Load Balancing Cluster Based on Linux Virtual Server

Shyam R. Ayyar[1], Pranita A. Bhagat[2], Vaishnavi D. Bante[3], Dr. A. K. Gaikwad[4]

[1,2,3]*Student, Dept. of Computer Science & Engineering, DES'S COET, Dhamangaon Rly*
[4]*Professor, Dept. of Computer Science & Engineering, DES'S COET, Dhamangaon Rly*

***Abstract** - With the explosive growth of the network, the workload on the servers is increasing rapidly, servers will be easily overloaded for a short time. Linux Virtual Server (LVS) which is an open source overcomes the overloading problem of the servers. Virtual server is a highly scalable and highly available server built on a cluster of real servers. The architecture of server cluster is fully transparent to end users, and the users interact with the cluster system as if it were only a single high-performance virtual server. The implemented ways of the virtual server are introduced. A LVS cluster of web servers based on VS/DR is presented for an online score system. The result of the application shows that the highly available LVS system is available and effective.*

Key Words: Internet-based laboratory, LVS, Quality-of-Service (QoS), real-time applications, .net Framework, Multicast path.

1. INTRODUCTION

Large-scale cluster-based network services are increasingly emerging to deliver highly scalable, available, and feature-rich user experiences. Inside those service clusters, a node can elect to provide services and it can also access services provided by other nodes. It serves as an internal server or client in each context respectively. Services are usually partitioned, replicated, aggregated, and then delivered to external clients through protocol gateways. Figure 1 illustrates the architecture of such a service cluster. In this example, the service cluster delivers a discussion group and a photo album service to wide-area browsers and wireless clients through web servers and WAP gateways. The discussion group service is delivered independently while the photo album service relies on an internal image store service. All the components (including protocol gateways) are replicated. In addition, the image store service is partitioned into two partition groups.

One of the advantages of a clustered system is that it has hardware and software redundancy, because the cluster system consists of a number of independent nodes, and each node runs a copy of operating Ease of Use system and application software. High availability can be achieved by detecting node or daemon failures and reconfiguring the system appropriately, so that the workload can be taken over by the remaining nodes in the cluster. High availability is a big field. An advanced highly available system may have a reliable group communication sub-system, membership management, quorum sub-systems, concurrent control sub-system and so on. There must be a lot of work to do. However, we can use some existing software packages to construct highly available LVS cluster systems now.

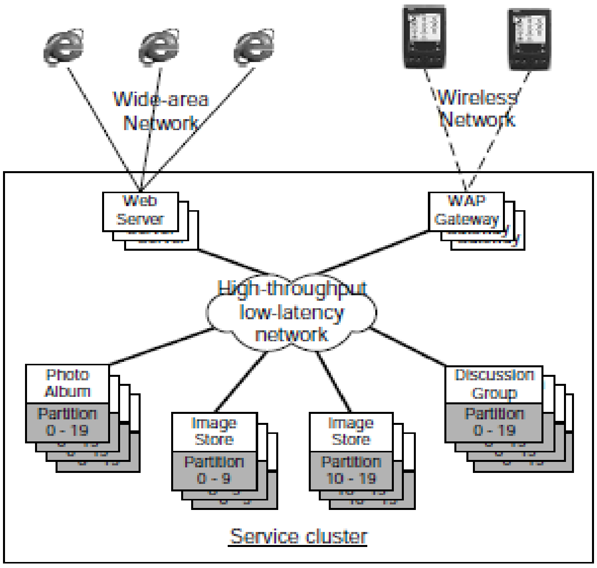

Wide-area Network
Wireless Network
Web Server
WAP Gateway
High-throughput low-latency network
Photo Album Partition 0 - 19
Image Store Partition 0 - 9
Image Store Partition 10 - 19
Discussion Group Partition 0 - 19
Service cluster

Figure 1: Architecture of a service cluster

While previous research has addressed the issues of scalability, availability, extensibility, and service replication support in building large-scale network service infrastructures, there is still a lack of comprehensive study on load balancing support in this context. This paper studies the issue of providing efficient load balancing support for accessing replicated services inside the service cluster. The request distribution between wide-area external clients and geographically distributed service clusters is out of the scope of this paper.

A large amount of work has been done by the industry and research community to optimize HTTP request distribution among a cluster of Web servers. This constraint calls for more complex load information dissemination schemes. Previous research has proposed and evaluated various load balancing policies for cluster-based distributed systems. Load balancing techniques in these studies are valuable in general, but not all of them can be applied for cluster-based network services. This is because they focus on coarse-grain computation and often ignore fine-grain jobs by simply processing them locally.

