



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное  
образовательное учреждение высшего образования  
“Национальный исследовательский университет ИТМО”

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ  
И КОМПЬЮТЕРНОЙ ТЕХНИКИ



## ЛАБОРАТОРНАЯ РАБОТА №4

по дисциплине  
“Тестирование программного обеспечения”

Работу выполнил:

Студент группы Р3311

Болорболд Аригуун

Лектор:

Клименков Сергей Викторович

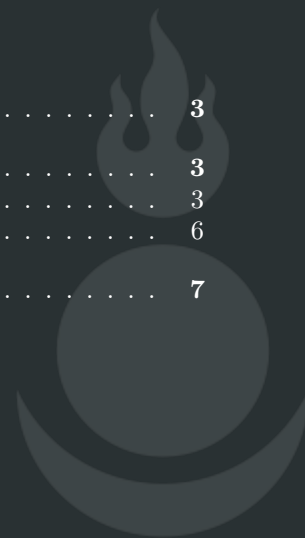
Практик:

Птицын Максим Евгеньевич

г. Санкт-Петербург  
2025 г.

## Содержимое

1 Текст задания	3
2 Выполнение	3
2.1 Порядок выполнения	3
2.2 Стресс-тестирование	6
3 Вывод	7



# 1 Текст задания

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 требованиям по максимальному времени отклика. Для этого необходимо построить график зависимости времени отклика приложения от нагрузки. Параметры тестируемого веб-приложения:

- URL первой конфигурации (\$ 3300) - `http://stload.se.ifmo.ru:8080?token=496132796&user=2104873452&config=1`;
- URL второй конфигурации (\$ 4600) - `http://stload.se.ifmo.ru:8080?token=496132796&user=2104873452&config=2`;
- URL третьей конфигурации (\$ 7000) - `http://stload.se.ifmo.ru:8080?token=496132796&user=2104873452&config=3`;
- Максимальное количество параллельных пользователей - 13;
- Средняя нагрузка, формируемая одним пользователем - 40 запр. в мин.;
- Максимально допустимое время обработки запроса - 670 мс.

## 2 Выполнение

Ссылка на репозиторию: [GitHub](#)

### 2.1 Порядок выполнения

Конфигурации (отличие только в `conf`):

Thread Group

Name:

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):

Ramp-up period (seconds):

Loop Count: ☐ Infinite

☒ Same user on each iteration

☐ Delay Thread creation until needed

☐ Specify Thread lifetime

Duration (seconds):

Startup delay (seconds):

HTTP Request

Name:

Comments:

Basic Advanced

Web Server

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

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	496132796	<input type="checkbox"/>	text/plain	<input checked="" type="checkbox"/>
user	-2104873452	<input type="checkbox"/>	text/plain	<input checked="" type="checkbox"/>
conf	1	<input type="checkbox"/>	text/plain	<input checked="" type="checkbox"/>

- Number of Threads — количество пользователей согласно заданию;
- Ramp-up period — время, за которое наращиваем нагрузку на сервис;
- Loop count — количество повторений;
- Duration in milliseconds — максимальное время ожидания согласно варианту;

Подключение:

```
ssh -p 2222 s372799@se.ifmo.ru -f -N -L 8079:stload.se.ifmo.ru:8080
```

Запуск сценария:

```
jmeter -n -t 'Test Plan.jmx' -l 'res.csv' -e -o 'Load Test Result'
```

Результат:

```
PS C:\Users\Admin\Downloads> jmeter -n -t 'Test Plan.jmx' -l 'res.csv' -e -o 'Load Test Result'
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
Creating summariser <summary>
Created the tree successfully using Test Plan.jmx
Starting standalone test @ 2025 Apr 11 11:13:30 MSK (1744359210597)
Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 4445
summary + 1 in 00:00:01 = 0.8/s Avg: 1136 Min: 1136 Max: 1136 Err: 1 (100.00%) Active: 3 Started: 3 Finished: 0
summary + 210 in 00:00:28 = 7.5/s Avg: 930 Min: 710 Max: 2460 Err: 33 (15.71%) Active: 21 Started: 21 Finished: 0
summary + 211 in 00:00:29 = 7.2/s Avg: 931 Min: 710 Max: 2460 Err: 34 (16.11%)
summary + 514 in 00:00:31 = 16.8/s Avg: 1481 Min: 821 Max: 3115 Err: 292 (56.81%) Active: 39 Started: 39 Finished: 0
summary + 725 in 00:01:00 = 12.1/s Avg: 1321 Min: 710 Max: 3115 Err: 326 (44.97%)
summary + 583 in 00:00:29 = 19.9/s Avg: 1441 Min: 897 Max: 2568 Err: 452 (77.53%) Active: 24 Started: 39 Finished: 15
summary + 1308 in 00:01:29 = 14.6/s Avg: 1375 Min: 710 Max: 3115 Err: 778 (59.48%)
summary + 248 in 00:00:34 = 7.2/s Avg: 1136 Min: 750 Max: 5240 Err: 101 (40.73%) Active: 3 Started: 39 Finished: 36
summary + 1556 in 00:02:04 = 12.6/s Avg: 1337 Min: 710 Max: 5240 Err: 879 (56.49%)
summary + 4 in 00:00:03 = 1.3/s Avg: 4381 Min: 1050 Max: 7754 Err: 3 (75.00%) Active: 0 Started: 39 Finished: 39
summary + 1560 in 00:02:07 = 12.3/s Avg: 1345 Min: 710 Max: 7754 Err: 882 (56.54%)
Tidying up ... @ 2025 Apr 11 11:15:37 MSK (1744359337488)
... end of run
```

