DOI: http://dx.doi.org/10.26483/ijarcs.v8i7.4555

ISSN No. 0976-5697

Volume 8, No. 7, July – August 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

AN SSL LOAD BALANCING TECHNIQUE IN CLUSTER WEB SERVERS

G.Srilakshmi
Research Scholar, Mother Teresa Women's University, Kodaikanal, TamilNadu, India

Dr. K. Kungumaraj
Assistant Professor,Department of Computer Applications, ArulmiguPalaniandavar Arts and Science College for Women, Palani, TamilNadu,India

Abstract: As the utilization of Internet fabricates well ordered Information security has transformed into a container neck. In spite of the way that the framework exchange speed continues growing speedier than as far as possible the server ranches/organize servers are obstructions in encouraging framework based organizations. Aggressors' usage of result item is creating and ending up being more present day. Server load adjusting gives adaptability and high openness to applications, Web districts and cloud benefits by watching the quality of servers, consistently circling loads transversely finished servers and keeping up session steadiness and a reliable customer inclusion if no less than one server get the opportunity to be overburdened or dormant. In any case, in late survey it has been watched that framework servers add to around 40 percent of the general delay, and this deferral is presumably going to create with the growing usage of component Web substance. In spite of the way that SSL is the genuine standard for transport layer security, its high overhead and poor flexibility are two significant issues in arranging secure far reaching scale sort out servers. The crucial purpose of this examination is to develop add to throughput under an extended load when customers are added and to enhance the passage of more number of records with greater in estimate. SSL-LB count is used to scatter thusly (load adjusting) the client requests over different servers giving a comparative help of the client, for instance, Internet Information Services (IIS).
For giving greater security SSL-LB load adjusting estimation is to vanquish the issues in the present system and decrease the latency time and augmentation the throughput than the present structures.

Keywords: Load balancing, Security, SSL- load balancing model, Throughput, Response time.

I. INTRODUCTION

Web server applications must have the capacity to keep running on various servers to concur an eternity developing number of clients and systems fundamental the capacity to scale execution to deal with enormous volumes of clients demands without producing irritating deferrals. Along these lines, load adjusting calculation must be acknowledged for enhanced execution and also to raise the ability to deal with more number of clients. For these thought processes, grouping is of colossal enthusiasm to the innovativeness. Clustering licenses an accumulation of autonomous servers to be prevailing as a cooler reasonability, single framework for higher accessibility, and better adaptability.

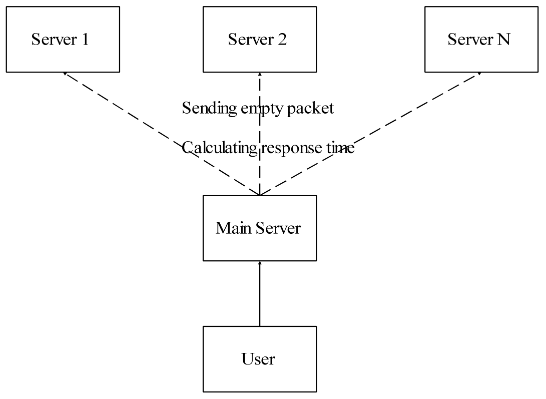

Figure 1: System Model

The Figure 1 symbolizes the complete structure model. End-user desires are directed to a load-balancing system that controls which server is greatest proficient of handling the request. Server load adjusting can likewise allot workloads to firewalls and readdress solicitations to reserving servers and intermediary servers. So as to accomplish web server adaptability, more servers fundamental to be added to dispense the heap between the gatherings of servers, this is otherwise called a server group. At the point when a few web servers are available in a server bunch, the HTTP movement needs to be similarly coursed between the servers. These servers need to execute as one web server to the web customer, for outline a web program. The heap adjusting component used for dispersing HTTP asks for is known as IP Spraying. The devices used for IP disseminating is likewise named the Network Dispatcher or Load Dispatcher or basically, the Load Balancer. In this circumstance, the IP sprayer interferes with a few HTTP ask for, and readdresses them to a server in the server gathering. Contingent upon the sort of sprayer entangled, the design can offer load adjusting, adaptability and failover prerequisites.

