

# Evaluating Performance of Web Services in Cloud Computing Environment with High Availability

Dilbag Singh<sup>1</sup>, Sukhamrit Kaur<sup>2</sup> and Dr. Kuljit Kaur<sup>3</sup>

<sup>1</sup> guru nanak dev university

*Received: 6 April 2012 Accepted: 30 April 2012 Published: 15 May 2012*

## Abstract

This paper presents an methodology for attaining high availability to the demands of the web clients. In order to improve in response time of web services during peak hours dynamic allocation of host nodes will be used in this research work. As web users are very demanding: they expect web services to be quickly accessible from the world 24\*7. Fast response time leads to high availability of web services, while slow response time degrades the performance of web services. With the increasing trend of internet, it becomes a part of life. People use internet to help in their studies, business, shopping and many more things. To achieve this objective LAMP platform is used which are Linux, Apache, My SQL, and PHP. LAMP is used to increase the quality of product by using open source software. The proposed strategy will work as middle layer and provide highly availability to the web clients.

**Index terms**— Failover, dynamic allocation, host service provider, High availability, LAMP, Web Services.

## 1 Introduction

The Internet and World Wide Web (WWW) have captured the world's imagination. Internet is represented in network as a cloud. Cloud computing is where application and files are hosted on a cloud consisting of thousand of computers and servers, all linked together and accessible via internet. Any web service or application offered via cloud computing is called a cloud service. With this simple but powerful interface, a user can download a file after accessing any web service from another computer with only a click of the mouse. Moreover, advances in technology continue to extend the functionality of the Internet. As Web services becomes increasingly popular, network congestion and server overloading have becoming significant problems. So efforts are being made to address these problems and improve web performance.

In this paper for providing high availability to the requests of web clients a new environment is developed. In order to improve in response time of web services during peak hour's dynamic distribution of host nodes will be used in this research work. As expectations of web users are at crest: they expect web services to be quickly accessible from the world 24\*7 Fast response time leads to high availability of web services, while slow response time degrades the performance of web services. With the mounting tendency of internet, it becomes a part of life. People use internet to help in their studies, industry, Shopping and many more things.

To attain this objective LAMP Technology is used in which are Linux is an operating system, Apache web server, My SQL database, and PHP scripting language. LAMP is used to increase the quality of product by using open source software.

## 2 II.

## 3 Research motivation

In existing Dynasoar do not provide the solution if request from the single client occur more than a time for the same web service. Moreover, it will not provide any solution to HSP failure. So after see the causes and benefits

of Dynasoar environment, try to implement it practically as Dynasoar is a theoretical concept, here trying to implement it.

### 4 III.

### 5 Scope of this research

The scope of the research is defined by the following: 1. This research work deals with load distribution and high availability for web services. 2. This research does not deal with management and security issue of web services. 3. Since it is not feasible to run the proposed strategy on large web hosting, small web sites are developed which will simulate the proposed algorithm using LAMP. 4. Different type of tests will be implemented using LAMP to test various aspects of the web services. 5. Visualization of the experimental results and drawing appropriate performance analysis. 6. Appropriate conclusion will be made based upon performance analysis. 7. For future work suitable future directions will be drawn considering limitations of existing work.

IV.

### 6 Problem definition

In this research paper a new approach is considered in which there is an active-active server. If a client accesses the same web service several times then it will consider as one and if one server fails, then client request redirected to another active server. Using active active server eliminated the problem of server enhancement and failover time. To achieve the objective some constraints have been setup, according to them, if access to the server goes down, another server has placed to accommodate that request. Preventing an interruption is the model of Active-Active high availability, which introduces multiple active, replicated, redundant components.

V.

## 7 Literature review

Web services (WS) [1] [2] [3] are self-contained software modules available in a network, such as the internet, which completes tasks, solves problems, or conducts transactions on behalf of a user or application. It is trajectory of communicating between two electronic devices over the web. WS interact with the sources of the information, changing the state of systems and causing real world processes to occur. As a WS network grows, its existence and performance becomes crucial to the business's core activities. So the management of WS is important for providing seamless access of the service to the user.

Web server (WSer) [setting up a web server by simon Collins [5][6] [7] delivers web services to the clients. Web server is connected to the web and can be accessed to the users. It is possible that user can setup its own web server. Web server can be connected to internet or it can be a private Intranet. Both require similar software, only Intranet works in a private network and internet connected to public internet. As the traffic on web server increases, congestion will increased which may results in low response time. So to reduce these problems concept of redirection is used.

Redirection (RD) [8][9] is the process of selecting the best server that can serve user request. Web server can redirect the browser to go elsewhere to proceed the user request. Redirection happens at client side. A client is redirected only after its request has reached the home server. When user request arrives, if there is congestion, the server can redirect the client to the other web page where same request to be processed.

Redirection of a client towards a given replica of a Web service is performed after the client's request has reached the Web server storing the requested service. To improve the overall systems throughout, redirection takes place. This one of the most common use is to route traffic while migrating a web page from one server to another.

Proxy Server (PS) [10][11][12]function is to forward traffic between clients and server. Here Mysar Squid Proxy server is used. Squid is an intermediary between clients and server. As Squid is a open source software. Mysar squid is a monitoring tool that constantly monitors the request of web client, and creates the database where clearly showing for which web service client made a request. PS could help provide adequate access and response time to large numbers of users requesting previously accessed page.

High availability (HA) [13], [14] also known as failover. The key to HA be redundancy is the most common approach to increase availability. If the primary fails, one of the back-ups is promoted into that role. HA ensures automated recovery in case of failure with two different approach 1+1 and 1:1. Over the time, the file management systems and registered data became complex, and database management systems were increasingly used to store metadata.

