

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

«Национальный исследовательский университет ИТМО»

Факультет Программной Инженерии и Компьютерной Техники

Дисциплина: Тестирование программного обеспечения

Лабораторная работа 4

336764

Выполнили:

Гурьянов Кирилл Алексеевич

Соловьев Павел Андреевич

Группа: Р33302

Преподаватель:

Гаврилов Антон Валерьевич

Санкт-Петербург

2024

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 перестает удовлетворять требованиями по максимальному времени отклика. Для этого необходимо построить график зависимости времени отклика приложения от нагрузки.

Параметры тестируемого веб-приложения:

URL первой конфигурации (\$ 3900) -

[http://stload.se.ifmo.ru:8080?token=492557778&user=2109712915&config=1;](http://stload.se.ifmo.ru:8080?token=492557778&user=2109712915&config=1)

URL второй конфигурации (\$ 7500) -

[http://stload.se.ifmo.ru:8080?token=492557778&user=2109712915&config=2;](http://stload.se.ifmo.ru:8080?token=492557778&user=2109712915&config=2)

URL третьей конфигурации (\$ 8100) -

[http://stload.se.ifmo.ru:8080?token=492557778&user=2109712915&config=3;](http://stload.se.ifmo.ru:8080?token=492557778&user=2109712915&config=3)

Максимальное количество параллельных пользователей - **6**;

Средняя нагрузка, формируемая одним пользователем - **40** запр. в мин.;

Максимально допустимое время обработки запроса - **890** мс.

Выполнение

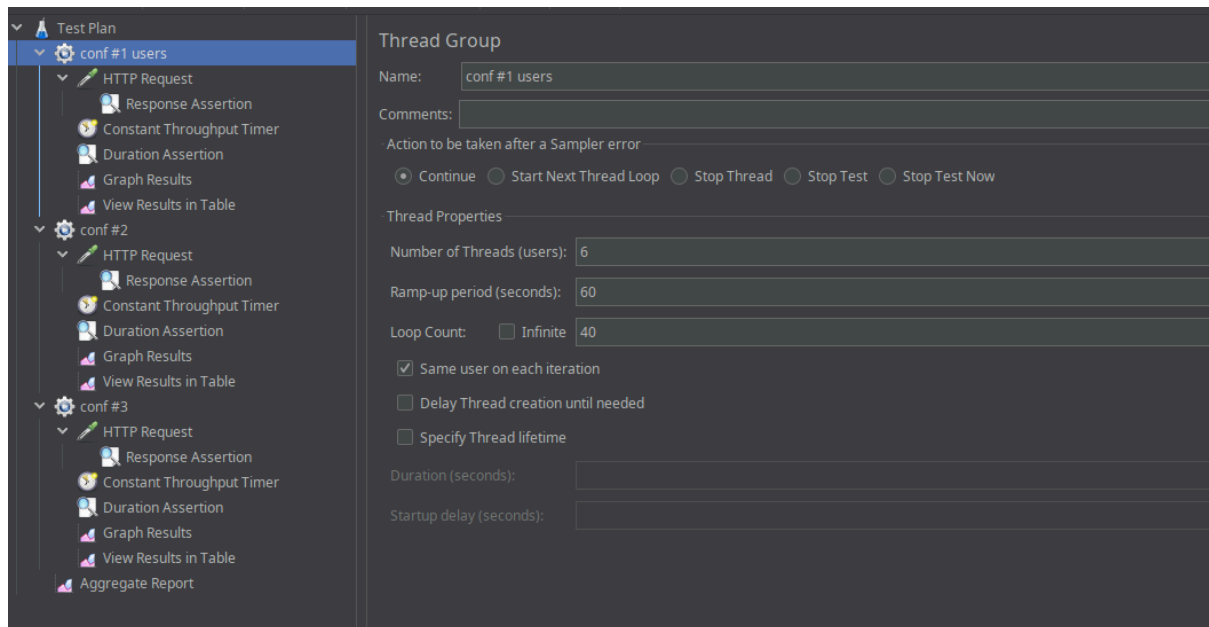
Описание конфигурации JMeter для нагрузочного тестирования.

