

PREPERFORMANCE TESTING OF A WEBSITE

Sushma Suryadevara and Shahid Ali

Department of Information Technology, AGI Institute, Auckland, New Zealand

ABSTRACT

This study was conducted on the importance of performance testing of web applications and analyzing the bottleneck applications. This paper highlights performance testing based on load tests. Everyone wants the application to be very fast, at the same time, reliability of the application also plays an important role, such that user's satisfaction is the push for performance testing of a given application. Performance testing determines a few aspects of system performance under the pre-defined workload. In this study JMeter performance testing tool was used to implement and execute the test cases. The first load test was calculated with 200 users which was increased to 500 users and their throughput, median, average response time and deviation were calculated.

KEYWORDS

Performance testing, load balancing, threads, throughput, JMeter, load test

1. INTRODUCTION

Flipmind company started in 2005 by Michael Jones Owner and Director of the company with a dream helping clients. Flipmind creates software to help the business succeed in E-commerce Consulting & Development, Website Design, Mobile Development, and Performance Analysis & Optimization, etc. The company had delivered a lot of projects in New Zealand such as Forza, Park & Fly, Money Place, My Ride, etc. The company has worked with Torpedo7 since 2006 providing BA, systems architecture, web & e-commerce development & load testing services, etc. Flipmind built the eCommerce platform powering Torpedo7 from the ground up. In New Zealand and Australia, Torpedo7 sells thousands of products and you would be hard-pressed to find a cyclist that does not buy from them. Flipmind company started in 2005 by Michael Jones Owner and Director of the company with a dream helping clients. Flipmind creates software to help the business succeed in E-commerce Consulting & Development, Website Design, Mobile Development, and Performance Analysis & Optimization, etc. The company had delivered a lot of projects in New Zealand such as Forza, Park & Fly, Money Place, My Ride, etc. The company has worked with Torpedo7 since 2006 providing BA, systems architecture, web & e-commerce development & load testing services, etc. Flipmind built the eCommerce platform powering Torpedo7 from the ground up. In New Zealand and Australia, Torpedo7 sells thousands of products and you would be hard-pressed to find a cyclist that does not buy from them. Flipmind is one of the most rapidly growing company in New Zealand as it offers high-class service such as website development, testing and its maintenance updating the data regularly to its stakeholders. Currently working on a project named as Torpedo7 website. Torpedo7 is the most popular outdoor store in New Zealand. We can browse our huge selection of snow and water gear, bikes comfortable outdoor clothing for women, men, and kids, camping equipment, hiking gear, bags & packs of all sizes, trampolines, and outdoor technology use in your next adventure. Nowadays, for every web application, everyone wants everything to be fast and at the same time, there is concern about the reliability of usage. Most of the users want to load their webpage

rapidly fast so that they can finish and move to another work. If a page took a long time to load, the users will end up the task. The economic growth of any organization depends on the web application. If the application is not running fast, people lost interest in that application. So, it is important to do the performance testing of a web application for any organization. The load testing is done to test the performance of a website. By doing this performance testing we can know the system behaviour while handling specific load given by the customer to the system [4].

The main challenge for this project was to conduct the is load test to torpedo7.co.nz website for Black Friday and Christmas to make sure the website does not crash on those days. Load testing before Black Friday and Christmas or on any big event is very important for companies. The performance testing helps to fix bottlenecks, errors, and bugs due to high traffic spikes on the website. The cost of failure is the direct loss of sales and money to retailers. Using load testing failure can be easily avoided.

The scope of my project is to do a load test of the Torpedo7 website using JMeter for the Homepage, Shop feature(Snow, Technology, Clothing, Water, Bike), Service feature (Bike workshop), Search functionality, adding the item to the cart and proceed to checkout with 200, 500 users hence compared and analysed the results using summary report, graphs, CSV file and HTML report. The core of this objective is to make sure that the current application is optimal for handling increased traffic on the website. Hence to analyze and measure the performance of the Toredpo7 website we will be using the JMeter tool.

This research is organized as follow: Section 2 focuses on the literature review of various studies concentrating on automation regression. Section 3 is focused on the research methodology for this research. Section 4 of this research is focused on research execution results. Discussion to results of this research are provided in section 5. In section 6 conclusion to the research is provided. Finally, section 7 is dedicated towards the future work recommendations.

2. LITERATURE REVIEW

Many studies have been conducted in the past in favour of performance and load testing, we will investigate those studies, respectively.

A research was conducted on the Importance of performance testing of web applications and analysing the bottleneck applications [4]. This study highlights the performance testing based on the load test. Everyone wants the application to be very fast, at the same time, reliability of the application also plays an important role, such that user's satisfaction is the push for performance testing of a given application. Performance testing determines a few aspects of system performance under the pre-defined workload. Performance testing is measured when the business gets peak by its hits. Another research was conducted on different tools to do performance testing [7]. Apache JMeter is a free java application performance testing tool. It has a lot of plugins to aid the testing tools. Performance testing using JMeter is a type of testing to determine the responsiveness, throughput, interoperability, reliability, and scalability of an

application under a given workload. In today's competitive world it has become critical to the organizations to test their web application [8]. Load testing is the process of subjecting a whole system to a work level approaching its limits. Load testing is done to determine the behaviour of the system under normal conditions and peak load conditions. The objective of load testing is to determine the maximum capacity of an application. Another research was done on web services used mostly in all aspects of social life [9]. For web applications, performance testing is gaining wide attention. This study highlights first we analyse and research the types, and methods of performance testing of the web and later we do some testing process methods.

A study was conducted that the performance tool was used to test web applications. This tool is used for performance, load and stress testing of web applications or websites [10].

Another research was conducted on web application performance testing which plays an important role in providing Quality of service [11]. This study highlights the performance testing of web applications using a reactive based framework which helps in reducing the cost and increases the efficiency of performance testing.

A performance testing framework for a rest-based web application was proposed [12]. This framework aims to provide software testers with an integrated process from test case design, test scripts, and test execution. Another research was conducted on ajax based web applications [13] which have gained more popularity since it brings the richness of desktop applications.

A research was conducted regarding challenges and experiences to identify a good solution for conducting performance testing on web applications [14]. There was another study conducted which described three open-source tools and compared their performance, usability and software requirements [12].

Testing web applications is nothing but finding errors [12]. This study highlights different performance testing tools and tried with JMeter to improve the performance of website or web application. There was another research conducted which provides the comparison of load testing tools (Sharma,2016). In this study the main load test tools available in the market and their advantages were discussed.

A study was conducted on performance testing concepts and comparative analysis of web applications [18]. The main objective of performance testing is not to identify bugs but to eliminate performance bottlenecks.

Performance testing and load testing are some of the means to evaluate web application performance [19]. This study highlighted the load test of web applications with JMeter using blaze meter in cloud-based load testing. Another study was conducted in comparison to tools [20]. In this research comparison of three different tools was compared with respect to response times.

All these studies highlight different aspects of performance testing. However, I believe that performance testing is very important for the success of the project. Hence in this project will perform performance testing on the torpedo7 website.

3. RESEARCH METHODOLOGY

Research methodology for performance testing of Torpedo website project has been discussed below.

3.1 Comparative Analysis of Performance Tools

Table 1 below shows the comparative analysis of performance tools. After doing comparative analysis selected JMeter as the best tool for my project. We have also selected JMeter since its open-source tool to use and have hands-on experience on JMeter.

CRITERIA	JMeter	LoadRunner	NeoLoad
Commercial License	Open Source Tool	Yes	Yes
Cost	Free	Expensive	Moderate
Launched By	Apache Foundations	HP	Neoload
Result Reporting	Minimum Support (Need additional plugins for graphs)	Maximum Support	Maximum Support
Browser Support	Support all browsers	Support all browsers	Support all browsers
Load Generation	Unlimited load GENERATION	Depending on the type of license	Depending on the type of license
Devise OS Support	Supports Android, Linux, Windows, Mac	Does not support Mac	Does not support Mac

Table 1:Comparitive analysis of performance tools

3.2.Selected Tools for Project

3.2.1 Apache JMeter

Apache JMeter is a most popular and open-source tool for testing the web applications which is best suited for the companies as they do not have to pay for it. It is 100% pure java application which is designed to load test functional behavior and measure performance of the application. It is specially designed for testing Web applications. Apache JMeter is used to test both on static and dynamic resources performance. It can be used to simulate a heavy load on the server to analyze overall performance under different load types. We can use to make a graphical analysis of performance test scripts under heavy load [3].

