International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 4, April - 2014

Efficient Distribution of Incoming Traffic with Load Balancer

Ashwini M. Hiremath
PG student, Dept of CSE,
CIT Gubbi,
Karnataka, India

Mr. Anilkumar G
Associate Proffessor, Dept of CSE,
CIT, Gubbi,
Karnataka, India,

Abstract: **The increase in traffic on the World Wide Web is augmenting users perceived response time from popular websites, especially in congestion with special events. A single server cannot provide the needed scalability to handle large traffic volumes to match rapid changes in the number of clients. In order to improve both throughput and response time load balancing algorithm for distributing session-initiation protocol requests to a clusters of SIP servers are introduced. Proposed load balancer improves the performance by integrating the features of introduced algorithms. i.e load on the back-end server ,Knowledge of the SIP protocol, Processing cost for dfferent transaction, variability in call length.**

Keywords- laod balancing, Session Initiation Protocol(SIP), VoiP, Dispatcher

I.INTRODUCTION

Session Initiation Protocol (SIP) is a general purpose signalling protocol,widely used to control various types of multimedia communications such as voice and video calls over Internet protocol[6]. The SIP protocol used for creating, modifying and terminating two party (unicast) or multiparty (multicast) session consists of one or several media streams.

SIP is widely used to establish and terminate Voice over Internet Protocol(VoIP) calls. A typical SIP session involves a client requesting a session with a SIP server. After the request is received the SIP server returns a response to the users indicating the availability of the session ,users are indentified by a SIP address which is similar to an email address.

Individual servers will handle hundred or thousands of users ,large scale ISPs need to support customers in the millions ,so in order to support millions of requests server clusters are introduced, server clusters improves system availability, application scaling and simplifies the system management ,server clusters have the ability to scale that service, but in order to manage the increasing load and customer demands is to use some form of load balancing dispatcher[2] that distributes load across multiple servers .here focusing on the evaluation of several algorithms for balancing load across multiple servers[1].

The Load balancer analysis the traffic in server clusters means, if the server is already processing the request the load balancer will detect the traffic in that particular server, then it will passes the request to the next available server.

SIP has important feature for load balancing i.e session oriented nature, Transaction corresponding to the same call must be routed to the same server otherwise the server will not recognize the call. SIP messages traverse the SIP overlay network routed by proxies to find eventual destinations. Once end points are found, communication is typically performed in a peer to peer fashion, an end point can also be a server providing such as voice mail, fire-walling, voice conferencing , mainly focusing on the scaling the server.

II.PROBLEM STATEMENT

HTTP [4] load balancer do not take sessions into account in making Load-Balancing decisions , and it is content unaware because it doesn't examine the contents of a request, while SIP has important implication for load balancing , Transaction corresponding to the same call must be routed to the same server , otherwise server will not recognize the call. This can be done by a process called SARA (Session Aware Request Assignment).

III.METHODOLOGY

Methodology consists of

A. SIP Users, Agents, Transaction and Messages

B. SIP Message flow.

C. Load Balancing algorithms.

A. SIP User, Agents, Transaction and Messages

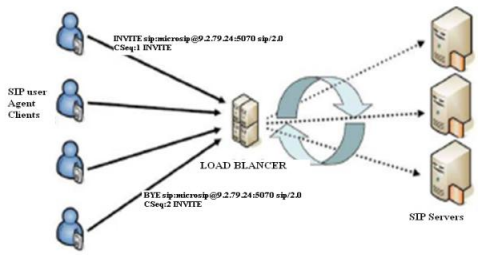

Figure 1:General Architecture.

A SIP Uniform Resource Identifier (URI) uniquely indentifies a SIP user e.g., sip:hongbo@gmail.com

SIP users employ endpoints known as User Agents, These entities initiate and receive sessions .User agents are further decomposed into User Agent Clients (UAC) and User Agent Server (UAS). SIP uses HTTP like request /response transaction, A transaction consists of a request to perform particular method (eg: INVITE, BYE, CANCEL). Response may be provisional (100 TRYING, 180 RINGING) and final (200 OK) and when final response is received then only the transaction is completed.

SIP is a text-based protocol that derives much of its syntax from HTTP [12]. Messages contains headers and additional bodies depending on the type of messages .An important header is to notice the call_ID header which is global unique indentifier for that session.

B. SIP message flow

An INVITE message creates a transaction , but also a session if the transaction completes successfully , A BYE message creates new transaction and when the transaction is complete ends the session . SIP messages are routed through the proxy server. Here a call is initiated with the INVITE message and accepted with the 200 OK message .Media is exchanged and then the call is terminated using the BYE message.

