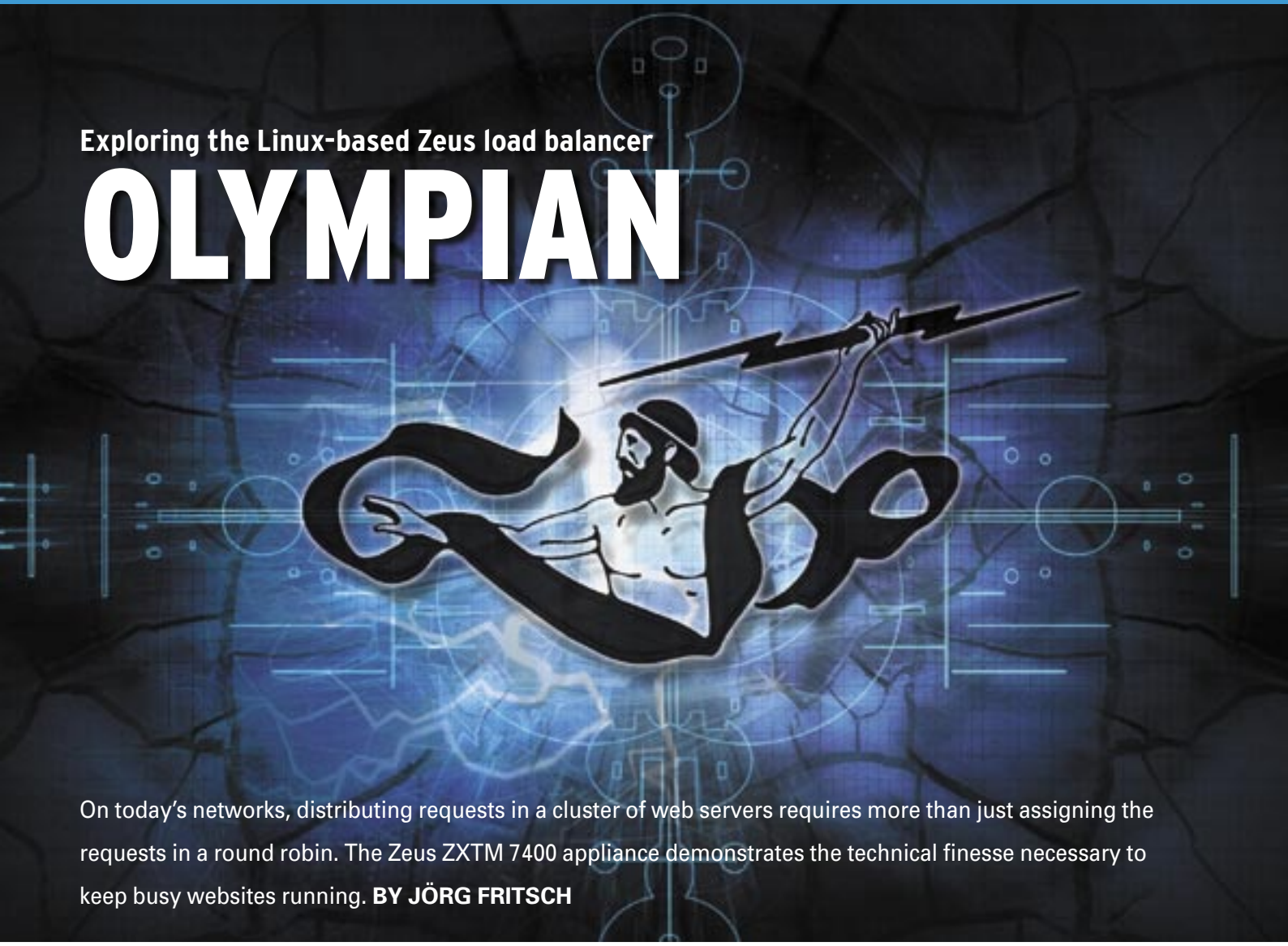
Exploring the Linux-based Zeus load balancer

OLYMPIAN



On today's networks, distributing requests in a cluster of web servers requires more than just assigning the requests in a round robin. The Zeus ZXTM 7400 appliance demonstrates the technical finesse necessary to keep busy websites running. **BY JÖRG FRITSCH**

Even the most powerful web server eventually reaches its limit. Getting help isn't a problem; the load redistribution can be complicated. Each server cluster needs to distribute requests intelligently to use resources in a meaningful way, and the client should not notice what is going on behind the scenes. One way of achieving this is the ZXTM 7400 appliance by Zeus Technology [1], which I recently tested.

