

# Purpose-Built Load Balancing The Advantages of Coyote Point Equalizer over Software-based Solutions



## **Abstract**

Coyote Point Equalizer appliances deliver traffic management solutions that provide high-availability, scalability and performance for servers or applications from a simple to manage plug-and-play device. Coyote Point load balancing products outperform software-based load balancers by delivering a cost-effective, OS independent traffic management solution that minimizes administrative overhead and maximizes site availability.

## **Introduction**

Early load balancing schemes provided website visitors the appearance of one server on a single URL while actually distributing traffic to a cluster of servers offering identical content, using crude load-sharing techniques such as round robin DNS. Load balancing quickly evolved into a more intelligent undertaking with software and devices making decisions about where to route incoming requests based on layer 4 information. As sites progressed to include dynamically generated content and applications that required session persistence, load balancing technology was required to advance once again, this time to include the ability to make layer 7 decisions when necessary.

Much like routers and servers, the market for load balancing has followed a path that has led to high performance purpose-built hardware and software combinations delivering the best price / performance. The appliance model has proven again and again to provide the most cost-effective solutions by eliminating much of the administrative overhead associated with building and maintaining systems. Appliance or hardware devices designed for a specific task allows site administrators to avoid the pitfalls of platform and OS dependent software solutions and implement a traffic management strategy that will deliver the highest availability for their servers regardless of the operating system or protocol.



## **Defining Load Balancing**

*definition* – Load balancing is a process in which client requests are distributed across multiple servers in an effort to increase the performance and availability of servers and network devices.

## **Types of Load Balancing**

> **Round-Robin DNS:** Delivers limited TCP/IP load balancing using DNS nameserver to map incoming IP requests one of several IP addresses defined. Typically used in small server farms or clusters where each server contains identical content.

> **Distributed Software:** A load balancing scheme where requests are received by any server in a cluster and then rerouted based on that servers ability to service the request. The term “distributed” software comes from the actual architecture of this type of load balancing, in that each server in a cluster must run either agents or an identical copy of the load balancing software, essentially making each server or node a load balancer itself.

> **Gateway or Dispatch Software:** Designed to run on general-purpose PCs or servers, gateway software is generally installed on a single machine that then becomes the point of traffic management or “gateway” managing all incoming client requests in a proxy like configuration. Requests are routed through this gateway server and then dispatched to the appropriate server in a cluster.

> **Hardware / Appliances:** Purpose-built devices that act as a gateway (typically in a multi-unit redundant configuration) to manage all incoming IP traffic requests as they reach the network. As an out-of-the-box solution, hardware/appliances generally eliminate many of the administrative tasks associated with software solutions and are designed to overcome the processor and speed limitations inherent in general-purpose servers.

Each of these solutions varies widely in the degree of effort they take to install, configure, manage and maintain. They also each have applications that the other may not be suited for. Administrators of large heterogeneous server farms will find their needs will vary widely from IT managers trying to optimize the response time of a few servers running the same operating system.



## **Software Load Balancing**

Software load balancing refers to any method of distributing server load that requires a software application to be installed on a single or multiple servers to create a server cluster. Software packages range from free open-source projects to those that require large licensing fees. One major disadvantage of all types of load balancing software are the hidden fees associated with this build-your-own method that requires hardware to run on, server resources and administrative time to install, configure, maintain and manage.

In the case of gateway software that intercepts and redirects requests to the servers physically located behind it, the positive effects of managing traffic can be easily outweighed by the processor limitations of that machines general-purpose hardware and operating system. Because of this, the throughput and subsequent response time of the entire cluster is limited to the speed and processing power of the gateway or dispatch server. In contrast, distributed software requires that an instance of that software be installed and running on every machine within a cluster. While this may eliminate the single processor limitations of gateway load balancing software, it can also create internal "heartbeat" and routing traffic. These network broadcasts from the load balancing software can place additional load on the network that does not directly translate to useful work. Distributed load balancing software also creates an administration issue by creating a scenario where updates and patches must be managed on not one server, but all servers within a cluster, a task that can become difficult as server farms grow.

