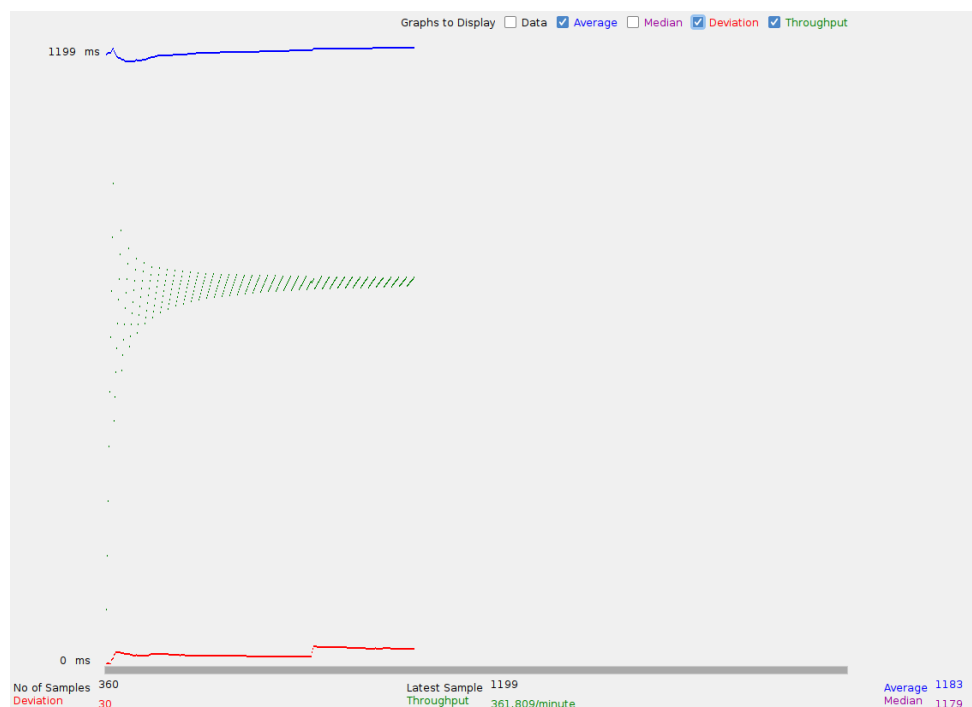
Итоговая структура

Созданы три группы потоков — каждая со своей конфигурацией HTTP Request. Дополнительно во всех группах есть графики с результатами, а Summary Report собирает статистику сразу со всех групп.