Techniques

Load balancers distinguish between physical servers and virtual IPs (VIPs). In this case, the physical servers are web servers. Each web server has a unique IP (real IP, or RIP). The VIP is only configured on the load balancer.

Web clients only see the IP address belonging to the website, and they connect to this address without realizing they're talking to a load balancer that uses a scheduling algorithm to assign client requests to servers.

A fairly simple implementation of this principle works like destination network address translation (NAT), modifying the target address in the request (from VIP to RIP) and modifying responses to match (from RIP to VIP).

This variant is typically found in application-specific integrated circuit (ASIC)-based devices, which use highly specialized hardware to manipulate packets at extremely high speeds. In the early days of web load balancing (see the box titled "Seven Years of Load Balancing"), ASICs were thought to virtually guarantee sufficient performance.

If you need more performance, algorithms quickly become too complex for ASICs. Fortunately, the CPU performance of today's server and PC hardware is sufficient to cope with demanding tasks, as the appliance that I tested

THE AUTHOR

Jörg Fritsch has a degree in chemistry and works in the field of software development and IT security. He has worked in his current position as the Engineer for Communication and Information Security for the Nato C3 Agency since 2003. Jörg has also published a variety of work on load balancing, TCP/IP, and security.

Zeus ZXTM 7400

Task: Load balancing and application optimization for web applications

Technology: PC-based appliance with Linux and proprietary software by Zeus

Version tested: 4.1r1 on ZXTM appliance 7400

Price: Starts at around EUR 15,000 (US$ 20,000) for the entry-level device, ZXTM 2000 LB, up to around EUR 57,000 (US$ 78,000) for the high-end ZXTM 7400 appliance including all software options for the device tested here); the software without appliance costs between EUR 5,500 and 28,400 (US$ 7,500 and 39,600).

just goes to prove. The ZXTM 7400 works more like a reverse proxy that doesn't convert IP addresses but, instead, terminates the client's TCP connection and opens up a new TCP connection to the physical server.

This approach helps the appliance retain control over the connection, with the ability to manipulate the data stream. The physical server sees the load balancer's source IP address and sends the response to the appliance, which in turn sends it to the client.

Single Process Per Core

To keep pace with ASICs, PCs need some clever programming. The UK-based manufacturer Zeus has put much effort into developing optimized network software, thanks to its own web server.

On this basis, programmers deduced that a legacy multi-processing or multi-threading model would be insufficient because it would lose too much time on context changes.

The Zeus approach also reflects Dan Kegel's recommendations [2] for fast network software. The program uses the *epoll* mechanism to search for data in all open connections without blocking and without context changes. The developers have used nonblocking functions for all processing steps.

Health Status

One of the things that load balancers do is check the load and availability of the physical servers and evaluate these parameters. The scheduler uses this information to decide to which server to assign which requests.

At the same time, the scheduler has to keep sessions persistent on the servers on which they are running: In many web applications, the server stores information about the client status when users log in or fill virtual shopping carts, for example.

The load balancer needs to take this into consideration to avoid tripping up the application. More advanced load balancers implement a variety of techniques to discover which requests belong to the shared session.

For example, the Zeus appliance investigates cookies, adding its own cookies if needed, or uses many other techniques. Thus, it can even accelerate SOAP applications by load balancing.