JMХ-план тестирования представляет собой xml-файл, поэтому описание конфигурации можно привести в двух видах: скриншоты gui или xml. В качестве примера будут приведены скриншоты.

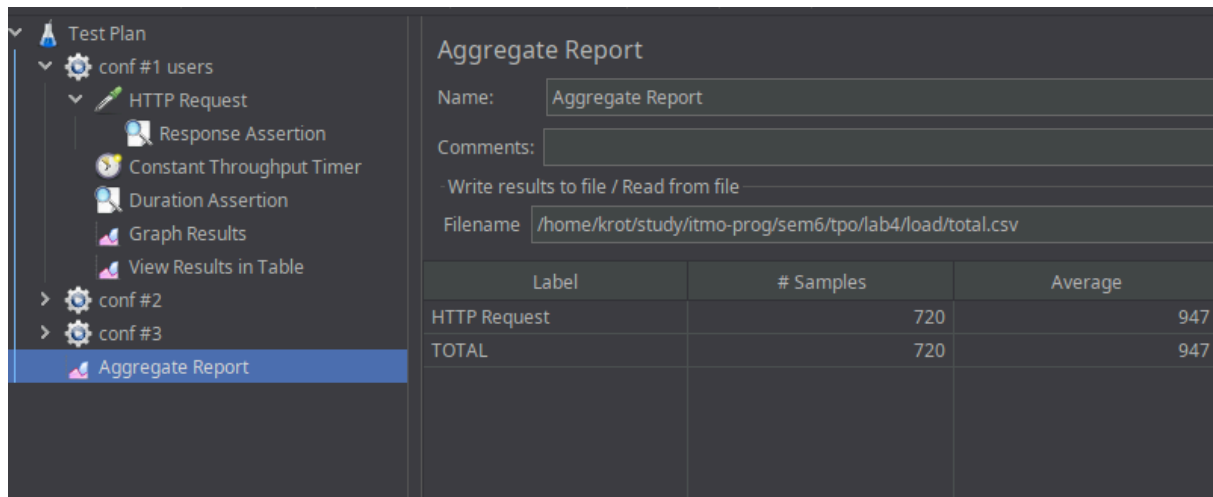
На верхнем уровне *Thread Group* объекты, описывающие количество

пользователей (Number of Threads) и количество запусков теста (Loop Count).

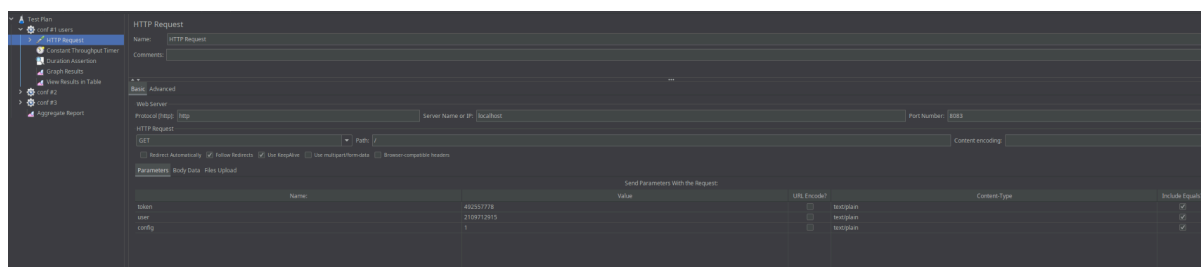
Между собой они отличаются только номером конфигурации, поэтому далее будут приведены примеры только первой группы



