Web Server Optimization: A DNS Based Approach

Ankur Chauhan
Information Technology Department
Medicaps Institute of Technology and Management
Indore, India
ankur.it007@gmail.com

Dheeraj Rane
Information Technology Department
Medicaps Institute of Technology and Management
Indore, India
ranedheeraj@gmail.com

***Abstract*—A group of web-server system can be deployed to support high request rate. The Domain Name System (DNS) servers act as a mediator between the client and web-servers. It dispatches the client request among the web-servers through the URL-name to IP-address mapping mechanism. In this paper, we propose a group of web-server system arranged in the multiple logical ring connection, in which the DNS evaluate the performance of the proposed web-server system by using adaptive load balancing algorithm. We show the performance measurements for the proposed system.**

***Keywords*—Load Balancing, Distributed Web-server system, DNS-based dispatcher, Web-server cluster.**

I. INTRODUCTION

A Web-Server system with one virtual URL-Name and multiple IPs is one of the possible approaches to handle ever increasing client requests to popular web sites. This system maintains a single interface to the users and has the potential to provide better Load Balancing [2]. In the translation process from the symbolic name (URL) to IP address, DNS can select any node of the Web-Server system. In particular, this translation process allows the DNS to implement various RR Set scheduling policies at Authoritative Name Server to select the appropriate Web-Server node [3, 4]. In this paper, we will focus on an approach that integrates the DNS-based dispatching mechanism with a redirection technique based on the load average (LA) information of the Web-Server system [5]. The local Name Server that is Authoritative for this domain collects the LA information from various Web-Servers node periodically and Sort the Zone records with increasing LA. By controlling RR Set Ordering at DNS, client will get reference of Web-Server node that is having least value of LA. Preserving the used hardware architecture and adding only a new Web-Server node with different IP, it will dynamically registered in the Zone records at DNS, no manual modification in Zone record is required. Firewall is used at DNS to prevent any unauthorized modification in Zone records. These techniques provide a scalable Web-Server system. In case of fault of any of the Web-Server node, it will be removed from the Zone records dynamically and no client will get reference of this Web-Server node thus making system fault tolerant. This paper describes the performance of the proposed Web-Server system. We experiment the system using the request generator, JMeter.

II. DNS-BASED LOAD BALANCING

In this section, we describe the implementation issues related to the proposed load balancing with dynamic Zone records in Authoritative Name Server. Many algorithms use some state information, such as the number of active connections on the Web-Servers, or the utilization of the Web-Server resources. In this paper, we consider the Load Average value [7] of various web-server nodes over a short interval of time. In order to collect the server load average value, we arrange the Web-server nodes in star connection as shown in Figure 2. Linux toolkit (Top, uptime) is used to collect LA value of various web-server nodes [5].In this figure Apache2.0 is configured at various web-server nodes. BIND 9 is used as DNS that is Authoritative for Experimental Domain. Zone file segments for experimental domain is

/var/named/example.com.zone

```
$ORIGIN example.com

        $TTL 86400@  IN SOA dns1.example.com.
                  hostmaster.example.com.
     2001062501 ; serial
     21600      ; refresh after 6 hours
     3600       ; retry after 1 hour
     604800     ; expire after 1 week
     86400 )    ; minimum TTL of 1 day

     IN    NS    example.com.
    IN    A     192.168.1.2
    example.com  IN    A     192.168.1.2
    example.com  IN    A     192.168.1.3
   example.com  IN    A     192.168.1.4
  example.com  IN    A     192.168.1.5
```

BIND versions after 9.2.3 fully implement the RRset-order statement, which can be used to control the order in which equal RRs, an RRset of *any* type are returned. The RRset-order statement can take a number of arguments, which may take the values fixed—the order the records were defined in the zone file; cyclic—start with the order defined in the zone file and round-robin for each subsequent query; and random—randomly order the responses for every query [6].

