



OS/390 Guide to e-business Connectors: Third Edition

This third edition of the e-business Connectors Guide provides more focus on the Java based connectors and an update on scalability. This white paper is available at

<http://www.s390.ibm.com/ebusiness/connectors/>.

Both technical explanations and customer examples are used to give performance advice. This edition supplements the prior edition. We expect to update this white paper periodically. For a higher level business view, please refer to the *OS/390 Web-Enablement Overview, GF22-5138*, paper.

Contents
2 Overview
5 Base Software Considerations
6 "2-Tier" versus "3 Tier" Considerations
9 Web Enablement Performance Overview
11 Programming Models
13 Language Choice Considerations for Performance
14 Server Scaling: WLM, Parallel Sysplex
16 OS/390 Release Considerations for Performance
19 Customer Scenarios
20 CICS
21 IMS
22 Second Generation (Java) Connectors
24 LINUX/390
25 Additional Information on Connectors

Overview

The objective of this white paper is to give OS/390® customers advice on how to choose the e-business connectors for their existing subsystem (CICS®, DB2®, IMS™, and MQSeries®) applications and data. “Connectors” in this paper refers to the software programs that connect a Web application or browser to an existing application and data. The key considerations for the e-business application being architected include: performance requirements (e.g. transactions per second), security requirements (e.g. SSL and its associated performance), “2-Tier” versus “3-Tier” considerations, existing levels of OS/390 and the subsystems (including whether an upgrade can be considered as part of the total solution), and the programming skills available for this solution (which could include contracted services). Some customer scenarios are provided as examples.

This white paper focuses on the “2-Tier” solution where the Web server, Web application server and existing application and data reside on an OS/390 server. The WebSphere™ Common Connector Framework (CCF) connectors are designed to support development and deployment on middle tier servers and portability to OS/390 without change. The deployment may benefit from additional local optimizations. S/390’s qualities of service are the added value of the deployment on OS/390. IBM has worked closely with JavaSoft™ and the industry to define a “Common Connector Framework.” An official JavaSoft specification drawn from this collaboration is available on their Web site.

The connectors have been categorized into three groups. A Web-based tool (*The OS/390 Planning Assistant for e-business*, www.s390.ibm.com/os390/wizards/ebiz) has been developed to help guide you through the decision process for picking the right connectors based on the criteria outlined in this white paper.

First Generation (API) connectors which have been available for some time, are characterized as being very quick to implement with existing programming skills. These include the API-based connections using CICS Web Support (CWS), IMS TCP/IP OTMA Connection (IMS TOC), Net.Data®, and CWS with 3270 Bridge and Templates. API connectors tend to have less standard tooling,

Goal: *Help customers choose how to access existing data and applications to meet their e-business objectives with the appropriate expectations.*

Audience: *This white paper is intended for technical personnel (architects, systems programmers, etc.).*

Frequency: *The intent is to update this paper twice a year or as necessary.*

less portability and typically better performance than server-side Java connectors. They are less focused on application re-engineering or technology upgrade and more on optimizing Web enablement to their existing application and data.

Second Generation (Java) connectors are characterized by the use of Java® and tooling. For relational databases, JDBC and SQLJ are provided. CICS, IMS and SAP have developed respectively the CICS Transaction Gateway (CTG), the IMS Connector for Java and the VisualAge® for Java Connector for SAP R/3 based on the IBM developed Common Connector Framework (CCF). This framework is supported by the Java application development tools of WebSphere, specifically VisualAge for Java. Host on-Demand is another solution for 3270-based applications. These connectors are generally available now.

Second Generation connectors are more focused on standards, tooling, portability and beginning the process of application re-engineering. Performance/scalability is currently below that of corresponding First Generation counterparts (e.g. CTG versus CWS, IMS TOC versus IMS Connector for Java, JDBC versus Net.Data).

Third Generation Enterprise Java connectors introduces Enterprise JavaBeans™ (EJBs) and the “Java 2” family of protocols (e.g. JDBC 2.0). The third generation connectors are an evolution of second generation with enhanced transactional and security context at the thread level, and data persistence built into the application programming model. Even more of the S/390 platform’s QoS (Qualities of Service) are provided to the application transparently. Examples of these key QoS advantages are Workload Management for Address Space management, RRS (Recoverable Resource Services) for distributed transactions and Parallel Sysplex® clustering for data sharing.

The connectors emphasized in this report are the Java based Common Connector Framework (CCF) connectors, namely the CICS Transaction Gateway

(CTG) and the Connector for Java. The CCF Connectors are part of IBM's Application Framework for e-business that is being used to focus our e-business activities. We will also discuss the relational database access connectors, JDBC and SQLJ, which are the strategic way to access both IBM and non-IBM relational data.

The IBM WebSphere family of products is a very powerful set of facilities for developing Web-enabled applications that combine a broad set of industry standard programming interfaces (e.g. Java servlets and Enterprise JavaBeans [EJBs]) with a set of application development and life cycle management tools to enhance the programmer's ability to rapidly compose business objects into e-business applications for deployment across multiple server platforms. With the delivery of the CICS Transaction Server and the Component Broker technology delivered as part of WebSphere Enterprise Edition on OS/390, EJBs will become the preferred connector technology. Combined with rapidly improving Java performance across the platforms, IBM believes solutions based on Java technology will quickly become the best total solution for enterprises requiring access to existing data and applications. As a result, many of our performance measurements and supporting services (e.g. UNIX® System Services, TCP/IP, JDK) are being focused on making this environment best of breed.

