International Journal of System and Software Engineering
7 (1), June 2019, 28-37
http://www.publishingindia.com/ijsse

Performance Testing and Enhancement of Java Web Applications

Samrat Sandesh Naik Gaonkar[1*], Anusha Pai[2] and Louella Mesquita Colaco[3]
[1]Student (M.Tech.), Information Technology Department, Padre Conceicao College of Engineering, Verna, Goa, India.
Email: samratng@gmail.com
[2]Associate Professor, Computer Department, Padre Conceicao College of Engineering, Verna, Goa, India.
Email: anusha.pai@gmail.com
[3]Assistant Professor, Computer Department, Padre Conceicao College of Engineering, Verna, Goa, India.
Email: lmesquita@rediffmail.com
*Corresponding Author

Abstract: In today's digital world no one likes slow loading web pages that test user's patience and waste his time; hence the performance of web application becomes principal constituent of the user experience. This paper intends to focus on techniques for evaluating and improving Java web application's performance iteratively. This work will make use of best in class performance testing tools to evaluate the performance of the developed Java web application and further attempt to improve the applications performance using the analysis made from the performance test results. The analysis will enable to target the particular bottlenecks in the application that hamper its performance. The paper will focus on different performance enhancing techniques that can be incorporated into Java web development enabling to improve the performance of the application in the targeted areas to build efficient, swift and robust Java web applications which will run smoothly under heavy user load during peak business hours.

Keywords: Java web application development, Load testing, Performance enhancement, Performance testing, Software development.

I. Introduction

Today every business wants to go online and be accessible to their customers irrespective of their geographic location. In such a growth driven scenario the performance of the web application holds the key. If the web application is not running fast it might hamper the growth of the business, its customers might lose interest and choose not to use the service in future.

Thus performance testing phase becomes immensely important in software development cycle. This work comprises of initially developing a Java web application using the standard Java web development tools.

The focus then will be to test the performance of the Java web application using Apache Jmeter tool for load testing. This tool will enable to generate scripts for various actions the virtual users will perform on the Java web application. The load that is the number of virtual users can be adjusted as needed.

The results from the load test will be used to improve the application's performance by highlighting the areas in the application that increase the response time under substantial load. It will focus on developing techniques to enhance the performance of the application at these points making it faster and more responsive.

These performance testing and improvement steps can be carried out iteratively till satisfactory performance is not achieved.

Thus this work will enable to deliver a fast, responsive and stable Java web application that performs satisfactorily during high user load at peak business hours.

II. Related Work

Rijwan Khan and Mohd. Amjad in their paper [1] discuss the importance of performance testing of web applications. It discusses that the monitoring of the system during load testing provides very useful information about the application and by using these results the customer can take major decisions. They carry out load test on their web application using HP ALM-LoadRunner version 11 with an aim to verify if the performance of the application meets the customer requirements mentioned in the Service Level Agreement for application development. They try to determine any performance bottlenecks like high CPU utilization, memory utilization or any hardware issues. They analyses the reports generated which are in tabular and graphical form to come to a conclusion.

In the conclusion the work verifies the fact that the application has an overall response time of less than 5 seconds as per the Service Level Agreement (SLA).

In our research work we try to take this work one step ahead by analyzing the reports and further making improvements our web application to exhibit enhanced levels of performance.

A lot of research has also gone into the various performance testing tools, trying to evaluate the tools to find the most suitable one.

Shikha Dhiman and Pratibha Sharma in their paper [2] compare three such tools namely Apache Jmeter, Grinder and HttpRider. In this paper they highlight the importance of the performance / load testing tools. The paper says that the performance testing tools automate testing process making it easier, as manual testing is costlier, requires too many efforts and is very time consuming. Thus the automated load testing tools reduce the cost, time and effort required.

The paper compares the three testing tools on parameters like response time, throughput and latency recorded during the tests run on the respective tools. This comparison helps us to choose a performance testing tool based on our specific requirements.

Similarly paper [3] compare another set of three such tools namely NeoLoad, loadstar, WAPT.

Another research paper [4] presents a new approach for performing load testing of web applications by simulating realistic user behavior with stochastic models of user behavior. The models are constructed from sample data and take into account effects of session history on user behavior and the existence of different categories of users.

Paper [5] introduces performance testing as a service for web applications, which provides all performance testing activities including automatic test case generation and test execution.

Papers [6], [7], [8], [9] also emphasize on the role of performance testing in software development and different ways to evaluate the performance of a web application under test.

III. Software Development

This research work focuses on the e-commerce domain. A Java web application is developed for an online gym store.

Fig. 1 below shows the system architecture for this application.

