Performance Analysis
February 2006

Using IBM® BladeCenter® with Microsoft® .NET 2.0 Applications and Microsoft SQL Server 2005

By Chris Floyd
IBM Systems and Technology Group

Abstract

In the world of application servers, moving to environments that can easily scale out is increasingly emphasized. At the same time, the desire to consolidate resources for ease of management, physical space, and power consumption is often at odds with the actual properties of traditional rack-optimized scaled-out application server clusters.

The IBM BladeCenter is a hardware solution that provides an infrastructure capable of application server scale-out without compromising on ease of systems management, physical space, or power requirements. BladeCenter management tools provide the ability to easily deploy and manage up to 14 blade servers such as the HS20, HS40, or LS20. IBM Director, which ships with BladeCenter, unleashes the power of BladeCenter, giving you comprehensive remote management from a single graphical console. IBM Director automates and simplifies IT and networking tasks, letting you deploy, configure, manage and maintain up to hundreds of blade servers. The BladeCenter architecture provides robust systems interconnect and the ability to support the storage and disk I/O requirements, as well as the network throughput, typical of enterprise application servers. Using BladeCenter, a large scale-out application server solution can be easily and cost-effectively realized and managed.

This paper presents a sample configuration illustrating IBM BladeCenter performance capabilities when using Microsoft .NET applications and Microsoft SQL Server 2005. A single HS20 blade was used to host the Microsoft .NET 2.0 Framework and application server environment. Typical customer implementations using this framework include technologies such as ASP.NET, C++, Visual Basic, Microsoft Internet Information Services (IIS) HTTP server, Web services, MSMQ messaging applications, database connectivity with ODBC, and Microsoft's Distributed Transaction Manager (DTC). As a further proof point of its versatility, the Bladecenter chassis also includes one HS40 blade, which is used as the database server with the SQL Server 2005 DBMS. An IBM TotalStorage® DS4300 disk subsystem is used for the HS40 database storage and I/O requirements, and is connected via a QLogic Fibre Channel adapter (daughter card) and QLogic Fibre Channel switch module inside the BladeCenter chassis.

BladeCenter Architecture and Features

The BladeCenter chassis design allows for high-density, high-performance, hot-swappable server enclosures such as 2- and 4-way Intel® processor-based and AMD Opteron processor-based blades. Each BladeCenter chassis supports up to 14 two-processor blades or 7 four-processor blades. The blades support the Microsoft Windows Server 2003 and Windows 2000 operating systems, enabling customers to migrate existing applications or deploy new Windows environments on this platform. IBM Director, an advanced central management suite, allows customers to easily control and manage every blade in the BladeCenter chassis, as well as the installed modules (e.g., Fibre Channel and network switch modules).

IBM has opened the design specification for BladeCenter, which has resulted in a large number of options, such as Cisco and Nortel network switch modules, high-speed blade interconnect expansion cards, Fibre Channel expansion cards, SCSI storage options, PCI I/O expansion units, Topspin InfiniBand® switch modules, and much more. BladeCenter also offers a variety of disk I/O solutions by supporting the high-performance IBM TotalStorage family of products and low-cost, network-attached storage.

Microsoft .NET 2.0 / SQL Server 2005 Application Test Scenario

The test performed demonstrates the performance capabilities of BladeCenter using an HS20 blade as an application server, and an HS40 as a database server, as well as the capabilities of the Microsoft .NET 2.0 Framework and Microsoft SQL Server 2005. The HS20 blade was configured with Microsoft Windows Server 2003 and Microsoft IIS 6.0 HTTP server using the Microsoft .NET 2.0 Framework. The application model represents a business-to-business application with functions such as order processing, stock/inventory warehouse processing, customer service operations, and inventory/catalog query interactions. The application consisted of several .NET Web service applications (accessed via IIS), and several message processing applications using Microsoft MSMQ for the message store. Each Web service interaction and messaging interaction accesses the Microsoft SQL Server DBMS installed on the HS40 blade. Many of the transactions involved two-phase commit operations (distributed transactions) between the message store (MSMQ) and the SQL Server DBMS. All transactions between the driver (workload generator) systems and the .NET application server used SSL to demonstrate an additional layer of complexity and security.

Figures 1 and 2 show models of the hardware environment used, as well as an overview of the Microsoft software technologies used in the test application.