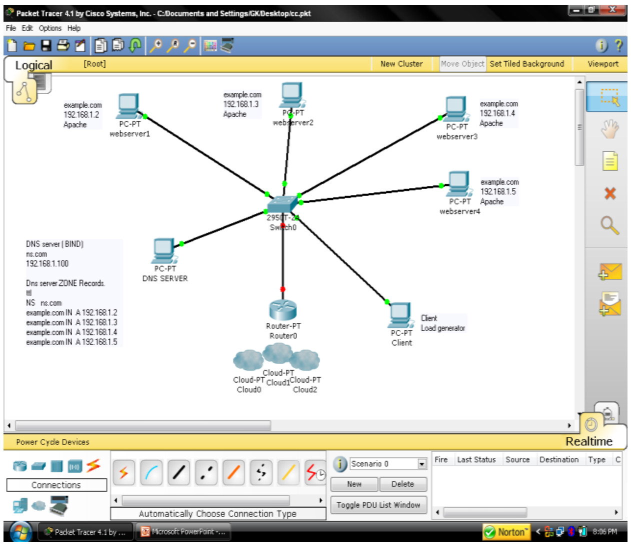

Figure 2 Experimental setup

In experimental setup, ***/etc/named.conf*** fragment follows, that will return any RRset (a set of equal RRs) in fixed order from the top of the Zone Records:

// named.conf fragment
options {
// other options
rrset-order {order fixed;};
};

In Figure 2, we show the main software components needed to implement the proposed Web-Server system. The software components include the Web-Server node Load Average generator and monitor, the Load Average collector at DNS from various Web-Servers nodes. Zone records re-shuffler at DNS on the basis of LA value of various Web-Server nodes. Firewall at Name Server to prevent any unauthorized modification in Zone records. HTTP request generator at web client.

After a fixed interval of time, Zone records are dynamically rewritten. A Web-Server node with least value of LA will come at top of Zone records.. Because of *fixed rrset ord*er, redirection module always response from top of the Zone records. So web client will always get reference of Web-Server node that is having least value of LA. By this algorithms load is uniquely distributed between various Web-Server nodes. If any of the Web-Server node is down, it will come at bottom of the Zone records and no client will get reference of this Web-Server node, it will provide fault tolerance in web-server system. Any existing web-server node can be removed and new can be added without any manually up gradation to Zone records. Zone records re-shuffler will listen response from any newly added Web-Server node and place it in the RR set, as per its value of LA. To prevent unauthorized modification at Zone records, a firewall with IPTABLES is configured at DNS. These techniques provide Network scalability. LINUX toolkit utility "XLOAD" is used to monitor the value of l Web-server nodes.

III. PERFORMANCE MEASUREMENTS

In this section, we describe the measured performance of the proposed Web-Server system. The tests described below were run on six PCs. All six machines connected via Ethernet were running Linux 2.4.2.Four of them as Web-Server node with Apache 2.0.Web-Server. One of them acts as the DNS server with BIND 9.0 and this DNS is Authoritative for experimental domain "example.com" and one as web client. The client HTTP requests were generated using the Apache JMeter. Apache JMeter may be used to test performance both on static and dynamic resources (files, Servlets, Perl scripts, Java Objects, Data Bases and Queries, FTP Servers and more)[4]. It can be used to simulate a heavy load on a server, network or object to test its strength or to analyze overall performance under different load types. One can use it to make a graphical analysis of performance or to test our server/script/object behavior under heavy concurrent load. Each test covered a period of 90 minutes, generating approximately 4500 HTTP requests and all the replies were HTML Web pages of the same size. We compared the LA value for the various Web-Server nodes based on dynamically varying Zone records in Authoritative Name Server with fixed RR set ordering vs the fixed Zone records and Round Robin scheduling policy in Authoritative Name Server. Figure 3 shows the LA value of each Web-server node in the Web-Server system using the fixed Zone records and Round-Robin scheduling policy at DNS.

Figure 4. Shows the LA values of Web-Server nodes with dynamically varying Zone records and Fixed RR set ordering on DNS. The proposed Web-server system based on varying Zone records and fixed RR set ordering at DNS-can achieve better performance than the system based on the DNS Round-Robin scheduling policy. This result is likely due to the Load Average information collected from each Web-server node by the authoritative name server.