Expanding reputation of the Internet, server farms/arrange servers are evaluated to be the bottleneck in facilitating system based administrations, despite the fact that the system transfer speed stays to rise speedier than the server ability [1]. It has been exploratory that system servers give to almost 40 percent of the entire postponement, and this deferral is required to ascend with the developing utilization of dynamic Web fillings [1].

For Web-based applications, a modest reaction time has imperative monetary impacts [2]. For representation, E-Biz [3] expressed about $1.9 billion misfortune in salary in 1998 because of the broad reaction time resultant from the Secure Sockets Layer (SSL) [4], which is ordinarily utilized for secure correspondence in the midst of Web servers group and customers. Despite the fact that SSL is the accepted consistent for transport layer wellbeing, its high overhead and humble versatility are two fundamental damages in conspiring secure imperative system servers. Arrangement of SSL can decrease a server's ability by up to two requests of enormity [5].

In computation, the above of SSL grows even extra extreme in application servers. Application servers give dynamic substance and the substance require secure systems for assurance. Producing dynamic substance takes around 100 to 1,000 times longer than basically perusing static substance. In addition, since static substance is from time to time refreshed, it can be effectively reserved. A few proficient storing calculations [6], [7] have been proposed to lessen inertness and increment throughput of front-end Web administrations. Notwithstanding, in light of the fact that dynamic substance is created amid the execution of a program, reserving dynamic substance is not a productive choice like storing static substance.

Recently, a crowd of network systems have been considered and evaluated utilizing bunch stages [8], [9]. Precisely, the proposition of coursed Web servers has remained a primary research push to propel the reaction time and throughput [10], [11]. PRESS is the first Web server models that accomplishments client level correspondence in a bunch based Web server. Our past work in [12] moderates the reaction time in a bunch based Web server utilizing co-planning plans.

To look at the impression of SSL commitment in bunch based system servers, focusing on application servers, which for the most part offer dynamic substance. To exhibit the potential execution advancement the SSL-session recover plot is used in bunch based servers. The SSL-session reuse game plan has been checked on a solitary Web server hub in [13] and stretched out to a group framework that comprised of three Web servers in [14]. To find the SSL-session reuse structure utilizing 16-hub and 32-hub group frameworks with a few levels of workload. Apostolopoulos [14] demonstrated that session reuse is fundamental to enhance the execution of Web server's groups. To propose back-end sending component by building up the low-overhead client level correspondence to enhance the SSL-empowered system server execution.

To relate three course models in clusters: ssl_with_session, Round Robin and ssl_with_bf. Ssl_with_session utilizes a more refined course calculation in which progressive solicitations of a same user are sent to a same server, staying away from costly SSL plan costs. The Round Robin show, habitually used in Web bunches, apportions demands from customers to servers utilizing the RR conspire. The expected ssl_with_bf hones the comparative flow arrangement as the ssl_with_session, however involves a shrewd load adjusting plan that advances customer demands from a profoundly loaded back-end hub to a daintily loaded hub to recoup the use over all hubs. This strategy utilizes the basic client level correspondence for quick correspondence.