It is often said that this generation of web services got it start from LAMP. LAMP is a stack of simple web technologies, powerful web technologies that power a lot of popular. LAMP is a popular open source solution used to run servers in which PHP is configured to run on Apache web server, using the MySQL database on Linux operating system. It is popular because of its open source nature, low cost, and its packages are easy to install and convenient to use [15] [16] [17] [18].

Availability [19], [20], [21] is a reoccurring and a growing concern in software intensive systems. Cloud systems services can be turned off-line due to conservation, power outages or possible denial of service invasions.

Fundamentally, its role is to determine the time that the system is up and running correctly; the length of time between failures and the length of time needed to resume operation after a failure. Availability needs to be analysed through the use of presence information, forecasting usage patterns and dynamic resource scaling.

Dynamic web service deployment functionality has been explored and developed in many different contexts, including J2EE [22], [23] and Web Services [24]. Rauch et al. [25] implemented partition cloning and partition repositories as well as a set of OS-independent tools for software maintenance using entire partitions, thus providing a clean abstraction of operating system configuration states. However, this approach is not suitable for service-oriented architectures. Moreover, the deployment of an entire OS image is expensive, and the deployment itself will seriously impact system availability. Chase et al. explore related ideas in their Cluster on Demand project [26].

Keahey et al. [27], [28] use virtual machine technology (e.g., Xen, VMware) to build virtual working environments and to provide for the dynamic management of the Grid job life cycle. Their use of virtual machines rather than JVMs to host user computations leads to somewhat different solutions from our service-oriented approach.

ROST [29], deployed in the CROWN Grid, focuses on dynamic and remote deployment for WSRF core with secure access. The developers evaluated remote deployment in the load balancing of local( D D D D ) B 2012

Year clusters. However, they did not discuss in detail the capability and availability of deployment. Weissman et al. present an architecture and implementation for a dynamic Grid service architecture based on Tomcat that extends GT3 to support dynamic service hosting (hosting and rehosting a service within the Grid in response to service demand and resource fluctuation) [30], [31].

Their implementation allows new services to be added or replaced without taking down a site for reconfiguration and allows a VO to respond effectively to dynamic resource availability and demand. But the implementation is based completely on Tomcat's container-level deployment capability, which suffers from poor performance. These and a few other projects [32], [33] are the main dynamic deployment efforts for Grid applications. Some of them clearly are not intended for a WSRF-enabled service-oriented architecture. Moreover, although some have implemented serviceoriented dynamic deployment, they do not address in detail the cost, namely, the capability brought from dynamic deployment itself and the availability in dynamic Grid environments.

## 8 VI.

## 9 Experimental set-up

In order to implement the fail-over strategy a suitable experimental set-up has been made as shown in Fig. ?? Fig. ?? take following steps to execute the jobs of the clients: The client request are monitored in such a way that each request can transparently monitor. For this process, the proxy server (Mysar) is implemented. The Network setup is as shown in Fig. ??.

Fig. ?? : Experimental set-up In Fig. ?? following steps are performed:

Step1: Initially clients submit their jobs to access some web services; it goes through some processes.

Step2: Requests goes to WSP, and then it will pass the given requests to the HSP.

Step3: HSP then fulfill the requests and give response to client according to desired requirement.

Step4: Multiple servers (HSP) are there in HSP end to provide services to their clients.

Step5: Policies are implemented in such a manner that the client's requests automatically redirected to other web servers. If number of requests range is reached to its peak, then it will be assumed that given server is busy or facing some sort of problem for responding client's requests.

Step6: Client's request is fulfilled with high availability with low response time.

PS: Server used for monitoring the service that user request. Clients in the network connected to the PS, after client request for web service, database is created into PS after monitoring. Then after passing through PS it goes to WSP.

## 10 VII.

## 11 Simulation results

a) Ideal server Fig. 2 is showing that when the web server was kept Ideal i.e. no request received from the client. This is the rare condition when server was kept ideal or can be possible during off-peak hours. All the request will be process on time-sharing basis and beyond this point the server load will increase which intern reduce the performance of the server. So to overcome this redirection of Web server is mandatory. d) Beyond Threshold Fig. ?? demonstrating that when 6th request received at server at same instant. The 6th request will be monitor by Wire shark as mentioned below. e) Beyond threshold In Fig. 7 client request for amrit.sukralamata.com so counter increase by 1 by received the request from Sukh.fossfoundation.com as given in Fig. ??.

After that if client request for amrit sub-domain page will redirect to such sub-domain. And both count increase by

## 12 Performance analysis

In order to do performance analysis, two comparisons table has been made in this research work. This section first give the performance comparison of developed simulator with existing methods (in which no dynamic deployment of host nodes is implemented) and later on comparison of different approaches is made using different performance metrics. II that proposed method gives better results than existing methods. As in existing method it not possible to achieve dynamic allocation of host nodes, and without log files may cause the problem of random allocation of nodes to the requests, which may increase response time. Fig. 8 is showing the graph of average response time using different intervals, which are shown in Table II. Fig. 8 shows the difference between existing methods graphically and it is clearly shown that the proposed method gives better results than existing methods. III. Fig. ?? shows the difference between existing methods graphically and it is clearly shown that the showing the graph of average response time on server 1, which is shown in Table IV. Fig. ??0 has demonstrate that as number of hits increased by some constant, average response time will increased rapidly (quite more than the increase in number of hits). Therefore proposed technique will prevent it by doing load balancing among available host nodes thus make average response time optimal as shown in Table V V. Fig. 12 has demonstrate that as number of requests increased by some constant, utilization is also increased by some multiple constant but after the threshold (the capacity of server) it will increases rapidly. Year Fig. 13 is showing the server utilization, which is shown in Table VI. Fig. 13 has demonstrate that as number of requests.