The CCF connector support is delivered in three parts:

- 1. VisualAge for Java provides the base connector classes,*
- 2. The WebSphere Application Server for OS/390 provides the optimized OS/390 classes, and*
- 3. The connectors themselves (CTG, IMS Connector for Java, etc.).*

Base Software Considerations

For a variety of reasons, current S/390® customers may need to consider tactical alternatives to full-scale WebSphere development and deployment (e.g. because of time delays in updating base software).

You have several considerations to make when choosing how to Web-enable. The first consideration is the subsystem (e.g. CICS, DB2, IMS, MQSeries) you need to connect to and what subsystem level you are currently running. For example, some of the Generation 2 connectors are not supported on older versions of CICS. Another consideration is the level of business application function implemented in the subsystem and the need for new application function to create a Web-ready application.

Other business considerations (e.g. application programmer skills, security/isolation, etc.) may not permit you to upgrade the subsystem to the most current level in time to establish a presence on the Internet and provide connectivity to production applications.

There are a number of ways to address this, such as running newer levels of software in a separate logical partition (LPAR) along with the IBM HTTP Server, and if needed, the WebSphere Application Server. CICS Transaction Server is particularly good in this area and offers a number of methods of interconnecting systems. Other potential solutions might include using MQSeries between LPARs.

You should consider implementing new business function in the WebSphere Java environment while re-using existing business function in the subsystem environment. The e-business connectors facilitate this by enabling Java applications to access applications and data in the subsystem environments. In general, the level of the subsystem is not a major performance consideration as it relates to the connection to the Web, but it may influence or limit the specific connectors that may be selected. Assessment of the performance / scalability of a specific connector given the application model (e.g. 3270 transactions) as well as a view on migration (e.g. EJBs) are key factors to be considered before selecting the best connector strategy to use.

The next thing to consider is the level of operating system in production now (and the next level planned, and when). In addition to functional dependencies that may require a certain level of OS/390, there are performance considerations of the underlying components (TCP/IP, Web server, Java, etc.) to factor in when selecting the best connector. One example would be the level of security and related performance required. OS/390 V2R6 with the IBM HTTP Server for OS/390 has enabled the use of hardware encryption (over software-based encryption) to improve SSL performance. R9 adds support for the new PCI crypto adapter, providing dramatic increase in SSL scalability. Another factor to consider is the use of the Workload Manager (WLM) to improve the scalability of some of the connector solutions. Some connectors are limited by storage, threads, or back-end connections to the number of connections that can be supported in a single address space. Using the WebSphere Application Server in scalable mode enables these applications to scale to significantly higher transaction rates. WAS 1.2 at a minimum is strongly recommended for scalable WLM mode. WebSphere 3.02 adds session state sharing (via DB2) and runs on R8 and above. WebSphere 3.02 provides current levels of industry standard Servlets and JSPs as well as consistency with the tooling and run-times on the workstation platforms. It also provides a significant performance improvement over WAS 1.2, both in terms of scalability and efficiency. Your performance and security attributes will be related to the level of the operating system on which you plan to deploy the connector.

“2-Tier” versus “3-Tier” Considerations

Many customers start out in the Web environment by simple page serving on a middle tier platform (UNIX or Windows NT®). As their requirements grow, such as connecting to back-end transactions or increased security concerns, the requirement for S/390's qualities of service becomes greater.

While most Web applications are inherently 3-tier applications, consisting of presentation, Web logic+connectors, and business logic, for the purposes of this

document, 2-tier vs. 3-tier is a discussion about hardware server tiers, not application tiers.

Some of the connectors are very portable (e.g. CCF) across multiple WebSphere platforms. Others that use distributed connections to the back-end subsystems have different interfaces when they run natively on OS/390 versus running on a middle tier server. Three-tier deployment, application development, and operational management characteristics of these configurations may be significantly different. Some multi-tier connector configurations may require both TCP/IP and SNA to be defined. In addition, there are some security considerations to take into account when deploying on a middle tier server. In the 3-tier model, security information must be configured on the middle tier and in some cases, security credentials must be communicated between the middle tier and S/390. Without physical network security or encryption of the connection, information will be transmitted in the clear.

The 2-tier WebSphere model provides for development of Java applications using VisualAge tools, simplified operational management within a single environment, improved scalability, and the mapping of security credentials by the Web server (working with the OS/390 Security Server) to the execution thread of the connector. The actual user ID/password required by the subsystem may require a different flow, but this is contained within the OS/390 environment. SSL communication between the browser and the Web server on OS/390 permits all client communication to be secure. The network protocol between the client and the server is all TCP/IP in the 2-tier model.

3-tier solutions can be very effective solutions. They don't require a change to existing subsystem or OS/390 levels. They leverage a variety of Web servers and

the applications they support. They provide scalability and/or more cost effective solutions for functions which do not require OS/390 qualities of service (QoS) such as Caching or CPU-intensive Java translation of data formats.

The key point is that development of new application business logic, especially when transactions or mission critical data access are involved, should be deployed on the platform where the required QoS are found.

