



ZLB Migration to ZXTM

A Zeus Technology white paper

Zeus Technology Limited
The Jeffreys Building
Cowley Road
Cambridge CB4 0WS
United Kingdom

Sales: +44 (0)1223 568555
Main: +44 (0)1223 525000
Fax: +44 (0)1223 525100
Email: info@zeus.com
Web: <http://www.zeus.com/>

Contents

Introduction	1
Copyright.....	2
Restricted Rights Legend	2
Trademarks	2
Contact Information	2
By Email.....	2
By Telephone	2
By Post or in Person.....	2
www.zeus.com	2
Differences between ZLB and ZXTM	3
Front-end Clustering	3
Virtual Servers instead of Ports	3
Pools.....	4
Load Balancing Algorithms.....	4
RuleBuilder and TrafficScript instead of Mapping Rules	4
Network Interfaces and Traffic IP groups	4
SSL.....	5
Tunables	5
Planning your migration.....	6
Front-end machines and traffic IP addresses	6
Back-end machines.....	6
Ports for non-HTTP(S) traffic.....	6
Ports for HTTP traffic.....	6
Ports for HTTPS traffic.....	7
SSL certificates	7
Traffic IP addresses	7
Performing a test migration.....	8
Install ZXTM and create your cluster	8
Create virtual servers for protocols other than HTTP and HTTPS	8
Create HTTP virtual servers.....	9
Import SSL certificates	9
Create your traffic IP group(s).....	10
Create HTTPS virtual servers.....	10
Tuning	10
File Descriptors and OS Tuning	10
Timeouts and responsiveness	11
Keepalives.....	11
Balancing algorithm tuning	11
Memory usage	11
Sessions.....	11
FTP.....	11
Other ZLB features	12
Miscellaneous Tunables	12
Test	12
After your migration.....	13
Optimising HTTP rules	13
Configuring the IP addresses a virtual server binds to	13
SSL decryption	13



Additional Notes	14
Switching between ZLB and ZXTM when both are installed.....	14
Conclusion	15



Introduction

Migrating from ZLB to ZXTM

The Zeus Extensible Traffic Manager product family replaces the Zeus Load Balancer product. Because there are significant differences in the way that ZXTM and ZLB operate and are managed, it is not possible to perform an automatic upgrade from ZLB to ZXTM.

This paper describes the ways in which the two products differ, and gives advice on how to plan and execute a migration of a ZLB cluster to ZXTM. It is intended for administrators of ZLB who are planning a migration to ZXTM.

We assume that the reader is familiar and comfortable with ZLB and its administration interface, and that the reader has attended the ZLB training course. We explain areas where ZLB and ZXTM differ and where these differences require consideration when planning a migration. We also point out features within ZXTM that may allow the administrator to improve the configuration.

The reader of this paper should also be familiar with ZXTM and its administration interface. This paper is not intended to replace the ZXTM manuals or on-line documentation. We recommend that if the reader has not already attended at least the ZXTM Standard training course, he or she attend it as soon as practicable.

You should read this paper thoroughly before starting to plan and execute your migration.

This paper is not designed to cover every single deployment of ZLB or every eventuality that may occur in the process of migrating to ZXTM. If your ZLB cluster has been tuned, either by yourselves or under guidance from Zeus Support or Zeus Professional Services, or if Zeus Professional Services was responsible for deploying your ZLB cluster, we recommend that you contact us prior to migrating so that we can advise you on the best course of action.



Copyright

© Zeus Technology Limited 2005. Copyright in this document belongs to Zeus Technology Limited. All rights are reserved.

Restricted Rights Legend

This document may not be reproduced in whole or in part in any manner or form (including photocopying or storing it in any medium by electronic means and whether or not transiently or incidentally to some or other use of this document) other than in accordance with any applicable licence agreement or with the prior written consent of Zeus Technology Limited. Any copies of this document must incorporate this notice.

