一個提供可變尺度及負載平衡的伺服機代理機
A Scalable Server Broker for Scaling and Load Balancing Services

蔡亮宙，蔡尚榮，盧彥如
Lian-Jou Tsai, Shang-Rong Tsai, Antony Y.R. Lu

國立成功大學電機研究所
Department of Electrical Engineering
National Cheng Kung University, Tainan Taiwan R.O.C.

摘要

本文敘述一個可變尺度伺服機代理者如何集合多個伺服機的能力,以單一伺服機的假象對外提供服務.
關鍵字:網際網路, WWW, TCP/IP, 可變尺度伺服機.

Abstract

This paper describes how the Scalable Server Broker (SSB) integrates multiple servers into a server group and presents a single server image to users.
Keywords: Internet, WWW, TCP/IP, Scalable Servers.

1.0 Introduction

Facing the amount of doubled Internet users each year, The conventional single server is getting unsatisfied for its workload [1]. The exponentially growing demand for Internet-based support, marketing, and software distribution services is increasingly driving companies or organizations to expand the capacities of their servers. This presents a serious challenge to Internet service providers and equipment suppliers in keeping up with this growth in traffic and number of users.

To increase the performance of a server, the first option that many companies use to scale their service is simply to move the server to a larger, faster machine. The replacement, however, is not a simple matter that may take time and sometimes interrupt services. The second option often emerges when increasing the server power, which is the speed of the connection to the Internet. Following a similar strategy to the server replacement, the connection upgrade shows the same disadvantages. When fault tolerance is taken into account, multiple mirrored servers can be placed to provide an acceptable level of service to network users.

2.0 General solutions and relative work

Currently, some mechanisms [2] are used to increase capability of scalability. There are caching, replication, dynamic request redirection, analyzing access pattern, etc [2]. A number of replicated servers, with location independence, can be selected either by manual or automatically. For example, some companies use manual selection in their WWW pages. With the user choice, users are prompted to identify either geographical locations or some other preference for servers, and then select the next Universal Resource Locator (URL) [3] appropriately. This solution shifts jobs of server selection to the user and may not balance the load among servers. Also, it is difficult to tell from the lists which services are actually available; that is, not already serving to capacity.

2.1 The NCSA's scalable WWW servers

The NCSA's way to scale their Web servers modifies the Domain Name Server (DNS) [4][5]. The modification of the DNS lookup process results in the DNS returning one of a set of IP addresses of servers providing the function. A DNS query for a named

service (for example, www.ncsa.uiuc.edu) is sent to the name server responsible for this particular domain. The DNS replies sequentially a different IP address each time for the same domain name. When more servers like to join the service, the modification of lookup table in the DNS is the only thing to do. As to the data sharing aspect, the AFS (Andrew File System) [6] is used in this system. The mechanism of whole file cache in the AFS promotes the efficiency in data access.

Using the round robin DNS, end users are connected to different servers in a cyclical pattern. Because the DNS knows nothing about network topology or server availability, end users can be connected to geographically distant or unavailable servers, resulting in poor access performance and increased transmissions costs. Also, it treats all servers as equal. As a result, less powerful servers may become oversubscribed, while larger servers remain underutilized. Lastly, the replied IP address may be hierarchically cached in DNSs or even in the client host itself and is used in subsequent name queries for a time.

2.2 ONE-IP

The ONE-IP, developed by Bell Lab. [7], is shown in Figure 1.

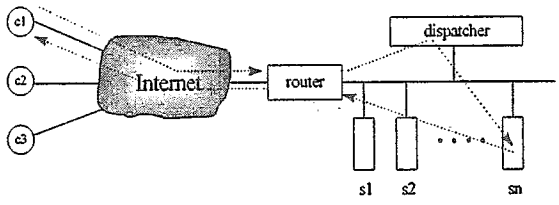

Figure 1: The System Architecture of ONE-IP

In Figure 1, the Dispatcher is responsible for delivering packets from clients (c1, c2 and c3) to web servers (s1, s2, sn). By using the function of IP alias, each web server keeps not only its own unique IP address but also a same alias IP address for the work group. It is this alias IP address that users think it as their server node. Every server uses this alias IP as its source address to reply incoming requests, sent from Dispatcher, to clients. The internal routing table of the router is modified, which directs packets with the destination of this alias IP address to the Dispatcher. Unlike the general ARP protocol, the router delivers packets to the Dispatcher with MAC address. The Dispatcher uses a hash function to switch the packet to a server chosen.

This solution works well without consuming too much internal network bandwidth. However, the function of *ifconfig IP alias* does not appear in all UNIX platforms. Moreover, the static hash function still can not balance the server group.

3.0 System design and implementation

The SSB is designed base on the criteria of load balance, single server image, high availability and application independent. Servers are replicated in a server group for the same service as shown in Figure 2.

