ISSN: 2321-7782 (Online)

Volume 3, Issue 4, April 2015

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

Load Balancing Techniques: A Comprehensive Study

Shilpi Pandey[1]
Dept. of CSE
BMSCE
Bengaluru, India

Shivika Prasanna[2]
Dept. of CSE
BMSCE
Bengaluru, India

Shreeya Kapil[3]
Dept. of CSE
BMSCE
Bengaluru, India

Rajeshwari B S[4]
Assistant Professor, Dept. of CSE
BMSCE
Bengaluru, India

***Abstract:** Due to the increase in usage of services provided over the Internet, there is excessive internet traffic. If the user's request is not distributed properly among the servers causes overloading or underutilization of some servers and an increase in the time taken to process the user requests. This calls for a need to distribute the load across the available servers evenly and optimally. Load balancing is a method of distributing the workload equally among multiple servers. By distributing the load equally among the servers, a load balancer provides a good response time, increases throughput and utilizes resources effectively. Thus load balancing is a key research issue. Many authors have proposed several load balancing techniques. In this paper, we discussed various load balancing techniques used on Session Initiation Protocol (SIP) server clusters, Cluster Computing and Cloud Computing.*

***Keywords:** Load balancer, SIP, Distributes and Cluster computing and SARA*

I. Introduction

Load balancing is an important aspect that distributes the workload across multiple servers optimally such that provides good response time and increase user's satisfaction, utilizes resources efficiently, thus improves overall performance. The load balancer accepts multiple requests from the client and distributes each of them across multiple servers based on current load on the server. Load balancing avoids a server or network device from getting overloaded with requests and helps to distribute the work evenly. For example, when the user sends a request to a server which is overloaded with some other processes, then the request needs to wait for till the serve is idle, these increases the waiting time of the request. Hence load balancer estimates the workload of each server identifies the least loaded server and schedules the request to a lightly loaded server. All requests from the clients pass through the load balancer, which forwards the requests to the appropriate server based on the current load of the server.

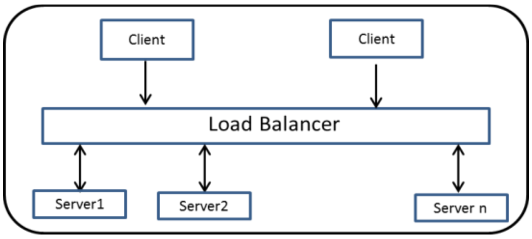

Fig.1: Basic functioning of a Load Balancer

The Session Initiation Protocol (SIP) is a general purpose signaling protocol used to control various types of media sessions. SIP protocol is used in Voice over IP (VoIP), Internet Protocol Television (IPTV), voice conferencing, instant messaging and video conferencing. SIP is a transaction-based protocol designed to establish and tear down media sessions, which is referred to as calls. Two types of state exist in SIP. The session state is created by the INVITE transaction and is destroyed by the BYE transaction. The state that exists for duration of the transaction is also created by each SIP transaction. The session-oriented nature of SIP has important effects for load balancing. Transactions corresponding to the same call must be transmitted to the same server otherwise the server will not recognize the call. Session-aware request assignment (SARA) is the process where a system assigns requests to servers, such that that server properly recognizes sessions, and subsequent requests corresponding to that same session are assigned to the same server. Thus SIP server has overheads associated with both transactions as well as sessions. These results in need of more effective SIP load balancing.

Computer clusters consist of computers connected based on their processing power, that work together to achieve a common goal. A computer cluster is a network of computers working together by creating a multi-processing environment. Clustering of computers have reduced the cost of processing power and also increases availability by making the cluster easily accessible at all times. If one system breaks down, there is no risk of data loss as the stability of the cluster is still maintained.

Cloud computing is a technology where computing resources are distributed in various data centers and these resources are provided to the customers on demand over the internet on pay-per-use basis. As the use of cloud computing increases, there is increase in traffic which needs to distribute the load equally among the servers. Thus load balancing is crucial in Cloud Computing.

Thus, in this paper, we discussed the various load balancing techniques used on Session Initiation Protocol (SIP) server clusters, Cluster Computing and Cloud Computing.

