appzero

Lab Test: AppZero Turbo-Charges Server Consolidation

March, 2009

Table of Contents

Abstract	1
Virtualization & Performance	1
Test Design	2
Test Procedures	3
Web Server Details	4
Database Server	4
Client Programs	4
Network and Throughput	4
Test Results	5
Conclusion: Servers and Server Applications	7
Technical Details	7

Lab Test: Server Consolidation

Abstract:

Server consolidation continues to be one of the strongest drivers of virtualization adoption, and one of its most significant pay-offs. However, when it comes to provisioning server-based applications, AppZero's Virtual Application Appliance (VAA) approach offers dramatic benefits in agility, cost, and performance -- whether deploying to physical or virtual machines.

This paper examines the impact of application virtualization technology on overall system performance. It documents a simple performance evaluation of application virtualization using AppZero virtual application appliance (VAA) compared to a VMware ESX-based system using virtual machines (VM).

Both approaches were compared to simultaneously running multiple applications natively on a Windows-based server. A series of tests were run to look at the performance characteristics of the systems based on a simple Web-based test application.

The test quantitatively demonstrates AppZero VAA performance at near-native level. Equally effective in physical, virtual, and cloud environments, AppZero's instant server-side application provisioning is a natural complement to server virtualization – driving higher server density and consolidation.

Virtualization and Performance

Recent trends in computer technology enable IT departments to deploy the highest performance systems with a small cost premium compared with the cost of software and ongoing operations. Technologies such as blade servers and storage arrays, with low power consumption, are changing the hardware footprint for many organizations. Virtualization is gaining rapid acceptance within IT as a means to achieve greater efficiency in both the use of server hardware and in ongoing operational support. There are two key benefits the technology provides:

- **Flexibility** – Virtualized systems may be moved among physical computers as needed, maximizing the use of high performance hardware resources and allowing for changes as needed to the physical infrastructure supporting the systems.
- **Isolation** – Applications and other software systems may be packaged and deployed in a modular fashion in complete isolation from each other, simplifying the overall lifecycle management within an organization.

Most virtualization technology today revolves around the use of a *hypervisor*-based approach, in which the entire operating system (OS) running an application is hosted as a guest in a dedicated virtual environment. While this approach completely isolates one OS instance from another, the amount of overhead involved in running many instances can be significant. That overhead can directly slow overall system performance. In addition, the proliferation of virtual machines and OS instances can create a virtual machine sprawl that requires almost as much work to manage as physical machines.

By contrast, AppZero's approach to application virtualization completely decouples applications from an OS, and still allows the configured application to be packaged and managed as a single construct – the virtual application appliance (VAA). In AppZero's VAA, the code base, dependent libraries, services, data, and other components of an application may be managed together and deployed as needed without an installation process or other system setup. This approach provides the benefits of flexibility and isolation, but without the overhead of a complete OS that is required in hypervisor-based products.