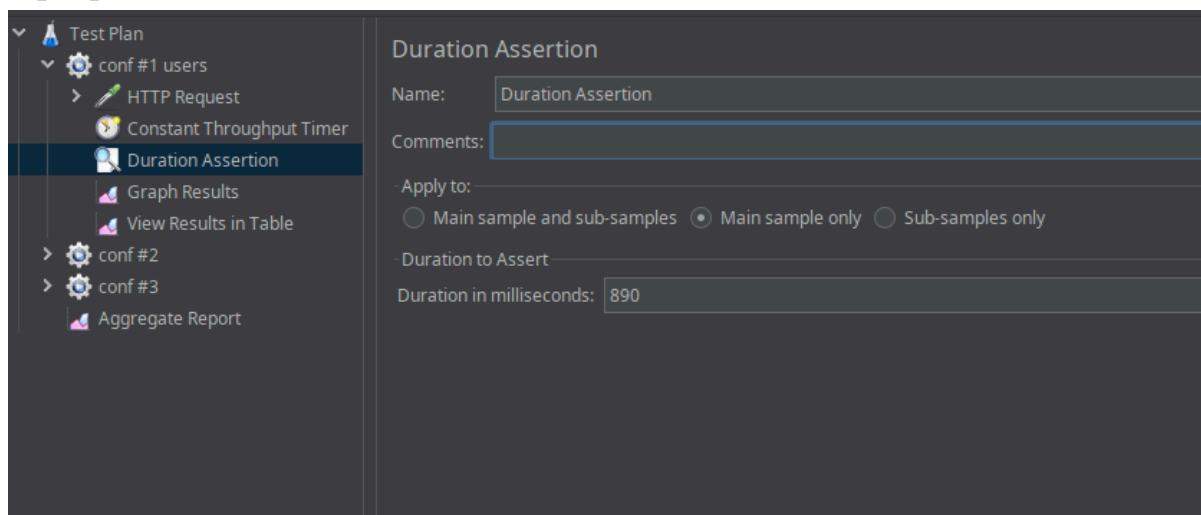
Элемент **Aggregate Report** верхнего уровня собирает результаты тестирования в csv-файл, по которому можно будет сгенерирован html-отчёт.



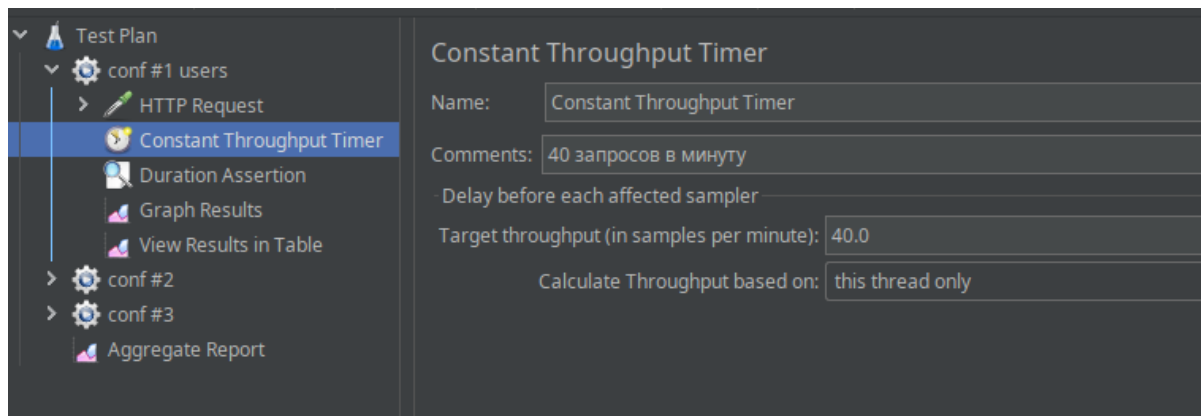
HTTP Request – осуществляет запросы к тестовому серверу.



Duration Assertion – осуществляет проверку времени отклика сервера.



Constant Throughput Timer – определяет среднюю нагрузку, формируемую одним пользователем в мин.