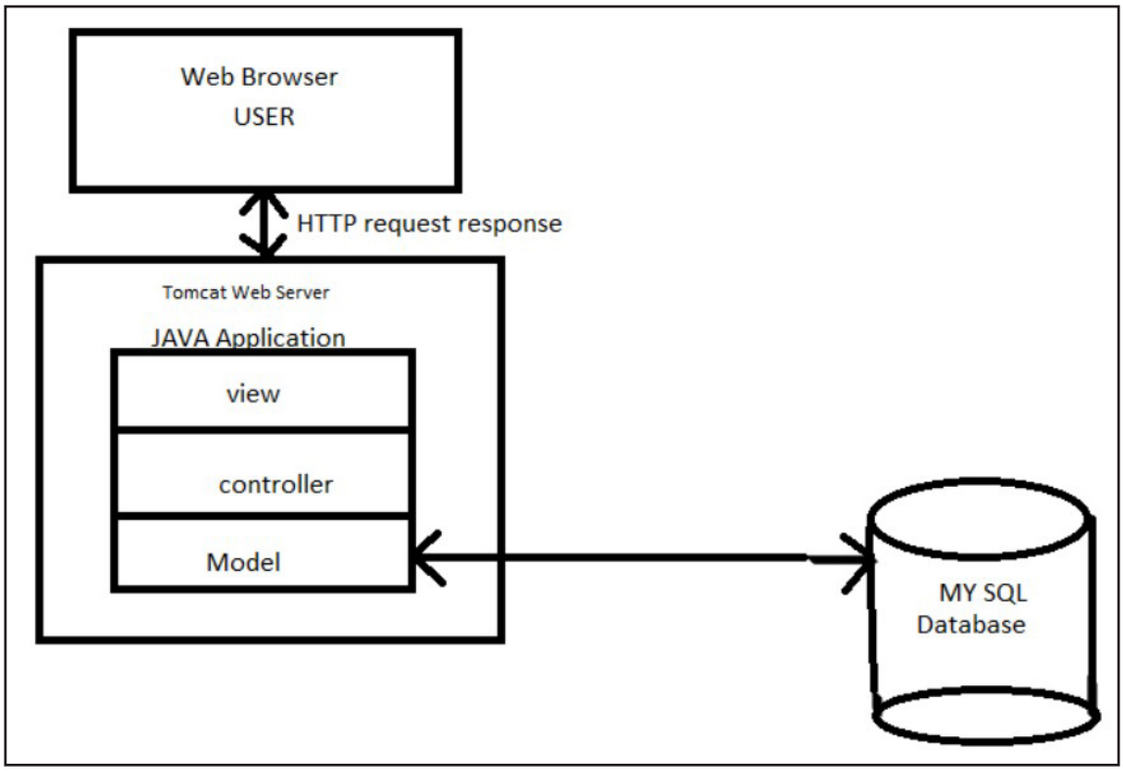


Fig. 1: Java Web Application System Architecture

The application is hosted on Apache Tomcat (version 7.0) server and uses MySQL (version 5.1) database.

The Application adheres to the MVC framework which divides the application into three interconnected modules namely model, view and controller.

Fig. 2 below depicts the flow of control within the application.

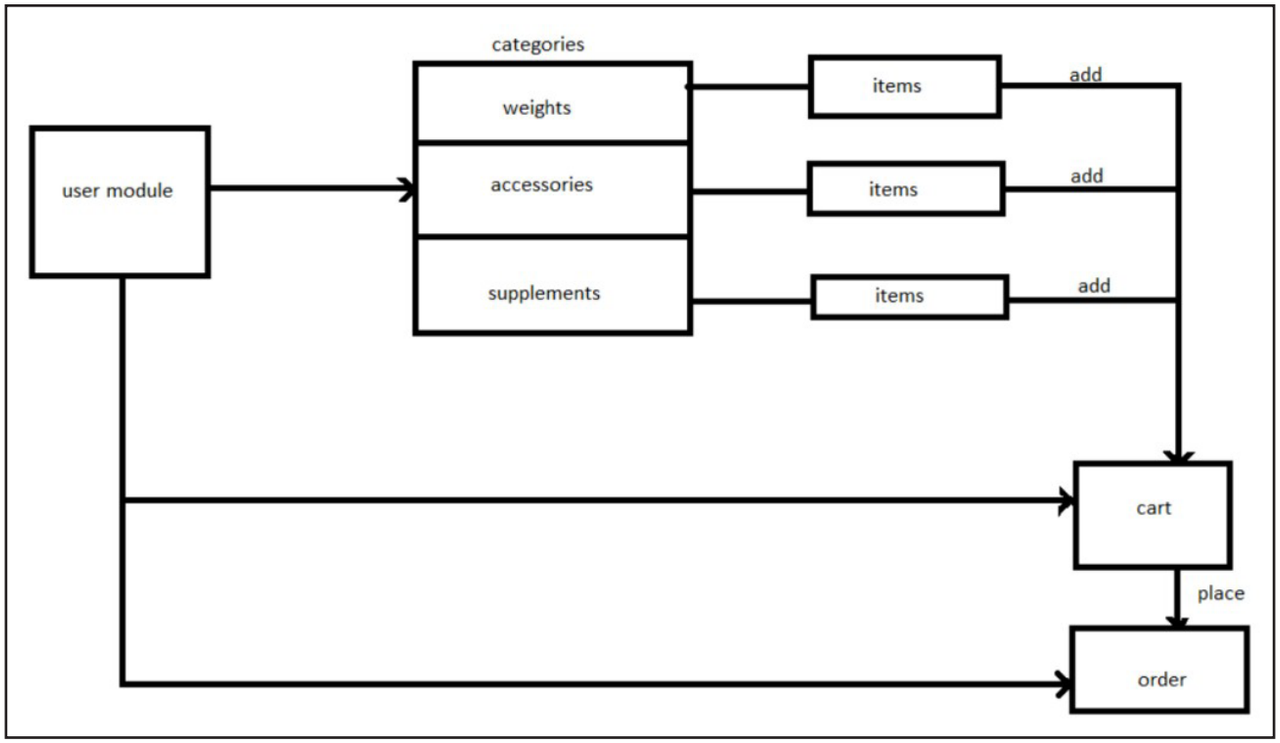


Fig. 2: The Control Flow within the Java Web Application

Fig. 3 and Fig. 4 depict the use case diagram and the Entity Relationship diagram drawn using SmartDraw, LLC web application for the Java web application respectively.