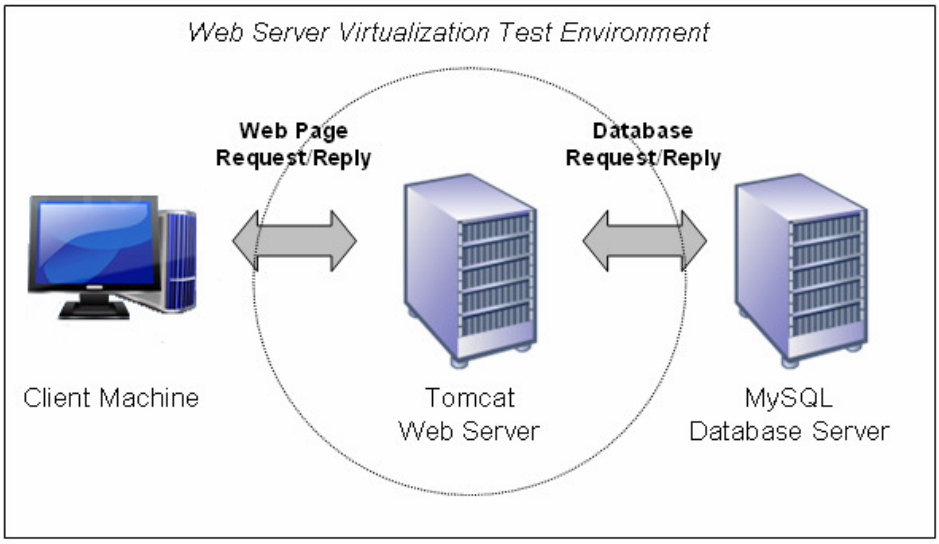
Test Design

The test was designed to simply illustrate performance and scaling in a context that is representative of a typical IT application.

The design was based on a simple three-tier Web-based application using a Web server and database server to supply data to multiple clients across a network. During the tests, the Web server was run both virtualized and as a native application to compare virtualization technologies and relate them to the non-virtual approach.

The number of Web servers run was varied from test to test to compare how the different technologies behave when multiple applications are deployed on the same physical machine. The clients and database server were each run on separate machines and were not virtualized.

The design is analogous to a Web server "farm", which is a common use for virtualization where multiple independent virtualized Web sites are run simultaneously on the same server host machine. The test was also designed to provide maximum load on the Web server, while not putting significant load on the client machine or database server. The Web server therefore governed the performance of the system as whole, isolating the direct impact the various virtual configurations had on system throughput.

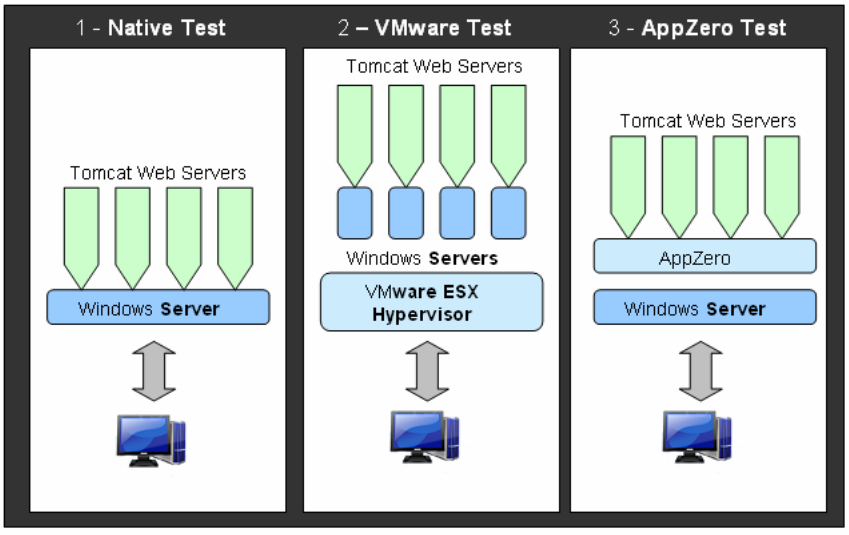


Test Procedures

Three sets of tests were run.

- First Set - The Tomcat Web server was run as a native application on Windows Server 2003 SP2. The number of Tomcat servers was increased for each test to simulate multiple applications running on the same physical host.
- Second Set – VMware ESX was used to virtualize each Tomcat server within a single virtual machine running Windows Server 2003 SP2. The number of running virtual machines was increased as the means to raise the number of Web servers for the test.
- Third Set - Tomcat servers were virtualized within an AppZero VAA and run on Windows Server 2003 SP2. The number of VAAs run was increased to run additional Web servers.

Each test was run at least twice for 20 minutes in duration to insure the stability of the systems and also to validate that repeatable results were obtained.



The number of pages delivered to the client machines was counted to determine the overall throughput of the Web server. Each test setup was given a warm up period of twenty minutes as well to insure items such as data caching and just in time compilation had occurred prior to the test runs.