View Results in Table и **Graph Results** отображают результаты тестирования в виде таблицы и графика соответственно

Test Plan

conf #1 users

conf #2

conf #3

HTTP Request

Response Assertion

Constant Throughput Timer

Duration Assertion

Graph Results

View Results in Table

Aggregate Report

View Results in Table

Name: View Results in Table

Comments:

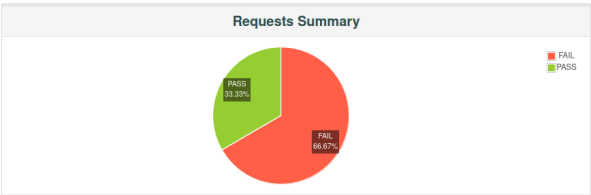
Write results to file / Read from file

Filename

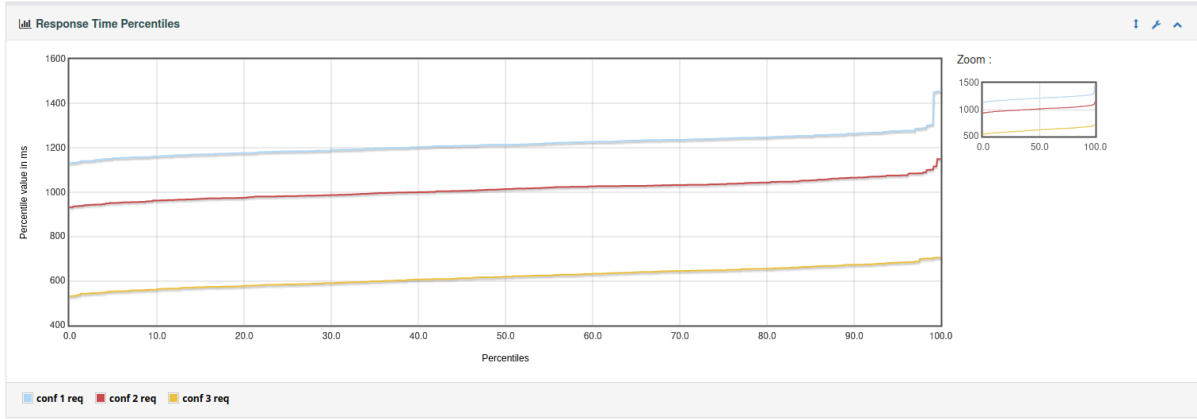
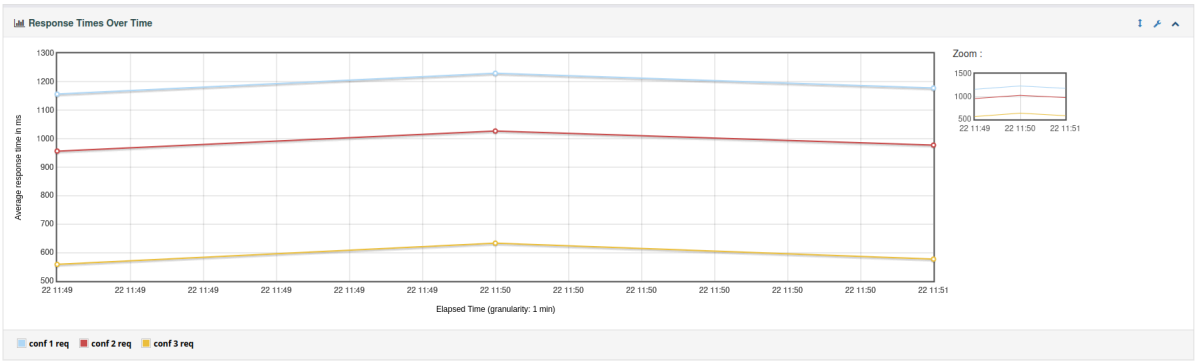
Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency
1	11:06:00.479	conf #3 3-1	HTTP Request	603	✓	231	158	603
2	11:06:01.978	conf #3 3-1	HTTP Request	534	✓	231	158	534
3	11:06:03.478	conf #3 3-1	HTTP Request	535	✓	231	158	535
4	11:06:04.978	conf #3 3-1	HTTP Request	539	✓	231	158	539
5	11:06:06.478	conf #3 3-1	HTTP Request	548	✓	231	158	548
6	11:06:07.978	conf #3 3-1	HTTP Request	547	✓	231	158	547
7	11:06:09.478	conf #3 3-1	HTTP Request	547	✓	231	158	547
8	11:06:10.476	conf #3 3-2	HTTP Request	595	✓	231	158	595
9	11:06:10.978	conf #3 3-1	HTTP Request	544	✓	231	158	544
10	11:06:11.975	conf #3 3-2	HTTP Request	551	✓	231	158	551
11	11:06:12.478	conf #3 3-1	HTTP Request	549	✓	231	158	549
12	11:06:13.475	conf #3 3-2	HTTP Request	547	✓	231	158	547
13	11:06:13.978	conf #3 3-1	HTTP Request	547	✓	231	158	547
14	11:06:14.975	conf #3 3-2	HTTP Request	558	✓	231	158	558
15	11:06:15.478	conf #3 3-1	HTTP Request	557	✓	231	158	557
16	11:06:16.475	conf #3 3-2	HTTP Request	564	✓	231	158	564
17	11:06:16.978	conf #3 3-1	HTTP Request	560	✓	231	158	560
18	11:06:17.975	conf #3 3-2	HTTP Request	562	✓	231	158	562
19	11:06:18.478	conf #3 3-1	HTTP Request	562	✓	231	158	562
20	11:06:19.474	conf #3 3-2	HTTP Request	564	✓	231	158	564