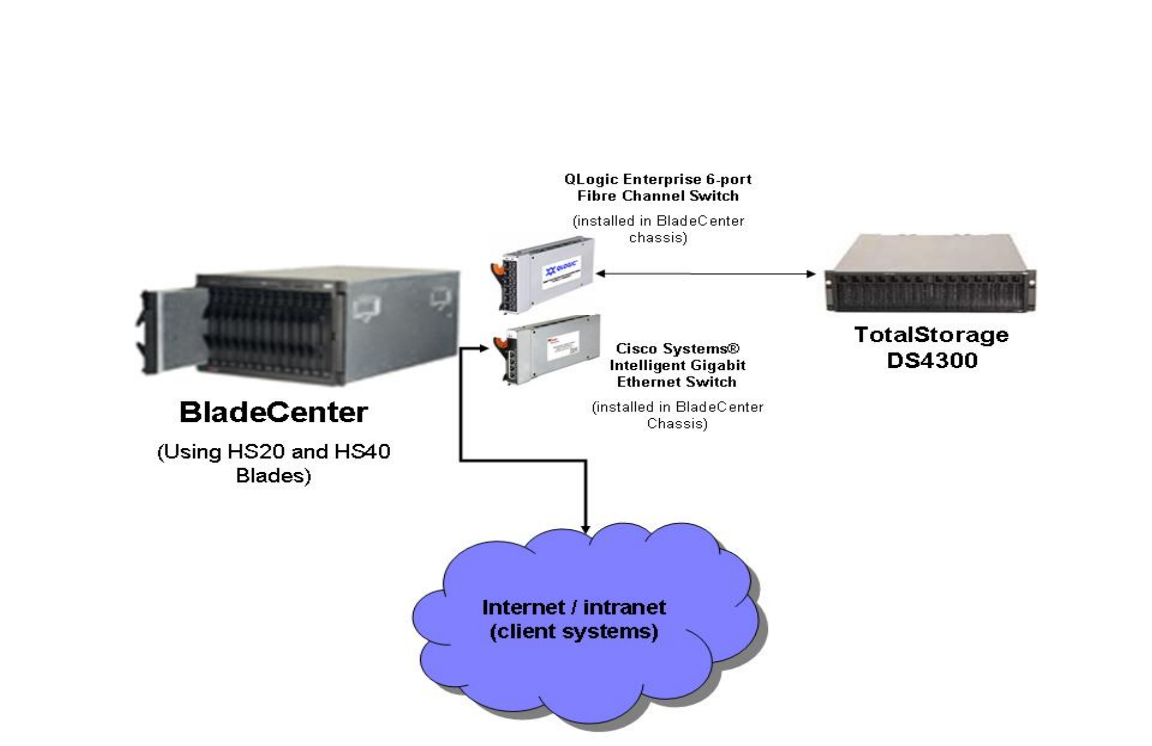
QLogic Enterprise 6-port Fibre Channel Switch
(installed in BladeCenter chassis)
TotalStorage DS4300
Cisco Systems® Intelligent Gigabit Ethernet Switch
(installed in BladeCenter Chassis)
BladeCenter
(Using HS20 and HS40 Blades)
Internet / intranet (client systems)

Figure 1: Hardware Test Environment Model

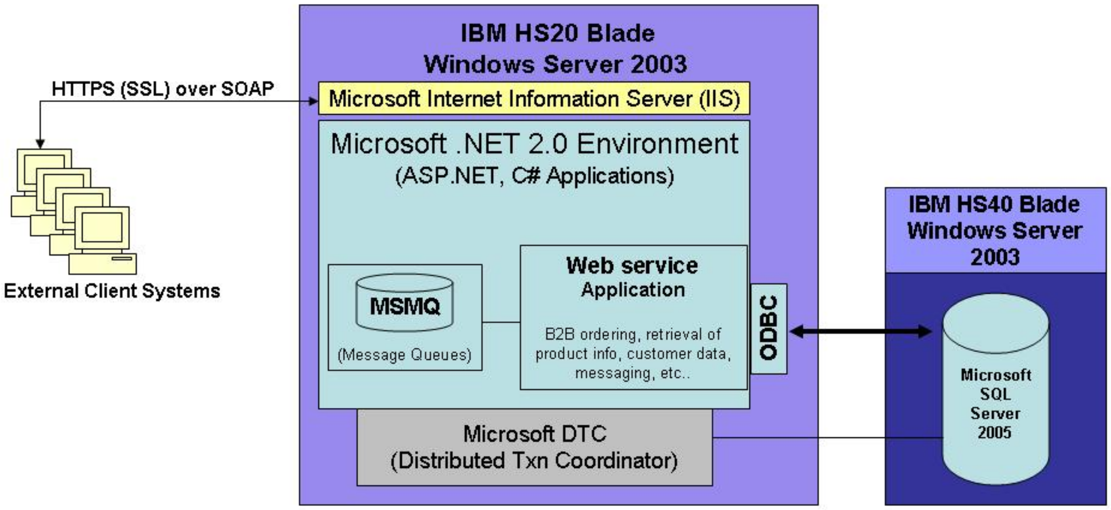
IBM HS20 Blade
Windows Server 2003
HTTPS (SSL) over SOAP
Microsoft Internet Information Server (IIS)
Microsoft .NET 2.0 Environment
(ASP.NET, C# Applications)
External Client Systems
MSMQ
(Message Queues)
Web service
Application
B2B ordering, retrieval of product info, customer data, messaging, etc..
ODBC
IBM HS40 Blade
Windows Server
2003
Microsoft
SQL
Server
2005
Microsoft DTC
(Distributed Txn Coordinator)

Figure 2: Software Test Environment Model

Test Scenario Results, Observations and Tuning Details

Results

The workload generator for the test scenario used 60 driver threads, which generated enough workload (i.e., Web service requests) to drive the HS20 blade .NET 2.0 application server to near maximum CPU utilization (96%). At this level, the HS20 was capable of handling ***420 Web service requests per second***, with an ***average response time of ~150ms***. The HS20 also ***simultaneously processed 231 messages per second*** from the local MSMQ message queues.

