Performance Testing and Evaluation of Web Applications Using WAPT Pro

Dr. Deepak Dagar[1], Dr. Amit Gupta[2]

Assistant Professor, Department of management, Maharaja Agrasen Institute of Management Studies, Rohini,

New Delhi, India [1]

Associate Professor, Department of management, Maharaja Agrasen Institute of Technology, Rohini, New Delhi,

India[2]

ABSTRACT: Web Applications are widely known as the building blocks of typical service oriented applications. Performance of such an application system is mainly dependent upon the components of web applications. Testing web application is nothing but to find out errors in its content, function, usability, navigability, performance, capacity, and security.
Performance testing is a used to determine the responsiveness, throughput, reliability, and/or scalability of a system under a given workload. This paper presents performance testing features, types and tools for testing web applications performance.

KEYWORDS: TPS, TTPS, CPS, PDPS.

I. INTRODUCTION

1. Performance Testing
The testing performed to determine the degree to which a system or component accomplishes its designated functions within given constraints regarding processing time and throughput rate."
The purpose of the test is to measure characteristics, such as response times, throughput or the mean time between failures (for reliability testing).

II. PERFORMANCE TESTING TOOL

A tool to support performance testing and that usually has two main facilities: load generation and test transaction measurement. Load generation can simulate either multiple users or high volumes of input data. During execution, response time measurements are taken from selected transactions and these are logged. Performance testing tools normally provide reports based on test logs and graphs of load against response times.
Features or characteristics of performance-testing tools include support for:
• generating a load on the system to be tested;
• measuring the timing of specific transactions as the load on the system varies;
• measuring average response times;
• producing graphs or charts of responses over time.

III. LOAD TEST

A test type concerned with measuring the behavior of a component or system with increasing load, e.g. number of parallel users and/or numbers of transactions to determine what load can be handled by the component or system.
While doing Performance testing we measure some of the following:

Characteristics (SLA)	Measurement (units)
Response Time	Seconds
Hits per Second	#Hits
Throughput	Bytes Per Second
Transactions per Second (TPS)	#Transactions of a Specific Business Process
Total TPS (TTPS)	Total no. of Transactions
Connections per Second (CPS)	#Connections/Sec
Pages Downloaded per Second PDPS)	#Pages/Sec

Table 1.1

IV.   OBJECTIVE OF PERFORMANCE TESTING

The objective of a performance test is to ensure that the application is working perfectly under load. However, the definition of "perfectly" under load may vary with different systems.
By defining an initial acceptable response time, we can benchmark the application if it is performing as anticipated.

V.   CHARACTERISTICS OF PERFORMANCE TESTING

The importance of Transaction Response Time is that it gives the project team/ application team an idea of how the application is performing in the measurement of time. With this information, they can relate to the users/customers on the expected time when processing request or understanding how their application performed.

5.1 Transaction Response time
The Transaction Response Time encompasses the time taken for the request made to the web server, there after being process by the Web Server and sent to the Application Server. This in most instances will make a request to the Database Server. All this will then be repeated again backward from the Database Server, Application Server, Web Server and back to the user. Take note that the time taken for the request or data in the network transmission is also factored in.The following diagram illustrates Transaction Response Time.
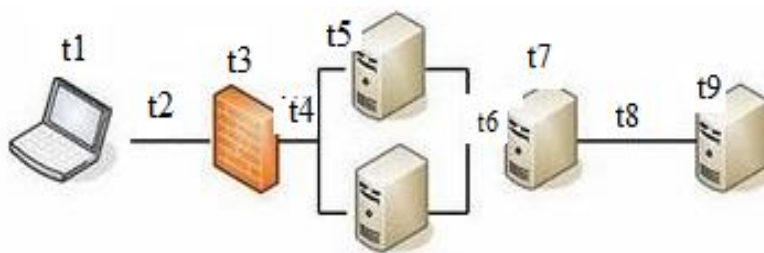

Fig 1.1
Transaction Response Time = (t1 + t2 + t3 + t4 + t5 + t6 + t7 + t8 + t9) X 2
Transaction Response

Time allows us to identify abnormalities when performance issues surface. This will be represented as slow response of the transaction, which differs significantly (or slightly) from the average of the Transaction Response Time.
With this, we can further drill down by correlation using other measurements such as the number of virtual users that is accessing the application at the point of time and the system-related metrics (e.g. CPU Utilization) to identify the root cause.

