REVIEW ON TLS OR SSL SESSION SHARING BASED WEB CLUSTER LOAD BALANCING

Dipesh Gupta[1], Hardeep Singh[2]

[1]*School of Computer Science & Engineering, Lovely Professional University, Punjab, India*
[2]*School of Computer Science & Engineering, Lovely Professional University, Punjab, India*

Abstract

Internet users increase the traffic on the servers and server security is the major concern with which the user's privacy needs to be protect. TLS (Transport Layer Security) is a widely deployed protocol that establishes a secure channel between communicating parties over the internet. But TLS/SSL has huge impact on webserver's performance by degrading it to a considerable amount. When TLS/SSL session is generated it is broadcasted to all servers in the cluster with which session reuse can be used to save time in negotiation. TLS Handshake and Session resume is occur at the server end so in future if client requests again and its session is not expired then it can again joins that its own session without renegotiating which saves the session initialization time. Ultimately a new load balancing cluster design is proposed that can share TLS sessions in the cluster to effectively improve the performance of TLS web cluster. The web cluster server shares the sessions of users within the cluster. The another technique for improving the latency and throughput of the server SSL/TLS with backend forwarding technique is compare and is analyzed. The traditional method has flaws in the load balancing of the server but with the new implanted technique on the server improves the performance during the high load .The results are reviewed with 16 and 32 node cluster system. With new technique the latency of system has been decreased by the 40 % and throughput of the system is extremely better than classical balancing technique.

Keywords: *TLS/SSL session sharing, Web cluster, TLS/SSL session reuse*

----------***----------

1. INTRODUCTION

With the enhancement in internet technologies, web based applications like ecommerce on shopping etc. are getting popular. These applications are insecure unless they use a secure channel to provide the data security. TLS (Transport Layer Security) is a widely deployed protocol to provide a secure channel between communicating parties. Although communicating using TLS, results in critical load on servers and degrade their overall performance. Load balancing cluster for TLS web system is a popular solution [5]. There is a higher probability that traditional load balancing of server leads to degrade the system performance because of the high load. Therefore, improving the TLS server performance is critical important and now it's a major issue in the research area how servers can more efficient and advance so that on load they can easily manage that all using the efficient load balancing technique.

In this paper we have analyzed and compare the processing of the TLS based cluster for load balancing. After that Handshaking protocol and resumption of session are modified to share the TLS in cluster. A new advance solution better than classical method is that which leads to reduce in latency and increase the throughput of the server. [6]

2. SSL/TLS WORKING

The SSL works on the application and transport layers. The TLS session is established with a handshake between the server and the client. The client starts the session by sending a "Client Hello" message with the *ciphersuites* to the client. The server replies "Server Hello" identifying the strongest cipher suite supported by both the parties and the server's certificate. The client application authenticates the certificate and generates a random number called the *pre-master key*. The client encrypts the pre-master key with the server's public key and sends it to the server. The server decrypts the pre-master key with its private key. Both parties use the pre-master key to generate the session key. At this point, the client and server exchange the Change Cipher Suite message to indicate that all future communications will be encrypted with the session key. Finally, both parties send a finished message to each other.

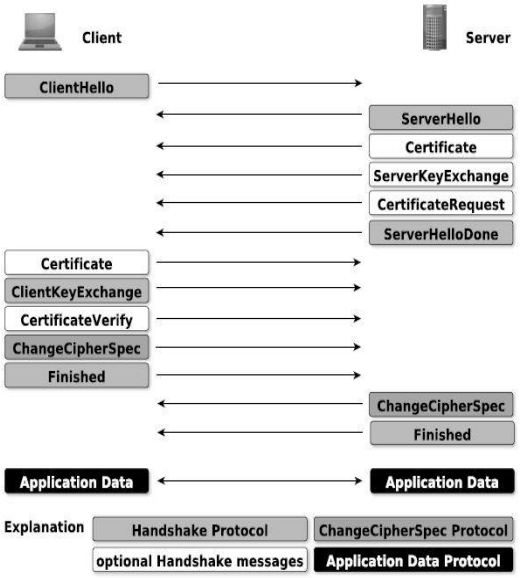

Fig 1: SSL/TLS Handshaking process

3. LOAD BALANCING WITH SESSION SHARING

3.1 Session Reuse

Whenever the client tries to connect with the server using the TLS, the session starts for it. Server sends the hello packet to initialize the communication so that if there is any closing connection which is going to terminate soon, that might reinitialize and the connecting time should be less. Mostly this happens when the connection time is not out. If again a client needs to connect with the server, then it needs to send only the session id as a hello message. If there is a session found on the server with that particular id, then that server will resume the session again. The main advantage is that, if an old session is started again, then the time which spends to renegotiate, that can be saved. If there is no session on the server with the sent session id, then new session will be initialized [9, 16].

