Managing IT growth
September 2008

Why to choose CICS Transaction Server for new IT projects

Author
Andrew Bates, IBM Software Group

Contributor
Timothy Sipples, IBM Enterprise Software Architect

Contents

Introduction

In the past, decisions about which deployment platform to use for new IT projects were often mandated, based on an organization's technology preferences at a given time. Now that the IT industry has established a set of common standards that enables applications to be delivered as services, this is usually no longer the case. Interoperable IT services can now be delivered regardless of the underlying technology implementation. Business drivers—such as the need to manage business growth without exponentially adding to IT cost and complexity—are becoming much more important decision factors. Frequently and increasingly these considerations favor highly virtualized platforms such as the IBM System z™ mainframe, running very efficient, modern applications in IBM CICS® Transaction Server for z/OS®.

This paper is aimed at IT architects, project mangers and other decision makers who are considering their options for new IT projects. It will be useful to those who need to choose a deployment platform for new application-development projects, as well as for anybody who is considering whether to consolidate distributed applications into a centralized environment. The paper does not assume that an organization already has a mainframe (although it certainly might), nor does it discuss the modernization or reuse of existing CICS applications (although many of the technologies covered certainly do support this). Rather, it focuses on the merits of deploying new applications to the System z platform, and in particular, to an environment managed by CICS Transaction Server.

From centralized to distributed computing—and back again

Although the environment in which businesses operate changes constantly, the fundamentals required for business growth do not. Businesses must innovate quickly to generate new revenue opportunities. They must expand into growing, profitable markets to ensure long-term success. And they must integrate their operations to control costs and maintain high levels of service to their customers.

Over many years, enterprise computing has evolved to address these challenges—initially by delivering a centralized mainframe platform that was the single, shared computing environment for an entire organization. This first computing wave established the foundations for delivering unprecedented levels of availability and scalability. Some time later, distributed (or *client/server*) computing gained widespread popularity, primarily due to perceived lower costs and flexibility. However, distributed computing also brought with it other, unexpected challenges.

In many distributed computing deployments, each new IT project required the procurement of many new distributed servers. Each of these servers tended to support discrete application functions (such as firewalls, Web servers, transaction processing servers or database servers), and each server had to provide enough capacity to handle its particular peak. Often servers were purchased many times over—for development, testing and production as well as for backup and failover. Then for each new IT project, new groups of servers were added, with separate hardware, software and networking. This deployment pattern has resulted in many organizations running hundreds, and sometimes even thousands, of distributed servers. This phenomenon has even been given its own name—*server sprawl*; the large, expensive and environmentally damaging data centers that are needed to house these servers are known as *server farms*.

Running such server farms presents a number of key challenges. Firstly, large numbers of distributed servers can prove to be surprisingly expensive to install, operate, maintain, power, cool and periodically replace. They also require additional IT staff to manage them. The total cost of owning these server farms can often be many times more than the initial cost of the technology acquisition. And in addition to the cost challenges, numerous technological problems can arise. Because distributed computing resources are not easy to share across networks, projects tend to be developed and designed separately from each other. This in turn encourages “siloed” information and low overall server utilization rates. One server might be busy (potentially affecting application response times), but another idle server cannot easily pick up that workload. Growth can be difficult to manage in these environments, and growing organizations must often partition, replicate and otherwise scatter their core operational databases and applications. This scattering makes privacy protection and delivery of consistent customer service more difficult. And the complexity associated with integrating such disparate data and applications can harm customer service and impede business flexibility.

Highlights

IBM System z software products such as CICS Transaction Server for z/OS and IBM DB2® for z/OS have enabled even the most ambitious of companies to easily cope with astonishing growth in transactions and information—without the need to keep adding more servers.

The end result is that many businesses with large and growing numbers of distributed servers are now reevaluating whether this server-farm approach is sustainable. In many cases, implementing highly virtualized computing environments, both for new application deployments and to consolidate their existing server farms, is a much more attractive alternative.

Get your IT environment under control

System z represents the most technically sophisticated enterprise-computing platform in the world. In particular, these servers are famously more reliable, more secure, and more space and energy efficient than other business computing platforms. Although many companies have struggled to contain distributed server farms, IBM System z software products such as CICS Transaction Server for z/OS and IBM DB2 for z/OS have enabled even the most ambitious of companies to easily cope with astonishing growth in transactions and information—without the need to keep adding more servers. In fact, a combination of the massive computing power and unparalleled reliability delivered by today's System z servers makes it common for even larger customers to run just three machines: two in a main-site, all-purpose cluster and one located at a remote site for near instant disaster recovery. Many System z customers run huge numbers of core business applications on just one or two machines. This remarkable system efficiency is enormously valuable to these organizations.

Additionally, IBM has continually reinvented the IBM mainframe platform by increasing its investments, promising and delivering continued improvements in value and by dramatically reducing the cost of mainframe computing. IBM and its partners have focused on modernizing and opening the platform with standards-based technologies. Today's System z servers are equally as happy running Java™ and Linux® workloads as they are running core banking applications—concurrently on the same machine. The IBM mainframe is no longer the costly, inflexible platform it was once perceived to be. In fact, increasingly, it is being seen as the solution to what has turned out to be an ever-more costly, inflexible and out-of-control distributed-server-sprawl environment.

System z is the world's most sophisticated virtualization platform, and as a result, many companies that had primarily deployed new applications on distributed platforms are now turning to a centralized System z server as the platform of choice to support their business growth. CICS Transaction Server for z/OS represents one important element that can help organizations move their distributed applications to the System z platform.

What makes System z different?

There is no single reason why System z servers are so much more sophisticated than their distributed brethren. Instead, the value of the platform is architected into every part of the hardware, operating system and software that it runs. This section covers just some of the combined capabilities that make IBM System z servers different from their distributed counterparts.

Unique IBM System z architecture

System z hardware design is fundamentally different. Distributed computers are usually designed for low cost of initial acquisition or raw, single-task computational speed. But a low purchase cost does not equate to low running costs; and performance for business applications depends on much more than just the straight-through mathematical speed of the processor. System z architecture is different. It embodies a balanced design that delivers maximum overall throughput, while embedding reliability, availability and serviceability (RAS) capabilities right into the processor chips themselves.

First, let's take overall throughput. Three factors affect actual performance of business applications:

- *The processor speed (usually measured in gigahertz)*
- *The speed of the memory hierarchy (processor cache and other memory)*
- *The speed at which the data can be moved to and from the processor (I/O for disk access, networking, printing and similar work)*

Although the latest System z servers have one of the fastest commercially available processors on the market, they also dedicate a significant amount of silicon chip space to processor memory (the L1 and L2 cache), and they devote additional processor resources entirely to I/O bandwidth. This

Highlights

The throughput of real business applications that submit high volumes of create, read, update and delete requests against very large databases (as is usual in enterprise computing) is unparalleled on the System z platform.

architecture dramatically reduces the time the system takes to move data in and out of memory, and the time that it takes for that data to travel to and from peripheral devices such as disk. As a result, the throughput of real business applications that submit high volumes of create, read, update and delete requests against very large databases (as is usual in enterprise computing) is unparalleled on the System z platform. This balanced performance makes it much easier to run multiple business applications and databases concurrently on the same system. System z customers routinely run their machines at over 80 or 90 percent utilization all year round. The System z platform gracefully copes with sustained periods of 100 percent utilization if the workload dramatically increases.