Table VII : Server Utilization (In %) Among Two Servers Fig. 13 : Server Utilization (in %) among two servers increased by some constant, utilization is also increased by some multiple constant but after the threshold (the capacity of server) proposed strategy will do balance the load among other active servers.

## 13 IX.

## 14 Conclusion and future directions

This paper proposes a smart strategy for web services using dynamic allocation techniques. Using LAMP, a new environment has been developed that implement the proposed method. Performance comparison of existing methods has been made with the proposed method. It has been concluded with the help of performance metric's comparison that the proposed failover strategy gives good results than existing methods.

In this paper homogeneous host nodes has been considered for simulation environment, in future work heterogeneous nodes will be used for better results.

## 15 X.

## 16 Contributions

This section will describe contributions of this research work to science and practice.

A literature review has been conducted on the approached of high availability in web services. The results of this literature research show that web services are in its infancy, because the structured search revealed minimal peer-reviewed information on these topics. Regardless, this research has refined three concepts from the literature and transformed them into dimensions that are usable in the context of web services. The most contributing dimension is based on concept from the research area of fail-over strategies.

Another contribution of this research is related to the LAMP. This research uses LAMP in order to provide high availability to the demands of the clients. With the identification of congestion in web services, this research shows which problems can occur when congestion occur. This type of research has not been conducted before and it is important for entities that wish to know what the high availability limitations of web services.

## 17 -45

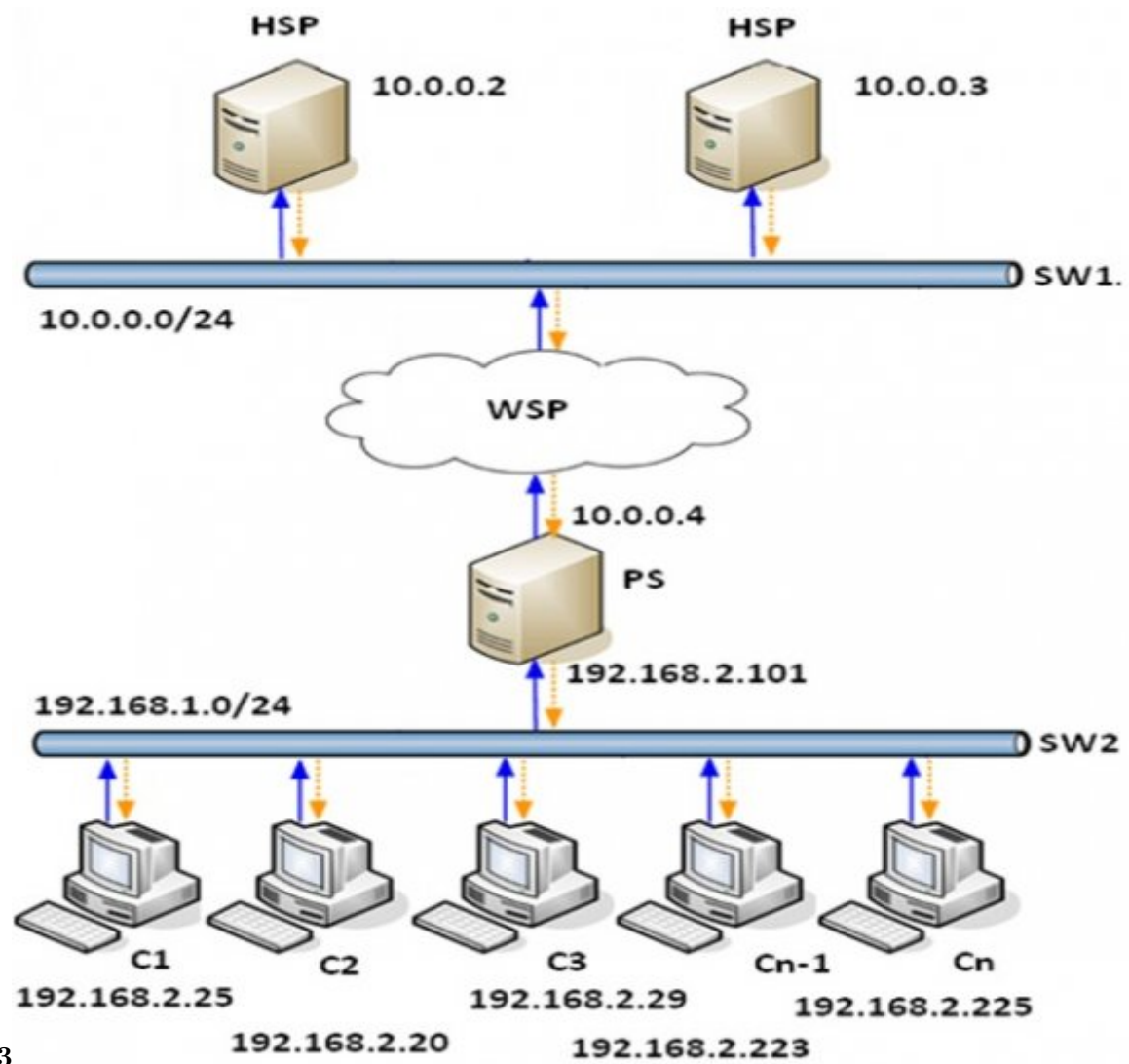
Year <sup>1 2</sup>

<sup>1</sup>© 2012 Global Journals Inc. (US)

<sup>2</sup>© 2012 Global Journals Inc. (US) Global Journal of Computer Science and Technology



Figure 1: Fig. 2 :



3

Figure 2: Fig. 3 :