Статистики:

Test and Report information	
Source file	"res.csv"
Start Time	"4/11/25, 11:13 AM"
End Time	"4/11/25, 11:15 AM"
Filter for display	***

APDEX (Application Performance Index)			
Apdex	T (Toleration threshold)	F (Frustration threshold)	Label
0.217	500 ms	1 sec 500 ms	Total
0.212	500 ms	1 sec 500 ms	Third Config Request
0.218	500 ms	1 sec 500 ms	Second Config Request
0.221	500 ms	1 sec 500 ms	First Config Request

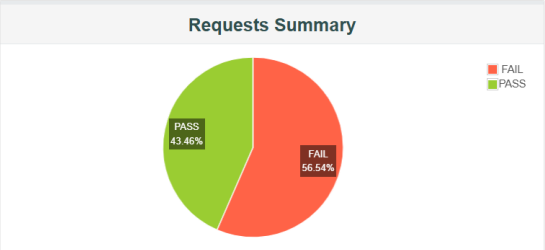
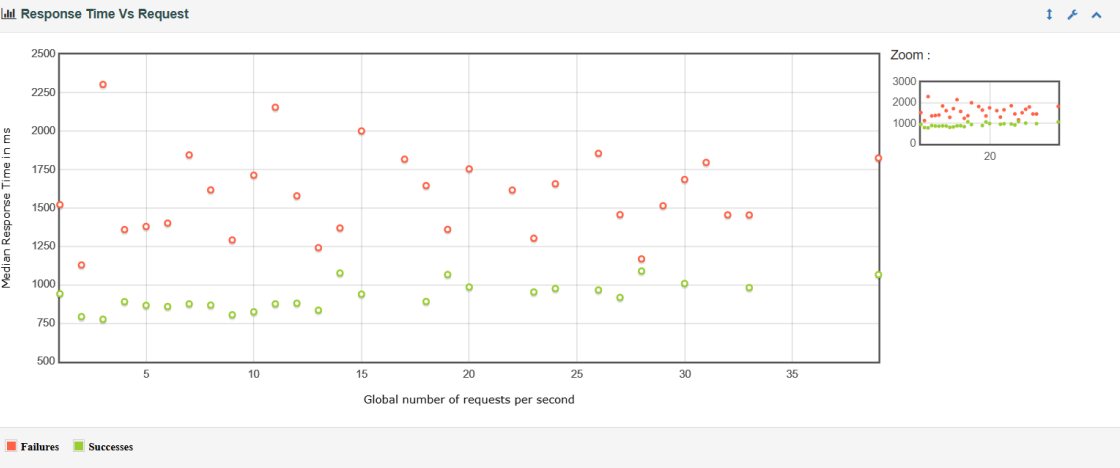


График зависимости времени отклика от числа запросов:



Процентные соотношения запросов и их времени обработки:

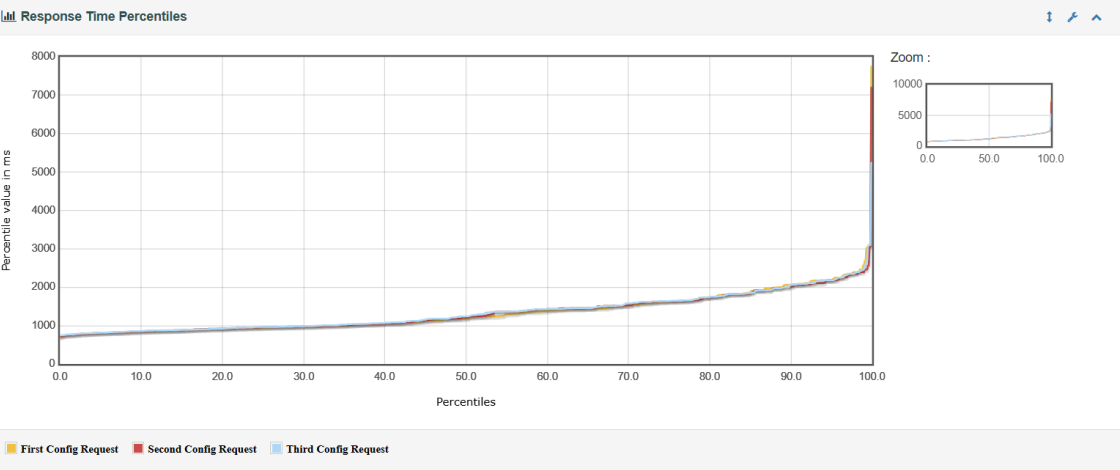
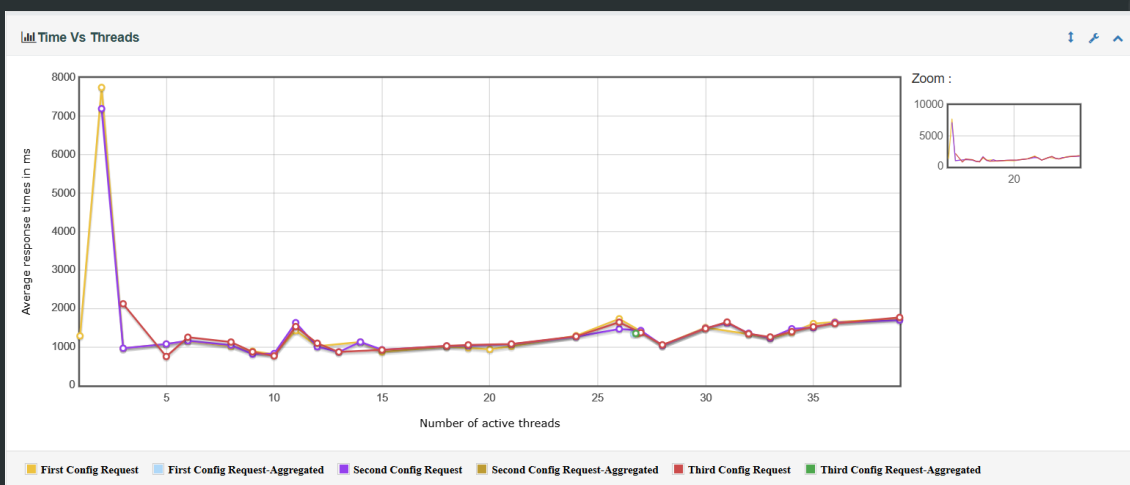


График зависимости времени отклика от числа пользовательских процессов:





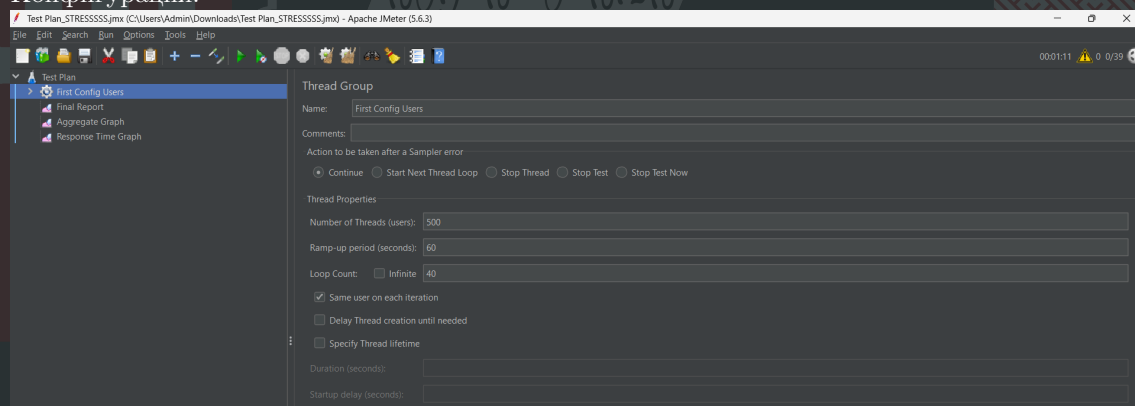
Все 3 конфига работают одинаково плохо.



Берем самый дешевый, к примеру.