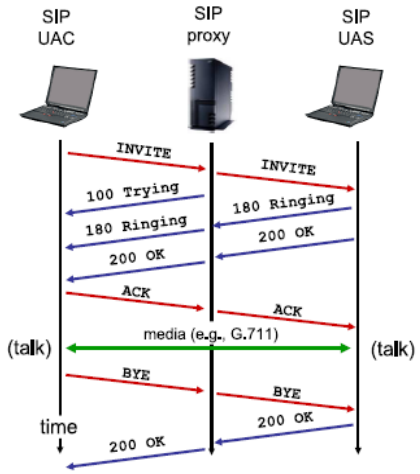

Figure 2: Message flow

C. Load Balancing Algorithms

Important factors of a load balancer:

- A important fact of a load balancer is that request corresponding to the same call are routed to the same server. The load balancer has the freedom to pick a sever only on the first request of a call and subsequent request corresponding to the same call must go to the same server.
- The method on which the load balancer is assigning calls to the server is by picking the server with the least amount of work assigned but not yet completed
- Here, the load balancer estimates the work assigned to a sever based on the request it has assigned to the server and the response it has received from the server.

Load balancing algorithms presents how the proxy server will choose the SIP server to handle the multiple requests[5], the algorithms are:

1) CALL-JOIN SHORTEST QUEUE (CJSQ): CJSQ algorithms estimates the work that the server has left to do based on the number of calls (sessions) assigned to the server , counters are maintained by the load balancer , It is incremented when the server receives a INVITE request and it is decremented by the BYE corresponding to the all.

An advantage of CJSQ is that can be used in environment in which the load balancer is aware of calls , and limitation of this approach is the number of calls assigned to a server is not always an accurate measure of a load on a sever.

2) TRANSACTION-JOIN SHORTEST QUEUE (TJSQ): TJSQ algorithm estimates the server load based on the number of transaction assigned to the server, As in CJSQ , here also counters are maintained based om INVITE and BYE transaction

A limitation of this approach is all transaction are weighted equally, as in the SIP protocol, INVITE requests are more expensive than any other NON-INVITE transaction.

3) TRANSACTION –LEAST WORL LEFT (TLWL): TLWL and TJSQ algorithms are same, counters are maintained by the load balancer indicating the weighted number of transaction a sever is currently handling. TLWL algorithm addresses the limitation of TJSQ by assigning the different weights to different transaction. Eg: INVITE transaction is more expensive than the BYE transaction ,so it is taken as in the ratio INVIT E : BYE i.e 1.75 : 1 so the total server has a load of 2.75 and it assigns the cost by relative overhead.

IV. ANALYSIS

Load balancer which acts as the proxy server assigns the request to the sever clusters based on the least amount of work left , if the server has large amount of work has left to do then the load balancer chooses the next server in the clusters which indicates traffic found in that particular server, before the traffic found both high throughput and low response time is achieved.

V. CONCLUSION

Load balancer performs SARA (Session Aware Request Assignment) to ensure that SIP transactions are routed to the proper back-end server that has the appropriate session- state thereby achieving lower response time and the Transaction –least work left algorithm results in the best performance as it calculates the cost by relative overhead that are associated both with sessions and transactions and taking advantage of this fact can result in more optimized load balancing.

VI. RESULTS ANALYSIS

The proposed load balancing algorithms are evaluated by considering the fact that if the server has large amount of work has to do then traffic found in servers in that case high throughput is not achieved, while before traffic found means the server is not busy high throughput is achieved and the lower response time is achieved.

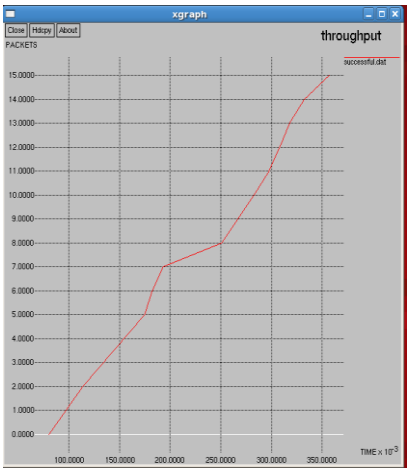

Figure3:Throughput is achieved before traffic found.

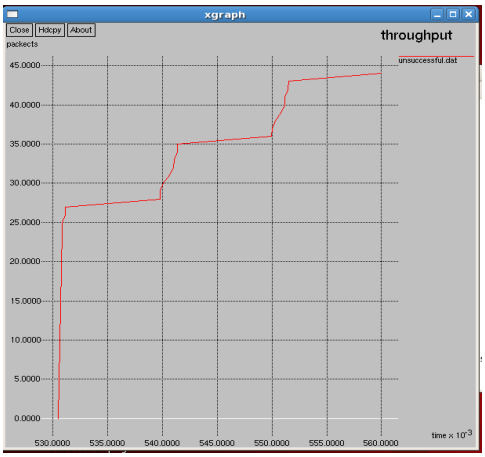

Figure 4: Throughput is achieved after traffic found

REFERENCES

1. 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
2. Gianfranco Ciardo, Alma Riska, and Evgenia Smirni. EQUILOAD:A load balancing policy for clustered Web servers. Performance Evaluation, 46(2-3):101–124, 2001
3. 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.
4. 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.
5. 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
6. Charles Shen Henning Schulzrinne, and Erich M. Nahum. Session initiation protocol (SIP) server overload control: Design and evaluation.In Principles, Systems and Applications of IPTelecommunications(IPTComm), pages 149–173, Heidelberg, Germany, July 2008.