3.2 Sharing Session

In this when the client connects with the server, firstly it connects with the load balancer which is the front end on the server cluster. Then that load balancer takes the request and send it forward to the cluster server which is the back end. In the normal TLS session, client send request to the server and starts the communication after verification. But in this method the each session which is created is broadcasted to all the servers in the cluster. That means session is inherited and another server can use that session also [5, 16].

Consider the follow Figure 2 Web Server B can reuse the session which is established on the Web Server A and User B, because the sever B inherits the session from the server A.

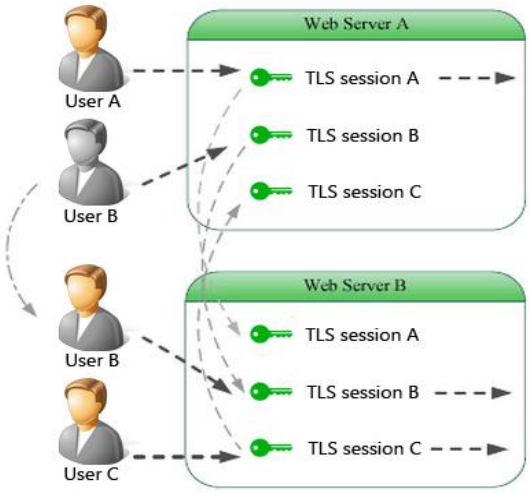

Fig 2: Sharing of SSL/TLS session within web server cluster

3.3 Experimental System Requirements

Hardware Requirements: TP-LINK TL-R480E; Switcher: 100Mbps; 5 Servers used to test with the same configuration: Pentium 4 2.0GHz C'PU, 256MB Memory, 10/100Mbps NIC and ordinary category 5 twisted pair.

Software Requirements:
Operating System: Redhat Linux 9.0; Web Server: Apache HTTP Server V2.2.4; WebBench 4.1; Modified OpenSSL 0.9.7d and added into to the code which can share the TLS session [11,16].

Performance Result: In this research the various algorithms are tested to perform the test and to obtain better results in different length of time and speed.

Table 1: Comparison of no. of requests per second in the cluster with no session reuse, session use and session sharing [8]

	2-node	3-node	5-node
No session reuse	26	39	55
Session reuse	247	352	588
Session sharing	266	383	624

The methodology may be used for better and most effective way for the TLS session among different webservers within a cluster. But storing all the sessions of the cluster of each individual server may lead to increase the load of the server and retrieval of the information of the session [8,16].

4. LOAD BALANCING WITH SSL WITH BACKEND FORWARDING

In the existing problem, the SSL with session technique is used for the load balancing in the web cluster and round robin (RR) is also used. This technique was not effective to handle the load .The major flaw was that the algorithm was not working as expected. This model has the latency problem.

Implemented Technique Advancements
The new technique has the session with back end forwarding instead of the session with SSL. With the advancement, the latency problem is overcome and gives better the throughput than the previous one. The new technique gives the 40 % better result i.e. it reduces the latency by 40 %.

System Specifications
The work is carried out on the 500 MHz Ultra Sparc uniprocessor, running on the Solaris 2.9 with 1 GB memory. The RSA, RC4, and MD5 are the most common web cipher suits supported and used by the web browser as well as by the servers. In this algorithm RSA 2048 bit key is used instead of 1024 bit [7,17].

4.1 SSL with session

The SSL with session distributor maintains the client request, receives and forward it to the application server. The advantage of SSL with the session is that once the connection is established, the session gets started.

If a server has a client which is frequent and require heavy computations, then that can't be forwarded to lightly loaded server [4,17].

