



white paper

Achieving High Throughput for Business-Critical Applications Using Microsoft BizTalk Server 2006 and SQL Server 2005

A Benchmark Report Demonstrating Unprecedented Performance with Unisys ES7000/one Enterprise Servers

performance

Table of Contents	
Executive Summary	4
Test Methodology and Environment	6
Application Software	6
Batch Process Testing	9
Results	11
Scenario I – Simple App	11
Scenario II – Latency App	13
Performance Analysis	14
Database Servers	14
BizTalk Servers	17
Conclusion	19
About the Authors	19

Executive Summary

The systems supporting complex business requirements and applications are often mission-critical and demand high throughput. Integrating related processes and data is an ever-increasing necessity for large organizations. As such, the systems themselves require software that provides reliable, high-performance data access from an extensive variety of sources — across wide area networks — including the Internet and rules-based processing, as well as routing and comprehensive management and auditing capabilities.

Consider for a moment that:

- Large conglomerates that are the result of corporate mergers and acquisitions require disparate systems to deliver consolidated information and reports.
- The volume of inter-institution financial transactions continues to rise as economies become more global.
- Law enforcement agencies need a consolidated view of offender information in order to effectively and efficiently apprehend suspects and prosecute crimes.
- The use of radio frequency identification (RFID) in supply chain management (SCM) is exploding at astronomical rates and generating even more exorbitant amounts of data.

Of the available solutions for these and other complex business situations, the combination of Microsoft® BizTalk® Server 2006 and SQL Server 2005 running on a Microsoft Windows® Server 2003 platform offers some of the industry's best mission-critical functionality and availability. And with its scaling on x64 instruction set architectures (x64), this configuration is capable of achieving very high levels of performance.

Unisys ES7000/one Enterprise Servers benefit enterprise BizTalk applications by providing an economical, highly scalable x64 platform that offers a much smaller management footprint than commodity servers. And, the ES7000 Real-Time Infrastructure capabilities provide agility to respond to changes in demand in minutes — rather than days or weeks.

This document presents an overview of an actual customer benchmark that achieved an unprecedented 1,156 orchestrations per second (defined for our purposes as execution of a message process) for an application designed to mimic a business-critical process within a large European financial institution. This level of performance, utilizing ES7000/one Enterprise Servers, is by far the highest throughput ever recorded by the Microsoft BizTalk team. The team believes that, with additional time and tuning, an even higher level of throughput could be achieved.

Unisys ES7000/one Enterprise Server

The ES7000/one Server offers a flexible, single server solution that can be configured to meet enterprise workloads as they change. With the ability to grow from 4 to 32 Intel® Xeon® processors MP or from 4 to 32 Intel® Itanium® 2 processors, the ES7000/one Server scales beyond conventional Intel® processor-based platforms while delivering improved price/performance.

The ES7000/one Server also has the ability to create up to eight independent partitions that concurrently run different operating systems, making it an excellent solution for consolidating applications, migrating from expensive proprietary RISC systems, deploying business intelligence applications, and running large-scale databases.

Additional and more detailed product information is available on the Unisys website at www.unisys.com

¹ For reasons of confidentiality the client is not named.

This specific institution is converting a mission-critical enterprise application integration (EAI) solution that provides support and services to all the companies included in its worldwide business group. Unisys and Microsoft personnel worked with the institution's team to design a single financial transactions system solution to replace multiple mainframe-based legacy systems. In order to determine if the new design and platform would meet the customer's performance requirements, a proof-of-concept (POC) performance benchmark was conducted at the Unisys ES7000 Performance Center in Mission Viejo, CA and with support of the Unisys and Microsoft Solutions Alliance Technology Center (SATC) in Bellevue, Washington.

During the POC, the team endeavored to determine and document the performance characteristics of the new solution, which was built with BizTalk Server 2006 and runs on ES7000 hardware. In addition, the client wished to use the POC to identify and verify ways to minimize the overall physical and administration/management footprint of the platform.

As such, the primary objectives for this test were to:

- Achieve the highest possible throughput with predefined solution constraints and optimal hardware
- Document the maximum sustained throughput of the solution
- Determine if the combination of ES7000 hardware, the Windows 2003 operating system, and BizTalk Server 2006 could meet the workload requirements of the client (which are a sustained throughput of 800 messages per second and production peak of 1,200 messages per second)
- Develop a detailed, documented understanding of the performance characteristics of the tested solution
- Provide guidance for tuning the BizTalk Server environment for optimal performance when running the solution
- Offer architecture and design guidance for a production implementation of the solution

The results demonstrate the scalability of Microsoft BizTalk Server 2006 on the ES7000/one Server x64 configuration and denote a remarkable level of throughput. And in fact, they represent the highest throughput that has ever been recorded by the Microsoft BizTalk team.

Test Methodology and Environment

The intention of this engagement was to drive as much load as possible through the tested configurations to determine a high-water benchmark for BizTalk server scalability. In order to allow for peaks in the workload in a production environment, neither the database nor the application servers were utilized at greater than 80 percent. In addition, such significant issues as fault-tolerance were not addressed in the test environment. (Because availability matters are very important to an organization, any production solution that is developed based on this engagement or the contents of this document should take these concerns into account.)

Only batch processing was evaluated during this engagement. All test cases were carefully defined and any changes made to configuration settings were tracked for each test case to ensure quality control.

Application Software

Two applications were constructed and used during the POC – Simple Application (Simple App) and Latency Application (Latency App). Both were based upon a conceptual application architecture developed by Microsoft Consulting Services (MCS), (see Figure 1), which reflected the customer's requirements.

Simple App and Latency App worked nicely for the performance benchmark because the goal was to assess “useful applications” that would mimic the performance characteristics of the actual application without hampering the testing by including custom, customer-specific code or third-party technologies.