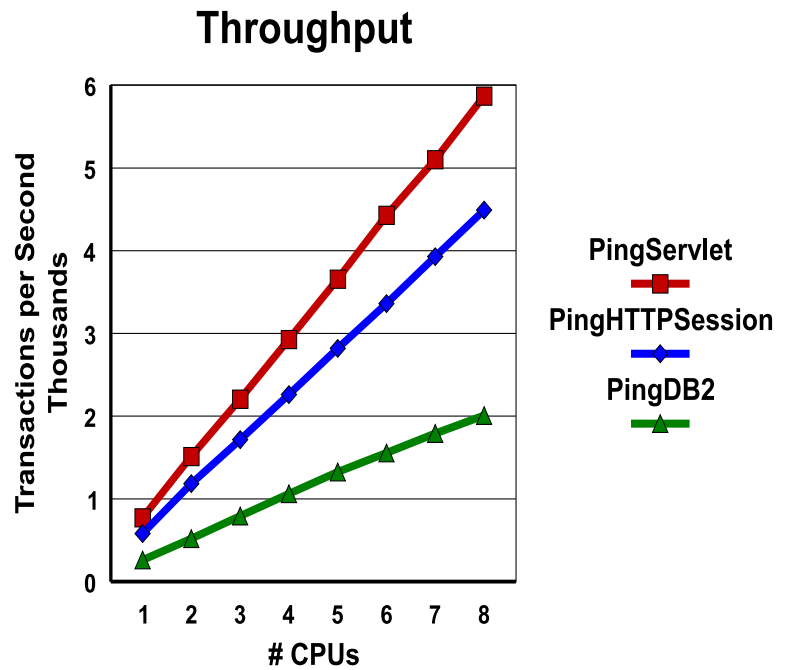
Skills Assessment

Once you understand what connector options exist for you, along with relative performance levels, you have to consider what programming skills exist in your shop and what tools exist to assist you to develop the required support. If you are at an OS/390 and subsystem level where WebSphere and Java perform well and you have Java skills, the preferred direction is use of VisualAge for Java with the Common Connector Framework. These are the configurations for which IBM will produce performance reports with frequent updates to assist in projecting throughput and assessing capacity planning. The key decision point is whether the current and/or expected performance improvements over time (including new hardware) will keep pace with your performance requirements. If you are not at a required level (OS/390 subsystem) or your performance requirements are already beyond the Java connector performance capabilities, then you need to as-

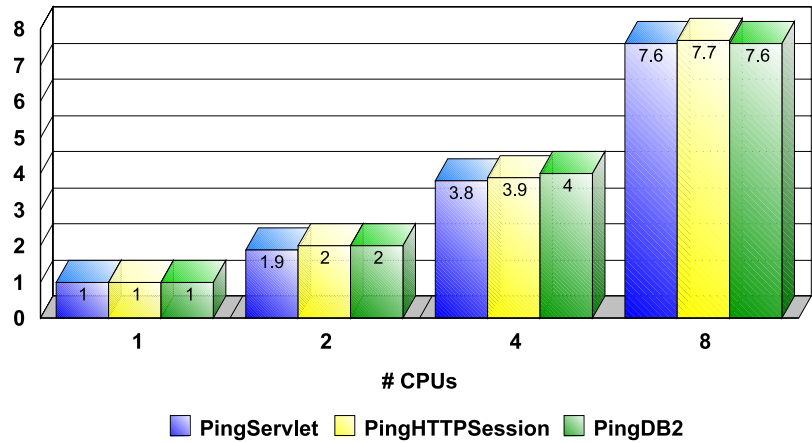
sess your options and skill requirements (e.g. develop/use GWAPI (Domino Go Webservice API) connectors in COBOL or C). If possible, look for connectors with application development tools to facilitate development as well as connectors that will run on multiple versions of the subsystem.

Web Enablement Performance Overview

Many customers approach Web enablement of existing transactions and data in an iterative fashion. They first look for the simplest method that doesn't require much investment or change to the existing infrastructure, including the operating system, the subsystem or the application itself. Tooling is usually the preferred route where the skills of the person required are more widely available. Security is almost never a concern initially although it becomes an issue when the new applica-



SMP Scalability



tion is prepared to go online. Performance and scalability are usually the last items considered and, unfortunately, most often only when they become a problem. The chart below indicates excellent scaling of the Web Application Server 3.0.2 when increasing numbers of CPUs are added to the configuration.

The main concern for most enterprise customers looking for business advantage from expanding application access to the Internet, or using Internet technology to re-engineer their business is the ability for those applications to scale as demand increases, as this is usually the motivation for investing in Web enablement. The ability to add hardware and realize proportional gains in throughput commensurate with that investment is a fundamental expectation and

a traditional strength of OS/390 subsystems (e.g. CICS, IMS, DB2). This includes scaling over a Parallel Sysplex configuration when the volume of transactions and/or data exceed the ability of one system to handle the load. This is especially true in e-business, where capacity peaks are typically significantly greater than in traditional intranet environments.

The next performance related concern to raise is the efficiency of the new connections, as they require additional capacity over that required for existing intranet connections. This is required for the additional layers of middleware (security, data translation, protocol conversion, etc.) needed to complete an end-to-end transaction, where the client is usually a Web browser. There are different models in the industry of how these connections are provided and they all have different performance characteristics which need to be understood further. The net effect is that the additional cost (hardware and software) must be competitive with other alternatives. Two-tier solutions, especially when a large amount of data is involved, tend to be more efficient (and somewhat more secure) than distributed, multi-tier solutions. However, there are cases when the interactions and data interchange are minimized and the middletier(s) are acting as simple dispatchers, where they could be more cost effective without sacrificing data integrity (e.g. no intermediate session state data which must be recovered on failure). The client protocols (e.g. HTTP, IIOP, RMI, 3270, etc.), the Web application server model (CGI, ASP, JSP/servlet, EJB, etc.) and the language choice (C, C++, Java, PERL, etc.) all have an impact on overall efficiency.