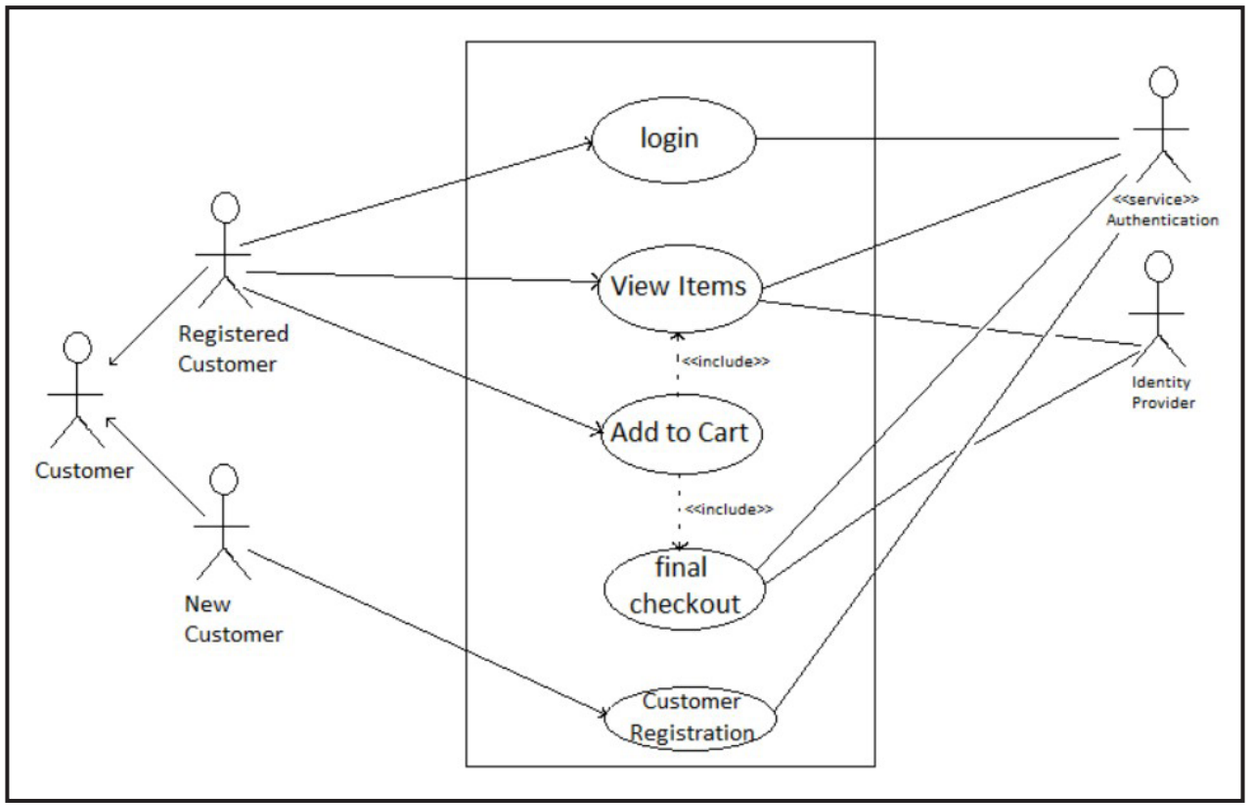
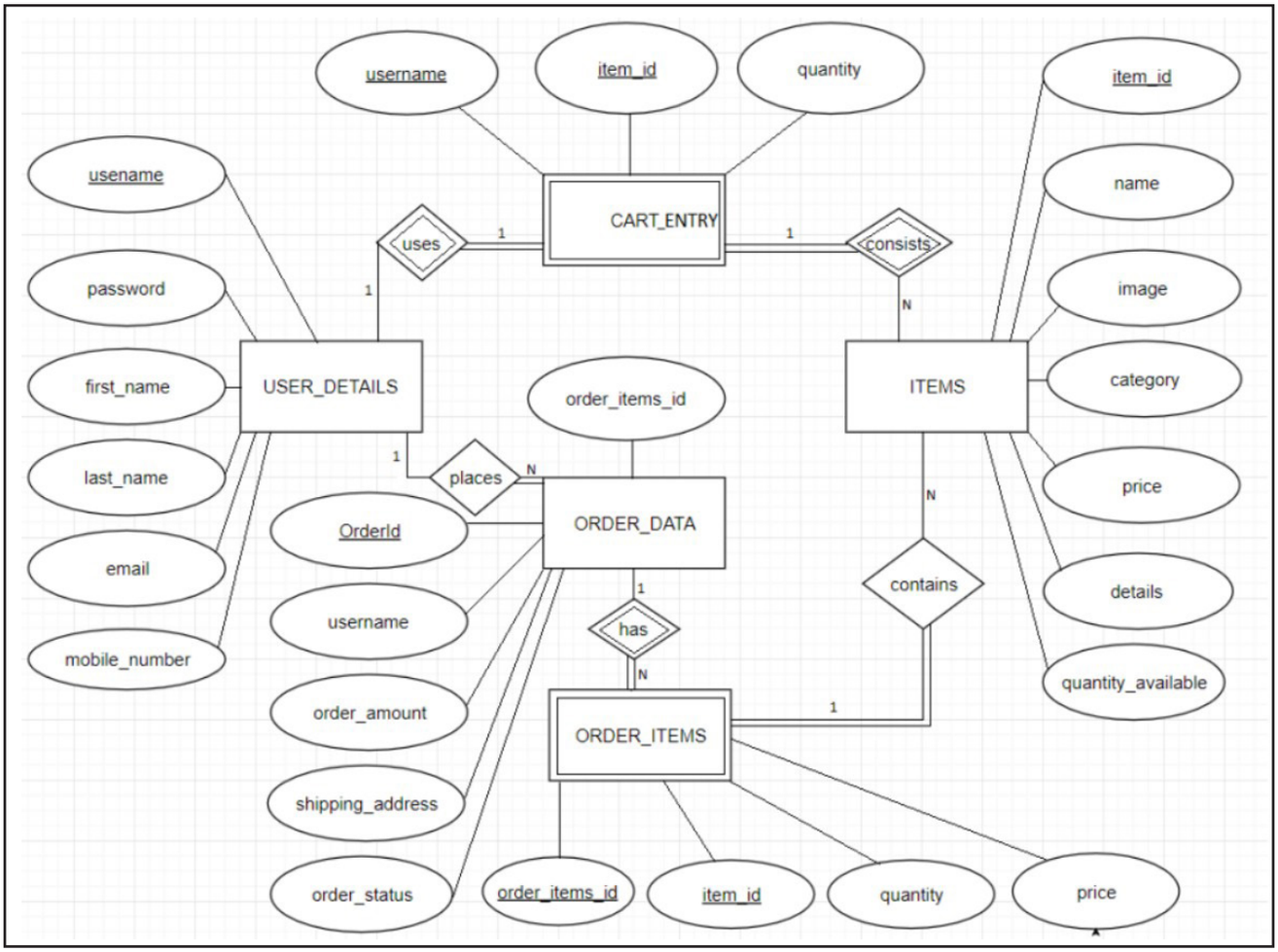


Fig. 3: Use Case Diagram for the Java Web Application



Source: Drawn using SmartDraw, LLC web application

Fig. 4: Entity Relationship Diagram for the Java Web Application

IV. Performance Testing and Improvement

The tool used for carrying out the performance testing in this work is Apache Jmeter (version 2.1).