II. Load Balancing Techniques

R. Jayabal et al., [1][20][21] discussed three algorithms. **Call-Join-Shortest-Queue (CJSQ)** algorithm estimates the amount of work a server has left to do, on the basis of the number of calls (sessions) assigned to the server. Counters are maintained by the load balancer indicating the number of calls assigned to each server. When a new INVITE request is received (which corresponds to a new call), the request is assigned to the server with the lowest counter, and the counter for the server is incremented by one. When the load balancer receives an acknowledgement to the BYE transaction corresponding to the call, it decrements the counter for the server. A drawback of this approach is that the number of calls assigned to a server is not always an accurate measure of the load on a server. There may be an extensive idle periods between the transactions in a call. An improved method is **Transaction-Join-Shortest-Queue (TJSQ)** which estimates the server load based on the number of transactions assigned to the servers. The TJSQ algorithm estimates the amount of work a server has left to do on the basis of the number of transactions assigned to the server. A limitation of this approach is that all transactions are weighted equally. In the SIP protocol, INVITE requests are more expensive than BYE requests, because the INVITE transaction state machine is more complex than BYE transactions. Thus an enhanced method **Transaction-Least-Work-Left (TLWL)** is proposed. The **TLWL** algorithm overcomes from this issue by assigning different weights to different transactions depending on their relative costs. It is similar to TJSQ with the enhancement that transactions are weighted by relative overhead in the special case that all transactions have the same expected overhead. The load balancer indicating the weighted number of transactions assigned to each server maintains counters. New calls are assigned to the server with the least count. A ratio is defined in terms of relative cost of INVITE to BYE transactions.

M. Ezhilvendan et al., [17] [27] explains another aspect of **SIP transaction types**, INVITE and BYE that have different overheads. INVITE has a higher overhead over BYE. This information on overheads improves decision making by the algorithms. Decision making can also be done by any of the existing queuing methods- **CJSQ**, **TJSQ**, and **TLWL**. The algorithm TLWL makes use of the information that INVITE is more expensive than BYE and it routes calls to the servers with the least work. In view of SIP, a system assigns requests to servers and SARA ensures the subsequent calls are assigned to the same servers. In aid to this SARA concept, a load balancer is used such that it distributes the work evenly and optimally among all the available servers. One of the key characteristic of VoIP using SIP is that it takes a completely different route over the Internet than the media of running calls.

Georgios Kambourakis et al., [15][25] suggests **Round Robin scheme** which plays an important role in load balancing using SIP server clusters. When there is a new SIP request, the SIP load balancer selects the next IP address for the specific SIP server name as stored in the Domain Name System (DNS).The load balancer forwards each SIP client's request to the most appropriate SIP proxy based on workload, to serve it. SIP clients firstly communicate with the load balancer entity to find out the SIP proxy server with least workload. If the load balancer is not responding, the SIP client can communicate directly with the DNS to retrieve all the available records corresponding to SIP servers in the domain and select one. Balancing the load of SIP transactions is very important in terms of redundancy, Quality of Service and high availability. Despite the different balancing approaches that have been proposed and developed for Web applications, until now, no SIP-oriented complete balancing solution has emerged.

Alireza Karimi et al., [15] bring out **two-stage architecture** to overcome the overloading problems in SIP servers during transactions. If the server fails, it is impossible to make new calls. To prevent this overloading problem, Two Stage Architecture is implemented. Stage 1 consists of cluster of dispatchers and algorithm for load balancing is implemented in dispatchers. Stage 2 consists of cluster of SIP proxy servers and probing mechanism is used to prevent server failures

Kundan Singh et al., [18] compare various **failovers and load sharing methods** for registration and call routing servers based on the Session Initiation Protocol (SIP). The authors discussed the different types of failovers which are Client Based, DNS Based, and Database Replication Based. In Client Based, client 1 knows the IP address of both the primary and back up servers say S1 and S2. Similarly, Client 2 also knows the IP address of PI (Primary Server) and P2 (Backup Server). This helps to continue the session requesting a call in the event of failure of any one of the servers. In DNS Based, a lower numeric value is assigned to P1. Here, dynamic DNS can be used to update the IP address of P1 to P2. P2 periodically monitor P1 and update the record when P1 is dead. In Database Replication Based, client 1 registers with the primary server P1, which stores the mapping in the database D1. The secondary server P2 uses the database D2. Any change in D1 is propagated to D2. When P1 fails, P2 can take over and use D2 to proxy the call to Client 1.