Additional factors affect the choice of application model and platform selection, such as availability, application/data integration, and security/integrity. They are not addressed specifically in this version of the white paper.

Programming Models

•**Java Servlets:** *Servlets are modules that extend request/response-oriented servers, such as Java-enabled Web servers. Servlets are to servers what applets are to browsers. Unlike applets, however, servlets have no graphical user interface. Servlets can be embedded in many different servers because the servlet API, which you use to write servlets, assumes nothing about the server's environ-*

ment or protocol. Servlets have become most widely used within HTTP servers; many Web servers support the Servlet API. Servlets are an effective replacement for CGI scripts. They provide a way to generate dynamic documents that is both easier to write and faster to run. Servlets also address the problem of doing server-side programming with platform-specific APIs: they are developed with the Java Servlet API, a standard Java extension.

• **JSP (Java Server Pages)** were initially an attempt to embed Java into static HTML pages, making the Java language like a scripting language for dynamic Web pages. This effort has matured to supporting both embedded Java and JSP tags (much like ASP) to support dynamic content. In WebSphere AppServer, and all other JSP supported Web/App Servers, when a request is made for a JSP page, the page is processed and converted into a dynamically constructed servlet. This servlet is compiled and loaded into the server much like any other servlet. This processing is called page compilation. The dynamically constructed servlet is called implicitly from then on, unless the JSP file has been changed on disk, in which case the JSP is re-page compiled.

There are two basic models for using JSP:

- An ASP-like model where a request is made for the JSP page and the page compiled servlet is run with dynamic content filled in.
- A servlet to JSP model, where the client calls a servlet which generates the dynamic content and then calls a JSP page to be sent in response to the client request. This JSP page gets the dynamic content from the original servlet and then sends the response to the client. This has the benefit of having the creator of dynamic content (the called servlet) be independent of the user of dynamic content (JSP page).

- *Enterprise JavaBeans - The key to the enterprise Java programming model is the concept of container-driven components.*

A container is a software run-time environment that provides common services on behalf of the Java components that run in the container. The container-driven component model allows the developer to defer the execution of critical computing tasks – such as maintaining a secure environment, executing transactions and persisting data to a resource manager– to the run-time containers. A business programmer merely specifies what services are required by an enterprise Java component, leaving the implementation of these services to the container. No plumbing required! Enterprise Java provides an environment in which business programmers with Web-oriented skills can produce meaningful, advanced enterprise applications without having platform-specific coding expertise.

The enterprise Java programming model defines three container-driven Java components: Java Server Pages (JSPs) to facilitate the visual, or client-side aspects of an application; Java Servlets, which are ideal for controlling the flow of execution; and Enterprise JavaBeans (EJBs), the powerful object-oriented technology for representing business logic and facilitating the access to non-Java systems and data. Enterprise Java also defines a set of useful services and connectors for direct access to back-end systems. The enterprise Java programming model is flexible enough to allow any combination of containers, services and connectors to be used in building an e-business application, on one, two, or "n" physical tiers. The use of enterprise Java containers very clearly separates the roles of business application developer, container provider, and application installer.

Language Choice Considerations for Performance

Most existing CICS/IMS applications are written in COBOL. Most CGI/plugin applications/gateways are written in C. As a result, they experience the additional overhead of Language Environment® (LE) transitioning between the two languages. Since they are also Posix (Pthread) applications, they have the limitations on the number of threads that can run in a process based on some below the line

storage limits. There is also character code page translation between ASCII and EBCDIC for the Web to incorporate into the existing transaction model.

The newer applications, generated by tooling, use the Java language, which presents a number of challenges. The first is that Java is an interpreted language, which requires translation of the byte codes as opposed to being compiled prior to execution. There exists a Just In Time (JIT) compiler, which will compile the Java code after a number of accesses to the classes. Java's native code page is Unicode, which creates additional overhead to bridge the environments. Another issue is linkage, where object oriented code typically creates more levels of nesting (methods call methods which call methods).

Then there's the issue of garbage collection, which is invoked periodically to deal with storage within the process. S/390 has enhanced the standard Java Garbage Collection by not requiring the process to be suspended while it's running. There's also the overhead to initialize the JVM, which is experienced on each process initialization. Servlets running on WebSphere for OS/390 do not suffer much from JVM initialization as OS/390 runs them in-process. There are a number of optimizations being created within Java and the supporting environment to reduce these overheads to make Java much closer to the current languages. However, for the same function, Java can be viewed currently as at least 2x the cost of the procedural languages, such as C or COBOL.

The Web Transaction RAD facility is an innovative and simple methodology that allows traditionally skilled programmers to develop, test and deploy multi-tiered Web systems without having to deal with the complexities of tier-2, such as designing and developing servlets and JSP that manage and contain data provided by tier-3 servers (mapping data between Java objects and flat data structures, understanding API for host connectivity, and so forth), and manage sessions and state.

Using this approach, developers can specify their system at a logical level, and let VisualAge generate both tier-2 (Java and JSP component) and tier-3 (C++ or COBOL) code that implements those specifications.