Trademarks

Zeus Technology, the Zeus logo, Zeus Web Server, Zeus Load Balancer, Zeus Mass Hosting Application, ZISP CGI Toolkit, Zeus Extensible Traffic Manager, TrafficScript and RuleBuilder are trademarks of Zeus Technology Limited. Other trademarks may be owned by third parties.

Contact Information

If you would like to learn more about any of the topics covered by this white paper, please feel free to contact us for more information. You can reach us in a variety of ways:

By Email

For general enquiries:	info@zeus.com
For commercial and technical enquiries:	sales@zeus.com
For reseller information:	partners@zeus.com
For press and public relations information:	press@zeus.com

By Telephone

Main switchboard:	+44 (0)1223 525000
Information line:	+44 (0)1223 568555
Fax:	+44 (0)1223 525100

By Post or in Person

Zeus Technology Limited
The Jeffreys Building
Cowley Road
Cambridge
CB4 0WS
United Kingdom

www.zeus.com

Our web site contains a wealth of information on our products, services and solutions, as well as customer case studies, press information and a comprehensive knowledge base. For more information, please visit <http://www.zeus.com/>.



Differences between ZLB and ZXTM

There are a number of differences in the ways that ZLB and ZXTM operate. You will need to be aware of those differences and bear them in mind when planning your migration, since they affect the way you will need to configure your cluster.

In most cases, the differences between the two products affect only the way the products are configured and how multiple installations of the product communicate: generally there is no difference as far as back-end services is concerned. We point out where this is not the case.

Once you have migrated to ZXTM, your cluster will continue to operate in essentially the same way as with ZLB, although you will have further scope to optimise your cluster's configuration.

No dedicated administration server

A ZLB installation has one administration server. Zeus recommend that the machine on which the administration server is installed does not also handle traffic.

With ZXTM, each ZXTM in a cluster provides an administration interface: the separate administration server is no longer required.

Front-end Clustering

ZLB clustered front-ends into pairs, where each pair would have its own set of traffic IP addresses. If one machine in a pair failed, the other machine would pick up the IP addresses. If both machines failed, other machines would not pick up those IP addresses.

With ZXTM, all of the machines are clustered together, and any machine can pick up any IP address. So if you had 4 front-ends, the failure of 2 will not cause any traffic IP addresses to be lost as it could with ZLB.

Virtual Servers instead of Ports

ZLB used the concept of a Port to differentiate between different kinds of incoming traffic, where the port was the TCP/IP port on which traffic arrived. A particular protocol was then associated with that port. For instance, all incoming web traffic would be handled by a configuration on port 80, using the HTTP protocol.

ZXTM uses the concept of a Virtual Server, which is a combination of IP address(es), port number and protocol. This means that you could run several different services (perhaps a public Internet site and an Extranet) on the same port but using different IP addresses, more easily than with ZLB.

A ZXTM Virtual Server is capable of performing SSL decryption on incoming requests, and encrypting responses as they go back to clients.

A virtual server can listen on all IP addresses, which gives it the same behaviour as ZLB. We recommend that you initially create one ZXTM virtual server for each ZLB port, and configure those virtual servers to listen on all IP addresses. Once your migration is complete you can then consider splitting your virtual servers.

One situation where you may need to split your virtual servers during your migration is if you are serving multiple SSL sites. Refer to the section [SSL](#) for further details.



Pools

ZLB maintained one list of all of your back-end servers. When you created a port, or an HTTP mapping rule, you could select whether to send traffic to all the back-end servers or just to a subset of them. A back-end server in ZLB was just a hostname or IP address: ZLB determined what port number to connect to based on the port number of the service being balanced.

With ZXTM, there is no single list of servers. Instead ZXTM uses pools, which are a collection of hostnames/IP addresses and port numbers. So a pool of web server machines would generally list hostnames and port 80, while a pool of FTP servers would generally list the hostnames and port 21. However, unlike ZLB, different machines in a pool can have different port numbers.