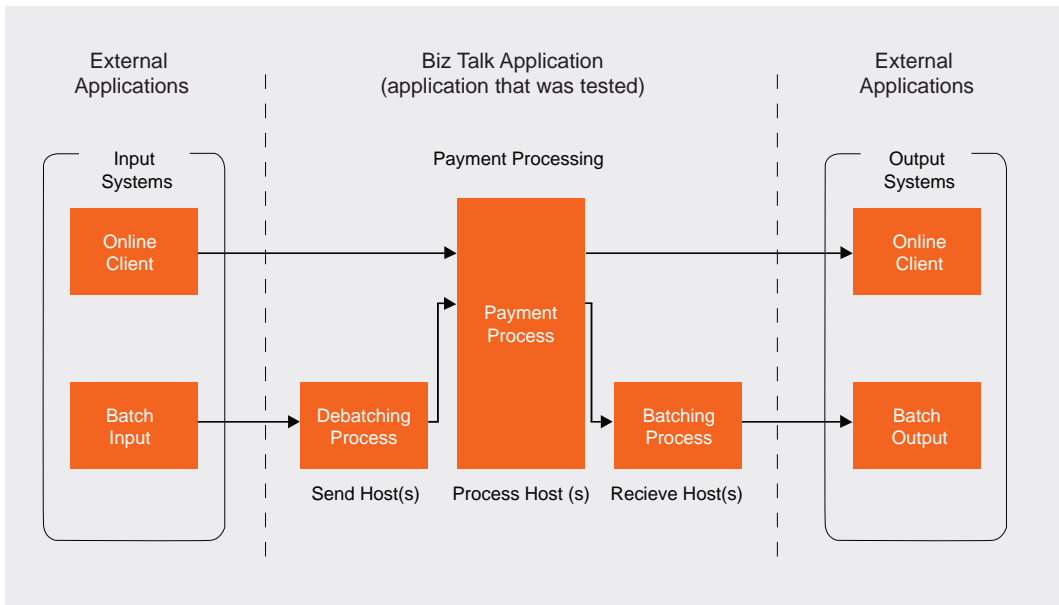


Figure 1: An overview of the conceptual application architecture

The Simple Application

The Simple App is the most straightforward orchestration that can run in BizTalk Server 2006. As pictured in Figure 2, it consists of only two steps; receive and send.

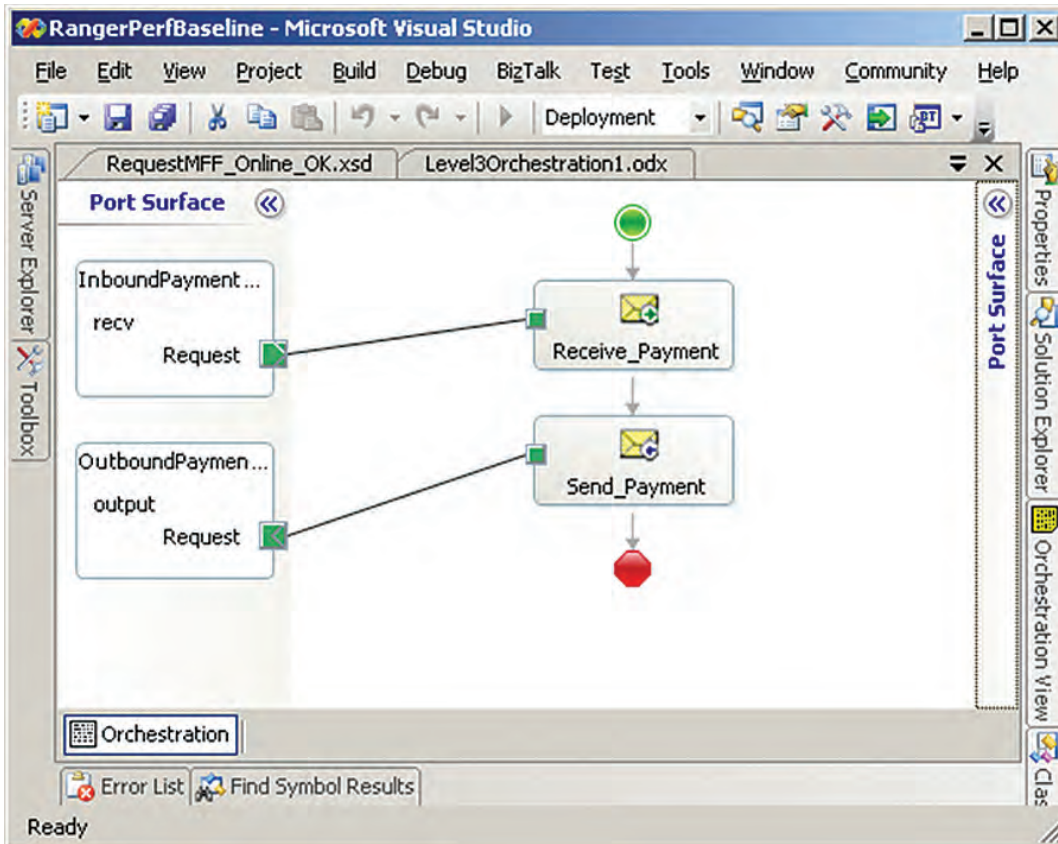


Figure 2: Simple App

The Latency Application

The Latency App is a more complicated orchestration, because it resembles the orchestration from the customer's original application. However, like the Simple App, it is unencumbered by connections to other technologies or data stores. Instead, latency values were selected based upon the expected length of each activity and are implemented by expression shapes that call out to a .NET component that will sleep for the configured amount of time. The purpose is to create a useful test application that resembles the performance characteristics of the actual application without complicating the testing with custom, customer-specific code or third-party technologies.

The Latency App, the orchestration of which can be seen in Figure 3, became the primary focus of the engagement. As such, the performance benchmark objectives were modified to concentrate on performance of the Latency App. The primary goal evolved into obtaining the highest possible maximum sustained throughput for this application as measured by orchestrations completed per second.