It can be used to simulate a heavy load on a server hosting a web application to test its strength and to analyze overall performance under different load types [10].

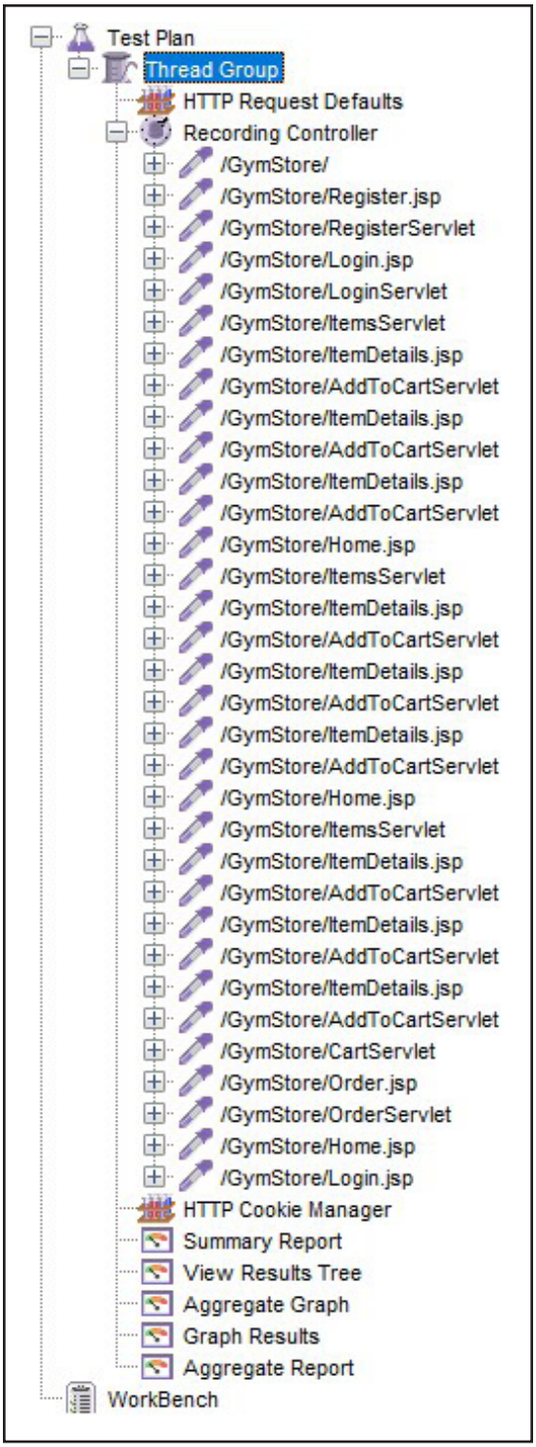
A. Initial Performance Test

The objective here is to formulate scripts for Apache Jmeter for the Java web application under test. The script will have the list of actions the virtual users will perform once the load test is started.

See Fig. 5 below for the list of actions to be performed in the Jmeter script rendered from Apache Jmeter 2.10.

Once the script is ready the Apache Jmeter is loaded with the number of concurrent virtual users that will perform the actions. These users will be added consecutively with a fixed interval of time known as the ramp up time, which is selected before running the tests. Thus when all the users get added the application, it will experience a virtual load of the selected number of users performing the set of actions given to them simultaneously.

It begins with 100 users then increased to 200 users, followed by 300 users, with ramp-up period of 10 that means each user will be added after an interval of 10 seconds.



Source: Apache Jmeter version 2.10

Fig. 5: List of Actions to be Performed in the Jmeter Script

It is observed that at 310 users the application fails to handle the user requests successfully and the application crashes due to its hardware and software resource limitation.

Table I, Table II, Table III show the stress test results for 100, 200 and 300 users respectively. This data is rendered by Apache Jmeter Version 2.10.

Fig. 6, Fig. 7, Fig. 8 show the graphical representation for the stress test results for 100, 200 and 300 users respectively. This data is rendered by Apache Jmeter Version 2.10.

TABLE I: PERFORMANCE TEST RESULTS FOR 100 USERS

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
/GymStore/	100	8	1	139	18.15	0.00%	10.0/sec	16.88	1732.0
/GymStore/Register.jsp	100	24	0	211	44.94	0.00%	10.0/sec	26.16	2684.0
/GymStore/RegisterSe...	100	365	74	921	149.50	0.00%	9.8/sec	5.69	595.0
/GymStore/Login.jsp	200	6	0	117	9.25	0.00%	6.7/sec	10.74	1649.0
/GymStore/LoginServlet	100	1360	32	4583	1383.06	0.00%	9.2/sec	11.74	1311.9
/GymStore/ItemsServlet	300	1403	25	4674	1365.78	0.00%	13.7/sec	60.75	4527.7
/GymStore/ItemDetails...	900	7	0	208	9.65	0.00%	39.3/sec	53.87	1405.3
/GymStore/AddToCart...	900	553	89	2031	273.66	0.00%	38.7/sec	171.75	4545.7
/GymStore/Home.jsp	300	6	0	94	6.90	0.00%	10.3/sec	13.14	1311.9
/GymStore/CartServlet	100	1781	28	5982	1096.75	0.00%	4.6/sec	35.96	7926.9
/GymStore/Order.jsp	100	8	0	24	5.68	0.00%	4.9/sec	5.80	1216.0
/GymStore/OrderServlet	100	8287	5066	13688	3356.74	0.00%	3.9/sec	3.17	829.9
TOTAL	3300	640	0	13688	1648.19	0.00%	109.1/sec	292.86	2747.6

