Remodel a Workload Profile for Web Application Performance Testing

C.Prathiba[1], T.Supriya[2]and N.Ezhil Arasu[3]

**[1, 2] Department of Computer Science & Engineering, Agni College of Technology,
Chennai ,Tamil Nadu, India – 603103**

**[3]Performance / Security Consultant, General Motors VSSM,
Capgemini, Chennai, Tamil Nadu, India -600096**

***Abstract*—Re modeling a current workload profile for a web application will provide the Business owners to match the Virtual user load to their present load condition. This section is detailing the number of virtual users, the scenario path, and kind of test data used in Performance Testing. The statistics derived from Web server access logs, Google analytics /Omniture data, DB transaction numbers will be really helpful to remodel the existing load profile.**

***Keywords*—Software Performance, Workload Profile, Load Model, Performance Test Data, Performance Evaluation, Performance Modeling**

I. INTRODUCTION

Web applications are now providing many solutions to all kind of customers in almost all industry, students, common peoples , government and corporate offices. Due to these kind of end users web applications need to be fast, reliable & availability. To meet these requirements, all the web applications need to be tested against their performance under load before released to the end user. For a small application with maximum of 5 or 10 users can be simulated with real users. Here also the time measurement will not be accurate. To do this all web applications are tested against simulated Virtual users..

To answer the question of number of Virtual users, we are relying on the business to get the expected load on the system. This paper will address how this assumption be validated against real numbers. This paper is applicable only for systems available on Production and used by peoples.

In this paper we are discussing the following 1. Number of real world uses accessing the application 2. Entry and Exit path of critical transactions 3. The amount of data (in number of records or bytes of data or number of lines) used by the real world users. By getting these data from the production system, we are comparing against the tested numbers in Pre production system. Based on these the existing load profile is re modeled so that to match with Real world Load[6] on the applicationSection II will determine the process of redefining the workflow, Section III will determine the process of redefining the number of users, section IV will give idea aboutredefine the functional scenario to the transactions, section V will give the idea about tuning the test data, Section VI explains the methodology used to collect the data, Section VII will give the advantages of doing this exercise.

II. REMODEL WORK FLOW

Work load profile [1][5] of an application under test will contain the following data. 1. Number of transactions or

Number of users / Transaction, 2. Response time 3.Failure Rate, 4. Test Scenario Document and 5. Test Data for performance Testing. To redefine the work load profile, we need to do the following processes. 1. Get the required statistics from the production system 2. Compare and analyze with the existing tested load profile. 3. Modify the Load profile based on step 2 4. Re test the application performance with the new load profile (most of the case not in production environment) 5. Analyze the results to identify the bottleneck 6. Tune thecode[7] or add infrastructure to resolve bottleneck (assuming +ve change in load profile[6]) 7. Test and deploy the changes to production environment. Refer figure II.1

This process need to run every six months or year to keep the application tuned to the present load conditions. These can be taken care along with other change requests to the applications. This process will make sure no performance degradations [2] over period time and make sure the end user to have a happy experience on the application. This process will ensure the application up to date with realistic user load and address the performance bottlenecks.

This paper insisting a constant observation of application after deployment to production environment for Performance and failures.Most of the time, organizations will find or target only few transaction performance, and never worried about the other transactions. This model will care all transactions and keep the organizations updated on the real profile.

This model is addressing not only the user load, but it is taking care of scenario tuning and test data tunings also. Based on this analysis , the performance test scenarios and test data also changed to match with production data. For example, earlierperformance tests done for a search transaction to retrieve less than 10 orders. But the production test data clarifies that 70% of users will retrieve at least 25 orders per search. This test data need to be taken care . Similar to that , we can get the peak load duration in a day, what are the functional areas will be most used during peak load on a given day.