II. BACKGROUND OF THE RESEARCH

Cluster Data Centers

To portray the typical engineering of a cluster based server pool or system server comprising of three layers: frontend Web server, mid-level application server, and back-end database server. A Web server level in a server farm is a Web framework outline that includes of different server hubs interrelated through a System Area Network (SAN). The Web server offerings the customers a solitary framework sees over a front-end Web switch, which allots the solicitations between the hubs. A request from a client goes over a Web change to begin an association among the Web server and the customer. At the point when a demand accomplishes at the Web switch, the Web switch assigns the demand to one of the servers utilizing either a substance careless (Layer-4) or a substance mindful (Layer-7) conveyance. The front-end Web server offers static or straightforward dynamic administrations. The Web assets conveyed by the main level are typically open to people in general and, along these lines, don't include confirmation or information encryption. Henceforth, the normal inactivity of customer asks for in this layer is typically shorter than in the application servers. The mid-level, called the application server, is situated between the Web servers and the back-end database. The application server has a different load balancer and a security framework, for example, a firewall and ought to be furnished with a help for databases, exchange administration, correspondence, heritage information, and different functionalities [15]. Subsequent to getting a customer's demand, an application server parses and changes over it to a question. At that point, it sends the produced inquiry to a database and gets back the reaction from the database. At long last, it changes over the reaction into a HTML-based record and sends it back to the customer. The application server gives imperative functionalities to online business, for example, web based charging, keeping money, and stock administration. Along these lines, most of the substance here is produced powerfully and requires a sufficient security component. The back-end database layers houses the most secret and secure information. The principle correspondence overhead of a database layer is the continuous plate access through the Storage Area Network [16].

User-Level Communication (VIA)

A Virtual Interface (VI) is a logical channel provided to an application by the VIA framework [17]. The VI allows direct message transmission to a Network Interface Card (NIC) without needful the data to be copied to the NIC driver or using the kernel space for virtual-to-physical memory conversions. An application directs a request to the VI-enabled API, also called the Virtual Interface Provider Library (VIPL). The VIPL plans the request, generates a VI queue or work queue pair, and submits it openly to the NIC, which has VI-enabled buffers to hold the data that are sent or received. The NIC, consuming its built-in processor, achieves a restricted processing of the packet and does the virtual-to-physical-address translation, which was traditionally done by the protocol stack and the NIC driver, respectively. The process now is performed much more quickly since 1) it is done by hardware rather than by software, 2) there are no user-kernel context switch and memory copy overheads, and 3) there is minimal protocol processing overhead. In addition, the processor is free to do more productive work as it no longer has to perform the bulk of the message passing and interrupt processing. VIA has been studied extensively as a standard user-level communication to design SANs. Further description of VIA can be found in [18].

SSL

SSL [19], which operates between the HTTP and Transmission Control Protocol (TCP) network layers, is the most popular tool that provides a secure channel between a Web server and a client. Specially, furthermost Web servers/data centers associate e-commerce applications deploy SSL to provide enhanced security to Web traffic. In 1999, SSL was approved by the Internet Engineering Task Force (IETF) as a regular and is newly named Transport Layer Security (TLS) [20]. SSL is composed of two components: a handshaking procedure and a bulk data encryption procedure.

The handshaking procedure triggers when a connection is initiated between Web servers and a client [21]. During this phase, a server and a client authenticate each other and negotiate encryption algorithms and the required session keys using an asymmetric key algorithm such as RSA before they send or receive data. Since all data between a server and a client are encrypted using symmetric keys, the channel between them is private. The bulk data encryption deals two services: message digest and data encryption. Message carrying contains a message reliability checked using a keyed message authentication code (MAC). Data encryption is done with a symmetric key algorithm such as TripleDES [22] or RC4. [23].A secure hash function such as the Secure Hash Algorithm Version 1.0 (SHA1) [24] or Message Digest 5 (MD5) [25] is used for MAC computations.

An exhaustive handshaking process essential to start a new session on web cluster. A client initiates a connection with a server by sending a Client Hello message that includes the session ID, a random number, cipher suites, and other required information. After receiving the Client Hello, the server sends a Server Hello including its certificate and other information as a reply. With the certification of the server, the client finishes the authentication of the server. Depending on the server side configuration, the next procedure for the client authentication is optional. If it is requested, the client needs to send its certificate to the server for verification. After the authenticating procedures, the client generates session keys for the encryption and decryption of data. The session is identified by the session ID that is shared between the clients and server.