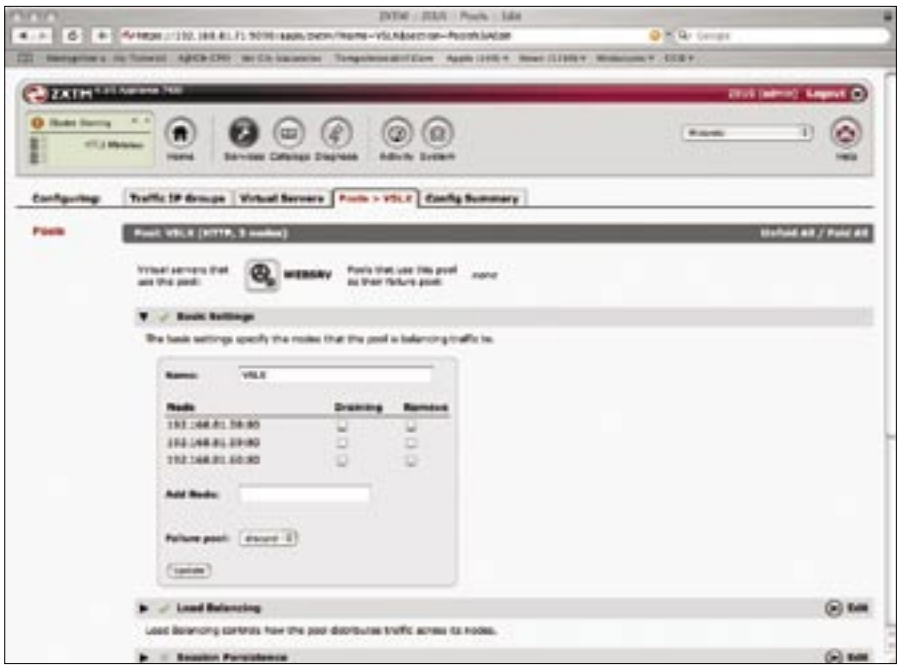


Figure 1: It is easy to create a simple pool with three physical web servers in the Zeus GUI. The load balancer selects one of these servers for each request.

To do so, the appliance analyzes the XML content of the messages.

ZXTM 7400

Zeus supplies the appliance with five network interfaces (see the "Hardware" box). One of these interfaces is mainly used for out-of-band management (OOB) – that is, for administrative access via a separate cable. A web GUI or a serial console are available for basic settings (network, default gateway). Because the appliance runs on Linux (a modified Debian Sarge with an Ubuntu kernel), Linux experts will soon feel at home at the command line.

The remaining configuration tasks are easy and intuitive: define a pool of physical servers (Figure 1), select Health-check, set up routing between physical servers and the load balancer, select VIP and scheduling algorithms (see Figure 2

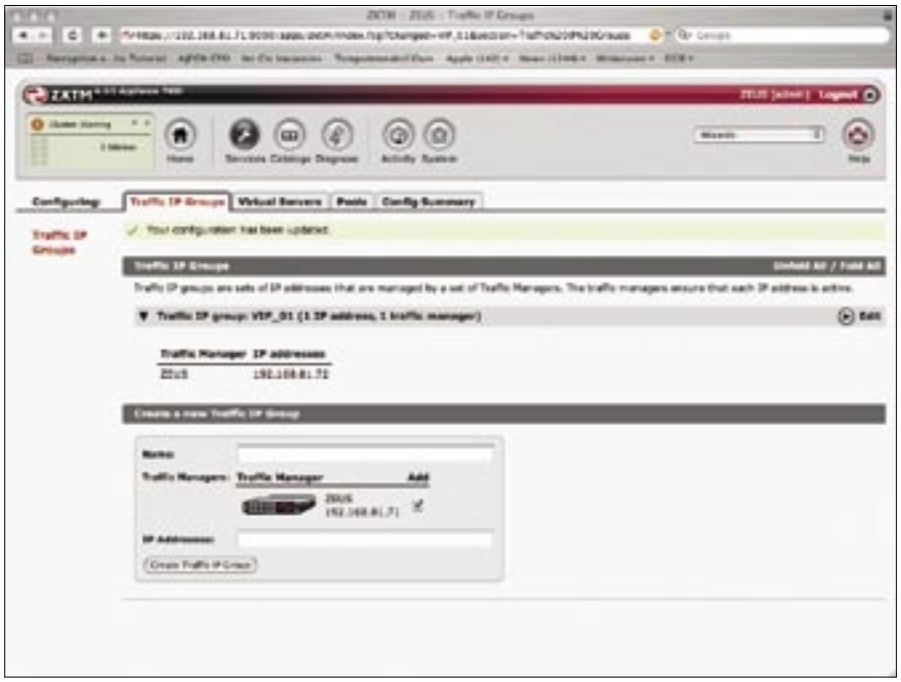


Figure 2: The load balancer needs a separate IP address to which clients connect. In our lab, the VIP was 192.168.81.72. The load balancer forwards requests just like a reverse proxy.

and Figure 3), decide whether some sessions need to be persistent on the server, and set up routing between the VIP and the clients.