Secondly, let's take the processor quality. The fundamental architecture of the microcode on the chip itself is different. IBM System z mainframes use a complex instruction set computer (CISC) chip architecture rather than, for example, a reduced instruction set computer (RISC) architecture, which is common on distributed UNIX® platforms. Although the debate between CISC and RISC is largely academic now (because both philosophies borrow from each other), the CISC architecture does help IBM System z chip designers to add more RAS directly onto the processor microchip itself. These CISC RAS attributes make the processor cores less likely to fail. For example, System z chips have a built-in capability called Dual Instruction and Execution Unit, which means that every processor instruction is effectively executed twice and the results compared. If the results are not identical, the System z server dynamically resolves the problem by retrying and switching the instruction set to a spare processor if necessary, without any impact to the application. Compare this with a distributed environment, where failures are likely to suddenly stop the entire server—along with all the work that is being processed on it. On a distributed server, much more of the RAS burden is placed on IT operational staff and application developers. Abnormal system terminations are generally handled through clustering, adding more machines and trying to architect a "fast failover" solution (with as little data loss and downtime as possible). This approach increases the overall cost and complexity of the solution and further contributes to the distributed-server-sprawl phenomenon.

Unique IBM System z virtualization and scalability

System z servers are inherently superior for highly virtualized environments, precisely because of their ability to reliably run mixed workloads at near 100 percent utilization on a sustained basis. However, it is the industry-leading System z partitioning and hypervisor technologies that make this virtualization a reality. The key technology that enables this is called a logical partition (LPAR). This approach allows a subset of the computer's hardware resources on a physical machine (processor resources, memory and I/O bandwidth) to be partitioned across multiple LPARs, each housing a separate operating system. System resources can be dynamically balanced across LPARs, with the system automatically assigning more processor resources or I/O channel bandwidth to LPARs according to predefined service level agreements.

System z owners can define up to 60 LPARs per machine (depending on the model), and each LPAR is logically and certifiably equivalent to a separate physical System z server. Because LPAR boundaries are completely isolated from each other, using LPARs safely allows multiple production environments to be defined on the same physical hardware. A single machine can host multiple, separate businesses without compromising the privacy of any of them. LPARs are also enormously useful for providing development and testing environments on the same machine as the production environment. Using LPARs offers several advantages, including lower costs, less data center space, better use of resources, faster deployment and greater convenience. The ability to run multiple LPARs on a single system has been a core part of the mainframe value proposition for almost two decades (although other forms of virtualization on the mainframe date back four decades). In fact, all modern System z servers must run in LPAR mode—even if they run only a single LPAR. In recent years, some distributed platforms have begun to implement some partitioning techniques. However, the ability to create a truly shared computing-system complex is unique to the System z platform.

IBM Parallel Sysplex® technology enables software and applications running in separate LPARs to conduct parallel processing using the same cluster wide state data (for both read and write access) using a direct memory

Highlights

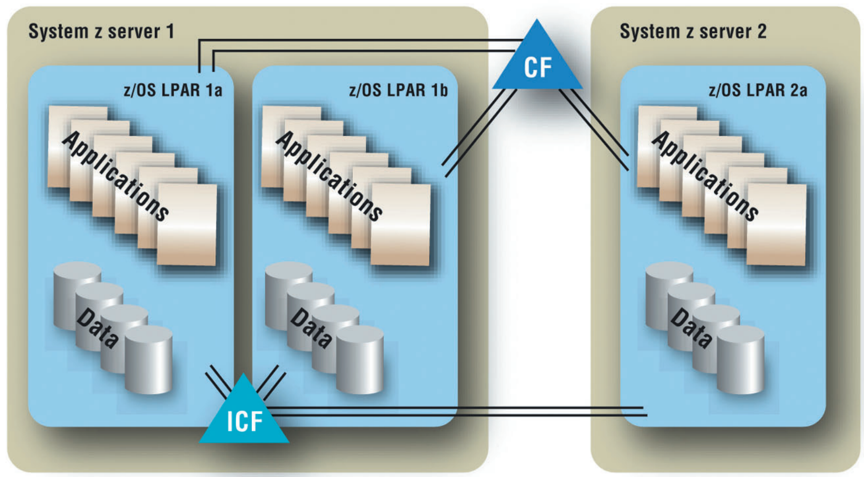
access-like mechanism, even across physically separate System z servers. This Parallel Sysplex technology is perceived by the operator, the user, the database administrator and anyone else who works with it as a single system image that can be managed from a single point of control.

The core technology that enables this unique “shared everything” environment—even across physically separate mainframes—is a called a coupling facility (CF). This refers to a specialized System z operating system called Coupling Facility Control Code (CFCC) that runs on a System z server. It can be an internal coupling facility (ICF) running inside a System z server that is doing other work, or an external coupling facility running on a dedicated System z server. In both options, the coupling facility has a large memory and extremely high-speed dedicated channels (CF links) that it uses to exchange data directly between its own memory and the memory of the attached systems. This enables all connected systems to share (read from and write to) a single source of memory, and pass information to each other at memory like speed.

Parallel Sysplex environments with multiple “shared everything” partitions solve many of the problems of distributed server sprawl.

Parallel Sysplex environments with multiple “shared everything” partitions solve many of the problems of distributed server sprawl. They can scale almost linearly, up to 32 physically separate System z servers (although in reality, no customer has ever needed anywhere near that much computing capacity). They enable parallel processing environments with no single points of failure. Workloads can be dynamically distributed across any node in a Parallel Sysplex cluster with spare capacity, thus minimizing response times and maximizing system utilization. The ability to have truly shared access to all computing resources avoids the need to partition data or applications, or to replicate databases across multiple servers in order to maintain performance. In fact, the underlying structure of the Parallel Sysplex environment remains virtually transparent to users, networks, applications and even operations.

Highlights



IBM Parallel Sysplex technology enables software and applications running in separate LPARs to conduct parallel processing using a mechanism similar to direct memory access, even across physically separate System z servers.

System z environments using Parallel Sysplex technologies are unique compared to every other system, solution or architecture available today.

System z environments using Parallel Sysplex technologies are unique compared to every other system, solution or architecture available today. IBM software products such as CICS Transaction Server for z/OS and DB2 for z/OS have been explicitly designed to exploit this Parallel Sysplex technology—delivering the potential for applications and data in these environments to scale (for all practical purposes) beyond any known requirement today. It is this capacity that makes the modern System z platform such an attractive proposition for companies who need to manage growth, or who need to centralize and consolidate multiple distributed servers.

Unique System z software

The virtualization capabilities of the System z platform and the z/OS operating system make it possible for software vendors to develop applications with functional and non-functional capabilities (such as availability, scalability or reliability) that far exceed what is possible in a distributed environment. Although there are literally thousands of System z software packages available from both IBM and its Business Partner community, most new IT projects involving CICS Transaction Server are likely to also include IBM WebSphere® Application Server and DB2 software. As we will see later, IBM Rational® application-development software can be used to model, assemble, deploy and manage composite modular applications with components from each of these key subsystems.

IBM CICS Transaction Server

IBM CICS Transaction Server for z/OS is a managed environment for hosting high-volume, highly efficient, transactional applications written in popular and productive business-oriented programming languages such as COBOL, PL/I, C/C++ and Java. It can process complex and demanding application workloads with optimal performance. CICS Transaction Server provides a rich set of functions for application developers and system administrators that enable them to build and manage highly efficient, high-volume applications that can use standards-based SOA interfaces such as Web services, Java connectivity and reliable WebSphere MQ messaging.

CICS Transaction Server has sophisticated virtualization capabilities that integrate with System z Parallel Sysplex technology to deliver the almost unlimited levels of scalability that are required to cost-effectively manage business growth. CICS Transaction Server—along with the comprehensive ecosystem of systems management, application development and SOA integration technologies that support it—is the primary focus of this paper.

IBM WebSphere Application Server

WebSphere Application Server is a family of application servers that delivers a consistent, standards-based Enterprise Java deployment environment across the broadest number of platforms, including z/OS and Linux on System z. It has comprehensive support for dynamic Web and Web services applications that rely on Java technologies such as JavaServer Pages (JSPs), Java Servlets, Enterprise JavaBeans (EJBs), Java Portlet applications, and Session Initiation Protocol (SIP) applications. Standards-based Java applications running in WebSphere Application Server can be ported across platforms unmodified, making it straightforward to move distributed applications to the System z platform.

