



Howard Operowsky
Karen Lawrence

IBM eX5 Technology and WebSphere Produce High-Performance Websites

Introduction to web-facing applications and accessibility

The world is becoming more instrumented, interconnected, and intelligent every day. *IBM® Smarter Commerce™* enables new business processes that help companies to buy, market, sell, and service their products and services. *IBM Smarter Cities®* are combining data from multiple systems to improve education, transportation, power management, and the delivery of social services. *IBM Smarter Healthcare®* turns data into clinical and business insights, and it instruments processes with those insights in real time for point-of-care decisions and productivity. For these systems to be useful, they must be accessible. Today, accessibility generally means availability on the web by using web-facing applications.

Application servers are the middleware that help enterprises to make their applications accessible on the Internet. However, application servers do much more than make applications available; they also do these tasks:

- ▶ Extend the capabilities of the operating system in managing applications.
- ▶ Optimize the system resources used by web-facing applications.
- ▶ Manage the connections between web-facing applications and external resources, such as databases, networks, and other applications.
- ▶ Provide quality-of-service features, such as availability, security, management, performance, and scalability to web-facing applications.

Clearly, a strong application server is absolutely essential for the smooth operation of your enterprise web-facing applications.

IBM provides servers and systems that enhance accessibility in small-, medium-, and enterprise-level businesses. The IBM Strategy and Technology Lab (STL) conducted studies with IBM WebSphere® and eX5 products to determine the improvements these products can offer.

First, we studied WebSphere Application Server, which provides an impressive combination of function and performance. The success of WebSphere Application Server in the marketplace is evidenced by the following statistics collected by IBM:

- ▶ More than 19,000 WebSphere Application Server customers exist, and more than 300 IBM products include WebSphere Application Server as a key enabling component.
- ▶ WebSphere is used in 90% of the 100 largest companies in the world.
- ▶ More than 4,330 IBM Business Partner applications are built on WebSphere Application Server and are listed in the IBM Global Solutions Directory.
- ▶ More than 80,000 WebSphere family feature packs have been downloaded to extend WebSphere systems.
- ▶ More than 9,000 independent software vendors write applications to run on WebSphere.

Second, we studied MAX5, a memory expansion unit for eX5 processors, which improves performance. Application consolidation is improved with MAX5, and so it is useful in virtual environments.

Third, we studied WebSphere eXtreme Scale, a product designed to improve application performance through the use of a scalable data grid for storing objects in computer memory rather than disks. This grid increases application performance by reducing the latency in accessing data.

This IBM Redpaper™ publication describes the studies done by the STL and describes the results with a focus on the accessibility needs of your business.

This publication is directed to IT professionals and decision-makers, such as CEOs, CIOs, CFOs, IBM clients, IBM Business Partners, information architects, business intelligence administrators, and database administrators.

WebSphere Application Server and eX5 have the power to support web-facing applications

WebSphere Application Server is optimized to take advantage of the power of eX5 multi-core processors. The WebSphere development team is constantly seeking to improve the performance of WebSphere Application Server.

WebSphere Application Server maximizes the efficiency of eX5 multi-core processors

Multiprocessors allow WebSphere Application Server to run multiple applications simultaneously. To ensure that applications do not interfere with each other, WebSphere Application Server uses a mechanism called *locks*. When an application locks a resource, all other applications that need the resource must wait until the lock is released before gaining access, which impacts overall system performance. In WebSphere Application Server v8, the algorithms are improved, so that the applications use locks less frequently. This improvement reduces overall wait-time, thus improving system performance.

Other techniques exist for making sure that applications do not interfere with each other. These techniques are collectively called *synchronizations*. WebSphere Application Server algorithms were also improved so that less time was spent synchronizing application threads, thus freeing up time for the applications to run business functions.

As a result of these changes, WebSphere Application Server v8 is 20% more efficient than WebSphere Application Server v7 when executing DayTrader 2.0¹, an open source benchmark for application servers².

WebSphere Application Server, on eX5, was 44% faster than the competitor server

We carried out a number of tests to demonstrate the performance of WebSphere Application Server on a production-level System x3850 X5. *Performance* was measured both in terms of transactions per second and in the cost per transaction per second. In one test, we ran a banking benchmark with WebSphere Application Server on a production-level System x3850 X5, and we compared the performance with that of a competitor's application server on a competitor's production-level server. The measured test results are shown in Figure 1. Here, the IBM and competitor hardware were built using the same Intel processor and were configured identically. We tuned the systems for maximum throughput.

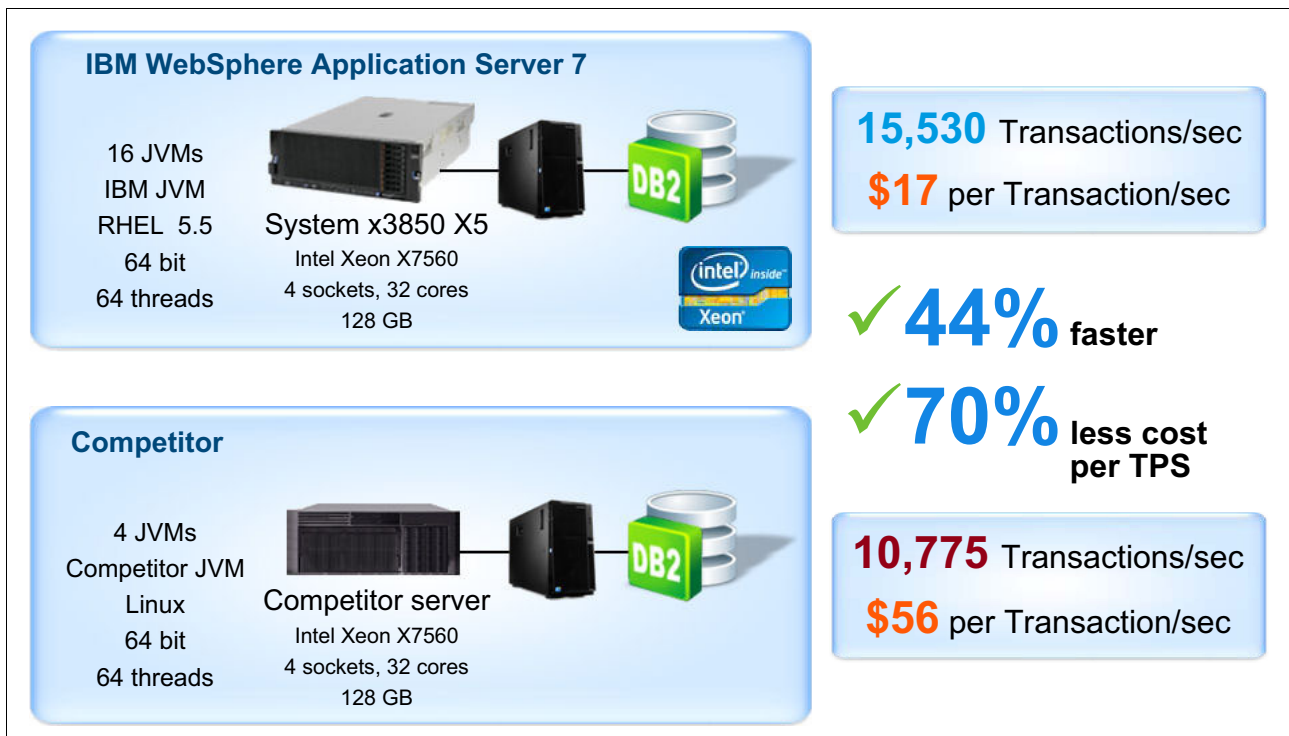


Figure 1 WebSphere Application Server on eX5 delivers more performance from 32 cores than competitor server

As you can see, the IBM system was 44% faster than the competitive platform, and the amortized cost of the IBM platform was 70% less than the amortized cost of the competitive platform.

¹ DayTrader - A More Complex Application:
<https://cwiki.apache.org/GMOxDOC22/daytrader-a-more-complex-application.html>

² Why IBM WebSphere Application Server V8.0?, IBM white paper, 2011,
<ftp://public.dhe.ibm.com/common/ssi/ecm/en/wsw14167usen/WSW14167USEN.PDF>

WebSphere Application Server can be clustered to handle heavier workloads

Many times, enterprises choose to run web-facing applications on a cluster of servers. Not only do clusters provide greater throughput than does a single server; they also provide better availability. If one of the servers in a cluster fails, the other servers ensure that the system remains active.

Two ways to create clustered systems exist. The traditional way is to run the same application natively, that is, non-virtually, on a collection of servers. With the growing use of blades and virtualization, there is another method to create clusters, as shown in Figure 2. In a blade system, the application can run in multiple virtual machines spread across the blades in the system. Not only does this approach, called *vertical stacking*, provide the same benefits as a traditional cluster, but it also allows for better workload density. Virtual machines can be moved between blades to maximize the utilization of each blade. See Figure 2 (in the figure, “WAS” represents WebSphere Application Server, and “IHS” represents IBM HTTP Server).

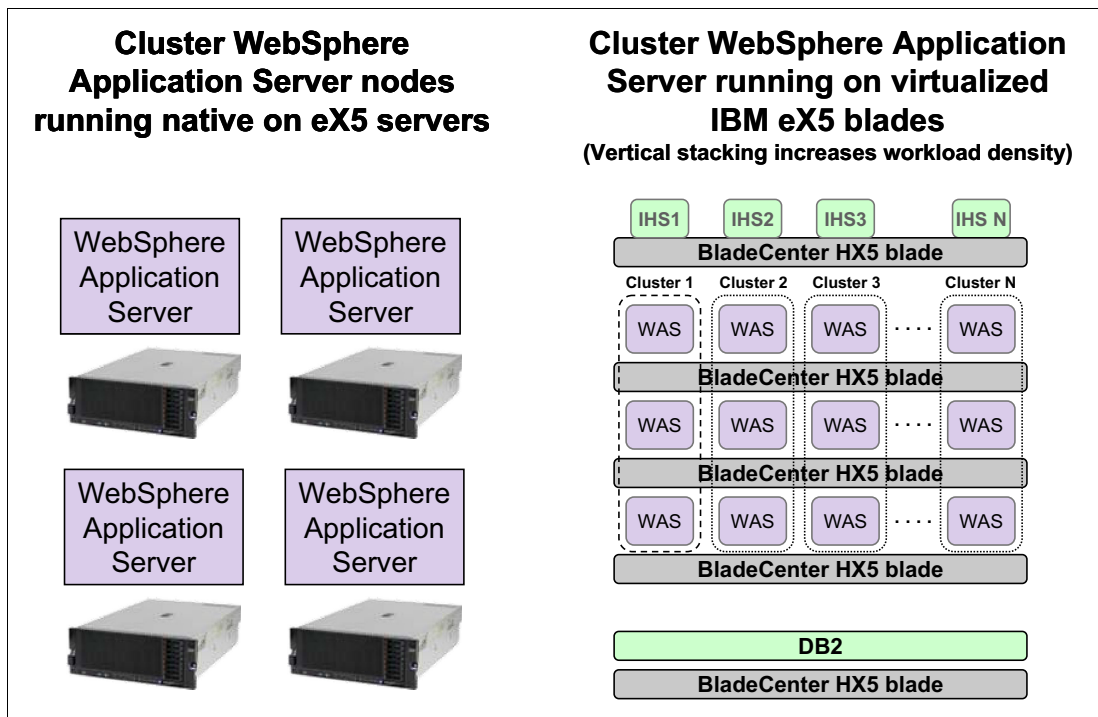


Figure 2 Creating a cluster with WebSphere eXtreme Scale

Consolidating applications with WebSphere on eX5

Now, we look at how you can consolidate WebSphere applications on eX5 servers to achieve the power you need to run your business at lower costs.