To generate load for the web servers and the appliance in our lab, I used the Apache benchmarking program, *ab*. It is included with the basic installation of many Linux distributions. Interestingly, Adam Twiss programmed the first version of *ab* in 1996, and he is one of the two founders of Zeus. Since then, the Apache Software Foundation has maintained the tool. Additionally, Twiss has not worked for Zeus for many years. Thus, you can be certain that Zeus does not manipulate the measured values.

The real point of performing the test was not to discover the response time for every single request. Users aren't going to notice whether the virtual server takes 0.4 or 0.2 milliseconds to respond, which are just normal delays on the Internet. What's more important is linearity of the measured values – that is, that the virtual web server just takes twice as long to respond to 200 simultaneous requests as it does to respond to 100, and not four times as long. We also wanted to find out whether it is possible to reproduce the effective connection management by means of keepalives, HTTP 1.1 compression and intelligent caching.

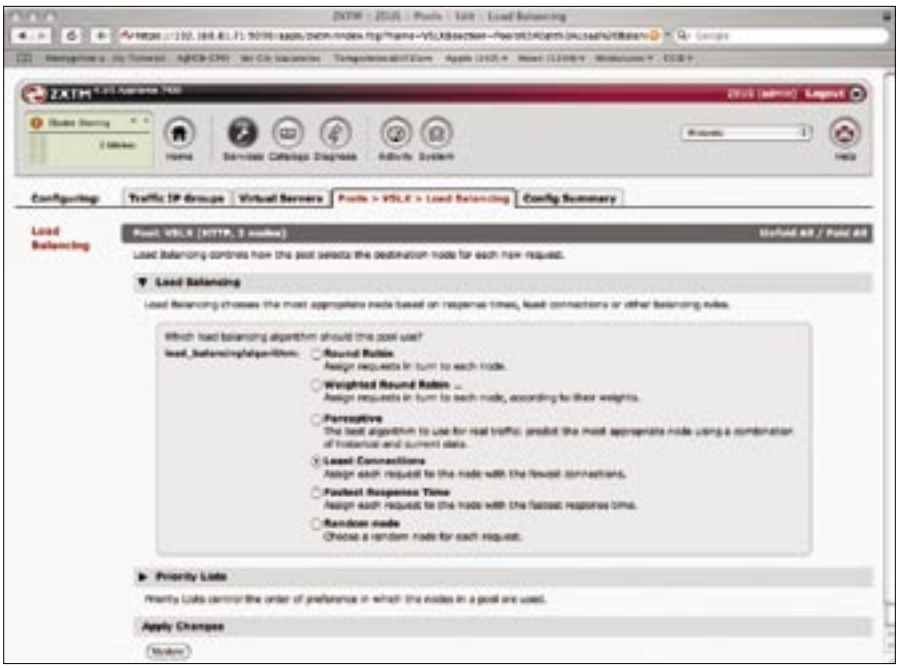


Figure 3: Zeus can handle basic load balancing tasks, offering six algorithms for selecting the physical server. Besides Layer 4 switching, the appliance also offers Layer 7 technology.

Excellent Measured Values

The measurements in Figure 4 were generated by requesting a 4KB web page from a cluster with two Apache servers (Figure 6) running on normal PC hardware. Load grew from 100 to 1000 concurrent HTTP requests. Of course, our lab conditions are fairly trivial compared with production use on e-commerce websites. An enterprise-level product like the Zeus ZXTM 7400 should be able to handle 50,000 to 100,000 HTTP requests per second.

According to the vendor, the appliance can handle up to 92,000 HTTP requests per second. Although I was unable to reproduce these results in our lab, I noted an interesting effect below the 1000 request level. Many CGI scripts and web applications start to slow down when faced with 70 or 100 simultaneous requests, instead demonstrating linear growth in response times.

The measured values show that keepalives do not offer any real gains if the load balancer only uses them in connections to the servers. Keepalives need to be enabled client-side to show any positive effect. In this case, the appliance simply opens a couple of connections to each physical server and routes any requests over the active connections.

