



Getting more from Server Virtualization

Building an Adaptive Computing Infrastructure with Virtualization and ZXTM

Zeus Technology Limited (UK)
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: www.zeus.com

Zeus Technology, Inc. (U.S.)
Suite 320 - 5201 Great America Parkway
Santa Clara
CA 95054
United States of America

Phone: 1-888-ZEUS-INC
Fax: (866) 628-7884
Email: info@zeus.com
Web: www.zeus.com



Contents

Summary	5
Server Consolidation	5
Adaptive Computing – the next step	5
It shouldn't be like this.....	6
Virtualization is the new order	6
Adaptive Computing	7
Virtualization	7
Management	7
Monitoring.....	8
Traffic Management.....	8
Introducing ZXTM	8
Managing Traffic	8
Monitoring.....	8
Integration	8
Building an Adaptive Computing architecture	9
Setting the scene	9
Adding reliability and scalability	10
Example 1: Fault Tolerance	11
A self-healing application	11
Example 2: Resource Scheduling	11
CPU resource scheduling is ineffective.....	11
ZXTM's Service Level Monitoring Capability.....	12
Example 3: Rate Shaping	12
Rate Shaping	12
Example 4: Preparing for VMotion.....	13
Connection Draining with ZXTM.....	13
Example 5: Application Updates	13
Giving application deployment control to the application team	13
Clone, Test, Deploy, Migrate and Reap.....	13
Conclusion	15
Further Reading	15

Are you getting the most from your virtualized infrastructure?

What comes **after server consolidation**?

How can your applications **self-scale**?

Keep within **Service Level Agreements**?

Restart themselves when they fail?

How can you **build and deploy** patches and new application versions –

without any downtime?

An **Adaptive Computing** design uses virtualization, monitoring and traffic management together to build a new environment for your applications – an environment that works with the applications to ensure that they meet the needs of your business.

Summary

Server Virtualization has dramatically changed the landscape in the datacenter. Organizations are consolidating workloads from underutilized servers and are seeing large reductions in datacenter space, power, cooling and administration.

However, server consolidation is just the beginning of what can be achieved using virtualization technologies. Virtualization unleashes applications and compute workloads, breaking the ties that hold them to physical servers. This new-found freedom makes possible an entirely new datacenter architecture where the hardware serves the applications and the applications serve the business, rather than the other way round.

Server Consolidation

Server Consolidation was enabled by virtualization technologies like VMware's Virtual Infrastructure 3. Servers running at 10-15% utilization were commonplace and a significant waste of valuable resources.

The most common driver for the adoption of virtualization was a desire to address these inefficiencies. Large numbers of server workloads could be consolidated onto a small number of powerful, reliable servers, running at higher utilizations. The availability of low-cost multi-core processors makes this a very cost effective proposition.

It's not just server applications that can be virtualized in this way. Many hardware appliances – load balancers, firewalls, security gateways, spam filters, etc - can be replaced by equally capable virtual appliances. VMware's Virtual Appliance program encourages this with a stringent Virtual Appliance certification program, where high quality virtual appliances like Zeus' ZXTM product are available through the Virtual Appliance Marketplace¹.

Adaptive Computing – the next step

The potential of server virtualization goes far further than simply the consolidation of server workloads.

Virtualized servers are much more flexible, mobile and efficient than their physically-bound predecessors. Perhaps the biggest change is that a server workload no longer needs to be regarded as a long-lived, expensive resource, tied to and limited by the hardware it is running on. Instead, hardware is simply a scalable resource that the compute fabric uses, and servers are completely disposable, created, restarted and destroyed on demand.

No longer does the business need to be enslaved by the capabilities of its applications. When applications can be deployed, resourced and scaled on demand, the workload running in the datacenter can be adapted to the needs of the business, rather than the other way round.

¹ <http://www.vmware.com/appliances/>

It shouldn't be like this...

Imagine a rather old-fashioned business operation, restricted in how it can use its human resources by shortsighted working practices. Each worker could only be employed to do one specific task, even if the task did not merit a full-time position. Sickness and holiday cover was provided for each employee on a full-time basis. There was no reason to employ skilled workers with multiple talents because they could not be used to their full potential. It was so difficult to change working practices that it was easier to employ new staff than retrain existing ones.