VisualAge Generator programs generated for CICS and IMS are COBOL programs and have scalability characteristics comparable to programs developed directly in COBOL. Server programs, including Web Transactions, that are created using VisualAge Generator Templates have the scalability and data integrity characteristics of well designed pseudo-conversational CICS programs. They do not hold database or file locks during user think time. They automatically reestablish database position following a user request, and use time stamps to check that updates are not done against records that have been changed by another transaction while the user was thinking.

Server Scaling: WLM, Parallel Sysplex

The existing transaction systems (such as CICS and IMS) have consistently scaled with the largest processors built (full SMP) and more recently, with data sharing, across the Parallel Sysplex environment with a predictable increase in throughput proportional with the increase in CPU power. The newer application environments are experiencing limitations on all platforms based on technologies that are not yet mature enough to reach the same levels of scaling as the systems they are connecting to. TCP/IP stacks, including OS/390 as of V2R5, typically scale well on most platforms up to at least 10 processors (four processors on NT). Web servers tend to scale well for static pages (although the file systems may experience some constraints when update ability is required within a cluster (where replication may be required). However, when they are accessing relational databases (e.g. JDBC) or they are running other programs (CGIs, servlets/JSPs), there are some overheads (process create/destroy, affinities, threading, context switching, etc.) that may limit scaling.

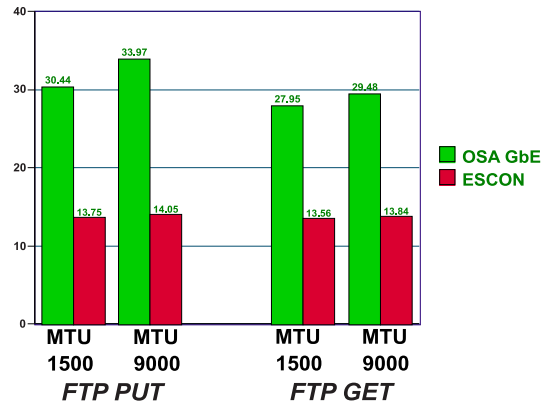
S/390 has some technologies that can assist these newer types of applications to scale to higher rates without requiring major changes to the applications them-

selves. The first is the Workload Manager (WLM), which can replicate application environments to provide expansion of the number of threads/processes without extra definitions or application awareness. It (WLM) can be used to manage address spaces and provide increased scalability when there are limits on the number of threads or connections to backend systems. CICS has an implementation defined limit of 100 concurrent EXCI connections per address space. This is intended to limit opportunities for “denial of service” attacks.

Another technology in OS/390 that facilitates server scaling is our Parallel Sysplex and data sharing support, such that applications can run on any processor without concern for data locality. Replicating Web server processes across a system or across the sysplex enables more transactions across a single logical application image. The shared HFS in OS/390 R9 enables access to content across any processor in the sysplex to reduce down time for publishing new content. Use of session state in servlets creates process and/or processor affinities which may affect scalability. There are several schemes to help address this (shared memory, use of shared data within a cluster, etc.) but they add additional overhead. WAS 3.0.2 on OS/390 supports storing session state in DB2, which is sharable across processes and processors. Network load distribution technologies (like the IBM SecureWay® Network Dispatcher or the Cisco® Local Director) enable this horizontal scaling with a single network definition (IP address) to all clients.

Another technology that OS/390 deploys in this space (which exists on multiple UNIX platforms) is “async I/O.” As opposed to Select, which incurs more overhead as the number of sockets increases, async I/O does not experience large system effects and requires less overhead on a per connection basis. The OS/390 Web server (as well as other application environments like Domino™, ADSM, EJBs,) utilizes this support. This is of particular value for functions like SSL and persistent connections, where the life of the transaction is longer.

FTP Performance with Gbit Ethernet Adapter versus ESCON

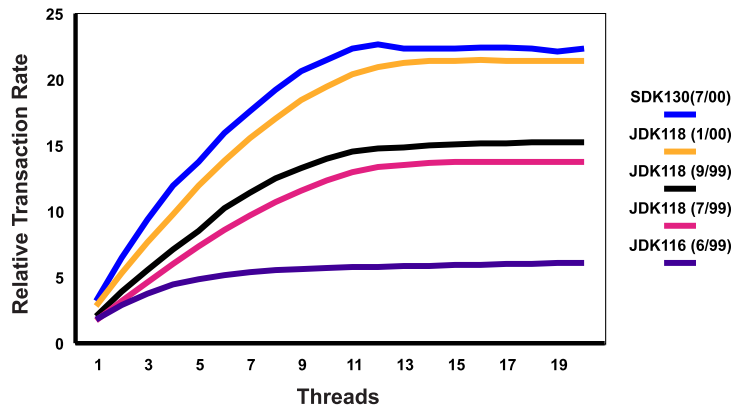


- 2.1-2.4X Throughput increase
- 11-25% CPU reduction

OS/390 Release Considerations for Performance

TCP/IP is obviously a major factor in the scalability and CPU efficiency of Web-based applications. Gigabit Ethernet with QDIO in OS/390 V2R7 greatly improves throughput, reduces latency and pathlength and increases the number of