Underutilized dedicated servers

Historically, x86 servers tended to be dedicated to a single function. Experience has shown that this is wasteful. Servers are bought on the basis of peak workloads, with the result that they are underutilized. Tests have shown that Windows servers run at 5 - 10% utilization on average, with UNIX servers only slightly better at 15 - 20% average utilization, as shown in Figure 3. As we saw with vertical stacking, virtualization provides a solution to this problem.

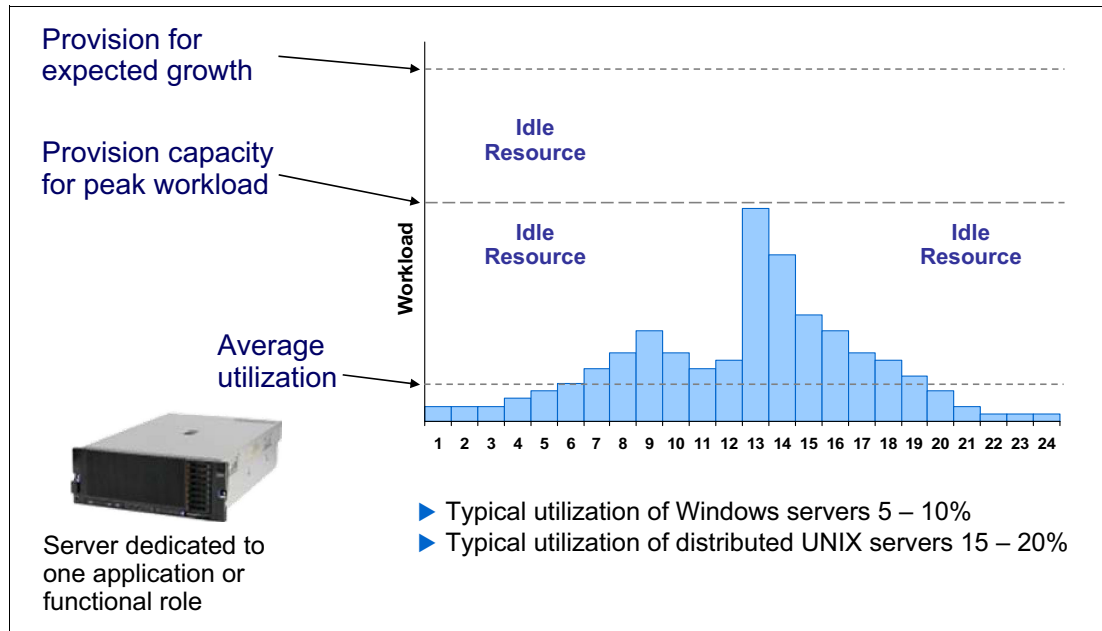


Figure 3 Underutilized dedicated servers

WebSphere and eX5 support applications of all sizes

Vertical stacking enables greater workload density by allowing virtual machines to be moved between blades to maximize CPU utilization. This approach applies equally well to applications that do not require the full power of eX5 servers. Using virtualization, these applications can be consolidated on a single eX5 server.

Consolidation reduces system costs in several ways:

- ▶ *Reduces software costs:* Software costs are based, in part, on the number of cores in the machine in which the software is installed. By removing applications from the underutilized servers and consolidating them on a single server, you can eliminate the underutilized servers, reducing the total number of cores in the enterprise system. The number of cores affects the cost of software, so lowering the number of cores reduces overall software and licensing costs.
- ▶ *Reduces maintenance costs:* With fewer servers, less space is required, which results in lower energy requirements. Also, fewer administrators are needed to manage the reduced number of physical machines.

Application consolidation is greatly facilitated by virtualization. Applications generally function as though they own all machine resources. Thus, it is difficult to install two such applications on the same operating system image. But virtualization allows such applications to be installed on the same server, continue to operate as before, and still not interfere with each other.

We looked at two forms of virtualization:

- ▶ *Java virtual machine (JVM):* WebSphere Application Server applications run in JVMs, helping to more easily consolidate them on a larger machine. When you install an application into WebSphere Application Server, the application will immediately execute in its own JVM, independent of the other applications running under WebSphere.
- ▶ *Hypervisor:* With a hypervisor, you can create many virtual machines on a single physical machine. Some hypervisors that are available for eX5 servers today are VMware ESXi, Microsoft Hyper-V, and KVM. Software running on a virtual machine behaves the same as it does on a physical machine. Applications that control all the resources in their machines can be installed in separate virtual machines and will run properly, even though they are sharing the same physical machine.

MAX5 improves consolidation

A new server technology available for the IBM eX5 class of servers is the MAX5 memory expansion unit. MAX5 is an IBM eX5 exclusive technology that adds memory to the server, independent of the processor. With no added processors, MAX5 can increase productivity and reduce costs in a way that other servers of this class cannot. This approach can be especially useful when consolidating servers in a virtual environment.

Physical memory can be a constraining factor

One problem with consolidating applications in virtual machines is that each application functions as though it has the same amount of physical memory that it had when it was running on a physical server. Thus, the combined demand of these applications can exceed the amount of physical memory installed on a server. The least expensive solution is to reduce the amount of memory each virtual machine can use; however, that way can generally affect application performance.

A better solution is to add additional physical memory to the server. This solution can work, but only to a point, because every server has a limit on the amount of physical memory that can be installed. When that limit is reached, the only solution, short of rewriting the applications to use less memory, is to buy additional servers. Then, some of the applications can be moved to the new machines. That, however, reduces the savings achievable through consolidation.

MAX5 increases server physical memory

IBM has addressed this problem with the MAX5 memory expansion unit. MAX5 increases the number of DIMMs that can be supported on eX5 computers, up to a total of 32. Using 32 GB DIMMS, MAX5 adds as much as 1 TB of memory to an existing eX5 server to support large, memory-constrained applications. Because MAX5 adds memory without the need for additional cores, it can reduce acquisition costs by 49%, largely because of reduced software costs. The details are listed in Table 1 on page 24.

Another way MAX5 can reduce costs is by using smaller-capacity DIMMs to meet memory requirements. For example, 16 GB DIMMs cost \$459 US; 32 GB DIMMs cost \$2,499 US. At these prices, populating one server with 1 TB would cost \$80,000 US by using 32 GB DIMMs, but only \$37,000 US by using 16 GB DIMMs and MAX5. These values are the US dollar list

prices in effect when the analysis was done (March, 2012) and might not represent current list prices. Also, prices can vary by country and will likely change over time as technology matures. Discuss your current environment with your IBM representative to determine how much you can save, at current prices, with MAX5.

How much does MAX5 improve virtualized environments?

We mentioned two types of virtualized environments (“WebSphere and eX5 support applications of all sizes” on page 5): hypervisors such as VMware ESXi and the JVMs provided by WebSphere. We conducted studies to demonstrate the advantages of MAX5 in both environments.

We ran the first test to demonstrate the benefits of MAX5 in a hypervisor-based environment. We used the industry standard, open Standard Performance Evaluation Corporation virtualization (SPECvirt) benchmark to measure the maximum performance of data center servers used for virtual server consolidation. Each unit of work, called a *tile*, is composed of virtual machines running applications that represent server virtualization, including web, mail, application, and database servers. The web server is particularly memory-intensive and was the primary focus of the test. The test consisted of gradually increasing the number of tiles that were executing until the total throughput of all tiles reached a maximum, or until workloads failed to meet some quality-of-service requirement.

For our study, we used a System x3690 X5 and the KVM hypervisor. Figure 4 describes our setup.

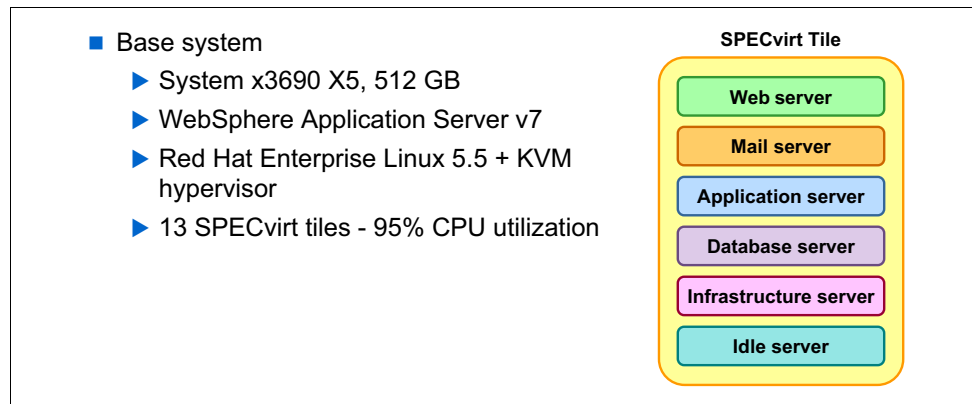


Figure 4 Setup for test to measure the effect of MAX5 in a virtualized environment

We obtained a baseline result, running SPECvirt on a System x3690 X5 with 512 GB RAM. Using the KVM hypervisor running on Red Hat Enterprise Linux 5.5, we achieved 95% utilization, executing 13 tiles. Each tile was six virtual machines, so we were actually executing 78 virtual machines on the system.

MAX5 is intended to improve virtual machine performance

Figure 5 demonstrates how the additional 256 GB of memory installed in the MAX5 can improve performance of the SPECvirt benchmark compared to the baseline result.