Bringing all the data that have been collected during the load test, we can correlate the measurements to find trends and bottlenecks between the response time, the amount of load that was generated and the payload of all the components of the application. Using Transaction Response Time, Project Team can better relate to their users using transactions as a

form of language protocol that their users can comprehend. Users will be able to know that transactions (or business processes) are performing at an acceptable level in terms of time. So, it is very important when looking at this metric to consider what and how the application is intended to work.

a.   Hits per second
A virtual client can request an HTML page, image, file, etc. Testing the application for Hits Per Second will tell you if there is a possible scalability issue with the application. For example, if the stress on an application increases but the Hits Per Second does not, there may be a scalability problem in the application. It is possible that out of a hundred hits on the application, the application server actually answered only one and all the rest were either cached on the web server or other caching mechanism.

5.3 Pages per second
Pages per Second measures the number of pages requested from the application per second. The higher the Page per Second the more work the application is doing per second. Measuring an explicit request in the script or a frame in a frameset provides a metric on how the application responds to actual work requests. Thus if a script contains a Navigate command to a URL, this request is considered a page. If the HTML that returns include frames they will also be considered pages, but any other elements retrieved such as images or JS Files, will be considered hits, not pages. This measurement is key to the end-user's experience of application performance.

5.4 Throughput
The amount of data transferred across the network is called throughput. It considers the amount of data transferred from the server to client only and is measured in Bytes/sec. This is an important baseline metric and is often used to check that the application and its server connection is working. Throughput measures the average number of bytes per second transmitted from the application being tested to the virtual clients running the test agenda during a specific reporting interval

5.5 Round Trips
Another useful scalability and performance metric is the testing of Round Trips. Round Trips find the total number of times the test agenda was executed versus the total number of times the virtual clients attempted to execute the Agenda. The more times the agenda is executed, the more work is done by the test and the application. The test scenario the agenda represents influences the round Trips measurement. This metric can provide all kinds of useful information from the benchmarking of an application to the end-user availability of a more complex application.

5.6 Hit Time
Hit time is the average time in seconds it took to successfully retrieve an element of any kind (image, HTML, etc). The time of a hit is the sum of the Connect Time, Send Time, Response Time and Process Time. It represents the responsiveness or performance of the application to the end user. The more stressed the application, the longer it should take to retrieve an average element. But, like Hits Per Second, caching technologies can influence this metric. Getting the most from this metric requires knowledge of how the application will respond to the end user.

5.7 Time to first byte
This measurement is important because end users often consider a site malfunctioning if it does not respond fast enough. Time to First Byte measures the number of seconds it takes a request to return its first byte of data to the test software's Load Generator. Time to First Byte represents the time it took after the user pushes the "enter" button in the browser until the user starts receiving results. Generally, more concurrent user connections will slow the response time of a request. But there are also other possible causes for a slowed response.

5.8 Page Time
Page Time calculates the average time in seconds it takes to successfully retrieve a page with all of its content. This statistic is similar to Hit Time but relates only to pages. In most cases this is a better statistic to work with because it deals with the true dynamics of the application. Since not all hits can be cached, this data is more helpful in terms of tracking a user's experience (positive or frustrated). It's important to note that in many test software application tools you can turn caching on or off depending on your application needs.

5.9 Failed Rounds/Failed Rounds per Second

During a load test it's important to know that the application requests perform as expected. The Failed Rounds and Failed Rounds per Second tests the number of rounds that fail. Sometimes, basic image or page missing errors (HTTP 404 error codes) could be set to fail a round, which would stop the execution of the test agenda at that point and start at the top of the agenda again, thus not completing that particular round.