3.2.2 Blazemeter

Blazemeter is a Chrome extension that enables you to record, browse, upload and run. Using Blazemeter we can create proper test scripts and load scenarios. Blazemeter enables us to write test scripts using JMeter and user experience test scripts using Selenium. All we need is to write the test-scripts in Blazemeter, choose the number of load-engines and run the test. The system takes care of everything else. An unlimited number of load-engines are preconfigured and available at our disposal. Detailed graphical reports are generated during the load [2].

3.3.Selected Methodology for Project

A project run by Flipmind company works on the agile methodology for the development of web applications. In the company, the project is divided into sprints and each sprint runs for 1 week. Daily stand-up meetings and discussions will be held in the company to give a brief description of project progress and about the issues facing [5].

3.3.1 Workflow of JMeter

Figure 1 shows the workflow of JMeter. When we start to perform the test of an application, JMeter creates requests to target servers and simulates the number of users by sending requests to the target server. Once the server starts responding to the requests JMeter saves all the responses. By using the response JMeter collects data to calculate statistical information. Finally, based on this information JMeter prepares a report about the performance of the Application under test.

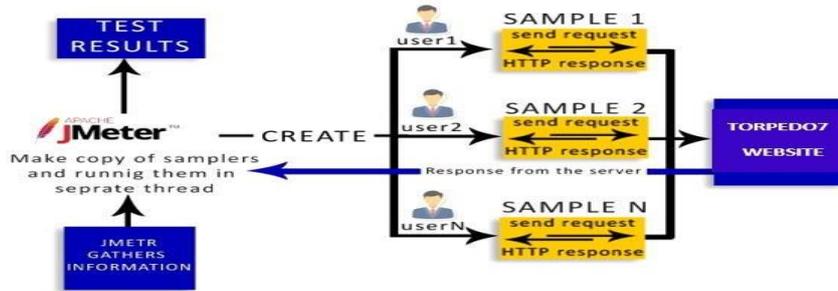


Figure 1: JMeter Workflow for Project

4 PROJECT EXECUTION

4.1 Test Plan

Table 2 below shows the test plan for my project. Table 2 is more focused on the test plan for the project of the Torpedo7 website and how I generated reports with different users.

Document	Performance Test Plan
Project	Torpedo7 website
Version	1.0.0
Objective	The objective of this document is to outline the environment and Performance test Plan for the Torpedo7 website and to ensure if the current system architecture is optimal for handling increased traffic on the website and also, to figure out if there are any upgrades that need to be done.
Scope	<ul style="list-style-type: none"> • Load test the Homepage, Shop feature (Snow, Technology, Clothing, Water, Bike), Service feature (Bike workshop), Search functionality, adding the item to the cart and proceed to checkout functions of Torpedo7 website for 200 users and 500 users • Analyze test results. • Identify bottlenecks. • Propose suggestions.
Out of Scope	<ul style="list-style-type: none"> • Functional or accuracy testing of the Torpedo7 website. • Browser or software compatibility testing. • Any other testing types not included in the Scope section.
Hardware	Load Generator – Dell Intel Core i7 -3.40 GHz., 16GB RAM

<p>Software Operating System – Windows 7 Professional</p>
<p>Performance Testing Tool Apache JMeter 5.1.1</p>
<p>Environment</p> <ul style="list-style-type: none"> • Java Runtime Environment (JRE) is installed to run JMeter. • JMeter is installed to create virtual users and do load testing. • jpgc – standard set plugin is added to JMeter to generate more graphs. • Chrome extension, Blaze Meter is added to chrome to record scripts. • A good internet connection is Confirmed.
<p>Risks Torpedo7 is a live website. At the time of performance testing, it is not isolated. The load JMeter applying is not the only load for the web site. Also, at the same time, there will be a website getting requests from the outside world. So, the report analysis might be not very correct.</p>
<p>Approach and Execution Strategy Performance Test Script Steps Performance Test scripts will be created using Blaze Meter. Those scripts will be developed to simulate the actions mentioned below.</p> <ul style="list-style-type: none"> • Torpedo7 website home page • Shop feature • Service feature • Search • Add to Cart • Checkout
<p>Performance Test Data Planning and Preparation</p> <ul style="list-style-type: none"> • Dynamic search data will be provided, and few searches will be done in one iteration. • Parameters are added to search data. <p>Performance Test Scenario</p> <ul style="list-style-type: none"> • First, the Load Test will be performed with 200 threads with 3000 seconds ramp-up period for 3000 seconds of duration. • Then the threads are increased to 500 with the same 3000 seconds ramp-up period for 3000 seconds of duration.

<p>Performance Test Reports and Metrics</p> <ul style="list-style-type: none"> Built-in Listeners in JMeter are used to create Summary Reports, Graphs and Tables. jp@gc – Standard Set plug-in will be added to the Test Plan to get more reports. Metrics like Response Time, Deviation, Throughput, Min, Max, Error%, Latency will be tracked using these reports. JMeter consumes less memory in Non-GUI mode which helps to generate CSV file and HTML Report in Windows <p>Execution and Analysis</p> <ul style="list-style-type: none"> Load Test will be performed with 200,500 users. Upgrades that need to be done will be identified.
<p>Deliverables</p> <ul style="list-style-type: none"> Test Plan – This project report document Test Results – Raw data captured from Performance Test execution Final Test Report – Test Metrics with findings and suggestions.
<p>Exit and Entry Criteria Entry Criteria: Functionally stable application. Data setup for the transactions. Exit Criteria: All the Performance objectives are met.</p>

Table 2:Test plan for the Project

4.2 Performance Test Cases

The performance test cases for this project are shown in table 3. Table 3 test cases 2,13,14,19 and 20 are the main features of this torpedo7 website.

Test Case ID	Step Description	Expected Output	Transaction Name
TC_001	1. Open the Chrome browser 2. Enter the URL 3. Enter the valid Email and Password 4. Click the sign-in button	User should be landed on the Torpedo7 Login Page and should be logged in successfully	User login
TC_002	User click on Shop icon and navigate to snow page	Snow page is displayed	Snow page
TC_003	User click on Snowboarding and navigate to snowboarding page	Snowboarding page is displayed	Snow Boarding page
TC_004	User click on Shop icon and navigate to technology page	Technology page is displayed	Technology page
TC_005	User click on phone and navigate to the phone page	Phone page is displayed	Phone page
TC_006	User click on Shop and navigate to clothing and footwear	Clothing page is displayed	Clothing page
TC_007	User click on women’s and navigate to women’s page	Women page is displayed	Women page
TC_008	User click on Shop icon and navigate to the water	Water page is displayed	Water page

TC_009	User click on surfing in water page and navigate to surfing page	Surfing page is displayed	Surfing page
TC_010	User click on Shop icon and navigate to the Bike feature page	Bike page is displayed	Bike page
TC_011	User click on MTB-Hardtail and navigate to Hardtail mountain bike page	Hardtail mountain bike page is displayed	Hardtail page
TC_012	User click on Shop icon and navigate to Bike & Frames feature	Bikes & Frames features page is displayed	Bikes & Frames page
TC_013	User click on service icon and navigate to Bike workshop page	Workshop page is displayed	Workshop page
TC_014	User click on the Search button and search for shoes	Shoe page is displayed	Shoe page
TC_015	User click on the Search button and search for socks	Socks page is displayed	Socks page
TC_016	User click on the Search button and search for jackets	Jackets page is displayed	Jacket page
TC_017	User click on the Search button and search for boats	Boat page is displayed	Boat page
TC_018	User click on the Search button and search for bike	Bike page is displayed	Bike page
TC_019	Click on add to cart for the item user selected	Item has been added to cart	Add to cart
TC_020	Click on proceed to checkout	My details page should be displayed	Checkout
TC_021	The user enters the email id and clicks on the next page	Delivery and payment details page should be displayed	Checkout
TC_022	User fills the details first name, last name, phone number, and clicks on continue to delivery	Delivery options page should be displayed	Checkout
TC_023	User fills the address details and clicks on continue to payment	Payment details page should be displayed	Checkout
TC_024	User enter the credit card number, expiry date, name on card, CVC and click on complete purchase	Your credit card number is invalid	Checkout