- Fewer processor cycles are spent on disk operations, freeing up 15% of processor utilization
- The freed up processor cycles could be used to add more tiles or virtual machines

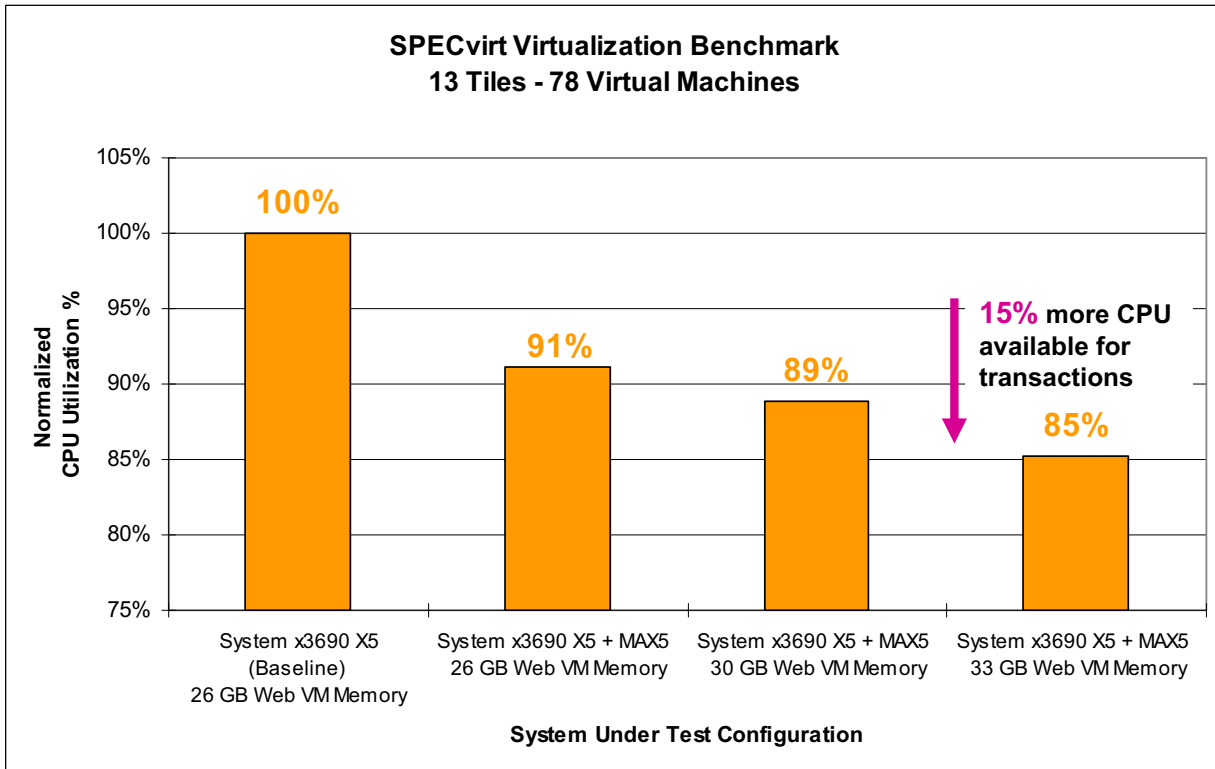


Figure 5 MAX5 is intended to improve virtual machine performance by 15%

To more easily compare the results, we first normalized the measured results of the tests so that the baseline test is shown as 100% CPU utilization in the left-most column in Figure 5. The second column shows that performance improved by 9%, just by adding MAX5 with 256 GB. This addition of MAX5 demonstrates that the hypervisor (KVM in our test) can use the additional memory to run more efficiently.

In “How much does MAX5 improve virtualized environments?” on page 7, we mentioned that web servers are memory-intensive. We looked at how the additional memory improved web server performance. In Figure 5, the two columns on the right show the measured performance improvement when memory was added to each of the virtual machines that was executing the web server. Thirteen tiles were executing in each case, so there were 13 web servers running on the system. In the two left columns, the web servers had 26 GB memory. When they were allocated, first 30 GB and then 33 GB, the measured system performance increased by 2% and 6%, respectively. Thus, MAX5 improved overall performance by 15% (from 100% CPU usage to 85% CPU usage), so that the CPU was 15% more available for transactions.

MAX5 enables more WebSphere Application Server workloads to be consolidated

JVM is the other form of virtualization mentioned in “WebSphere and eX5 support applications of all sizes” on page 5. Every WebSphere application runs within its own JVM.

In this test, we created a workload of four virtual machines. Each workload consisted of one deployment manager, one node agent, and two application servers running the open standard, DayTrader 2.0 benchmark³. Using a System x3850 X5 with 256 GB memory, we added multiple copies of this workload, and found that we were able to consolidate 85 workloads. After adding MAX5 with an additional 128 GB, we were able to consolidate 124 workloads. In this test, MAX5 improved system consolidation by 46% for an additional \$13,500 US. That amount translates into a savings of 30% per workload, based on the costs as indicated in Figure 6. The additional cost of \$13,500 US was computed by adding the cost of MAX5 (\$7,500 US) and the cost of 128 GB of memory (\$6,000 US). These list prices are from mid-2011 and might not represent current prices. Discuss your current environment with your IBM representative to determine how much you can save, at current prices, with MAX5.

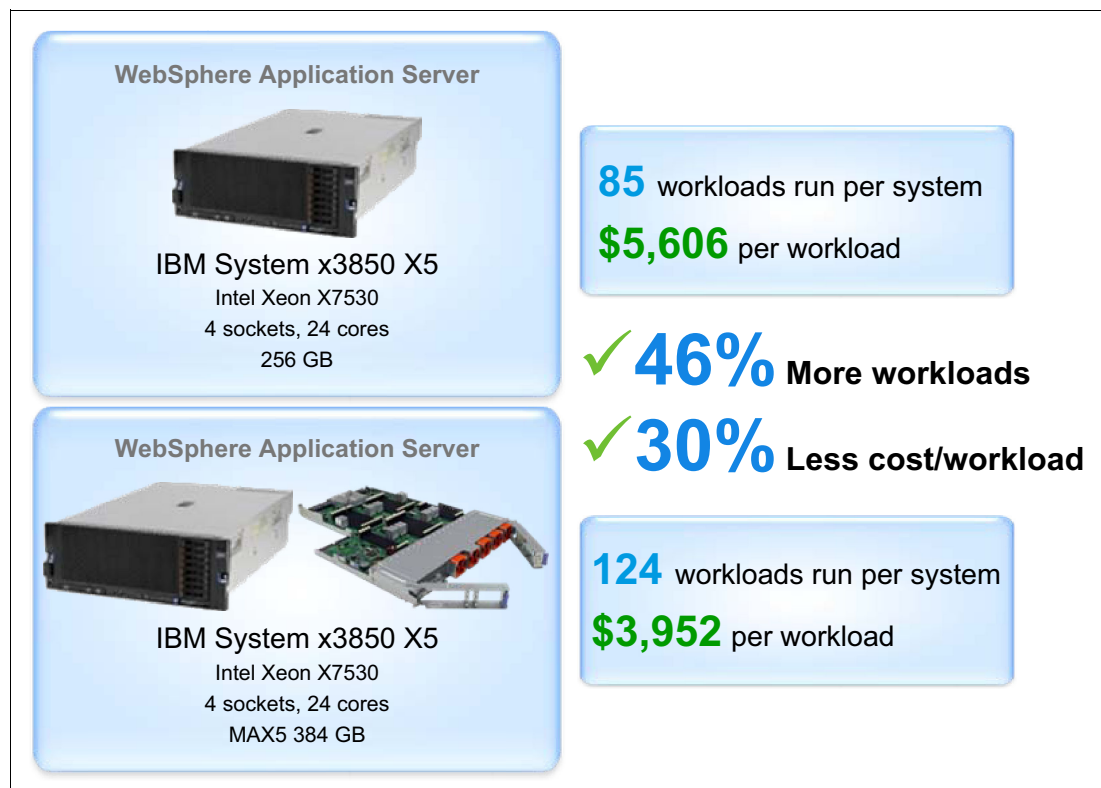


Figure 6 MAX5 enables more workload consolidation

IBM Workload Deployer accelerates time to value

Virtualization has done much to advance cloud computing. IBM Workload Deployer is a hardware appliance that uses virtualization to help users build private clouds easily and efficiently.

³ DayTrader - A More Complex Application:
<https://cwiki.apache.org/GMOxDOC22/daytrader-a-more-complex-application.html>

Virtual server lifecycle management is essential for environments with extensive virtualization

Several advantages of cloud computing are that users can acquire and provision their own virtual servers, and monitoring and billing of those virtual servers can be automated. With a private cloud, you can obtain the benefits of cloud computing but behind the safety of your firewall on your own private network.

IBM Workload Deployer accelerates time to value in eX5 virtualized environments

IBM Workload Deployer is a hardware appliance that provides access to IBM middleware virtual images and patterns to easily, quickly, and repeatedly create application environments. These environments can be deployed more securely and managed in a private cloud composed of eX5 servers.

Workload Deployer patterns contain preconfigured virtual machines with IBM middleware products installed, such as WebSphere and IBM DB2®, which simplify the deployment and management of virtual machines. *Patterns* are hardened systems that incorporate the best practices IBM has learned from our experience with thousands of customers over a decade. Workload Deployer patterns have the additional benefit of providing consistent and repeatable deployments in minutes.

Workload Deployer uses a standardized image for automated deployment

This section describes how IBM Workload Deployer can automatically create a virtual machine running WebSphere Application Server in an eX5 private cloud with almost no user interaction.

Automated deployment

Here is what happens:

1. As the user, you access Workload Deployer through a browser. You begin by logging in to Workload Deployer and selecting a pattern to deploy. With the pattern selected, you can configure the virtual machines. You may also install script packages that will automatically install your applications on the virtual machine.
2. Workload Deployer downloads, to cloud storage, the components necessary for the hypervisor to deploy the virtual machine on the physical machine. Currently, the hypervisor used by Workload Deployer on eX5 is VMware ESXi. The downloaded components include the WebSphere virtual machine image, the configuration information specified by you, and any additional applications and associated shell scripts that you want to install on the virtual machine. Workload Deployer remembers what virtual machines it downloaded to the hypervisor, and it will not download them a second time. These images are large, so a significant amount of time is saved by not downloading virtual machine images on subsequent deployments.
3. Workload Deployer issues a request to the hypervisor to create a virtual machine from the virtual image that was just downloaded.
4. The hypervisor reads the virtual machine image and configuration information that were downloaded and builds a properly configured virtual machine with IBM middleware, such as WebSphere, preinstalled. The hypervisor then informs Workload Deployer that the virtual machine is ready.
5. Workload Deployer sends a command to the hypervisor to boot the virtual machine.

6. During the boot process, any one-time customization of the system is performed, such as setting the root password or specifying the IP address of the virtual machine. These actions are managed by an IBM-developed software component, called the Activation Engine.
7. When the virtual machine is up and running, Workload Deployer issues a command to the hypervisor to execute the script packages and install the applications specified by the user.