## 2.2 Стресс-тестирование

Конфигурация:



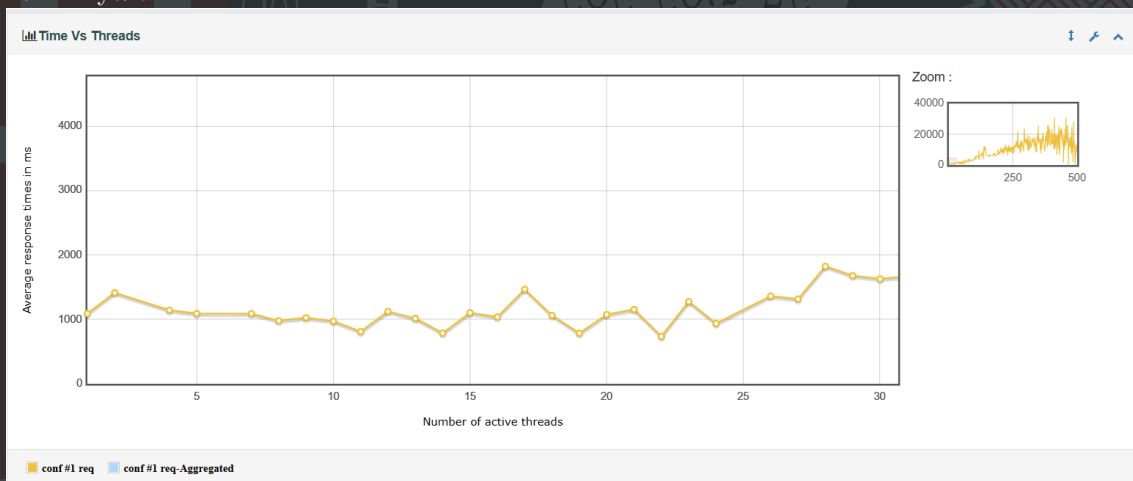
Поставил 500 пользователей. Результат:

```

PS C:\Users\Admin\Downloads> jmeter -n -t 'Test Plan_STRESSSSS.jmx' -l 'stRESS.csv' -e -o 'Stress Test Result'
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release
Creating summariser <summary>
Created the tree successfully using Test Plan_STRESSSSS.jmx
Starting standalone test @ 2025 Apr 11 11:20:29 MSK (1744359629442)
Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 4445
summary + 1 in 00:00:01 = 1.0/s Avg: 772 Min: 772 Max: 772 Err: 0 (0.00%) Active: 8 Started: 8 Finished: 0
summary + 659 in 00:00:32 = 20.4/s Avg: 4059 Min: 722 Max: 9526 Err: 536 (81.34%) Active: 278 Started: 278 Finished: 0
summary + 660 in 00:00:33 = 19.8/s Avg: 4054 Min: 722 Max: 9526 Err: 536 (81.21%)
summary + 804 in 00:00:27 = 29.6/s Avg: 10013 Min: 1 Max: 22083 Err: 804 (100.00%) Active: 500 Started: 500 Finished: 0
summary + 1464 in 00:01:00 = 24.2/s Avg: 7327 Min: 1 Max: 22083 Err: 1340 (91.53%)
summary + 1329 in 00:00:32 = 41.0/s Avg: 10171 Min: 0 Max: 42069 Err: 1329 (100.00%) Active: 500 Started: 500 Finished: 0
summary + 2793 in 00:01:33 = 30.1/s Avg: 8680 Min: 0 Max: 42069 Err: 2669 (95.56%)
summary + 1335 in 00:00:31 = 43.7/s Avg: 10338 Min: 0 Max: 37766 Err: 1335 (100.00%) Active: 500 Started: 500 Finished: 0
summary + 4128 in 00:02:04 = 33.4/s Avg: 9216 Min: 0 Max: 42069 Err: 4004 (97.00%)
summary + 1265 in 00:00:27 = 46.2/s Avg: 11543 Min: 0 Max: 37856 Err: 1265 (100.00%) Active: 500 Started: 500 Finished: 0
summary + 5393 in 00:02:31 = 35.7/s Avg: 9762 Min: 0 Max: 42069 Err: 5269 (97.70%)
summary + 1325 in 00:00:33 = 40.0/s Avg: 10591 Min: 0 Max: 37357 Err: 1325 (100.00%) Active: 500 Started: 500 Finished: 0
summary + 6718 in 00:03:04 = 36.5/s Avg: 9926 Min: 0 Max: 42069 Err: 6594 (98.15%)
summary + 1295 in 00:00:28 = 47.0/s Avg: 11083 Min: 0 Max: 38645 Err: 1295 (100.00%) Active: 500 Started: 500 Finished: 0
summary + 8013 in 00:03:32 = 37.9/s Avg: 10113 Min: 0 Max: 42069 Err: 7889 (98.45%)
summary + 1397 in 00:00:30 = 46.9/s Avg: 10253 Min: 0 Max: 38270 Err: 1397 (100.00%) Active: 500 Started: 500 Finished: 0
summary + 9410 in 00:04:01 = 39.0/s Avg: 10133 Min: 0 Max: 42069 Err: 9286 (98.68%)
summary + 1268 in 00:00:30 = 42.9/s Avg: 10221 Min: 0 Max: 37814 Err: 1268 (100.00%) Active: 497 Started: 500 Finished: 3
summary + 10678 in 00:04:31 = 39.4/s Avg: 10144 Min: 0 Max: 42069 Err: 10554 (98.84%)
