Network Load Balancing

This chapter will cover the concepts involved with network load balancing. Network load balancing involves spreading the load of processing a web site out over multiple servers in a network. After this chapter you should be familiar with

- TCP/IP
- Basic routing
- Basic network connectivity testing
- Installation and administration of the IPVS application
- Common task scheduling methods, their advantages, disadvantages, and uses

Sharing the Workload

So, you have configured your servers and set up load balancing on your HTTP web servers and your database. You can connect to your web server and you can see the loads increasing on both the server and the database. Your servers are live but occasionally you load your web site and it’s slow. Yes, the load optimization you have performed is working. The issue is not with your software—it’s your hardware! Having balanced your HTTP servers and your databases, you have set up your system to function well but it’s still not enough.

It’s time to add new hardware.

New hardware means another server to share the load with your existing server, to share the work. The problem is how do you share this workload? How do you tell one server to take a connection while the other doesn’t? To understand, it’s best to start looking at the process of a user connection as a whole. The process flows like this:

- A user sends a connection to your server.
- The load balancer receives the connection and then fires another connection to the appropriate HTTP server.
- The HTTP server processes this request.
- If needed, the server connects to the database.
- If connected, the database responds.
The HTTP server forms a response.

The response is sent out to the initial recipient.

Looking at these steps, you can see what you have and have not load balanced. In this instance, you have modified the HTTP connections and the database connections to spread the system load.

This leaves you with two portions of the connection left to be balanced: the user’s connection in and the data transmission out that need to be configured to share load. To do this, you will add another server to the mix to divvy up the work and efficiently organize who will be doing what. But before you add this server, you should look more deeply at the connections that are being made to your server so you can understand exactly what’s going on when you start shuffling these connections around.

**TCP/IP**

TCP/IP refers to a pair of protocols that represent the most common stream of organized traffic over the Internet. In order to load balance data being transmitted by this protocol set, you should understand the two subprotocols that combine to form it.

**TCP**

Network load balancing (NLB) distributes network traffic across a number of servers to enhance scalability and availability of applications. For the purpose under discussion, NLB is about distributing the incoming connections that are made so that no single server of yours becomes overloaded. As part of load balancing any system, it is important to understand exactly what it is you are balancing. This means an investigation into the networking of the Internet!

Across the Internet all messages are broken up into tiny fragments called packets. This allows a much higher level of control over how the packets are transmitted and how they reach their destination; it also provides a very high level of redundancy. Moreover, since the Internet is so heavily diverse in the number of pathways it contains, having small packets instead of large ones allows for changes in the path a connection takes between a client and a server.

Imagine trying to send a file that will take three hours to send internationally over an Internet connection. Then imagine trying to send it in one big hit with no breaks. Add to that the risk that if any points of connection between you and your destination drop out, your entire connection is lost! You can see the advantages of using a method like packet switching.

However, packets are simply the way to fragment information to make it easier. What about managing these connections as a whole? The Internet uses many, many protocols for information sharing, but for HTTP web servers the connections use TCP (Transmission Control Protocol). TCP connections are the primary and most common connection established on the Internet.

TCP connections work like this:

- A client sends an initial message to a server requesting a connection.
- The server responds; the client and server perform a “handshake,” which allows the client and server to set up the rules of their communication.
- Following the handshake, the client and server enter the data transmission state, where they transfer data between them.