IBM Workload Deployer is fast

As you can see, deploying a virtual system by using Workload Deployer is easy. Virtual system deployment by using Workload Deployer is also fast. Figure 7 shows the results of a test in which we installed a WebSphere cluster of virtual machines, both manually and by using a production-level Workload Deployer; we measured the amount of time for each installation. The manual installation was performed on an IBM Power 750. The automated installation was performed on a System x3650 X5 and Workload Deployer 3.0.0.1. Figure 7 shows the results. In this figure, “IWD” represents IBM Workload Deployer.

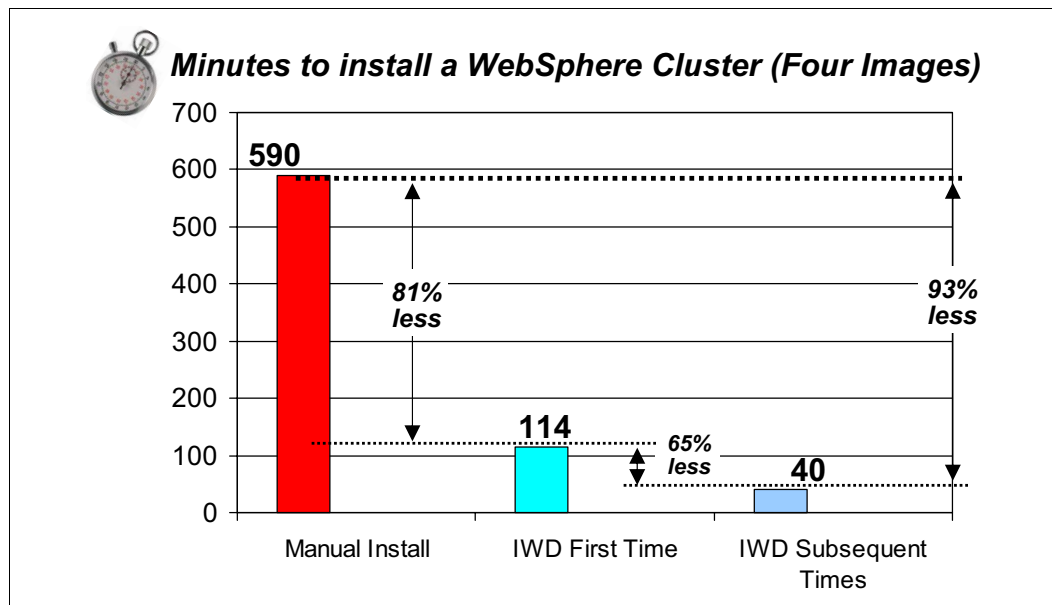


Figure 7 IBM Workload Deployer reduces deployment time for virtual machine images by ten times

In this case, manual installation took 590 minutes. When we first used Workload Deployer to install the same system, the amount of time was 114 minutes. The second deployment of the system with Workload Deployer took 40 minutes, for an overall reduction in deployment time of 93%. The reason for the improvement between the first and subsequent deployments by using Workload Deployer is that the virtual machine image was downloaded the first time it was used, but not thereafter.

Furthermore, not only are deployments faster with Workload Deployer, they are identical, so you do not have to be concerned about differences that result from human error.

WebSphere eXtreme Scale improves eX5 performance

WebSphere eXtreme Scale transactional data grids further improve eX5 performance.

Customers are won or lost in a fraction of a second

With the increasing availability of high-speed networks, Internet users are increasingly demanding faster response times from the websites they visit. The New York Times⁴ reported that users will abandon one website for a competitor's if the difference in response time is a mere 250 milliseconds. So it is clear that users demand extremely low response times, even under heavy workload.

What causes response time delays

Several factors can cause delays in response time. One factor is the latency in reading and writing to a database stored on disk. In addition, relational databases store data in interconnected tables, however most applications today are written in Java and reference data objects. The conversion of tabular data to objects is another source of delay. When data is stored as memory objects, as it is with Java, rather than in file-based tables, performance is dramatically improved. See Figure 8.

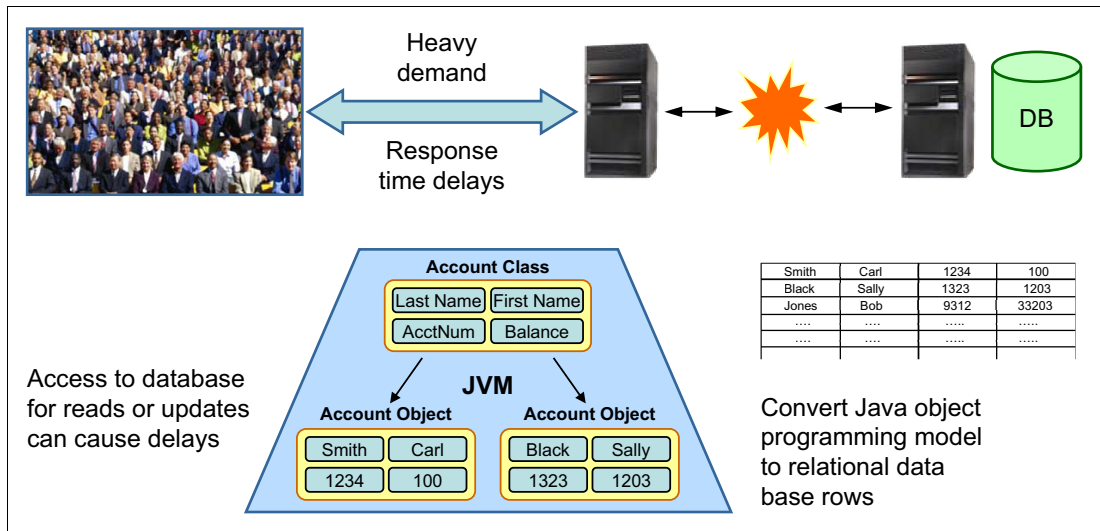


Figure 8 Several causes of response time delays

WebSphere eXtreme Scale accelerates performance

WebSphere eXtreme Scale accelerates performance by providing a low-latency, elastic data grid for storing objects. The data grid appears to applications as a single, virtualized, in-memory cache. A WebSphere eXtreme Scale grid can be effectively used as a data cache for a database or other data source, which is generally slower to respond to requests because it has to access data that resides on a hard disk.

⁴ For *Impatient Web Users, an Eye Blink is Just Too Long to Wait*, Steve Lohr, New York Times, Feb. 29, 2012 http://www.nytimes.com/2012/03/01/technology/impatient-web-users-flee-slow-loading-sites.html?_r=1&agewanted=all

This elastic scalability is possible by the use of distributed-object caching. The grid is implemented by combining the memory contained in a collection of JVMs as depicted in Figure 9. In this figure, “WXS” represents WebSphere eXtreme Scale.

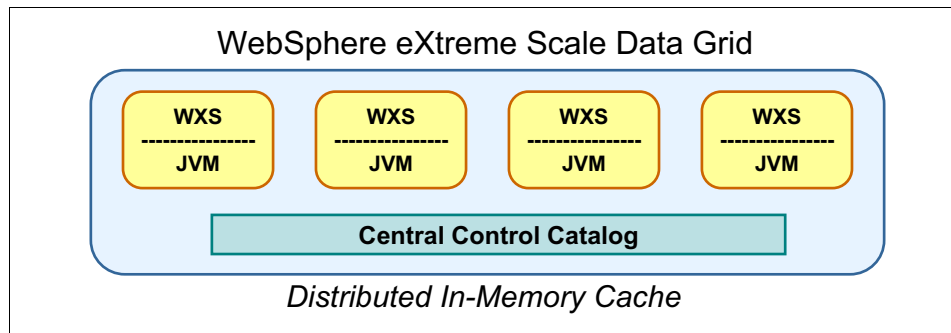


Figure 9 WebSphere eXtreme Scale accelerates performance using a scalable grid

These JVMs can be located in the same physical machine, or they can be dispersed geographically among multiple physical machines. The data objects in the grid are partitioned among these JVMs. Applications, however, do not need to detect which JVM contains the data, because the grid is presented as a single entity, managed by a single central catalog server. The grid can grow by adding JVMs, or shrink by removing JVMs. Data in the grid can be replicated to enhance reliability and performance. In addition, the high performance of the grid is able to support the most demanding systems, such as extreme transaction processing (XTP) systems.

WebSphere eXtreme Scale usage patterns

This section describes various ways that a WebSphere eXtreme Scale grid can be used. We also describe the WebSphere eXtreme Scale write-behind technology for improving performance.

Side cache

The simplest way to use WebSphere eXtreme Scale is as a *side cache*, which is completely managed by the programmer. When an application requires some data, it first tries to retrieve the data from the side cache. If the data is not in the side cache, the application loads the data into the side cache from the database and then uses that data. The side cache offers the following benefits:

- ▶ Can be simple to use
- ▶ Can provide exceptional performance
- ▶ Can reduce the load on the external data source
- ▶ Can improve response time

However, the side cache does require that the programmer be aware of when the cache is filling up and take corrective action. It also requires that the programmer write any changed data back to the external source.

Inline cache

Another pattern is to use the WebSphere eXtreme Scale data grid as the “front end” of a database, or inline cache. An inline cache is similar to a side cache, with the important difference that WebSphere eXtreme Scale manages the contents of the inline cache. The application always accesses the inline cache to obtain the data; it never goes to the external source. If the data is not available in the inline cache, a WebSphere eXtreme Scale loader program obtains the data from the database, stores it in the inline cache for later reuse and

then returns the data to the program. If the data is changed in the inline cache, the loader automatically writes the change back to the database.

The loader program operates without programmer intervention. Because of the loader program, some users treat the WebSphere eXtreme Scale inline cache as the database of record and view the database as a way to persist the data for backup or catastrophic failure.

An inline cache improves performance by reducing the latency caused by the database overhead and network traffic that is necessary to access application data. By reducing the number of requests on both the database and the network, overall system performance improves.

Write-behind technology

Users can further improve the performance of their inline caches by using WebSphere eXtreme Scale's write-behind technology. With the commonly-used write-through technology, changes to the grid are immediately written back to the external data source. Control is not returned back to the program until the write has completed. With *write-behind* technology, grid changes are not written back immediately. Instead, they are collected and later written back as a group, asynchronously, without halting program operation. For programs that can tolerate the delay, write-behind technology improves performance for several reasons:

- ▶ The program does not have to wait while the loader writes each change to the “back end.” Write-behind technology writes the grid changes to the back end concurrently, with the program running.
- ▶ Write-behind technology typically results in fewer write operations to the database. When a grid element is updated multiple times, write-behind writes only the last change to the back end.