Java for OS/390 Application Performance Comparisons

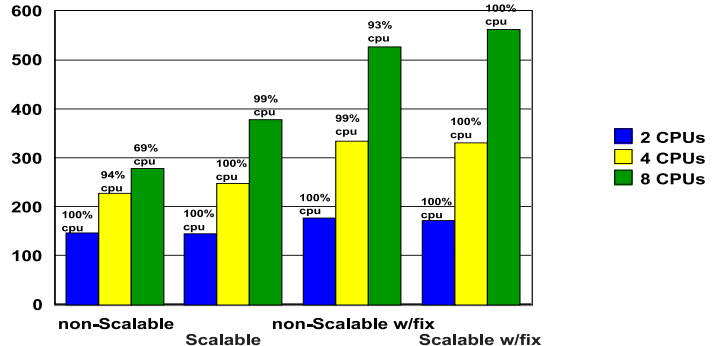


The tests shown here were conducted using an S/390 G6 system with **12 active processors**. While results will vary for specific applications, this graph illustrates results from IBM internal tests for a representative pure Java server application using varying numbers of threads. Apart from the level of Java for OS/390, all hardware and software was held constant.

connections through a single connection by multiple factors (2-5) over other existing options. Given the unpredictable nature of e-business and increasing need for bandwidth, we recommend OSA-Express for high volume Web solutions. The next chart illustrates the value of OSA-Express Gbit Ethernet when compared to ESCON communication channels.

The Java based connectors will see significant improvements in performance on the latest level of the JDK (1.1.8). These are available on OS/390 releases (V1R3 thru V2R8). The chart below shows that the JDK performance will continue to show significant performance improvements with JDK 1.3.0.

Customer Scalable Web Experiments w/IMS Connect for Java and WAS 3.0.2



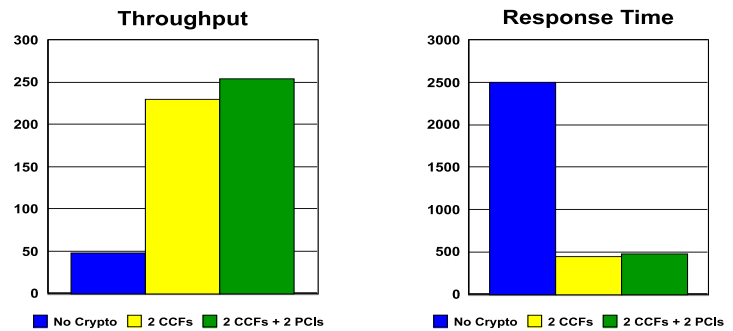
- Single copy of WAS scales very well up to at least 8 CPUs w/ PQ40030
 - Scalable mode transparently relieves constraints with minimal CPU cost per trx
- Note: Each servlet drives 2 IMS transactions on same system**

The Web server provides significant performance improvements for static Web serving in V2R7 through use of the Fast Response Cache Accelerator (FRCA) cache. The Web Application Server 1.2 has addressed some issues working with the WebSphere Application Server in scalable mode. The following chart shows data from a customer test where scaling was tested going from 2,4, to 8 CPUs with and without scalable mode. Without scalable mode, resource constraints limit CPU utilization and throughput at the 4 and 8 CPU points. These constraints are relieved and the full power of the processors utilized when scalable Web is used showing excellent scaling without any degradation at the two CPU measurement point.

The V2R8 Web Server has improved support for scalable mode, including persistent connections and SSL support, which in V2R10 has been enhanced to use system SSL (which enables a common key ring and improves the secu-

WebSphere SSL TDES Performance

Integrated Cryptographic Service Facility
Cryptographic Co-processor with PCI Crypto Cards



Full handshake w/ TDES approximately 50% of transactions

Measurements on 9672-R46 - 4 CPUs - 416 Mhz

Conceptual Chart - Data incomplete

urity/integrity of the key). For those applications that use SSL, the Lotus® Domino Go Web server (renamed to the IBM HTTP Server for OS/390 in OS/390 V2R6) has been enhanced to work with hardware crypto, which offers a dramatic CPU reduction, in OS/390 V2R6 release. This reduction is achieved by off loading the software encryption instructions from the general purpose processors to a separate crypto coprocessor. This can provide up to a ten-fold improvement in CPU utilization for SSL handshakes (public key operations). In addition, with OS/390 V2R9, we've added support for the new PCI crypto adapter, which increases the scalability of SSL to over 1000 SSL transactions/second.

The above charts show the throughput and response time benefits when two hardware crypto co-processors (CCF) are compared to software-only encryption and when two hardware crypto PCI cards are used along with the two CCFs. Each PCI CC card can perform approximately 100 full handshakes per second and up to eight PCI CC cards can be installed per CEC.

Recent tests with WebSphere 3.0.2 using System SSL on OS/390 V2R10 with four PCI cards and one CCF have shown that the application scales linearly and is able to fully utilize the hardware. We expect to verify the application scaling to the full eight PCI cards and two CCFs within the next month or two. We are also working on enhancements that will improve the throughput on the PCI cards by another 30%.

Customer Scenarios

This section describes the customer configurations where they are achieving satisfactory performance with one of the e-business connector options. We hope this information will be helpful for you in selecting an alternative.

Whenever possible, we have attempted to include OS/390, subsystem and other relevant product levels as well as rough performance/capacity information.

CICS

CTG (CICS Transaction Gateway) Customer reference example: A customer in Europe is using CTG with the servlet model to drive CICS TS 1.2 with CTG 3.0.2 on OS/390 V2R6.

Rabobank in Europe is using CTG with the applet model to drive CICS TS 1.2. A Java applet, containing the application and infrastructural components (e.g. IBM CICS JavaBeans) is served (pushed) from the OS/390 Domino Go Web server to the local proxies. Local clients start the client applet within the browser and JVM. A direct connection is then started between the Java applet and the CTG (CICS Transaction Gateway) on the OS/390 (no middle tier!). The gateway then transfers the Java request as an external call interface (EXCI) request to CICS TS1.2, and the appropriate COBOL transaction is started.