### **Limitations of Software Load Balancing Solutions:**

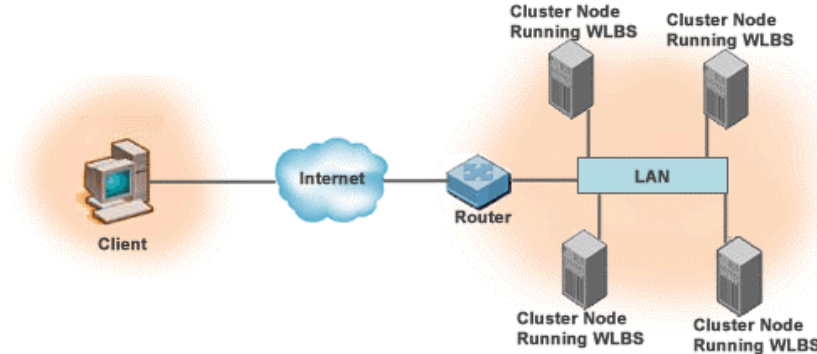
- Relies on general purpose commodity hardware to run
- Requires installation and maintenance, often on multiple servers
- Performance is limited by processor and speed of server it is installed on
- Creates hidden costs; hardware, management, maintenance
- Distributed software can create unnecessary internal traffic
- Limited intelligence in distributed software model as requests reach any server first

## **Popular Software Load Balancers**

### **Microsoft Windows NT Load Balancing Service (WLBS)**

Microsoft's WLBS is a feature of NT Server 4.0, Enterprise edition that allows for limited load balancing and clustering functions on Windows servers only. This software is a good example of the distributed software model, requiring that the application and drivers be installed and running on every server in a cluster. As with all load balancers, WLBS distributes IP traffic across servers in a defined cluster, but with limited functionality in that users are limited to a single predefined algorithm to map incoming requests to the appropriate server. Understanding that each server farm is different, this limits the applications of this software. While load balancing at its core is implemented to minimize response times and maximize availability, WLBS will only do so Microsoft web server applications and even then, only on Microsoft servers.



**Chart – Typical deployment of WLBS**

The illustration above depicts a typical installation of WLBS in a four server cluster. Each node or real server in the cluster must run its own copy of the WLBS software and driver. Incoming IP traffic requests reach any available node as they act as a single virtual server. Requests that cannot be served by the node they reach are then redirected to a server or node in that cluster that can.

**Key WLBS Advantages:**

- Designed for Microsoft only datacenter integration
- COMS+ and MSCS plug-ins manage COM+ objects and SQL database traffic

**Key WLBS Limitations:**

- OS dependent – only run on NT 4.0 server, Enterprise Edition
- Limited to 32 servers per cluster
- Nodes generate internal “heartbeat” traffic
- Run on general purpose PC hardware
- Subject to throughput and speed limitations of nodes/servers
- Hidden hardware costs
- Single proprietary load balancing algorithm
- Administration costs – installing, configuring and maintaining multiple instances

**Linux Virtual Server**

Linux Virtual Server (LVS) is the product of an open source Linux community project started to create a clustering and load balancing solution for Linux users. LVS is an example of gateway or dispatch software in that it utilizes a single PC or server running linux as the load balancer or gateway that accepts and distributes client requests to servers within a cluster.

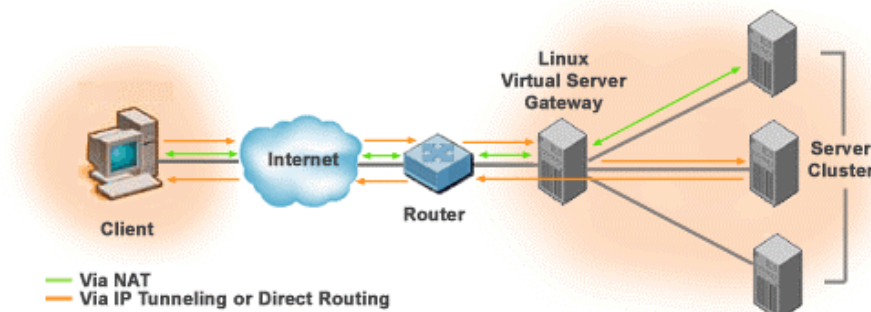
LVS can be implemented in three ways, each with its own advantages and limitations. The first is to run virtual server via NAT in which all requests and responses are required to pass through the load balancer gateway which requires significant hardware to keep that machine becoming a bottleneck. Based on this shortcoming, a load balancing implementation using LVS via NAT can theoretically only manage the load of 20 real servers. The second deployment option is LVS via IP tunneling in which requests are routed through the load balancer and real servers respond



directly to the client. While this helps eliminate the bottleneck created by NAT, it does require each server to have IP tunneling enabled, a feature, which at this point has only been demonstrated on Linux servers. The final scenario is LVS via direct routing. Direct routing works in the same way as IP tunneling, where only client requests (not responses) are processed, however it requires that the load balancer and real servers all have interfaces on the same physical segment.