21	11:06:19.978	conf #3 3-1	HTTP Request	593	✓	231	158	593
22	11:06:20.477	conf #3 3-3	HTTP Request	727	✓	231	158	727
23	11:06:20.974	conf #3 3-2	HTTP Request	568	✓	231	158	568
24	11:06:21.478	conf #3 3-1	HTTP Request	565	✓	231	158	565
25	11:06:21.977	conf #3 3-3	HTTP Request	561	✓	231	158	561
26	11:06:22.475	conf #3 3-2	HTTP Request	564	✓	231	158	564
27	11:06:22.978	conf #3 3-1	HTTP Request	566	✓	231	158	566
28	11:06:23.477	conf #3 3-3	HTTP Request	564	✓	231	158	564
29	11:06:23.975	conf #3 3-2	HTTP Request	568	✓	231	158	568
30	11:06:24.478	conf #3 3-1	HTTP Request	570	✓	231	158	570
31	11:06:24.976	conf #3 3-3	HTTP Request	569	✓	231	158	569
32	11:06:25.475	conf #3 3-2	HTTP Request	570	✓	231	158	570
33	11:06:25.978	conf #3 3-1	HTTP Request	582	✓	231	158	582
34	11:06:26.477	conf #3 3-3	HTTP Request	585	✓	231	158	585
35	11:06:26.975	conf #3 3-2	HTTP Request	589	✓	231	158	589
36	11:06:27.478	conf #3 3-1	HTTP Request	602	✓	231	158	602
37	11:06:27.977	conf #3 3-3	HTTP Request	590	✓	231	158	590
38	11:06:28.475	conf #3 3-2	HTTP Request	593	✓	231	158	593
39	11:06:28.977	conf #3 3-1	HTTP Request	593	✓	231	158	593
40	11:06:29.477	conf #3 3-3	HTTP Request	588	✓	231	158	588
41	11:06:29.975	conf #3 3-2	HTTP Request	590	✓	231	158	590
42	11:06:30.478	conf #3 3-1	HTTP Request	589	✓	231	158	589
43	11:06:30.478	conf #3 3-4	HTTP Request	660	✓	231	158	660
44	11:06:30.977	conf #3 3-3	HTTP Request	601	✓	231	158	601
45	11:06:31.475	conf #3 3-2	HTTP Request	582	✓	231	158	582
46	11:06:31.978	conf #3 3-1	HTTP Request	583	✓	231	158	583
47	11:06:31.977	conf #3 3-4	HTTP Request	659	✓	231	158	659
48	11:06:32.478	conf #3 3-3	HTTP Request	593	✓	231	158	593
49	11:06:32.976	conf #3 3-2	HTTP Request	588	✓	231	158	588
50	11:06:33.479	conf #3 3-1	HTTP Request	585	✓	231	158	585