WebSphere Application Server offers a family of proven, robust Java runtime environments that can meet the needs of customers of all sizes, from the entry-level Community Edition (which has no license charges) to the

Highlights

advanced editions and topologies that provide the workload management, scalability, high availability and central management of multiple application servers required for enterprise deployments. WebSphere Application Server provides the highest qualities of service when deployed to a z/OS environment, where it can integrate with Parallel Sysplex technologies to deliver the maximum levels of availability and scalability.

IBM DB2 for z/OS

IBM DB2 for z/OS is a high-performing and extremely scalable relational-database management system that excels in the management of vast quantities of information, with ultra-high-speed access to it, and the ability to perform complex query processing upon it. It manages access from many concurrent users to large volumes of data in a secure and "always available" manner, while ensuring that the integrity of data is maintained at all times. DB2 for z/OS is also supported by a comprehensive suite of data management tools covering all areas of database maintenance, performance, business intelligence, governance and much more.

DB2 on z/OS in a Parallel Sysplex configuration is unique in that it can provide a truly "shared everything" architecture by using the coupling facility technology to share database state information (locks and disk cache).

DB2 on z/OS in a Parallel Sysplex configuration is unique in that it can provide a truly "shared everything" architecture by using the coupling facility technology to share database state information (locks and disk cache). All systems and applications that have access to this coupling facility have read and write access to it at memory-like speed, regardless of their logical or physical location. To really appreciate the benefits of this unique capability, it must be contrasted with the distributed alternatives: "shared nothing" (which leads to inflexible data and application partitioning) or "shared disk" (in which synchronization overhead causes scalability and performance problems). Using "shared everything" technology, DB2 for z/OS can deliver unparalleled levels of near-linear scalability, performance and availability. That includes avoiding planned outages that other databases require to apply patches or to upgrade versions.

A word about network latency and network overhead

Before we turn our attention to focus on CICS Transaction Server and the ecosystem that supports it, it is important to say a final word on the value of deploying all application components to the System z platform. Although most new IT projects involving CICS Transaction Server will access data that is also hosted in the System z environment (in DB2 for z/OS), deciding where to locate the WebSphere Application Server environment is not always as

Highlights

straightforward. Although the qualities of service of WebSphere Application Server are certainly highest running under z/OS, it can also work very well running on Linux on System z or on entirely distributed environments. When considering which deployment platform to use, some important considerations are network latency and network overhead.

Network latency is the delay between the initiation of a network transmission by a sender and the receipt of that transmission by a receiver—basically, it is the duration it takes for a sent message to be received. Network overhead is the cost of processing a piece of work (for example, the processor usage and network bandwidth) that is entirely due to the network rather than the useful content of the message. In cases where the quantity of data being requested and returned over a network is small (such as an account balance enquiry), the vast majority of the information produced, transmitted and processed is simply that which is required to establish, manage and encrypt the network connection. Additionally, networks can fail, so each network hop is a potential additional point of failure. Moreover, network failure can affect how the application behaves—potentially requiring additional coding and thus imposing a costly burden on application developers.

One key advantage of deploying multiple products in a z/OS environment is that all subsystems and applications can communicate with each other across shared memory rather than a network.

One key advantage of deploying multiple products in a z/OS environment is that all subsystems and applications can communicate with each other across shared memory rather than a network. No network means no network latency, no network overhead and no network failures. Additionally, z/OS has a feature called Resource Recovery Services (RRS) that can manage the overall success or failure of a unit of work across multiple subsystems, including between WebSphere Application Server and CICS Transaction Server—so failure compensation logic need not be programmed into the applications. It is for these reasons, in addition to the higher qualities of service, that z/OS is such an attractive platform to host WebSphere Application Server deployments.

WebSphere Application Server for Linux on System z also has advantages over distributed deployments, because it has access to a unique System z technology called IBM HiperSockets™, which can enable IP networking to flow across memory between virtual servers on the same hardware, significantly reducing network latency and network overhead. Additionally, Linux on System z also exploits another unique System z virtualization technology called IBM z/VM®, which can be used to support large numbers (many hundreds) of Linux virtual machines running WebSphere Application Server and other workloads, all on a single physical IBM System z server.

These are all important deployment considerations that should not be overlooked when deciding which topologies to use in new IT projects that include WebSphere Application Server deployments both for current and new System z installations.

CICS Transaction Server and transaction processing fundamentals

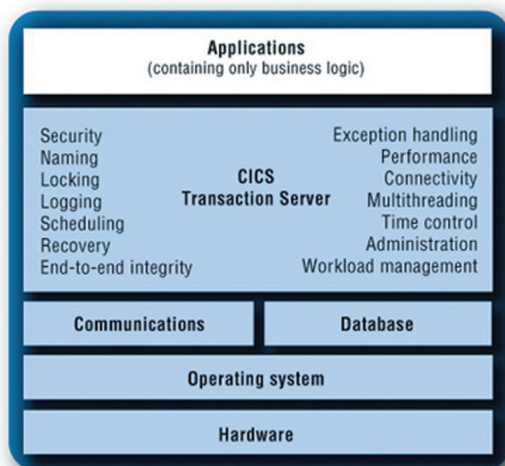
The global economy is made up of an immense number of business transactions, usually involving the exchange of goods or services for money. These business transactions tend to be short and repetitive. Common characteristics include concurrent access by many users, through multiple interfaces, to shared information—without compromising data integrity. Some examples of such transactions include a customer information inquiry, a financial transfer, an inventory update, a point-of-sale transaction, a travel reservation, an insurance claim, stock trading, order processing, account billing, a credit check or any number of similar operations. Because processing these transactions represents a direct cost to the business, the cost for each transaction must be minimized—without compromising future business growth.

It is possible to write standalone applications (in various programming languages) to perform these tasks in shared, multiple-user environments; however, it is complex to do so. Application developers would have to address such issues as the communications with the user interface, ensuring that only authorized users access the system, preventing concurrent changes, handling failures and backing out partial updates, managing the database connectivity, and numerous other complicated tasks—each of which would have to be rewritten for each new application. All of this is extra and unnecessary work in addition to writing the actual business logic required to process the business transaction.

Highlights

CICS Transaction Server for z/OS is a managed environment for hosting high-volume, highly efficient, transactional applications written in popular business-oriented programming languages.

CICS Transaction Server for z/OS is a managed environment for hosting high-volume, highly efficient, transactional applications written in popular business-oriented programming languages. Instead of requiring developers to write system functions into their applications, CICS Transaction Server performs these services either automatically, or upon request from the application. This allows developers to focus on solving business problems rather than implementing system functions. This approach offers several advantages, including shorter development cycles, easier maintenance and easier incorporation of new functions (such as adding Web services interfaces to applications). Applications do not have to be changed when new versions or releases of operating systems or databases become available—because CICS Transaction Server handles the interface with these systems.



CICS Transaction Server performs system functions either automatically or upon request from the application, allowing developers to focus on solving business problems.

Because CICS Transaction Server itself is developed to exploit the full potential of the System z platform, applications running in CICS Transaction Server inherit many of these qualities of service, with little or no additional programming required. This enables these applications to be massively scalable, available and high performing. Applications that run in CICS Transaction Server are referred to as *CICS applications* (regardless of the programming language they are implemented in).

Best practices for CICS application design

To users of business services accessing IT systems, perhaps through a Web browser, the systems they access can seem to be one service that has many capabilities. From an IT application perspective, the view is very different. For reasons of performance and extensibility, large business solution areas such as online banking that seem like a single service to the user are actually made up of hundreds (perhaps thousands) of linked applications that each perform discrete functions. For example, an online banking solution might have multiple applications to display accounts, check balances, transfer money or perform other functions. All of these applications are linked together so that when a user of online banking transfers money from one account to another, the new and correct updated balances are instantly reflected back to the user.