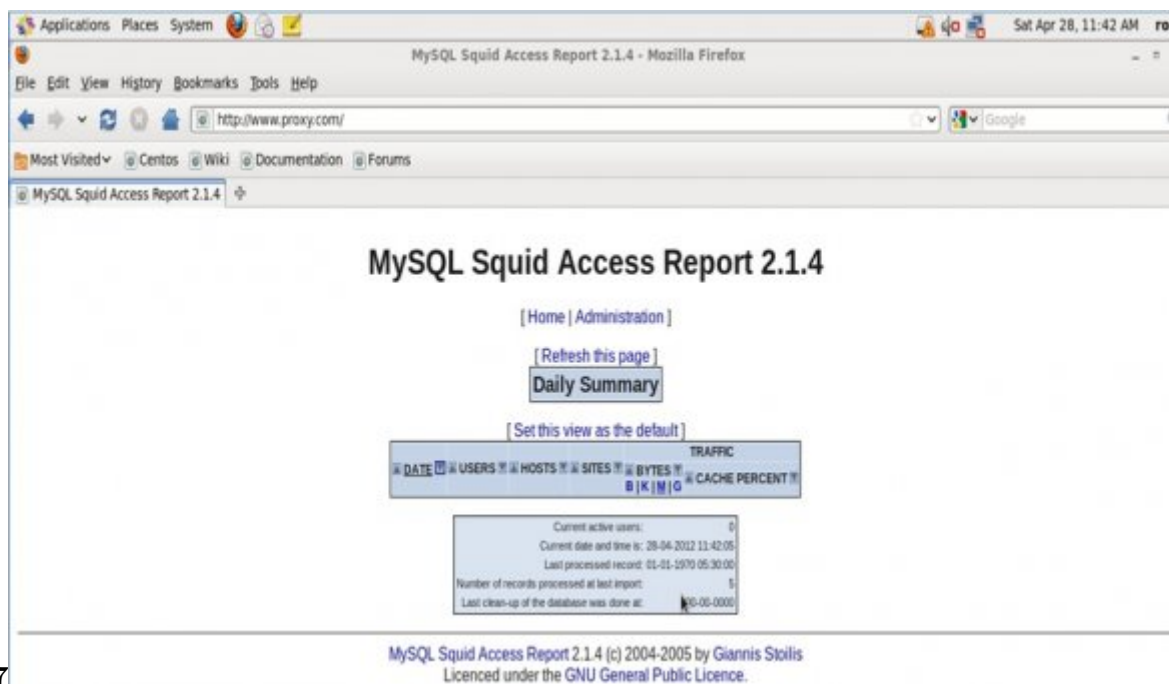


Figure 3: Fig. 7 :