This picture may seem implausibly antiquated, but it illustrates the limitations and inefficiencies that current datacenter architectures impose on business applications today:

- Businesses wish to cut costs, but the one-workload-per-server model is wasteful and inefficient;
- Businesses wish to be agile, but the need to tie server workloads to physical hardware makes change a slow and expensive proposition;
- Businesses wish to invest in hardware wisely, but differing workloads have different requirements. A powerful multi-core server may be more cost efficient but is a poor investment if few of the current workloads merit it;
- Businesses wish to deploy redundant servers to maintain business continuity during planned or unplanned downtime, but this redundancy multiplies the inefficiencies.

Virtualization is the new order

Virtualization decouples the server workload from the physical server and allows physical servers to run multiple workloads when resources allow. In terms of human resources, each worker can perform multiple tasks during their working hours and resources can be quickly redistributed to where they are needed most.

An **Adaptive Computing** architecture uses this metaphor. It redistributes workloads across the virtualized compute fabric on demand, driven by service level agreements and business requirements to serve the needs of the business.

Adaptive Computing

An Adaptive Computing architecture can be used with server applications that are made up of multiple components. This includes a wide range of enterprise applications; multi-tiered services may be comprised of web server, application server and database tiers, and SOA (Service Oriented Architecture) applications may be made up of tens or hundreds of discrete SOA components.

In a traditional, server-based environment, each service tier or group of SOA components is likely to be deployed on a dedicated physical server. Each tier or component communicates with the next using the network, most commonly using HTTP or an HTTP-based protocol such as SOAP.

Adaptive Computing breaks the ties with the physical servers making the application mobile, responsive and much easier to manage. There are four cornerstones to an Adaptive Computing architecture:

1. **Virtualization** of the workloads
2. **Management** of the virtual machines
3. **Monitoring** of correct operation and SLA conformance
4. **Traffic Management** between scalable and reliable components

Virtualization

The first cornerstone of Adaptive Computing architectures is the virtualization layer. Virtualization makes it possible to deploy and resource server workloads flexibly and with minimal delay.

Server Virtualization, such as VMware's Virtual Infrastructure 3, has made adaptive computing possible, but other virtualization technologies can be used to equal effect. For example, applications can be broken into discrete SOA components, and these components deployed across a cluster of Java application servers.

Tools like VMware's VirtualCenter™ provide a central point to manage virtual machines. A SOAP-based API (or equivalent) makes it very easy to drive the virtual infrastructure from other applications.

Management

The second cornerstone is the management layer. This lightweight layer contains the business and application management logic that controls how applications are deployed, being aware of the application limits and interdependencies. Each application will have its own unique requirements. As there are no commonly accepted standards as to how these are specified and deployed, the management layer is often built using custom code that interfaces to the standard APIs of the other parts of the architecture.

Monitoring

The monitoring capability detects when applications are performing outside their desired service level (for example, poor response times) or if an application component has failed completely. It informs the management layer, which can then choose to take remedial action such as deploying a new service instance or creating a new virtual machine.

Traffic Management

The traffic management capability allocates tasks to the individual workers in the virtualized compute fabric. It makes application scaling possible by load-balancing traffic across virtual machines, applying session persistence and migrating users from one generation to another.

The Monitoring and Routing functions can be provided by a sophisticated application traffic manager like Zeus' ZXTM product.

Introducing ZXTM

ZXTM is a non-intrusive, software traffic manager that load balances network services across clusters of servers, both physical and virtual. It is the only traffic management product available as a virtual appliance, and can be run directly on virtualization platforms from VMware, Microsoft, Xen and VirtualIron.

Managing Traffic

ZXTM provides a wide range of functions to manage traffic, from content-based routing to bandwidth and rate control. It contains a uniquely powerful traffic inspection engine and a configuration language called TrafficScript™ that allows the end user to construct very sophisticated traffic management policies to meet their specific requirements.

Monitoring

ZXTM's Service Level Monitoring capability monitors the response time of server nodes and services, raising alerts and modifying traffic management logic if services begin to fall outside desired Service Level Agreement (SLA) standards. Custom health monitors can perform complex tests against server nodes, detecting application and network-level errors.

Integration

The standards-based SOAP Control API makes it easy to integrate ZXTM with other management systems, reconfiguring ZXTM when a new server node is added. The SNMP interface allows another system to monitor ZXTM's operating parameters, including service health and performance statistics. The pluggable alerting framework makes it possible for ZXTM to initiate a custom action when a critical event such as a server failure occurs.