Using commodity servers as the load balancing component, LVS is subject to the throughput and processing limitations of that machine and OS. The biggest drawback to using LVS in a mission critical enterprise environment is that as an open source solution there is no technical support available and no single point of accountability.

#### Chart – Typical LVS deployment



Typical deployment of LVS where load balancer is the front end for a Linux only cluster

#### Linux Virtual Server Advantages:

- No software investment – Free open source project
- Source code available to build custom solutions

#### Linux Virtual Server Limitations:

- Limited to 20 servers per cluster
- Must be installed on general purpose commodity hardware
- Subject to throughput and speed limitations of PC and OS
- OS dependent – must run on Linux and handle requests for Linux servers only
- Open Source – No technical support available
- Not application aware – simple L4 load balancing only

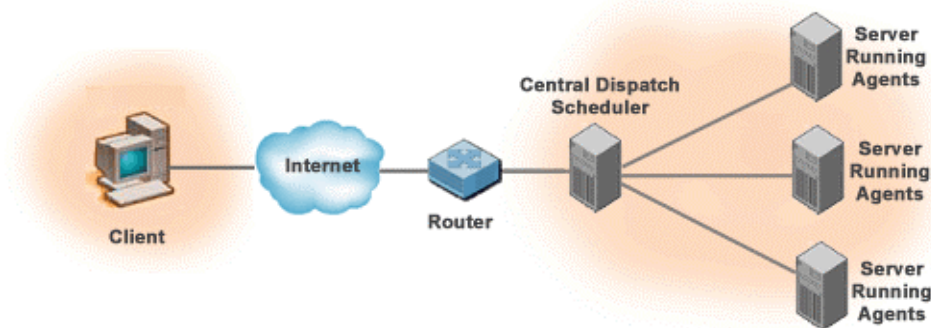


## Resonate Central Dispatch

Central Dispatch is another example of software load balancing. This package is a hybrid of the gateway/dispatch model and the distributed software model in that incoming requests are handled through a dedicated load balancer or “scheduler” but also requires that “agents” be installed on all servers within a cluster to communicate with the scheduler.

While this application may deliver more functionality than WLBS or LVS, it introduces the same network overhead and requires similar hardware investments. Resonates Central Dispatch is not as OS dependent as other software solutions (runs on Solaris, Windows NT, Windows 2000, AIX) but still requires that users install on a general purpose PC or server, using a single machine as the gateway which again is typically insufficient to handle aggressively increasing traffic loads.

### Chart – Typical Central Dispatch Installation



Client Requests are received by the scheduler and routed to the servers in the cluster or “central dispatch site” based on administrator defined rules and agent / scheduler communications.

### Central Dispatch Advantages:

- Free – limited function – version available for trial
- Manages XML traffic

### Central Dispatch Limitations:

- Requires installation and management of components on every machine in cluster
- Handling of concurrent requests limited by speed and throughput of scheduler processor
- Runs on general purpose commodity hardware
- OS dependent – Windows NT, Windows 2000, Solaris, AIX only
- Generate internal “heartbeat” or “health check” traffic
- Administration costs – installing, configuring and maintaining multiple instances



## **Hardware Load Balancing**

Hardware load balancing refers to a pre-configured PC, switch or router that is purpose-built to deliver a traffic management solution that enhances the response time, performance and availability of multiple servers within a network. The key concept behind hardware load balancing is to provide seamless network integration, working with any OS or platform, simplifying administrative tasks, including installation.

Hardware load balancing has continued to deliver increasingly more comprehensive feature sets than software load balancing. From using higher level algorithms to content-awareness, purpose-built hardware solutions continue to achieve higher performance and integrate more of the critical features required for today's high traffic sites. By eliminating the performance bottlenecks, hidden costs and administrative overhead of software solutions, hardware load balancers like the Coyote Point Equalizer product line have emerged as the clear price/performance leader in load balancing and traffic management.