VI. WEB APPLICATION ARCHITECTURE

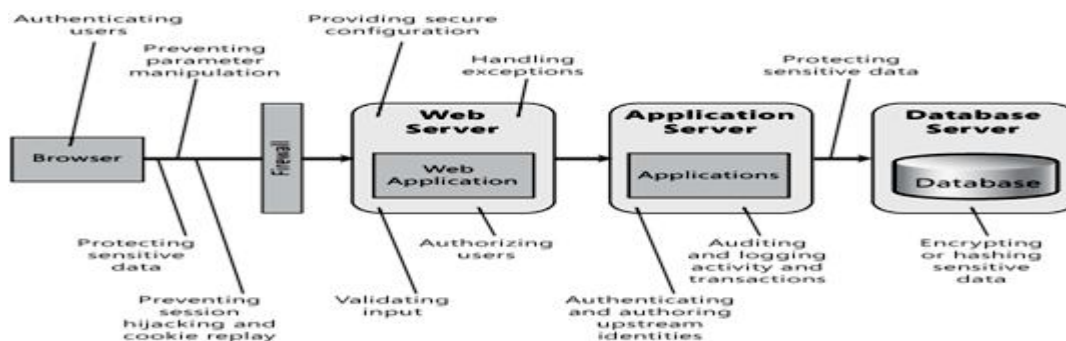A Web application is an application that can be accessed by the users through a Web browser or a specialized user

Fig 1.2

agent. The browser creates HTTP requests for specific URLs that map to resources on a Web server. The server renders and returns HTML pages to the client, which the browser can display. The core of a Web application is its server-side logic. The application can contain several distinct layers. The typical example is a three-layered architecture comprised of presentation, business, and data layers. Figure 1.2 illustrates a typical Web application architecture with common components grouped by different areas of concern.

VII. PERFORMANCE TESTING DONE ON FLIPKART.COM

Summary

Profile	Successful sessions	Failed sessions	Successful pages	Failed pages	Successful hits	Failed hits	Total KBytes sent	Total KBytes received	Avg response time, sec (with page elements)
Profile_flipkart.com_july5	8	109	1780	109	6894	140	3974	85322	0.98(2.78)

Successful sessions (Failed sessions)

Profile	0:00:00-0:01:00	0:01:00-0:02:00	0:02:00-0:03:00	0:03:00-0:04:00	0:04:00-0:05:00	0:05:00-0:06:00	0:06:00-0:07:00	0:07:00-0:08:00	0:08:00-0:09:00	0:09:00-0:10:00	Total
Profile_flipkart.com_july5	0(0)	0(0)	3(0)	2(9)	2(5)	0(20)	0(41)	1(9)	0(9)	0(16)	8(109)

Successful pages (Failed pages)

Profile	0:00:00-0:01:00	0:01:00-0:02:00	0:02:00-0:03:00	0:03:00-0:04:00	0:04:00-0:05:00	0:05:00-0:06:00	0:06:00-0:07:00	0:07:00-0:08:00	0:08:00-0:09:00	0:09:00-0:10:00	Total
Profile_flipkart.com_july5	73(0)	232(0)	385(0)	131(9)	251(5)	26(20)	2(41)	299(9)	292(9)	89(16)	1780(109)

Successful hits (Failed hits)

Profile	0:00:00-0:01:00	0:01:00-0:02:00	0:02:00-0:03:00	0:03:00-0:04:00	0:04:00-0:05:00	0:05:00-0:06:00	0:06:00-0:07:00	0:07:00-0:08:00	0:08:00-0:09:00	0:09:00-0:10:00	Total
Profile_flipkart.com_july5	308(0)	825(0)	1382(0)	773(14)	934(8)	100(25)	22(50)	1200(14)	845(9)	505(20)	6894(140)

Total errors on pages (hits) as a % of all completed pages (hits)