Table 3:Performance Test Cases

4.3 Test Scripts

For accurate performance testing results, the test environment should be similar to the production environment. To achieve that, the whole application infrastructure should be thoroughly analyzed. Torpedo7 website which is subjected to the performance testing is already in the production environment since it is a live website. To avoid the external heavy load as much as possible, it is better to conduct the performance test after 10 p.m. as there will be fewer users accessing the Torpedo7 website at that time. Performance testing is conducted using JMeter to check the behavior of the Torpedo7 website on different load conditions. First, the performance test is conducted for 200 users with 3000 seconds ramp-up period for 3000 seconds.

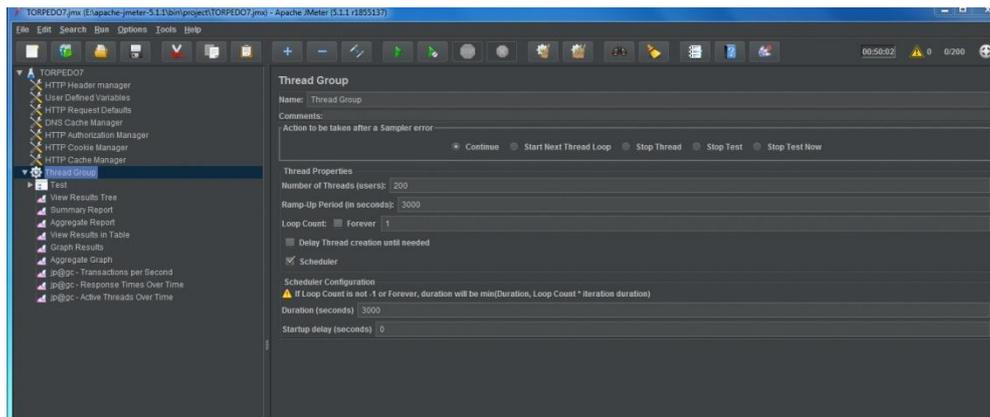


Figure 2: Test Scenario 1

Secondly, users were increased up to 500 and performance testing is conducted with 3000 seconds ramp-up period for 3000 seconds.

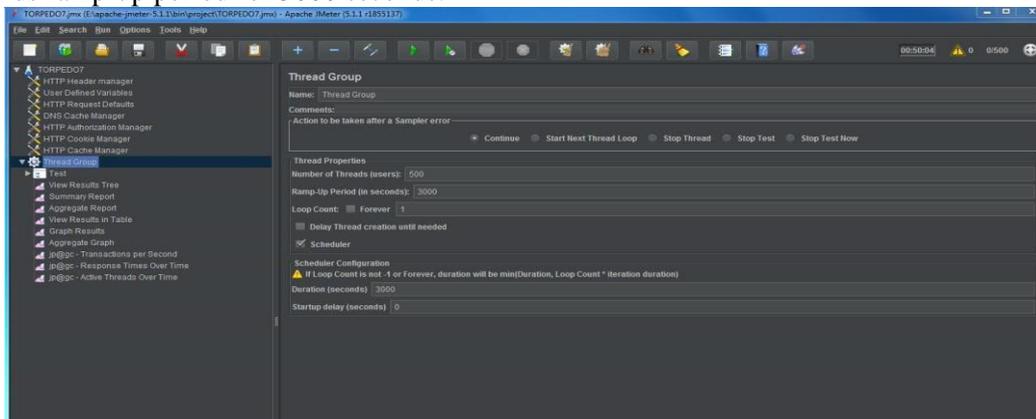


Figure 3: Test Scenario 2

Test Plan named TORPEDO7 was created to test the performance of the www.torpedo7.co.nz website using JMeter. Test scripts were recorded using Blaze Meter in the Chrome browser and exported to JMeter as JMX file.

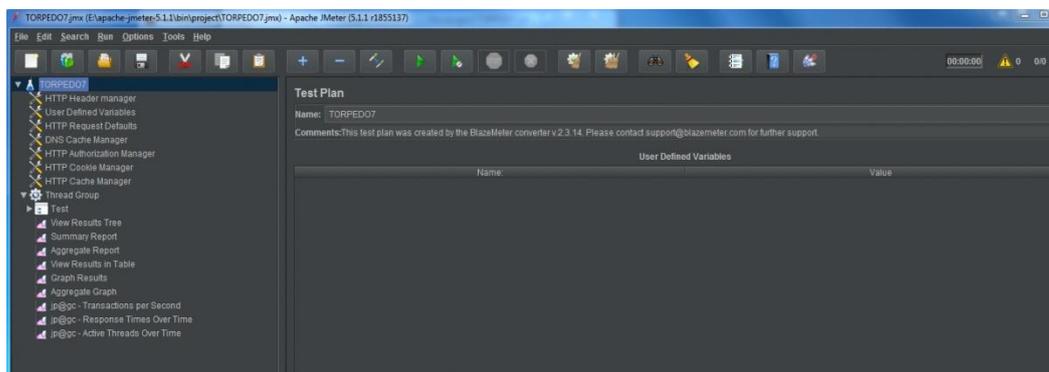


Figure 4: Test Plan

There are few Configuration Elements added to the Test Plan. Some of them are listed below:

1. HTTP Header Manager is added to the test plan at the Thread Group level which carries header information of requests.

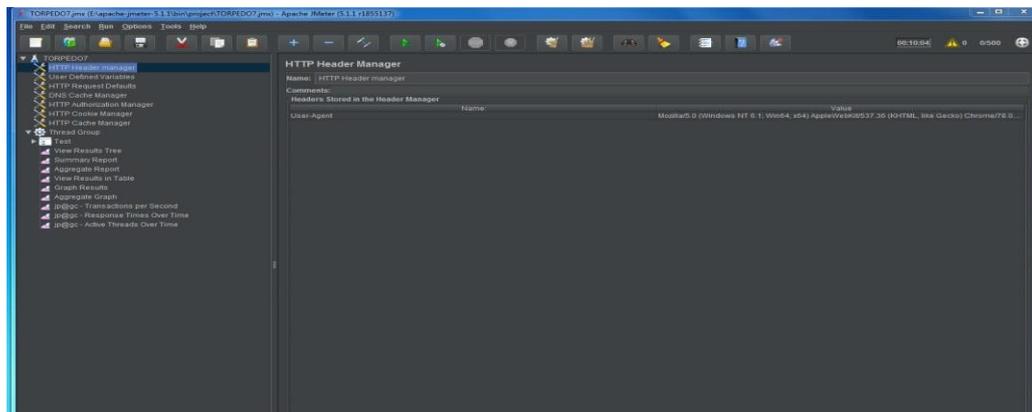


Figure 5: HTTP Header Manager

2. User-Defined Variables have been added to the Thread Group.

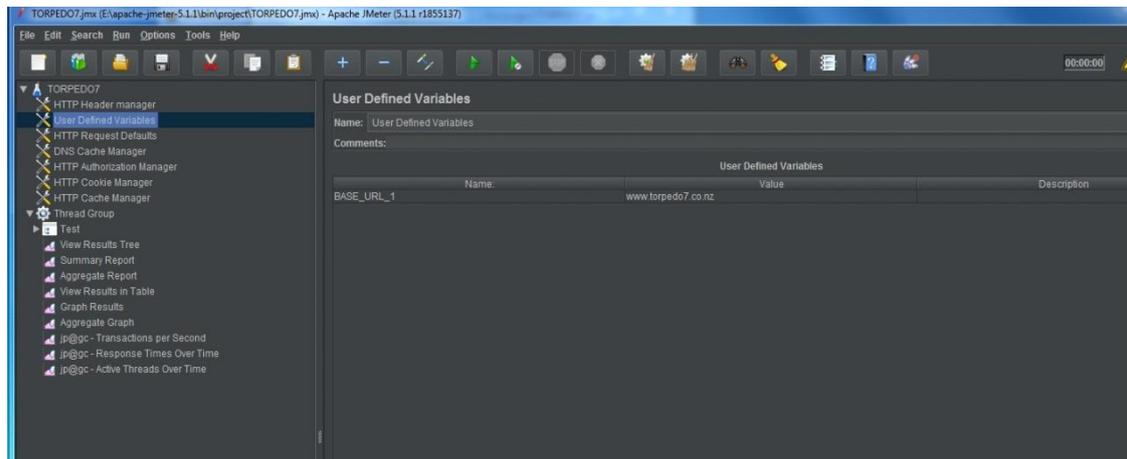


Figure 6: User-Defined Variables

3. HTTP Request Defaults have been added at the Thread Group level. Base URL is given there which is passed to every request.

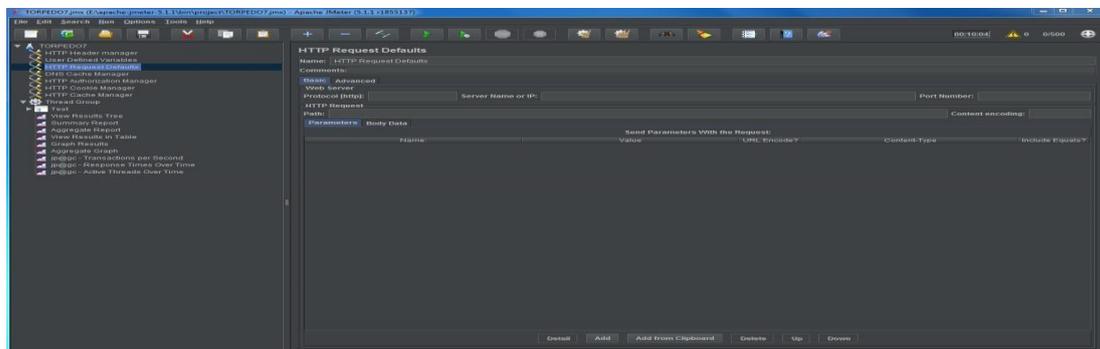


Figure 7: HTTP Request Defaults

4. DNS Cache Manager is added to the test plan at the Thread Group level. Clear cache each iteration option is selected to clear the DNS cache after each iteration.

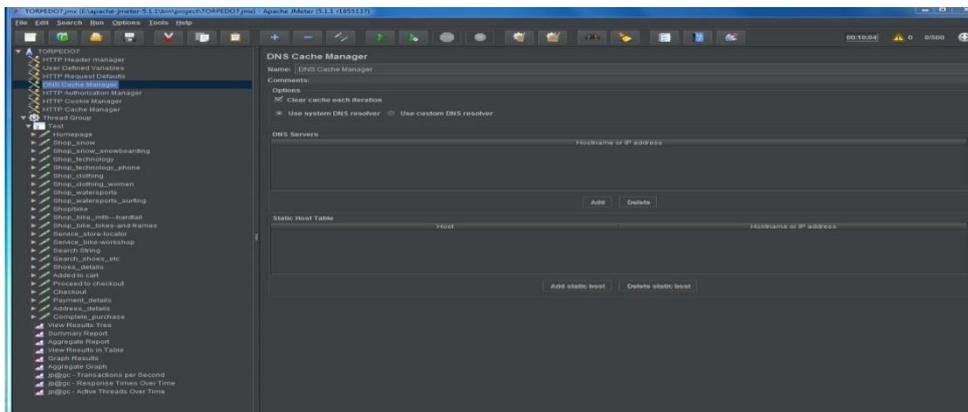


Figure 8: DNS Cache Manager

5. HTTP Cookie Manager is added at the Thread Group level. Clear cookies each iteration option is selected to clear HTTP Cookie data after each iteration.

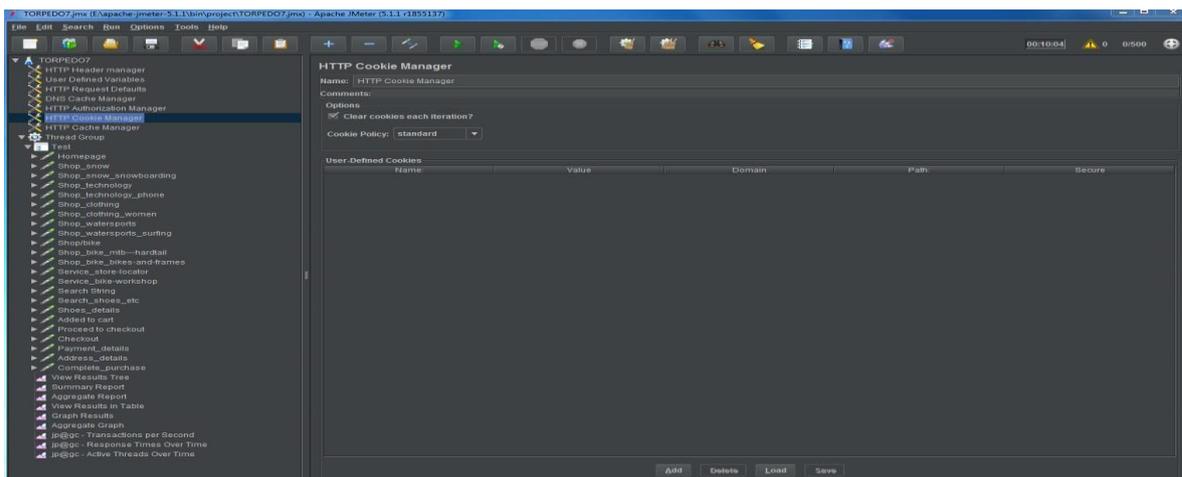


Figure 9: HTTP Cookie Manager

6. HTTP Cache Manager is added to the test plan at the Thread Group level. Clear cache each iteration option is selected to clear HTTP cache after each iteration and a maximum number of elements in the cache is set to 5000.

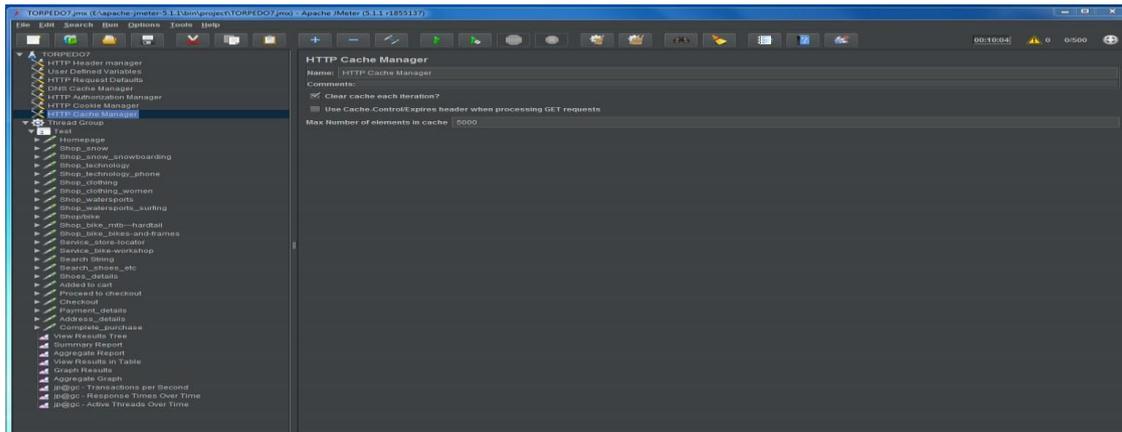


Figure 10: HTTP Cache Manager

Adding Parameters: Added parameters in Search String like shoes, socks, jackets, boats, bike

