International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878, Impact Factor (2014): 3.05

Simulation of Environment of Geographically Distributed Server Nodes for Testing of Load Balancing Algorithms

Shivaprasad Balaji[1], Sneha M[2]

[1]Dept. of Computer Science, R. V. College of Engineering, Bangalore, India
[2]Assistant Professor, Dept. of Computer Science, R. V. College of Engineering, Bangalore, India

Abstract: ***In the present era, web applications are being deployed on more than one server node for ensuring high availability. The server nodes are geographically distributed and are usually connected to a load balancer. Setting up of this environment is expensive since it involves creating server nodes at actual geographic locations with its own computing resources. In this paper, we explore the method to simulate this environment so as to reduce the cost for testing load balancing algorithms.***

Keywords: Web traffic modeling, traffic distribution, response time delaying

1. Introduction

The web applications have changed a lot from their birth in 1991 to what they are presently. From Web 1.0 to current Web 5.0, the traffic experienced by the web applications have scaled up massively. This is firstly due to the type of content the web applications have started to serve. While it was simple static HTML pages in Web 1.0, the content includes Video on demand, multimedia applications, High quality images, user specific dynamic content in Web 5.0.

Secondly, the no. of users using the internet has also scaled up significantly in the last decade. From about a few million users in 1995, the current internet users is estimated to be at about 3 billion and is expected to grow with more pace. This has brought about a paradigm shift in the way web applications are written, deployed and managed.

In the present era, web applications are usually being deployed over multiple server nodes that are geographically distributed. This is to accommodate the users who are also distributed geographically. These server nodes are connected to a load balancer. The load balancing algorithm judiciously uses the resources of the server that are distributed geographically.

The setting up of the environment for testing of the load balancing algorithms is expensive because of the following reasons. Firstly this requires systems which are actually geographically distributed. This is usually achieved by purchasing virtual machines and ensuring that they do not belong to the same geographical area. Also since the application is still under testing, it requires huge resources to produce the traffic that is experienced by the server nodes in production environment. This again adds up to the cost of the network requirements. An environment of about 10 Amazon EC2 nodes with 8 GB RAM and Linux OS is about 900$ per month and 11000$/year.

This paper aims to setup an environment for testing of the load balancing algorithms by simulation of web traffic and a method of delaying response time of the web applications.

2. Survey of Existing Approaches to Model Web Application Traffic

A lot of research has been carried out to model the traffic experienced by web application. Some of the important methods are listed below.

2.1 Gaussian Distribution Model [1]

This method varies the web traffic by varying the length of the HTTP packets. The lengths of the HTTP packets are varied by using Gaussian distribution. The method is also extended for FTP (File Transfer Protocol) traffic.

2.2 Behavioural Model [2]

Web-requests area unit known by analyzing not simply the communications protocol header in the trace however conjointly the HTTP headers. The impact of internet caching is incorporated into the model. The model is evaluated by comparing freelance statistics from the model and from the trace. The explanations for variations between the model and therefore the traces are given.

2.3 Light Weight Traffic Source [3]

The source for this model is an integrated packet session model that captures along with user characteristics, the user behaviour. It also reduces the number of events required for generation of TCP source.

2.4 Poission Distribution [4]

The web traffic is modeled using Poisson distribution here. The traffic bursts in the internet can be better explained by using the Poisson model. Hence we will be using the Poisson model in our proposed method.

3. Proposed Method

3.1 Simulation of Web Application Traffic

The traffic experienced by the web application can be estimated using Poisson distribution. This is because we can model the traffic at different intervals of the day effectively using Poisson distribution. The Throughput of the web

application at different times of the day can be seen in Figure 1.