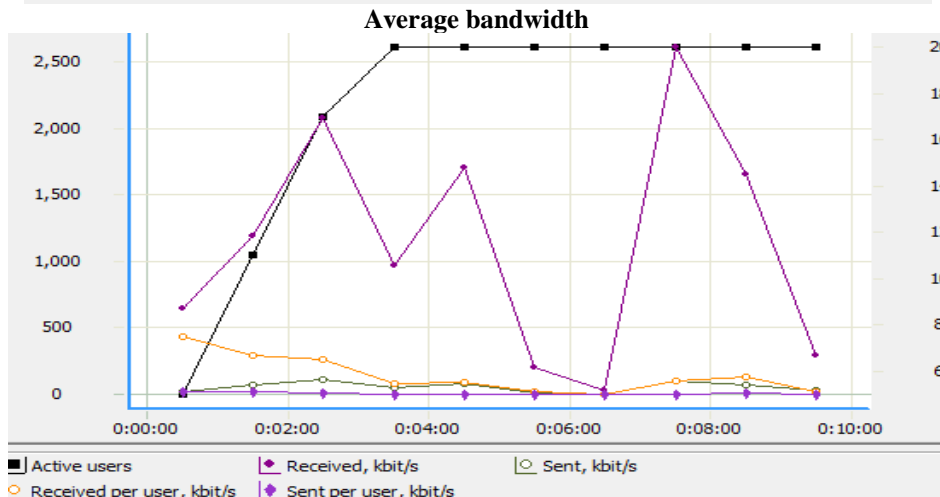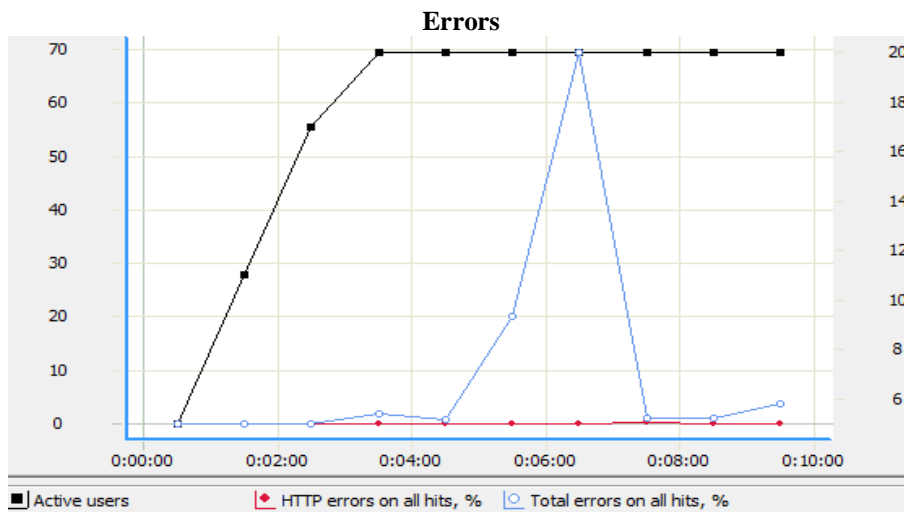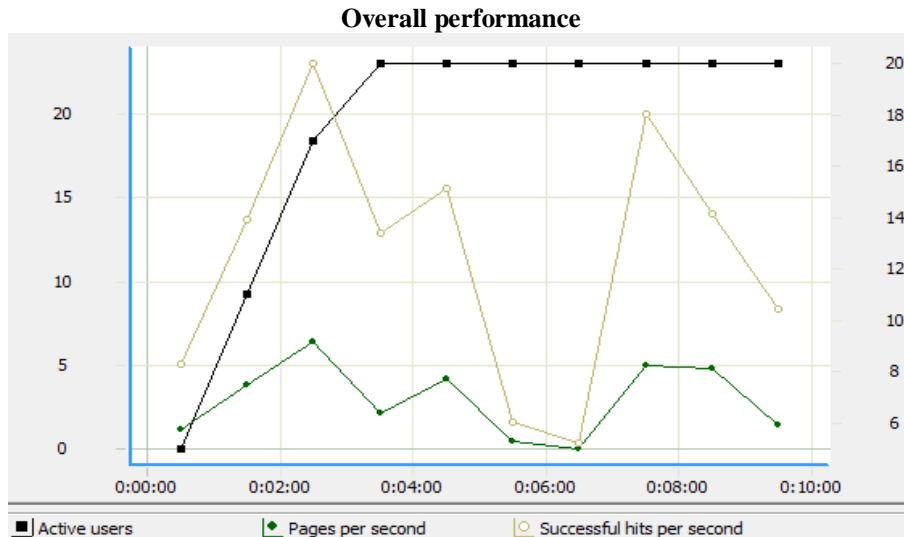| Profile	0:00:00-0:01:00	0:01:00-0:02:00	0:02:00-0:03:00	0:03:00-0:04:00	0:04:00-0:05:00	0:05:00-0:06:00	0:06:00-0:07:00	0:07:00-0:08:00	0:08:00-0:09:00	0:09:00-0:10:00	Total
Profile_flipkart.com_july5	0(0)	0(0)	0(0)	6.43 (1.78)	1.95 (0.85)	43.5 (20)	95.3 (69.4)	2.92 (1.15)	2.99 (1.05)	15.2 (3.81)	5.77 (1.99)

