International Journal of Engineering Research and Development
***e*-ISSN: 2278-067X, *p*-ISSN: 2278-800X, www.ijerd.com**
Volume 3, Issue 9 (September 2012), PP. 58-61

Software Performance Testing Tools – A Comparative Analysis

Mrs. Charmy Patel[1], Dr. Ravi Gulati[2]
[1]*Shree Ramkrishna Institute of Computer Education and Applied Sciences, M.T.B College Campus, Athwalines, Surat, Gujarat, India.*
[2]*Department of Computer Science,Veer Narmad South Gujarat University,Surat, Gujarat, India.*

Abstract—Performance is one of the most important aspects concerned with the quality of software. It indicates how well a software system or component meets its requirements for timeliness. Nearly everyone runs into performance problems at one time or the other. Today's software development organizations are being asked to do more with less. For performance driven development we propose a framework for performance analysis and performance evaluation. So, in this paper, we have discussed comparative analysis of various software performance testing tools and their limitations and a new approach for software performance testing. With an upbeat approach, we can produce software that can meet performance objectives and can be delivered on time and within budget, avoiding the project crises.

Keywords—Performance analysis, testing tools, performance indicator.

I. INTRODUCTION

Often, the fact is that although the software system has gone through extensive functionality testing, it was never really been tested to assess its expected performance.

Managing system performance facilitates the effective delivery of strategic and operational goals. There is a clear and immediate correlation between using performance management programs or software and improved business and organizational results.

In an ideal world, performance would be engineered into software starting early in the development process. The reality, however, is that budget and schedule constraints often lead developers, to adopt a "make it run, make it run right, make it run fast" strategy. But, the result is that, somewhere near the end of the project, performance problems appear [2].

To make the software engineering process efficient in the field of performance management following aspects has been studied and reviewed. Following approach will clearly identify problem domain and a systematic, quantitative approach to performance tuning that will help in quickly finding problems, identify potential solutions, and prioritize efforts to achieve the improvements with the least effort.

II. ANALYSIS OF SOFTWARE PERFORMANCE TESTING TOOLS

To meet the requirements of performance testing, different software are available in the market but they have their own limitations in different areas.

For example,

- Firefox browser's add-on firebug provides the facility of network monitoring. If web pages are taking long time to load then, Firebug breaks it all down file-by-file. It provides the features like,
  - Watching timeline for scripts, image, multimedia and all other files.
  - Network monitor will examine HTTP headers, cache, functions and file calls, XmlHTTPRequest etc. and will give idea about its load time only.
- SQL Query analyzer is used only for calculating actual query execution time.
- Nunit test tool is used to check performance of entire application which processes only executable files and dll files.
- Web Page Analyzer - Calculates total number of http requests, total size of HTML files, Scripts, CSS, multimedia files and image files with context to connection speed.
- Load impact tool which Calculates only the load time of images, scripts, style sheets and number of requests per second.
- Web page test tool, which was developed by AOL, calculates memory usage of page, css, HTML, js file and image & displays the result in chart format but is not able to calculate load time.
- Slocop which is used to display data like total number of elements, page size, download time only.
- Pingdom which Tests your page load speeds The Full Page Test gives you a detailed visual report on the load time of each element (images, CSS, JavaScripts, RSS, Flash and frames/iframes). Objects included in javascript are ignored.
- FWPTT, which is a web application tester program for load testing web applications, can record normal and Ajax requests.
- Different application monitors are available for desktop application monitoring but they have their own limitations like they monitor limited aspects of application.

A. Some Examples:

Object Size Totals

Object type	Size (bytes)	Download @ 56K (seconds)	Download @ T1 (seconds)
HTML:	38890	8.35	0.81
HTML Images:	138199	30.34	3.53
CSS Images:	217399	49.93	7.75
Total Images:	355598	80.27	11.28
Javascript:	91426	19.22	1.48
CSS:	26969	7.17	1.94
Multimedia:	0	0.00	0.00
Other:	0	0.00	0.00

External Objects

External Object	QTY
Total HTML:	3
Total HTML Images:	14
Total CSS Images:	33
Total Images:	47
Total Scripts:	5
Total CSS imports:	9
Total Frames:	0
Total Iframes:	2