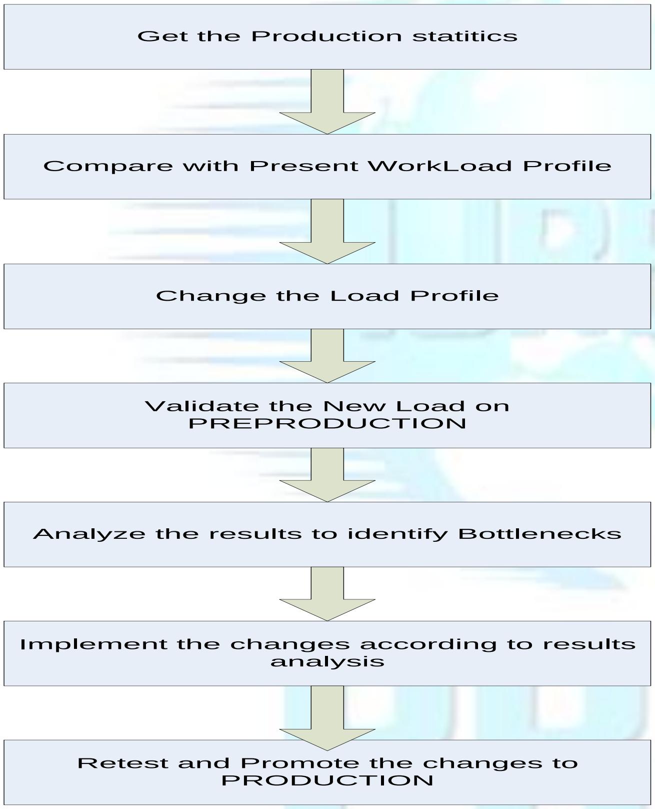

Get the Production statitics
Compare with Present WorkLoad Profile
Change the Load Profile
Validate the New Load on PREPRODUCTION
Analyze the results to identify Bottlenecks
Implement the changes according to results analysis
Retest and Promote the changes to PRODUCTION

Fig. 1.Process flow for Work load Profile remodeling.

III. REDEFINE USER /TRANSACTION NUMBERS

The main part to the load model[4] is .number of users / Transaction or number of transactions / hour [7]. This data can be gathered by analyzing web server logs or Google analytics / Omnituredata or by gathering Database update details

Suggestion to get the data for at least six months or one year data.This will help us to identify the busiest month of a year. The next goal to identify the busiest day in a year. Here we will face a conflict. Some of the business functions will be in peak in different days. For example a Delivery transaction will be busy in Month end, Salary related transactions will be busy in starting of every month.

To address this, we need to find the Peak load day per transaction wise not by the whole hits for a website. Since if a user selected 15th of June as a peak day which may have less number of Vehicle delivery transactions (since Vehicle delivery will be busy only in month end). So every transaction will have its own peak load day. Some of the transactions may be grouped into functional area level to reduce work.

Next goal is to identify the peak transaction per hour data. Here are the options available to pick up.

1. Peak number from Peak hour in the peak day
2. 90th Percentile of data
3. Average of maximum N numbers.

Approach 1 Peak Number from Peak Hour in the Peak day will be an aggressive approach which yields always have higher value. There may be many reasons for a transaction to go peak in an hour of peak day. Thus we are considering only one hour data from one year which may not be accurate and over predicted. This will be more than a peak load and less than or equal to double peak load. I told this is aggressive, since this will be a worst hour in a year of time.

Approach 2 90th percentile of peak data will take the 90th percentile value of all peak hours in a year. This will be close to the peak value. If this value is too large with average value, we can go for 80th percentile. This number will make sure the number we got is more than 90 percentage of time and less than the number we got from approach 1.

Approach 3 Average of N numbers is taking the average of hourly hits over one year. This average should exclude non peak days like holidays, Sundays and night time and system down times. Otherwise the calculated number will be very less.

For all these calculations, we should exclude non business hours, holidays and Sundays. Approach 1 will not be affected, but approach 2 & Approach 3 need theses exclusions to depict correct value. At the end of this process, we will get the number of transactions in a peak hour. In addition to that we got the peak days for different functional areas and peak month of a year.
Based on the number of transactions per hour, Performance engineers can calculate the required number of Virtual users needed to simulate the load.

IV. REDEFINE PERFORMANCE TEST SCENARIOS

Performance Test scenario[5] is application usage scenarios. During the initial design of load profile, these inputs are taken from the business analysts. These are the inputs for creating a Virtual user script and core part of the performance Test execution.

This information should be accurate and it should cover all critical valid paths to a transaction. For example in Indian railways ticket booking system, there are two valid paths to book a ticket. One is Plan My Travel and the other is Quick Book Ticket. A performance engineer should divide the Ticket booking transaction load between these scenarios. This number has to be arrived from business Analysts initially.

Apart from that, any additional sub flow in the scenario path need to be discussed with the stake holders to ensure or exclude in the Performance testing scenario.

Now from production data, we may get this statistics from Entry / Exit path[2],[4]to a transaction metrics. This will give the exact division of load between different paths to a same transaction. In addition to that we can identify the most used and least used sub flows to take a decision of Exclude / include them on the revised scenario.