Load Balancing Algorithms

ZLB provided only one load balancing algorithm, with a number of tunables to control its behaviour. ZXTM provides several algorithms, including the ZLB algorithm, which is called 'Perceptive'. The tunables controlling this algorithm have been removed, since in the situations where those tunables need to be modified an alternative algorithm such as 'Least Connections' will generally be more appropriate.

RuleBuilder and TrafficScript instead of Mapping Rules

For HTTP traffic, ZLB allowed you to define mapping rules, which would match a particular host header or URL and direct the request to a particular subset of your back-end machines. ZLB did not provide mapping rules for any other protocols.

ZXTM uses Rules, which are a more flexible approach to the mapping rules that work for all protocols. RuleBuilder gives you an easy way to create rules that behave in the same way as ZLB's mapping rules, whilst giving you greater flexibility and scope to make your rules simpler.

We give some examples of how you can optimise your rules in the section [After your migration](#).

Network Interfaces and Traffic IP groups

ZLB required each front-end machine to have two network interfaces, which you nominated as the 'front' and 'back' interfaces. Traffic IP addresses could only be assigned to the front network interface.

ZXTM does not require two network interfaces, although we continue to recommend two network interfaces for performance reasons. Interfaces are no longer explicitly nominated as front and back, and traffic IP addresses can be assigned to any interface.

Traffic IP addresses belong to 'Traffic IP groups', and a group can be handled on any combination of ZXTMs within the cluster. You can create any number of Traffic IP groups on ZXTM.

We recommend that initially you create one Traffic IP group that contains all of your traffic IP addresses, however if you are serving multiple SSL sites you will need to read the section [SSL](#) in detail.



SSL

ZXTM provides substantially different facilities for handling SSL compared to ZLB. To take advantage of those facilities certain other features of ZXTM have been modified compared to their counterparts in ZLB. Therefore if you are balancing SSL traffic with ZLB it is essential that you pay particular attention to this section, since the migration of SSL traffic from ZLB to ZXTM is somewhat more involved than for non-SSL traffic.

ZLB could load balance SSL-wrapped protocols such as HTTPS and IMAPS, and it used SSL session IDs for session affinity purposes, but it could not decrypt or encrypt SSL traffic. This meant that you could not use the HTTP mapping rules for HTTPS traffic.

ZXTM can perform this SSL pass-through, but it is also capable of decrypting and re-encrypting SSL traffic. Specifically, a virtual server can be configured to accept SSL-encrypted traffic from a client and decrypt it, while encrypting traffic being sent back to the client. Also, a pool can be configured to encrypt the traffic it sends to a node, and decrypt the responses.

By decrypting requests from the client, ZXTM is now able to perform content inspection even for encrypted protocols.

If you choose to decrypt traffic on ZXTM instead of using SSL passthrough, you may choose to have the pool re-encrypt the traffic before sending it on to the node. This means that you do not have to reconfigure your nodes to accept unencrypted traffic. However, sending the traffic through unencrypted saves you from the additional encrypt/decrypt stages in both directions (client -> server and server -> client).

When ZLB was used to load-balance a cluster of Zeus Web Servers handling HTTPS traffic, the web servers could be configured to receive a special prefix from ZLB containing the client and destination IP addresses. This allowed the web server to log the client IP address correctly, rather than believing the client to be ZLB. It also allowed the web server, based on the destination IP address, to choose the correct SSL certificate to present to the client. Remember that SSL negotiation – including the certificate exchange – takes place before the HTTP request is sent. This means that the server must commit to a certificate before it knows what Host header the client will use, which in turn means that every SSL certificate you use must have its own distinct set of IP addresses.

ZXTM can also add this prefix, provided you are using SSL passthrough in your virtual server. If you choose to decrypt traffic in the virtual server, and re-encrypt it in the pool, ZXTM will not add this prefix. You will therefore need to reconfigure ZWS. If you are in this situation, please contact Zeus Support who will be happy to advise you on your options.