Download Times*

Connection Rate	Download Time
14.4K	410.31 seconds
28.8K	211.55 seconds
33.6K	183.16 seconds
56K	115.02 seconds
ISDN 128K	44.10 seconds
T1 1.44Mbps	15.52 seconds

Global Statistics

Total HTTP Requests:	3
Total Size:	6407 bytes

Fig 1. Webpage Analyzer 0.98

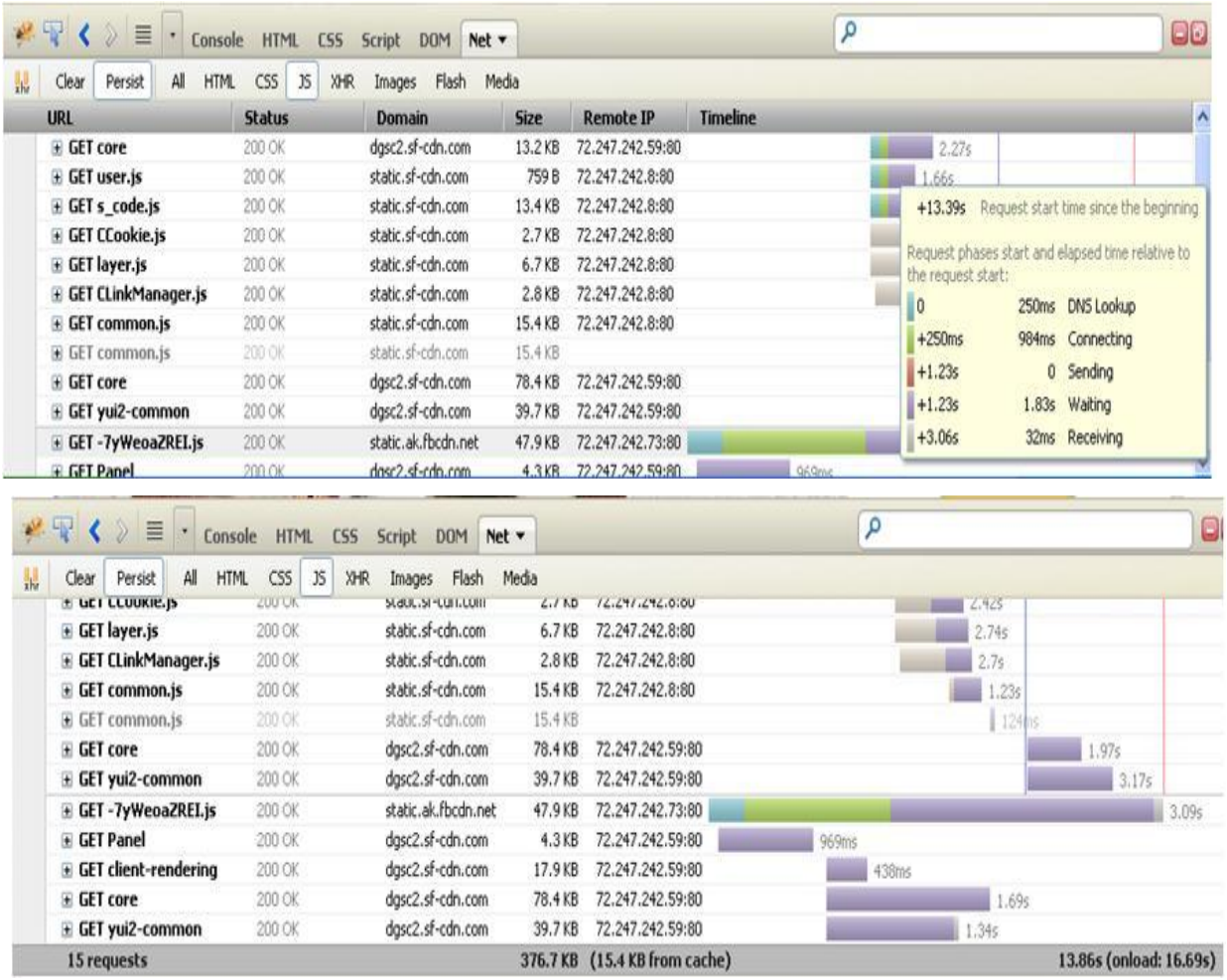

Fig 2.Firebug

To achieve performance driven development, at the time of application development (coding phase), developer requires an indication about the performance of a page/file created by him. Before application deployment quality assessment of whole application is required to get the idea about functioning and performance of project. Hence we propose the following architecture.