Графики пропускной способности приложения, полученные в ходе нагрузочного тестирования (HTML report)

APDEX (Application Performance Index)			
Apdex	T (Toleration threshold)	F (Frustration threshold)	Label
0.167	500 ms	1 sec 500 ms	Total
0.000	500 ms	1 sec 500 ms	conf 1 req
0.000	500 ms	1 sec 500 ms	conf 2 req
0.500	500 ms	1 sec 500 ms	conf 3 req



Statistics														
Requests	Executions				Response Times (ms)							Throughput	Network (KB/sec)	
Label	#Samples	FAIL	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent	
Total	720	480	66.67%	948.48	531	1455	1014.50	1237.00	1254.00	1285.11	6.57	1.48	1.01	
conf 1 req	240	240	100.00%	1215.14	1132	1455	1214.00	1264.00	1276.00	1391.73	2.19	0.49	0.34	
conf 2 req	240	240	100.00%	1012.96	933	1150	1014.50	1066.00	1076.00	1110.85	2.19	0.49	0.34	
conf 3 req	240	0	0.00%	617.33	531	705	618.50	672.90	682.00	702.77	2.20	0.50	0.34	





Выводы по выбранной конфигурации аппаратного обеспечения.

Для нагрузочного тестирования обратимся к графикам времени отклика запроса к количеству успешных откликов. На них можно увидеть, что в заданные варианты временные рамки **890** мс и **6** пользователями одновременно, может справиться лишь 3-я – самая дорогая конфигурация, а первая и вторая не подходит так как в пике достигается **752** мс. По этой же причине, наибольший процент неудачных запросов во время тестирования имеет первая и вторая конфигурации - 100%.

Описание конфигурации JMeter для стресс-тестирования.

Для стресс-тестирования была выбрана **третья** конфигурация оборудования, так как только она показала достойный результат. Duration Assertion можно убрать, так как из прошлого тестирования мы уже знаем, что и при **6** пользователях временные рамки будут соблюдены с запасом **260** мс. Установим кол-во пользователей равным **20** и ramp-up period в **30** секунд, тогда мы сможем отследить постепенное увеличение задержек при увеличении кол-ва пользователей.