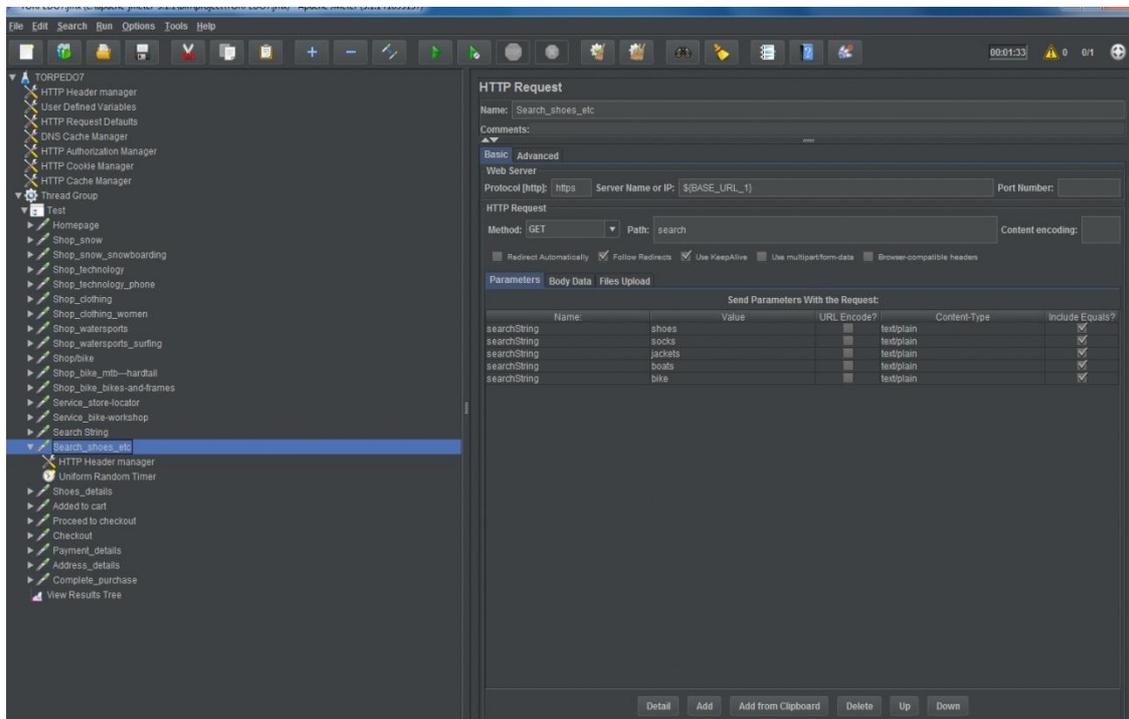


Figure 11: Adding Parameters

Generation of CSV file and HTML report: Using this command prompt generated CSV file for 200 users.

```

Administrator: Command Prompt - jmeter -n -t E:/apache-jmeter-5.1.1/bin/project/TORPEDO7.jm...
C:\Users\administrator>cd..
C:\Users>cd..
C:\>E:
E:\>apache-jmeter-5.1.1
'apache-jmeter-5.1.1' is not recognized as an internal or external command,
operable program or batch file.
E:\>cd apache-jmeter-5.1.1
E:\apache-jmeter-5.1.1>cd bin
E:\apache-jmeter-5.1.1\bin>jmeter -n -t E:/apache-jmeter-5.1.1/bin/project/TORPE
DO7.jmx -l E:/apache-jmeter-5.1.1/bin/project/200.csv -e -o E:/apache-jmeter-5.1
.1/bin/project/200
Creating summariser <summary>
Created the tree successfully using E:/apache-jmeter-5.1.1/bin/project/TORPEDO7.
jmx
Starting the test @ Wed Nov 27 13:36:25 NZDT 2019 <1574814985946>
Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 44
45

```

Figure 12: Command Prompt for 200 users

Generation of CSV file and HTML report: Using this command prompt generated CSV file for 500 users.

```

Administrator: Command Prompt - jmeter -n -t E:/apache-jmeter-5.1.1/bin/project/TORPEDO7.jm...
E:\>cd apache-jmeter-5.1.1
E:\apache-jmeter-5.1.1>cd bin
E:\apache-jmeter-5.1.1\bin>jmeter -n -t E:/apache-jmeter-5.1.1/bin/project/TORPE
DO7.jmx -l E:/apache-jmeter-5.1.1/bin/project/500.csv -e -o E:/apache-jmeter-5.1
.1/bin/project/500
Creating summariser <summary>
Created the tree successfully using E:/apache-jmeter-5.1.1/bin/project/TORPEDO7.
jmx
Starting the test @ Wed Nov 27 12:54:02 NZDT 2019 <1574812442445>
Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 44
45
summary +      1 in 00:00:01 =    0.7/s Avg:   1119 Min:   1119 Max:   1119 Err:
0 (0.00%) Active: 1 Started: 1 Finished: 0
summary +     16 in 00:00:27 =    0.6/s Avg:    511 Min:   323 Max:   1010 Err:
0 (0.00%) Active: 5 Started: 5 Finished: 0
summary =     17 in 00:00:29 =    0.6/s Avg:    547 Min:   323 Max:   1119 Err:
0 (0.00%)
summary +     54 in 00:00:28 =    1.9/s Avg:    606 Min:   245 Max:   1446 Err:
0 (0.00%) Active: 10 Started: 10 Finished: 0
summary =     71 in 00:00:57 =    1.2/s Avg:    592 Min:   245 Max:   1446 Err:
0 (0.00%)

```

Figure 13: Command Prompt for 500 users

To create CSV file and HTML report in NON-GUI Mode we use the following command:

```
Jmeter -n -t E:/apache-jmeter-5.1.1/bin/documents/Homepage.jmx -l E:/apache-jmeter-5.1.1/bin/documents/test2.csv -e -o E:/apache-jmeter-5.1.1/bin/documents/htmlreport
```

Here

- n Non-GUI mode
- t location of Jmeter script
- l location of the result file
- e generate report dashboard after the load test
- o output folder for report dashboard

5 DISCUSSION

5.1 Analysis of Web Application Performance

5.1.1 Aggregate Report Comparison

Aggregate Report

Name: Aggregate Report

Comments:

Write results to file / Read from file

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/s	Sent KB/sec
Homepage	200	622	479	719	1299	3558	362	6445	0.00%	4.0/min	146.43	1.19
Shop_snow	200	398	328	478	598	1388	278	3188	0.00%	4.0/min	35.55	1.91
Shop_snow_	200	1204	848	1708	3764	7468	304	11303	0.00%	4.0/min	54.20	2.42
Shop_techno_	200	402	324	481	581	1448	276	4510	0.00%	4.0/min	27.56	1.70
Shop_techno_	199	926	691	964	1598	6810	297	9837	0.00%	4.0/min	40.28	2.35
Shop_clothing	198	389	321	507	561	1378	217	3235	0.00%	4.0/min	35.30	1.64
Shop_clothing_	198	1362	1028	1836	1991	5838	238	8939	0.00%	4.0/min	55.91	2.43
Shop_water	198	401	328	480	519	1424	232	3124	0.00%	4.0/min	35.18	1.81
Shop_water_	198	810	724	885	1152	3076	226	6645	0.00%	4.0/min	41.02	2.43
Shopfakes	198	431	333	474	636	3156	288	4381	0.00%	4.0/min	44.18	1.95
Shop_bike_	198	558	336	467	758	1375	304	3911	0.00%	4.0/min	30.55	1.44
Service_stor_	197	773	713	970	1329	3545	235	4588	0.00%	4.0/min	49.11	2.45
Search_string	198	342	254	343	452	3010	160	3900	0.00%	4.0/min	5.97	0.60
Search_shoe	198	1275	1114	1885	1827	6880	712	7430	0.00%	4.0/min	47.11	2.47
Shoes_details	198	1278	1102	1608	2134	3367	980	5550	0.00%	4.0/min	45.04	2.10
Added to cart	198	18	17	24	29	44	12	88	0.00%	4.0/min	0.06	0.03
Proceeded to c.	198	437	336	548	898	1723	278	3161	0.00%	4.0/min	23.71	1.43
Checkout	198	18	17	24	29	44	12	88	0.00%	4.0/min	0.07	0.03
Payment_det	195	447	353	615	920	1784	281	3126	0.00%	4.0/min	23.64	1.43
Address_det	194	123	81	120	150	375	55	387	0.00%	4.0/min	0.11	0.03
Complete_p	194	504	354	615	1311	3153	285	4989	0.00%	4.0/min	44.29	1.63
Test	194	1408	1264	1947	2149	2867	714	39256	0.00%	4.9/min	89.68	35.62
TOTAL	4729	1168	444	1306	4059	15435	12	39256	0.00%	1.9/sec	1790.27	71.54