To remunerate the great overhead of the handshaking protocol, a session can be reused when re establishing the connection. The SSL protocol allows a server to configure the session time. During the session time, a server caches the session information of clients. Whenever a client requests a new connection within its session time, the server reuses the cached session state to generate a set of keys for the new session and saves the computation time required for authentication and negotiation procedures.

Cryptographic Algorithms

The cost of cryptographic algorithms used in SSL. To measure the execution time of the algorithms in a 500-MHz Ultra-Sparc uni processor, running on Solaris 2.9 with a 1-Gbyte memory. The algorithms we measured are RSA, RC4, and MD5, which are the most widely used cipher suites provided by a Web server and a Web browser. RSA is restrained using SSL’s RSA_private_enc(), RSA_ public_dec() functions, and RC4 and MD5 are measured using RC4() and MD5() functions, respectively [26]. Since the RSA algorithm is optimized for the public key operation, the operation time for the public key is relatively fast compared to the private key operation. In the SSL protocol, a server needs to compute a private key to establish a secure channel with a client. Thus, to negotiate the session key between a server and a client, the client uses the server’s public key to encrypt a session key, and the server has to decrypt it using its own private key. This process increases the server-side overhead. The key size of RSA impacts the level of the security. The 2,048-bit RSA key can offer a greatly stronger security level than a 1,024-bit RSA key, whereas the previous needs more than six times the computation cost. The computation costs for MD5, RC4 are the message digesting and encryption times for the 1-Kbyte data block with a 128-bit key. These costs are relatively small compared to the cost for an RSA algorithm.

III. PROBLEM DEFINITION

Difficult systems make growing loads on web servers. Multiple Objects can delay, and high capacities can overcome Systems. Fixes essential to be recognized early in this research, and Clients have scalability disquiets, and must warrantee some level of scalability with industry accepted metrics. The basic routine tests for both the browser and server sides of the calculation and recommends on an overall approach for recognizing and attacking performance bottlenecks. Recognizing load of the servers is more difficult process. The identification of load refers to the repetition of modeling the predictable usage of a software program by simulating multiple users retrieving the program simultaneously. As such load proof of identity is greatest relevant for multi-user systems; often one built using a client/server model, such as web servers. Still, other kinds of software systems can also be utilized for load testing.

There are few simple symptoms shows network server load. If the server load exceeds its limit then the application will automatically get slow down and response from the server will be very low.

Specific physical Origin or Proxy Server will most likely be unable to deal with its heap. For Web-based applications, a modest reaction time has fundamental money related inductions because of the broad reaction time coming about because of the Secure Sockets Layer (SSL), which is generally utilized for secure correspondence amongst customers and Web servers. Though the SSL are the accepted steady for transportation level wellbeing, its extraordinary overhead and poor versatility are two noteworthy issues in planning secure expansive scale organize servers. Arrangement of SSL can diminish a server's ability by up to two requests of greatness. In computation, the overhead of SSL grows considerably starker in application servers. Application servers give Dynamic substance and the substance requires secure components for insurance. To make all the requests more secured with the advancement of SSL algorithm in form of generating random session keys. This development of security can enhance the progress of E-Commerce sites and other busy and highly loaded servers.

IV.OBJECTIVES OF PROPOSED WORK

According to what is expected from the development of this research, concrete objectives are the following:

- ✓ Review the typical of the proxy servers and web servers.
- ✓ Review the various aspects of decreasing the web server loads.
- ✓ Propose a new concept called ESSL_LB technique.
- ✓ Analysis, design and find the new solution.
- ✓ To study the impact of SSL offering distribution in cluster network servers.
- ✓ To study the ESSL-session-aware circulation in cluster network servers.