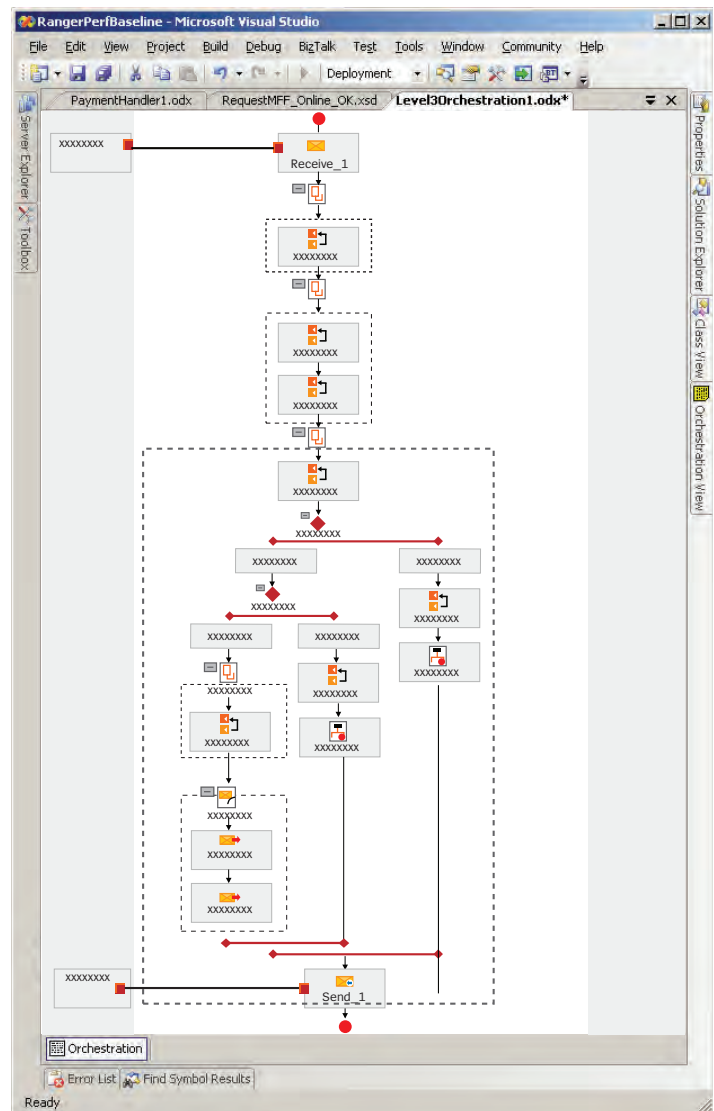


Figure 3: Latency App

Batch Process Testing

For the batch system, two test scenarios were run: Simple App and Latency App. Each scenario included three variations with a different number of processing systems for a total of six test cases. Multiple passes of each test case were often required before the optimal configuration was determined. These tests were defined as S1, S2, and S3 for the Simple App and L1, L2, and L3 for the Latency App.

The first test scenario, which used Simple App, was to confirm the following:

- Is the BizTalk solution deployed correctly and functioning as expected?
- What are the performance characteristics of the batch solution in the Simple App?
 - What is the processor usage of the BizTalk Flat File Dissassembler pipeline component?
 - What is the ideal method of measuring batch throughput?
- What is the maximum sustained throughput that BizTalk Server 2006 can achieve?

The second test scenario was to determine the performance characteristics of an application that is similar to the ultimate client production application. To that aim, the BizTalk Ranger team wrote an application that simulated the latencies of the client application (Latency App), and tests similar to those that were run during the first scenario were executed. The goals of this phase were to confirm the following for the simulated application:

- What are the performance characteristics?
- What tuning parameters can be used to optimize the configuration?
- What is the maximum sustained throughput of this configuration?

Hardware and Software Configurations

Following is a detailed list of the hardware and software utilized in the POC.

Hardware Utilized

Database Tier (Scenario I: Simple App)

Master Database

One ES7000 Model 420 Enterprise Server:

- 16 Intel® Itanium® 2 processors at 1.6GHz and 9MB cache
- 256GB memory
- PCI-X 133MHz I/O
- Four Intel® Gigabit Ethernet ports
- Up to eight Emulex fibre channel connections
- Storage Area Network (SAN): 2.5 TB (shared with Worker Database)

Worker Database

One ES7000 Model 440 Enterprise Server:

- 32 Intel® Itanium® 2 processors at 1.6GHz and 9MB cache
- 256GB memory
- PCI-X 133MHz I/O
- Up to eight Intel® Gigabit Ethernet ports
- Up to 16 Emulex fibre channel connections
- Configurable as up to four partitions
- SAN: 2.5 TB (shared with Master Database)

Database Tier (Scenario II – Latency App)

Master Database and Worker Database

One ES7000/one Server:

Microsoft BizTalk Tier

Two ES7000/one Servers, each configured with:

- 32 Intel® Dual-Core Xeon® 7041 processors MP at 3.0GHz and 2x2MB cache
- 256GB memory
- PCI-X 133MHz I/O
- 16 Intel® Gigabit Ethernet ports
- SAN: No SAN connectivity required
- 76GB local storage for each partition

Load Drivers

16 commodity servers, each configured with:

- Four Intel® Xeon® processors at 2.7GHz and 1MB cache
- SAN: No SAN connectivity required
- 4GB memory
- 36GB local storage

Storage

Two EMC CLARiiON CX700 storage subsystems:

- 90 36GB 15,000 RPM fibre channel drives

System Software Utilized

Database Tier

- Microsoft Windows Server 2003, Enterprise Edition
- Microsoft SQL Server 2005

BizTalk Tier

- Microsoft Windows Server 2003 Datacenter x64 Edition (ES7000/one Servers)
- Microsoft Windows Server 2003 Enterprise Edition (Commodity servers)
- Microsoft SQL Server 2005
- 11 Microsoft Outlook® Web Components
- Three Microsoft Management Consoles
- Microsoft BizTalk Server 2006, Enterprise Edition
- BizTalk redistributable CAB file (contains many of the prerequisites listed above, which the BizTalk installation process includes automatically)

Results

Using the Simple App, a sustained throughput of 1,605 messages per second was obtained with all performance metrics within acceptable limits on both the database servers and the BizTalk servers. With the Latency App, a sustained throughput of 1,156 messages per second was obtained with all performance metrics within acceptable limits on both the database and BizTalk servers.