Tunables

ZLB provided a number of tunables, such as those for timeouts and numbers of retries when attempting to connect to back-end servers, and the sizes of input and output buffers. In ZLB those tunables applied to all traffic being handled by ZLB, and they were adjusted by editing lines in the ZLB global configuration file.

For ZXTM, the tunables relating to communication with clients are now a property of a virtual server – and adjustable for each virtual server individually. Those tunables to do with communication with nodes are a property of a pool, and they are adjustable on a per-pool basis. The administration interface shows the current value of each tunable and allows them to be changed.



Planning your migration

Front-end machines and traffic IP addresses

We recommend that before you attempt to migrate a live ZLB cluster to ZXTM, you ensure that you are familiar with managing ZXTM by creating a test cluster on a separate set of machines. You can use this cluster to test that the virtual servers, pools and rules that you will be creating work as expected.

It is not possible to have ZLB and ZXTM running on the same machine simultaneously. If you wish to install ZXTM on the hardware that is currently providing your live service you will therefore appreciate that there will be some downtime while ZXTM is installed and configured on that hardware. You can leave ZLB installed when you install ZXTM, so once ZXTM is running you can switch back to ZLB if you need to – refer to the section [Switching between ZLB and ZXTM](#) for further details.

It is obviously very important should you take that approach that you thoroughly test your test cluster before shutting down ZLB, and similarly that you thoroughly test the newly-installed live ZXTM cluster.

If you have enough hardware to enable you to create a new live cluster using ZXTM while leaving your ZLB cluster in place, we would recommend taking that approach.

If you have sufficient available IP addresses to be able to assign new traffic IP addresses for ZXTM, we would recommend this too, especially if you are setting up new hardware for ZXTM. That way both of your clusters – ZLB and ZXTM – can be operational at the same time, allowing you to keep ZLB immediately available should you need to revert to it for any reason.

Back-end machines

Make a list of all of your back-end machines (those that run your web servers etc). This list can also be obtained from the ZLB admin interface. For instance,

<u>web1</u>
<u>web2</u>
<u>web3</u>
<u>asp1</u>
<u>asp2</u>
<u>ftp1</u>
<u>ftp2</u>

Ports for non-HTTP(S) traffic

Make a list of the ports on which you are balancing traffic that is neither HTTP nor HTTPS. For each of those ports, also list the back-end machines handling traffic for that port.

<u>Port</u>	<u>Machines</u>
21	ftp1 ftp2

Ports for HTTP traffic

Make a list of the ports on which you are balancing HTTP traffic. For each of those ports, also list the rules you are using (including the default rule), which machines are used by each rule, and whether transparent session affinity is enabled.



Port	Machines	Host Header	Rule	Session Affinity?
80	asp1 asp2	<i>blank</i>	<i>(.*)\.asp</i>	Yes
	web1 web2	www.mysite.com	<i>blank</i>	No
	web2 web3	<i>default rule</i>		No

Ports for HTTPS traffic

Make a list of the ports on which you are balancing HTTPS traffic. For each port list the back-end machines handling traffic for that port, the number of different sites (as identified by certificates) handled and the certificates used.

Port	Machines	Sites	Certificates
443	web1 web2	1	secure.mycompany.com

SSL certificates

If you have more than one site on an HTTPS port, you will need to obtain the private key and certificate for each site, in DER or PEM format, to install them into ZXTM. Although it is not necessary for HTTPS ports with only one site, or other SSL-wrapped protocols, we recommend that you do this for all SSL certificates covering traffic being handled by your ZLB cluster, since this will give you the opportunity to decrypt that traffic within ZXTM and take advantage of ZXTM's content inspection capabilities.