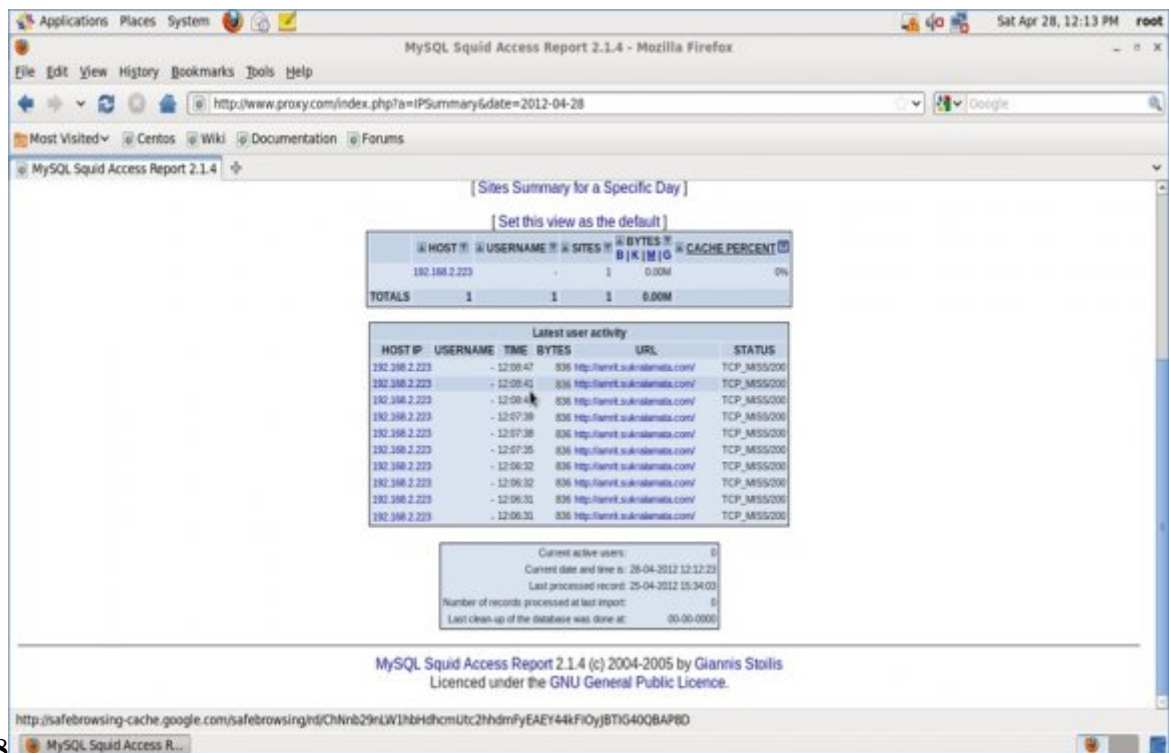
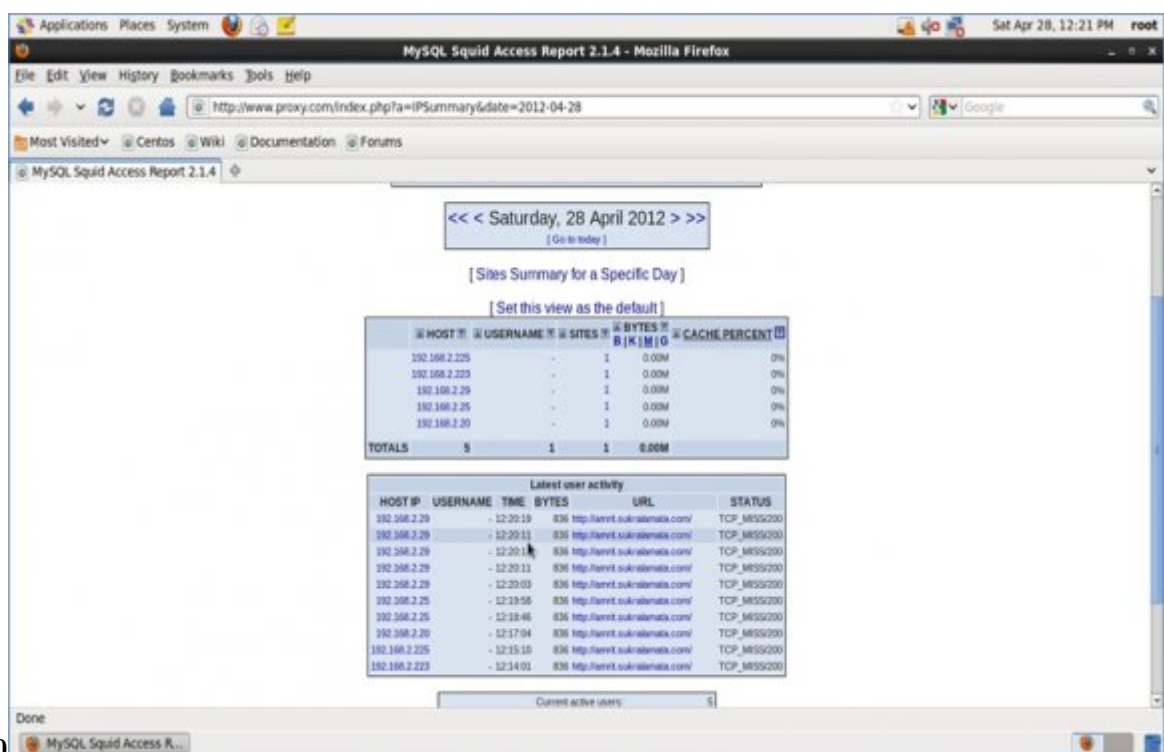