Jiani Guo et al., [19] introduced two **scheduling techniques First Fit and Stream Based Mapping** for routing incoming requests. **First Fit (FF)** schedules a media unit using round-robin method. The unit is scheduled to the first computing server whose corresponding queue has a vacancy. If all queues are full, overload is indicated on all servers and the unit will not be scheduled until one of the queues is drained. In **Stream Based Mapping (SM),** the unit is mapped to a server according to the function f(c) =c mod N, where c is the stream number to which the unit belongs and N is the total number of servers in the cluster. Therefore, all the units belonging to one stream will be sent to the same server. This paper also discussed about prediction based processing time. The incoming media units are separated into Group of pictures (GOP) and processing is done based on the types of frames each GOP has. Based on execution time taken on single or heterogeneous Personal Computers, results are derived. This paper demonstrates results which are on the basis of System Scalability, Load Sharing Overhead and Video Quality.

Viney Rana et al., [22] explained load balancing concept by clustering of computers. In this paper, author states that computer clusters are better than single computer in terms of cost and performance. The major challenge is Cluster Management; the cost of administering the cluster is quite high. In this paper, a cluster based hierarchical architecture for load balancing in distributed systems is proposed. Nodes with similar processing capacity are grouped into cluster. A group of heterogeneous cluster is made for load balancing. When there is a request, algorithm randomly selects a cluster head and arranges the load of nodes in the cluster in ascending order and transfer load from overloaded node to idle node to manage the load of cluster. Cluster heads are connected with each other. Algorithm computes the total load at each cluster and transfer load between clusters at same level through cluster head. Finally to balance the load, transfers load from one level to another .Thus the whole clustered structure is balanced.

Rutuja Jadhav et al., [29] proposed a dynamic load balancing algorithm in distributed computing system. The proposed architecture takes n nodes and each node is associated with back up node. Back up nodes will not serve the tasks, but will transfer the unserved tasks to other under loaded nodes. The overloaded nodes are transferring extra load to the under loaded nodes, thus performing load balancing of the system. In case of node failure, back up nodes broadcast the failure notice and redistribute unserved tasks to under loaded nodes.

A B M Moniruzzaman [24] proposed **shared storage technology** and **two-tier architecture model** for high availability cluster with load balance infrastructure for web servers. The architecture includes four types of nodes – Load Balancer nodes, Cluster nodes, Network File System (NFS) Server nodes and SAN Box. The two-tier architecture has a load balancer as the front-end machine that balances load and routes requests to different web servers. Shared storage refers to a storage space shared by all severs to simplify the services provided by each server. Shared storage can be network file system, distributed file system or database systems.

N. S. Raghava et al., [28] discussed some of the issues encountered while designing any load balancing algorithm. **1) Distribution of nodes,** which occurs in case of Face book, Gmail etc. A well distributed system of nodes helps in handling fault tolerance and maintains the efficiency of the system. **2) Designing an algorithm** based on the state or behavior of the system which can be static or dynamic. Static algorithms do not depend on the current state of the system and have prior knowledge of the resources, but in case of sudden failure of system resources, static algorithms fail. Dynamic algorithms depend on the state of the system and do not require prior knowledge of the system. Dynamic algorithms are complex to design, but have better fault tolerance and overall performance. **3) Algorithm complexity,** a complex algorithm provides better resource utilization and throughput while simpler ones may give poor performance in terms of fault tolerance, migration and response time. Thus author suggests, based on the system requirements, care should be taken to decide a better or suitable load balancing algorithm. **4) Traffic Analyses,** peak hours differ because of the different time zones around the globe. The load balancer must be able to handle the load at all times in every location.

Divya Thazhathethil et al., [23][2] discussed load balancing by **partitioning the public cloud**. A switch mechanism is used to choose different strategies for different situations. The system has a main controller, balancers and servers. The main controller chooses the appropriate load balancer for a particular job. The balancer further selects the least loaded server and forwards the request to the identified server. Hence, this system will help dynamically allocate request to the least loaded server. Thus the entire cloud system is balancing efficiently.