V. PROPOSED WORK METHODOLOGY

The server, which receives the request from another node, generates and encrypts the dynamic content using the forwarded session key. Finally, it returns the reply to the initial node, which sends the response back to the client. To agree that all the intra transportations in a cluster are secure consequently these nodes are connected through the user-level communication and are located very closely.

```
If CR$_1$ S then
Check load of S.
If LS exceeds then
Calculate R and then Send Packet to  $S_1, S_2, S_3 \ldots S_n$
CR$_1 \rightarrow$ Si.
```

The requests arriving at the Web switch of the system server are sent to either the Web server layer or the application server layer as per the asked for benefit by the customer. Since the SSL association is served by an alternate kind of HTTP server(Hypertext Transfer Protocol Secure (HTTPS)) and an alternate port number, the solicitations for the SSL association are passed on to the wholesaler in the application server layer. To extraordinarily fixation on the routine of the application server, it neglects the idleness between the Web switch and the wholesaler and coherently speaks to them as extraordinary unit. At the point when a demand touches base at the wholesaler, it looks through its query table to decide if there is a server that has the session data of the customer and after that advances the demand to the server. Else, it grabs another server to forward the demand. The sent server builds up another SSL association with the customer. In the event that the demand is sent to a profoundly stacked server, the server thusly sends the demand with the session data to a gently stacked server.

The server recognizes the accessible server by sending an unfilled bundle. In the event that the sub server is free then it will responds in a flash, finished the reaction time it doles out the customers to the intermediary servers. End-client demands are coordinated to a heap adjusting framework that directs which server is furthermost fit for preparing the demand. It already advances the call to that server. Server stack adjusting can likewise distribute workloads to firewalls and divert solicitations to intermediary servers and storing servers.

The below Figure-2 represents the architecture of the proposed mechanism. The requests from several clients are collected in the server side; the server weight will be considered using the ESSL_LB structure if the server load overdoes the sub servers' details can be collected. After that the server sends as empty packet to all the sub servers, depends on the response time the client request will be navigated to the particular sub server. The client navigation will be considered.

ESSL-LB (Enhanced Secure Socket Layer-Load Balancing) Algorithm:

Step 1: Signifying the server and its sub servers by essential the IP address and Host name. This helps to distribute and create the network structure with main server and proxy server before proceeding.

Step 2: Save those Network construction with relevant details. This can be done with proper authentication.

Step 3: Select the files to be shared with the proxy servers.

Step 4: Encrypt the files with the help of private and public keys. This can be done with the help of RSA algorithm.

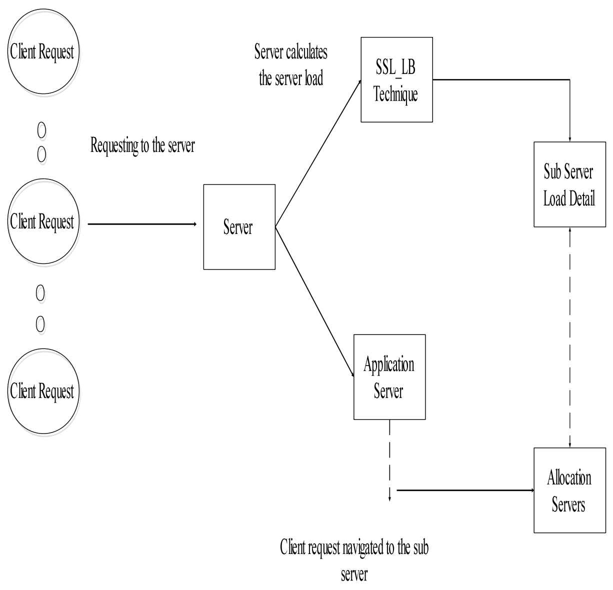

Figure 2: SSL_Load Balancing Architecture

Step 5: Save those encrypted files on the sub servers. These files will be routinely stored on the proxy's, the proxy servers could be recognized by the network construction module, which stores the IP addresses.

