IJRRECS/October 2013/Volume-1/Issue-6/1429-1434 **ISSN 2321-5461**

INTERNATIONAL JOURNAL OF REVIEWS ON RECENT ELECTRONICS AND COMPUTER SCIENCE

AN APPROACH TOWARDS THE LOAD BALANCING STRATEGY FOR WEB SERVER CLUSTERS

B.Divya Bharathi[1], N.A. Muneer[2], Ch.Srinivasulu[3]

[1]Dept of IT, J.B. Institute of Engineering & Technology, Hyderabad, A.P, India

[2]Assistant Professor, Dept of IT, J.B. Institute of Engineering & Technology, Hyderabad, A.P, India

[3]Associate Professor, Dept of IT, J.B. Institute of Engineering & Technology, Hyderabad, A.P, India

ABSTRACT:

The practice that directs numerous types of media sessions is known as session initiation protocol and it is expanding in the fields of voice conferencing, instant messaging and video conferencing. Intended for load balancing, the session oriented nature of session initiation protocol has significant inference. In the direction to set up and tear down media sessions that are often known as calls, session initiation protocol which is a transaction based protocol was projected. The process in which a system allocates requests to servers so that the sessions are appropriately predicted by that server, and the requests of the following that are equivalent to the identical session are allocated to the similar server is known to be a Session-aware request assignment. Load balancing algorithms of the novel type depends on allocating of calls to servers by means of picking the server with the least quantity of work allocated are the new algorithms of load balancing based. In the session initiation protocol two types of session states were present. By means of the INVITE transaction the initial session state is constructed and is destroyed by the transaction of the BYE transaction. Numerous novel load balancing algorithms were introduced for allocating the requests of Session Initiation Protocol to a cluster of session initiation protocol servers and among them a few are Call-Join-Shortest-Queue, Transaction-Least-Work-Left, and Transaction-Join-Shortest-Queue.

Keywords: Session initiation protocol, Session-aware request assignment, Load balancing algorithms, Media session.

1. INTRODUCTION:

The protocol that directs numerous types of media sessions is known as session initiation protocol and it is expanding in the fields of voice conferencing, instant messaging and video conferencing. Towards both with transactions and with sessions SIP has overheads that are linked to and can consequence in more optimized session initiation protocol balancing of load [4]. Session initiation protocol has a lot of characteristics that makes it unique from the protocols such as hypertext transfer protocol. The transaction of every session initiation protocol constructs a state that subsists for the interval of that transaction. To set up and tear down media sessions that are often known as calls, session initiation protocol which is a transaction based protocol was designed [11]. In the session initiation protocol two types of session states were present. By means of the INVITE transaction the initial session state is constructed and is destroyed by the transaction of the BYE transaction. For load balancing, the session oriented nature of session initiation protocol has significant inference. [9] The process in which a system allocates requests to servers so that the sessions are appropriately predicted by that server, and the requests of the following that are equivalent to the identical session are allocated to the similar server is known to be a Session-aware request assignment. Transactions that correspond to the similar call have to be routed to the similar server; or else, the server will not be familiar with the call [14]. Numerous novel load balancing algorithms were introduced for allocating the requests of Session Initiation Protocol to a cluster of session initiation protocol servers and among them a few are Call-Join-Shortest-Queue, Transaction-Least-Work-Left, and Transaction-Join-Shortest-Queue [3]. To pick the least loaded server, the load balancer has the freedom for the initial `INVITE` transaction of a call the load balancer. The important improvements in response time that Transaction-Join-Shortest-Queue and Transaction-Least-Work-Left make available a compelling reason for systems such as these to use the

algorithms. Intended for quite a lot of the load balancing algorithms, was put into practice, and these assignments may possibly be based on the estimates for each server [12].

2. METHODOLOGY:

The arrangement of the load balancer was shown in fig1. The requests are received by the receiver and are subsequently parsed by means of the Parser [1] [6]. The request that corresponds to a previously existing session was determined by the session recognition module by means of querying the state of Session that is put into practice as a hash table.

