A Study on Traffic Distribution Processing using the Round Robin Monitoring System

Seok-Soo Kim, *Member, KIMICS*

Abstract – As the result of the study shows, Round Robin Algorithm allows servers to ensure definite traffic distribution, unless incoming data loads differ much. Although error levels are high in some cases, they were eventually alleviated by repeated tests for a long period of time. The study uses two software packages (Packet Capture and Round Robin Test Package) to check packet volume from Virtual Network Structure (data generator, virtual server, Server 1, 2, 3), and finds out traffic distribution toward Server 1, 2, and 3. The functions of implemented Round Robin Load Balancing Monitoring System include round robin testing, system monitoring, and graphical indication of data transmission and packet volume.

Index Terms – Round Robin, Monitor, Traffic, Network, Distribution,

I. INTRODUCTION

As e-commerce transactions become more popular, demand for data service is drastically increasing via the Web. With adding varied features to the network, which carry special purposes, workloads of routers and switches comprising the network are gradually increasing. Given the fact, without considering transmission speed, the transmission performance of the overall network will naturally deteriorate.

As the interest and demand in e-commerce using web server explode, the number of network that accommodates this web server cluster architecture is increasing. Therefore, establishing a web server cluster network environment that guarantees scalability becomes an important technical issue [4, 5, 6]. Namely, when an exterior client sees a web server cluster, it is important for dispatcher to implement server management functions toward the back-end servers so that the web server cluster is seen as a single server. At this time, the dispatcher should perform an efficient session distribution policy so that back-end servers can have balanced loads among them.

[1,2] Meeting the tendency of high-speed network, the technology of establishing and managing high-speed TCP connections based on the technology of managing the existing multiple back-end servers in the web server cluster environment are gaining its impotence.

For this purpose, TCP Splicing has emerged as a higher speed TCP connection technique. Utilizing this technique, such technique that efficiently creates and manages sessions that are fast and have special purposes in the web sever cluster environment, will be the core technology that strengthens the current e-commerce service.[3]

Namely, TCP Splicing is not a technique to apply to TCP/IP stack in a single system such as client and server, but a technique to connect TCP at the proxy (belongs to L4 switch) that connects client with server. In the existing proxy server, the proxy processes the headers of incoming packets as they are received from client and uploads up to application layer process that resides in the main processor in order to create connection between client and server.[4,5]

Because of this reason, this study selects web server cluster as a target for transmission performance optimization. The issue of study focuses on the implementation of high-speed data transmission between client and server in such an environment of the web server cluster.

II. ROUND ROBIN MONITORING SYSTEM

A. Overview of System Architecture

(1) It analyzes header files of random data sent through data generator and shows the results on the generator's monitor.
(2) It sends the analyzed files to the virtual server and distributes across the Web servers using Round Robin algorithm.
(3) It reanalyzes the distributed files and displays the results on the monitor of each server.

B. System Architecture (Linux Cluster Implementation)

The Linux Virtual Server [6] is a highly scalable and highly available server built on a cluster of real servers, with the load balancer running on the Linux operating system [7, 8].

Shown as the figure below, it depicts general virtual server architecture with one load balancer and several real servers. The real servers are configured to operate the same services and the service contents are copied into a local disk at each server or provide service by sharing a distributed file system. The load balancer sends client's requests to one real server that is interlinked by adequate scheduling technique (i.e. Round Robin), whenever a client requests.

Manuscript received April 3, 2005.

Seok-Soo Kim is with Department of Multimedia, Hannam University, Daejeon, Korea (e-mail : sskim@ hannam.ac.kr)

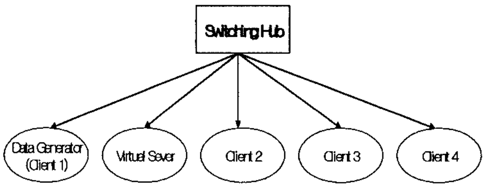


Fig. 1 Virtual connection architecture, Physical connection architecture

C. Scenario

(1) Build a virtual server to analyze incoming packet information from data generator.
(2) Develop software to display analyzed packet information on window screen.
(3) As a final experiment, analyze such packets inputted through data generator as total packet volume, TCP, IP, ARP, and other protocols on the virtual server and find out whether the results can be displayed on the data input computer.
(4) In addition, check what values return when the virtual server sends the analyzed data to server 1, 2, and 3.

D. Limitation

(1) Linux operating system is required to test on the real server environment.
(2) A switching hub is required to combine computers into one network environment for Round Robin test.
(3) Data implementation is required to analyze header files of packet data and display the results

III. SYSTEM IMPLEMENTATION AND FUNCTION MODULES

This study builds the following two packages to implement round robin load balancing technique. They are named Packet Capture Package and Round Robin Test Package.