III. PROPOSED ARCHITECTURE

Good software performance testing facilitates the effective delivery of strategic and operational goals. Hence performance testing is performed to determine how a system performs in terms of responsiveness and stability under a particular workload. It can also serve to investigate measure, validate or verify other quality attributes of the system, such as scalability, reliability and resource usage. To meet the requirements of performance testing different software tools are available in the market. There is no such framework in the market which covers all the aspects to test system performance, which will process the code and indicate the developer about following parameters at coding phase.

- Environment
- Memory usage and time
- Website load time
- etc.

The above parameters further include different parameters like code execution time, Memory Usage, External file (scripts, images, multimedia) load time, Ideal website load time, Ideal image format and size, Ideal Script size, Style sheet size, Multimedia file size, Number of HTTP Requests, Ideal Compression technique, Browser compatibility, S/W compatibility, H/W compatibility, etc.

Application can be scanned with this new approach which in turn will give detailed report on the status of application in terms of network, server response, memory usage etc. and can also suggest for performance and quality improvement for application/software. So, team leaders can point out issues easily which can cause performance degradation and hence major issues can be solved during the development phase. A performance driven approach will lead industry to quality and will help in stress reduction, since quality is important for a product and stress is harmful for creativity. So this framework will be helpful in both the areas and in future will result into good productivity, quality and financial gain.

IV. CONCLUSION

Performance problems are introduced early in the development process but are typically not found until later (during integration test or when the system is in use) when they are more difficult and more expensive to fix [3].There is no such kind of tool available which will help to test the performance of the software at coding phase i.e. at local network. Available tools and techniques are used very rarely because of their variations. There is no integrated framework available to analyze and evaluate application performance in every aspects related to images, scripts, audio, video, multimedia, code execution, query execution, external resources integration, server response time, memory usage, CPU usage, cache utilization etc.

So, the results generated by this proposed approach will give clear idea to the developer about performance of the application at coding phase itself. These results will also help in identifying the actual area causing performance degradation or can cause in future and will provide possible optimum solution.

REFERENCES

[1]. Software Performance Testing By, Xiang Gan
[2]. Five Steps to Solving Software Performance Problems By Lloyd G. Williams, Ph.D., Connie U. Smith, Ph.D. June, 2002
[3]. Best Practices for Software Performance Engineering, Connie U. Smith, Performance Engineering Services
[4]. The Business Case for Software Performance Engineering Lloyd G. Williams, Ph.D., Connie U. Smith, Ph.D.
[5]. Development Practices: Exploring the Trade-offs Between Productivity and Quality, forthcoming, IEEE Software
[6]. Software Maintenance as Part of the Software Life Cycle Comp180: Software Engineering Prof. Stafford
[7]. Banker, R.D., S.M. Datar, and C.F. Kemerer, A Model to Evaluate Variables Impacting Productivity on Software Maintenance Projects. Management Science, 1991. vol. 37
[8]. Gilb, T., Principles of Software Engineering Management, Addison-Wesley, New York
[9]. J. D. Musa, A. Iannino, and K. Okumoto, Software Reliability Measurement, Predication, Aplication, McGraw Hill, 1987.
[10]. Kemerer, C., MacCormack, A., Cusumano, M. and W. Crandall, "Practice and Performance in Software Development: A Global Study", working paper, 2003.
[11]. Lloyd G. Williams, Ph.D., Connie U. Smith, Ph.D. Five Steps to Solving Software Performance Problems June, 2002
[12]. MacCormack, A. "Product-Development Processes that Work: How Internet Companies Build Software" MIT Sloan Management Review, vol. 42 no. 2, 2001, pp. 75-84.
[13]. International Journal of Recent Trends in Engineering, Vol 2, No. 4, November 2009
[14]. Performance Analysis Framework for Large Software Intensive Systems with a Message Passing Paradigm Christian Del Rosso
[15]. M. J. A. Smith Performance-Driven Development, Microsoft Research Event 2007