Source: Apache Jmeter version 2.10

TABLE II: PERFORMANCE TEST RESULTS FOR 200 USERS

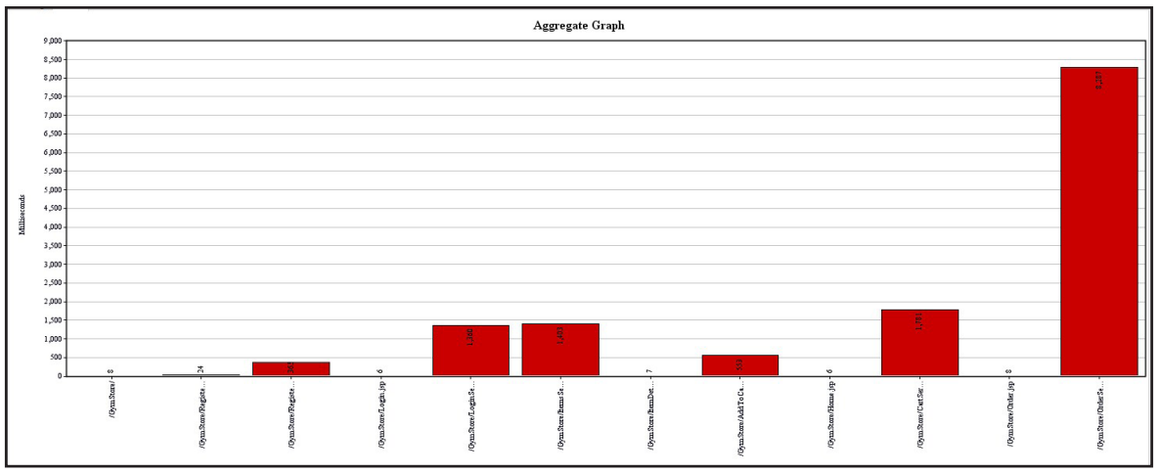
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
/GymStore/	200	613	6	4317	799.99	0.00%	16.8/sec	28.51	1736.1
/GymStore/Register.jsp	200	578	4	4352	976.31	0.00%	17.3/sec	45.34	2689.4
/GymStore/RegisterSe...	200	3339	77	8727	2415.32	0.00%	14.1/sec	8.28	601.6
/GymStore/Login.jsp	400	140	3	4300	409.50	0.00%	4.6/sec	7.39	1657.4
/GymStore/LoginServlet	200	5053	41	11746	2808.03	0.00%	13.0/sec	16.78	1325.1
/GymStore/ItemsServlet	600	5429	580	14345	3357.88	0.00%	10.8/sec	48.06	4542.8
/GymStore/ItemDetails...	1800	128	3	5865	311.28	0.00%	32.1/sec	44.65	1422.2
/GymStore/AddToCart...	1800	2796	171	5694	1195.35	0.00%	30.1/sec	134.36	4564.4
/GymStore/Home.jsp	600	23	3	903	64.21	0.00%	8.0/sec	10.41	1326.5
/GymStore/CartServlet	200	3803	52	11824	2189.82	0.00%	5.7/sec	44.11	7946.4
/GymStore/Order.jsp	200	171	3	1045	292.25	0.00%	6.2/sec	7.51	1235.0
/GymStore/OrderServlet	200	23037	14100	38921	8128.38	0.00%	4.3/sec	3.60	849.5
TOTAL	6600	2410	3	38921	4552.58	0.00%	73.1/sec	197.13	2763.2

Source: Apache Jmeter version 2.10

TABLE III: PERFORMANCE TEST RESULTS FOR 300 USERS

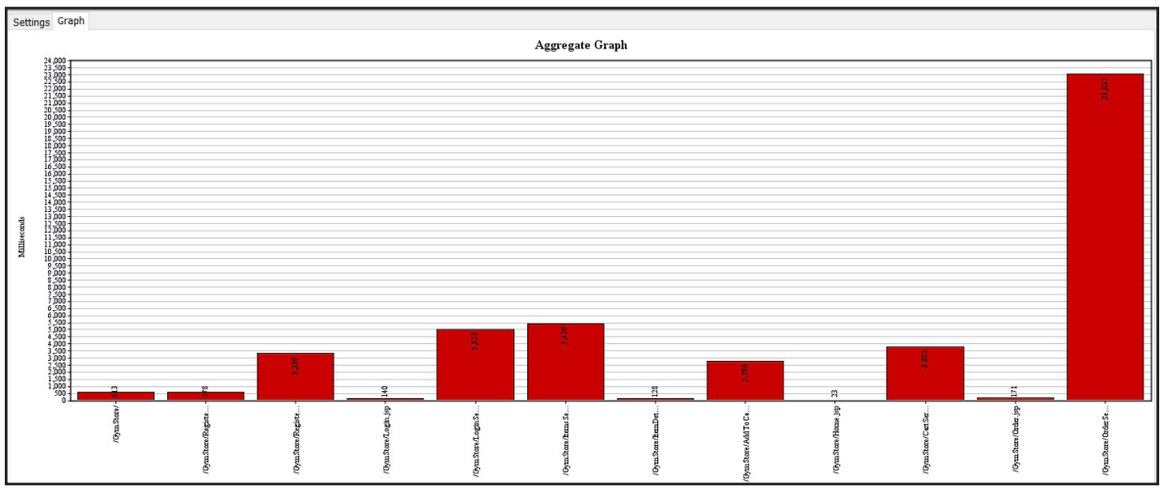
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
/GymStore/	300	485	4	2318	620.40	0.00%	25.3/sec	42.92	1738.3
/GymStore/Register.jsp	300	336	3	2300	522.40	0.00%	22.5/sec	59.04	2690.5
/GymStore/RegisterSe...	300	2358	91	4731	782.88	0.00%	19.1/sec	11.25	604.4
/GymStore/Login.jsp	600	297	3	2637	468.16	0.00%	5.3/sec	8.63	1660.4
/GymStore/LoginServlet	300	10099	36	21981	7273.31	0.00%	9.4/sec	12.14	1326.1
/GymStore/ItemsServlet	900	5271	32	15298	2527.34	0.00%	11.2/sec	49.90	4545.4
/GymStore/ItemDetails...	2700	977	3	4093	678.13	0.00%	29.7/sec	41.25	1423.6
/GymStore/AddToCart...	2700	2886	69	8867	1402.34	0.00%	29.3/sec	130.72	4564.1
/GymStore/Home.jsp	900	884	2	4006	900.49	0.00%	8.5/sec	10.97	1328.3
/GymStore/CartServlet	300	4943	1847	17228	2572.61	0.00%	5.3/sec	40.95	7945.9
/GymStore/Order.jsp	300	573	3	3836	740.05	0.00%	6.0/sec	7.24	1229.6
/GymStore/OrderServlet	300	21848	14783	29432	4723.67	0.00%	4.6/sec	3.77	846.6
TOTAL	9900	2863	2	29432	4391.86	0.00%	87.5/sec	236.10	2764.0