4.2 SSL_with_backend (Backend_forwarding)

The SSL with backend forwarding is making less severe the limitation of the backend with the session in the load balancing module in the distributor to obtain. The load of the i^{th} server Li is calculated by number of open connections. The servers are denoted with the N. If the cluster consists of N servers where the i^{th} node is denoted by ni. Here there are two requests, static and dynamic rstat and rdyn respectively with wstat and wdyn. Here i refers the some value for some server.

The average processing time for the static and dynamic requests are weighted with the values for the calculation of the load of the i^{th} server and the threshold values T1 and T2.

If Li>T1, then ni forwards the request along with negotiated session key to one of the servers. Finally the server which receives the request from the request generated node, that server then encrypts the dynamic content using the forward session key and returns it to the initial node which further sends the response back to the client [7,17].

The SSL_WITH_BF is focused at mitigating the problem of SSL with session with back end forwarding technique to balance the load on the servers within the cluster. The distributor which is the front end on the application server is updating itself every 300 ms .The load on the server Li is calculated on the behalf of number of the open connections .The cluster has the N servers and the i^{th} server is denoted as i^{th} server by ni.

Application server with back end forwarding technique. The average processing time for the static and dynamic requests are weighted with the values for calculation of load of the i^{th} server and the threshold values T1 and T2.

If Li>T1, then ni forwards the request along with the negotiated session key to one of the servers.[17]

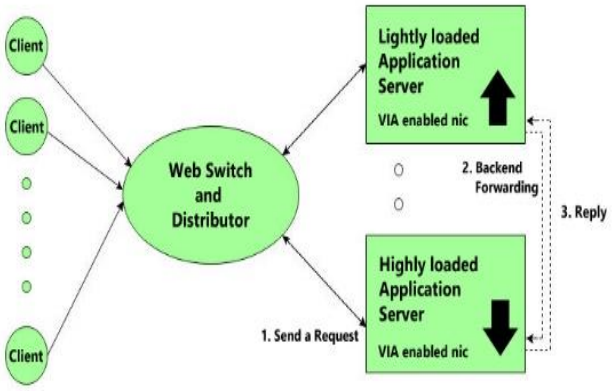

Fig 3: Working of SSL Based Cluster Load Balancing

5. METHODOLGY ANAYLYSIS

In the SSL with session sharing methodology the web clusters use the session sharing process to share the load session among different servers within the same cluster. The sessions which are established on server is shared onto another.

When a client requests to as sever the load balancer check the loads on the server. If the server has the peak load then balancer switches the user on to another server. The client which requested the server for session, and if that session Id for particular client found on server than server will use that session before its time out with this its result into time decrease in renegotiation. But the problem with system is the server has to store the all session's ids, with this server takes time to store and fetch the session information stored on the server. The overall system performs degrade during the operation of the storing and fetching the session information during the process.

But as comparison with SSL_WITH_BF technique the load balancing is done at backend that is at sever end. In this the load of the server is calculated by the open no. of connections at that time. The average time varies on nature of the request made on the server whether the requested one is static or dynamic request. If the request is the static that is simply e.g. html part or some url that may also know as the light load applications. But with the dynamic application this system gets the busy i.e system gets busy with high load. The balancer at server cluster end choose the nature of the requested application and forwards it to the server. The lightly loaded applications are forwarded to lightly loaded server and heavy applications like dynamic webpages e.g ASP , .Net or server executable code pages are move to the heavy load server so that server can execute them accordingly. This technique has the load balancing at the server end and it is done by load balancer of the cluster

server .The whole balancing is done on the server via NIC in between the servers. In this technique the previous problem like session storing and fetching which was the major issue of system performance degradation on the server that is overcome because now the serve is managing the whole load on the basis of the application nature. Therefore now the session reuse is not a big problem and session information can be reuse easily.

With the implantation of technique SSL_WITH_BF the server performance is improved very well, the server latency is degraded which results into the better throughput. Overall with this new method the latency is decreased by up to 40 % and the server has highest throughput. That means now the server has the better resource utilization. Better resource utilization leads to better performance. The method has been experimented on the different server cluster nodes. The server cluster nodes are 16 and 32 nodes. With this as the nodes on the cluster is getting increased and also the load on the server is also increased that results into efficiently working of the system with lower latency and better throughput of the system.