Test Plan

conf #3

conf 3 req

Duration Assertion

Response Assertion

Graph Results

View Results Tree

Constant Throughput Timer

Aggregate Report

Thread Group

Name: conf #3

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): 30

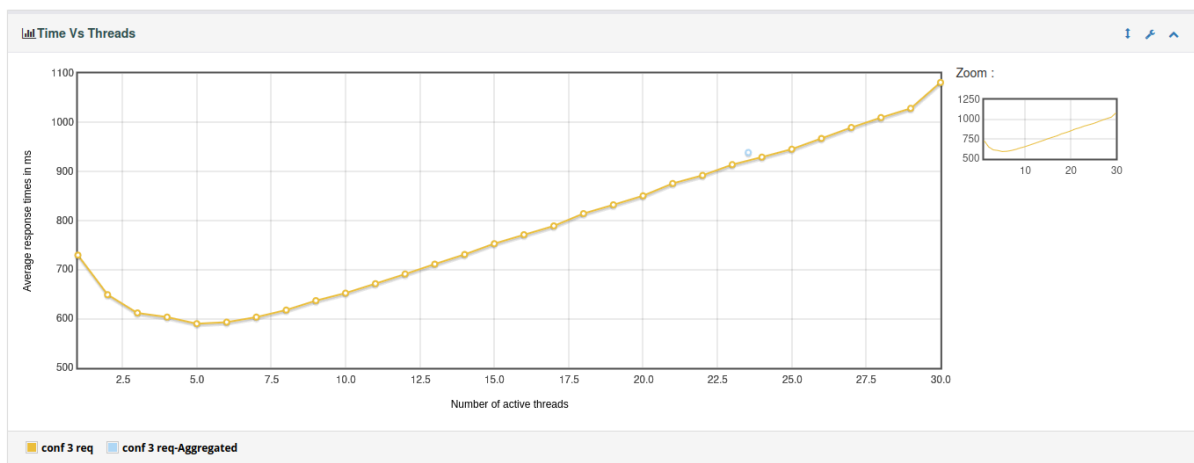
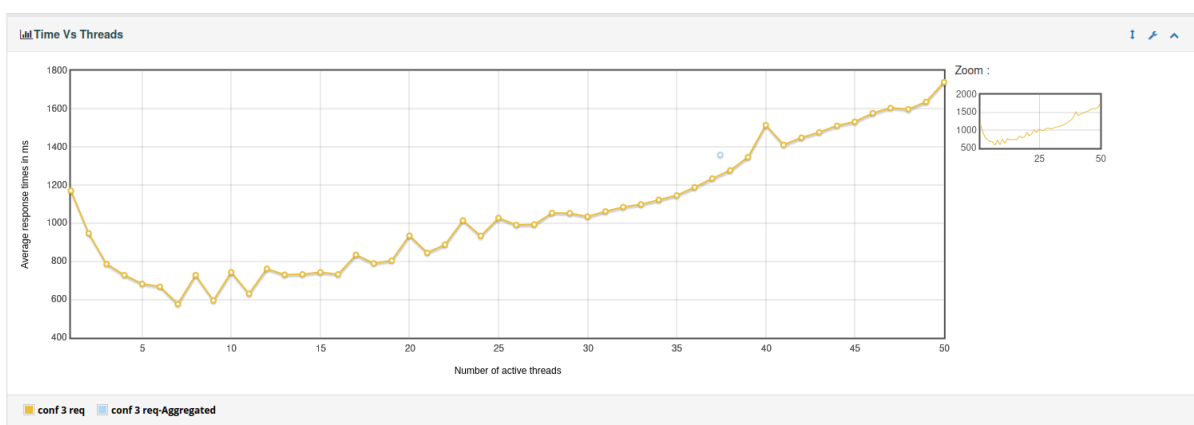
Loop Count: ☒ Infinite

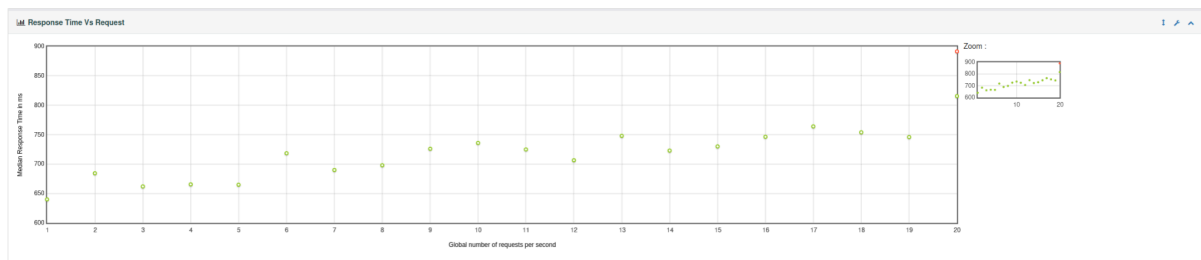
☐ Same user on each iteration
 ☐ Delay Thread creation until needed
 ☒ Specify Thread lifetime

Duration (seconds): 60

Startup delay (seconds):

График изменения времени отклика от нагрузки для выбранной конфигурации, полученный в ходе стресс-тестирования системы.





Приложение максимально выдерживает **20** пользователей, при задержке меньше **890** мс.

Выводы по работе

В ходе выполнения лабораторной работы были проведены нагрузочное и стресс тестирования сервером, с целью выявления самой дешевой конфигурации, удовлетворяющей требованиям. JMeter - удобный и гибкий инструмент для проведения нагрузочного тестирования. В результате выполнения лабораторной работы была выбрана третья конфигурация, удовлетворяющая требованиям и при этом имеющая наименьшую стоимость. В ходе стресс-тестирования было определено, при какой нагрузке выбранная на предыдущем шаге конфигурация — перестаёт удовлетворять требованиям по максимальному времени отклика. Для этого был построен график зависимости времени отклика приложения от нагрузки.