Step 6: The uploaded files are sharable and the client can download those files which they needed.

Step 7: The next step is evaluating the server load. When client requests the server, server calculates the load by the number of open connections. The request of the clients will be categorized into 2 types such as dynamic and static requests.

Step 8: The server distributes an empty packet to all sub servers and gathers the response. The response time will be calculated through queuing method.

Step 9: The response time will be calculated and compared by using the total number of requests and download time of the sub servers.

Step 10: The user request is redirected to the sub server and the user can download the files and decrypt using the private key.

The server load will be calculated by determining some elements such as number of connections made in the network, proxy server allocation. The approximate server weight consist with the number of open connections, here the concept cluster has been proposed to combine all requesting clients for a particular server at a time.

VI. PERFORMANCE IMPACT WITH SSL

To study the impact of the SSL algorithms on server performance, we measured the throughput and latency using trace files in a single Web server environment. The Web server for this experiment is Apache v2.0 with mod_ssl_cluster, deals robust web cluster cryptography for the Transport Layer Security (TLS) (TLS v1) protocols and Apache Web server through the SSL (SSL v2/v3) [27]. The trace files used for the experiment are the Clark net and NASA trace files. The trace files have about 3,000,000 client desires for the Web resource. The normal file size of NASA is about 28 Kbytes, while the normal file size of Clark net is about 15 Kbytes. To used 10 workstations as client nodes. One client process runs on each workstation. A client process sends a new request as soon as it receives the reply for the previous request. The server node is equipped with 1-GHz UltraSparcllli dual processors running on Solaris 2.9 with a 4-Gbyte memory system.

To depicts the excessive overhead due to SSL offering. Subsequently the normal file size of the NASA trace file is greater than the Clarknet, it shows a poorer latency and throughput than the Clarknet trace without SSL. With SSL, the potentials of the Clarknet and NASA are corrupted about 76 times and 38 times, correspondingly. The NASA trace file shows less poverty because it handovers more bytes per RSA authentication. The throughput results are similar to the latency results.

VII. CONCLUSION

To adopt that 80 percent of the requests are active and are routinely focussed to the application servers. The remaining 20 percent are serviced by the Web servers. It will be more adaptable to the 99 percent dynamic requests in the future and effective load distribution will be made. To carry out for pay load decline of the web server when the server being busy and examined the performance implications of the SSL protocol for providing a protected service in a cluster web application server. A vital aim of this research is to motivate this situation as a real time project, analysis and evaluate results. Using three proxy servers, proposed a back-end forwarding scheme for distributing load to the lightly loaded server and improving server performance to obtain the better results. In this system examined different aspects of using IP any cast as a mechanism for load distribution and service location in the Internet.

VII. REFERENCES

[1] E. Cecchet, J. Marguerite, and W. Zwaenepoel, “Performance and Scalability of EJB Applications,” Proc. 17th ACM SIGPLAN Conf. Object-Oriented Programming, Systems, Languages, and Applications,pp. 246-261, 2002.

[2] E.V. Carrera, S. Rao, L. Iftode, and R. Bianchini, “User-Level Communication in Cluster-Based Servers,” Proc. Eighth Int’l Symp. High-Performance Computer Architecture (HPCA ’02), pp. 248-259, 2002.

[3] Network Working Group, The MD5 Message-Digest Algorithm, IETF RFC 1321, http://www.ietf.org/rfc/rfc1321.txt, 1992.

[4] M. Mitzenmacher, “Dynamic Models for File Sizes and Double Pareto Distributions,” Internet Math., 2004.

[5] S. Abbott, “On the Performance of SSL and an Evolution to Cryptographic Coprocessors,” Proc. RSA Conf., Jan. 1997.

[6] Q. Li and B. Moon, “Distributed Cooperative Apache Web Server,” Proc. 10th Int’l World Wide Web Conf., pp. 1215-1229, 2001.