The content cache also led to considerably improved measured values. In contrast, I could not detect any noticeable gains after enabling HTTP 1.1 compression, probably because of the HTTP implementation in *ab* and because I had network bandwidth to spare.

Seven Years of Load Balancing

The idea of using load balancing to run multiple parallel web servers has been through several reincarnations over the last seven years. When the dotcom bubble was at its peak (in 1999 to 2001), load balancing was justified because, in the tough world of e-commerce, traders with the fastest systems would make deals. This form of load balancing typically took place in OSI Layer 4, the transport layer.

After the events of September 11, the arguments started to change. Flash Events (FEs) or Flash Crowds generated sudden peak loads that could bring an underequipped server to its knees. An FE could be anything from a successful advertising campaigns to a popular news story.

As a potential bottleneck, a load balancer couldn't afford to be slow, which led to vendors opting for special ASIC-based hardware for their load balancers and application switches. Sophisticated scheduling helped them achieve maximum performance with the existing web server hardware. Few of the manufactures that offered PC-based systems at the time are still in business today. Two companies that survived are F5 Networks and Arraynetworks.

The Long Way Up

A modern Layer 7 load balancer, like the Zeus ZXTM 7400 appliance tested here, offers more performance and stability than the sum of its individual components. If you have four web servers and connect into a modern load balancer, you get more than four times the performance – at least, that's what the manufacturers promise. The new systems use a variety of approaches to achieve their goals. For example, they harmonize the handling of TCP connections and use Layer 7 technologies such as compression and intelligent content caching.

In addition, manufactures of Layer 7 load balancers can rely on most web presentations not being perfectly programmed and thus offering some scope for optimization. Because tricks of this kind require complex logic, many of today's vendors do without ASICs and use PC-based server hardware instead.

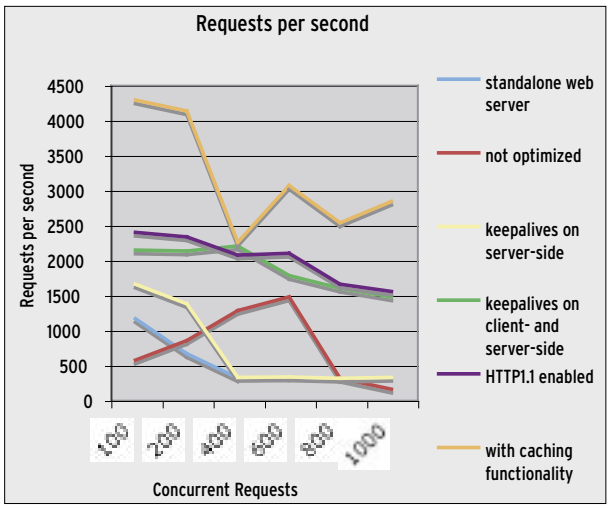


Figure 4: A cluster of two servers can serve up more than twice the number of pages per second thanks to the Zeus load balancer.

Performance-only values are no longer all that relevant because most load balancers do little to distinguish themselves. It makes little difference whether a load balancer achieves 92,000 or 100,000 requests per second. Additionally, vendor specifications are difficult to compare. If you look at ASIC systems, the results apply to Layer 4 load balancing, whereas manufactures of PC-based systems will tend to demonstrate performance in Layer 7. Flexibility, features, and stability are far more important. Today's intelligent load balancers dig deep into data communications, and even into content if necessary.

Script-Based Optimization

Zeus really starts to shine with its Trafficscript. For simple tasks, like evaluating the HTTP header fields, the browser-based GUI has a Rule Builder Wizard (Figure 7). For more demanding tasks, you need to write the rules yourself. To help you do so, Zeus gives you a 154-page reference manual. You can then use the web GUI to copy your script to the appliance.

Trafficscript is a scripting language that uses information from OSI Layers 3 through 7 to support decision making and to manipulate data. Zeus distinguishes between request rules for incoming requests and response rules for responses from physical servers. The following example takes information from Layer 7 (the */downloads* URL here) and adds a Layer 3 parameter (the type of service [TOS] bit in the IP header). The script is readable helps the admin by hiding the complexity of the protocols:

```
$url = http.getPath();
if (string.startsWith ➥
($url, "/downloads"))
{
    response.setToS ➥
("THROUGHPUT");
}
```

Figure 5: A separate appliance available this summer will distribute client requests over geographically diverse data centers.