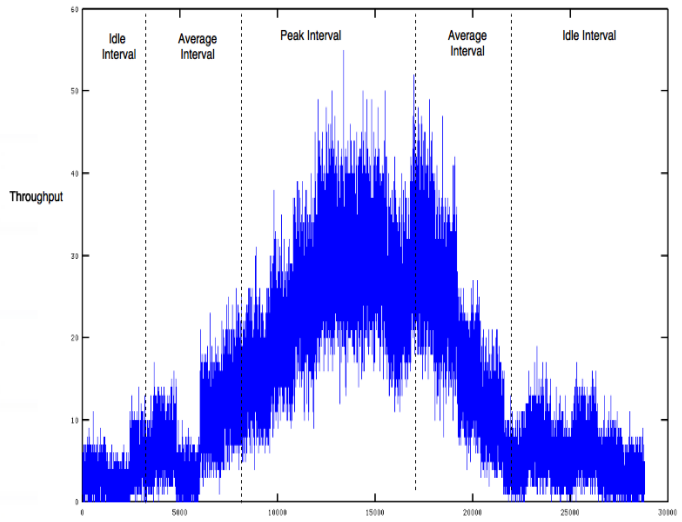

Figure 1: Throughput variation of the web application at different intervals of the day

The time duration of the day can be broken into intervals as shown in Figure 1. The intervals are:

1. Busy/ Peak Interval: Most of the user base of the web application is active at this time. Hence at this interval, the traffic experienced is maximum.
2. Average Interval: Moderate no. of users are active in this interval and the traffic experienced is not too high nor too low.
3. Idle Interval: The traffic experienced by the website is very low and sometimes zero or idle.

The load balancing algorithms exploit the above traffic distribution characteristics to judiciously use system resources. This is because of the time difference for the geographically distributed server nodes. Figure 2 shows the distribution of traffic of for two server nodes that are placed geographically apart according to Poisson distribution.

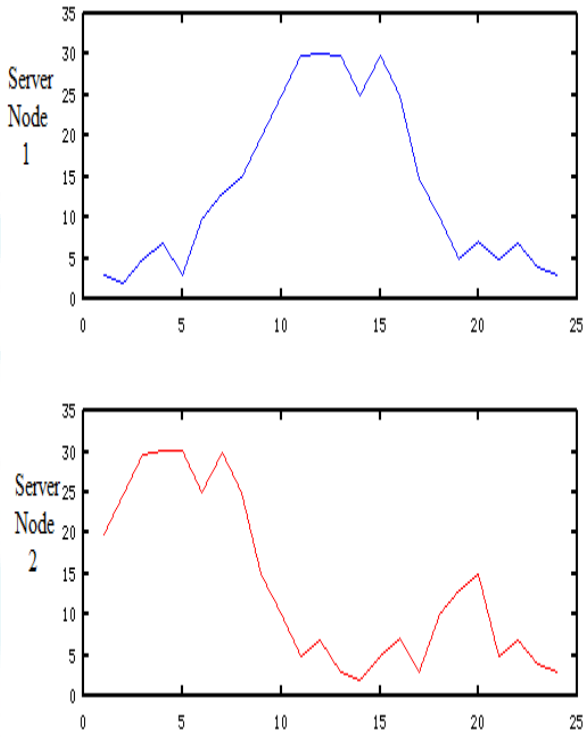

Figure 2: Time difference between the busy hours of two servers situated geographically apart

The Poisson distribution is time delayed according to the required time difference. The time difference between the busy hours can be seen in Figure 2.

3.2 Response Delaying

The response time of the web application very widely when deployed on server nodes those are geographically distributed. To simulate this behavior, the following algorithm is used.

Algorithm for Delaying Response

Step 1: Start processing request
Step 2: Read the delay parameter from the request
Step 3: IF delay parameter is not zero
Block thread for no. of seconds required as specified in delay parameter
ELSE
Move Ahead
End IF
Step 4: Process request and send Response

By the above algorithm, the response of the application is delayed by the required time.

4. Results

For testing the delaying of the response times for the web application, a sample application was deployed. Rails [5][6] was used for developing the application and the application was deployed on a Web Brick Server.

Figure 3 shows the normal response time of the server node for a GET request of the home page and it can be seen it is about 315 milliseconds. The page is requested by using chrome browser.