Figure 4: Fig. 8 :



1010

Figure 5: Fig. 10 isFig. 10 :



11

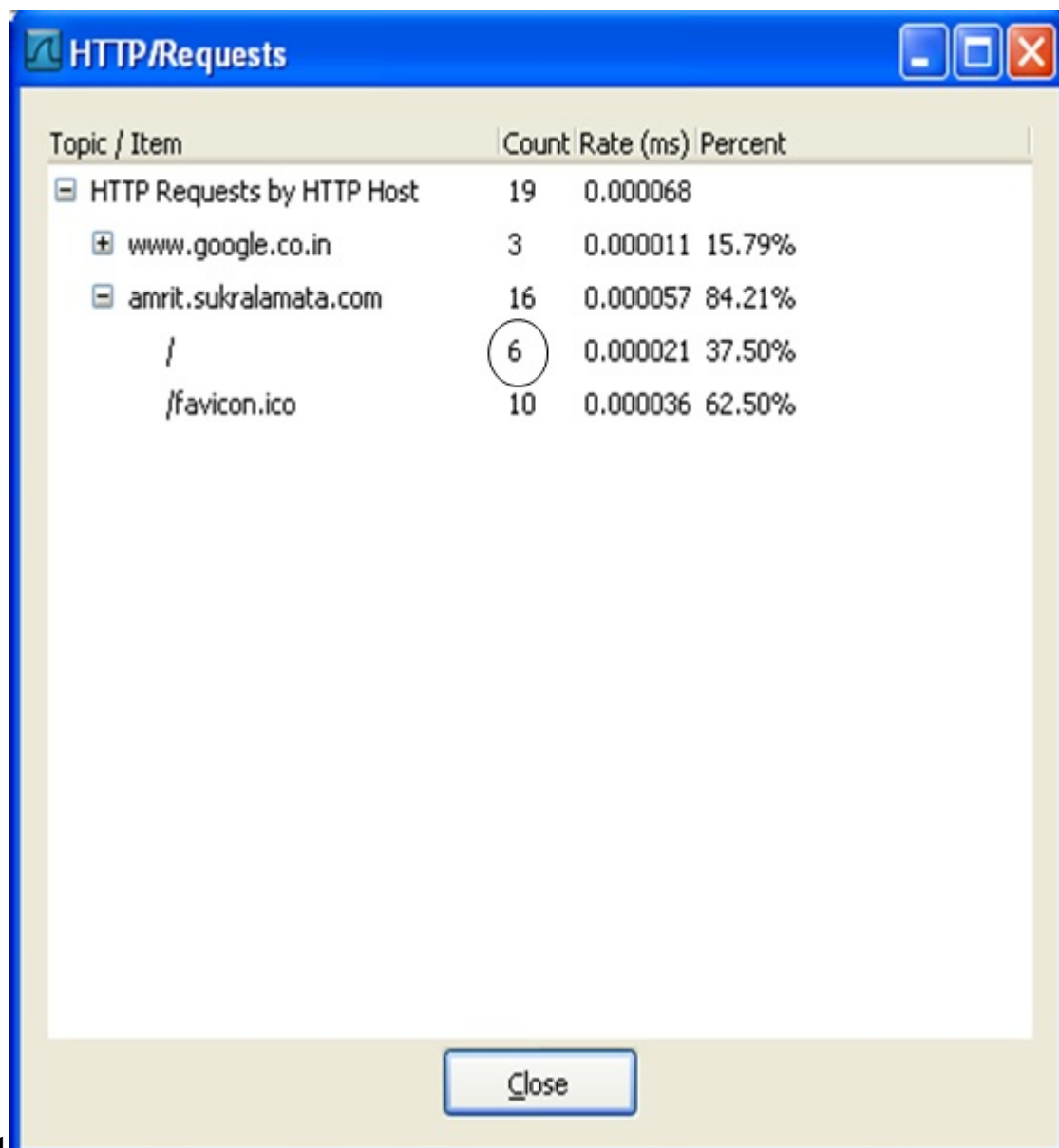
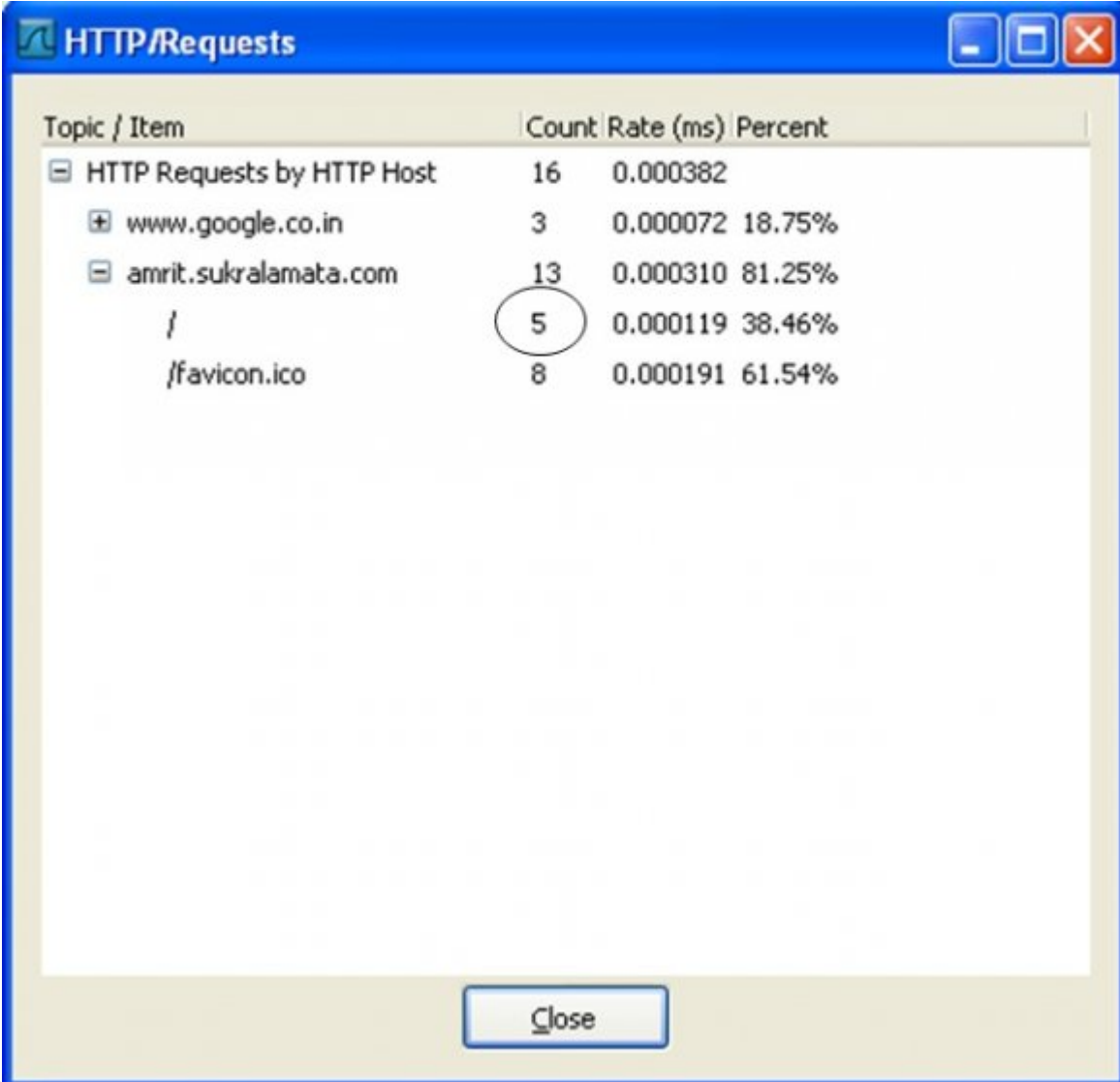


Figure 6: Fig. 11 :



11

Topic / Item	Count	Rate (ms)	Percent
[-] HTTP Requests by HTTP Host	16	0.000382	
[+] www.google.co.in	3	0.000072	18.75%
[-] amrit.sukralamata.com	13	0.000310	81.25%
/	5	0.000119	38.46%
/favicon.ico	8	0.000191	61.54%

Close

Figure 7: Fig. 11 :