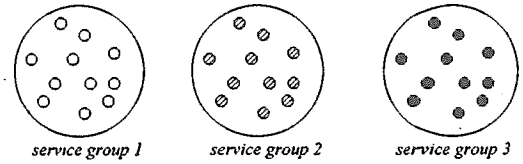

Figure 2: The server replication

3.1 System configuration

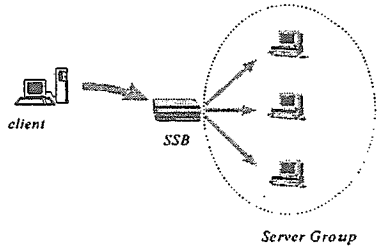

Figure 3: The Basic function of SSB

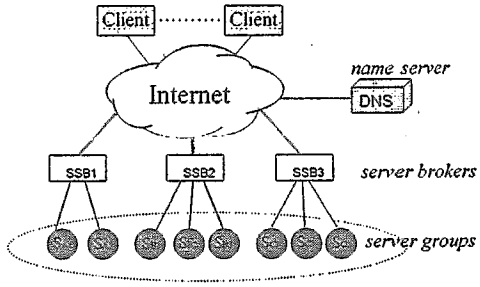

Figure 4: The System Configuration of SSRs

Figure 3 shows the basic function of SSB. The SSB gathers some server nodes and presents as a single service access point. Operations of joining/leaving a server can be easily performed. The system configuration of SSB is shown in Figure 4.

The SSB presents its unique IP address to the outside world as if it is the server node. When requested, it picks a lightest loaded server in the server group to have the service, while keeping transparent to the user. Upon receiving an IP packet, SSB modifies the destination IP address and switches the packet to the desired server. For continuing packets in a TCP connection without changing the server, SSB has to keep a Translation Table which records TCP connections established. Because SSB is somewhat like a gateway to the net, a ghost IP address and some virtual servers can be used in the system.

3.2 Address translation

The address translation is somehow like NAT specified in RFC 1631 [8] that is used to short term overcome the shortage of 32-bit IP addresses. The NAT separates global IP addresses and Local IP addresses with a gateway as shown in Figure 5.

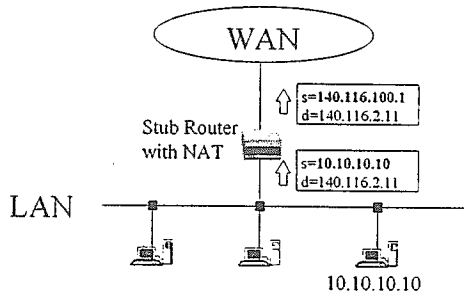

Figure 5: The Address Translation of NAT

The portions need to be modified in an IP packet are shadowed in Figure 6. The modification of IP packets actually does not satisfy the upper layer such as TCP or UDP. For example, The shadowed part of a TCP segment in Figure 7 needs to be modified too.

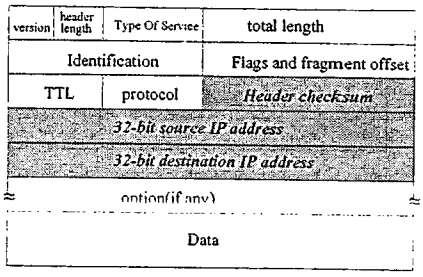

Figure 6: The Structure of an IP packet

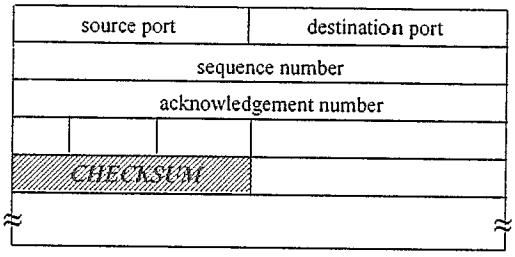

Figure 7: The Structure of a TCP Segment

The *Checksum* in the TCP segment is calculated as follows.

CHECKSUM = ip_checksum(*PSEUDO HEADER* + TCP Segment + Padding);

The *PSEUDO HEADER* consists of data structure shown in Figure 8.

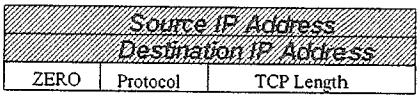

Figure 8: The format of PSEUDO HEADER

Like the modification in the IP packet, the *checksum* in the TCP segment needs to be modified. The calculation of checksum uses the method of incremental update described in RFC 1624 [9].

3.3 Modifications of FreeBSD

FreeBSD operating system is selected to be the platform of SSB implementation. The flow of packets in FreeBSD is shown in Figure 9, where the darkened arrows are the normal flow when configured as a gateway.