HTTP session management

A WebSphere eXtreme Scale data grid can also be used for hypertext transfer protocol (HTTP) session management. HTTP session management is a function offered by Java Platform, Enterprise Edition (Java EE) application servers, such as WebSphere Application Server. HTTP session management allows web applications to store session state information about a given user across many web requests. A typical use of an HTTP session is for a website shopping cart, where information about a user's intended purchases is stored until the user checks out.

To increase reliability, most application servers provide some type of session replication. This approach allows other application servers in the environment to access session data. Thus, if one application server goes down, another one can continue to process requests by using the saved session data. Session replication is normally performed by writing to a database or by memory-to-memory replication. These two options, however, can pose problems. Using a database can have performance implications, and memory-to-memory replication can be restricted by the amount of memory available in a single JVM.

Using WebSphere eXtreme Scale can overcome these problems. Saving HTTP session state information in a data grid provides for the sharing of user session data by any set of application servers, even across data centers. Furthermore, the WebSphere eXtreme Scale grid can be accessed using standard HTTPsession application program interfaces (APIs). In this way, the data grid can be used without making any changes to the application code. The elastic capability of WebSphere eXtreme Scale eliminates any restrictions on the size of session data.

The HTTP session store of WebSphere eXtreme Scale offers the following benefits:

- ▶ Provides HTTP session failover support when application servers are shut down or end unexpectedly.
- ▶ Supports environments that run various versions of WebSphere Application Server, or application servers from various vendors.
- ▶ Keeps users logged on during data center failover in geographically dispersed locations.

WebSphere eXtreme Scale increases application performance dramatically

Figure 10 shows how WebSphere eXtreme Scale can dramatically increase application performance. Here, WebSphere eXtreme Scale is used as a side cache. When the application starts, the data grid is completely empty and is filled as the application requires data during execution. The bottom axis represents time. Notice that, as time increases and as the WebSphere eXtreme Scale side cache fills with data, both throughput and response time improve dramatically.

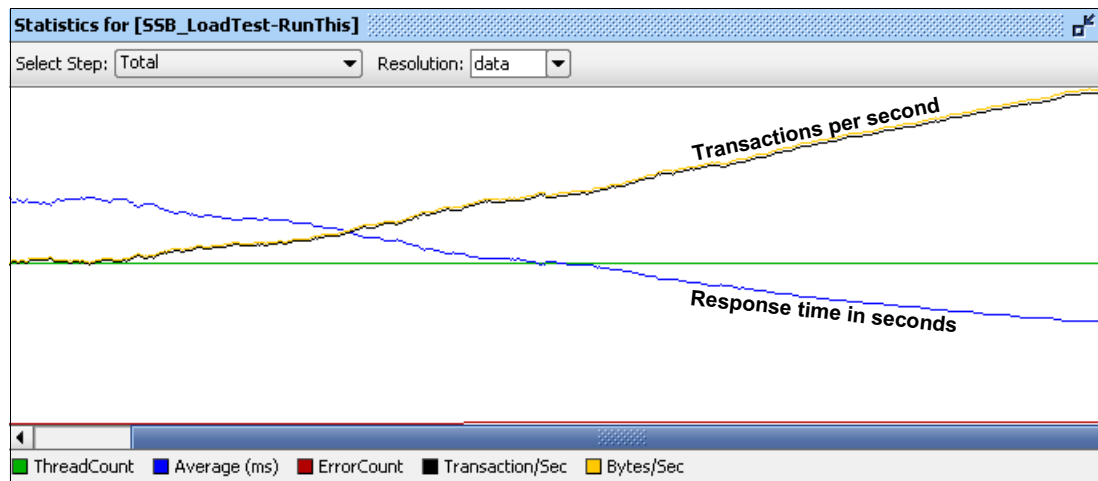


Figure 10 WebSphere eXtreme Scale increases application performance dramatically

The WebSphere eXtreme Scale grid is transactional

The WebSphere eXtreme Scale cache is *transactional*, meaning that an application can make several updates to the cache and have it appear as a single update. Either all of the updates will be made to the cache or none of them will be made. The beginning and end of each transaction is defined by the programmer, and all updates that occur between those two endpoints are treated as a single update. The WebSphere eXtreme Scale transactional grid guarantees that either all the changes are made or none of them are. Transactional grids, therefore, can ensure data consistency within the grid in the event of failure.

Transactional grid without database: normal operation

How does the WebSphere eXtreme Scale transactional data grid work? We begin by describing a typical case (normal operation) in which no database is used. All data resides within the grid. Figure 11 illustrates the process.

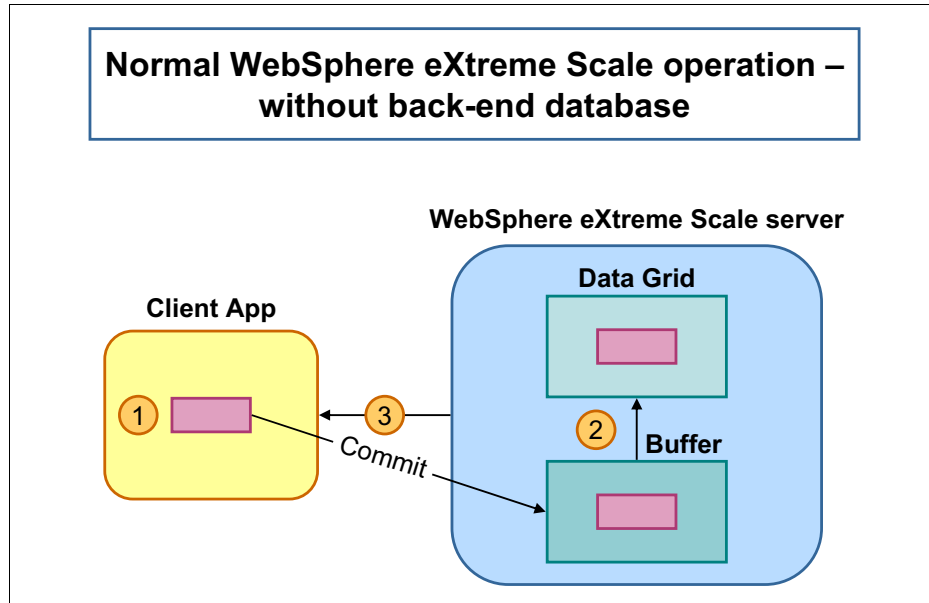


Figure 11 A transactional grid guarantees data consistency in the event of a system failure

The box in the *Client App* represents user data that will be written to the grid. The figure shows the following steps:

1. The user issues the **Commit** command, and the modified user data is transferred to a buffer in the WebSphere eXtreme Scale server.
2. WebSphere eXtreme Scale moves the data to the proper location in the grid.
3. WebSphere eXtreme Scale notifies the user that the operation has completed normally.

Transactional grid is protected during system failure (no back-end database)

How does a WebSphere eXtreme Scale transactional grid function if a failure occurs before the data can be saved on the server? Figure 12 illustrates the process.

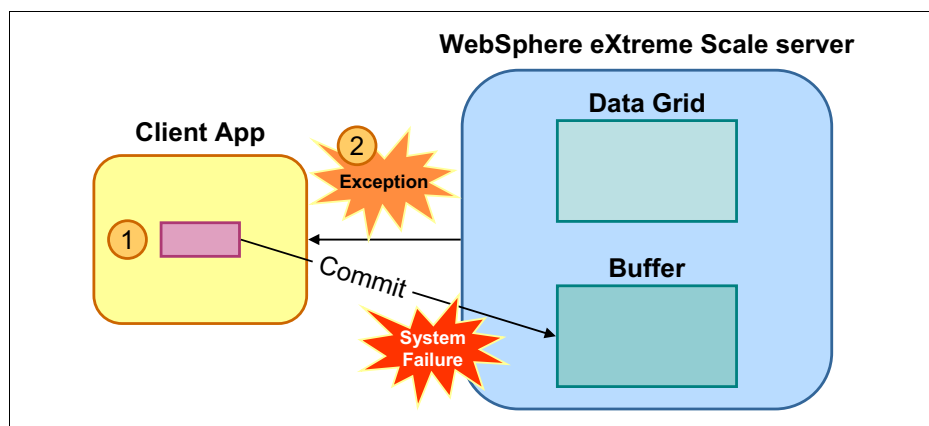


Figure 12 A transactional grid is protected during system failure (no back-end database)

Again, the box in the *Client App* represents user data that will be written to the grid. The figure shows the following process:

1. The user issues the **Commit** command, and a system failure occurs before all of the data has been transmitted to the WebSphere eXtreme Scale server.
2. WebSphere eXtreme Scale raises an exception, and none of the data changes are saved within the grid (the data in the grid remains consistent).

WebSphere eXtreme Scale transactional support extends to back-end databases

In most cases, a back-end database will be used to save the data in the grid. Databases also support transactions. How does the WebSphere eXtreme Scale transactional grid interoperate with the database? Figure 13 illustrates the process.

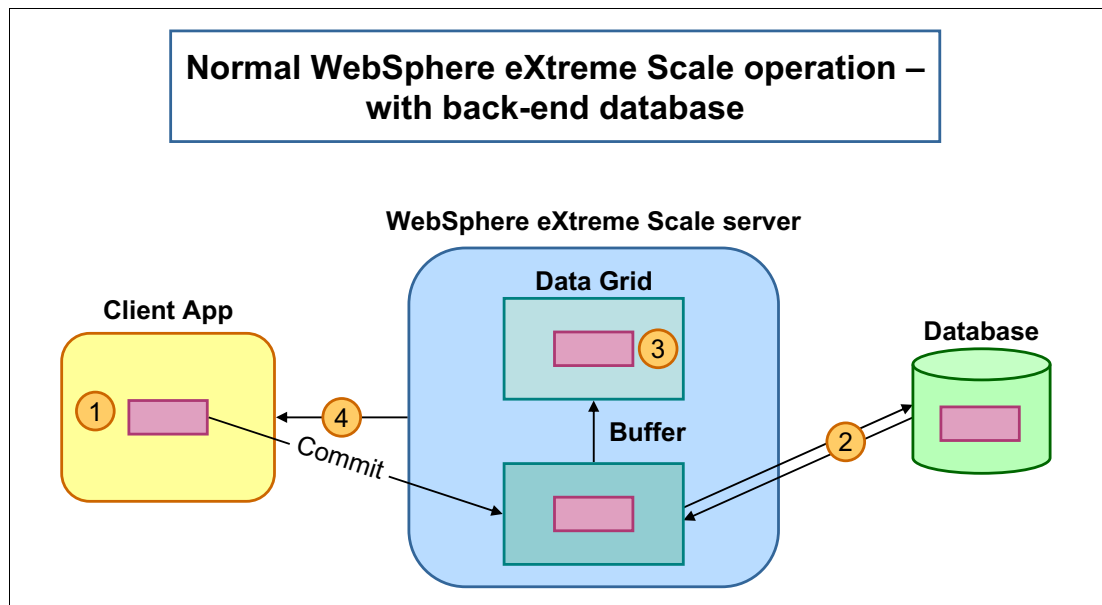


Figure 13 WebSphere eXtreme Scale transactional support extends to back-end databases

The box in the *Client App* represents user data that will be written to the grid. The figure shows the following steps:

1. The user issues the **Commit** command, and the modified data is transferred to a buffer on the WebSphere eXtreme Scale server.
2. WebSphere eXtreme Scale writes the modified data to the database and waits until a completion message is received.
3. The modified data is written to the proper location in the grid.
4. WebSphere eXtreme Scales signals completion to the user.