summary + 1309 in 00:00:31 = 42.9/s Avg: 11787 Min: 0 Max: 36957 Err: 1309 (100.00%) Active: 494 Started: 500 Finished: 6
summary + 11987 in 00:05:01 = 39.8/s Avg: 10323 Min: 0 Max: 42069 Err: 11863 (98.97%)
summary + 1335 in 00:00:33 = 39.9/s Avg: 10295 Min: 0 Max: 37122 Err: 1335 (100.00%) Active: 491 Started: 500 Finished: 9
summary + 13322 in 00:05:35 = 39.8/s Avg: 10320 Min: 0 Max: 42069 Err: 13198 (99.07%)
summary + 1195 in 00:00:30 = 40.4/s Avg: 11931 Min: 0 Max: 37714 Err: 1195 (100.00%) Active: 482 Started: 500 Finished: 18
summary + 14517 in 00:06:04 = 39.8/s Avg: 10453 Min: 0 Max: 42069 Err: 14393 (99.15%)
summary + 972 in 00:00:31 = 31.6/s Avg: 14642 Min: 0 Max: 38235 Err: 972 (100.00%) Active: 458 Started: 500 Finished: 42
summary + 15489 in 00:06:35 = 39.2/s Avg: 10716 Min: 0 Max: 42069 Err: 15365 (99.20%)
summary + 737 in 00:00:26 = 28.8/s Avg: 15776 Min: 0 Max: 37379 Err: 737 (100.00%) Active: 442 Started: 500 Finished: 58
summary + 16226 in 00:07:01 = 38.6/s Avg: 10946 Min: 0 Max: 42069 Err: 16102 (99.24%)
summary + 807 in 00:00:31 = 26.0/s Avg: 17674 Min: 6809 Max: 36365 Err: 807 (100.00%) Active: 413 Started: 500 Finished: 87
summary + 17033 in 00:07:32 = 37.7/s Avg: 11264 Min: 0 Max: 42069 Err: 16909 (99.27%)
summary + 749 in 00:00:30 = 25.2/s Avg: 16001 Min: 7776 Max: 35943 Err: 749 (100.00%) Active: 368 Started: 500 Finished: 132
summary + 17782 in 00:08:01 = 36.9/s Avg: 11464 Min: 0 Max: 42069 Err: 17658 (99.30%)
summary + 781 in 00:00:30 = 26.4/s Avg: 15093 Min: 7756 Max: 35163 Err: 781 (100.00%) Active: 294 Started: 500 Finished: 206
summary + 18563 in 00:08:31 = 36.3/s Avg: 11617 Min: 0 Max: 42069 Err: 18439 (99.33%)
summary + 847 in 00:00:31 = 27.4/s Avg: 10563 Min: 6653 Max: 25665 Err: 847 (100.00%) Active: 176 Started: 500 Finished: 324
summary + 19410 in 00:09:02 = 35.8/s Avg: 11571 Min: 0 Max: 42069 Err: 19286 (99.36%)
summary + 590 in 00:00:23 = 25.9/s Avg: 4816 Min: 990 Max: 11935 Err: 584 (98.98%) Active: 0 Started: 500 Finished: 500
summary + 20000 in 00:09:25 = 35.4/s Avg: 11371 Min: 0 Max: 42069 Err: 19870 (99.35%)
Tidying up ... @ 2025 Apr 11 11:29:54 MSK (1744360194279)
... end of run

```

В итоге при времени отклика 1100, максимум число пользователей, при котором система удовлетворяет нашим условиям — 24.



### 3 Вывод

В рамках этой лабораторной работы я упражнял навыки работы с JMeter при соответствии работы с нагрузочным тестированием. Если всё работают на одинаковом уровне, то следует выбрать самую дешёвую из них (логика :D).