III. Conclusion

A Load Balancer is used to distribute the workload among various available servers. By distributing the load among the servers based on current load on the server, provides good response time, Increases throughput, utilizes resources effectively. In this paper, we have discussed different load balancing algorithms used on various platforms like Session Initiation Protocol (SIP) server clusters, Cluster Computing and Cloud Computing.

Acknowledgement

The work reported in this paper is supported by the college through the TECHNICAL EDUCATION QUALITY IMPROVEMENT PROGRAMME [TEQIP-II] of the MHRD, Government of India.

References

1. Hongbo Jiang, ArunIyengar, Erich Nahum, Wolfgang Segmuller, Asser N. Tantawi, Charles P. Wright, “Design, Implementation, and Performance of a Load Balancer for SIP Server Clusters”, Networking, IEEE/ACM Transactions, Volume 20, Issue 4, August 2012, pp 1190-1202,DigitalObject Identifier:10.1109/TNET.2012.2183612

2. Kilroy Hughes, “The Future of Cloud-Based Entertainment”, Proceedings of the IEEE, May 13th, 2012, ISSN 0018-9219, 2012,Volume 100, pp 1391-1394, Digital Object Identifier: 10.1109/JPROC.2012.2189790

3. Mintu M. Ladani, Vinit Kumar Gupta, “A Framework for Performance Analysis of Computing Clouds”, International Journal of Innovative Technology and Exploring Engineering, ISSN 2278-3075, Volume 2, Issue 6, May 2013,pp 245-247

4. Dr.Sudha Sadhasivam, Dr. N. Nagaveni, “Design and Implementation of an efficient Two-level Scheduler for Cloud Computing Environment”, International Conference on Advances in Recent Technologies in Communication and Computing, 27-28 October 2009, IEEE, pp 884-886, Digital Object Identifier:10.1109/ARTCom.2009.148

5. R. Jayarani, Dept of Computer Technology & Applications, Coimbatore Institute of Technology, Coimbatore, India “Optimizing Transactions In Load Balancer Using Least Work Left Concept” International Conference on Electronics and Communications Engineering,28th April,2013,ISBN:978-93-83060-04-7, pp 162-166

6. Mayank Mishra, Anwesha Das, Purushottam Kulkarni, and Anirudha Sahoo, “Dynamic Resource Management Using Virtual Machine Migrations”, Communications Magazine, IEEE , Volume 50, Issue 9, ISSN 0163-6804, September 2012, pp 34– 40, Digital Object Identifier10.1109/MCOM.2012.6295709

7. Argha Roy, Diptam Dutta, “Dynamic Load Balancing: Improve Efficiency in Cloud Computing”, International Journal of Emerging Research in Management Technology, 2013, ISSN: 2278-9359 Volume 2, Issue 4, pp 78-82

8. Meenakshi Sharma, Pankaj Sharma, Sandeep Sharma, “Efficient Load Balancing Algorithm in VM Cloud Environment”, International Journal of Computer Science and Technology, pp 439-441, Volume 3, Issue 1, ISSN: 0976-8491[online], ISSN: 2229-433[print], Jan-March 2012

9. Dr.BhupendraVerma, “Efficient VM Load Balancing Algorithm for a Cloud Computing Environment”, International Journal, 2012, pp 1658-1663

10. Awadam, Uchechukwu, keqiu Li, “Improving Cloud Computing Energy Efficiency”, Cloud Computing Congress (APCloudCC), 2012 IEEE Asia Pacific. IEEE, pp 53-58, 2012, Digital Object Identifier: 10.1109/APCloudCC.2012.6486511

11. Prof.Meenakshi Sharma and Pankaj Sharma “Performance Evaluation of Adaptive Virtual Machine Load Balancing Algorithm”, International Journal of Advanced Computer Science and Applications, 2012, Volume 3, Issue 2, pp 86-88

12. Abhay Bhadani, Sanjay Chaudhary, “Performance Evaluation of Web Servers using Central Load Balancing Policy over Virtual Machines on Cloud”, Proceedings of the Third Annual ACM Bangalore Conference. ACM, 2010, ISBN: 978-1-4503-0001-8, Digital Object Identifier:10.1145/1754288.1754304