A. Packet Capture Package

It includes two modules: packet capture module and graphic module. The packet capture module evaluates the volume of packets and displays text data by protocols. The graphic module creates a window box and saves data handed over from the packet capture module, showing the results in graphs.

B. Round Robin Test Package

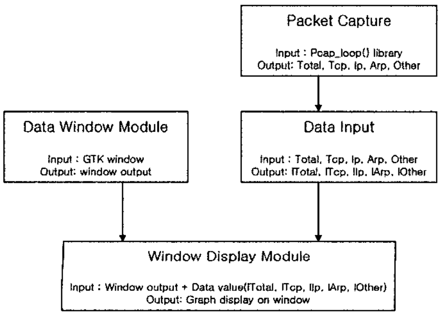
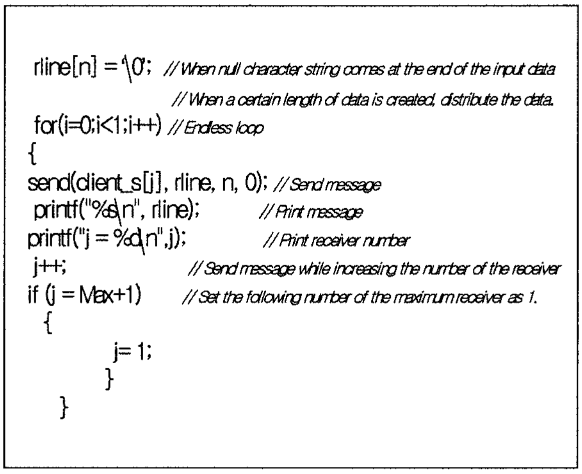

Packet Capture
Input : Pcap_loop() library
Output: Total, Tcp, Ip, Arp, Other
Data Window Module
Input : GTK window
Output: window output
Data Input
Input : Total, Tcp, Ip, Arp, Other
Output: ITotal, ITcp, IIp, IArp, IOther
Window Display Module
Input : Window output + Data value(ITotal, ITcp, IIp, IArp, IOther)
Output: Graph display on window

Fig. 2 Packet Capture Package

It includes three modules : data generator module, virtual server module, and data receiver module. The data generator module creates data of random value and transfers them to the virtual server. The virtual server module distributes the data received from the data generator module using round robin algorithm and transfers them to the data receiver. The data receiver module saves incoming data from the virtual server module and analyzes them..



```
rline[n] = '\0';  //When null character string comes at the end of the input data
                   //When a certain length of data is created, distribute the data.
for(i=0;i<1;i++) //Endless loop
{
send(client_s[j], rline, n, 0); //Send message
 printf("%s\n", rline);          //Print message
printf("j = %d\n",j);             //Print receiver number
 j++;                           //Send message while increasing the number of the receiver
if (j = Max+1)                  //Set the following number of the maximum receiver as 1.
  {
           j= 1;
          }
    }
```

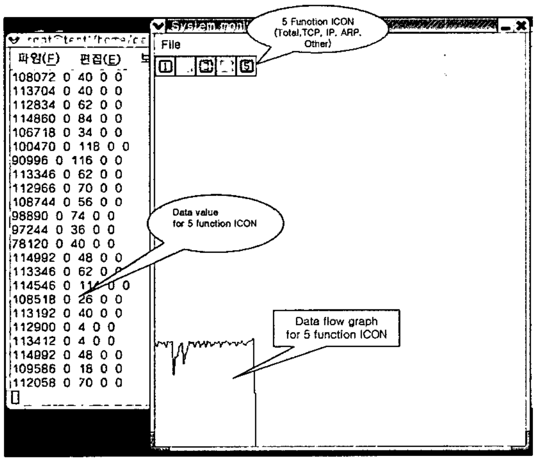
Fig. 3 Round Robin Load Balancing Implemented Algorithm

IV. SYSTEM INTERFACE

1) Receiving data volume
   Save the receiving data as a file to check the size of the file by timeline.
2) Packet volume display
   As a numerical display, use to check the incoming data whether it comes well or not by renewing data value every second.
3) Packet volume graphs display
   As a final interface, display various packet types and packet volume graphs. It helps to see the relations of data generator, server, and data receiver and allows to evaluate round robin algorithm.
4) Internal contents architecture
   - Data architecture: variable length array
   - Socket communication: character string
   - Packet capture: packet structure (Display the volume of each protocol by analyzing header), Evaluate the incoming volume for a second and process in realtime
   - Graphs display: Save all the current and past packet volumes and display realtime graphs. It draws the graphs in a second interval.
5) Basic function
   - Round robin test: Display the distribution process of network traffic loads by round robin algorithm
   - System monitoring: It means packet monitoring. It is done by a software program that evaluates the packet volume coming from the five protocols: Total, TCP, IP, ARP, and Other. The program has five buttons and a menu.