This also highlights any missed flow / Transaction in the existing load profile.

This will helpful in getting the below metrics & Suggestions

1. Suggesting the Most used & Least used Flows for a same transaction
2. It identifies any missed scenario that has been used significantly in the production
3. It tells about what can be excluded and what can be added into the scenario.

V. REDEFINE PERFORMANCE TEST DATA

Performance test data[3] is the data set used for Virtual users during an initial load test. By carefully understanding the statistics from production application, we can redefine the test data to an accurate one. Performance Test data is the important during Performance Test runs since it controls the throughput, and the performance of the back end systems. It also creates much effect on the application server in case of xml to xslt conversion or any kind of sorting done by the client side.

For Example, if the Performance test data always create a train ticket with one passenger detail, but in real time scenario, an average of 3 passenger /ticket booked and the 90th percentile would be 4 passenger /ticket.

In this case, we are not generating the exact load on the backend systems while writing the ticket details to the Database. Similarly while retrieving the ticket details also the read is only retrieving single line detail. But the real time scenario is not like that and it would take considerable time to book the ticket with 3 lines of detail. Here are the data we got from the production data:

1. 55% of tickets booked with 3 lines of passenger detail
2. 25% of tickets booked with 4 or more lines of passenger Detail
3. 20% of tickets booked with less than 3 lines of passenger detail.

Based on this data we can go with all performance Test data need to be adjusted to write 3 lines of ticket. The optimal solution should be using the same mixture of data file in the testing also.. The parameter file used for Virtual users also need to be have the same proportion.

Here is another example with Indian Railways Ticket Booking system:

Here we assumed that during initial performance test, the test performed with mixture of 70% of General tickets and 30% of Tatkal Tickets. But the statistics we got from production system is as follows:

1. Peak hour is between 10 AM to 11 AM (90% of time)
2. Peak hour between 8 AM to 9 AM (8% of time)
3. Peak hour between 9 AM to 10 AM (1% of time) and so on…

Now it’s clear that the peak load generated only by Tatkal ticket booking which has different flow and restrictions compared to general ticket booking such as number of lines, opening time and number of ticket limitations/user. To optimize the performance Test data based on these metrics the parameter file need to change to load 90% of Tatkal tickets and 10% of General tickets.

VI. DATA COLLECTING METHODS

To collect the statistics from the production system, many methods are available. Here are the few important to discuss

1. Web server Access Log
2. Omniture / Google Analytics data [8][10]
3. DB transaction details

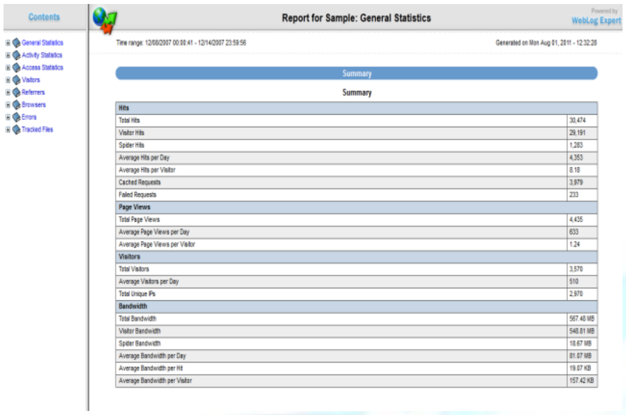
A. Web Server Access Log

There are more tools available in the market to analyze the webserver access logs. Access logs containing the information about URL, HTTP Response code& time . Analyzing this log file with web log analysis programs will give us the idea about how many users using the web site transactions, what was the busiest hour /week /day/month in a year. Apart from that we can find many useful information about http response codes, error & Failure rates, Top 50 URL’s, Content size, average duration of a user, entry & exit path and many. I am mentioning few of the tools in interest here:

1. Web Log Analyzer (Freeware) [11]
2. Sawmill (Extensive but licensed)[12]
3. Web Log expert[13]

4. Webalizer[14]

These are few but you can search and found many of these kinds.