These results demonstrate the impressive scalability of Microsoft BizTalk server in this configuration and denote a remarkable level of throughput. And in fact, they represent the highest throughput for Microsoft BizTalk ever recorded by Microsoft. Moreover, additional database optimizations, which were not implemented due to the time constraints placed on this engagement, were identified during the course of the testing. (The Microsoft team fully believes that an even higher level of throughput could easily be achieved had these optimizations been implemented.)

Scenario I – Simple App

The Simple App test was designed to show the “raw” throughput capability of the actual BizTalk Server 2006 applications. Since the application itself did nothing, however, these results are not real-world and can only be used to demonstrate BizTalk Server 2006 throughput. The results for the test of the Simple App are detailed in Figure 4.

Test Configuration	Receive Hosts	Process Hosts	Send Hosts	Orchestrations per Second
S1	1	1	1	280
S2	3	4	4	800
S3	6	8	8	1,552

Figure 4: Scenario I – Simple App test results in orchestrations per second

Due to the tight schedules and as previously noted, the team was not able to perform regression testing of optimizations discovered during the second phase of testing using the Latency App. But, implementing these optimizations would have resulted in even higher results than those shown in Figure 4.

The hardware landscape for this scenario is shown in Figure 5.

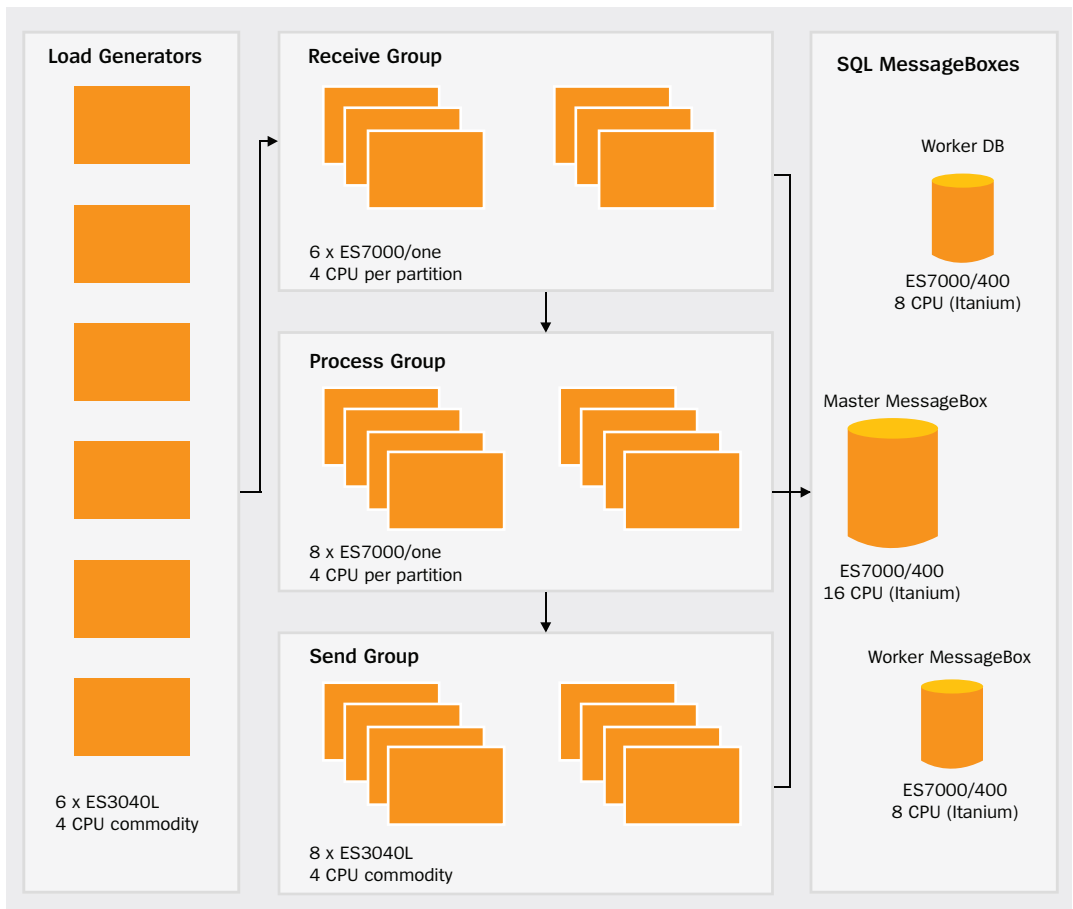


Figure 5: Hardware landscape for Scenario I – Simple App

Scenario II – Latency App

The Latency App was designed to demonstrate the throughput of BizTalk Server 2006 with a real-world application. As such, it closely models delays present in the customer’s actual payment process. The results of the test of the Latency App measured in orchestrations per second are detailed in Figure 6.

Test Configuration	Receive Hosts	Process Hosts	Send Hosts	Orchestrations per Second
L1	1	1	1	250
L2	4	4	4	684
L3	5	10	10	1,156

Figure 6: Scenario II – Latency App test results in orchestrations per second

For the L3 configuration, additional SQL optimizations were identified by the Microsoft SQL Server Team, however time constraints prevented their application. It is believed that these optimizations would have provided even better results (although the actual throughput improvements are not known) and the customer plans to implement these optimizations in the production deployment of the solution.

The hardware landscape for this scenario is shown in Figure 7.