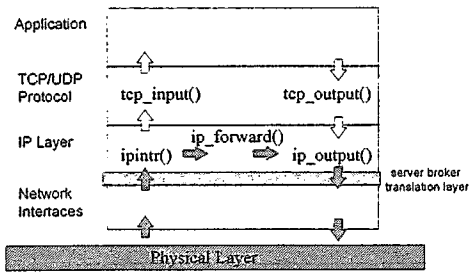

Figure 9: Flow of packets in FreeBSD

The hardware interrupt service routine packs the incoming data into an IP input queue, and then issues the software interrupt routine *ipintr( )* to proceed the queue. What the im0plementation needs to do is to modify the code before checking of IP layer. Therefore, A server broker translation layer is added. This keeps all functions in the original system work as normal.

A new function, named *ip_ssbin( )*, is added in the very beginning of the function *ipintr( )*, which lookups members, maintains translation table and modifies relative data in the packet. Similarly, a new function, named *ip_ssbout( )* is added in the very last of the function *ip_output( )*.

3.4 Load balancing and fault tolerance

As Figure 10 shows, every node in a server group has a monitor agent that monitors system status including information of CPU load, number of processes and I/O condition. The status will be sent to SSB in an optional time interval. The monitor in SSB collects information from monitor agents and maintains a status table of the service group. It is the weight data in the status table that is used to determine which server to redirect to. However, there are still some options that can be used such as round robin or random selection.

The monitor treats every message from monitor agent as a heartbeat of alive. When the interval of heartbeats exceeds the time-out value set in the status table, it is considered failed and can no longer provide services. The monitor in SSB can automatically insert or delete entries in the status table of server group. This simplifies work when a new server joins the service or a server leaves the group for maintenance because the system changes nothing except the server joined or left.

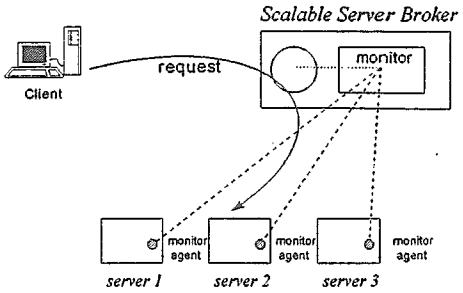

Figure 10: Load balancing in SSB system

3.5 Data consistency

Server replicas are necessary to keep data consistency. There are two options in this SSB system as follows:

a. Identical mode: Every server has enough capacity to retain all data. All modifications of data are applied on a master server while all other servers just mirror it from time to time.

b. File server mode: AFS or NFS are chosen to be the back ends of file system, which provide good data distribution performance to servers. Besides, TOFF (Transparent Operation Fault-tolerant File System) [10][11], developed in our lab, is a choice too.

4.0 Benchmark

There are several packages can be used to benchmark web servers [12]. For example, Gstone in Web66 project measures response latency and data throughput of web servers. SPECweb96 measures the number of HTTP operations and the average response time of web servers. NetPerf, by Hewlett-Packard, not dedicated for HTTP, provides indexes of network performance for variable UNIX and Windows NT environments. Webstone, used in this paper, has abilities to measure connection time, response time, data throughput and number of files/pages retrieved.

4.1 Testing environment

Webstone is developed by Silicon Graphics, Inc., with the source code included, which provides system simulation testing on Web servers. Figure 11 shows the testing environment of Webstone.

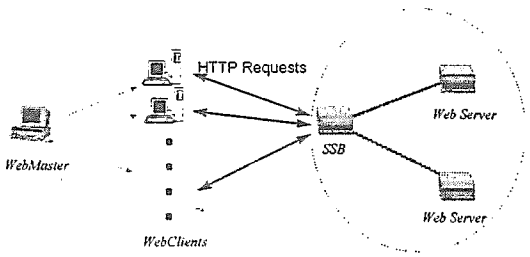

Figure 11: The testing environment of Webstone

The baseline of environment must contain at least two machines. One is the server to be tested while the other is responsible for simulating the workload of clients. The WebMaster uses system tools such as *rsh* or *rcp* to duplicate workload program to WebClients. After all programs in WebClients are ready, the WebMaster issues commands to ask WebClients send requests simultaneously. Operations of web browser can be simulated according the setting file in the WebMaster. The WebMaster collects reports from Webclients

4.2 Test results

Two identical servers on a 10Mbps Ethernet bus are used as the server group. A CGI program of electrical dictionary is used to simulate actions of a search engine. The test results are shown in Figure 12, 13, 14 and 15.