2. LOAD BALANCING

These days, LVS (Linux Virtual Server) software, which is free of charge and has good performance, has commonly been used to construct web server cluster. But when requests are increased, LVS can raise a bottleneck and can make the cluster system unable after all, because it has only single front-end. In this paper, we suggest a new architecture for web server cluster based on LVS with multiple front-ends which can act as back-ends simultaneously.

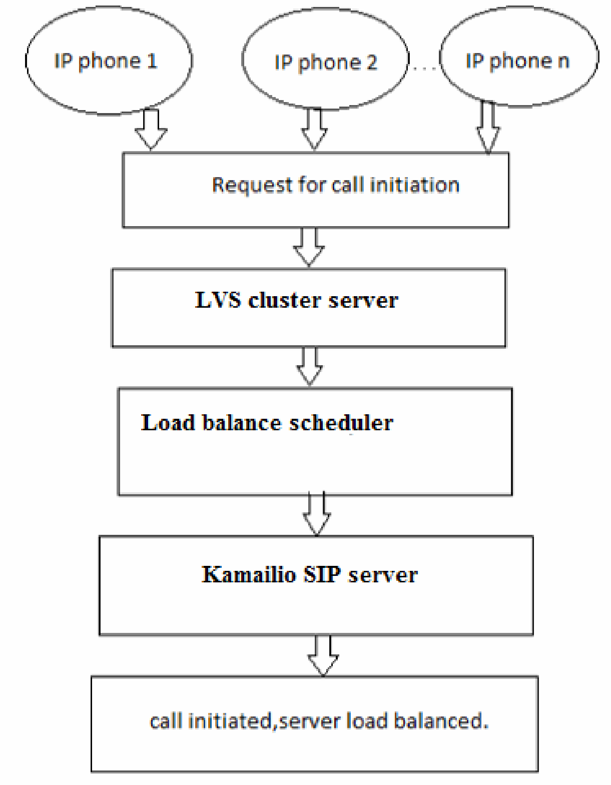


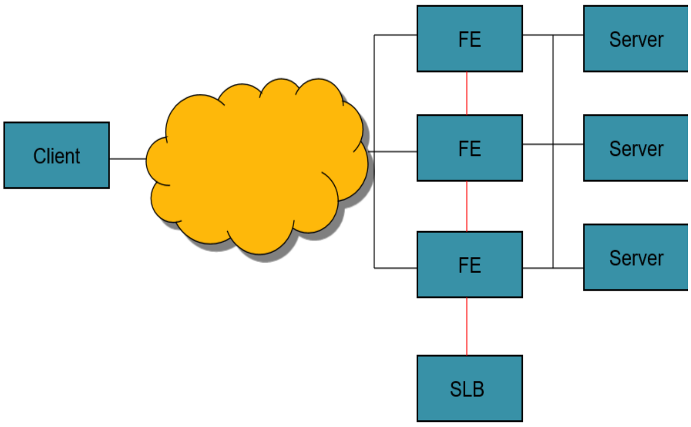
Figure 2: Flowchart for LVS clustering

We also propose a scheduling algorithm to distribute requests equally to servers with considering their load. The proposed system and algorithm remove the bottleneck, and are useful in constructing small and middle-sized applications. In the proposed system, a server can make response directly to clients request only when its load is not large enough. Otherwise, the server should redirect the request to a selected back-end with the lowest load, so it can avoid load.

3. WORKING PRINCIPLE

3.1: Network Load Balancing Architecture

Network Load Balancing runs as a network driver logically beneath higher-level application protocols, such as HTTP and FTP. On each cluster host, the driver acts as a filter between the network adapter's driver and the TCP/IP stack, allowing a portion of the incoming network traffic to be received by the host. This is how incoming client requests are partitioned and load-balanced among the cluster hosts. To maximize throughput and availability, Network Load Balancing uses fully distributed software architecture, and an identical copy of the Network Load Balancing driver that runs in parallel on each cluster host [2]. The fig.2 given below shows the implementation of Network Load Balancing as an intermediate driver in the Windows Server 2003 network stack.



This architecture maximizes throughput by using the broadcast subnet to deliver incoming network traffic to all cluster hosts and by eliminating the need to route incoming packets to individual cluster hosts. Since filtering unwanted packets is faster than routing packets (which involves receiving, examining, rewriting, and resending), Network Load Balancing delivers higher network throughput than dispatcher-based solutions. As network and server speeds grow, its throughput also grows proportionally, thus eliminating any dependency on a particular hardware routing implementation. For example, Network Load Balancing has demonstrated 250 megabits per second (Mbps) throughput on Gigabit networks.