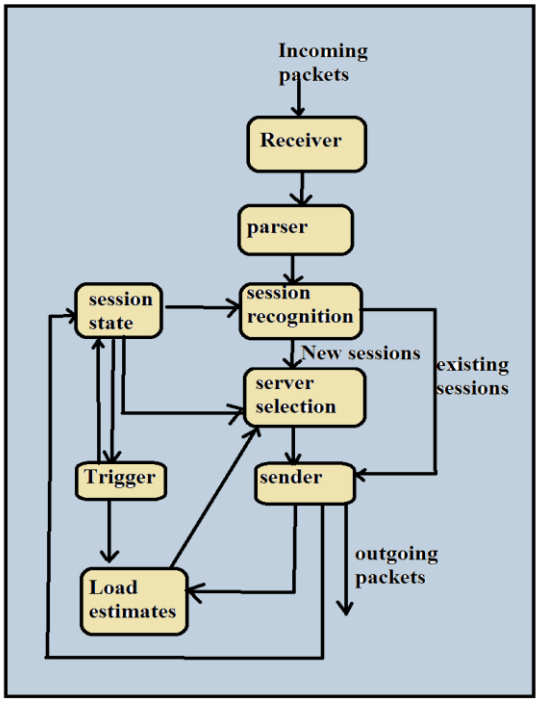

Fig1: An overview of load balancer architecture

If so, to the server to which the session was formerly allocated, the request is consequently forwarded. If not, the module of the Server Selection allocates the new session to a server by means of one of the algorithms [8]. Intended for quite a lot of the load balancing algorithms, was put into practice, and these assignments may possibly be based on the estimates for each server [15]. The requests were forwarded to the servers by the sender and update the estimates of the load and session states as required. Responses were also received by the receiver that are send by the servers and the response is recognized by the module of the session recognition that was to be received by the client and by means of querying the session state the information was obtained [13]. The response was sent by the sender to the client and updates the estimates of the Load and Session State as necessary. The Session State and the estimates of the Load Estimates were updated by the trigger module subsequent to the expiry of the session. Load balancing algorithms of the novel type depends on allocating of calls to servers by means of picking the server with the least quantity of work allocated are the new algorithms of load balancing based. Some of them include

Call-Join-Shortest-Queue: The quantity of work, a server has left to perform was estimated by this algorithm based on the number of calls that are assigned to the server [2] [5]. By the load balancer the counters are preserved that indicates the number of calls allocated to each server. The request is assigned to the server after receiving a request of new INVITE with the least counter, and subsequently the counter is incresed by means of one. The number of calls allocated to the server is not forever an appropriate evaluate of the load on a server is the limitation of this approach [7]. Transaction-Join-Shortest-Queue: To approximate the server load on the basis of number of transactions allocated to the servers is the alternative method. Based on the number of transactions that are allocated to the server the algorithm approximates the work that was remained by the server [10]. All the transactions are subjected uniformly is the limitation of this approach. Transaction-Least-Work-Left: By means of assigning various weights to different transactions on the basis of their comparative costs is the issue addressed. The transactions are weighted in relative overhead; it is comparable to Transaction-Join-Shortest-Queue with the improvement in the particular case that all transactions have the same accepted transparency; Transaction-Least-Work-Left and Transaction-Join-Shortest-Queue and are the same.

3. RESULTS:

Significant differences were observed in the response times of the various load balancing algorithms. Performance is limited by means of the CPU processing power of the servers performance is limited and not by memory. To pick the least loaded server, the load balancer has the freedom for the initial `INVITE` transaction of a call the load balancer. The important improvements in response time are that the algorithms of Transaction-Least-Work-Left and Transaction-Join-Shortest-Queue algorithm make available a compelling reason for systems such as these to use the algorithms. In terms of how well throughput scales with perform with the increasing numbers of back-end servers; the load balancing algorithms were examined. Calls-Join-Shortest-Queue is considerably not as good as than the others; in view of the fact that it does not differentiate times of the call hold in the approach that the algorithms of the transaction-based carry out.

