Load Balancing Cluster Based on Linux Virtual Server

Shyam R. Ayyar¹, Pranita A. Bhagat², Vaishnavi D. Bante³, Dr. A. K. Gaikwad⁴

¹²³Student, Dept. of Computer Science & Engineering, DES’S COET, Dhamangaon Rly
⁴Professor, 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 one 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.

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.

![Flowchart for LVS clustering](image)

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. 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 figure given below shows the implementation of Network Load Balancing as an intermediate driver in the Windows Server 2003 network stack.

![Network Load Balancing](image)
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 kamaillio server.

![Figure 3: Architecture of Linux Virtual Server](image)

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