Traffic IP addresses

If you are reusing your ZLB traffic IP addresses for ZXTM, make a list of them. If you are using multiple addresses to handle multiple SSL sites, record which addresses relate to which SSL site.



Performing a test migration

This chapter takes you through installing ZXTM on a test cluster and creating a configuration that matches the one you have on ZLB, based on the information you gathered in the previous chapter.

Install ZXTM and create your cluster

On each of the machines you have decided to run ZXTM on, install the software as described in the Getting Started manual. Perform the first-time configuration on each of those machines.

When you install the first machine, you will need to create a new cluster. On all the other machines you will need to join the cluster you created on the first machine.

Use the ZXTM admin interface – on any of the machines – to verify that ZXTM has been installed on each machine and that they are all in the same cluster.

Create virtual servers for protocols other than HTTP and HTTPS

For each port that is handling a protocol other than HTTP or HTTPS, use the 'Manage a new Service' wizard to create a virtual server and the associated pool.

In Step 2 of the wizard, enter a descriptive name – this will be used on the virtual server and the pool. You can change both of these names later should you wish. Select the protocol that you are balancing – for instance SMTP if you are balancing port 25, or FTP on port 21. Confirm that the suggested port is correct, and change it if not.

If you are balancing an SSL-wrapped protocol, such as IMAPS, select SSL, and then select the protocol being wrapped. This will choose the port number appropriate for the protocol, again allowing you to change it if necessary.

In Step 3, add the hostnames (or IP addresses) of the back-end servers you used for this port in ZLB. ZXTM automatically offers the port number you selected in step 2.

Step 4 lets you review the virtual server and pool that will be created. If you gave your server the name 'SMTP', ZXTM will create a pool called 'SMTP pool' containing the hostnames you entered in Step 3.

The virtual servers that you have created are configured to listen on all IP addresses – click on the virtual server in the admin interface's home page to see its configuration. ZLB always listens on all IP addresses, so you don't need to change this setting.

The pools that ZXTM creates use the Ping monitor by default – this monitor attempts to ping the node, and treats it as live if the ping is returned. For a closer match to ZLB's behaviour, you should delete the Ping monitor and add a Connect monitor, which attempts to make a connection to the service itself. This is the mechanism ZLB uses to determine if a back-end server is live.

To add and delete monitors, click on Services on the top menu, then Pools. Click on the Edit button for the pool in question, then Health Monitoring. From there you can add new monitors and delete existing ones.



Create HTTP virtual servers

Create virtual servers for the HTTP ports you were balancing in ZLB in the same way as above. If you are using mapping rules, when the wizard asks you for the back-end nodes that will handle the service, enter only those nodes that are used by the default mapping rule.

For each of your mapping rules, you will need to create a Request Rule in the virtual server. You will also need to create an additional pool for each distinct set of back-end nodes used by your mapping rules – for instance `asp1` and `asp2`, or `web1`, `web2`, `web3` and `web4`.

Create the pools first. Remember that you will need to add the port number to each hostname, so you would add `asp1:80` and `asp2:80` as nodes in your ASP pool. Change the monitor from Ping to Connect.

To create Request Rules, go to the virtual server in question (click on Services, then Virtual Servers). Click on Rules, then 'Manage Rules in Catalog'. Make sure you select the link under Request Rules, not Response Rules.

Under 'Create new rule', type in a name for the rule, perhaps 'ASP Requests'. Make sure that RuleBuilder is selected, and that the 'associate with virtual server' box is ticked. Click the Create Rule button. This will bring up the RuleBuilder.

For mapping rules that match host headers, select HTTP Header under Condition. Set the HTTP Header to Host, and type in the host header into the rightmost box.

For mapping rules that examine the URL, select URL Path under Condition. Select 'matches regex' instead of 'equals', and copy the regular expression that you used in ZLB.

Under Actions, select Choose Pool and select the pool that you created with the nodes for that rule. Finally, use the Update button to confirm the rule.