The monitoring, traffic management and integration capabilities of ZXTM can be used with the reconfigurable virtualized compute resources to build an architecture that reacts and adapts to the needs of the applications it hosts.

Building an Adaptive Computing architecture

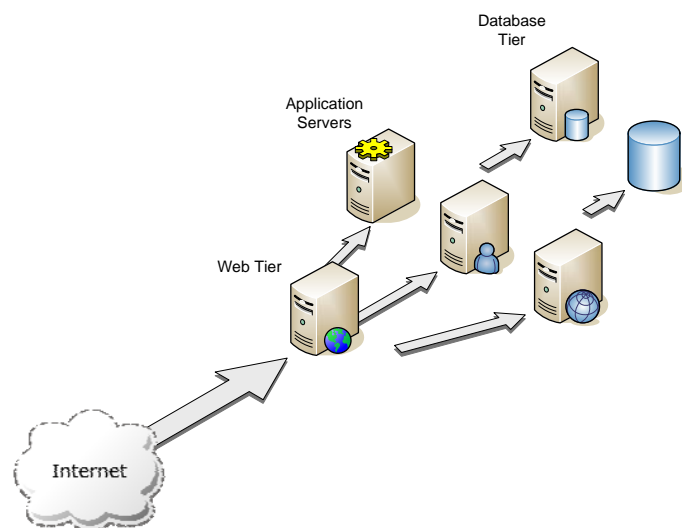
Setting the scene

Let's consider how we could take an existing application and construct an Adaptive Computing architecture around it.

Our application is comprised of:

- A web-based front end, running Apache or IIS. The front end hosts static pages and content and proxies application requests back to one of the application servers.
- Several application servers, authentication servers and custom gateways that perform different functions for the application.
- Several data sources – local databases and remote services.

Each component is running as a virtual machine on a virtualization platform like VMware VI3:

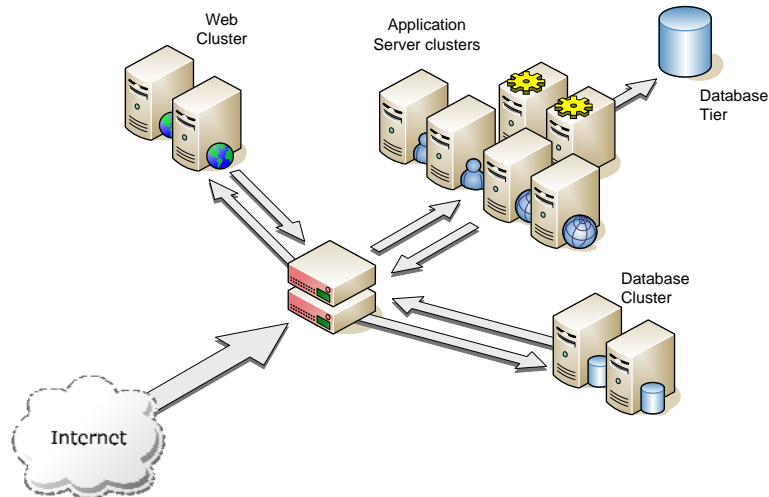


This configuration is fragile and un-scalable. If any one virtual machine were to fail or be overloaded with traffic, the entire application would fail. It is difficult to manage. Application upgrades cannot be easily tested and deployed without disrupting the live system.

Adding reliability and scalability

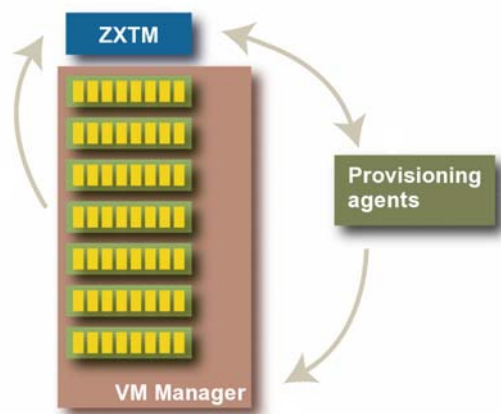
The first step in building the adaptive infrastructure is to use a traffic management device like ZXTM to load-balance and scale traffic within the infrastructure.