12

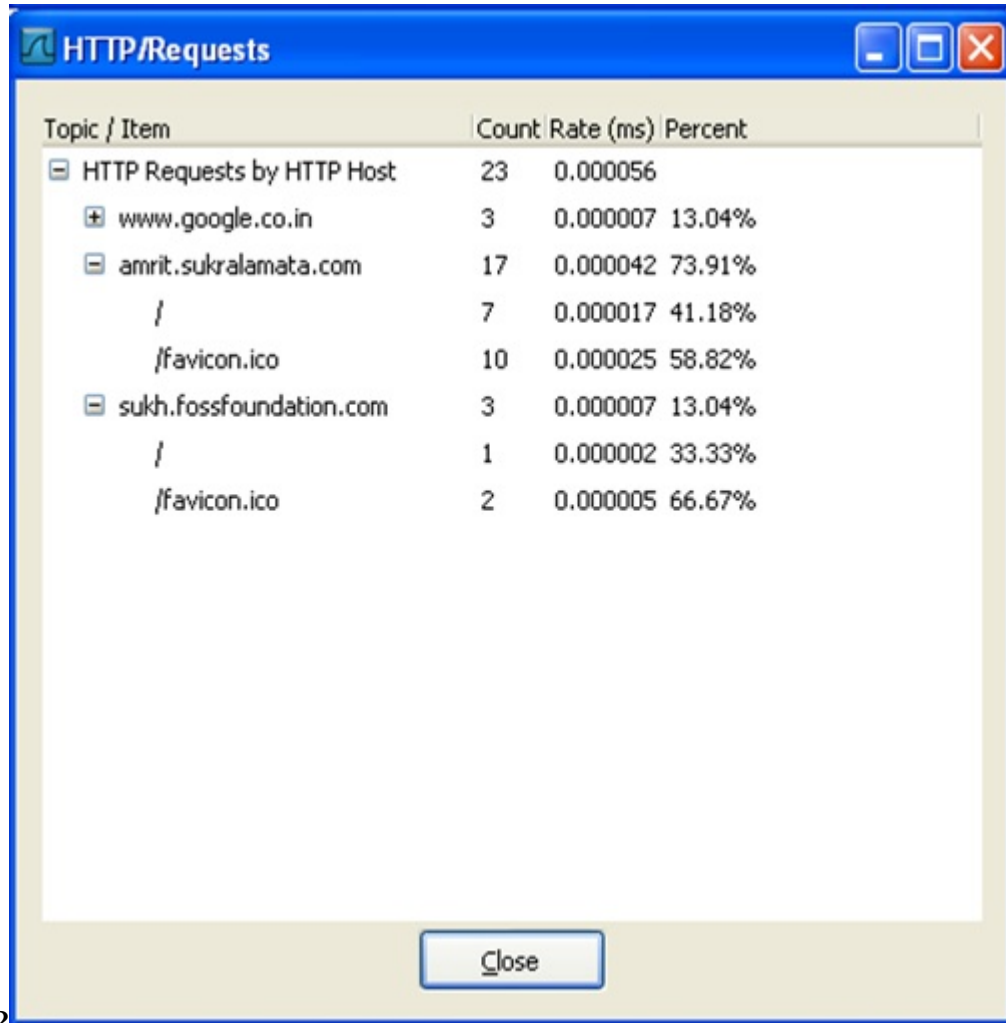


Figure 8: Fig. 12 :

III

Figure 9: Table III :

II

proposed method gives better results than existing methods.

d) Average response time on server 1  
Year

Figure 10: Table II :

V

Figure 11: Table V :

Figure 12: Table

## VI

Figure 13: Table VI :

2012	Number of hits Server 1			
	1	4		
	2	68		
	3	128		
6	4	130		
Volume XII Issue XI ... 9		... 1645	Number of re-	Utilization
Version I			quests	(%)
D D D D ) B			1 2	6 12
(				
			3	18
			4	22
			...	...
			8	92
			9	95
Number of hits	Server 1	Server 2	Server 3	
1	4	-	-	
2	68	-	-	
...	...	...	...	
6	144	-	-	
7	-	16	-	
8	-	42	-	
...	...	...	...	
13	-	-	42	
14	-	-	78	
...	...	...	...	
18	-	-	140	

Figure 14:

- 
- 198 [Weissman et al. ()] 'A Framework for Dynamic Service Adaptation in the Grid: Next Generation Software  
199 Program Progress Report'. J Weissman , S Kim , D England . *19th IEEE International Parallel and Distributed  
200 Processing Symposium (IPDPS'05)*, 2005.
- 201 [Parastatidis et al. ()] *A Grid Application Framework based on Web Services Specifications and Practices*, S  
202 Parastatidis , J Webber , P Watson , T Rischbeck . <http://www.neresc.ac.uk/ws-gaf> 2003.
- 203 [Watson and Fowler] *An Architecture for the Dynamic Deployment of Web Services on a Grid or the Internet*,  
204 P Watson , C Fowler . CSTR-890. University of Newcastle upon Tyne (Technical Report Series)
- 205 [Chen et al. (1997)] *Benchmarking the Next Generation of Internet Servers*, J Chen , M Day , " A Wharton .  
206 <http://domino.lotus.com> March 1997.
- 207 [Etal (2002)] *Bottleneck Characterization of Dynamic Web Site Benchmarks*, Amza Etal . TR02-398. January  
208 2002. Rice University (Technical Report)
- 209 [David et al. (2004)] *Computer Organization and Design: The Hardware/Software Interface*, A David , John L  
210 Patterson , Hennessy . August 2004. Morgan Kaufmann. (3 rd edition)
- 211 [Sridhar et al. (2004)] *Container-based component deployment: A Case Study*, N Sridhar , J O Hallstrom , P A  
212 Sivilotti . OSU-CISRC-2/04-TR08. February 2004. Columbus, OH. Computer Science and Engineering, The  
213 Ohio State University (Technical Report)
- 214 [Benatallah et al. (2002)] 'Declarative Composition and Peer-to-Peer Provisioning of Dynamic Web Services'. B  
215 Benatallah , M Dumas , Q Z Sheng , A H H Ngu . *18th Int. Conference on Data Engineering (ICDE)*, (San  
216 Jose, CA) February 2002. IEEE Computer Society. p. 297308.
- 217 [Chatterjee and Webber ()] *Developing Enterprise Web Services: An Architects Guide: Penguin Books*, S  
218 Chatterjee , J Webber . 2003.
- 219 [Baggio (2004)] *Distributed redirection for the Globule platform*, A Baggio . IR-CS-010. [http://www.cs.vu.  
220 nl/globe/techreps.html](http://www.cs.vu.nl/globe/techreps.html) Oct. 2004. Vrije Universiteit (Technical Report)
- 221 [Szymaniak (2002)] *DNS-based client redirector for the Apache HTTP server*, M Szymaniak . June 2002. Warsaw  
222 University and Vrije Universiteit (Master's thesis)
- 223 [Reverbel et al. (2004)] 'Dynamic Deployment of IIOP Enabled Components in the JBoss Server'. F Reverbel ,  
224 B Burke , M Fleury . *Component Deployment: Second International Working Conference*, (Edinburgh, UK)  
225 2004. May 20-21, 2004. p. .
- 226 [Chase et al. ()] 'Dynamic Virtual Clusters in a Grid Site Manager'. J Chase , L Grit , D Irwin , J Moore , S  
227 Sprenkle . *12th International Symposium on High Performance Distributed Computing*, 2003.
- 228 [Sun et al.] 'Early Experience of Remote and Hot Service Deployment with Trustworthiness in CROWN Grid'.  
229 H Sun , Y Zhu , C Hu . *APPT 2005*, p. .
- 230 [Sheldon (2001)] *Encyclopedia of Networking & Telecommunications*, Tom Sheldon . June 2001. New York, NY:  
231 McGraw-Hill. (3 rd edition)
- 232 [Budrean and Li (2003)] 'High availability solutions for transactional database systems'. S Budrean , Yanhong  
233 Li , DesaiB . *Database Engineering and Applications Symposium*, 2003. July 2003. 347355 p. . (Proceedings.  
234 Seventh International)
- 235 [Sloan ; Rocks et al. (2004)] *High Performance Linux Clusters With Oscar*, J D Sloan ; Rocks , Open Mosix , O'a  
236 Mpi , Reilly . Available:[gec.di.uminho.pt/discip/minf/cpd0910/PAC/livro-hpl-cluster.  
237 pdf](http://gec.di.uminho.pt/discip/minf/cpd0910/PAC/livro-hpl-cluster.pdf) Nov.2004. p. .
- 238 [Ukuug and Conference ()] *High-Availability and Reliability. The Evolution of the Linux-HA Project*, Lisa/Winter  
239 Ukuug , Conference . February 25-26 2004. Bournemouth.
- 240 [ J ()] 'Network Behavior of a Busy Web Server and its Clients'. J . *DEC WRL RR* 2006.
- 241 [Rauch et al. ()] 'Partition Repositories for Partition Cloning OS Independent Software Maintenance in Large  
242 Clusters of PCs'. F Rauch , C Kurmann , T M Stricker . *IEEE International Conference on Cluster Computing*,  
243 2000. p. .
- 244 [Etal (2002)] *Performance and scalability of EJB applications*, Cecchet Etal . 2002. November 2002. Seattle, WA,  
245 USA. 17 p. .
- 246 [Etal ()] 'Performance Comparison of Middleware Architectures for Generating Dynamic Web Content'. Cecchet  
247 Etal . *4th ACM /IFIP/USENIX International Middleware Conference*, (Rio de Janeiro, Brazil) June 16-20,  
248 2003.
- 249 [Wooster and Abrams (1997)] 'Proxy Caching that Estimates Page Load Delays'. R P Wooster , M Abrams .  
250 *Proceedings of the 6th International WWW Conference*, (the 6th International WWW Conference) April  
251 1997.
- 252 [Keidl et al. ()] *Reliable Web Service Execution and Deployment in Dynamic Environments*, M Keidl , S Seltzsam  
253 , A Kemper . 2003. Berlin. (presented at Technologies for E-Services)

- 254 [Crovela and Bestavros ()] 'Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes'. M  
255 Crovela , A Bestavros . *IEEE/ACM Transactions on Networking* 1997.
- 256 [Etal (2002)] 'Specification and Implementation of Dynamic Web Site Benchmarks'. Amza Etal . *IEEE 5th Annual*  
257 *WWC-5*, (Austin, TX, USA) November 2002.
- 258 [Weissman et al. ()] *Supporting the Dynamic Grid Service Lifecycle*, J Weissman , S Kim , D England . CCGrid04,  
259 2004.
- 260 [Smith et al. ()] 'Towards a Service-Oriented Ad Hoc Grid'. M Smith , T Frieze , B Freisleben . *3rd International*  
261 *Symposium on Parallel and Distributed Computing/ Third International Workshop on Algorithms, Models*  
262 *and Tools for Parallel Computing on Heterogeneous Networks (ISPD/HeteroPar'04)*, 2004. p. .
- 263 [Zhang et al. (2010)] 'VirtCFT: A Transparent VMLevel Fault-Tolerant System for Virtual Clusters'. M Zhang ,  
264 H Jin , X Shi , S Wu . *Proceedings of Parallel, Distributed Systems (ICPADS)*, (Parallel, Distributed Systems  
265 (ICPADS)) Dec. 2010.
- 266 [Keahey et al. (2005)] 'Virtual Workspaces in the Grid'. K Keahey , I Foster , T Freeman , X Zhang , D Galron  
267 . *Europar* 2005. September, 2005.
- 268 [Keahey et al. ()] *Virtual Workspaces: Achieving Quality of Service and Quality of Life in the Grid*, K Keahey ,  
269 I Foster , T Freeman , X Zhang . 2006. (Scientific Programming)
- 270 [Badrinath et al. (2007)] 'Virtualization aware job schedulers for checkpointrestart'. R Badrinath , R Krishnakumar ,  
271 R Rajan . *13th International Conference on Parallel, Distributed Systems (ICPADS'07)*, (Hsinchu,  
272 Taiwan) Dec. 5-7 2007. 2 p. .
- 273 [Trent and Sate] *WebSTONE: The First Generation in HTTP Server Benchmarking*, G Trent , M Sate .  
274 <http://www.sgi.com/Products/WebFORCE/WebStone/paper.html> (White paper available at)