Figure 14: Aggregate Report (200 Threads)

Aggregate Report

Name: Aggregate Report

Comments:

Write results to file / Read from file

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/s	Sent KB/sec
Homepage	500	1470	560	1366	3192	21481	351	26154	0.00%	10.0/min	365.02	2.97
Shop_snow	500	1214	398	801	1530	21425	200	58743	0.00%	10.0/min	86.54	4.78
Shop_snow_	499	1878	864	2570	5272	21987	300	39134	0.00%	10.0/min	136.04	6.03
Shop_techno_	498	1063	383	696	1330	21235	273	21454	0.00%	10.0/min	68.60	4.24
Shop_techno_	498	1872	702	1743	5226	21497	387	22421	0.00%	10.0/min	100.83	5.86
Shop_clothing	496	1392	383	929	2813	21244	226	21890	0.00%	10.0/min	87.50	4.06
Shop_clothing_	495	1722	1201	1903	4101	22227	256	23555	0.00%	10.0/min	137.34	6.04
Shop_waters	495	1298	359	804	2031	21304	228	42298	0.00%	10.0/min	87.34	4.50
Shop_waters_	495	1981	751	2487	6015	21852	228	42455	0.00%	10.0/min	101.88	6.03
Shopfakes	494	1390	378	890	1513	21362	289	42173	0.00%	10.0/min	109.76	4.85
Shop_bike_	493	1818	742	1747	4187	21667	289	73623	0.00%	10.0/min	129.24	6.69
Service_stor_	492	1738	748	1384	3889	22482	237	45391	0.00%	10.0/min	122.83	6.12
Search_string	492	1454	570	891	1613	21275	399	21624	0.00%	10.0/min	76.26	3.53
Search_shoe	491	1148	419	832	1405	21168	223	46167	0.00%	10.0/min	207.23	8.64
Search_shoe	490	557	249	595	1431	4869	138	45800	0.00%	10.0/min	14.59	0.48
Shoes_details	490	2408	1197	2926	7398	22254	725	78958	0.00%	10.0/min	116.97	6.14
Added to cart	488	224	171	274	543	21495	883	24917	0.00%	10.0/min	112.11	5.21
Proceeded to c.	488	28	19	34	41	248	12	1115	0.00%	10.0/min	0.15	0.07
Checkout	488	1124	365	977	1462	21251	278	22287	0.00%	10.0/min	59.07	3.56
Payment_det	486	1296	398	1077	1480	21217	286	21760	0.00%	10.0/min	59.12	3.57
Address_det	486	201	87	373	875	1855	55	4000	0.00%	10.0/min	0.29	0.06
Complete_p	486	1339	432	1518	1761	21866	269	22569	0.00%	10.0/min	110.59	4.68
Test	486	30543	27735	57647	66597	95648	9018	117989	0.00%	9.7/min	2219.85	88.73
TOTAL	11908	2524	481	2010	21126	39542	12	117989	0.00%	3.9/sec	4472.85	178.65

Figure 15: Aggregate Report (500 Threads)

- The aggregate report is the default report in JMeter.
- The aggregate report creates a table row for each differently named request in our test, which is similar to the summary report.
- We should mention the label names correctly to get the best results from the report.

- For both the scenarios, labels are the same as homepage, shop, service, add to cart and checkout.
- And when comparing samples, average, min, max, sent, received, std dev, throughput, avg bytes all values are different due to network issues.
- Errors for both the scenarios are showing 0% because all the tests are passed here.
- If anyone of the tests is failed, we get values in errors.

5.1.2 Aggregate Graph Comparison

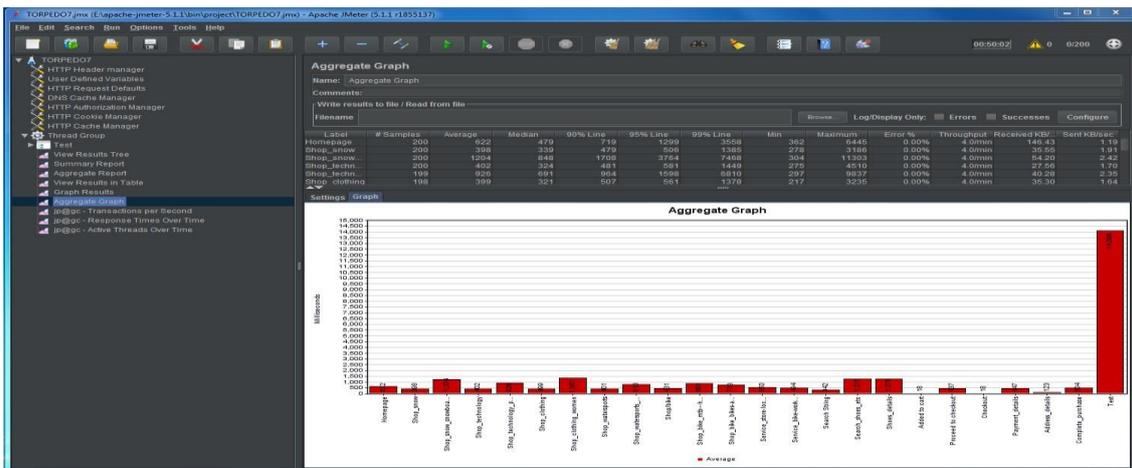


Figure 16: Aggregate Graph (200 Threads)

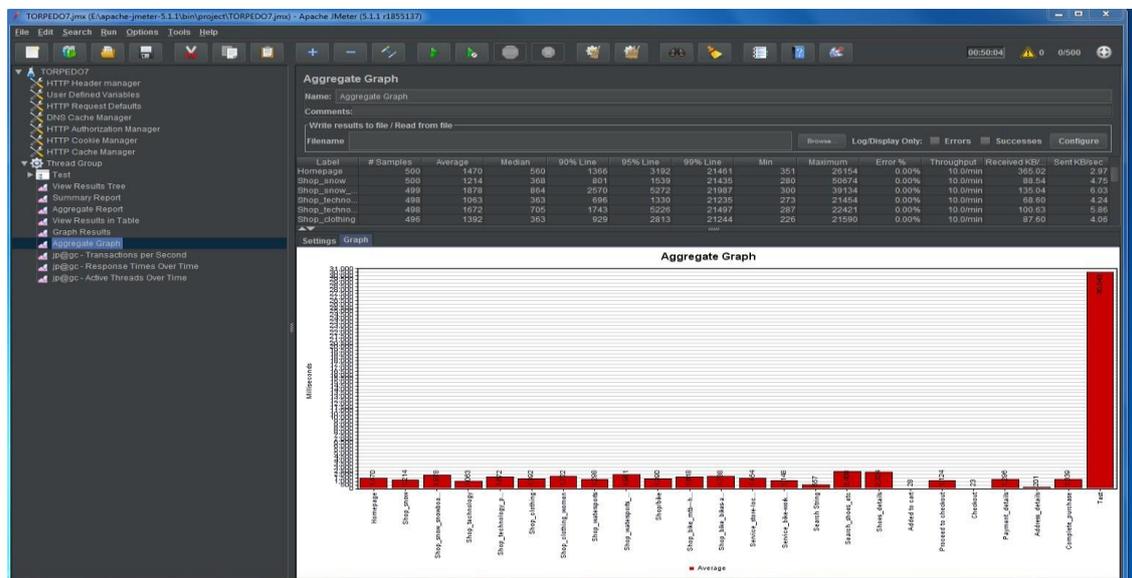


Figure 17: Aggregate Graph (500 Threads)

- The above figures compare the Average Response time and Median Response Time for every single request.