Load agent utilization, %

Name	Utilization	0:00:00-0:01:00	0:01:00-0:02:00	0:02:00-0:03:00	0:03:00-0:04:00	0:04:00-0:05:00	0:05:00-0:06:00	0:06:00-0:07:00	0:07:00-0:08:00	0:08:00-0:09:00	0:09:00-0:10:00	Total
localhost	CPU	12	14	18	18	20	11	10	19	18	16	15
	Memory Mb(%)	100(4)	102(4)	115(5)	124(6)	124(6)	124(6)	124(6)	124(6)	124(6)	124(6)	118(5)
	Network	0	0	1	1	1	0	0	1	1	0	0

Overall performance

Errors

Average bandwidth

Average response time (without page elements)
Profile_flipkart.com_july5.page_1: http://www.flipkart.com/

■ Active users   ♦ Avg response time, sec   ○ Avg 90% response time, sec
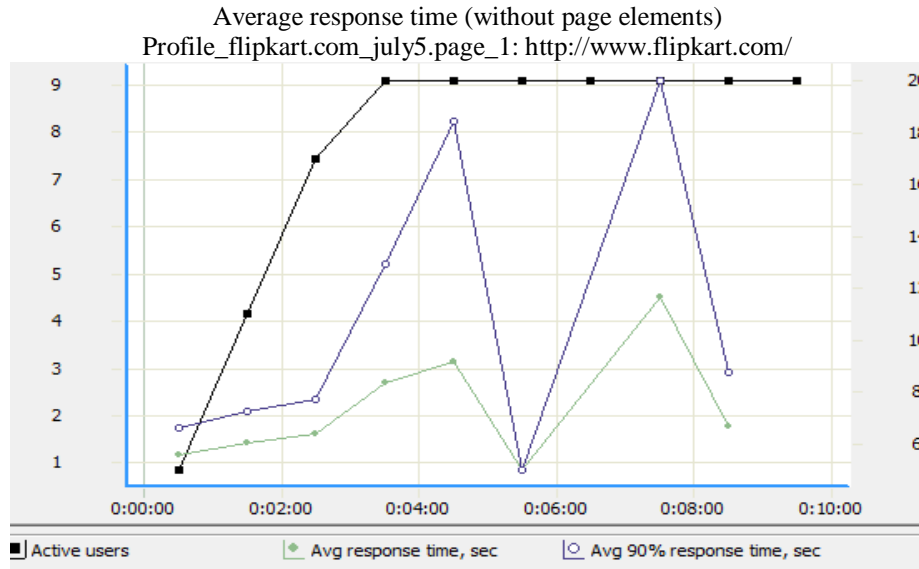
VII.   PERFORMANCE TESTING USING WAPT PRO (SNAPDEAL.COM)

Summary

Profile	Successful sessions	Failed sessions	Successful pages	Failed pages	Successful hits	Failed hits	Total KBytes sent	Total KBytes received	Avg response time, sec (with page elements)
Profile_snapdeal_5july	0	305	286	305	3661	415	1562	100524	1.95(26.2)

Successful sessions (Failed sessions)

Profile	0:00:00-0:01:00	0:01:00-0:02:00	0:02:00-0:03:00	0:03:00-0:04:00	0:04:00-0:05:00	0:05:00-0:06:00	0:06:00-0:07:00	0:07:00-0:08:00	0:08:00-0:09:00	0:09:00-0:10:00	Total
Profile_snapdeal_5july	0(8)	0(28)	0(43)	0(35)	0(31)	0(34)	0(24)	0(32)	0(38)	0(32)	0(305)

Successful pages (Failed pages)

Profile	0:00:00-0:01:00	0:01:00-0:02:00	0:02:00-0:03:00	0:03:00-0:04:00	0:04:00-0:05:00	0:05:00-0:06:00	0:06:00-0:07:00	0:07:00-0:08:00	0:08:00-0:09:00	0:09:00-0:10:00	Total
Profile_snapdeal_5july	8(8)	28(28)	44(43)	36(35)	23(31)	34(34)	19(24)	25(32)	39(38)	30(32)	286(305)