Transactional grid is protected during system failure (with back-end database)

How does WebSphere eXtreme Scale transactional grid function if a failure that occurs before the data can be saved to the database? Figure 14 illustrates the process.

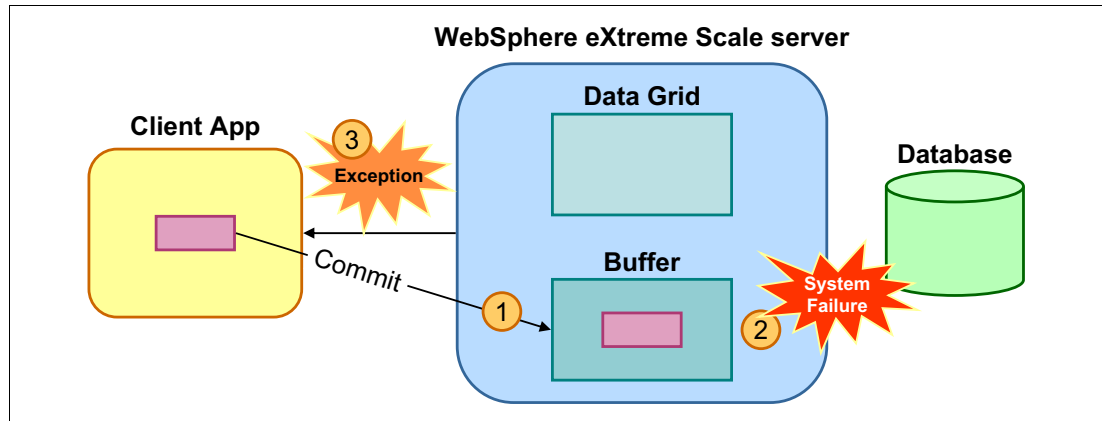


Figure 14 A transactional grid is protected during system failure (with back-end database)

The box in the *Client App* represents user data that will be written to the grid. The figure shows the following steps:

1. The user issues the **Commit** command, and the changed data is moved to a buffer on the WebSphere eXtreme Scale server.
2. WebSphere eXtreme Scale attempts to write the data to the database, but a system failure occurs, preventing the data from being stored in the database.
3. WebSphere eXtreme Scale raises an exception and discards the new data, so that the grid is not updated and the data remains consistent.

WebSphere eXtreme Scale studies

We conducted studies to determine the ability of WebSphere eXtreme Scale and MAX5 to improve processing environments. This section has an overview of two studies with results, and an experiment with MAX5.

Large investment bank increases transactions by 40 times

An investment bank wanted to develop a “next generation” order system. The system that the bank was using processed only 300,000 transactions per day, and their competitor database system could not scale sufficiently to meet demands. After deploying WebSphere eXtreme Scale as a database front-end, the bank was able to process 40 times more transactions per day, increase its revenue by a factor of four, and reduce response time to 2.5 milliseconds.

Figure 15 summarizes these results.

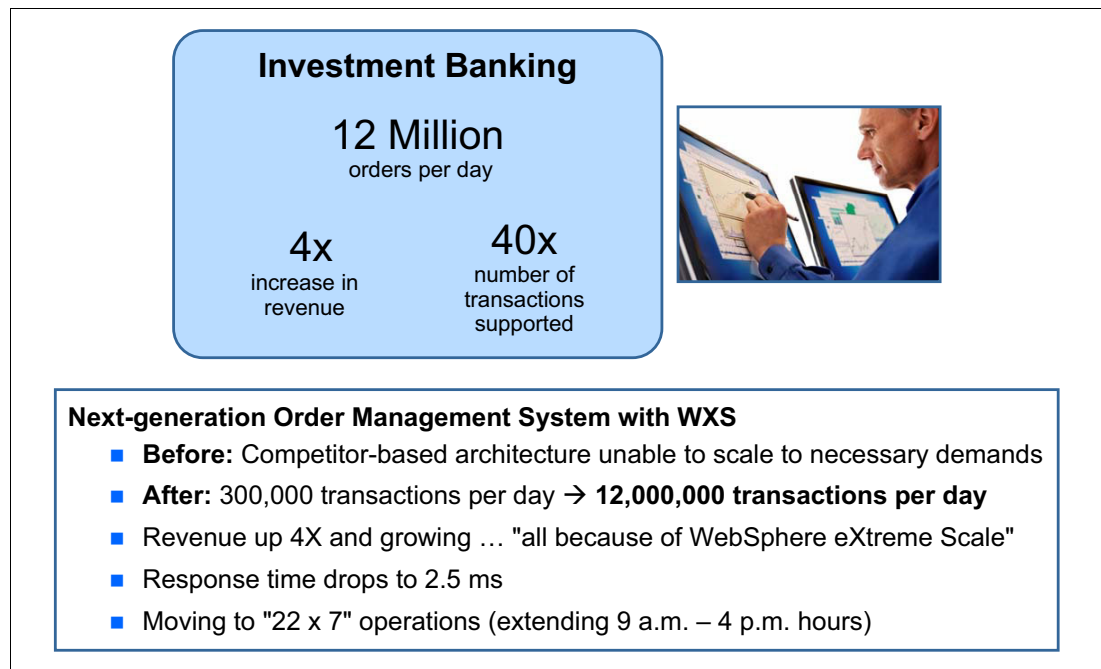


Figure 15 A large investment bank increases transactions by 40 times with WebSphere eXtreme Scale

Social media website reduces response time by 90%

A social media website chose WebSphere eXtreme Scale to improve performance of the site. This website records a vast amount of data for its users. Before implementing WebSphere eXtreme Scale, users were experiencing 60-millisecond response times because of the heavy demand placed on the databases. The site was moved to WebSphere eXtreme Scale to improve performance by maintaining user data in memory, using an inline cache. After the change was made, response time dropped by 90%, to 6 milliseconds. In 2011, the site was able to support 450,000 concurrent users and handle close to one million requests per second.

WebSphere eXtreme Scale performance experiments with MAX5

The need for reduced response time on the social media website (see “Social media website reduces response time by 90%” on page 19) was just the kind of application that could benefit from MAX5 because of the large data requirements. We decided to study this further.

Cache larger databases in WebSphere eXtreme Scale with MAX5 on eX5

To demonstrate the benefits of MAX5, we used a system verification test, developed by the WebSphere eXtreme Scale development team, that simulated how the social media website used WebSphere eXtreme Scale.

Figure 16 illustrates our test system. Our base system was a production-level System x3690 X5 with 16 cores and 256 GB of RAM.

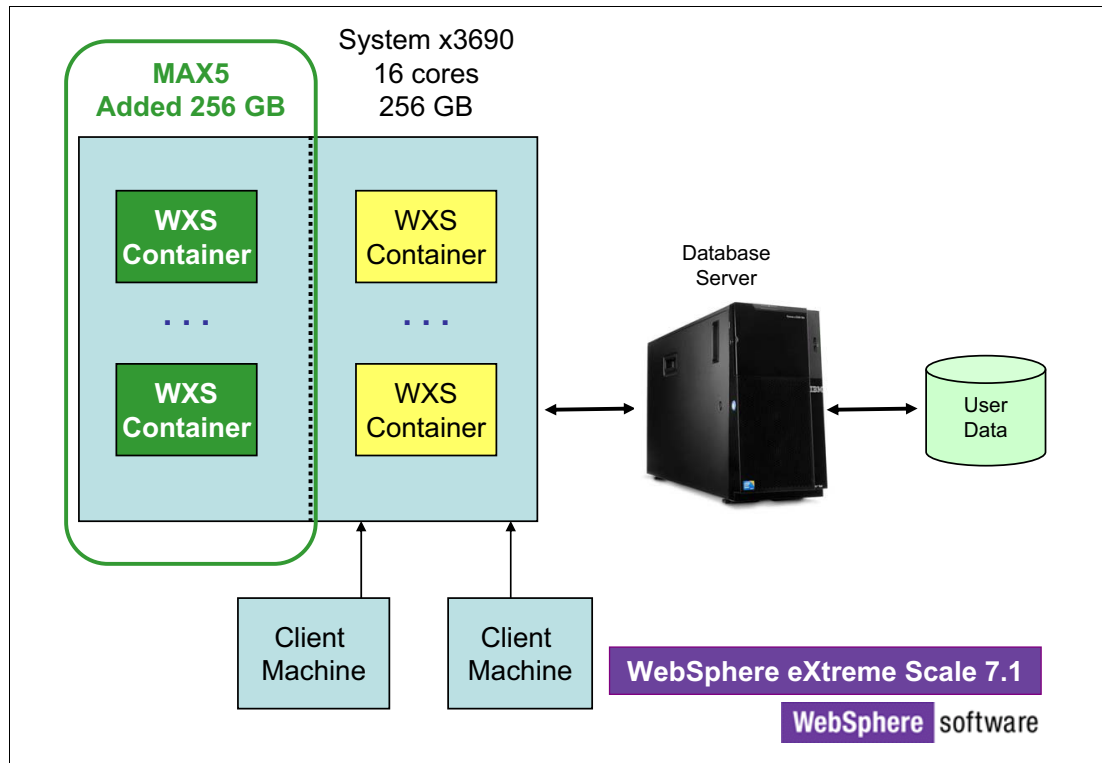


Figure 16 Our base system was a System x3690 X5

The yellow boxes, marked *WXS Container* inside the System x3690 X5, represent the components of the WebSphere eXtreme Scale distributed grid. Each container is a JVM, and the WebSphere eXtreme Scale grid is distributed across the memory in those JVMs. In our test, all of the containers resided in one System x3690 X5, but WebSphere eXtreme Scale lets you distribute the grid across multiple physical machines that can be located in different data centers or geographies. Then, to our base system, we added MAX5 with 256 GB, for a total of 512 GB.