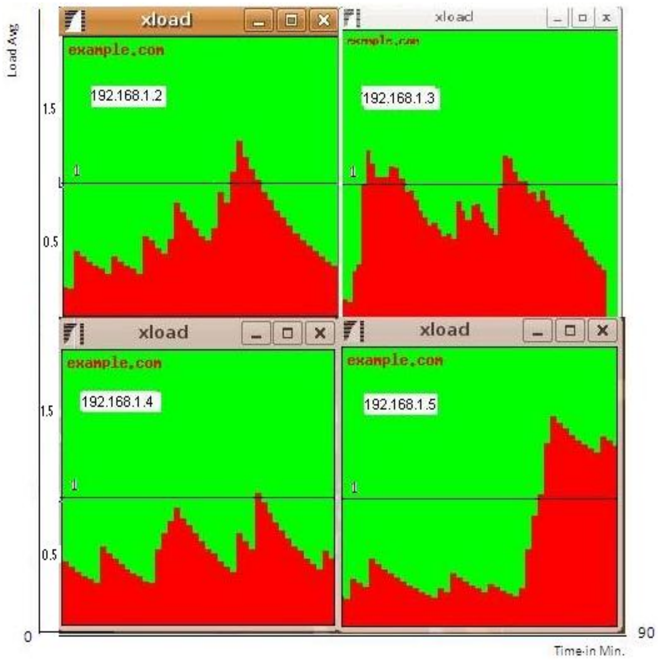

Figure 3 LA values of Web-servers nodes based on fixed Zone records and Round-Robin RRset policy at DNS.

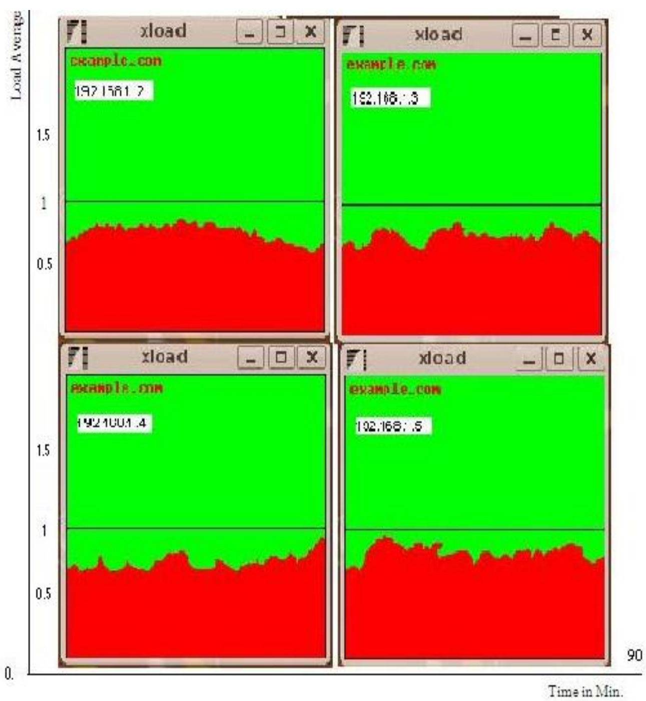

Figure 4 LA values of Web-server nodes with dynamically varying zone records and Fixed RR set ordering on DNS.

IV. CONCLUSION

We have measured the performance of Web-server system, where Authoritative Name Server distribute client request between various Web-server nodes and thereby provide Load Balancing. No client request will be forwarded to a Web-Server node that is down and new web-server node will automatically registered in Zone records thereby providing fault tolerant and scalable Web-Server system. We used the open source request generator, JMeter in order to generate client requests. The experimental results show that the proposed scheme achieves better performance and provide scalable and fault tolerant system than the default Load Balancing scheme based on the Round-Robin policy. In the future, these techniques can be club with TTL so that cashing of records at intermediate Name Server can take advantage of this proposed scheme.

REFERENCES

[1] Y. S. Hong, J. H. No and S.Y. Kim, "DNS-Based Load Balancing in Distributed Web-server Systems" Fourth IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems and Second International Workshop on Collaborative Computing, Integration, and Assurance (SEUS-WCCIA'06)

[2] M. Colajanni, P. S. Yu, and D. M. Dias, "Analysis of task assignment policies in scalable distributed Web-server Systems", *IEEE transactions on Parallel and Distributed Systems*, Vol. 9, No. 6, pp. 585-600, (June 1998).

[3] Y.S. Hong, J.H. No, and In Han "Evaluation of fault-tolerant distributed Web systems", Proc. WORDS 2005 (IEEE CS 2005 Workshop on Object-Oriented Real-Time Dependable Systems), Sedona, U.S.A. (Feb. 2005)

[4] http:// jakarta.apache.org

[5] http://linux.com

[6] Pro DNS and BIND: Ron Aitcheson

[7] UNIX Load Average: How It Works by Dr. Neil Gunther.

[8] RFC 1794 (DNS support for Load Balancing).

[9] Java computing, http://sun.com/java

[10] Apache's site http:// apache.org

[11] RFC 1032 Domain Administrators Guide.

[12] Wikipedia DNS- Wikipedia, the free encyclopedia.