Report for Sample: General Statistics

Time range: 12/08/2007 00:00:41 - 12/14/2007 23:59:56

Generated on Mon Aug 01, 2011 - 12:32:28

Summary

Hits	
Total Hits	30,474
Visitor Hits	29,191
Spider Hits	1,283
Average Hits per Day	4,353
Average Hits per Visitor	8.18
Cached Requests	3,979
Failed Requests	233
Page Views	
Total Page Views	4,435
Average Page Views per Day	633
Average Page Views per Visitor	1.24
Visitors	
Total Visitors	3,570
Average Visitors per Day	510
Total Unique IPs	2,970
Bandwidth	
Total Bandwidth	567.48 MB
Visitor Bandwidth	548.81 MB
Spider Bandwidth	18.67 MB
Average Bandwidth per Day	81.07 MB
Average Bandwidth per Hit	19.07 KB
Average Bandwidth per Visitor	157.42 KB

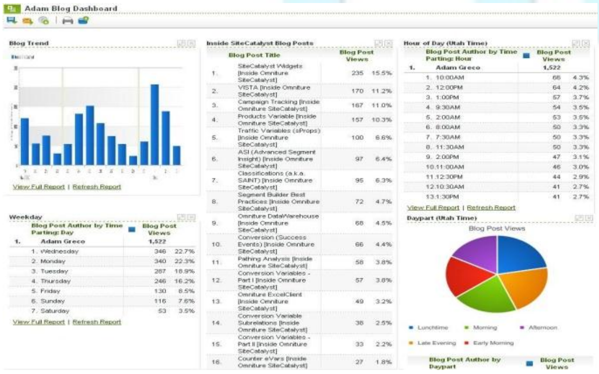
Fig 2 A Sample Web Log Expert Report

A sample web log expert report one can find here: http://www.weblogexpert.com/sample/index.htm

B. Omniture / Google Analytics Data

The other way to track these data with custom field is through Google analytics (free & Licensed Versions), Omniture (Licensed from Adobe Systems) and few more are available in the market. This needs some change in the html /jsp/asp/aspx pages to load the tracking functions. In addition to that additional fields can be tracked such as application specific fields like User Name, Region, City; number of records, lines in an order can also be tracked. This will make a web / service call to the Omniture /Google site to increment the hits every time a user load the corresponding page. A separate non functional requirement will track the tags to be used etc.

Extensive reporting & Dashboards are available in the Omniture / Google side to gather the metrics. This would allow the metrics collecting people to gather the data by choosing various filters. If you have Omniture / Google Analytics data included in your web application you no need to go for any web log analyzer tools.



Adam Blog Dashboard

Weekday

	Blog Post Author by Time Parting: Day	Blog Post Views	
1.	Adam Greco	1,522	
	1. Wednesday	346	22.7%
	2. Monday	340	22.3%
	3. Tuesday	287	18.9%
	4. Thursday	246	16.2%
	5. Friday	130	8.5%
	6. Sunday	116	7.6%
	7. Saturday	53	3.5%

View Full Report | Refresh Report

Inside SiteCatalyst Blog Posts

	Blog Post Title	Blog Post Views	
1.	SiteCatalyst Widgets [Inside Omniture SiteCatalyst]	235	15.5%
2.	VISTA [Inside Omniture SiteCatalyst]	170	11.2%
3.	Campaign Tracking [Inside Omniture SiteCatalyst]	167	11.0%
4.	Products Variable [Inside Omniture SiteCatalyst]	157	10.3%
5.	Traffic Variables (sProps) [Inside Omniture SiteCatalyst]	100	6.6%
6.	ASI (Advanced Segment Insight) [Inside Omniture SiteCatalyst]	97	6.4%
7.	Classifications (a.k.a. SAINT) [Inside Omniture SiteCatalyst]	95	6.3%
8.	Segment Builder Best Practices [Inside Omniture SiteCatalyst]	72	4.7%
9.	Omniture DataWarehouse [Inside Omniture SiteCatalyst]	68	4.5%
10.	Conversion (Success Events) [Inside Omniture SiteCatalyst]	66	4.4%
11.	Pathing Analysis [Inside Omniture SiteCatalyst]	58	3.8%
12.	Conversion Variables - Part I [Inside Omniture SiteCatalyst]	57	3.8%
13.	Omniture ExcelClient [Inside Omniture SiteCatalyst]	49	3.2%
14.	Conversion Variable Subrelations [Inside Omniture SiteCatalyst]	38	2.5%
15.	Conversion Variables - Part II [Inside Omniture SiteCatalyst]	33	2.2%
16.	Counter eVars [Inside Omniture SiteCatalyst]	27	1.8%

Hour of Day (Utah Time)