Source: Apache Jmeter version 2.10



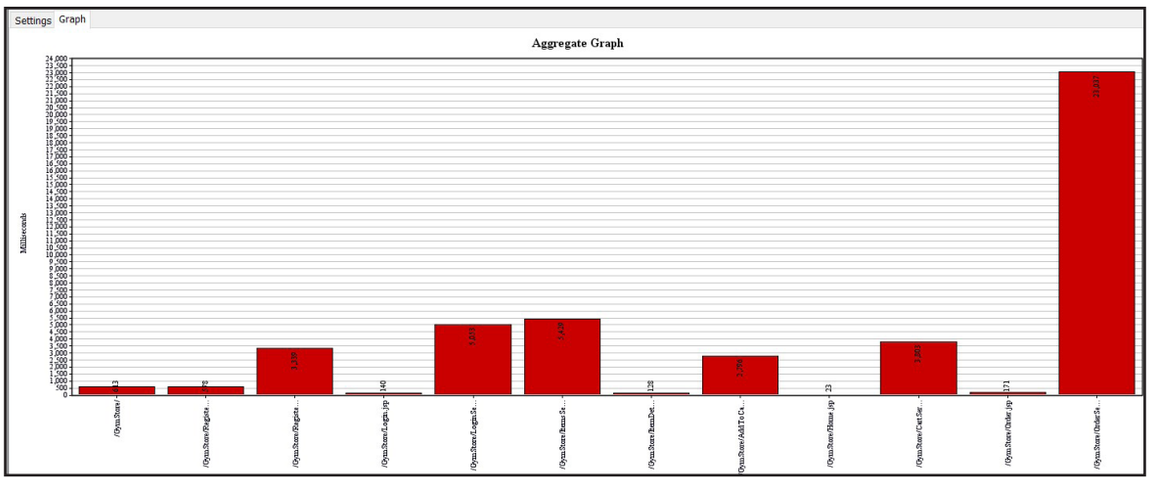
Source: Apache Jmeter version 2.10

Fig. 6: Graphical Representation for the Stress Test Results for 100 Users



Source: Apache Jmeter version 2.10

Fig. 7: Graphical Representation for the Stress Test Results for 300 Users



Source: Apache Jmeter version 2.10

Fig. 8: Graphical Representation for the Stress Test Results for 300 Users

B. Analysis of the Performance Test Results

After the Stress test it is evident that the Order servlet is taking the longest time to respond, with average response time of 17724.00 milliseconds for the tests carried out.

Thus Order servlet acts as bottleneck for the application and needs to be taken care of in order to improve the performance of the application.

C. Improvement in the Application

The high response times observed in Order servlet are attributed to the lengthy Database transactions.

To counter this, the concepts of query optimization are used.

The Database queries for Order module are modified to incorporate Inner joins wherever possible to reduce the amount of data processed for select queries.

Changes are also made for the insert queries, individual insert queries have been replaced by bulk insert queries using Java arrays and StringBuilder classes. As it was observed that the insert queries take the longest time to execute. The idea is to save on the time required to make individual database transactions and have the data stored in arrays and execute bulk database inserts using StringBuilders taking care that the consistency and availability of data is maintained.

D. Performance Testing After Improvement

Same three tests with 100, 200 and 300 users using Apache Jmeter are carried out for the Updated Java web application.

Table IV, Table V, Table VI show the stress test results for 100, 200 and 300 users post improvement respectively. This data is rendered by Apache Jmeter Version 2.10.

Fig. 9, Fig. 10, Fig. 11 show the graphical representation for the stress test results for 100, 200 and 300 users post improvement respectively. This data is rendered by Apache Jmeter Version 2.10.

Table IV: Performance Test Results for 100 Users Post Improvement

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
/GymStore/	100	51	2	902	135.49	0.00%	9.3/sec	15.67	1732.0
/GymStore/Register.jsp	100	94	1	919	161.58	0.00%	9.3/sec	24.29	2684.0
/GymStore/RegisterSe...	100	681	69	2428	442.46	0.00%	8.5/sec	4.95	595.0
/GymStore/Login.jsp	200	14	1	408	37.33	0.00%	7.3/sec	11.69	1649.0
/GymStore/LoginServlet	100	1808	41	7475	2023.75	0.00%	6.2/sec	7.96	1311.9
/GymStore/ItemsServlet	300	1325	27	4316	1003.02	0.00%	12.6/sec	55.68	4527.7
/GymStore/ItemDetails...	900	47	0	2508	237.40	0.00%	37.2/sec	51.00	1405.3
/GymStore/AddToCart...	900	720	115	2587	386.84	0.00%	40.1/sec	178.03	4545.7
/GymStore/Home.jsp	300	9	1	87	7.97	0.00%	12.5/sec	16.07	1311.9
/GymStore/CartServlet	100	1867	56	3377	840.09	0.00%	7.0/sec	53.89	7926.9
/GymStore/Order.jsp	100	29	1	1562	154.28	0.00%	7.5/sec	8.89	1216.0
/GymStore/OrderServlet	100	4794	1084	8141	2155.16	0.00%	7.5/sec	6.09	829.9
TOTAL	3300	614	0	8141	1132.44	0.00%	116.9/sec	313.79	2747.6