Web Server Details

The Tomcat Web server was used for all testing. A servlet was run in Tomcat to accept client requests and retrieve 100 rows of data from the MySQL database server. The database results returned were used by the servlet to format a response to the client formatted as an XML page. The size of the XML page returned was 25K. The database request and page formatting resulted in full CPU utilization during the test, resulting in a significantly higher processing load on this server as compared with the other machines

The Web server machine is based on the latest quad-core Intel XEON server processor and contains enough memory to eliminate any memory paging or memory contention among Web servers or the Windows operating system.

Database Server

The database server is based on dual XEON processors running the MySQL database. After startup, the database table being requested was cached in memory, eliminating the need to access the disk and reducing the load on this server. During the tests, this machine never exceeded 10% of its available CPU and was lightly loaded compared with the Web server machine

Client Programs

The client machine contained 16 identical page request programs simulating 16 users. Each program retrieved pages as quickly as possible from the Web server and did no processing with the data to insure maximum throughput. Each program accessed each of the available Web servers in round-robin fashion to create an even client load.

The client machine used a dual-core E6600 Core 2 Intel processor. The number of client programs running was sufficient to fully load the Web server in all tests. The client machine never exceeded 30% CPU utilization in any test.

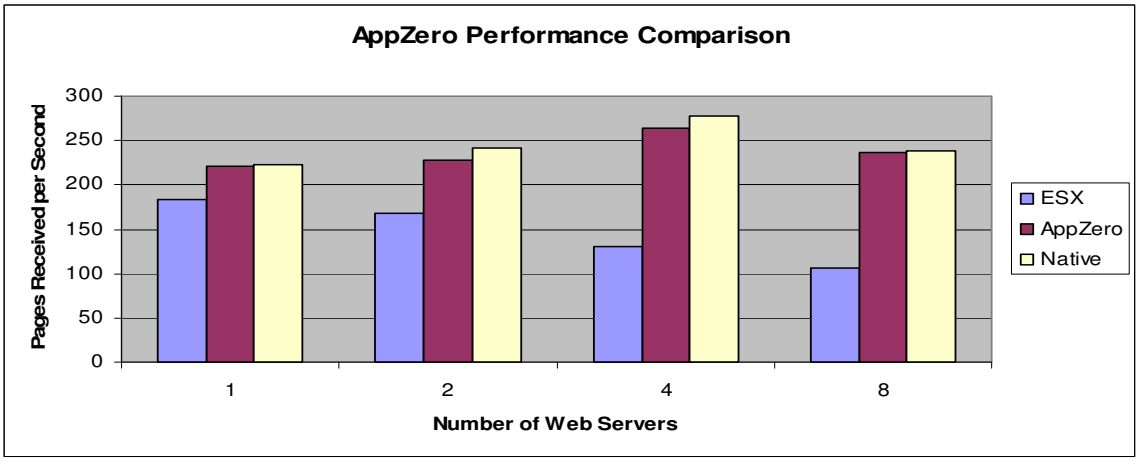
Network and Throughput

Gigabit Ethernet was used for all network connections. A Gigabit switch connected each machine within an isolated network. Simple additional testing was performed to insure that the network imposed no slowdown on the system. Both the client connections

and database connections were additionally evaluated in standalone tests to insure that it was possible to run both the clients and database at much faster rates than those achieved by the Web servers in the virtualized Web server testing.

Test Results

For each Web server configuration, results were collected for an increasing number of running Tomcat instances. The number of Web servers was increased from 1 to 2, 4, and 8. As the number of servers was increased, the impact of the different virtualization technologies became apparent. The ability to efficiently use hardware resources running multiple Web applications was shown to vary significantly based on the technique chosen.



The results above show AppZero at near native level performance as the number of Web servers was increased. In contrast, the VMware ESX performance continually degraded with additional Web servers.

Hardware Utilization – The ability to fully utilize hardware resources is one of the key values of virtualization. Increasing the number of applications running on a physical machine allows hardware to be used to its fullest.