Where possible, each virtual machine should be cloned and a pair (or more) of machines run in its place. ZXTM can load-balance traffic across the cloned machines:



In this configuration, the ZXTM traffic manager sits 'in line', observing all transactions within the application, verifying the performance and correct operation of each tier and load-balancing transactions to the most appropriate virtual servers.

Web and Application servers in a compound application can always be scaled in this way. ZXTM's session persistence capabilities can be used if a series of transactions must be pinned to a particular web or application server. Active-passive clustering with replication can be used for components such as databases that are less easily scaled because of state sharing requirements.



ZXTM monitors and manages all application transactions, routing each network request to the most appropriate, fastest responding virtual machine.

ZXTM can detect when a service is performing poorly, or if a virtual machine is producing errors or has locked up.

When services need re-provisioning, ZXTM informs the appropriate agent which restarts or redeploys servers in the virtualized environment.

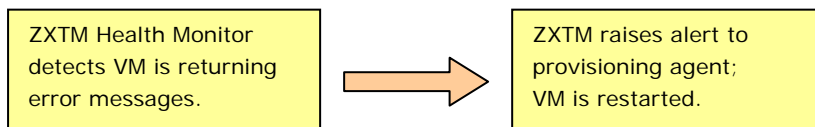
When a new server is started, the provisioning agent informs ZXTM so that it can load-balance traffic to it.

Example 1: Fault Tolerance

A single virtual machine can be easily duplicated to a pair of identical machines, with little additional resources required. In general, these virtual machines should be run on separate hardware in the virtualized cluster, using appropriate affinity rules. In this case, even if one of the hardware servers hosting the virtual machines were to fail, the application will be unaffected.

A self-healing application

Virtual machines may fail in a variety of ways, most of which are undetectable by hardware watchdogs and heartbeats. For example, an unattended server may run out of local disk space (perhaps exacerbated by a denial of service attack) and the server software may lock up or return error messages rather than valid responses. When faced with such a problem, the quickest resolution is generally to reboot the server or (preferably) restart it from a trusted snapshot.



The customizable health monitors in ZXTM are able to detect a wide range of application errors, whether the application has hung or is returning invalid responses. When an error is detected, ZXTM will stop routing transactions to the failed machine until the monitors detect that it is operating correctly again. ZXTM raises an alert, and you can plug into the alerting framework to invoke custom actions when a server fails.

For example, if an administrator desired that a node is restarted from a snapshot when it failed, he could provide a script or application that communicated with VMware's VirtualCenter™ management software using the management API to invoke this action.

In another case, it may be sufficient to provide a script that logs on to the failed server and restarts the software.

Example 2: Resource Scheduling

CPU resource scheduling is ineffective

Virtual machines run on shared servers and compete for resources – CPU, Memory and Network bandwidth. To prevent a runaway virtual machine from impacting the performance of other virtual machines, resource limits can be applied, but these are often ineffective.

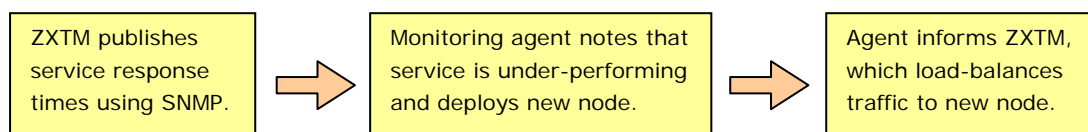
CPU resource allocations can be split between virtual machines, but unless these are over-allocated, you'll never achieve good hardware utilization. On the other hand, over-allocation carries the real risk that a small number of busy virtual machines will use all the available resources.

Furthermore, static CPU restrictions have very little bearing on the performance of the application. End users measure application performance in terms of response time, and organizations will wish to impose Service Level Agreements (SLAs) to define acceptable response times for each application or component.

ZXTM's Service Level Monitoring Capability

ZXTM's Service Level Monitoring capability monitors the response times for each service, and measures the service's performance against a predefined SLA target.

Response times and SLA conformance statistics are published via SNMP, so an external agent could monitor these figures and re-resource services on the fly. Alternatively, ZXTM will raise an alert if a service falls outside of an SLA target. This alert could initiate a custom provisioning action, such as re-balancing resource allocations within the virtual machine manager, or deploying a new virtual machine and informing ZXTM (via the SOAP-based Control API) to load-balance traffic to that new machine.