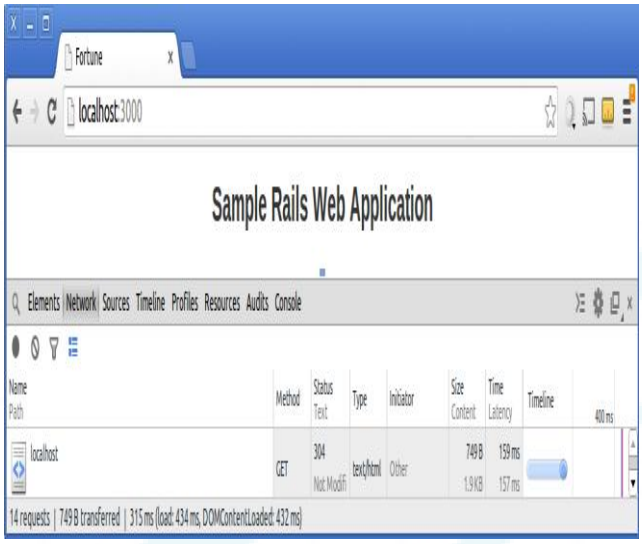

Figure 3: Web Application Response Time without delaying

In the next case a delay parameter is passed with the GET request for the home page and its value is 3s. We can see in Figure 4 that the response time is around 3.38s. Hence the response time of the application is delayed by the requested time.

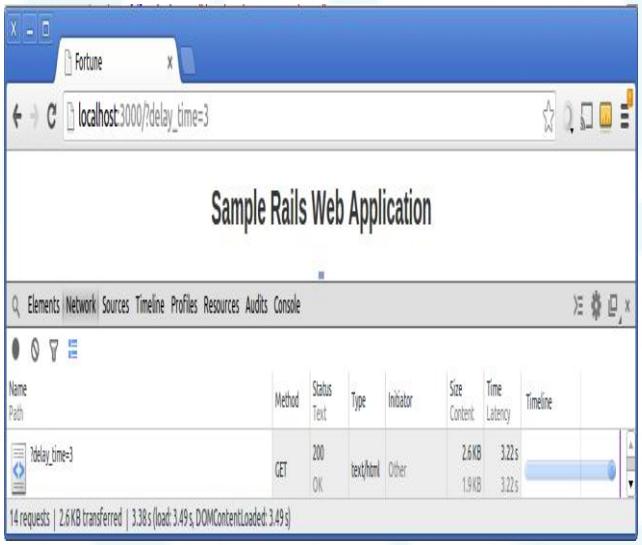

Figure 4: Web Application Response Time with delaying

5. Conclusion

The proposed method for simulating the geographically distributed server node environment is very cost effective for testing of load balancing algorithms. The whole environment can be setup in a single virtual machine and with no requirement of high capacity network connections.

References

[1] Z. Sun, D. He, L. Liang and H. Cruickshank, "Internet QoS and traffic modelling," Software, IEE Proceedings (Volume:151, Issue: 5) pp 248-255
[2] Hyoung-Kee Choi , A Behaviour Model of Web Traffic, Network Protocols, 1999. (ICNP '99) Proceedings. Seventh International Conference on, pp32-334
[3] Sireen Malik, Ulrich Killat, "A Light Weight Traffic Source for Web Traffic Simulation," in Global Telecommunications Conference Workshops, 2004. GlobeCom Workshops 2004. IEEE, pp312-319
[4] Park, C. ; University of Georgia, USA ; Shen, H. ; Marron, J.S. ; Hernandez-Campos, F., "Capturing the Elusive Poissonity in Web Traffic," in Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2006. MASCOTS 2006. 14th IEEE International Symposium on, pp189-196
[5] Viswanathan, V., "Rapid web Application Development: A Ruby on Rails Tutorial" in Software, IEEE (Volume:25, Issue: 6) pp98-106
[6] Bachle, M. ; Univ. of Cooperative Educ., Ravensburg; Kirchberg, P. "Ruby on Rails" in Software, IEEE (Volume:24, Issue: 6)pp105-108

Author Profile

Shivaprasad Balaji is currently pursuing M. in R.V. College of Engineering, Bangalore with specialization in Computer Network Engineering. He has completed his B.E in Electronics and Communication from K. S. Institute of Technology in 2012.