The applet was built using VisualAge for Java 2.1 enterprise edition. The entire applet operates stateless. The back-end application was built using IBM VisualAge COBOL 2.2 enterprise edition. The back end transactions were developed using a Component Based Development approach where the application

layer was separated from the business layer, which was separated from the data layer. These last two are stateless as well. This makes it possible to reuse the built business layer transactions, as well as address the business transactions from any platform. Initial production tests show that the gateway is stable and performs well. One entire transaction from front-to-back-to-front takes about two seconds, with just a fraction of that spent on the S/390.

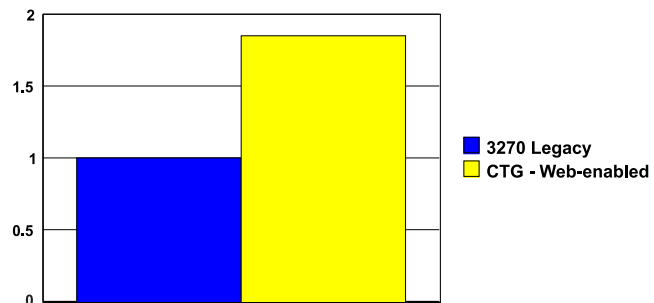
To reduce data flow over the lines, they implemented the standard Java zip functionality in the minimal compression configuration. The performance of the JVM on OS/390 is one of the limiting factors. We expect (and have seen) that the current setup should be able to handle around 10 transactions per second without the CTG taking up an excessive amount of mainframe resources and stifling other critical processes. In general it can be said that the performance of the JVM on S/390 is the critical factor. This will get better in future releases.

Please see the following Web sites for more CICS customer examples including a sysplex example. [.software.ibm.com/casestudies/swcs.nsf/customername/E3B6A022F22F9CE20025678B000659B9](http://software.ibm.com/casestudies/swcs.nsf/customername/E3B6A022F22F9CE20025678B000659B9) (CWS - Health Ins Corp)

Cost of WebSphere CCF Enablement

CICS TS1.3, CTG 3.1.1, WAS 1.2, JDK 1.1.8 PTF6,VAJ3.0.2

CICS/DB2 - 10 Transaction Order Inventory Workload



- Cost of CTG w/ WAS amortized across a mix of transactions is less than 2X
 - More enhancements in the pipeline
- Servlets generated with VAJava - no JSPs used

.software.ibm.com/solutions/internet/g325-1336-00.pdf (CWS - Den Danske Bank)

IMS

IMS Connector for Java

Another domain of investigation today, is the use of servlets provided by VisualAge for Java and IMS. The new IMS Connector for Java provides a way to create Java applications or servlets that can access IMS transactions through use of WebSphere and CCF with VisualAge for Java. The Java application or servlet acts as a TCP/IP client to IMS TOC. Significant performance and scale improvements are being seen with the latest WebSphere 3.0.2 and VA-Java servlets with the IMS Connector for Java.

Second Generation (Java) Connectors

•Connector: CICS Transaction Gateway (CTG)

This connector works with WebSphere and supports Java servlets and applications. It provides CCF classes for interfacing to CICS applications. Creation of connector applications is supported by VisualAge for Java (VA-Java) 2.0 and above. The key performance consideration for CTG is Java/Servlet performance. Up to 100 concurrent requests to CICS may be made from a single WebSphere address space, enabling many hundreds of browser connections to be supported. In order to scale beyond this, the WebSphere Application Server in scalable mode is recommended. CTG requires CICS Transaction Server (CTS) V1R3 and OS/390 V2R5 or greater. OS/390 V2R7 and JDK 1.1.8 are recommended for their performance improvements. It supports SSL by virtue of the underlying IBM HTTP Server. CTG is the logical follow-on for CICS Java Gateway.

Ordering information: *CTG is available from the following Web site:*

.software.ibm.com/ts/cics/downloads/

•Connector: IMS Connector for Java

This connector works with WebSphere and supports Java servlets and applications. It provides CCF classes for interfacing to IMS applications. Creation of connector applications is supported by VisualAge for Java 2.0 and above. The key performance consideration for the IMS Connector for Java is Java/Servlet performance. It requires IMS V5 and IMS TOC. It supports SSL by virtue of the underlying IBM HTTP Server. The WebSphere Application Server in scalable mode is recommended for scalability.

Ordering information: *The new IMS Connector for Java is currently available with VisualAge for Java, Enterprise Edition, Version 3.0 for Windows NT. It is available for download from the VisualAge Developer Domain to registered owners of VisualAge for Java, Enterprise Edition, Version 3.0.*

•Connector: SQLJ

SQLJ offers an easy to use static SQL syntax (stored procedures) that requires the application developer to know the table and column names when the application is written. The primary performance consideration is Java/Servlet performance. SQLJ can provide better performance and security than JDBC, based on the program author's SQL Privileges. SQLJ is pre-processed by the DB2 SQLJ Translator into JDBC. Coding SQLJ is much simpler, more efficient and performs better at run-time. This means more programmer productivity and better system utilization. SQLJ is an open standard created by IBM, Oracle and others. It is supported by VA-Java based tools. It requires DB2 V5 and OS/390 V2 R5. SSL is supported by virtue of the underlying IBM HTTP Server.