3.2 Software Design

A software design is a description of the structure of the software to be implemented, the data which is part of the system, the interfaces between the system components and sometimes the algorithms used.

A. Architectural Design

In this section we present system architecture for building highly scalable and highly available network services on clusters.

1) Load balancer is the front end to the service as seen by the outside world. The load balancer directs network connections from clients who know a single IP address for services, to a set of servers that actually perform the work.

2) Server pool, consist of a cluster of servers that implement the actual services.

3) Users: users make call from one IP to IP, which is scheduled by Linux virtual server to the kamailio server

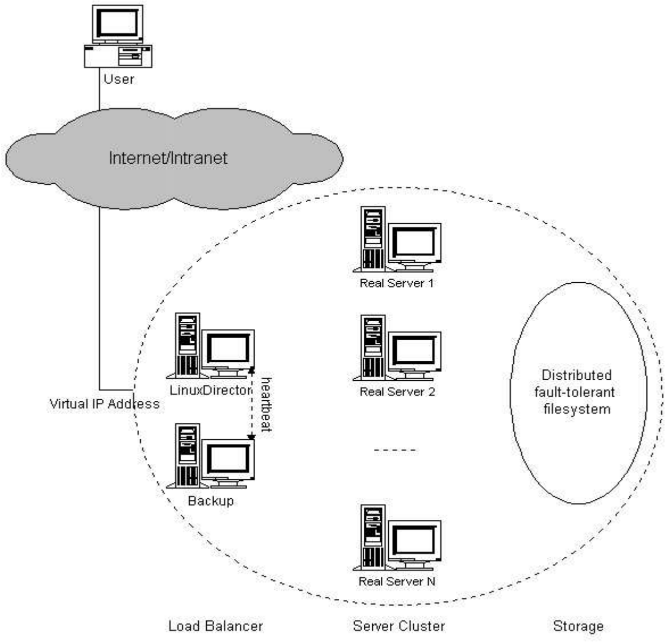


Figure 3: Architecture of Linux Virtual Server

4. APPLICATIONS

[1] Achieve optimal resource utilization

Minimum resources are required to achieve the network load.

[2] Maximize throughput

All the resources are shared and work in parallel manner so the output is high.

[3] Minimize response time

Here one backup server is used so that if the actual server fall down does not affects the response time.

[4] Avoid overload

As the cluster architecture is used in this load balancing so task are divided equally and the overload is avoided.

[5] Avoid crashing

Here we are using scalable servers that shares all the incoming traffic across various nodes due to which crashing is avoided.

[6] Reduces the job idle time

No any processor is in stable state the job is assigned on the serve base so the job idle time is reducing.

[7] Increase scalability

Network Load Balancing scales the performance of server based program, such as web server, by distributing its client requests across multiple server within the cluster. As traffic increases, additional server is added to the cluster.

5. CONCLUSION

In this paper it is concluded that, the load balancing in distributed systems is the most thrust area in research today as the demand of heterogeneous computing due to the wide use of internet. There exists no absolutely perfect balancing algorithm but one can use depending on the need. Finally, we studied some important dynamic load balancing algorithms and made their comparison to focus their importance in different situations. More efficient load balancing algorithm more is the performance of the computing system.

REFERENCES

[1] Yufeng Hu, Zhihai Sun, Shanan Zhu. Internet-based Electric and Electronic Experiment System[J]. Education Theory and Practice, 2007, 27(Special Issue):258-260

[2] J.J. Zhao, J.W. Ding, F.L. Zhou and L.P. CHEN: Journal of System Simulation, Vol. 18 (2006) No.2, p. 570

[3] Zhang W. Linux virtual server for scalable network services[C] // Ottawa Linux Symposium. 2000, 2000.

[4] Cassen A. Keepalived: Health checking for LVS & high availability[J]. URL http://www. linuxvirtualserver. org, 2002.

[5] Suntae Hwang, Naksoo Jung. Dynamic Scheduling of Web Server Cluster[C]. Proc. Of the 9th International Conference on Parallel and Distributed System, 2002: 563-568P

[6] Zhaolei Duan, Zhimin Gu. Dynamic Load Balancing in Web Cache Cluster[C]. 7th International Conf. on Grid and Cooperative Computing, 2008:147-150P

[7] Shi L, He Z H. Adaptive load balancing model based on prediction mechanism[J]. Jisuanji Yingyong/ Journal of Computer Applications, 2010, 30(7).

[8] Liu B, Xu J M, Dai S H, et al. Load Balancing Algorithm Based on Linux Virtual Server[J]. Computer Engineering, 2011, 23: 096.

[9] Hecht-Nielsen R. Theory of the backpropagation neural network[C]// Neural Networks, 1989.