Successful hits (Failed hits)

Profile	0:00:00- 0:01:00	0:01:00- 0:02:00	0:02:00- 0:03:00	0:03:00- 0:04:00	0:04:00- 0:05:00	0:05:00- 0:06:00	0:06:00- 0:07:00	0:07:00- 0:08:00	0:08:00- 0:09:00	0:09:00- 0:10:00	Total
Profile_snapdeal_5 july	151(8)	384(32)	621(43)	423(48)	312(44)	436(48)	257(38)	281(58)	456(50)	340(46)	3661(415)

Successful sessions per second

Profile	0:00:00- 0:01:00	0:01:00- 0:02:00	0:02:00- 0:03:00	0:03:00- 0:04:00	0:04:00- 0:05:00	0:05:00- 0:06:00	0:06:00- 0:07:00	0:07:00- 0:08:00	0:08:00- 0:09:00	0:09:00- 0:10:00	Total
Profile_snapdeal_5july	0	0	0	0	0	0	0	0	0	0	0

Successful pages per second

Profile	0:00:00- 0:01:00	0:01:00- 0:02:00	0:02:00- 0:03:00	0:03:00- 0:04:00	0:04:00- 0:05:00	0:05:00- 0:06:00	0:06:00- 0:07:00	0:07:00- 0:08:00	0:08:00- 0:09:00	0:09:00- 0:10:00	Total
Profile_snapdeal_5july	0.13	0.47	0.73	0.60	0.38	0.57	0.32	0.42	0.65	0.50	0.48

Successful hits per second

Profile	0:00:00- 0:01:00	0:01:00- 0:02:00	0:02:00- 0:03:00	0:03:00- 0:04:00	0:04:00- 0:05:00	0:05:00- 0:06:00	0:06:00- 0:07:00	0:07:00- 0:08:00	0:08:00- 0:09:00	0:09:00- 0:10:00	Total
Profile_snapdeal_5july	2.52	6.40	10.4	7.05	5.20	7.27	4.28	4.68	7.60	5.67	6.10

Total errors on pages (hits) as a % of all completed pages (hits)

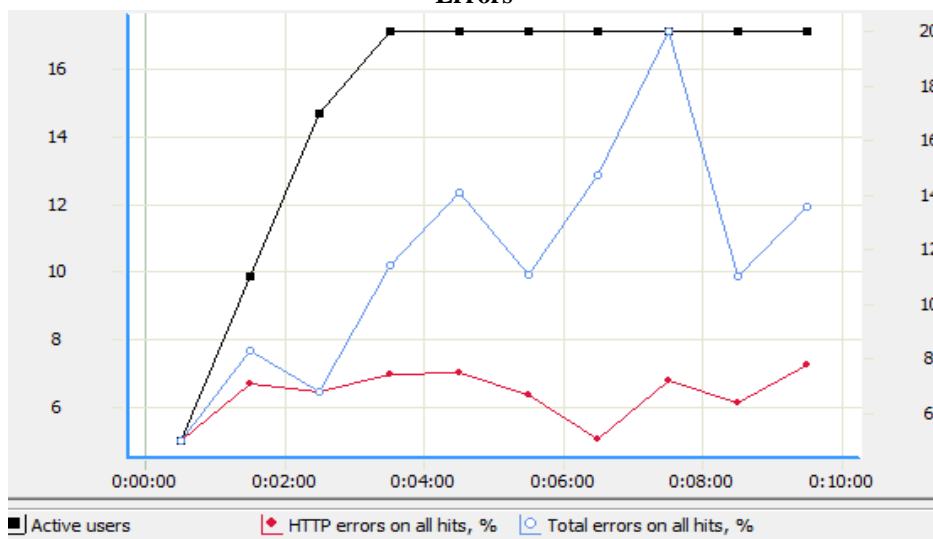| Profile	0:00:00- 0:01:00	0:01:00- 0:02:00	0:02:00- 0:03:00	0:03:00- 0:04:00	0:04:00- 0:05:00	0:05:00- 0:06:00	0:06:00- 0:07:00	0:07:00- 0:08:00	0:08:00- 0:09:00	0:09:00- 0:10:00	Total
Profile_snapdeal_5july	50(5.03)	50(7.69)	49.4(6.48)	49.3(10.2)	57.4(12.4)	50(9.92)	55.8(12.9)	56.1(17.1)	49.4(9.88)	51.6(11.9)	51.6(10.2)