So, just as a traditional application may fork a new worker thread or process when it becomes overloaded, with ZXTM you can fork a new virtual machine on demand.

Example 3: Rate Shaping

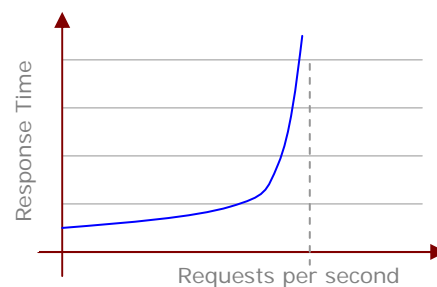
SLA-based resource balancing goes a long way towards ensuring that all applications receive the resources that they need, but an organization must be ready for exceptional situations where there aren't sufficient resources to go round.

Consider the case where a virtual infrastructure is managing several applications. One of these applications comes under a denial of service attack, where users unintentionally or maliciously flood it with large numbers of requests. Attempts to drag it back to within its SLA by stealing resources from other applications causes all applications to suffer as there are not enough resources for all the applications.

Rate Shaping

In this case, it's necessary to use a 'brake' that limits the number of transactions that the application is asked to perform. ZXTM's rate shaping functionality can be used to prioritize and limit the number of requests to the application. This has the effect that the application operates at peak efficiency within its resource allocation and does not become overwhelmed.

Rate shaping limits can be combined with Service Level Monitoring to build dynamic, adaptable traffic management policies that are only applied when service performance falls outside the desired SLA.



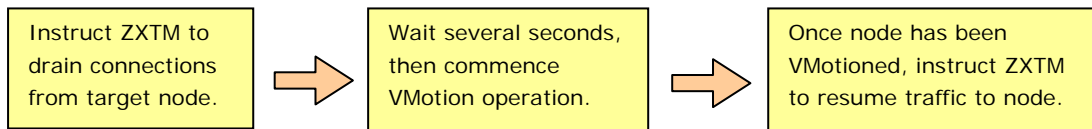
Towards a critical request rate, performance degrades asymptotically. Beyond this rate, the application is unusable.

Example 4: Preparing for VMotion

A virtual machine may become unavailable during a short period of disruption. For example, if it is to be migrated from one VMware ESX host to another, network connections may stall and time out. Additionally, the VMotion process is hampered when the source VM is busy with large quantities of memory writes, which is common on a busy web or application server.

Connection Draining with ZXTM

ZXTM's Connection Draining capability is used to take a machine out of service in a managed way. When a machine's connections are drained, ZXTM does not route any new requests to the machine; only requests in already established sessions are routed to it. Once these sessions complete, the machine can be safely removed from service without any disruption.



A virtual machine may be prepared for VMotion by draining connections from it for several seconds beforehand. This will minimize the change of dropped transactions and speed up the VMotion process.

Example 5: Application Updates

When applications are tied to physical servers, upgrades and server maintenance are problematic. Invariably, there is downtime when a live operating system is patched or live applications are upgraded, and there is a risk of serious disruption if the upgrade of the live system fails.

Giving application deployment control to the application team

A process like the one described in this section was used by a major Telco who suffered several hours of downtime each time they upgraded an application on their production servers. Using ZXTM, they were able to upload new application versions to their live servers, test them in the live environment and then migrate users smoothly from the old generation to the new one without interrupting any user sessions.

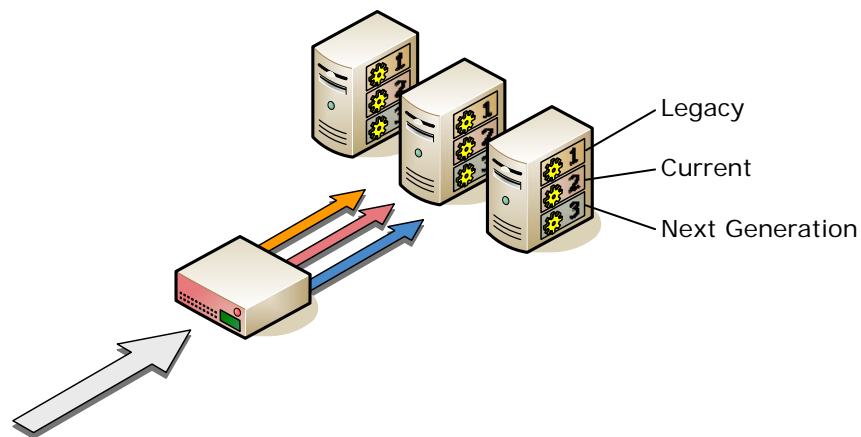
They reported that this gave full application deployment control back to the application support team, with no involvement of network teams. The application team increased their control of application deployment operations, increased their deployment speed, and improved efficiency by bypassing established procedural barriers.