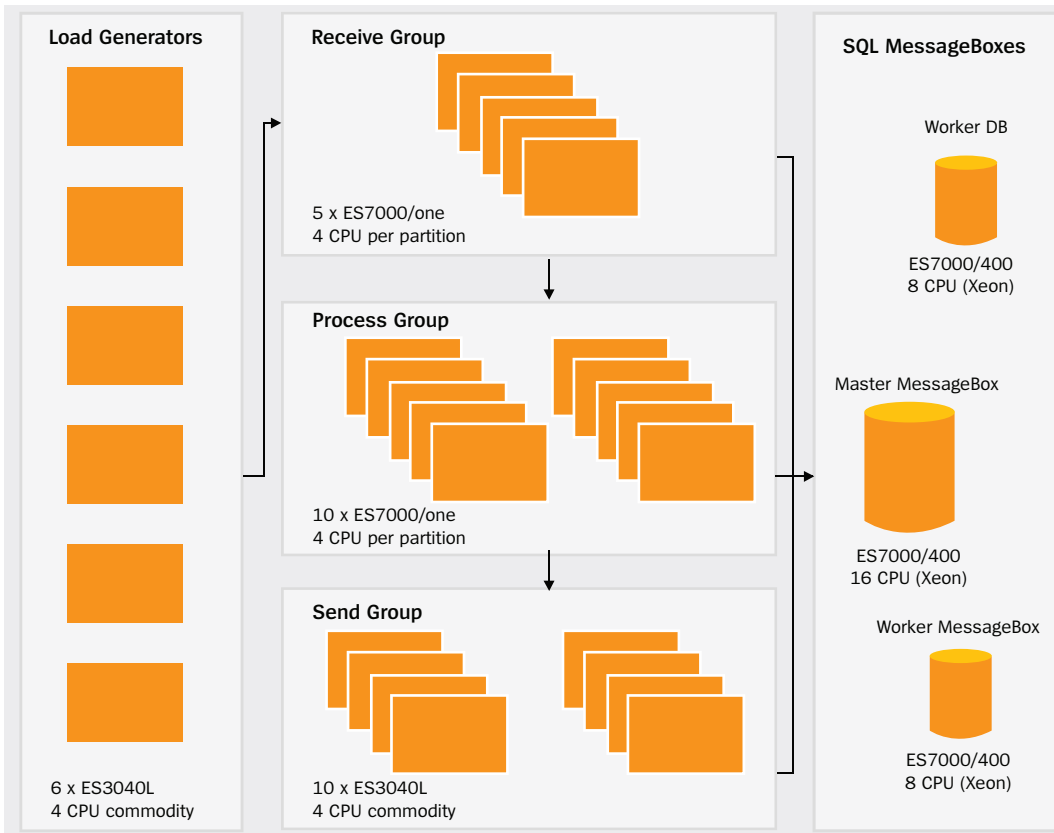


Figure 7: Hardware landscape for Scenario II – Latency App

Performance Analysis

Prior to this performance test, reports documented BizTalk Server 2006 scaling to about 200 messages (end-to-end Orchestrations) per second. The results in this lab demonstrated that almost 1,200 messages per second is achievable with a financial transaction application. As such, the results can be generalized to other applications with similar characteristics.

The performance metrics were all at an acceptable level for this engagement. Processor utilization on the database servers was high, with average utilization at around 65 percent and peaks that were over 90 percent. All of the BizTalk servers had average utilization of less than 50 percent, with one server showing a peak processor utilization of about 85 percent. Memory utilization, paging activity, and disk queue depths were all well within acceptable limits.

These metrics indicate that, in the test environment, the database servers were a potential bottleneck. As noted earlier in this paper, there were some database optimizations that were not applied due to time constraints, which may have reduced some of the load on these servers. Since the Master MessageBox server had the highest overall load, it is logical to conclude that augmenting more processors to this server – and then adding another Worker MessageBox server – could provide significant headroom for the database.

Database Servers

As demonstrated in Figure 8, processor utilization on the three database servers was relatively high. The Master MessageBox and Worker MessageBox 1 showed the highest processor utilization, which averaged about 65 percent and demonstrated peaks near 100 percent. Processor utilization on Worker MessageBox 2 was lower, with an average of about 35 percent and peak usage close to 75 percent.

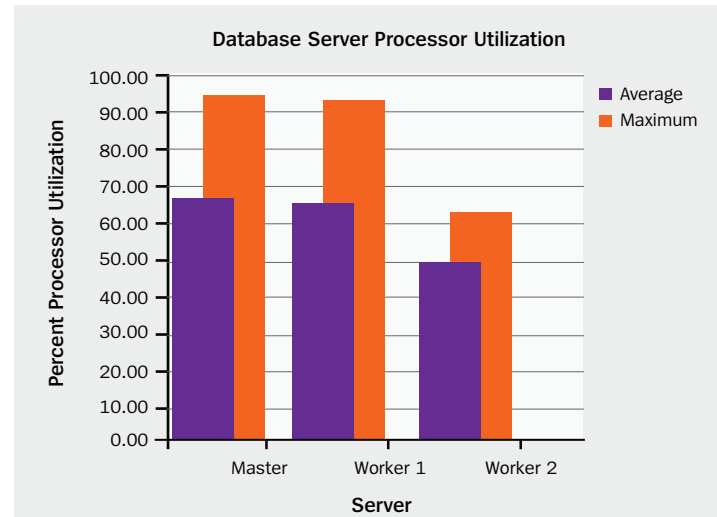


Figure 8: Database server processor utilization

The SQL Server databases running on the MessageBoxes showed a high level of throughput and achieved close to 5,000 database transactions per second on the Master MessageBox and 3,500 database transactions per second on each of the Worker MessageBoxes. The database throughput tracked closely with the processor utilization on each server – as well as across servers – indicating that the workload was evenly balanced. Figure 9 shows the transaction rate compared to the processor utilization on each of the MessageBoxes.