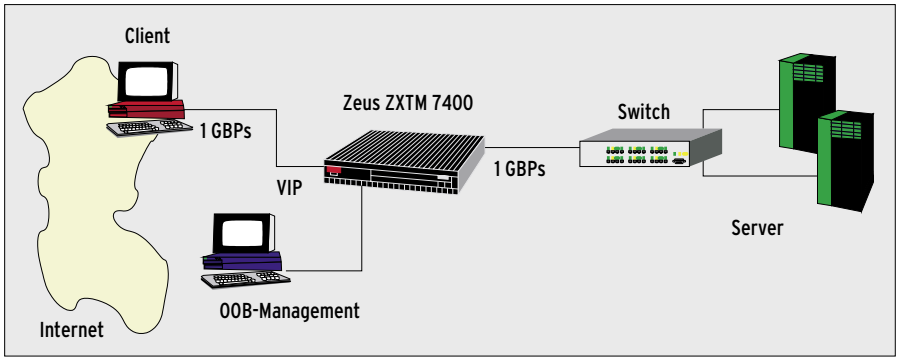


Figure 6: In our lab, the Zeus load balancer supported two physical servers. The administrator can control the appliance via OOB. Clients only see the external VIP; load balancing is completely invisible to them.

Global View

Global server load balancing (GSLB) is designed for really large sites. The idea behind GSLB is to distribute requests for clients over geographically spread data centers to ensure the availability of the website in case of a disaster. At the same time, clients benefit from receiving a response from the data center that they can reach most quickly. Adaptive content delivery networks (CDNs) really have no alternative to GSLB.

DNS Travels

The technology is based on the fact that clients use DNS requests to resolve the host and domain names to IP addresses for any websites they visit. The DNS server passes the request on to the load balancer, which retrieves the *CNAME* or *IN A* record for the VIP. This technology just has one weakness: The load balancer does not see the requesting clients directly, but only the request from the name server to which the client turned. Thus, it actually discovers the best VIP for the client's name server and hopes that the clients and the name server are not too far apart. This assumption is normally sensible because it is in the client's and the network carrier's best interest to locate DNS servers as close as possible to clients.

GSLB will not be included as a standard feature of the ZXTM Appliance series; instead, Zeus has announced a separate appliance for the summer of 2007.

If this rule is bound to a virtual server that a client addresses to request a URL starting with */downloads*, the appliance tags all the IP packets that belong to this connection with the TOS bit for throughput. The aim is to achieve maximum throughput for download files. Although the tag is only effective within a single Internet connection, this is often all it takes. The example also shows that the manufacturer has a very universal approach capable of connecting and manipulating Layers 3 through 7.

Other possible deployment scenarios might include inserting meta tags, evaluating HTTP headers (User-Agent, Accept-Language, …), or restricting the bandwidth for downloads. Zeus does not just distribute the load; the appliance also protects physical servers from excessive load. The techniques used to do this range from simple white- and blacklists of known clients, to connection limiting (maximum number of connections per client), to investigation of HTTP headers, to arbitrary rules implemented in Trafficscript.

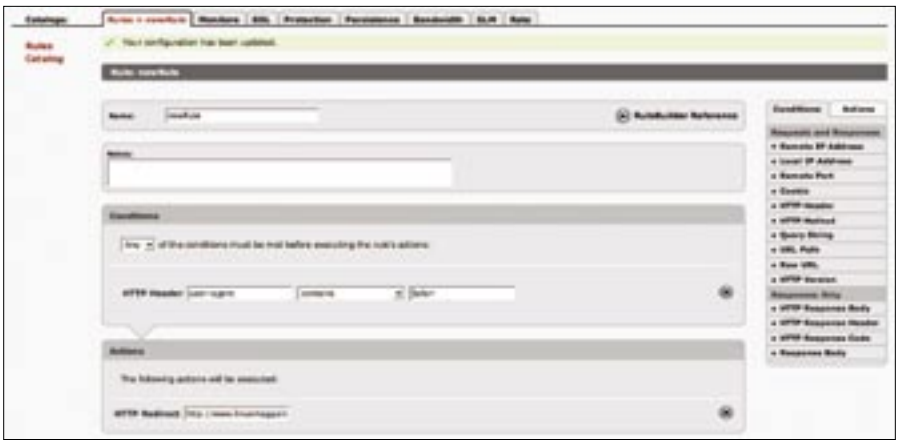


Figure 7: The Rule Builder wizard helps administrators quickly set up new Trafficscript rules. If you need more sophisticated manipulation, you can add your own scripts.