- According to the above two graphs, there is a huge difference between Average Response Time for 200 threads and for 500 threads for each request.
- There is also a huge difference between Median Response Time for 200 threads and for 500 threads for every single request.
- Finally, there is more deviation between Average Response Time and Median Response Time for each request.

5.1.3 Transaction per Second Graph comparison

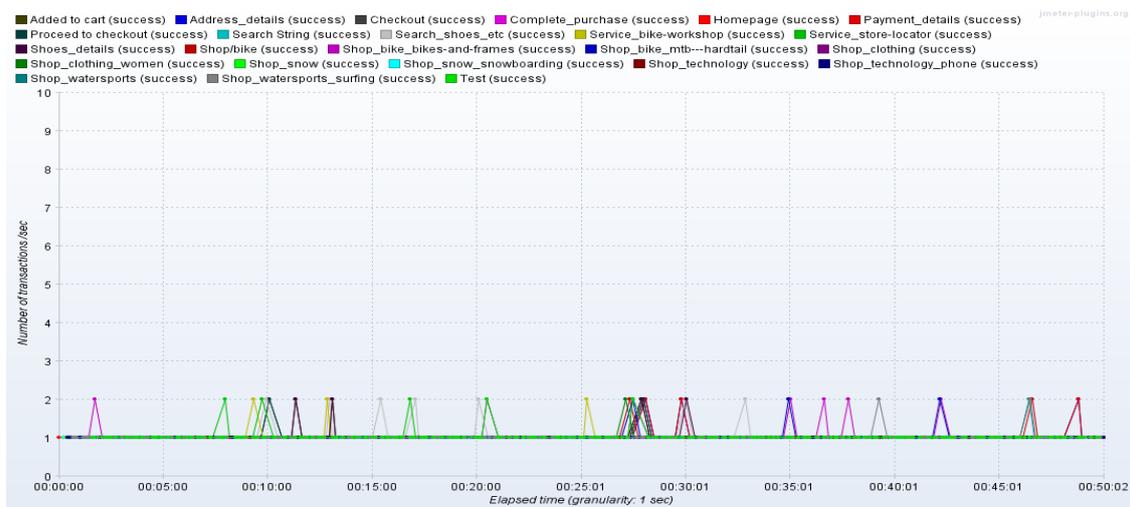


Figure 18: Transaction per Second Graph (200 Threads)

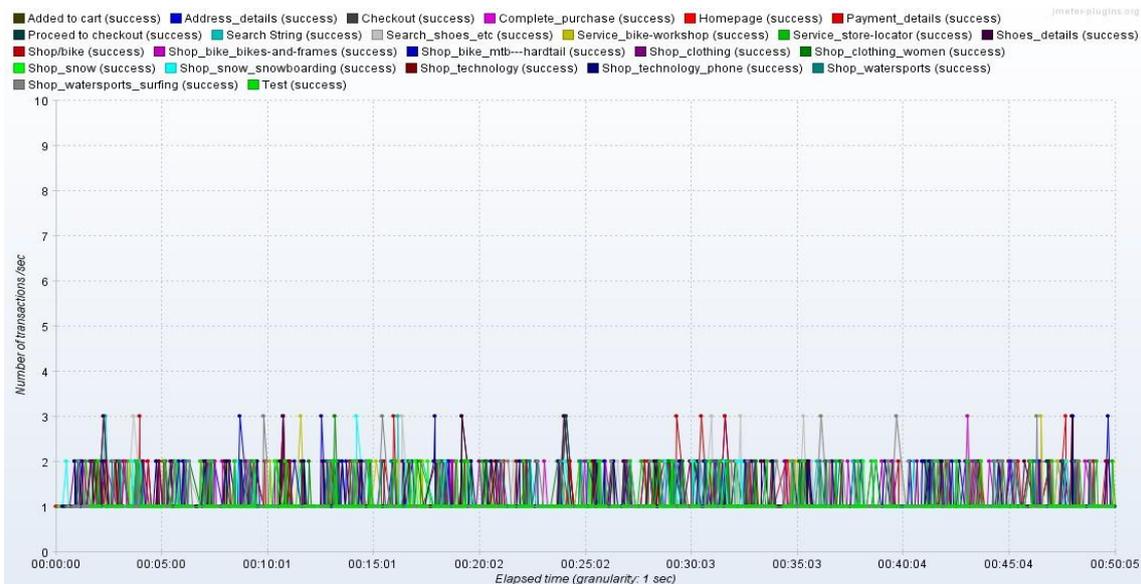


Figure 19: Transactions per Second Graph (500 Threads)

- The above figures 57,58 show the Transactions per Second for 200,500 Threads

- In figure 57 the transactions per second are 1 or 2 all the time.
- But when the threads increased from 200 to 500 transactions per second also increased mostly and reached to 3.

5.1.4 Graph Results comparison

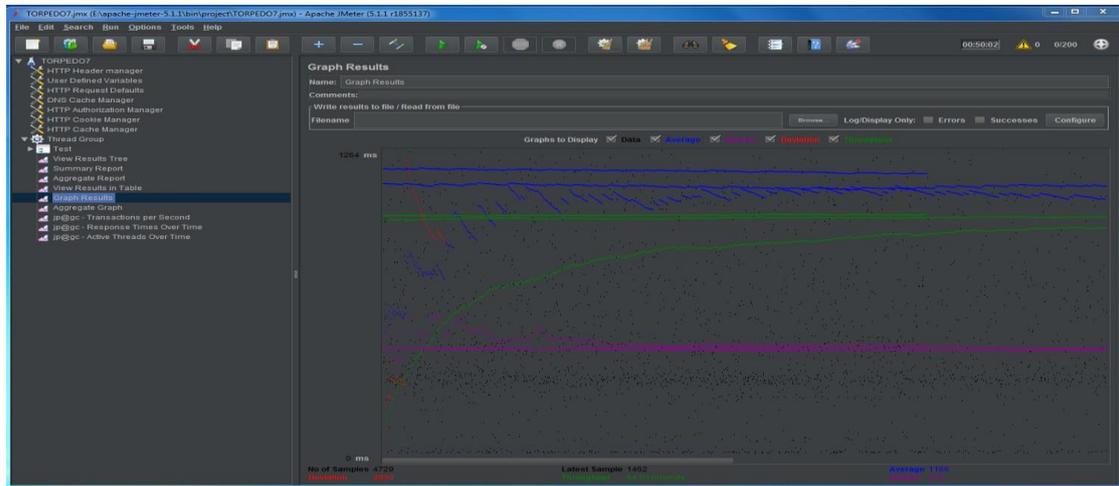


Figure 20: Graph Results (200 Threads)