You can create several conditions, for instance if you needed to match particular URLs within a specific host header. By default ZXTM requires that all the conditions succeed for the rule to take effect.

The last thing to do is to configure session affinity for those rules that require it. In ZXTM, session affinity is handled by the pool rather than in rules.

Go to Catalogs, and Persistence. Create a new Session Persistence class called Transparent Session Affinity, and set its type to be Transparent session affinity. Click Update to confirm that change.

Then, for each of your pools that require session affinity, configure the pool to use the Transparent Session Affinity persistence class: go to Services, then Pools. Click the Edit link against the pool you want to edit, then Session Persistence. Select the Transparent Session Affinity class and click Update.

Import SSL certificates

If you have more than one SSL site on a port in ZLB, you will need to decrypt all traffic arriving at the corresponding virtual server in ZXTM, as explained in the section [SSL](#).



To import an SSL certificate, go to Catalogs, SSL catalogs, then SSL Certificates catalog. Click on 'Import Certificate' and follow the prompts.

Create your traffic IP group(s)

If you do not have more than one SSL site on a port in ZLB, then you do not need to split your traffic IP addresses into multiple groups. You can therefore create just one traffic IP group with all of the traffic IP addresses you used in ZLB.

However, if you do have more than one HTTPS site, you will have created more than one traffic IP address for each machine in your ZLB cluster. You can create a single traffic IP group containing all the traffic IP addresses, but we would recommend creating separate traffic IP groups for each distinct HTTPS site. This will make it easier to alter the ZXTM configuration for these sites at a later date.

To create a traffic IP group, go to Services, Traffic IP Groups. Choose a name for the group (perhaps ZLB Migration), and enter all your traffic IP addresses. Then click Create Traffic IP group.

ZXTM distributes the IP addresses within the group and brings them up as soon as the group is created.

Create HTTPS virtual servers

If you wish to use SSL passthrough, where ZXTM passes the traffic between client and back-end server without doing any decryption of its own, then HTTPS virtual servers are treated in the same way as other SSL-wrapped virtual servers, using the instructions in the section [Create Virtual Servers for Protocols other than HTTP and HTTPS](#). If you are using ZWS, you will also need to enable the 'ssl_enhance' option in the pool configuration. This enables the SSL extensions supported by ZLB, ZWS and ZXTM that allow ZWS to determine the client IP address correctly.

If you are not using ZWS, you should not enable the ssl_enhance option. Doing so will cause your servers not to work correctly.

Tuning

All tunables in ZLB apply to all traffic being balanced. In ZXTM, tunables that relate to the communication between clients and ZXTM become the responsibility of a virtual server, and the tunables can have different settings in different virtual servers. Tunables that relate to communication between ZXTM and back-end servers are the responsibility of a pool. Again, they can have different settings for different pools.

If you have modified any of the settings that are now specific to a virtual server or a pool, you should modify those settings only on the virtual servers or pools that require them.

File Descriptors and OS Tuning

ZXTM automatically attempts to configure the operating system to give it as many file descriptors as possible.

Other operating system tuning, for instance timeout settings on the TCP/IP stack, should be the same for ZXTM as they are for ZLB. Your own records should show what tunings you have made to these parameters.



Timeouts and responsiveness

ZLB Tunable	Location in ZXTM
timeout	Virtual Server, Connection Management, Timeout Settings
max_connect_time	Pool, Connection Management
max_reply_time	Pool, Connection Management
max_retries	Settings, Global Settings, Connection Settings
max_connect_tries	No equivalent
dead_time	Settings, Global Settings, Other Settings
monitoring_time	Monitor, Basic Settings Now known as timeout

Keepalives

ZLB Tunable	Location in ZXTM
keepalive	Virtual Server, Connection Management, HTTP-Specific Settings
keepalive_backend	Pool, Connection Management Now known as keepalive
keepalive_timeout	Virtual Server, Connection Management, HTTP-Specific Settings
backend_timeout	Settings, Global Settings, Connection Settings
max_keepalives	Settings, Global Settings, Connection Settings
use_chunked	No equivalent

HTTP-specific settings are only available on virtual servers that are handling the HTTP protocol.