Web Servers	AppZero vs Native	ESX vs Native
1	-.1%	-18%
2	-5.7%	-31%
4	-4.7%	-53.1%
8	-1.2%	-55.8%

The test showed that to support throughput equivalent to AppZero, VMware ESX will need roughly twice the hardware resources for an application similar to that tested as the number of applications is increased.

I/O Throughput – In addition to slowdowns in handling multiple virtual machines, the ESX environment showed a performance slowdown of 18% for even a single Web server as shown in the first entry above.

This slowdown is the result of input and output (I/O) operations for the Web server. In general, hypervisor-based virtualization such as ESX introduces two layers of I/O drivers through which data must be handled: one for the Hypervisor, and one for the guest operating system.

Peak Performance -- The peak performance achieved by AppZero and the native processes occurred when four Web servers were run. This performance is due to the full utilization of the quad-core processor where each Web server essentially has a dedicated processor core. However ESX was not able to take advantage of this configuration and continued to degrade as virtual machines were added, ultimately running less than half the throughput of AppZero.

Memory Utilization – The need for ESX to load an entire operating system for each application introduces significant memory overhead. At a minimum, the recommended memory for the guest operating system must be allocated along with the memory required for the application.

With AppZero, code modules may be loaded to support an application, but in general this is significantly smaller compared with an entire operating system. In the tests, ESX was allocated 320MB of memory for each virtual machine to allow for the Tomcat Web server to run in 60MB without running out of memory. AppZero required only the 60MB alone on top of the native OS.

Total Memory Use for Eight Web Servers

VMware ESX -	2560MB
AppZero -	480MB

The memory required will vary depending on the application deployed.

Conclusion: Servers and Server Applications

Hypervisor virtualization technology has proven its worth in accomplishing its design purpose: server consolidation. But server-based application virtualization is a different matter – exactly the one that AppZero was architected to address.

The testing described above demonstrates that the choice of application virtualization technology can have a significant impact on the ability to meet performance goals on new hardware platforms. The test shows that application virtualization using AppZero VAA performs at near native rates – orders of magnitude more efficient than the hypervisor approach tested.

When multiple applications are deployed in a hypervisor-based system, many more systems may be required to achieve the desired level of performance for applications. This requirement increases both software license costs and operational expenses when using hypervisor-based systems.

In addition to superior performance, AppZero reduces costs and simplifies ongoing management of applications by eliminating the need to manage an operating system for each virtual instance.

AppZero provides the flexibility and isolation required to fully utilize the high performance systems being deployed with IT departments today – whether physical or virtual, in the datacenter or cloud, in any combination.

Technical Detail

Web Server

Hardware:

Intel XEON 5450 Quad-core 3.0 GHz 12MB Cache 1.333 GHz FSB

Intel S5000PSL motherboard 4GB FBDIMM Memory

Intel I/O Acceleration Technology

Intel Virtualization Technology extensions enabled

500GB Western Digital SATA (1 dedicated per environment)

Gigabit Ethernet onboard

Software:

Windows Server 2003 Enterprise SP2

VMware ESX Server 3i 3.5.0 build 110271

Apache Tomcat 5.5.26

Java 1.6.0_06-b02

Database Server

Hardware:

Dual Intel XEON 3.0Ghz HT 2MB Cache 800Mhz FSB

Supermicro X5DPE motherboard 2GB DDR Memory

500GB Western Digital SATA

Gigabit Ethernet

Software:

Windows Server 2003 Enterprise SP2

MySQL 5.0.67 Community Edition

Client Machine

Hardware:

Intel Core 2 Duo E6600 2.4GHz 4MB Cache 1.066 GHz FSB

Intel DP965LT motherboard 2GB

1TB Western Digital SATA

Gigabit Ethernet onboard

Software:

Windows XP SP2

Java 1.6.0_06-b02

AppZero
300 Brickstone Square, Suite 201
Andover, MA 01810
+1-617-820-5126
+1-866-444-6670

750 Palladium Dr., Suite 210
Ottawa, ON K2V 1C7
+1-613-254-5432
+1-866-444-6670

info@appzero.com
www.appzero.com