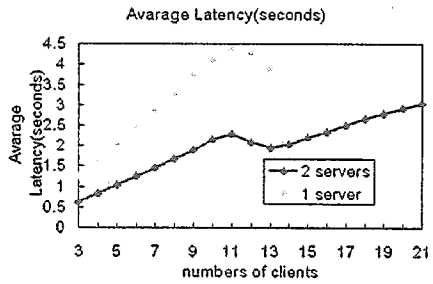

Figure 12: Average latency for the CGI program

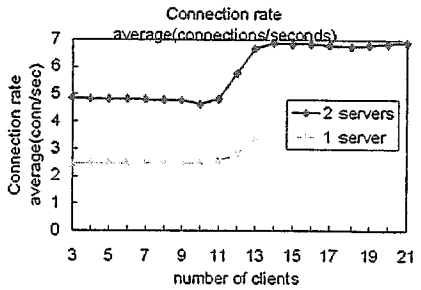

Figure 13: Connection rate for the CGI program

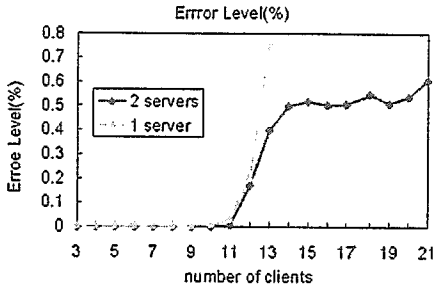

Figure 14: Error level for the CGI program

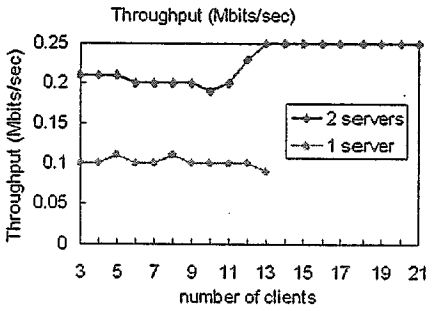

Figure 15: Throughput for the CGI program

The data of error level shown in Figure 14 happen when the server does not have enough resources or can not afford the workload and therefore replies a temporary error code to the client. Figure 12, 13, 14 and 15 show that the two-server system promotes almost twice in performance comparing to the single-server system. This is because the test condition is CPU bound. However, another opposite test condition of I/O bound is also applied that a number of clients continuously and randomly access 4 HTML files of different sizes. The test results show similar profiles as Figure 12, 13, 14 and 15 except that the performance of 2-server system is improved 1.5 times comparing to the

one of 1-server system due to the interference of network messages.

5.0 Conclusion

The scalability of servicing capability may be the problem in current Internet environment. It is preferable if the power of service can be easily adjusted according to the load. It is not necessary to replace a old server with a new powerful one and wastes the resource.

The SSB gathers the power of server group that can be on-line increased or decreased the number of servers. It is transparent to all TCP-based applications such as gopher, telnet, WWW, rlogin, etc.

The SSB also provides the ability of fault-tolerance that automatically exempts the bad one or joins the new one. However, the SSB itself is the weak point in fault-tolerance to the system. A primary-backup model to the SSB will be added in the next stage of the project. This shall overcome the problem when the SSB fail.

6.0 References

[1] LaLiberte, D. and Braverman, A. 1995. "A Protocol for scalable group and public annotations", *Computer networks and ISDN systems: Proceedings of the 3rd International WWW Conference*, v27, No. 6, P911-918.

[2] Mari Korkea-aho, "Scalability in Distributed Multimedia System", Master's thesis, Helsinki University, November 5,1995.

[3] T. Berners-Lee, M. McCahill, "Uniform Resource Locators (URL)", RFC 1738, December, 1994

[4] T.T. Kwan, R.E. McGrath, D.A. Read, "NCSA's World Wide Web Server: Design and Performance", *IEEE Computer*, Nov. 1995, pp. 68-74.

[5] E.D. Katz, M. Butler, R. McGrath, "A Scalable HTTP Server: The NCSA Prototype", *Computers networks and ISDN systems*, Vol. 27, 1994, pp. 155-164.

[6] J.H.Morris, M.Satyanarayanan, et al., "Andrew: A Distributed Personal Computing Environment", *Commun. of the ACM*, vol. 29, Mar 1986, pp. 184-201

[7] Yennun Huang, Om P. Damani, P. E. Chun, "ONE-IP: Techniques for Hosting a Service on a Cluster of Machines"

[8] K. Egevang, P. Francis, "The IP Network Address Translator (NAT)", RFC 1631.

[9] A. Rijsinghani, "Computaion of the Internet Checksum via Incremental Update" RFC1624.

[10] C.C. Chin, "Design and Implementation of a Fault Tolerant Network File Service", Master Dissertation, Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, June 1992.

[11] C.C.Chin, S.R. Tsai, "Transparency in a Replicated Network File System", *Proceedings 24th EUROMICRO conference*, 1996.

[12] Ed Tittel , "Benchmarking The Web", *SunWorld Online Feature*, September,1996.