Balancing algorithm tuning

The balancing algorithm tunables in ZLB have no direct equivalent in ZXTM. Instead you should choose the appropriate balancing algorithm for each pool.

Memory usage

ZLB Tunable	Location in ZXTM
client_buffer_size	Virtual Server, Connection Management, Memory Limits Now known as max_client_buffer
server_buffer_size	Virtual Server, Connection Management, Memory Limits Now known as max_server_buffer
client_buffer_ext	No equivalent
max_header_size	No equivalent

Sessions

Session management is now the responsibility of a pool. Each pool can have a different session persistence class, which controls how sessions are detected and managed.

FTP

ZLB Tunable	Location in ZXTM
ftp_buffer_size	No equivalent
ftp_force_client_secure	Virtual Server, Connection Management, FTP-Specific Settings
ftp_force_server_secure	Virtual Server, Connection Management, FTP-Specific Settings

FTP-specific settings are only available on virtual servers that are handling the FTP protocol.

There is also the ftp_bind_20 option in Settings, Global Settings, which determines whether or not ZXTM should initiate all data connections from port 20. ZLB did not offer this facility, and by default ZXTM does not do it.



The following tunables in ZLB have corresponding settings in ZXTM's Global Settings page (Settings -> Global Settings). These settings apply to all traffic being handled by ZXTM.

Other ZLB features

ZLB Tunable	Location in ZXTM
port_offset	Use specific port numbers in pools instead
mail_interval	
history_size	No equivalent
ssl_zws_protocol	Pool, SSL Now known as ssl_enhance
add_cluster_ip	Virtual Server, Connection Management, HTTP-Specific Settings
read_zeus_backend	No equivalent
logtime	Not directly supported – use Bandwidth Management instead

ZLB's ssl_zws_protocol setting was only applied to nodes that ZLB knew to be Zeus Web Server. This took place automatically, and for a server that was not ZWS the setting had no effect. ZXTM only supports the ZWS v4 version of the protocol – sending both client and server IP addresses – and it must be enabled manually for an entire pool.

Miscellaneous Tunables

ZLB Tunable	Location in ZXTM
so_nagle_off	Settings, Global Settings, System Settings Now known as so_nagle – note that the sense is reversed
socket_opt	Settings, Global Settings, System Settings
multiple_accept	Settings, Global Settings, System Settings
num_children	Handled in global.cfg
use_poll/devpoll/ select/kevent/backoff	No equivalents
so_rbuff_size	Settings, Global Settings, System Settings
so_wbuff_size	Settings, Global Settings, System Settings
maxfds	Automatically configured
listen_queue_size	Settings, Global Settings, System Settings
read_on_connect	No equivalent
write_on_connect	Virtual Server, Connection Management, Low-Level Settings

Test

Once you have configured ZXTM, you should test to make sure that all of your services are being handled correctly: that you can access all of your servers, that sessions work as expected, and that your rules direct traffic correctly.

If you are having problems you will find ZXTM's logging to be more comprehensive than ZLB's, so you should be able to find and fix the problem.

You should ensure that you are entirely happy with ZXTM's behaviour and performance in your test environment, by simulating the levels of load that you are seeing on your live environment, before you deploy ZXTM in the live environment. Likewise, you should ensure that the live deployment is performing as required before decommissioning ZLB.



After your migration

There are steps that you can now take to improve your configuration, for instance to combine many mapping rules into one RuleBuilder or TrafficScript rule, or to enable SSL decryption for HTTPS – giving you all the HTTP inspection capabilities of ZXTM on your SSL sites too. This section gives advice on some of the things that you can do with your new ZXTM configuration.