Social media website benchmark characteristics

We populated our system with 400 clients operating concurrently. Each client was able to read and write social media objects chosen at random. We used a 350 GB database. In our MAX5 system, we created a 450 GB grid that was large enough to contain our entire 350 GB database. Thus, all data requests from the website application could be satisfied directly from the grid.

In our base system, with only 256 GB, we created a 225 GB grid, which was too small to contain our database. Thus some data requests would require a call to the database system to be satisfied.

Social media website throughput increases from 2.4 to 4.9 times

We ran two separate tests on both the base system and the MAX5 system and measured the results. In one test, our workload consisted of 80% read requests and 20% write requests. In the second test, the workload consisted of 100% reads.

For our workload of 80% reads and 20% writes, throughput on the MAX5 system was 2.4 times the throughput of the base system. For our workload of 100% reads, throughput on the MAX5 system was 4.9 times the throughput of the base system. See Figure 17.

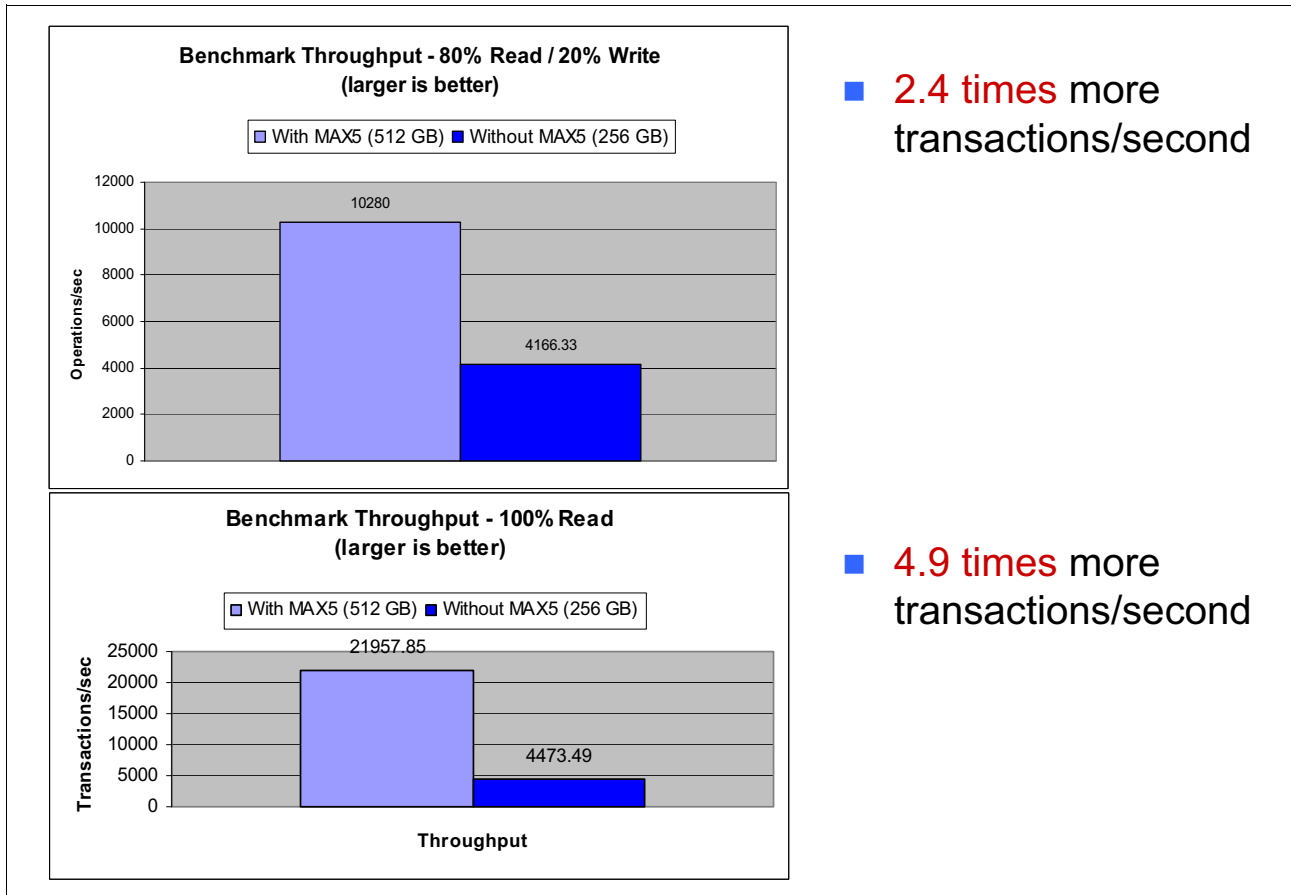


Figure 17 Social media website throughput increased from 2.4 times to 4.9 times

Social media website user-response time reduced by 60 - 80%

For our workload of 80% reads and 20% writes, response time on the MAX5 system was 60% less than the base system response time. When we tested with 100% reads, response time was 80% less than the response time on the base system. See Figure 18.

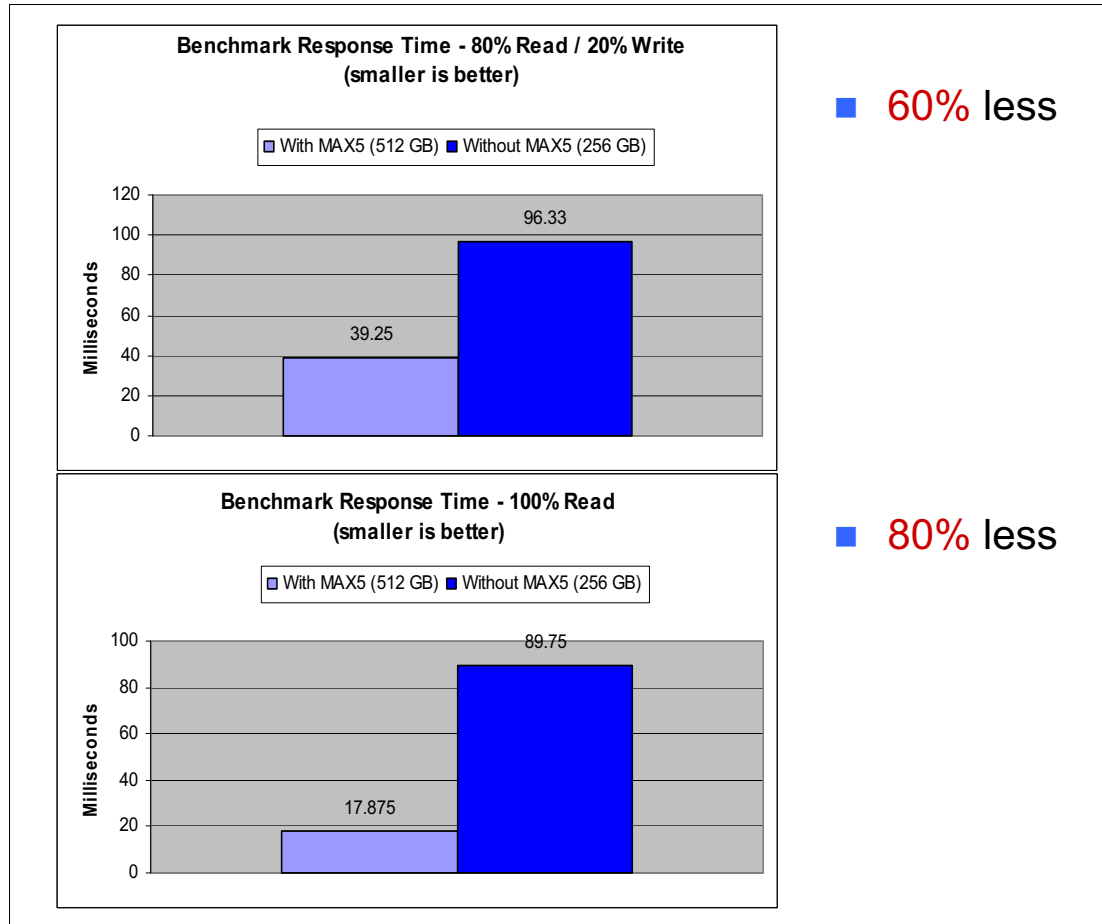


Figure 18 Social media website response time was reduced by from 60% to 80%

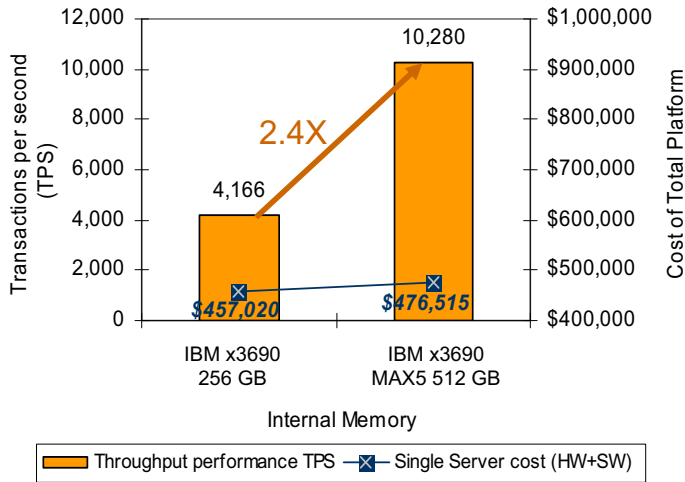
MAX5 improves the price/performance of the social media website

We next looked at the cost benefits of MAX5. We first compared the total cost of acquisition of the MAX5 system, compared to the total cost of acquisition of the base system. The MAX5 system cost only 4% more than the base system. In other words, for an additional 4%, MAX5 can improve throughput by 2.4 times. On a price/performance basis, each transaction/second on the base system costs \$109.70, but it costs only \$46.35 on the MAX5 system.

See Figure 19 on page 23, which shows the list prices that were in effect when the analysis was performed in mid-2011. These prices might not represent current list prices. Discuss your current environment with your IBM representative to determine how much you can save, at current prices, with MAX5.

For only 4% increase in cost, MAX5 yields **2.4X** throughput improvement

Price/Performance of MAX5 with WXS



	IBM x3690 X5 256 GB	IBM x3690 X5 MAX5 512 GB
System	\$17,364	\$17,364
Added MAX5 & Memory		\$19,495
Storage	\$202,856	\$202,856
Total HW	\$220,220	\$239,715
WXS license PVU=70	\$238,800	\$236,800
Total SW	\$236,800	\$236,800
Total Platform	\$457,020	\$476,515
Throughput TPS	4,166	10,280
Dollars per TPS	\$109.70	\$46.35

Figure 19 Adding MAX5 improves price and performance

Adding MAX5 can be a better business decision than adding another server

We also looked at how much hardware we would have to add to the base system to achieve the same 2.4 times throughput gain that was observed in the MAX5 system for the 80% read-test case. We determined that we required a second System x3690 X5 with 256 GB of RAM. The cost of this new system can be as much as 49% more than the cost of the MAX5 system that was used in our tests.