[7] C. Allen and T. Dierks, The TLS Protocol Version 1.0, IETF Internet draft, work in progress, Nov. 1997.

[8] C. Cunha, A. Bestavros, and M. Crovella, “Characteristics of WWW Client-Based Traces,” Technical Report BU-CS-95-010,Boston Univ., 1995.

[9] M. Crovella and P. Barford, “Self-Similarity in the World Wide Web Traffic-Evidence and Possible Cause,” Proc. ACM Int’l Conf.Measurement and Modeling of Computer Systems (SIGMETRICS ’96),pp. 160-169, 1996.

[10] “SSLeay Description and Source,” http://www2.psy.uq.edu.au/ftp/Crypto/, 2007.

[11] Compaq Computer Corp., Intel Corp., and Microsoft Corp.,Virtual Interface Architecture Specification. Version 1.0, http://www.vidf.org, Dec. 1997.

[12] M. Colajanni, P.S. Yu, and D.M. Dias, “Analysis of Task Assignment Policies in Scalable Distributed Web-Server Systems,” IEEETrans. Parallel and Distributed Systems, vol. 9, no. 6, pp. 585-600, June 1998.

[13] G. Apostolopoulos, V. Peris, and D. Saha, “Transport Layer Security: How Much Does It Really Cost?” Proc. INFOCOM, 1999.

[14] G. Apostolopoulos, D. Aubespin, V. Peris, P. Pradhan, and D. Saha, “Design, Implementation and Performance of a Content- Based Switch,” Proc.INFOCOM, 2000.

[15] Network Working Group, The Use of HMAC-SHA-1-96 within ESP and AH, IETF RFC 2404, http://www.ietf.org/rfc/rfc2404.txt,1998.

[16] Network Working Group, Triple-DES and RC2 Key Wrapping, IETFRFC 3217, http://www.ietf.org/rfc/rfc3217.txt, 2001.

[17] M.Yousif, Characterizing Datacenter Applications, www.intel.com/idg,Feb.2003..Zhou,

[18] A.Bilas,S.Jagannathan, C. Dubnicki, J.F.Philbin and K.Li , “Experiences with VI Communication for Data Storage,” Proc. 29th Ann.Int’l Symp. Computer Architecture,pp. 257-268, May 2002.

[19] T.Wilson, E-Biz Bucks Lost Under SSL Strain,http://www.internetwk.com/lead/lead0520.htm,May 1999.

[20] Transport Layer Security Working Group, The SSL Protocol Version3.0, Internet draft, work in progress, Mar 1996.

[21] R. Rivest, the RC4 Encryption Algorithm, RSA Data Security, Inc., Mar. 1992.

[22] J.E. Pitkow, "Summary of WWW Characterizations," Proc. Seventh Int'l Conf. World Wide Web, pp. 551-558, 1998.

[23] V.S. Pai, M. Aron, G. Banga, M. Svendsen, P. Druschel, W.Zwaenepoel, and E. Nahum, "Locality-Aware Request Distribution in Cluster-Based Network Servers," Proc. Eighth Int'l Conf.Architectural Support for Programming Languages and Operating Systems, pp. 205-216, 1998.

[24] J. Guitart, D. Carrera, V. Beltran, J. Torres, and E. Ayuade," Session-Based Adaptive Overload Control for Secure Dynamic Web Applications," Proc. Int'l Conf. Parallel Processing (ICPP '05),2005.

[25] G. Gousios and D. Spinellis, "A Comparison of Portable Dynamic Web Content Technologies for the Apache Server," Proc. Third Int'l System Administration and Network Eng. Conf. (SANE '02), pp. 103-119, 2002.

[26] "SSLeay Description and Source," http://www2.psy.uq.edu.au/ftp/Crypto/,2007.

[27] C. Allen and T. Dierks, The TLS Protocol Version 1.0, IETF Internet draft, work in progress, Nov. 1997.