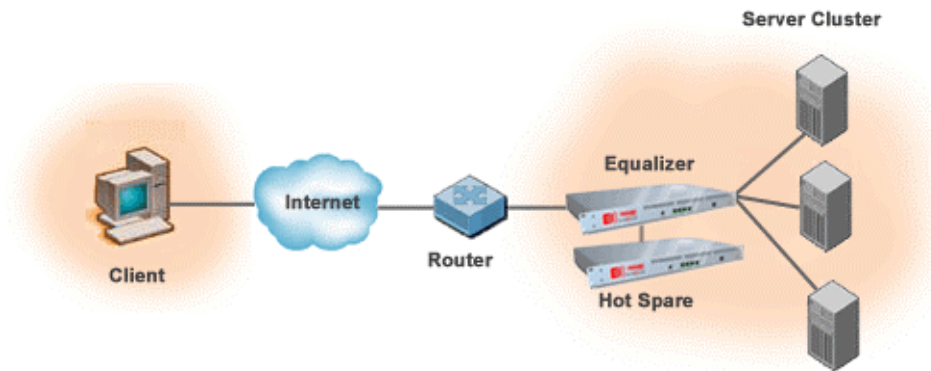
### **Hardware Load Balancing Strengths:**

- Seamless integration – often plug and play devices
- No server agents or software to install
- Purpose built for load balancing – no GP server limitations
- Simple to install, manage and maintain
- No hidden software and administrative costs

## **Coyote Point Equalizer**

Coyote Point Equalizer load balancers are a cost-effective appliance-based solution for managing the scalability, availability and performance requirements of any network infrastructure. By effectively managing Internet traffic, the Equalizer product line maximizes network potential by minimizing response times and ensuring site availability. The ability to operate independent of operating system and Internet protocol allows network administrators to deliver maximum performance for any mission-critical service or application in any network that requires high-availability and fault-tolerance.

### **Chart – Typical Equalizer Deployment**



As a gateway appliance, Coyote Point load balancers are typically deployed in a redundant configuration that includes a hot backup. Client requests are routed through the Equalizer to the appropriate server based on rules set by the administrator.



### **Coyote Point Equalizer Strengths**

- Compatible with any server or OS
- Manages Layer 4 and Layer 7 traffic
- Supports all standard TCP/IP client server protocols
- Supports unlimited number of virtual clusters
- Plug and Play – simplified installation and management
- No software or agents to install
- Can support up to 64 servers per virtual cluster and unlimited clusters
- Handles up to 4,000,000 simultaneous active connections

### **Why Equalizer?**

In this cost conscious IT environment, it is important to evaluate not only the purchase price but underlying costs of applications and devices being deployed in the network. Coyote Point has established itself as the price/performance leader in Internet traffic management (ITM) by delivering ITM solutions with a product line that provides ITM essentials – internet load balancing, reliable failover, intuitive configuration, robust reporting for servers and clusters, transport layer (L4) and application layer (L7) load balancing – at a fraction of the prices charged by hardware and software competitors. Beyond delivering an award-winning product, Coyote Point Equalizers simply outperform the competition.

### **Competitive Analysis Scalability and Performance**

	Coyote Point E450	Windows Load Balancing Service	Linux Virtual Server	Resonate Central Dispatch
Servers per Cluster	64	32	20	64
# of Virtual Clusters	unlimited	N/A	N/A	16
Recommended Bandwidth	Up to 500 Mb/s	Up to 250 Mb/s	Network dependent	Network dependent
Simultaneous Connections	4,000,000	Hardware dependent	Hardware dependent	Hardware dependent

### **Summary:**

Coyote Point's L4 and L7 load balancing solutions offer unmatched price/performance to meet the reliability, scalability and performance requirements of today's Internet-based economy. The Equalizer's time tested load balancing appliance architectures provide dramatic capital cost savings over software load balancing solutions in purchase price, resource utilization, management and overall operational efficiency. Coyote Point offers many distinct advantages that together deliver a highly efficient cost-effective traffic management solution for any network infrastructure and budget. Simply put, Equalizer lets you do more for less.