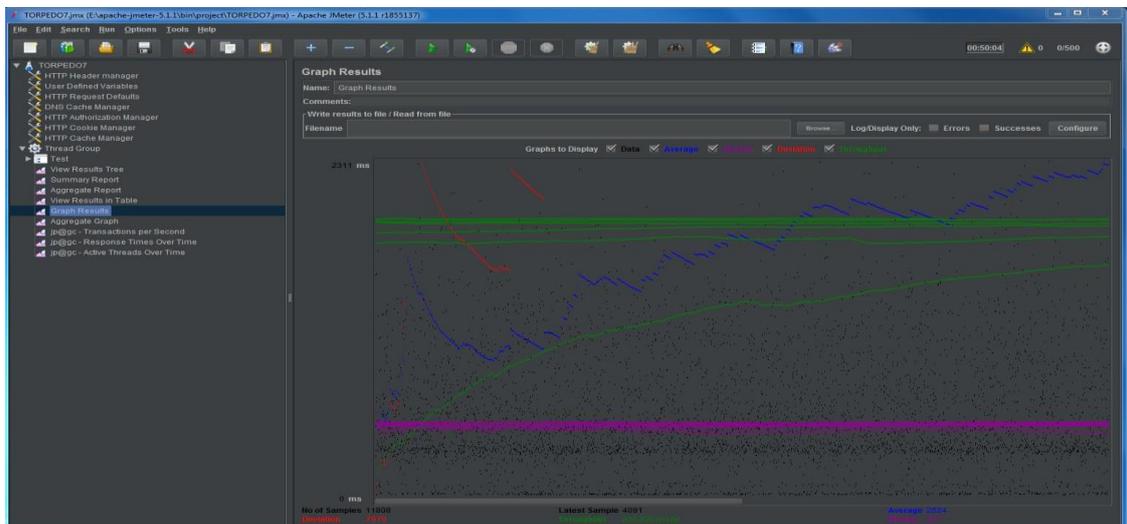


Figure 21: Graph Results Graph (500 Threads)

- This is the most important graph that shows the Median, Average response time, Deviation from the response time and Throughput.
- In figure 60 throughput gradually increased very high nearly two times compared to figure 59, because the first graph is for 200 threads and the second is for 500 threads.

	(200 Threads)	(500 Threads)
Throughput	94.531/minute	235.82/minute

- There is some difference between 200 threads and 500 threads in average response time, median and deviation.

	(200 Threads)	(500 Threads)
Average Response Time	1166	2524
Median	444	481
Deviation	2892	7979

In performance testing we start our script with recording, once we finish with recording, then we open our script and run. However, the first step we need to do in performance testing after recording is to identify the main major requests and delete all the redirected ones, otherwise we will end up with the issues frequently.

This research project report was generated with 200 users and 500 users, but when we started our test with 10 and 50 users we did not face any issues. But once we increased the test from 200 users to 500 users at the time of execution, we faced some issues then we performed brainstorming why are we getting issues for 200 users but not for 20,50 users?

In our research we came to know that in performance testing there are some issues, for example the server should be able to handle all token ids because each run creates freshly ids and server can not handle 500 requests. For this we slowly increased the number of threads and we observed firstly for 50 users then for 100 users so on and if errors were encountered after X numbers of users then it meant that the server was not ready to accept after that thread. We increased the number of users slowly like 20,30,40,50, 100 for 500 so on. Further it required more understanding. So, we increased the ramp-up time and duration and observed for 50 users at the beginning then increased the threads to 100 users then later to 200 and 500.

6. CONCLUSION

Torpedo7 website report covers a brief introduction about Flipmind Company and what they do in New Zealand and Torpedo7 website. Flipmind Company is using an agile methodology for its Torpedo7 website project. Torpedo7 website needs performance testing for Black Friday and Christmas deals. Using the JMeter tool created test scripts and analyzed the results. A load test is done for the Torpedo7 website using JMeter for the Homepage, Shop feature, Service feature, Search functionality, add to cart and checkout with 200, 500 users and hence analyzed the results using summary report, graphs, CSV file, and HTML report.

7. RECOMMENDATIONS

Torpedo7 website was tested with 200, 500 threads. With the analysis of test results, the current system is optimal for handling increased traffic on the website. Further, it does not need any upgrades now. The graphs will not remain the same every time because of the load stress or network issue. To get the proper and exact results it recommends executing the test night after 10 pm. Especially if working on the performance testing on live websites, it is always recommended to run the load test at night-time to get accurate results.

REFERENCES

- [1] Patil, S. S., & Joshi, S. D. (2012). Identification of Performance Improving Factors for Web Application by Performance Testing. *Int. J. Emerg. Technol. Adv. Eng.*, 2(8), 433-436.
- [2] Arul, D. P., & Asokan, M. (2014). Load testing for query based e-commerce web applications with cloud performance Testing Tools. *International Journal of Computer Engineering & Technology (IJCET)*, 5(10), 01-10.
- [3] Chandel, V., Patial, S., & Guleria, S. (2013). Comparative Study of Testing Tools: Apache JMeter and Load Runner. *International Journal of Computing and Corporate Research*, 3(3).
- [4] Khan, R., & Amjad, M. (2016). Performance testing (load) of web applications based on test case management. *Perspectives in Science*, 8, 355-357.
- [5] Laanti, M., Salo, O., & Abrahamsson, P. (2011). Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation. *Information and Software Technology*, 53(3), 276-290.
- [6] Khan, R., & Amjad, M. (2016). Performance testing (load) of web applications based on test case management. *Perspectives in Science*, 8, 355-357.
- [7] Erinle, B. (2013). *Performance testing with JMeter 2.9*. Packt Publishing Ltd.
- [8] Arul, D. P., & Asokan, M. (2014). Load testing for query based e-commerce web applications with cloud performance Testing Tools. *International Journal of Computer Engineering & Technology (IJCET)*, 5(10), 01-10.
- [9] Zhu, K., Fu, J., & Li, Y. (2010). Research the performance testing and performance improvement strategy in web applications. In *2010 2nd international Conference on Education Technology and Computer (Vol. 2, pp. V2-328)*. IEEE.
- [10] Van, P. H. O., Phipps, D. A., & Lin, S. (2016). U.S. Patent Application No. 14/730,692.
- [11] Rasal, Y. M., & Nagpure, S. (2015). Web application: Performance testing using the reactive based framework. *IJRCCCT*, 4(2), 114-118.
- [12] Kao, C. H., Lin, C. C., & Chen, J. N. (2013). Performance testing framework for rest-based web applications. In *2013 13th International Conference on Quality Software (pp. 349-354)*. IEEE.
- [13] Dhote, M. R., & Sarate, G. G. (2012). Performance testing complexity analysis on Ajax-based web applications. *IEEE Software*, 30(6), 70-74.
- [14] Kiran, S., Mohapatra, A., & Swamy, R. (2015). Experiences in performance testing of web applications with Unified Authentication platform using Jmeter. In *2015 international symposium on technology management and emerging technologies (ISTMET) (pp. 74-78)*. IEEE
- [15] Hussain, S., Wang, Z., Toure, I. K., & Diop, A. (2013). Web service testing tools: A comparative study. *arXiv preprint arXiv:1306.4063*.
- [16] Patil, S. S., & Joshi, S. D. (2012). Identification of Performance Improving Factors for Web Application by Performance Testing. *Int. J. Emerg. Technol. Adv. Eng.*, 2(8), 433-436.
- [17] Sharma, M., Vaishnavi, S. I., Sugandhi, S., & Abhinandhan, S. (2016). A comparative study of load testing tools. *International Journal of Innovative Research in Computer and Communication Engineering*, 4(2), 1906-1912.
- [18] Jha, N., & Popli, R. (2017). Comparative analysis of web applications using JMeter. *International Journal of Advanced Research in Computer Science*, 8(3).
- [19] Arslan, M., Qamar, U., Hassan, S., & Ayub, S. (2015). Automatic performance analysis of cloud-based load testing of web-application & its comparison with traditional load testing. In *2015 6th*

IEEE International Conference on Software Engineering and Service Science (ICSESS) (pp. 140-144). IEEE.

- [20] Bhardwaj, S., & Sharma, A. K. (2015). Performance Testing Tools: A Comparative Analysis. International Journal of Engineering Technology, Management and Applied Sciences, 3(4).

AUTHORS

First Author

I am Sushma. I have completed my Bachelor's in Computers Science from JNTU University India. I have recently completed my graduate diploma in Software Testing from AGI Education Limited New Zealand. I am passionate about testing and love to deliver quality products to customers.



Second Author

Dr. Shahid Ali is a senior lecturer and IT program leader at AGI Education Limited, Auckland, New Zealand. He has published number of research papers on ensemble learning. His expertise and research interests include ensemble learning, machine learning, data mining and knowledge discovery.