The resulting load on the HS40 blade running the Microsoft SQL Server DBMS generated an 18% CPU utilization. The Microsoft Distributed Transaction Coordinator (DTC) on the HS40 system processed **244 distributed transactions per second**.

Figure 3 highlights the performance throughput characteristics achieved using this configuration:

IBM BladeCenter HS20 / .NET Application Server Results	
Processors	2 x 3.6GHz Intel Xeon (HyperThreading Enabled)
CPU Utilization	96%
Throughput	420 Web Service Requests / Sec
Average Interaction Response Time	150 ms
MSMQ Messages per Sec	231
Network Traffic	35.6 MB / Sec (285 Mbit / Sec)

Figure 3: HS20 / Microsoft .Net 2.0 Application Server Results

IBM BladeCenter HS40 / Microsoft SQL Server Results	
Processors	4 x 3.0GHz Intel Xeon (HyperThreading enabled)
CPU Utilization	18%
Distributed Transactions / Sec (DTC)	244
Network Traffic	2.3 MB / Sec (18 Mbit / Sec)

Figure 4: HS40 / Microsoft SQL Server Results

Observations and Tuning Details

The HS20 blade .NET 2.0 application server performed quite impressively with relatively few configuration and tuning changes. The Web service application was designed entirely using Microsoft Visual Studio 2005. No changes related to compilation or intermediary files were made to the Web services or message queue processing applications. The same can be said for the HS40 blade running SQL Server; in fact, no changes were made to the SQL Server DBMS options (e.g., sp_configuration parameters).

For the .NET 2.0 application server system, a few changes to the global .NET machine.config file were necessary to eliminate potential threading and TCP/IP connection bottlenecks.

Tuning Changes Made to the .NET machine.config Application Server

For the HttpRuntime section, the MinFreeThreads setting was changed to 176, and the MinLocalRequestFreeThreads was changed to 152. This increases the number of worker threads and completion port threads that must be available to start a remote request or a local request from within the ASP.NET process. If these threads pose a potential bottleneck, Microsoft recommends setting the values of the minFreeThreads parameter to 88*N and the minLocalRequestFreeThreads parameter to 76*N, where N is the number of processors on the application server system.

For this application workload, since the Web service applications executing on the .NET 2.0 system performed Web service calls to other external Web services, the maximum number of allowable “outbound” TCP/IP connections to a given Web site needed to be increased. If a given Web application needs to make a synchronous or asynchronous Web service call to another Web application to retrieve data, by default, the framework limits the outbound connections to a given site to two connections. If it is necessary to increase this value, Microsoft recommends setting the value of the maxconnection parameter to 12*N (where N is the number of CPUs in the system).

Finally, to increase the number of ASP.NET processing threads, the maxWorkerThreads and maxIoThreads value was increased to 100.

For more details about the rationale for these .NET Web service application changes see:

http://support.microsoft.com/kb/821268/en-us

Summary

Given the impressively high performance for a relatively complex business-to-business application for a *single* HS20 blade .NET application server and HS40 blade running Microsoft SQL Server, it is easy to see how a more heavily populated BladeCenter system would provide a scale-out, end-to-end, high-performance, enterprise-class Microsoft solution platform. Dramatic code-development, deployment, and management cost savings are realized by embracing managed-environment applications hosted by the .NET 2.0 Framework. Similarly, the enterprise-class, high-performance and award-winning management features of BladeCenter provide cost savings for managing and deploying the hardware solution. The ability to simply and effectively add more .NET application servers as your company needs grow provides for a “pay as you grow” infrastructure.

With available options such as layer-7 gigabit switches, Fibre Channel SAN and network-attached storage (NAS) disk connectivity, high-speed blade interconnects, ease of management, space and power, the IBM BladeCenter provides an ideal, all-around cost-effective solution.

IBM Systems and Technology Group
3039 Cornwallis Road
Research Triangle Park, NC 27709

Produced in the USA
01-06

Warranty Information: For a copy of applicable product warranties, write to: Warranty Information, P.O. Box 12195, RTP, NC 27709, Attn: Dept. JDJA/B203. IBM makes no representation or warranty regarding third-party products or services including those designated as ServerProven or ClusterProven.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the 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.

Performance is based on measurements using industry standard or IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve performance levels equivalent to those stated here.

IBM reserves the right to change specifications or other product information without notice. References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates. IBM PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.