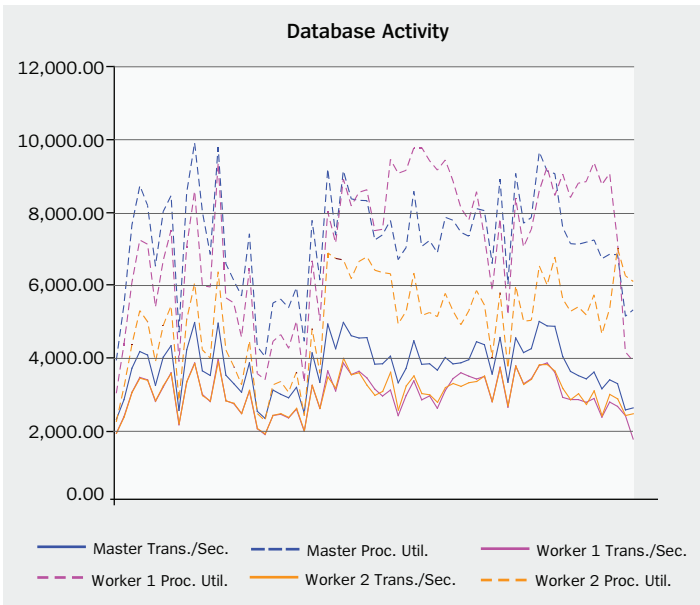


Figure 9: SQL Server transaction rate compared to processor utilization

As evidenced by the low incidence of paging, memory on the database servers was more than adequate. As depicted in Figure 10, total page faults on the Master MessageBox never exceeded 1,400 pages per second. And as shown in Figure 11, hard-page faults, which required an I/O-operation-to-disk to retrieve the requested data, were scarcely over one page per second

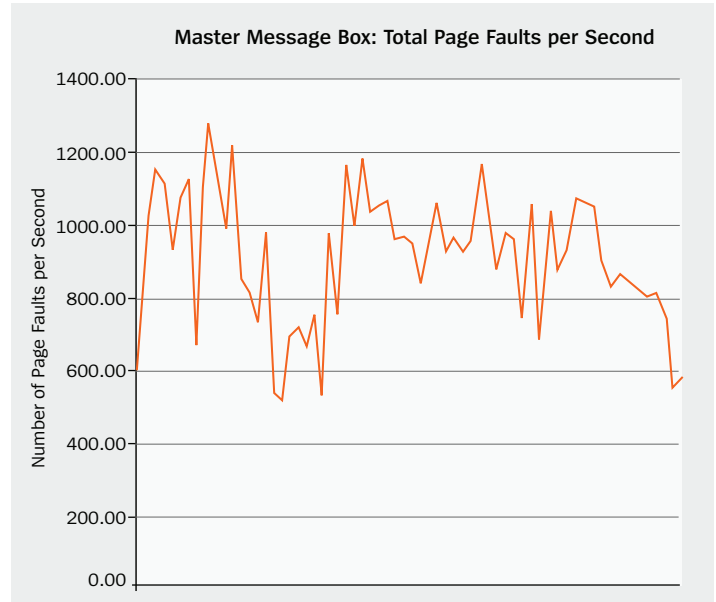


Figure 10: Master MessageBox page faults

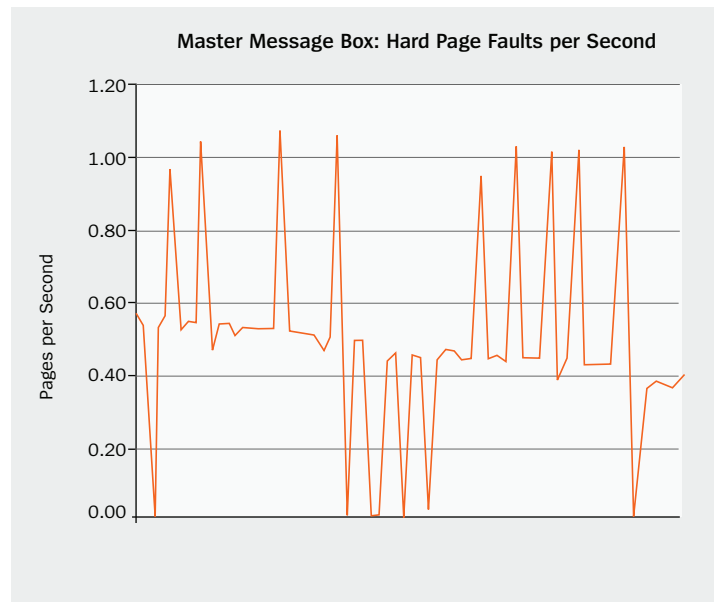


Figure 11: Master MessageBox hard-page faults

Disk queue depth on the Master MessageBox was extremely low – with none of the volumes showing anything greater than two. This is indicative of extremely efficient use of storage resources. Figure 12 shows the average and maximum disk queue depths for all of the storage drives on the Master MessageBox.

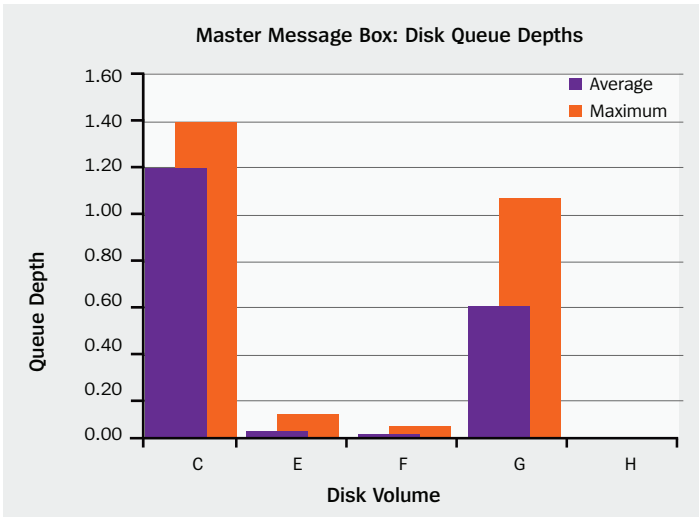


Figure 12: Master MessageBox disk queue depths

On Worker MessageBox 1 and Worker MessageBox 2, disk activity was also good, although overall queue depths were slightly higher than on the Master MessageBox. There was an anomalous peak queue depth of about 13 on Drive H: of Worker MessageBox 1, but it was brief and transient. Figure 13 shows the disk queue depths for Worker MessageBox 1, and Figure 14 shows the queue depths for Worker MessageBox 2.