6. CONCLUSIONS

Load Balancing with TLS session provides the better cluster load balancing in which session reuse can be used effectively but the problem is that storing the all session of clusters on the server is huge big problem which will increase the load on the server and performs degradation in fetching the information in future. But with the SSL_with_bf technique experimental results are provided with 16 node and 32 node cluster shows that the session reuse with SSL with session is critical to improve the performance of the servers with implanted algorithm performs better and provides the efficient performance.

This Technique enhances the performance by 40% better than the SSL with session. Also in this algorithm, the session storage doesn't need to resume the session as it was required in the SSL with session to resume the existing session.

We are now in the further comparing the new methodologies for better load balancing and we will conduct further study on reviewing the more efficient algorithms for load balancing.

REFERENCES

[1]. Allen C, Dierks. The Us Protocol. [S].RFC 2246, 1999-01.
[2]. Trinitis C, Markus M W, Leberecht M. Balanced high availability in layered distributed computing systems. [C] 14th International Workshop on Database and Expert Systems Applications (DESNO3). Prague, Czech Republic: IEEE Computer Society, 2003: 713-717.
[3]. Hou Zonghao, Huang Yongxiang, Zheng Shouqi. Design and implementation of heartbeat in multi-machine environment [C] //Advanced Information Networking and Application, 17th International Conference on Advanced Information Networking and Applications. Xi'an, China: ISTP.2003:583-586.
[4]. Hatsugai, Ryosuke, Saito, Takamichi. Load-Balancing SSL Cluster Using Session Migration. [C] Advanced Information Networking and Applications, 2007. AINA '07. 21. International Conference on 2123 May 2007 Page(s):62 — 67.
[5]. Schroeder T, Goddard S, Ramamurthy B. Scalable Web server clustering technologies [J]. IEEE Network, 2000, 14(3):38-45.
[6]. Casalicchio E. and Colajanni. M, "A Client-Aware Dispatching Algorithm for Web Clusters Providing Multiple Services," Proceedings. 10th lnt'l World Wide Web Conference., May 2001.
[7]. Chita R. Das, Jin-Ha Kim, Member, IEEE, Gyu Sang Choi, Member, IEEE, Fellow, IEEE "An SSL Back- End Forwarding Scheme in Cluster-Based Web Servers" IEEE transactions on parallel and distributed systems, volume. 18, no. 7, July 2007.
[8]. Carrera. D, Guitart.J, Beltran. V, Torres.J, "Session-Based Adaptive Overload Control for Secure Dynamic Web Applications," Proceedings. Int'l Conf. Parallel Processing (ICPP '05), 2005.
[9]. Balaji P , Narravula S, Vaidyanathan K, Krishnamoorthy S, Wu J, and Panda D.K, "Sockets Direct Protocol over InfmiBand in Clusters: Is It Beneficial?" Proceedings. IEEE Int'l Symp. Performance Analysis of Systems and Software (ISPASS '04), Mar. 2004.
[10]. Bunt R., Oke. A and "Hierarchical Workload Characterization for a Busy Web Server," LNCS, volume. 2324/2002, Aug. 2003.
[11]. Choi G.S, Kim.J.-H, Ersoz.D, and Das. C.R, "Improving Response Time in Cluster-Based Web Servers through Co scheduling," Proceedings. 18th Int' I Parallel and Distributed Processing Symp. 2004.
[12]. Downey. A.B, "The Structural Cause of File Size Distributions," Proceedings. ACM Int'l Conf. Measurement and Modeling of Computer Systems (SIGMETRICS '01), 2001.
[13]. Gousios. G. and Spinellis. D, "A Comparison of Portable Dynamic Web Content Technologies for the Apache Server," Proceedings. Third Int'l System Administration and Network Eng. Conference. (SANE '02), 2002.
[14]. Keynote speech at Proc. Performance and Architecture of Web Servers Workshop, June 2000.
[15]. RFC Transport Layer Security (TLS) Protocol TLS 1.2
[16]. Ziyu Wang,Lixin Pang,YunFei Fan,"Analysis of Load Balancing of Web Cluster Based on TLS Session Sharing",2009
[17].V.M Suresh,D.Karthikeswaran,V.M Sudha,D .Murali Chandraseker,"Web Server Load Management Using Back-End Forwarding Method",2012