Source: Apache Jmeter version 2.10

Table V: Performance Test Results for 200 Users Post Improvement

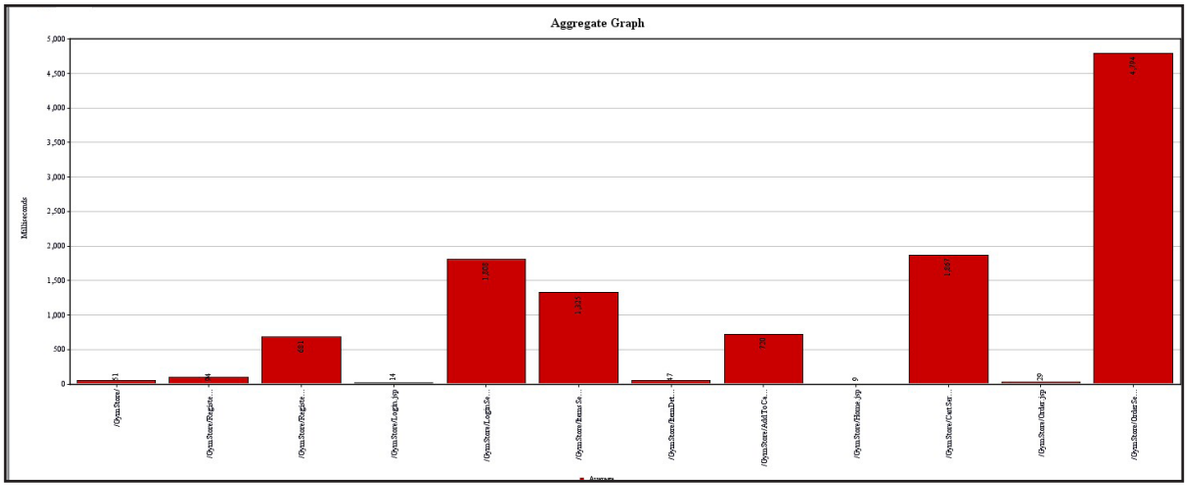
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
/GymStore/	200	56	1	654	120.73	0.00%	19.5/sec	33.11	1736.7
/GymStore/Register.jsp	200	106	1	862	178.68	0.00%	19.5/sec	51.25	2688.8
/GymStore/RegisterSe...	200	950	119	2119	462.61	0.00%	18.7/sec	10.95	599.6
/GymStore/Login.jsp	400	13	0	615	45.49	0.00%	7.3/sec	11.82	1653.8
/GymStore/LoginServlet	200	1763	46	9034	1812.09	0.00%	10.2/sec	13.11	1318.9
/GymStore/ItemsServlet	600	2873	24	9164	2796.91	0.00%	13.6/sec	60.34	4540.7
/GymStore/ItemDetails...	1800	19	0	886	53.14	0.00%	40.1/sec	55.51	1419.1
/GymStore/AddToCart...	1800	2118	73	5388	1066.11	0.00%	39.8/sec	177.19	4559.3
/GymStore/Home.jsp	600	10	0	142	17.63	0.00%	11.3/sec	14.64	1323.0
/GymStore/CartServlet	200	1731	16	4456	1501.58	0.00%	5.6/sec	43.45	7939.5
/GymStore/Order.jsp	200	19	1	137	20.64	0.00%	5.8/sec	6.89	1223.0
/GymStore/OrderServlet	200	11631	4350	24127	2685.35	0.00%	4.6/sec	3.75	839.1
TOTAL	6600	1338	0	24127	2431.43	0.00%	120.5/sec	324.64	2759.1

Source: Apache Jmeter version 2.10

Table VI: Performance Test Results for 300 Users Post Improvement

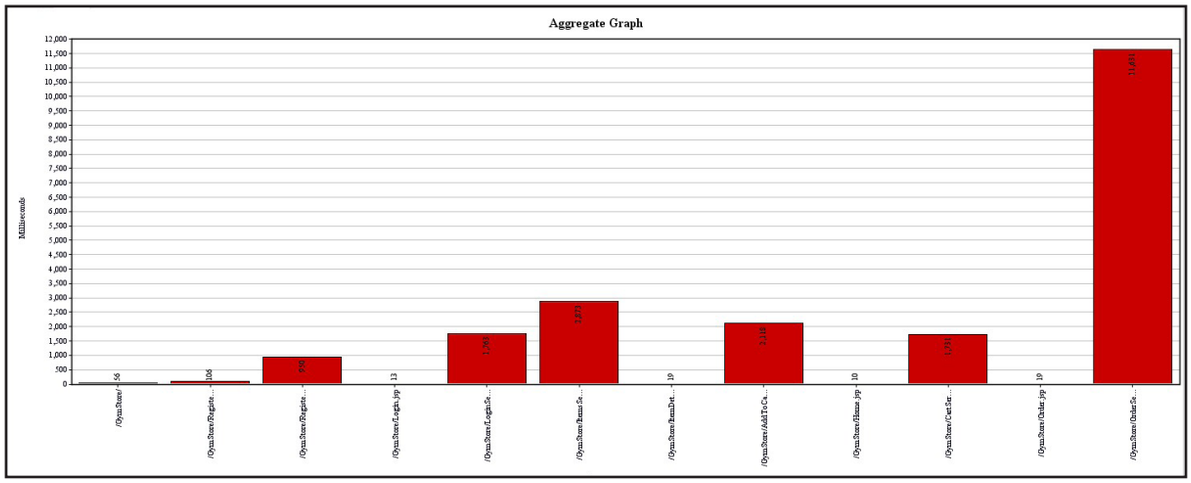
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
/GymStore/	300	260	1	1675	420.64	0.00%	24.6/sec	41.85	1740.4
/GymStore/Register.jsp	300	219	0	1568	332.01	0.00%	23.9/sec	62.76	2692.4
/GymStore/RegisterSe...	300	1463	51	4918	1036.40	0.00%	20.6/sec	12.14	603.4
/GymStore/Login.jsp	600	170	0	1623	320.20	0.00%	8.9/sec	14.39	1659.3
/GymStore/LoginServlet	300	2373	19	9821	1958.12	0.00%	18.5/sec	23.86	1323.7
/GymStore/ItemsServlet	900	4109	20	12364	3022.94	0.00%	16.7/sec	73.97	4540.9
/GymStore/ItemDetails...	2700	556	0	3885	590.58	0.00%	45.3/sec	63.02	1423.3
/GymStore/AddToCart...	2700	2031	66	5937	907.23	0.00%	45.0/sec	200.50	4563.0
/GymStore/Home.jsp	900	505	0	2401	474.66	0.00%	13.5/sec	17.48	1327.3
/GymStore/CartServlet	300	3684	35	14487	2453.59	0.00%	6.5/sec	50.17	7946.4
/GymStore/Order.jsp	300	226	0	2069	312.82	0.00%	6.8/sec	8.17	1231.3
/GymStore/OrderServlet	300	10777	3793	31120	4710.42	0.00%	5.9/sec	4.88	848.2
TOTAL	9900	1711	0	31120	2490.98	0.00%	146.4/sec	395.08	2763.2