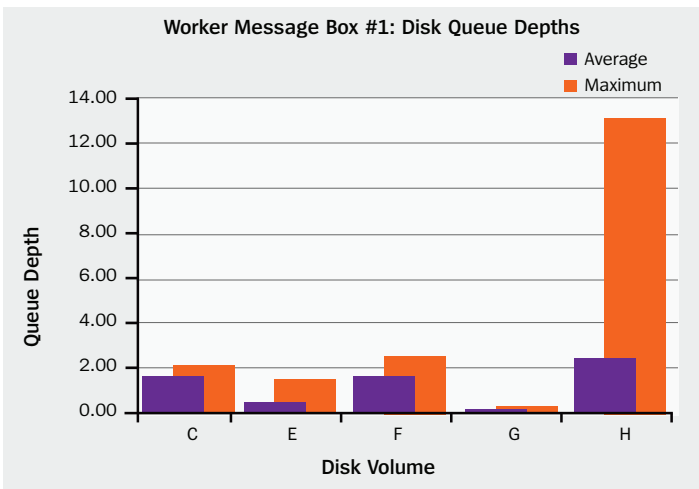


Figure 13: Worker MessageBox 1 disk queue depths

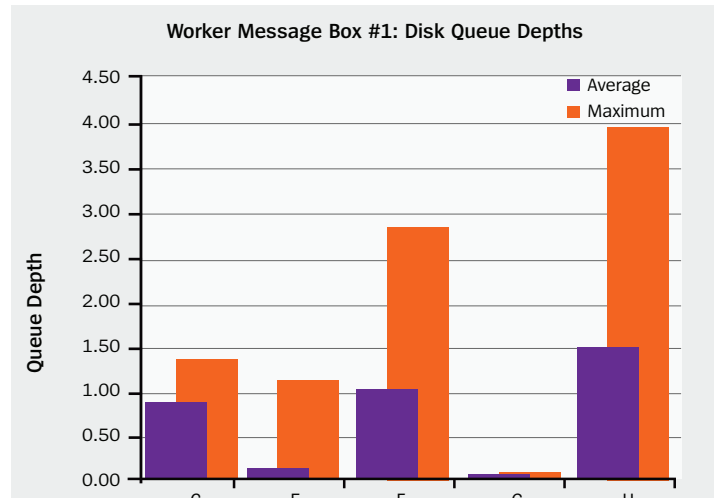


Figure 14: Worker MessageBox 2 disk queue depths

On both Worker MessageBoxes, Drive F: and Drive H: saw the most activity. But the peaks were still within accepted limits, because the disk volumes were on a high-performance storage subsystem and all volumes were striped across multiple physical disks. The accepted standard is that the disk queue depth should not exceed the number of physical disks contained in a logical volume. And in this case, Volume F: was striped across eight physical drives, and Volume H: was striped across six physical drives.

BizTalk Servers

The main performance counters examined in the BizTalk servers were the BizTalk message counters and processor utilization. The BizTalk message counters vary depending on the host. For the Receive Hosts, the counter is BizTalk: MessagingDocuments received/sec. For the Process Hosts, the counter is XLANG/s Orchestrations\Orchestrations completed/sec. For the Send Hosts, the counter is BizTalk: Messaging\Documents processed/Sec. These counters indicate the number of messages coming into the system via batch interface, the number of orchestrations being completed, and the number of messages going out of the system via the batch interface. Figure 15 shows these three counters plotted for a 30-minute run.

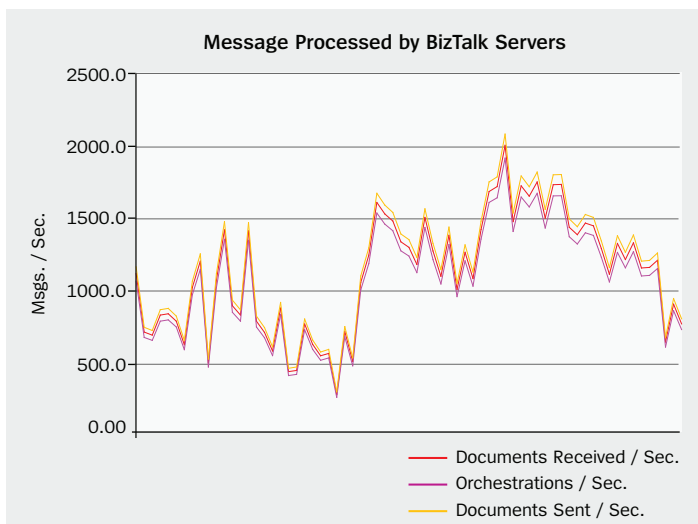


Figure 15: BizTalk server message processing

As can be seen from Figure 15, the overall average number of messages processed is around 1,100 orchestrations per second (1,116.3) with a peak of around 1,800 orchestrations per second (1,842).

The BizTalk servers do not consume large amounts of memory, and are not I/O intensive. On these servers, processor utilization was well within the acceptable range. In fact, none of the servers showed average processor utilization greater than 50 percent, and none demonstrated peak processor utilization greater than 85 percent. This indicates that there were processing resources available to handle any transient spikes in loads. Figure 16 depicts processor utilization across all of the BizTalk servers.