When designing applications for IT projects, it is important to ensure that the design does not inhibit scalability as the business grows or hamper flexibility as business requirements evolve and new technology offers new opportunities. CICS Transaction Server encourages a componentized approach to application design, development and deployment. Elements of programs (for example, the presentation logic, the business logic and the data access logic) are deployed separately. These elements can be further componentized into separate business-logic components for separate business rules. All program elements are linked together using CICS Transaction Server capabilities, either internally, between CICS applications within the CICS Transaction Server environment, or externally, to databases, or to other IT systems through standards-based interfaces such as Web services.

Designing CICS applications for performance and efficiency

Performance encompasses many areas, including the efficiency of the deployed code (which can be optimized by calling CICS Transaction Server functions) and also the efficiency of the way in which the system interacts with those applications. Because every computer system waits at the same speed (that is, stopped) it is critical to reduce the amount of time that the system spends waiting. CICS Transaction Server promotes two important design rules to reduce the time the system spends waiting: tasks should exist in the system for the minimum time possible, and resources should be locked (unavailable to others) for the shortest time possible.

CICS Transaction Server does this by enabling a *pseudo-conversational* design model, where separate application components run independently of each other. Each component performs a discrete business function and then ends, with CICS Transaction Server itself taking responsibility for linking these application components together. In the case of a failure at any point in the process, CICS Transaction Server ensures that all the effects of an application are completed in their entirety or returned to their original state (and the requester informed). Complete integrity of data is maintained at all times, even in the case of total system failures such as those due to loss of power to the system (in which case the transactions are processed correctly when power is returned to the system).

This pseudo-conversational model of programming also reduces the time during which an application requires exclusive access to a shared resource. Exclusive access requires that a lock be placed on a resource (such as database lock) to prevent other applications from concurrently updating it and corrupting the data. Reducing the time a resource is locked can reduce the time other applications have to wait for it and increase the overall throughput of the system. It also reduces the potential for a deadlock situation, where two or more applications are waiting for each other to release locks in order to finish (and thus in theory neither will ever be completed). In the relatively rare situation when a deadlock does occur, CICS Transaction Server dynamically takes action to resolve the deadlock.

Highlights

The componentization of CICS applications provides huge potential for reuse, because new technology can be implemented as it becomes available without the need to change the underlying applications themselves.

Designing CICS applications for reuse and flexibility

Although this paper is not focused on the reuse of existing CICS applications, the potential to reuse new CICS applications in the future is a large part of the value proposition of CICS Transaction Server. The componentization of CICS applications provides huge potential for reuse, because new technology can be implemented as it becomes available without the need to change the underlying applications themselves. The best way to illustrate this is with an example. A well-structured, terminal-based CICS application that might have been written over twenty years ago will have had the opportunity to be reused in numerous technology evolutions including client/sever, enterprise messaging, the Internet, Enterprise Java and Web services (to name just a few). Today, that same application could fully participate in a modern SOA by using capability introduced in recent versions of CICS Transaction Server. Compare this with the time, cost and risk of rewriting applications for each new technology evolution.

The potential to reuse highly componentized CICS business logic goes beyond evolutions in the external interface technology. As the requirements of the business grow, so will the need to add new functions that interoperate with existing functions. With highly componentized code, it is much easier to create applications that add new features, or to update programs to add new functions to them. New modules can be added to provide new features and functions with minimal disruption to existing CICS applications. CICS Transaction Server provides options to introduce new applications or to upgrade existing applications while the system is running. It is possible to make these changes without the system being unavailable to users at any time. Additionally, certain CICS application modules can be assigned a priority to ensure that critical applications continue to achieve predefined service levels, such as a certain response time, even when the system is running at 100 percent utilization.

Channels and containers

The pseudo-conversational design that maximizes performance and flexibility requires a method of passing information between application modules that is consistent across different programming languages. This information also needs to be exchanged under the control of CICS Transaction Server, so that

Highlights

any errors at any stage in the process can be handled correctly. Modern CICS applications can use a CICS Transaction Server technology called channels and containers to pass information between each other. This technology is structured and extensible, allowing data of any size, in any format, to be passed between CICS applications. For example, large quantities of structured parameter data in both XML and non-XML formats can be exchanged between applications. Channels and containers technology within CICS Transaction Server is ideally suited for structuring large quantities of data such as that which is often needed to provide meaningful Web services from a CICS Transaction Server environment.

The value of CICS Transaction Server and highly virtualized workloads

CICS Transaction Server provides the ability to reliably run many thousands of CICS applications in a single system, running continuously at near 100 percent utilization. This means dozens or even hundreds of physically distributed smaller servers can be consolidated into a single System z server.

CICS Transaction Server provides the ability to reliably run many thousands of CICS applications in a single system, running continuously at near 100 percent utilization. This means dozens or even hundreds of physically distributed smaller servers can be consolidated into a single System z server. Customers can run CICS Transaction Server on multiple LPARs, for multiple production environments (for example, banking corporations might run sets of LPARs for core banking and sets of LPARs for credit card processing), as well as separate LPARs for development, test and quality assurance. All applications running under CICS Transaction Server can be integrated into a single disaster recovery environment that covers the entire System z environment, eliminating the need to purchase dozens of additional distributed servers for backup and disaster recovery purposes.

CICS Transaction Server delivers an environment that is designed for virtualization by providing the ability to run many virtualized CICS application runtime environments known as CICS regions within a single CICS Transaction Server installation.

CICS Transaction Server delivers an environment that is designed for virtualization by providing the ability to run many virtualized CICS application runtime environments known as CICS regions within a single CICS Transaction Server installation. Running CICS applications in separate CICS regions provides a separation of workloads that isolates applications from each other. Any application-based problems that occur in one CICS region do not affect the performance or availability of applications running in other CICS regions. This design based on CICS regions also allows for the cloning of entire CICS regions so that workloads can be evenly balanced across the overall environment, delivering a highly available and highly scalable infrastructure that can respond appropriately to spikes in workload. It

Highlights

is not uncommon for large CICS Transaction Server customers to run many hundreds of CICS regions hosting many thousands of CICS applications, all in a single CICS Transaction Server system image on a single machine or in a Parallel Sysplex environment.

Managing a virtualized CICS environment

Running multiple CICS regions in a single system image is often referred to as running a *CICSplex*. This can significantly simplify the task of systems management compared to running dozens of distributed servers, but good management tools and operations procedures are still essential to achieve operational efficiencies and reduce costs.

IBM CICSPlex® System Manager is an integrated component of CICS Transaction Server that enables highly virtualized CICS environments with many CICS regions to be logically grouped into environments that work together to achieve predefined processing goals.

IBM CICSPlex System Manager is an integrated component of CICS Transaction Server that enables highly virtualized CICS environments with many CICS regions to be logically grouped into environments that work together to achieve predefined processing goals. It provides a single system image and a single point of control for the selected groups, offering exceptional levels of control over a company's application infrastructure. CICSPlex System Manager also provides sophisticated workload management capability that can integrate with the workload management capabilities of the z/OS operating environment to route workload to the CICS region that is most able to complete it, according to predefined business priorities and goals. This helps ensure the best response time for the user, and the most efficient utilization of system resources. Additionally, if certain "under stress" conditions occur, CICSPlex System Manager can trigger notifications to provide information to operators and even recover automatically from certain error conditions, such as a lost database connection.

CICSPlex System Manager is managed through a highly secure, simple-to-use, and easily customizable modern user interface. It is this level of management and control over highly virtualized environments that brings significant advantages over running and managing hundreds of separate distributed environments with no single view and no single point of control over the distributed, highly siloed infrastructure.

