



By Roger Howorth, 10 Sep 2008



VERDICT

Rating: ★★★★★

The Zeus Extensible Traffic Manager Load Balancer (ZXTM LB) is an enterprise class traffic management suite for web based applications. It's available in a range of formats and sizes to suit every requirement, deployment is fast and it also extremely easy to use.

EVALUATION

When it comes to load balancing Zeus has always had an Olympian reputation. Can its entry-level virtual appliance maintain this status?

Zeus is a British company with an enviable reputation for high performance web servers and load balancing systems. Earlier this year the Zeus Extensible Traffic Manager Load Balancer (ZXTM LB) was upgraded to version 5.

At a simple level load balancers work by taking traffic sent to a particular URL and redirecting it to a pool of several back-end servers. The ZXTM LB also handles more sophisticated functions, such as decrypting SSL requests so the back-end servers don't need to, or rewriting parts of incoming requests so software on client systems remains compatible with newer applications.

ZXTM LB works with pretty much any internet software, so besides helping with web sites it could be used with email systems or just about anything else that uses TCP/IP. New features in version 5 include support for IPv6, RTSP for streaming media, and SIP for voice over IP systems.

The Zeus suite is available in a variety of formats, including two hardware appliances based on Sun Fire servers for handling medium and large scale loads. Zeus also offers ZXTM LB as a virtual appliance, which is a virtual machine preconfigured with all the

necessary software ready to run on VMware ESX 3 or Microsoft Virtual Server 2005 R2. Pricing is based on the number of back-end servers that are to be load-balanced, and a range of support options are available to suite every taste.

The ESX 3 version of the Zeus virtual appliance uses 512MB RAM, one virtual NIC and a 4GB virtual hard disk, and a demo version is available for people that want to try before they buy. The virtual appliance could also be purchased and used for applications that have relatively little traffic.

Busier web sites would probably be better off either buying one of the hardware appliances, or buying the ZXTM LB 5 software and running it on their own x86, x86_64 and SPARC servers running supported versions of Linux, Solaris or FreeBSD.

A more advanced version of the suite, called ZXTM, includes a sophisticated programming language called TrafficScript, and ZXTM 5 includes new support for combining third party Java software with TrafficScript. This combination could be used, for example, to integrate Microsoft Active Directory into TrafficScript software. ZXTM LB can be upgraded to the full featured version simply by adding a licence number.

We tested the virtual appliance version of ZXTM Load Balancer running on VMware Workstation, but the same virtual appliance would run on VMware ESX Server. Regardless of which version you use, all ZXTMs are managed the same way, using an attractive and easy to use browser based interface.

We started by downloading and unzipping the ZXTM Load Balancer virtual appliance, and then added it to our VMware Workstation inventory and booted the system. The ZXTM LB welcome screen (see gallery) displayed information telling us how to connect to its browser based management console. As this was the first time we had run ZXTM LB, the system used DHCP to acquire an IP address and the management interface started with an Initial Configuration wizard. This presented the license agreement, and then asked for various parameters, such as the hostname and a suitable static IP address. The wizard finished by presenting a neat summary of the information we had entered (see gallery).

Having completed the initial setup, the Zeus main screen displayed a few status messages confirming that all was well, and told us it wasn't managing any traffic yet because we hadn't defined any virtual servers. The term "virtual server" could be a little confusing, especially for novice users running the virtual appliance. However, a ZXTM "virtual server" is simply a service running on ZXTM that passes incoming requests to the back-end servers.

Like most customers, the first thing we wanted to do with our new ZXTM LB was configure it to handle traffic to our web servers. The newly created ZXTM LB made this very easy for us by placing a link to the "Manage a new service" wizard on the

management console's front page. We clicked this link and the wizard asked for the IP address details of the back-end web servers and then created and started the our new virtual server.

We could then click on links in the administration console to manage its settings. For example, as our web servers were running WebDAV file services, we configured our service with session persistence so traffic from a particular browser would go to the same back-end server. We also configured our service to accept SSL connections. In effect, this added SSL support to our web site in one easy step, without us having to change any settings or configure individual SSL certificates on the back-end servers. For web sites with high security requirements we could have configured the SSL support so that ZXTM LB encrypted traffic between the virtual appliance and the back-end servers. However, for our tests we left this traffic unencrypted. There was a wide range of options for session persistence, but we chose the one to insert transparent cookies into the response to track sessions.

Although ZXTM LB runs on a Linux [operating system](#), the stylish web based user interface means administrators do not need Linux skills. In fact, the management console does a remarkably good job of monitoring the system configuration and warning about potential problems. For example, it issued a warning when we unplugged the network connection to one of our web servers, and even told us when we set an invalid default gateway. Likewise we found setting up and editing virtual servers extremely easy. It took us around one minute to configure SSL support for our site and generate a self signed certificate, which is much quicker than if we'd done this by hand on a Linux web server.

As well as providing fault tolerance for web services by spreading workloads across several servers, the ZXTM software makes it very easy to build fault tolerance into the actual load balancing architecture. Several ZXTM LBs can be cluster together extremely easily both to provide fault tolerance and to improve performance, such as if it was being used to handle lots of SSL traffic. Once they are clustered together, administrators need only update the configuration on one ZXTM, and the changes would be automatically replicated to the other systems in the cluster.