Figure 20 on page 24 shows what a system would need without MAX5 (the left side of the figure) to provide the same performance as a system with MAX5 (on the right side of the figure). In the figure, “WXS” represents WebSphere eXtreme Scale.

Table 1 on page 24 shows our calculations for each of these systems. Notice that a large portion of the additional cost is from the purchase of additional WebSphere eXtreme Scale licensing.

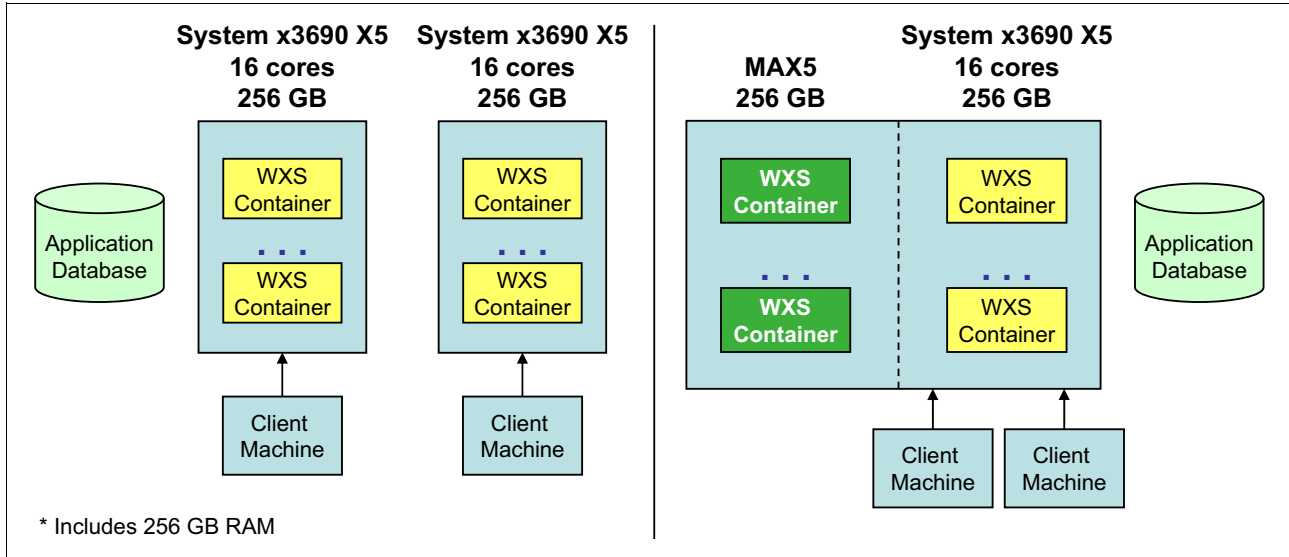


Figure 20 The left side shows a system without MAX5 that would be needed to provide the same performance as the system with MAX5 on the right

Table 1 Details of purchase requirement calculations

	Cost	Quantity without MAX5	Total	Quantity with MAX5	Total
System x3690 X5 (includes 256 GB RAM)	\$17,364.00	2	\$34,728.00	1	\$17,364.00
MAX5	7,495.00	0	0.00	1	7,495.00
256 GB RAM	12,000.00	0	0.00	1	12,000.00
Disks			202,856.00		202,856.00
Red Hat Enterprise Linux	0.00	2	0.00	1	0.00
WebSphere eXtreme Scale	236,800.00	2	473,600.00	1	236,800.00
Total			\$711,184.00 ^a		\$476,515.00 ^a

a. Values in this table are list prices that were published on the IBM website when the analysis was done in mid-2011. They might not represent current list prices. Discuss your current environment with your IBM representative to determine how much you can save, at current prices, with MAX5.

Summary

This paper describes how various IBM WebSphere products are optimized to make use of eX5 multiprocessors to provide accessibility for web-facing applications. WebSphere Application Server, MAX5, and WebSphere eXtreme Scale include the following benefits:

- ▶ *WebSphere Application Server* has been designed to take advantage of eX5 multiprocessors. With WebSphere Application Server on a System x3850 X5, compared with a competitor's application server, on a competitor's identically configured hardware, we showed that the IBM system was 44% faster, and the amortized cost was determined to be 70% less than that of the comparison system.
- ▶ *WebSphere Application Server and eX5* support applications of all sizes. Although large applications might require all the resources of an eX5 system, WebSphere and eX5 also provide an exceptional platform for consolidating smaller applications. Consolidation reduces system costs by reducing software costs, space, energy costs, and administrative costs.
- ▶ *MAX5*, which adds memory apart from the processor, increases productivity, and reduces costs in a way that other servers of this class cannot. It can expand memory capacity, up to 1 TB of additional memory, and can reduce costs by using smaller-capacity DIMMS.

The additional memory provided by MAX5 improves virtual machine performance. This paper shows that, in our testing, performance increased by 15%, and that MAX5 enables more WebSphere Application Server workloads to be consolidated. In one study, comparing System x3850 X5 with and without MAX5, we observed that, by adding MAX5 with 128 GB, we were able to increase the number of consolidated workloads by 46%.
- ▶ *WebSphere eXtreme Scale* provides a transactional data grid for even higher performance on eX5. WebSphere eXtreme Scale accelerates performance by providing a low-latency, elastic data grid for storing objects. WebSphere eXtreme Scale usage patterns enable the use of side caches, inline caches, and HTTP session caches. Cache performance can be enhanced by using write-behind technology. Furthermore, by using MAX5 with WebSphere eXtreme Scale, we showed that it was possible to improve the throughput of a web-facing application by 2.4 times to 4.9 times for an additional cost of only 4%.
- ▶ *IBM Workload Deployer* accelerates time-to-value in eX5 virtualized environments. IBM Workload Deployer provides access to IBM middleware virtual images and patterns to more easily, quickly, and securely create application environments.

Based on the information in this paper, we believe that WebSphere Application Server and WebSphere eXtreme Scale running on eX5 multiprocessors with MAX5 provide the power, memory, and virtualization features necessary for supporting web-facing applications in the most demanding environments.

For more information

For more information about the products covered in this publication, visit the web addresses:

- ▶ IBM System x®
<http://www.ibm.com/systems/x/>
- ▶ IBM Software
<http://www.ibm.com/software>
- ▶ WebSphere software
<http://www.ibm.com/software/websphere/>
- ▶ WebSphere Application Server
<http://www.ibm.com/software/webservers/appserv/was/>
- ▶ WebSphere eXtreme Scale
<http://www.ibm.com/software/webservers/appserv/extremescale/>
- ▶ IBM eX5 Enterprise Systems:
 - *IBM eX5 enterprise systems: Breakthrough innovation for smarter computing, 2012*
<http://www.ibm.com/systems/info/x86servers/ex5/>
 - *IBM eX5 Portfolio Overview: IBM System x3850 X5, x3950 X5, x3690 X5, and IBM BladeCenter HX5, REDP-4650-04*
<http://www.redbooks.ibm.com/abstracts/redp4650.html>
- ▶ IBM MAX5
 - *Add Memory, Improve Performance, and Lower Costs with IBM MAX5 Technology, REDP-4846*
<http://www.redbooks.ibm.com/abstracts/redp4846.html>
 - IBM MAX5 for System x
<http://www.ibm.com/systems/x/options/memory/max5/index.html>

The team who wrote this paper

This paper was produced by a team of specialists from around the world working at the International Technical Support Organization (ITSO), Poughkeepsie Center.



Dr. Howard Operowsky is a Senior Technical Staff Member in the IBM Strategy and Technology Lab where he evaluates server and middleware technology and is currently focused on virtualization. Since joining IBM, he has been manager or technical leader of a variety of IBM products, including WebSphere Commerce architecture, collaborative multimedia applications, IBM AIX/ESA® operating system architecture, the design and development of compilers and programming languages for high-end parallel systems, and performance analysis of mainframe operating systems. He has co-authored multiple patents in the area of multimedia applications and is the recipient of an IBM Outstanding Technical Achievement Award for work on operating system architecture. He received his Ph.D. in Computer Science from the Courant Institute of Mathematical Sciences at New York University, and is a Senior Member of the Institute of Electrical and Electronics Engineers (IEEE).



Karen Lawrence is a Technical Writer with the IBM ITSO team in North Carolina. She has 25 years' experience in IT, with expertise in application design, change management, and the software development lifecycle (SDLC). She has worked with SMEs on leading technologies in global, regulated enterprises, in the areas of storage systems, security, and disaster recovery, and the applications to support these. Karen holds a Bachelor's degree in Business Administration from Centenary College in New Jersey.

Thanks to the following people for their contributions to this project:

- ▶ Randall W. (Randy) Lundin, Product Marketing - High End System x, for his contributions and sponsorship of this project
- ▶ Vijay Bhadriraju, SWG Strategy and Technology Lab
- ▶ Matt Dickens, SWG Strategy and Technology Lab
- ▶ Doug Larson, WebSphere Performance Team
- ▶ Monica Martinez, System x Competitive Technology Analyst
- ▶ Steve Worley, Software Engineer
- ▶ Linda Robinson, Graphics Specialist
- ▶ Diane Sherman, Editor

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Stay connected to IBM Redbooks

- ▶ Find us on Facebook:
<http://www.facebook.com/IBMRedbooks>
- ▶ Follow us on Twitter:
<http://twitter.com/ibmredbooks>
- ▶ Look for us on LinkedIn:
<http://www.linkedin.com/groups?home=&gid=2130806>
- ▶ Explore new IBM Redbooks® publications, residencies, and workshops with the IBM Redbooks weekly newsletter:
<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>
- ▶ Stay current on recent Redbooks publications with RSS Feeds:
<http://www.redbooks.ibm.com/rss.html>

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

© Copyright International Business Machines Corporation 2012. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

This document REDP-4847-00 was created or updated on August 10, 2012.



Send us your comments in one of the following ways:


- ▶ Use the online **Contact us** review Redbooks form found at:
ibm.com/redbooks
- ▶ Send your comments in an email to:
redbooks@us.ibm.com
- ▶ Mail your comments to:
IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400 U.S.A.



Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX/ESA®	Redpaper™	Smarter Healthcare®
DB2®	Redbooks (logo)  ®	System x®
IBM®	Smarter Cities®	WebSphere®
Redbooks®	Smarter Commerce™	

The following terms are trademarks of other companies:

Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.