Highlights

Batch processing

A common issue than many businesses face today, especially those with highly partitioned distributed environments, is the ability to complete their offline processing requirements without affecting their online business. Examples of such offline, or *batch*, processing include nightly inventory updating, monthly customer billing, or quarterly creation of customer statements. In highly siloed distributed environments, batch processing often uses dedicated servers and data partitioning. However, the law of diminishing returns means that in some cases, to cope with just a small increase in the volumes of data being processed, significantly more computing power is required. For some businesses, adding more processing capability is not enough because the platform or the middleware simply does not scale well. To work around this growth inhibitor, regular repartitioning of data resulting in expensive and disruptive application changes is becoming commonplace.

CICS Transaction Server for z/OS and DB2 for z/OS together help ensure faster and more reliable on-time completion of batch cycles.

CICS Transaction Server for z/OS and DB2 for z/OS together help ensure faster and more reliable on-time completion of batch cycles, by providing the ability to concurrently and efficiently process complex transaction requests in both batch and online mode. This eliminates the need to procure and maintain numerous distributed servers with highly partitioned data just for batch processing. By assigning priority to online work, and having batch work run when the system is less busy (for example, at night or on weekends), large volumes of batch processing can be done very efficiently, without affecting the higher-priority online-transaction processing. Online transactions can be completed without affecting response times, while also allowing batch processing jobs to run within an efficient timeframe.

Additionally, z/OS has its own sophisticated, file-based data storage environment, called VSAM, which is designed for extremely high-performing access to data, for both online and batch applications. Before relational databases became common, VSAM was the most popular method of storing data in mainframe environments. Although it can be more difficult to provide true concurrent and fully shared access to data stored using VSAM (especially for data updates), it often retains an advantage in terms of raw performance and can be appropriate for certain applications, usually as an adjunct to DB2. VSAM and other file types can also be useful to support migrations from other

Highlights

environments to CICS Transaction Server. VSAM record-level sharing and Transactional VSAM provide advanced VSAM sharing and concurrent access capabilities for Parallel Sysplex environments, and can greatly enhance the availability of VSAM data.

CICS Transaction Server also provides first-class support for data that is stored in VSAM environments, and it continues to support new developments in VSAM technology, especially with regard to the ever-increasing size of overall data and the performance of access to that data. Additionally, a number of CICS tools are available to enhance the capability and efficiency of both online and batch CICS and VSAM applications. CICS VSAM Recovery and CICS Batch Application Control can reduce planned and unplanned downtime, reduce human error and make recovery faster, easier and more reliable. For customers who want the option to move to DB2 for z/OS in the future, a tool called CICS VSAM Transparency provides utilities to migrate data from VSAM to DB2 for z/OS, without the need to change the CICS applications themselves.

Integrating CICS applications into an enterprise-class SOA

CICS Transaction Server is an application hosting environment that has been designed to efficiently process heavy and complex mixed workloads with ease. It is the transaction processing engine of choice for the core business applications of thousands of the world's largest companies. However, typically no single IT system can provide everything that every customer needs. As a result, integration between heterogeneous systems will always be a requirement, and the IT industry has converged on a set of common standards and principles that has led to the emergence and maturity of the service oriented architecture (SOA) model of IT integration.

SOA is an IT architectural style that enables integration of business by linking services, and CICS Transaction Server is a high-quality hosting environment for SOA-compliant business services.

SOA is an IT architectural style that enables integration of business by linking services, and CICS Transaction Server is a high-quality hosting environment for SOA-compliant business services. CICS Transaction Server extensively supports standards-based interfaces for complete SOA integration, both as a service provider and also a service requester. When CICS Transaction Server acts as a service provider, CICS applications are called by some other IT system, and the service within CICS Transaction

Highlights

To enable all-inclusive integration of CICS applications within an SOA, CICS Transaction Server supports standards-based SOA integration options, including comprehensive native Web services capability, connectivity based on Java Enterprise Edition (EE) standards, and WebSphere MQ messaging.

Server runs some business logic and provides a response back to that system. When acting as a service requester, CICS Transaction Server can request information from some other service on some other IT system, receive the response and then do something useful with it.

To enable all-inclusive integration of CICS applications within an SOA, CICS Transaction Server supports standards-based SOA integration options, including comprehensive native Web services capability, connectivity based on Java Enterprise Edition (EE) standards, and WebSphere MQ messaging. CICS Transaction Server enables CICS applications to be used by any other IT system that supports SOA connectivity, enabling core CICS business processing to be extended to new business channels.

Web services

Web services are an industry-wide family of integration technologies based on open standards. The comprehensive Web services support in CICS Transaction Server include the core XML, SOAP and Web Services Description Language (WSDL) technologies, as well as comprehensive support for numerous Web services specifications including Web services distributed transactions (WS-Atomic Transaction), Web Services Security (WS-Security), and the Web Services-Interoperability (WS-I) Basic Profiles. CICS Web services support enables SOAP messages to be delivered over HTTP or WebSphere MQ transports. These capabilities run natively inside CICS Transaction Server.

CICS Transaction Server enables applications to be providers or requesters of Web services. Web services interact in a highly secure and reliable manner, independent of platform, environment or application language. Developers can rapidly build open-standards-based applications independent of the CICS business-logic program they will interact with. Web services are widely supported in almost every new IT application environment, and so the native Web services capability of CICS Transaction Server enables it to interoperate with the broadest range of other IT environments.

Java EE Connector Architecture

The Java EE Connector Architecture (JCA) defines a common standard for connecting from the Java EE specification to heterogeneous enterprise-information systems, such as CICS Transaction Server. The underlying infrastructure of the JCA automatically manages the connection pooling, transactional scope and security qualities of composite applications, so that these capabilities do not have to be individually coded into each application. This enables application developers to concentrate on the development of business logic rather than on quality-of-service provisioning.

IBM CICS Transaction Gateway provides a JCA connector that enables CICS Transaction Server to be a service provider for JCA requests. It delivers an adapter that plugs into the Java EE application server, providing connectivity between the Java application, the application server and CICS Transaction Server. Its attributes as a high-performing, highly secure and vastly scalable connectivity method has made CICS Transaction Gateway an extremely popular connector from WebSphere Application Server to CICS Transaction Server. Like WebSphere Application Server, CICS Transaction Gateway is available for multiple operating systems but has numerous advantages when running on z/OS, alongside CICS Transaction Server.

Enterprise messaging and WebSphere MQ

Enterprise messaging is an approach to SOA connectivity based on asynchronous queuing. Messages are sent to a queue by one IT system and retrieved from that queue by another IT system. This is an inherently scalable method of SOA connectivity because the requesting system does not have to wait for a response before it can continue processing other work.
IBM WebSphere MQ is the market leader in enterprise messaging. It can provide assured, once-and-once-only delivery of messages between CICS Transaction Server and over 80 other platform configurations, even if a temporary network failure occurs. It greatly simplifies application development for connectivity-related code by providing common, highly portable interfaces to send and receive messages. This eliminates most of the networking and error-handling complexities that programmers must otherwise handle themselves, in their code.

WebSphere MQ supports rapid growth and workload peaks by dynamically distributing work across available resources. Uniquely on the z/OS platform, WebSphere MQ delivers the highest levels of availability possible by taking advantage of Parallel Sysplex coupling facility technology to hold shared queues, thus removing any single point of failure. CICS Transaction Server has built-in interfaces that allow WebSphere MQ to optimize performance, integrate diagnostics and provide rich statistical information about the interactions between CICS Transaction Server and WebSphere MQ environments. A large percentage of CICS customers rely on WebSphere MQ for application integration.

Integrating CICS applications through an enterprise service bus

It is not the purpose of this paper to go into depth on the concepts and evolution of SOA, nor some of the technologies (such as business process management) that are enabled by it. However, to demonstrate how applications hosted by CICS Transaction Server can fully participate as first-class services within an SOA, it is important to discuss the concept of the enterprise service bus (ESB).