Hardware

The Zeus ZXTM 7400 appliance hardware comes from Germany's Pyramid Computer [3]. The appliance has a height of two rack units and includes a redundant power supply and five network interfaces, all of which support speeds of 10/100/1000MBps. The design is the typical Pyramid 2-HU server chassis with customized front panel for Zeus.

The basic hardware is quite popular on the European market. When I looked under the hood (Figure 8), I noted that the Supermicro motherboard has a clear-cut layout, and everything else is nicely arranged and connected. A plastic separator, which you can hardly see in the figure, divides the chassis into two halves – the CPUs and everything else. In each of these halves are two fans for cooling.

The ZXTM 7400 appliance has two AMD Opteron 280 CPUs (dual core, 2.64GHz, 64-bit). The 8GB of RAM comprises PC3200 modules (DDR1-400), although the more recent DDR2 667 RAM would have been faster. However, the vendor would have had to opt for the AMD Opteron 2000 series for DDR2 support.

The Supermicro motherboard has two 64-bit network cards on board. One of the four 64-bit PCI-X slots is occupied by a dual-port NIC, and another slot has a single-channel RAID controller. The single-channel controller appears to be fairly ancient on close inspection: it doesn't have a type label, and you need to set jumpers to configure it. According to the vendor, the controller is an ICP Vortex GDT8114RZ. Two 73GB disks are attached to it.

The system also has a PCI Express slot with a 32-bit network adapter sitting in it. You can expect lower throughput rates from this, so the manufacturer recommends using it for OOB management (i.e., via separate lines). The CD ROM drive in the appliance is not state of the art. A DVD drive would create a better impression.

Although the system is professionally built and promises good performance, the manufacturer loses a couple of points for its choice of CPU and RAM.

(*Norbert Landowski*)

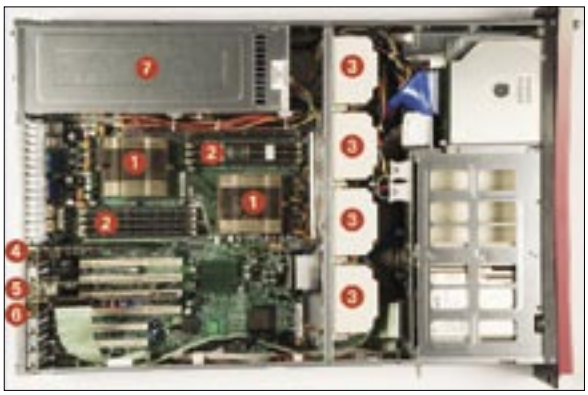


Figure 8: Inside the Zeus appliance: ❶ CPU, ❷ RAM, ❸ fans, ❹ 32-bit NIC, ❺ 64-bit dual-port NIC, ❻ raid controller, ❼ power supply.

Well Balanced

The Zeus ZXTM appliance impressed me with its flexibility, simple configuration, and excellent test results. A cluster can be more than the sum of all its servers thanks to Web Application Optimization. In particular, Trafficscript gives administrators plenty of scope for customization.

Zeus loses a couple of points compared with F5 and Nortel Networks when it comes to IPv6. Whereas the competitors already advertise full IPv6 support, Zeus refers to future releases, but without a tangible schedule. On the other hand, GSLB (see the "Global View" box) just goes to show that Zeus is on the right track and headed for a top spot in the major league. The hardware performed perfectly and also deserves top marks. You can expect this well-designed box to provide stable service.

For administrators, the ZXTM 7400 is a universal device that can handle any aspect of load balancing in a web server cluster, offering an impressive portfolio of techniques, from simple Layer 4 switching to low-level manipulation of the HTTP datastream and even of HTML documents. ■

INFO

[1] Zeus: *http://www.zeus.com*

[2] Dan Kegel, "The C10K problem": *http://www.kegel.com/c10k.html*

[3] Pyramid: *http://www.pyramid.de/en/index.php*