Clone, Test, Deploy, Migrate and Reap

In virtualized compute architectures, virtual machines are not tied to individual physical machines. It's entirely possible to consider running two or more generations of a virtual machine in parallel.

With ZXTM, you can manage traffic to these virtual machines and achieve a seamless application upgrade process with absolutely no downtime. You can run legacy, current and

next generation clusters of virtual machines together and ZXTM can load-balance each users' traffic to the correct cluster:



Multiple different generations of an application can coexist, and requests are smoothly migrated as the transaction sessions complete.

Imagine that you had a cluster of three virtual machines running an application server. You need to apply an operating system patch, or make some updates to the application server configuration.

The virtualization layer and ZXTM can assist you by enabling the following process:

- **Clone:** Clone one of the virtual machines and designate it the 'development' machine. You now have four virtual machines running; three active and one development.
- **Test:** Using ZXTM, route traffic from the development staff to the new virtual machine; all other traffic continues to be routed to the current, active machines. The development staff can make changes to their virtual machine and test it on the live system without any risk of disrupting current users of the application.
- **Deploy:** Once the new virtual machine has been fully tested and is ready for live traffic, clone it so there are three virtual machines ready to accept user traffic.
- **Migrate:** Keep users pinned to the original cluster while their session is valid. Send new users to the new active cluster; old users are routed to the new cluster when they access the application after their session has expired.
- **Reap:** Once the original cluster is no longer in use, reap the virtual machines.

If the new version fails for any reason, you can easily roll back to the previous version without any difficulty – something that would not be possible if you upgraded a physical server in place.

ZXTM can identify development staff by their source IP address, login details or other criteria. It can pin end users to the correct generation of the application by automatically tagging them with a non-invasive cookie that lasts for the duration of their browser session.

Conclusion

Virtualization is a long-term investment. Consolidation is a short-term gain, and Adaptive Computing is big prize to follow.

You can realize the benefits of Adaptive Computing when you understand the needs of your applications, the needs of your business and match the two together. A traffic management device like Zeus' ZXTM Virtual Appliance is a key component of the Adaptive Computing architecture, managing application traffic and monitoring the health and performance of the applications.

Ultimately, this architecture can deliver an SLA-driven computing fabric that meets the needs of your business by managing the needs of your applications.

Further Reading

ZXTM is a sophisticated, programmable Application Traffic Manager. It is used in virtual and physical environments to accelerate, make reliable, secure and manage network-based applications.

Useful documents for more information are:

- www.zeus.com/library

Traffic Valuation and Prioritization white paper

Service Delivery Controller fact sheet

Building a Service Oriented Network white paper

- knowledgehub.zeus.com

The ZXTM KnowledgeHub (<http://knowledgehub.zeus.com>) is the core Developers' resource for ZXTM, containing practical examples, technical documentation and code samples that illustrate the capabilities of the ZXTM traffic manager.

Copyright

© Zeus Technology Limited 2008. Copyright in this document belongs to Zeus Technology Limited. All rights are reserved.

Trademarks

Zeus Technology, the Zeus logo, Zeus Web Server, Zeus Load Balancer, Zeus Extensible Traffic Manager, ZXTM, ZXTM Global Load Balancer, ZXTM Virtual Desktop Broker and associated logos and abbreviations, TrafficScript, TrafficCluster 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

Zeus Technology UK:	+44 1223 525000
Zeus Technology US:	1-888-ZEUS-INC <i>or</i> +1 650 965 4627
Fax:	+44 1223 525100

By Post or in Person

Zeus Technology Limited	Zeus Technology
The Jeffreys Building	1955 Landings Drive
Cowley Road	Mountain View
Cambridge CB4 0WS	CA 94043
United Kingdom	United States

www.zeus.com and knowledgehub.zeus.com

Our web site contains a wealth of information on our products, services and solutions, as well as customer case studies and press information. For more information, please visit www.zeus.com and knowledgehub.zeus.com.