Creating CICS applications with standards-based SOA interfaces such as Web services, JCA or WebSphere MQ can dramatically increase the value of the CICS applications by promoting reuse. Reusing applications can significantly reduce the application development and quality assurance process. Supporting standards-based SOA connectivity increases the likelihood that your applications can be reused by other IT systems. However, if these applications are all directly connected to each other (for example, if the service provider is directly coupled to the service requester), the resulting point-to-point dependencies can increase complexity and reduce flexibility. Changes in one application might have to be synchronized with changes in all other directly connected applications. Additionally, as service provider applications are reused by multiple service requesters, the effects of application changes such as routine maintenance are multiplied. In these scenarios, despite applications having standard SOA interfaces, the direct coupling of applications becomes an inhibitor to growth rather than an enabler of it.

In an SOA where an ESB has been implemented, the service providers and service requesters never interact directly. Instead, they all connect to the ESB, and this ESB acts as the mediator between the service requesters and service providers. This decouples the requester's view of a service from the actual

Highlights

CICS applications that interact with other applications and services through an ESB can be considered truly loosely coupled, providing the most flexibility and the greatest potential for reuse.

implementation, greatly increasing the flexibility of the architecture. For example, a CICS application that is connected to other IT systems through an ESB can be upgraded, changed, moved or replaced without necessarily requiring the connected applications themselves to be changed. This is because the ESB contains elements of the connectivity logic, and the service provider applications and service requester applications are connected to this connectivity logic, rather than directly to each other. CICS applications that interact with other applications and services through an ESB can be considered truly loosely coupled, providing the most flexibility and the greatest potential for reuse.

IBM offers a variety of enterprise service bus solutions that provide different connectivity capabilities to meet the needs of different integration requirements. These capabilities include service routing, data transformations, aggregations, publish and subscribe, and much more. Integration with registry and repository technologies delivers the ability to store information about CICS services, and even to dynamically choose the most appropriate service at run time. It is not the purpose of this paper to describe these capabilities in detail, or to compare and contrast between these options. However, it is important to show that applications running in CICS Transaction Server can fully participate in an enterprise SOA by connecting to these technologies through standards-based SOA interfaces such as Web services, JCA or WebSphere MQ. It is also worth noting that in some cases, such as with WebSphere Message Broker and WebSphere Service Registry and Repository, advanced integration with CICS Transaction Server has been explicitly added to provide additional value beyond the core capabilities of these technologies.

Through the use of SOA interfaces and ESB technologies, applications hosted in CICS Transaction Server can be first-class participants in an SOA. This enables CICS applications to solve today's business problems of managing business growth, while also using the standards-based SOA connectivity technology that helps ensure interoperability with other IT systems. This makes CICS Transaction Server an attractive option for developing new SOA applications, or for porting existing applications that might currently be running in lower quality, less flexible distributed environments.

CICS tools to help you understand, optimize and manage your systems

As business requirements grow, so does the number of applications and the number of interactions between them. If not well understood, this can lead to decreases in flexibility that make it difficult to continue to add new capabilities. It can also lead to decreases in overall system performance that directly affect customer satisfaction. To assist CICS Transaction Server customers manage this business growth, IBM provides a number of tools to help them understand, optimize and manage their systems as their IT infrastructure grows.

CICS Performance Analyzer

In highly utilized CICS Transaction Server environments, understanding the performance and performance trends is critical to maintaining non-disruptive business growth. IBM CICS Performance Analyzer is a powerful tool that allows customers to monitor and improve the performance of CICS systems. It produces a wide range of reports and extracts to help tune and manage CICS systems, measure and understand the impact of changes, and plan capacity for new workloads. CICS Performance Analyzer works by analyzing System Management Facilities (SMF) records, which are data points that are produced by z/OS, CICS Transaction Server, DB2 and other subsystems and tools based on z/OS. It uses the results to provide useful information that enables CICS Transaction Server customers to make informed decisions about their environment.

CICS Performance Analyzer allows users to create their own reports, or to use more than 150 preconfigured (but customizable) reports to profile the performance of CICS systems. It is easy to tell how systems are responding to increased system workloads, application changes, or the addition of new applications to the system. If an application suddenly fails to meet its service level agreements, such as responding to the user within two seconds, this information is captured and can be displayed in a variety of reports. The historical database capability enables you to identify changes in application behavior over time, and thus better predict and manage CICS system-capacity requirements. Performance data can even be exported to spreadsheets for increased data manipulation, charting and reporting capabilities.

CICS Interdependency Analyzer

CICS Transaction Server environments usually run thousands of applications, many of which rely on each other for successful operations. So, understanding the relationships between these applications is vital. IBM CICS Interdependency Analyzer is an application-understanding tool that automates the detection of runtime relationships between CICS applications and resources. It enables IT staff to easily see all of the upstream and downstream dependencies on applications, so that when changes are made, the impact of those changes is clearly understood. CICS Interdependency Analyzer performs dynamic analysis that does not require source code to identify relationships between CICS applications and resources. It can also identify relationships to non-CICS resources, such a DB2 or WebSphere MQ. It builds a database of relationships that can be easily interrogated. It also provides an interface that includes a number of standard queries, along with the ability to easily customize these queries or build new queries from scratch.

CICS Interdependency Analyzer provides a deep understanding of applications that can dramatically reduce the time and cost of application changes, because dependencies do not have to be manually discovered and worked through. It can also significantly reduce the risk of unplanned outages or, worse still, undetected data corruptions, which can occur if application changes are made without a complete understanding of the impact of those changes. CICS Interdependency Analyzer is especially useful for customers that are experiencing rapid growth, because it allows programmers and managers to make informed decisions about the best way to split workloads for improved availability. This can help when implementing workload balancing across CICSplex and Parallel Sysplex environments to improve performance and provide continuous availability.

CICS Configuration Manager

Growing organizations need to grow their IT systems to support business expansions. From an IT perspective, this can mean making frequent system and application configuration changes—quickly and with minimal risk.
IBM CICS Configuration Manager is a tool to help control CICS Transaction Server system definitions from a single interface, even across a combination of environments that are managed by CICSPlex SM and those that are not. The

Highlights

tool provides the ability to create, edit, rename, compare, copy, move and remove definitions, either individually or in groups. Definitions can be searched and filtered by criteria such as name, type, group and attribute values, making changes much easier. Additionally, changes that are made in one environment can be automatically populated to other environments (such as newly cloned production environments), eliminating the opportunity for manual entry errors. Changes can be backed out, reverting to previous definitions even if they are several states back.

CICS Configuration Manager can also help organizations implement new regulations and organizational best practices that are increasingly focused on robust approval, auditing, and reporting processes, including IT configuration changes. In addition to making it quicker and easier to manage configuration changes, CICS Configuration Manager also provides comprehensive reporting capabilities for your resource definitions, along with audit log records for every new or modified configuration definition. There is broad change-control capability that can be optionally implemented to enforce procedures and require approval before certain changes can be made. This increases the rigor in configuration management and reduces the chance for configuration errors to lead to unplanned outages.

Summary of tools for the CICS environment

CICS Performance Analyzer, CICS Interdependency Analyzer and CICS Configuration Manager are just three of the products that IBM offers to make it easier to understand, manage and control CICS environments.

Another core advantage of running applications on the System z platform is the abundance of tools on offer to support these highly virtualized environments. CICS Performance Analyzer, CICS Interdependency Analyzer and CICS Configuration Manager are just three of the products that IBM offers to make it easier to understand, manage and control CICS environments. These three tools can also help system administrators perform CICS Transaction Server upgrades as new versions and releases become available, and are likely to be essential elements of any CICS Transaction Server administrator's toolkit.