Ordering information: *SQLJ is available as a download for DB2V5 and V6 from: .software.ibm.com/data/db2/os390/sqlj.html/*

•Connector: JDBC

JDBC uses dynamic SQL so it can deal with situations where you don't know the table and column names at the time the application is written. The primary performance consideration is Java/Servlet performance. It is supported by VA-Java based tools. It requires DB2V5 and OS/390 V2R5. SSL is supported by virtue of the underlying IBM HTTP Server.

Ordering information: *JDBC is shipped with DB2 V5 and V6.*

This has the advantage that applications built on top of the Connector for SAP R/3 can be used on any server infrastructure which provides an implementation of the CCF infrastructure. Implementing the CCF connector interfaces makes the Connector for SAP R/3 consistent with other e-business connectors. Therefore writing applications using the Connector for SAP R/3 follows the same programming paradigm as programming the other connectors. The Access Builder for SAP R/3 is the tool you use at development time to generate the SAP specific classes. You can then use the Enterprise Access Builder (EAB) tool together with these generated classes to build EAB specific Commands and Navigators. This gives you the full flexibility of combining functions from different SAP objects and Remote Function Calls (RFCs) in one EAB Command or Navigator.

LINUX/390

IBM has announced its intent to provide WebSphere Advanced Edition 3.5 on Linux/390 by YE00. Included on top of this support will be connector support to access CICS, IMS, DB2 and MQ. The CICS solution will consist of a CICS Java client on Linux/390 connecting via TCP/IP to the CICS Transaction Gateway on OS/390. The IMS solution will consist of the IMS Connector for Java running on Linux/390 connecting via TCP/IP to the IMS Connect feature on OS/390. The DB2 solution will consist of the DB2 Connect feature (along with UDB) running on Linux/390 connecting via TCP/IP to DB2 on OS/390. The MQ solution will consist of the MQ Java Bindings running on Linux/390 connecting via TCP/IP to MQ/ESA on OS/390.

Future Directions WAS EE/390 will not incorporate an HTTP server. Instead, WAS EE/390 will be able to be configured to work in conjunction with HTTP servers that execute either on or off the OS/390 platform. For customers looking to develop two-tier solutions by hosting the HTTP server on OS/390, the IBM HTTP Server for OS/390 (once known as Domino Go WebServer) is already a part of OS/390 and will remain available for e-business applications.

The IBM HTTP server for OS/390 (DGW) will perform and scale better initially as well as have more OS/390 qualities of service (like WLM scalable mode) for running high volume transactions. Over time, we expect the Apache version (with enhancements like threading) will improve and lessen the gap. More and more, new functions will be made available on the Apache base (although both servers will support the application interfaces required to run servlets, JSPs and EJBs), consistent with the evolution in the industry.

Additional Information on Connectors

Links to more information on e-business connectors for OS/390 is available on our connectors Web site: s390.ibm.com/ebusiness/connectors/

Redbooks

- *Revealed! CICS Transaction Gateway and more CICS Clients Unmasked, SG24-5277*
- *Revealed! Architecting Web Access to CICS, SG24-5466*
- *Web-to-Host Integration Solutions, SG24-5237-00 1998-10-01 (an update is in progress)*
- *IMS e-business Connect Using the IMS Connectors, SG24-5427-00 1999-07-02*
- *The Trader Story, A Performance Study of Web Access to CICS, SG24-5748-00 (coming soon)*

Other sources for help:

- *The OS/390 e-business Integration Test Report:*
- *.s390.ibm.com/ebusiness/techinfo/ebitintr.html*
- *Performance and tuning guidance: .s390.ibm.com/ebusiness/*
Select the performance category on the navigation bar on the left. This category will be available at the end of December, 1999.
- *CICS Web Selection Guide:*
.software.ibm.com/ts/cics/library/white_papers/cicsweb/
- *The JavaSoft Web Page for the latest specification on the Common Connector Framework: .javasoft.com/*
- *The OS/390 Planning Assistant for e-business: .s390.ibm.com/os390/wizards/ebiz*
- *The OS/390 Web-Enablement Overview, GF22-5138*



© Copyright IBM Corporation 2000

IBM Corporation
Integrated Marketing Communications,
Server Group
Route 100
Somers, NY 10589

Produced in the United States of America
10-00
All Rights Reserved

References in this publication to IBM products or services do not imply that IBM intends to make them available in every country in which IBM operates. Consult your local IBM business contact for information on the products, features, and services available in your area.

AIX, CICS, DB2, DB2Connect, ESCON, IBM, IBM logo, IMS, @server logo, Language Environment, MQSeries, Net-Data, OS/390, Parallel Sysplex, SecureWay, S/390, VisualAge and WebSphere are trademarks or registered trademarks of IBM Corporation in the United States, other countries or both.

Java and all Java-based trademarks are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

LINUX is a registered trademark of Linus Torvalds.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through The Open Group.

Windows NT is a registered trademark of the Microsoft Corporation.

Other trademarks and registered trademarks are the properties of their respective companies.

IBM hardware products are manufactured from new parts, or new and used parts. Regardless, our warranty terms apply.

Photographs shown are of engineering prototypes. Changes may be incorporated in production models.

This equipment is subject to all applicable FCC rules and will comply with them upon delivery.

Information concerning non-IBM products was obtained from the suppliers of those products. Questions concerning those products should be directed to those suppliers.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.