	Blog Post Author by Time Parting: Hour	Blog Post Views	
1.	Adam Greco	1,522	
	1. 10:00AM	66	4.3%
	2. 12:00PM	64	4.2%
	3. 1:00PM	57	3.7%
	4. 9:30AM	54	3.5%
	5. 2:00AM	53	3.5%
	6. 8:00AM	50	3.3%
	7. 7:30AM	50	3.3%
	8. 11:30AM	50	3.3%
	9. 2:00PM	47	3.1%
	10.11:00AM	46	3.0%
	11.12:30PM	44	2.9%
	12.10:30AM	41	2.7%
	13.1:30PM	41	2.7%

View Full Report | Refresh Report

Daypart (Utah Time)

Blog Post Author by Daypart	Blog Post Views

Fig 2 Sample Omniture Dashboard

C. Database Statistics

This can be retrieved from DB access history. By including number of writes / updates to a particular table over a period of time. This can be done through by enabling and configuring audit logs from application servers.

Apart from these, we can get information from Audit logs of application servers. But there are no log analysis tools available to analyze audit logs since they don't have a common format across different vendors.

VII. ADVANTAGES OF REMODELING LOAD PROFILE

This process needs some time to gather data, and need statistical analysis and comparing against what we done earlier. But by doing this we have many advantages and here it described.

1. Workload Profile updated with the Present Load
2. Response time increase will be monitored continuously
3. Failure rate due to Increased loads will be detected early
4. Need for a hardware addition / upgrade identified at the right time
5. Providing a fast & reliable website to end customers
6. Need for Transaction tuning identified and tested effectively without failures
7. Any assumption in the future load on a web application studied and revised every time.
8. Tuning think time and pacing time in the load scripts

To implement this on a website needs a onetime investment on Web log analysis tools or Implementing Omniture or Google analytics. Later on it needs the person's time to gather and analyze.

In addition to that the project team can aware of most sensitive usage time in a day, and most active day in a month and peak month etc. This will give the idea about designing the application down time and maintenance window. It also gives the average time a user spends in the application and average number of concurrent users on the application etc. These times used to figure out the think time and pacing time in designing load test[9]. By doing these exercise, we will simulate a test which is very similar to actual production environment

VIII. CONCLUSION

We discussed the approach of redefine the work load profile for web applications once or twice in every year. Apart from that we discussed the ways to do that and how it will be helpful in maintain the reliability of the web application.

- This is helpful in designing the load test

- This is helpful in simulating the load test
- This is helpful in early detection of Performance bottlenecks in Production Environment
- This will be helpful in keeping the project team aware of what is going on Production.

By Overall , this exercise aims to educate the business owners & the Project Development teams about the present load pattern and gives enough room to correct the work load profile before failures.

IX. References

[1] Modeling a Realistic Workload for Performance Testing ChristofLutteroth, Gerald Weber

[2] Testing Object-Oriented Systems: Models, Patterns, and Tools - By Robert Binder

[3] Mercury Interactive Corporation. Load Testing to Predict Web Performance. Technical Report WP-1079-0604, Mercury Interactive Corporation, 2004

[4] FilippoRicca and Paolo Tonella. Analysis and Testing of Web Applications.In ICSE '01: Proceedings of the 23rd International Conference on Software Engineering, pages 25–34, Washington, DC, USA, 2001.IEEE Computer Society

[5] Y. Cai, J. Grundy, and J. Hosking. Synthesizing client load models for performance engineering via web crawling. In ASE 2007 – Proceedings of the 22th International Conference on Automated Software Engineering,pages 353–362, 2007.

.

[6] Daniel A. Menasc´e. Load Testing of Web Sites. IEEE InternetComputing, 6(4):70–74, July 2002.

[7] Performance Testing for Web Based Application Architectures(.NET vs. Java EE)Osama Hamed1 and Nedal Kafri2

[8] Omniture available at *www.**omniture**.com/en/products*

[9] D.A. Menasc´e, "Load Testing, Benchmarking, and Application Performance Management for the Web," inProc. 2002 Computer Management Group Conference, pp.,271-281, Reno, Nevada, December 2002. Available at:http://www.cmg.org

[10] Google Analytics available @ www.**google**.co.in/**analytics**/

[11] weblog analyzer tool available to download from *indihiang.codeplex.com/*

[12] SAWMILL log analyzer tool available www.**sawmill**.net/

[13] web log expert tool available to download from www.**weblogexpert**.com/

[14] webalizer tool available to download from www.**webalizer**.org/