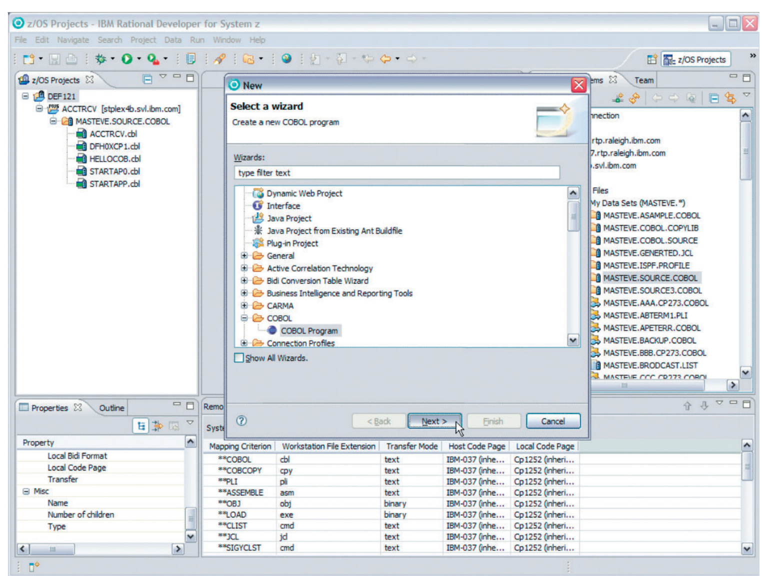
Modern application development using Rational Developer for System z

Throughout this paper, we have seen how deploying modern CICS applications within an enterprise-class SOA can facilitate rapid business growth. This section covers the development environment that facilitates the creation of these modern CICS business applications.

Highlights

IBM Rational Developer for System z is a modern Eclipse-based integrated development environment (IDE) that consists of a common workbench and an integrated set of development tools.

All developers need to be able to rapidly construct well-built, scalable composite business services, and this is no different for developers on the System z platform. IBM Rational Developer for System z is a modern Eclipse-based integrated development environment (IDE) that consists of a common workbench and an integrated set of development tools. It supports the end-to-end, model-based development, runtime testing, and rapid deployment of both simple and complex, mixed-language applications that can be deployed to the System z platform. Application components in various programming languages can be created and be deployed to a range of runtime environments including WebSphere Application Server and CICS Transaction Server. This means that developers with different technical backgrounds can use the same IDE to easily participate in important technology projects. CICS application developers can use the same modern tooling as their non-CICS counterparts to develop applications, facilitating shorter development cycles and fostering better teamwork.



Rational Developer for System z provides an interactive workstation-based development environment that helps developers create, maintain and enhance application components that can be deployed to a range of runtime environments including CICS Transaction Server and WebSphere Application Server.

Rational Developer for System z is a complete tool for the entire development community. Specialists with expertise in modern browser-based user-interface development can use visual layout tools to help them write creative JavaServer Faces (JSF), JavaServer Pages (JSP), and HTML Web-based applications that can be deployed to WebSphere Application Server. WebSphere business application developers can use tools to develop, test and debug Java and Java EE code such as Java servlets, Enterprise JavaBeans (EJB) and enterprise archive (EAR) files. Extremely efficient and high-performing applications and transactions hosted by CICS Transaction Server can be easily developed in core business languages such as COBOL, PL/I, C/C++ and Java. These applications can be integrated through tool-generated, standard SOA interfaces such as JCA and Web service interfaces. The entire project can be developed, tested and debugged end-to-end from a common, integrated and extensible Eclipse-based development workbench.

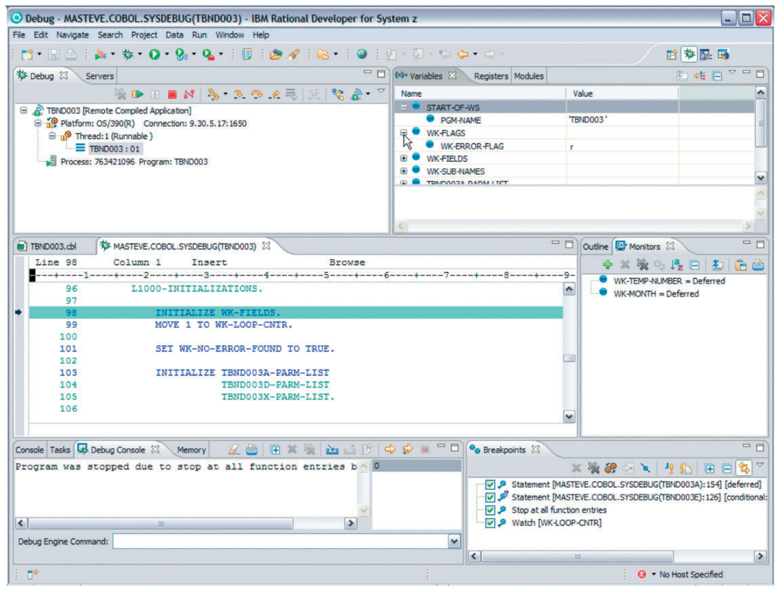
Rational Developer for System z supports increased developer productivity, shorter development cycles and better quality applications in numerous ways. For example, the workstation-based environment supports both local and remote development, meaning that developers can continue being productive, regardless of whether they have a live connection. With the Remote System Explorer (RSE) capability, developers can quickly and easily browse, create and manipulate System z file systems and data sets, as well as create and save predefined filters that speed access to remote resources. And tools such as the System z application pattern generator enables increased productivity and code quality by enabling composite applications to be created without the need to write any code. Applications with JSF-based front-end processing in WebSphere Application Server, connecting to core business COBOL processing in CICS Transaction Server, and accessing data that is stored in a DB2 database, can be entirely tool generated from Uniform Modeling Language (UML) and database schemas.

In addition to the core application-generation capabilities, extensive support is provided for developing, editing, assisting with content, checking syntax and building hand-written CICS Transaction Server applications coded in COBOL, PL/I, C, C++, and Java. Color-coded visual editing and the ability to easily jump to different sections of a program make it quick and easy to navigate application code. The predictive suggestion and real-time syntax validation of both the application language and the latest CICS Transaction Server application programming interfaces (APIs) can significantly speed up development and reduce coding errors. Support is provided for remote z/OS artifacts and dependency identification, such as remote copybooks, reducing the potential of compilation errors. And Rational Developer for System z can automatically generate and submit the job control language (JCL) scripts required to build and deploy applications on the z/OS platform, reducing the need to maintain scripts for routine tasks.

A set of enterprise service tools (graphical code-generation wizards) are provided to speed the development of various CICS projects. There is a tool that supports the creation of CICS Web services, providing bottom-up, top-down and meet-in-the-middle options. Bottom-up development enables the automatic generation of Web services components from existing CICS applications. Top-down development can build a CICS application structure from a Web Services Description Language (WSDL) file. And meet-in-the-middle is a combination of both, in which a mediation program maps a WSDL document to a CICS application. Once created, these CICS Web services can be tested within the tool using the Web services explorer function. Another tool, the Service Flow Modeler, supports the model-driven orchestration of micro-flows that are managed by CICS Transaction Server itself. This capability promotes code componentization and reuse, by linking highly reusable, fine-grained CICS applications together to create business-focused services. These services can then be exposed though numerous standard SOA connectivity options, including Web services, JCA and WebSphere MQ interfaces.

Integrated problem determination

Another major aspect of application development is the ability to quickly debug problems through rapid identification and resolution of application errors. IBM has developed an integrated set of problem determination tools specifically for the System z platform to help the application development community in areas such as source-code debugging, application abnormal-end-of-task (abend) analysis, data management and application-performance analysis.



Rational Developer for System z software provides integration with the IBM Problem Determination tooling, allowing developers to more productively test and resolve application problems.

Application Performance Analyzer

In highly virtualized environments where system resources are shared across multiple subsystems and applications, it is even more important to make sure that applications are designed for optimum efficiency. IBM Application Performance Analyzer is an application-performance measurement tool designed for use on z/OS systems to help ensure that wasteful code is not excessively using system resources such as processing time, real and virtual storage, or I/O bandwidth. Measuring performance helps provide the best response time for the system's users (for example, an organization's customers). The product's key function is to measure and report on how system resources are used by applications running in multiple languages including COBOL, PL/I, Java and C/C++, and across multiple subsystems, including CICS Transaction Server, WebSphere Application Server, DB2 and WebSphere MQ. It can also analyze and report on the performance of batch applications, helping for example, to reducing the time needed for nightly processing, or monthly processing.