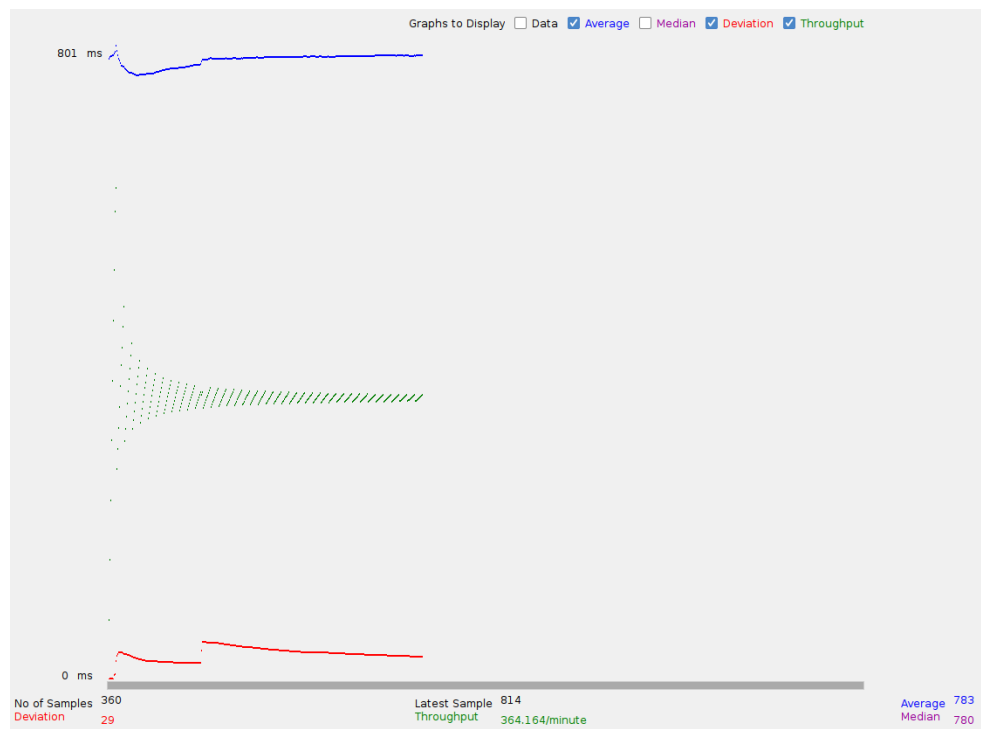


Нагрузочное тестирование

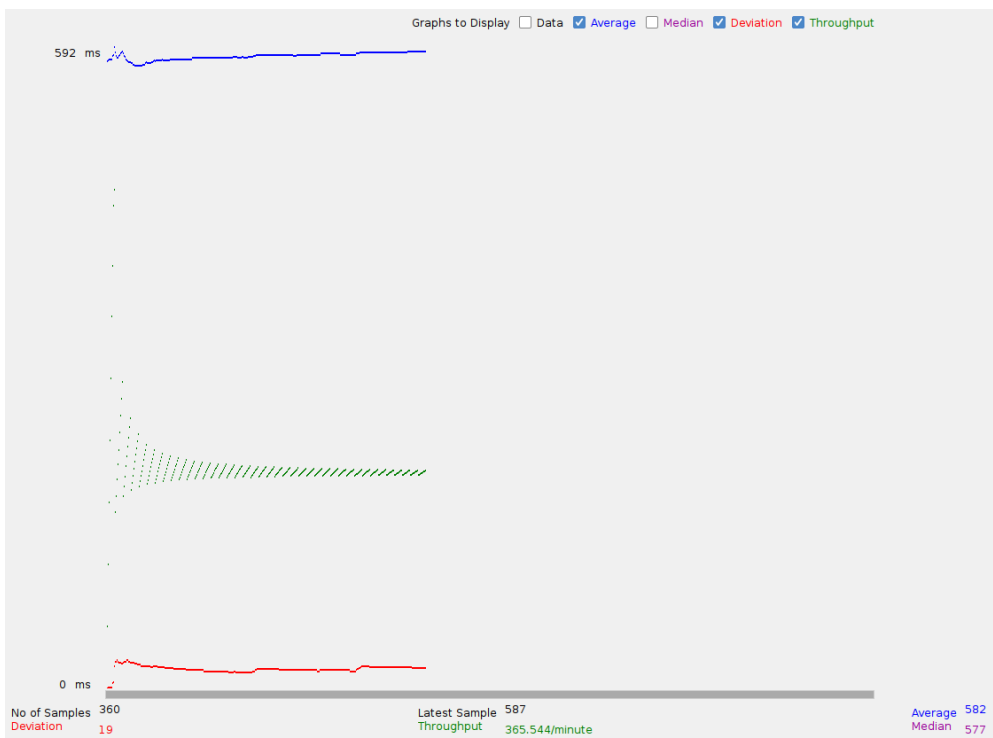
Конфигурация 1



Конфигурация 2



Конфигурация 3



Выводы по нагрузочному тестированию

Общая таблица результатов выглядит так:

Label 1	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request 1	360	1183	1131	1465	30.95	100.00%	6.0/sec	1.36	0.90	231.0
HTTP Request 2	360	783	728	1112	29.56	0.83%	6.1/sec	1.37	0.91	231.0
HTTP Request 3	360	582	534	680	19.70	0.00%	6.1/sec	1.37	0.91	231.0

Первая конфигурация не удовлетворяет ограничению по времени даже в наилучшем случае — минимальное время равно 1131 мс.

Вторая конфигурация в целом удовлетворяет ограничениям. Разве что в какой-то момент задержка увеличилась до 1112 мс, хотя в среднем она держалась на уровне 728 мс.

Третья конфигурация полностью удовлетворяет ограничениям. Даже в худшем из случаев задержка не превышала условленной.

Конфигурация для стресс-тестирования

Для плавного повышения нагрузки решено использовать ramp-up параметр. Максимальное количество пользователей равно 20, ramp-up равен 120, значит каждые 6 секунд появляется один пользователь:

Thread Group

Name:

Thread Group stress

Comments:

Action to be taken after a Sampler error

☒ Continue ☐ Start Next Thread Loop ☐ Stop Thread ☐ Stop Test ☐ Stop Test Now

Thread Properties

Number of Threads (users):

20

Ramp-up period (seconds):

120

Loop Count:

☒ Infinite

☒ Same user on each iteration

☐ Delay Thread creation until needed

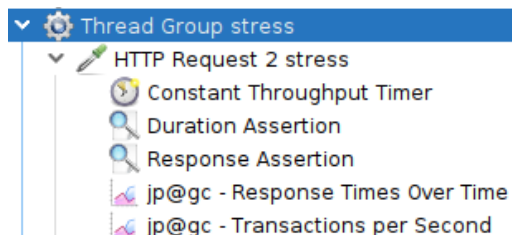
☐ Specify Thread lifetime

Duration (seconds):

30

Startup delay (seconds):

Дополнительно был установлен плагин “3 Basic Graphs”. Из него использовались графики времени ожидания ответа и количества успешных/безуспешных запросов:

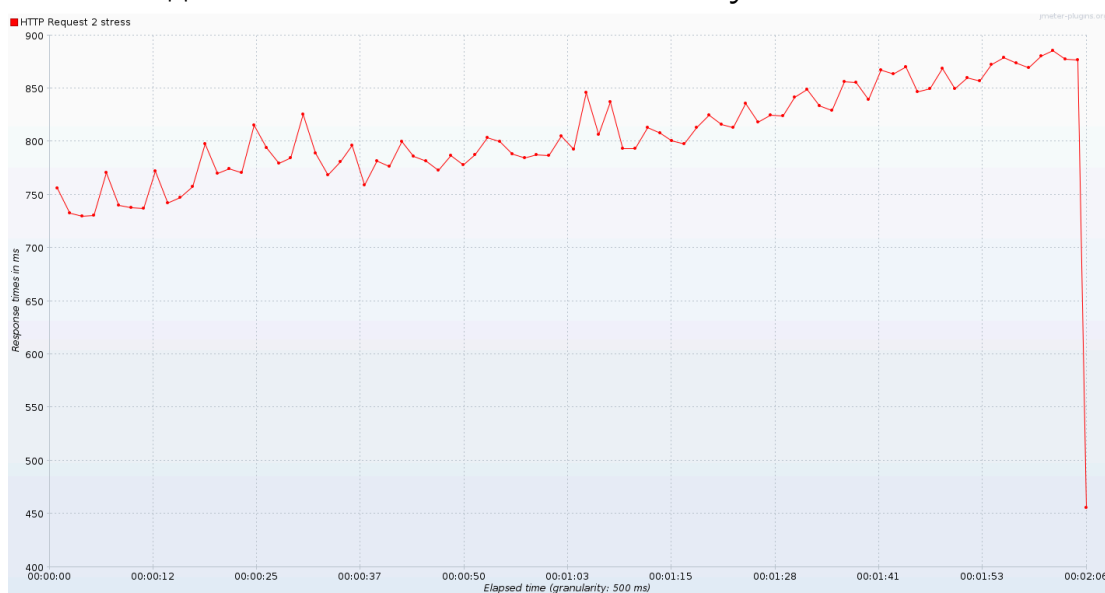


Идея тестирования такова: каждые 6 секунд увеличивается количество пользователей, а следовательно увеличивается нагрузка. По графикам можно понять, в какой момент времени сервер перестает справляться — время ответа превышает 840 мс. По времени уже можно вычислить, сколько в этот момент работает пользователей (т.к. они запускаются в работу с одинаковым интервалом).

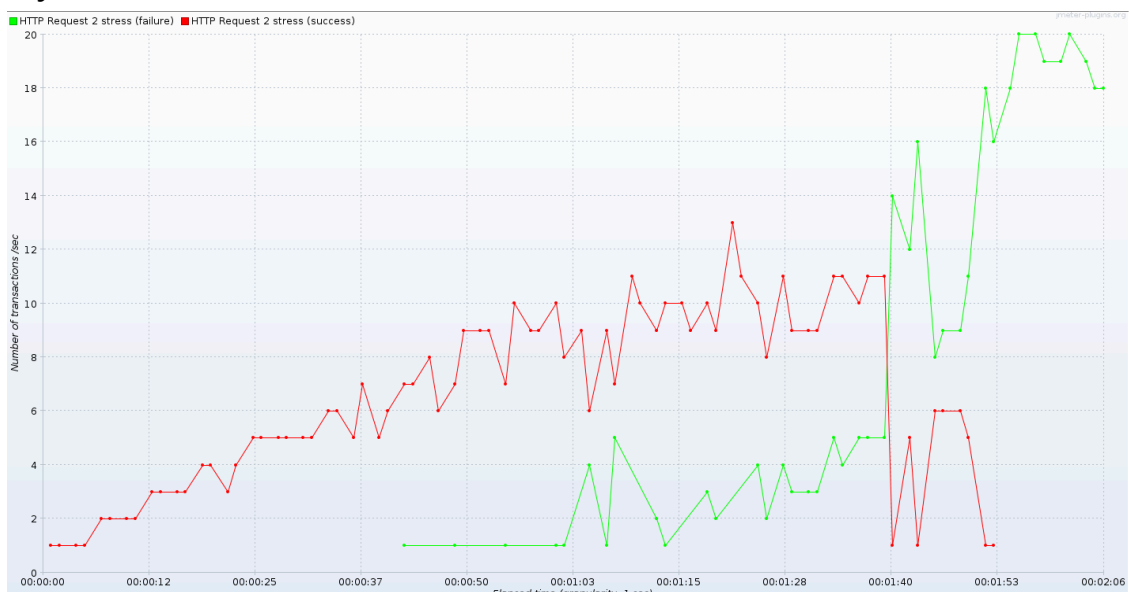
Стресс-тестирование

Как видно из графика, время ответа постепенно увеличивается. Примерно на 1:30 значение отклика пробивает отметку в 840 мс.

Т.к. новый пользователь появляется каждые 6 секунд, то на момент времени 1:30 должен был появиться 15-й по счету пользователь.



Если судить по этому графику, то количество превышений лимита в 840 мс внезапно повышается в момент времени 1:40. Это где-то 17-й по счету пользователь:



По итогу максимальное количество пользователей, которое способно выдержать конфигурация примерно равна 16. А т.к. каждый пользователь создает нагрузку в 40 запросов в минуту, то суммарное количество запросов в минуту равно 640.

Вывод

Мы провели нагрузочное и стресс тестирования, познакомились с приложением JMeter и научились с ним работать. Уверены, эти навыки пригодятся нам в дальнейшем.