Load agent utilization, %

Name	Utilization	0:00:00-0:01:00	0:01:00-0:02:00	0:02:00-0:03:00	0:03:00-0:04:00	0:04:00-0:05:00	0:05:00-0:06:00	0:06:00-0:07:00	0:07:00-0:08:00	0:08:00-0:09:00	0:09:00-0:10:00	Total
localhost	CPU	34	38	38	34	38	27	30	28	19	14	30
	Memory Mb(%)	91(4)	95(4)	104(4)	109(5)	109(5)	109(5)	109(5)	109(5)	109(5)	109(5)	105(4)
	Network	0	1	2	0	1	1	1	0			

Overall Performance
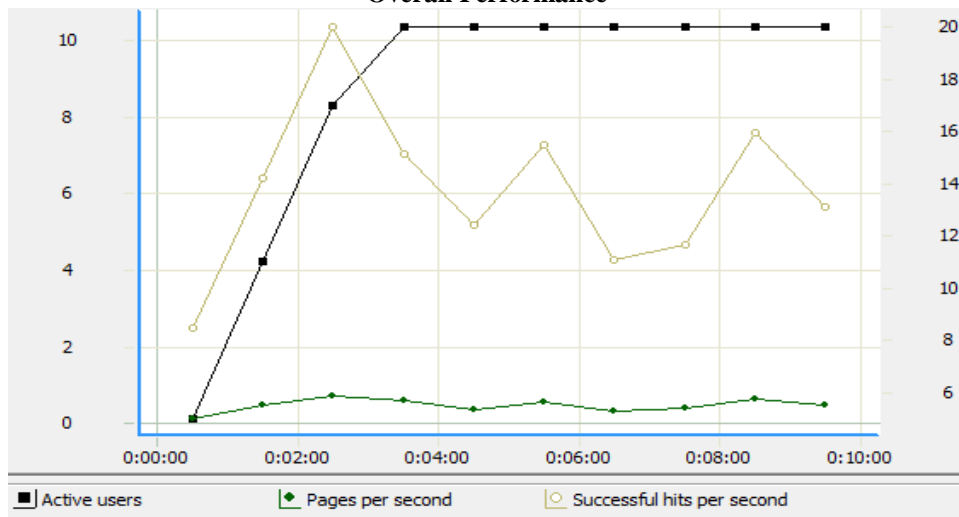
Errors

Average bandwidth

Average response time (without page elements)
Profile_snapdeal.page_1: http://www.snapdeal.com/
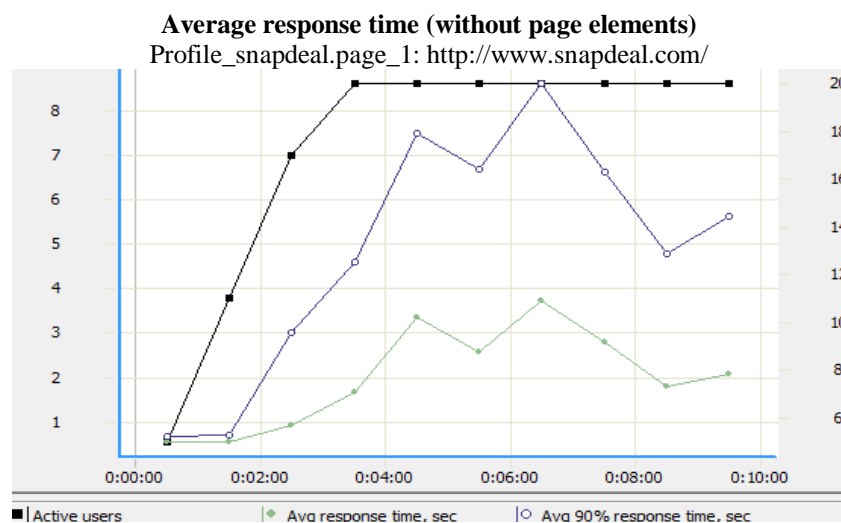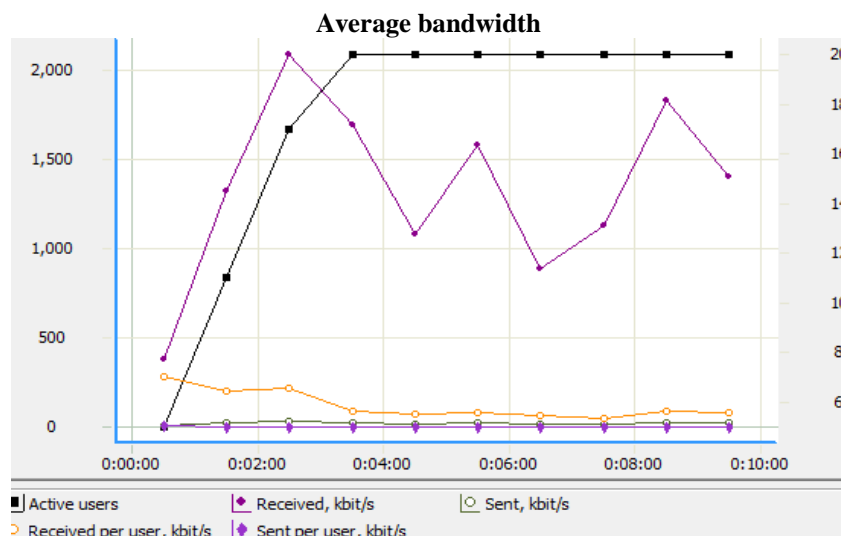
IX. RESULT AND CONCLUSION

Performance Criteria	Flipkart	Snapdeal
Successful Session	8	0
Failed Session	109	305
Successful Pages	1780	286
Failed Pages	109	305
Successful Hits	6894	3661
Failed Hits	140	415
Total KB Sent	3974	1562
Total KB Received	85322	100524
Average Response Time	0.98	1.95
Load Utilization		
CPU	15	30
Memory	118	105
N/w	0	0

The result based on the comparisons of performance testing conducted using WAPT shows that both the website shows good performance in terms of overall performance, average response time, average bandwidth and error rate. However, WAPT pro shows good result for flipkart in terms of successful hits, failed hits, average response time and CPU Utilization.

REFERENCES

1. Shariq Hussain, Zhaoshun Wang, Ibrahima Kalil Toure and Abdoulaye Diop, "Web Service Testing Tools: A Comparative Study", IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 1, No 3, January 2013.