Application Performance Analyzer helps you evaluate application prototypes and review the impact of changes in program source code, data volumes or increased workload. It provides online analysis capability to create displays of overall system activity, enabling you to check on all jobs running online. It can also generate historical data and reports to analyze performance trends over time. Application Performance Analyzer can measure the effects of the performance of all CICS applications, a group of CICS applications, or just those that meet specific criteria. Threshold monitoring can also be configured, so that statistics are gathered only when CICS applications exceed certain predefined thresholds for processor usage, I/O bandwidth or job duration, enabling you to identify potential issues before they affect production operations. It also integrates with other products in IBM's portfolio of problem determination tools for complete, end-to-end problem identification and resolution.

Debug Tool Utilities and Advanced Functions

To effectively build and maintain applications, System z developers need robust and easy-to-use tools to compile, test and debug them. IBM Debug Tool Utilities and Advanced Functions for z/OS provides a complete solution for System z developers that includes runtime debugging of applications in CICS, batch and DB2 environments. COBOL, PL/I, C and C++ programs can be compiled and link-edited, and CICS applications that query DB2 databases can be preprocessed and compiled. CICS applications can be interactively debugged as they are running, enabling developers to seamlessly debug mixed-language applications. Adjustments can be made during program execution to display, monitor and alter program variables. Debug Tool Utilities and Advanced Functions also provides coverage tools that enable test cases to be analyzed to determine how thoroughly they validate the quality of applications before they are promoted into a production environment.

Debug Tool Utilities and Advanced Functions can integrate with Rational Developer for System z to provide a complete development, test and debugging environment for all applications running on z/OS. Source-level debugging capabilities allow developers to view the source code and focus on problem areas by stepping through it line-by-line during the execution of an application. Developers can set breakpoints in an application program, monitor variables for changes, and watch for specified exceptions and conditions during program execution. Dynamic patching provides the ability to make adjustments to applications while they are being debugged. Using the Rational Developer for System z integration, it is possible to end-to-end debug a composite application that has components running in both CICS Transaction Server and WebSphere Application Server environments—all in a single Eclipse-based workbench.

Fault Analyzer

When an application ends abnormally, it is important to be able to quickly fix the problem by identifying the cause and analyzing the failure. IBM Fault Analyzer is a powerful abend-analysis tool that enables developers to determine the source of system and application failures across multiple subsystems, including CICS Transaction Server, WebSphere Application Server, WebSphere MQ and DB2. It supports applications running in multiple languages including COBOL, PL/I, C/C++ and Java. Fault Analyzer helps to repair failures quickly by gathering information about an application and its environment at the time of failure. The data can be used to analyze failures either as they occur for immediate problem determination, or later, to help speed future application development. The tool is designed to reduce costs and increase productivity of application development and testing on the System z platform.

Fault Analyzer records each application abend in a fault-history file that includes the details such as job name and failure code, along with the analysis report and storage pages referenced during the analysis. Fault Analyzer also extracts message and failure-code descriptions from product manuals and inserts them into the analysis report where applicable, freeing developers from researching the explanation of message and failure details. Fault Analyzer can also integrate with Rational Developer for System z to allow application developers to work with fault entries. Developers can browse fault entries, link to source listings of a program experiencing abends and display formatted dump storage directly from the Eclipse-based environment of Rational Developer for System z.

Highlights

Application Performance Analyzer, Debug Tools Utilities and Advanced Functions, Fault Analyzer and File Manager are key problem determination tools, each of which has extensive support for multiple languages and subsystems. They integrate well with each other, and also with IBM's strategic System z application development workbench, Rational Developer for System z.

File Manager

Application developers need to work with data, such as that stored in files or databases, or data in storage that is being used by an application. IBM File Manager provides a user interface to work with these file and data resources on z/OS, such as program data sets and VSAM files, as well as data under the management of z/OS subsystems such as CICS Transaction Server and DB2. Additional support is provided to view and manipulate COBOL and PL/I copybook-record layouts. File Manager can be used to find, copy, edit, print, compare and display data quickly and easily, helping to increase productivity and lower the costs of building and testing applications. File Manager also provides a range of control facilities that can help the application development community to respond to increasing governance and compliance requirements. A scrambling algorithm modifies data in a field while maintaining its system data type, allowing developers and testers to use production data without exposing sensitive information.

File Manager has excellent support for the DB2 environment, providing utilities for editing, browsing, printing, copying and maintaining DB2 data. It also provides a number of utilities that perform functions such as listing DB2 objects, managing DB2 privileges, creating data to populate DB2 tables, prototyping SQL SELECT statements and much more. For the CICS Transaction Server environment, File Manager has a powerful feature that provides access to VSAM files as well as the temporary storage queues and transient data queues that are under the management of CICS Transaction Server. It also provides services to inquire and modify the status and the various attributes of the supported CICS resources. File Manager can also optionally integrate with Rational Developer for System z for editing and browsing of z/OS file-based data, and for viewing and editing of template-based and copybook-based data.

Summary of problem determination tools

Application Performance Analyzer, Debug Tools Utilities and Advanced Functions, Fault Analyzer and File Manager are key problem determination tools, each of which has extensive support for multiple languages and subsystems. They integrate well with each other, and also with IBM's strategic

System z application development workbench, Rational Developer for System z. For System z application developers, these tools can help to deliver the complete end-to-end problem identification and resolution capabilities that increases productivity and overall application quality.

Additional products in the CICS Transaction Server ecosystem

During the course of this paper we have mentioned a number of key IBM product offerings that can support the development, deployment and SOA integration of new IT projects involving CICS Transaction Server. This list, however, is by no means exhaustive, and there are literally hundreds of IBM products that can support IT projects in which CICS Transaction Server is a core component.

IBM WebSphere solutions provide wide-ranging capabilities for application integration, business process management and composite business services, enabling CICS applications to participate in the most sophisticated of SOA solutions.

IBM Tivoli® solutions offer complete, end-to-end solutions that deliver the visibility, control and automation needed to deliver consistent quality in all areas of IT management that include CICS systems and applications.

IBM Rational solutions provide the application development community with the ability to help model, develop, deploy and manage composite applications that include CICS components, throughout the application life cycle.

IBM Information Management solutions provide the database servers and data management tools to store, manage and analyze the single view of data that is always concurrently available to CICS Transaction Server and other systems.

IBM Lotus solutions provide the communication and collaboration software that can draw upon the valuable information provided by CICS systems and applications to enable teams of people to take full advantage of their collective knowledge.

Summary

Companies that need to be sure that their new IT projects can support their future business growth should consider deploying new applications to a centralized System z platform and an environment that is managed by CICS Transaction Server for z/OS.

- *Organizations that are experiencing business expansion are increasingly turning to IBM System z servers and software to help them manage IT growth without exponentially increasing IT cost and complexity.*
- *CICS Transaction Server delivers a managed environment on the System z platform that supports extremely high-performing and ultra-efficient transactional applications.*
- *The virtualization capabilities of CICS Transaction Sever and the System z platform can provide levels of scalability and availability required to meet the needs of even the largest and fastest growing enterprises, both today and in the future.*
- *There is a wealth of supporting software and tooling to support the modern development, deployment and SOA integration of CICS applications.*

For more information

To learn more about IBM CICS Transaction Server for z/OS, please contact your IBM marketing representative or IBM Business Partner, or visit the following Web site: **ibm.com**/cics

IBM United Kingdom Limited
Hursley Park
Winchester
Hampshire
SO21 2JN
United Kingdom

Produced in the United States of America
July 2008

TAKE BACK CONTROL WITH **IBM® Middleware**

WSW14027-USEN-00