Optimising HTTP rules

With ZLB, if you had many different host headers or URL patterns that you needed to match, and send all matching requests to the same set of back-ends, you would need many mapping rules. This is particularly true if you had many host headers in use.

With ZXTM's RuleBuilder you can combine rules – so a rule could match traffic for one pattern, or for another pattern, or for a third. You can also examine any header in a request, or use the client's IP address in making your routing decision. TrafficScript gives you even more control over request and response processing.

Configuring the IP addresses a virtual server binds to

With ZLB, a service binds to all IP addresses. This means that if you wanted to run several different services on the same port – perhaps internal-facing and external-facing SMTP servers, you could not achieve this without complicated workarounds.

With ZXTM, a virtual server can listen on all IP addresses, or a specific list of IP addresses, or one or more traffic IP groups. Continuing with the SMTP example, you could create two traffic IP groups – one with external IP addresses for external clients, and one with internal IP addresses for internal clients – and two virtual servers, one for each traffic IP group. The SMTP servers that are used by the two virtual servers can then be configured appropriately for their intended purpose.

Note that with ZXTM you can create traffic IP addresses on internal interfaces. This is useful if for example you need to provide a load-balanced service to your own servers. For instance, web servers may need to access an LDAP directory, which because of the load needs to be balanced across several LDAP servers. By providing a traffic IP address on an internal interface, the availability of the LDAP service to the web servers can be guaranteed in the same way as a service to external clients.

SSL decryption

ZLB handled SSL traffic by passing it straight through to a back-end server. With ZXTM, you can decrypt SSL traffic on ZXTM itself, giving you access to the power of RuleBuilder and TrafficScript. The traffic to your back-end servers can either be left unencrypted, or re-encrypted.

Modifying an SSL-passthrough virtual server to decrypt and re-encrypt its traffic is as easy as selecting the 'SSL Decrypt a service' wizard and following the prompts.

For HTTPS virtual servers passing their traffic to ZWS, you will need to make changes to your ZWS configuration that will render it incompatible with ZLB. You should therefore not attempt this change until you have decided to decommission ZLB. Please contact Zeus Support who will be happy to advise you of the changes you will need to make to your ZXTM and ZWS configurations.



Additional Notes

Switching between ZLB and ZXTM when both are installed

Although we recommend that ZLB and ZXTM not be installed on the same machine, there may be situations where this is unavoidable.

We strongly recommend that you do not run ZLB and ZXTM concurrently on the same machine, and it is essential that you do not run them together if both products are configured to manage the same traffic IP addresses. Please note that in this case (the same traffic IP addresses configured on ZLB and ZXTM) you must not run ZLB on some machines and ZXTM on others.

You can stop either product by running `$ZEUSHOME/stop-zeus`, where `$ZEUSHOME` is the directory in which the product was installed. Similarly a product can be started with the `$ZEUSHOME/start-zeus` script.

You will probably have files and symbolic links in `/etc` to start a product automatically at boot. These can be edited to select the correct product to start.



Conclusion

This paper has taken you through the process of migrating your ZLB configuration to ZXTM, giving you access to the many new features of ZXTM.

If you have had problems following these instructions, and need help getting your configuration working, please contact Zeus Support, on support@zeus.com. You can create a Technical Support report from within ZXTM, and this will be extremely useful. Go to Diagnose, then Technical Support.

If your ZLB cluster has had significant tuning either by yourselves or by Zeus Support or Professional Services, or it has been designed and implemented as part of a Professional Services engagement, a migration to ZXTM will require additional planning and consultation with Zeus that is not described in this paper. If a situation such as this, or if you are in any doubt, we would recommend that you contact your account manager who will be happy to discuss the options available to you.

To get the most out of ZXTM, we would strongly recommend signing up for either the ZXTM Standard training course, or the ZXTM Advanced training course with certification. Please contact your account manager for further details.

If you have any comments on this paper, we would be delighted to hear from you.