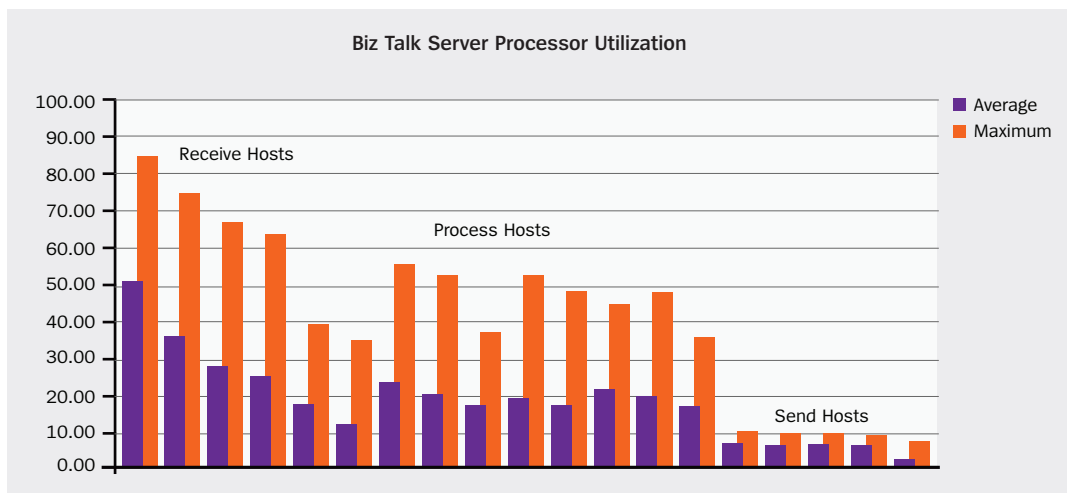


Figure 16: BizTalk server processor utilization

Conclusion

During the POC, the team endeavored to determine, and document the performance characteristics of its new solution, which was built with BizTalk Server 2006, and running on Unisys ES7000/one Enterprise Servers. In addition, the benchmark sought to confirm the use of a scale-up server architecture to minimize the overall physical and administration / management footprint of the platform.

As such, this POC demonstrates that Microsoft BizTalk Server 2006 running on ES7000/one Enterprise Servers can reach an amazing level of performance and can deftly handle a real-world, high-volume application requirement. This combination is not only extremely scalable, but also a very flexible and cost-effective solution. For financial institutions like the benchmark client, these results represent incredible performance for critical applications and processes.

About the Authors

Mike Curnutt

Mike Curnutt is a Unisys Senior Architect specializing in both Linux and Windows scale-up ES7000 environments, including server technology, storage sub-systems, and networking. Mike joined Unisys in 1995 as a Senior Consultant in the Technical Services Group in Mission Viejo, CA. In 2002, he was promoted to System Architect and was the first team member to join the ES7000 Performance Center, which was formed to spearhead large-scale benchmarks, proofs-of-concept, and system-tuning engagements for both clients and partners. In his six years with the ES7000 Performance Center, Mike has worked on a number of successful projects using a variety of products and across many different industries. Prior to Unisys, Mike worked ten years as a business owner and independent consultant specializing in mainframe databases, communications, and distributed applications. Mike has also worked for several end-user customers during his 27-year career in the industry and holds several certifications, including Microsoft Certified Systems Engineer and SAP Basis Consultant.

Michael O'Connell

Michael J. O'Connell joined Unisys in 2004 as a Senior Microsoft Architect in the Managed Services organization. Later, he joined the Microsoft Alliance team in Bellevue, WA in order to lead the Unisys and Microsoft Solutions Alliance Technology Center, which evaluates new Microsoft products and technologies and develops proofs-of-concept for Unisys solutions and enterprise clients. During his 27-year career in information technology, Michael has been employed by Microsoft, Hewlett Packard, and Digital Equipment Corporation, where he has focused on database and transaction processing software and server technology.

Contributors

Petr Kratochvil

Petr Kratochvil joined Microsoft in 2001 as Senior Consultant in the National Practices Group in Redmond, WA. He entered the BizTalk Product Group in 2004 as a Program Manager and was a founding member of a BizTalk Ranger Team, which focuses on facilitating large BizTalk implementations and developing related best practices. Prior to Microsoft, Petr worked at Ncompass Labs, which was acquired by Microsoft in 2001 and is the company that developed the Microsoft Content Management Server (which powers some of the largest high-traffic websites on the Internet).

Rob Steel

Rob Steel joined Microsoft in 2003 as a Principal Consultant with Microsoft Consulting Services in Dallas, TX. In 2006, he joined the BizTalk Product Group as a Principal Program Manager on the BizTalk Ranger Team. In this role, he primarily focuses on conducting BizTalk Performance Lab engagements with strategic customers, performing research into BizTalk Server solution performance, and producing performance-related content for the BizTalk community. From 1989 to 2003, Rob worked as an independent consultant doing Win32, COM, .NET, and BizTalk Server development.

About the Unisys and Microsoft Solutions Alliance

Unisys and Microsoft deliver secure, scalable solutions that harness the power of people and technology so you can realize the value of Enterprise Visibility – total alignment among Business Strategy, Process, Applications, and Infrastructure. Unlike other companies that deliver on isolated technology projects, Unisys and Microsoft help you see and understand all the technology influences and interactions that affect your business. Together, Unisys and Microsoft provide proven industry solutions, the visibility to align business and technology, and dedicated expertise for delivery assurance.

For more information, contact your Unisys representative.

Or call:

1-800-874-8674 ext 365 (U.S. and Canada)

00-1-585-487-2430 ext 365 (Other countries)

© 2008 Unisys Corporation. All rights reserved

Unisys and the Unisys logo are registered trademarks of Unisys Corporation. Microsoft, BizTalk, Outlook, and Windows are registered trademarks of Microsoft Corporation. Intel, Itanium, and Xeon are registered trademarks of Intel Corporation. EMC and CLARiiON are registered trademarks of EMC Corporation. Linux is a registered trademark of Linus Torvalds. All other brands and products referenced herein are acknowledged to be trademarks or registered trademarks of their respective holders.

Printed in US America 6/08



BL100200-100