6) Window/Menu/Icon

- Window: In the Linux environment, an execution is mainly carried on the text mode. To display graphs, Linux using GTK is used. The program has a menu and five buttons. With 100 pixels applied, it is able to display 100 steps horizontally and vertically.
  The real volume of packet traffic may vary greatly from 10 to 1 million. When 100 users are simultaneously connected, the packet volume normally shows 300,000. Due to large fluctuation, it is difficult to accurately display in graphs. It only shows a comparison in the same context. The software is enabled only to show a comparison. If further functions are necessary, it can be expanded with different color display to give more accurate display for evaluation.
- Menu: It has a Close menu under File that closes the window.
- Icon: It has five buttons by different colors and shows five protocols.



```
108072 0 40 0 0
113704 0 40 0 0
112834 0 62 0 0
114860 0 84 0 0
106718 0 34 0 0
100470 0 118 0 0
90996 0 116 0 0
113346 0 62 0 0
112966 0 70 0 0
108744 0 56 0 0
98890 0 74 0 0
97244 0 36 0 0
78120 0 40 0 0
114992 0 48 0 0
113346 0 62 0 0
114546 0 114 0 0
108518 0 26 0 0
113192 0 40 0 0
112900 0 4 0 0
113412 0 4 0 0
114992 0 48 0 0
109586 0 18 0 0
112058 0 70 0 0
```

파일(F)
편집(E)
File
5 Function ICON (Total,TCP, IP, ARP, Other)
Data value for 5 function ICON
Data flow graph for 5 function ICON

Fig. 4 Execution screen (Window, Icon, Menu)

7) Advanced features: Display the volume of five kinds of protocols (Total, TCP, IP, ARP, and Other) to see the use of protocols.

- This shows values in the protocols of Total, TCP, IP, ARP, and Other as packets on the real network.
- This graph shows the trends of data transmission. In this window, it only shows the volume of Total in red.

V. CONCLUSION

In order to handle client's requests while multiple servers seamlessly work in the web server cluster environment, it is vital to implement a router that execute a routing using TCP information and requested target content.

This study analyzes a load balancing technique using Round Robin Algorithm. For this technique, the study used two software packages (Packet Capture Package and Round Robin Test Package). Packet capture module measures packet volume as it outputs text data by protocol, while graphic module creates a window, saves data received from packet capture module, and displays them into visual graphs. Round robin test package includes three modules : data generator module, virtual server module, and data receiver module. Data generator module creates data at random value, which transmits them to virtual server module. Virtual server module distributes the data received from the data generator by round robin algorithm and sends them to data receiver. Data receiver module saves incoming data from virtual server module as files and analyzes them.

ACKNOWLEDGMENT

This work was supported by a grant No. (R12-2003-004-03003-0) from Ministry of Commerce, Industry and Energy.

REFERENCES

[1] P.Gupta, S.Lin, and N.Mckeown, "Routing lookups in hard ware at Memory Access Speeds," Proceedings of the Conference on Computer Communications (IEEE INFO-COM), pp. 1241-1248, 1998.

[2] E. Levy-Abegnoli, A. Iyengar, J. Song, and D. Dias, "Design and Performance of a Web Server Accelerator," IEEE INFOCOM'99, pp. 135-143,1999.

[3] T. Schroeder, S.Goddard, and B. Ramamurthy, "Scalable Web Server Clustering Technologies," IEEE Network, pp. 38-45, 2000.

[4] D.A. Maltz. and P, Bhagwat, TCP Splicing for Application Layer Proxy Performance, Jurnal of High Speed Networks, pp. 225-240, 1999.

[5] A, Cohen, S.Rangarajan, and H, Slye, On the Performance of TCP Splicing for URL aware Redirection, Proceedings of the 2nd USENIX symposium on Internet technologies and Systems, 1999.

[6] W. Zhang and et al., Linux virtual server project, 1998.

[7] Darwin Streaming Server, http://www. publicsource. apple.com/projects/streaming

[8] W. Zhang, "Linux Virtual Server for Scalable Network Services."

Seok-Soo Kim
Received a B.S. degree in computer engineering from Kyungnam University 1989, and M.S. degree in Information engineering from Sung kyun-kwan University 1991 and Ph D. degree in Information engineering from Sung kyunk-wan University 2002.
In 2003 he joined the faculty of Hannam University where he is currently a professor in Department of Computer & Multimedia Engineering. His research interests include Multimedia Communication systems, Distance learning, Multimedia Authoring, Telemedicine, Multimedia Programming, Computer Networking. Information Security. He is a Member of KICS, KIMICS, KIPS, KMS, and DCS.