2. VCAA (2005), Stress, Load, Volume, Performance, Baseline Testing Tool Evaluation and comparison [Online].Available:http://www.vcaa.com/tools/loadtesttoolevaluationchart-023.pdf.
3. AgileloadBlog (2012), Application Performance testing basics [Online].Available:http://www.agileload.com/agileload//blog/2012/11/09/application-performance-testing-basics.\
4. Lakshmi Narayan (2011), Overview of Performance Testing Concepts[Online].Available:http://nonfunctionaltestingtools.blogspot.in/2011/01/overview-of-performance-testing.html
5. Dr. S. M. Afroz, N. Elezabeth Rani and N. Indira Priyadarshini, "Web Application– A Study on Comparing
6. Software Testing Tools", International Journal of Computer Science and Telecommunications, Volume 2, Issue 3, June 2011.
7. IDS Australia (2009), A comparison of Open Source Load Testing tools [On line].Available: http://www.jds.net.au/tech-tips/load testing tool comparison.
8. Vinod, P. (2008), Open Source & Commercial Performance Testing Tools [Online].Available:
9. http://www.scribd.com/doc/27765939/Performance -Test-Tools-Comparison.
10. Kualitatem (2012), Comparison between HP, IBM and APACHE –Performance Testing Tools
11. [Online].Avialable:http://www.kualitatem.com/comparison-between-performance-testing-tool/.
12. [9] VCAA (2005), Stress, Load, Volume, Performance, Baseline Testing Tool Evaluation and comparison
13. [Online].Available:http://www.vcaa.com/tools/loadtesttoolevaluationchart-023.pdf.