4. CONCLUSION:

The convention that directs numerous types of media sessions is known as session initiation protocol. In the session initiation protocol two types of session states were present. Intended for quite a lot of the load balancing algorithms, was put into practice, and these assignments may possibly be based on the estimates for each server. By means of the INVITE transaction the initial session state is constructed and is destroyed by the transaction of the BYE transaction. Numerous novel load balancing algorithms were introduced for allocating the requests of Session Initiation Protocol to a cluster of session initiation protocol servers and among them a few are Call-Join-Shortest-Queue, Transaction-Least-Work-Left, and Transaction-Join-Shortest-Queue. The important improvements in response time are that the algorithms of Transaction-Join-Shortest-Queue and Transaction-Least-Work-Left algorithm make available a compelling reason for systems such as these to use the algorithms. Significant differences were observed in the response times of the various load balancing algorithms. To pick the least loaded server, the load balancer has the freedom for the initial `INVITE` transaction of a call the load balancer.

REFERENCES:

[1] Zongming Fei, Samrat Bhattacharjee, Ellen Zegura, and Mustapha Ammar. A novel server selection technique for improving the response time of a replicated service. In *Proceedings of IEEE INFOCOM*, 1998.

[2] Darrell C. Anderson, Jeffrey S. Chase, and Amin Vahdat. Interposed request routing for scalable network storage. In *USENIX OperatingSystems Design and Implementation (OSDI)*, San Diego, California, USA, October 2000

[3] Charles Shen, Henning Schulzrinne, and Erich M. Nahum. Session initiation protocol (SIP) server overload control: Design and evaluation. In *Principles, Systems and Applications of IP Telecommunications (IPTComm)*, pages 149–173, Heidelberg, Germany, July 2008.

[4] G. Goldszmidt, G. Hunt, R. King, and R. Mukherjee. Network dispatcher: A connection router for scalable Internet services. In *Proceedings of the 7th International World Wide Web Conference*, Brisbane, Australia, April 1998.

[5] Jonathan Rosenberg, Henning Schulzrinne, Gonzalo Camarillo, Alan Johnston, Jon Peterson, Robert Sparks, Mark Handley, and Eve Schooler. SIP: Session initiation protocol. RFC 3261, Internet Engineering Task Force, June 2002.

[6] Erich Nahum, John Tracey, and Charles P. Wright. Evaluating SIP proxy server performance. In *17th International Workshop on Networking and*

Operating Systems Support for Digital Audio and Video (NOSSDAV), Urbana-Champaign, Illinois, USA, June 2007.

[7]Mohit Aron, Darren Sanders, Peter Druschel, and Willy Zwaenepoel. Scalable content-aware request distribution in cluster-based network servers. In *Proceedings of the USENIX 2000 Annual Technical Conference*, San Diego, CA, June 2000.

[8] Kundan Singh and Henning Schulzrinne. Failover and load sharing in SIP telephony. In *Proceedings of the 2005 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'05)*, July 2005.

[9] Nortel Networks. Layer 2-7 GbE switch module for IBM BladeCenter. http://www-132.ibm.com/webapp/wcs/stores/servlet/ProductDisplay?productId=4611686018425170446&storeId=1&langId=-1&catalogId=-840.

[10] F5. F5 introduces intelligent traffic management solution to power service providers' rollout of multimedia services. http://www.f5.com/ news-press-events/press/2007/20070924.html.

[11] Jim Challenger, Paul Dantzig, and Arun Iyengar. A scalable and highly available system for serving dynamic data at frequently accessed Web sites. In *Proceedings of ACM/IEEE SC98*, November 1998.

[12] Vivek S. Pai, Mohit Aron, Gaurav Banga, Michael Svendsen, Peter Druschel, Willy Zwaenepoel, and Erich M. Nahum. Locality-aware request distribution in cluster-based network servers. In *Architectural Support for Programming Languages and Operating Systems*, pages 205–216, 1998.

[13] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, and T. Berners-Lee. Hypertext transfer protocol – HTTP/1.1. RFC 2068, Internet Engineering Task Force, January 1997.

[14] IBM. Application switching with Nortel Networks layer 2-7 gigabit ethernet switch module for IBM BladeCenter. http://www.redbooks. ibm.com/abstracts/redp3589.html?Open.

[15] Henning Schulzrinne, Stephen Casner, Ron Frederick, and Van Jacobson. RTP: a transport protocol for real-time applications. RFC 3550, Internet Engineering Task Force, July 2003.