13. Tamara Celime Winegust, “The Impact of Cloud Computing and content streaming on copyright in the entertainment industry”, Vol 4, Issue 1, Art 1, 2012

14. Schulzrinne, Henning Rosenberg, J. “The Session Initiation Protocol: Internet-centric signaling”, Communications Magazine, IEEE Journal, Volume 38, Issue 10, ISSN 0163-6804, pp 134-141, 2000, Digital Object Identifier: 10.1109/35.874980

15. Alireza,Karimi, Mehdi AgahSarram, Mohammad Ghasemzadeh “Two stage architecture for load balancing and failover in SIP networks”, Middle-East Journal of Scientific Research 6, Volume 6, Issue 1, ISSN 1990- 9233, pp: 88-92, 2010

16. “Cloud Computing Technology Spotlight Articles”, cloudcomputing.ieee.org/publications/technology-spotlight

17. M. Ezhilvendan, J. Gunasekaran, P. Vijayanand, B. Sivakumar and J. Nagaraj, “The State of the Art in Locally Distributed Web-Server Systems”, International Journal of Electronics Communication and Computer Technology, January 2013,Volume 3, Issue 1, ISSN:2249-7838, pp 341-346.

18. Kundan Singh and Henning Schulzrinne, “Failover and Load Sharing in SIP Telephony”, Technical Report, 2004

19. Jiani Guo and Laxmi Narayan Bhuyan, “Load Balancing in a Cluster-Based Web Server for Multimedia Applications” , IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, November 2006, Volume 17, Issue 11, ISSN 1045-9219, pp 1321-1334, 2006, Digital Object Identifier: 10.1109/TPDS.2006.159

20. Sandeep Kholambe, “Implementation of Load Balancing Using SIP”, International Journal of Computer Science and Mobile Computing, Volume 3, Issue 7, July 2014, pp 817-822

21. R. Jayaba, R. Mohan Raj, “Design and Implementation of Locally Distributed Web Server Systems Using Load Balancer”, International Journal of Engineering Sciences & Research Technology, February 2014 ISSN 2277-9655, pp 708-713

22. Viney Rana, Sunil Kumar Nandal, “Efficient Load Balancing in Clusters in Hierarchical Structure”, International Journal of Engineering and Computer Science ISSN: 2319-7242, Volume 3, Issue 7, July 2014, pp 7177-7180

23. DivyaThazhathethil, NishatKatre, Jyoti Mane-Deshmukh, Mahesh Kshirsagar, “A Model for load balancing by Partitioning the Public Cloud”, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 3, March 2014, ISSN 2277 128X

24. A B M Moniruzzaman, Syed Akther Hossain, “A Low Cost Two-Tier Architecture Model For High Availability Clusters Application Load Balancing”, arXiv preprint arXiv:1406.5761. 22nd June, 2014

25. Georgios Kambourakis, Dimitris Geneiatakis, Tasos Dagiuklas, Costas Lambrinoudakis, Stefanos Gritzalis, “Towards effective SIP load Balancing”, Third Annual VoIP Security Workshop, 7 June 2006

26. Jong Yul Kim, Gregory W. Bond, Eric Cheung, Thomas M. Smith, Henning Schulzrinne, “An Evaluation Framework for Highly Available and Scalable SIP Server Clusters”, Proceedings of the 5th International Conference on Principles, Systems and Applications of IP Telecommunications. ACM, 2011, ISBN: 978-1-4503-0975-2, Digital Object Identifier:10.1145/2124436.2124438

27. P. Lekha Chandra, A. Rama Satish, “Implementation of a Load Balancer for Instant Messaging over SIP Server Clusters with Improved Response time”, International Journal of Engineering Research and Applications,ISSN: 2248-9622, Volume. 4, Issue 1 (Version 3), January 2014, pp.58-62 [28] N. S. Raghava and Deepti Singh, “Comparative Study on Load Balancing Techniques in Cloud Computing”, Open Journal of Mobile Computing and Cloud Computing, Volume 1, August 2014, pp 18-25

28. Rutuja Jadhav, priyadarshini, Snehal Kamlapur “Performance Evaluation in Distributed System using Dynamic Load Balancing”,International Journal of Applied Information Systems (IJAIS), 2012,volume. 2, issue 7, pp 36-41