Source: Apache Jmeter version 2.10



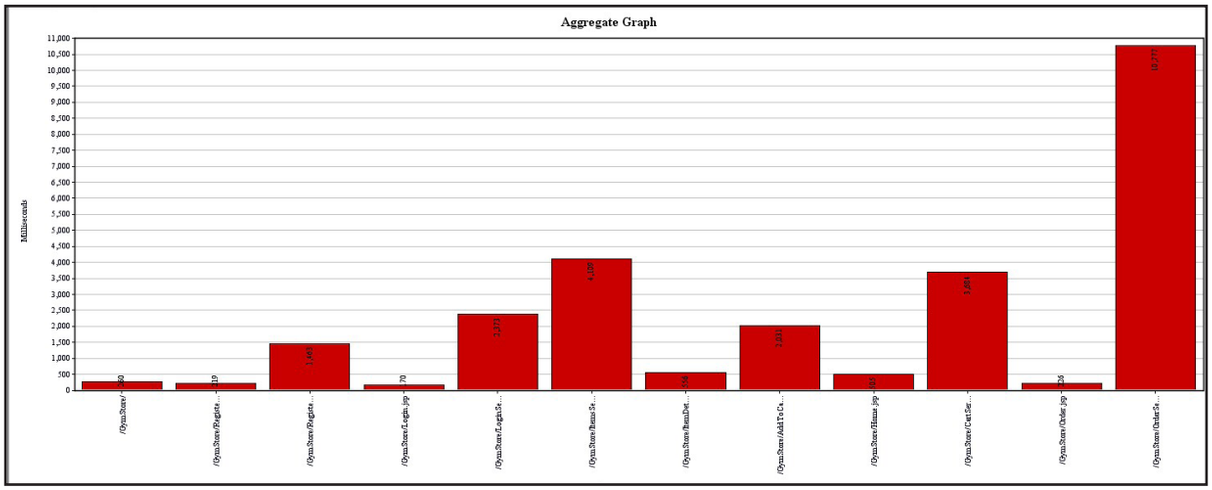
Source: Apache Jmeter version 2.10

Fig. 9: Graphical Representation of Performance Test Results for 100 Users Post Improvement



Source: Apache Jmeter version 2.10

Fig. 10: Graphical Representation of Performance Test Results for 200 Users Post Improvement



Source: Apache Jmeter version 2.10

Fig. 11: Graphical Representation of Performance Test Results for 300 Users Post Improvement

E. Comparison of the Performance Test Results Before and After Improvement

It is clearly seen that the average response time of 17724.00 milliseconds for Order servlet has now dropped to 9067.33 milliseconds, an improvement of 48.84%.

Also observed that the average time for the complete performance test has dropped from 36625.00 milliseconds to 19700.33 milliseconds, thus making an improvement of 46.21% overall.

Thus the overall performance of the application has been enhanced. It is thus observed that bulk database inserts save a lot of time if the amount of data to be inserted is large as compared to the individual inserts.

V. Conclusion

In this paper we developed a Java web application, evaluated its performance and improved the same. How well a web application responds to user requests especially when there is large concurrent user load, determines the customer's satisfaction from that application. Considering the importance and criticality a web application's performance holds, performance testing and improvement becomes a pivotal step in the web application development process. Techniques of query optimization were used to improve the performance in this paper; additional techniques like load sharing using multiple servers can be harnessed to further improve the performance.

References

[1] R. Khan, and M. Amjad, "Performance testing (load) of web applications based on test case management," *Perspectives in Science*, vol. 8, pp. 355-357, September 2016.

[2] S. Dhiman, and P. Sharma. "Performance testing: A comparative study and analysis of web service testing tools," *International Journal of Computer Science and Mobile Computing,* vol. 5, no. 6, pp. 507-512, June 2016.

[3] Rina, and S. Tyagi, "Comparative study of performance testing tools," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 5, pp. 1300-1307, May 2013.

[4] D. Draheim, J. Grundy, J. Hosking, C. Lutteroth, and G. Weber, "Realistic load testing of web applications," *Proceedings of the Conference on Software Maintenance and Reengineering (CSMR'06),* 2006.

[5] A. Ali, and N. Badr, "Performance testing as a service for web applications," *Proceedings of the Seventh International Conference on Intelligent Computing and Information Systems (ICICIS'15),* 2015.

[6] S. Sharmila, and E. Ramadevi, "Analysis of performance testing on web applications," *International Journal of Advance Research in Computer and Communication Engineering,* vol. 3, no. 3, pp. 5258-5260, March 2014.

[7] H.-L. Zhang, S. Zhang, X.-J. Li, P. Zhang, and S.-B. Liu, "Research of load testing and result application based on LoadRunner," *Proceedings of National Conference on Information Technology and Computer Science (CITCS 2012),* 2012.

[8] O. Hamed, and N. Kafri, "Performance testing for web based application architectures (.NET vs. Java EE)," *Proceedings of First International Conference on Networked Digital Technologies,* 2019.

[9] R. Kumar, and K. Bhargav, "A survey on performance testing approaches of web application and importance of WAN simulation in performance testing," *International Journal on Computer Science and Engineering*, vol. 4, no. 5, pp. 883-885, May 2012.

[10] Apache Software Foundation, *Apache JMeter.* [Online]. Available: https://jmeter.apache.org/