April 2005

Experience with Linux on zSeries running InformationBuilders Inc. (IBI) Business Intelligence Workloads

Table of Contents

Acknowledgments

IBM would like to thank the following for their assistance in the creation of this white paper.

Information Builders Inc. http://www.informationbuilders.com/

IBM New York City Account Team - Jeff Sklar

Washington System Center - Richard Lewis

IBM Poughkeepsie zSeries Benchmark Center - Tangy Ward and Nin Lei

Background

An account team from the IBM New York office was helping a customer explore the possibility of using Business Intelligence software from InformationBuilders Inc. (IBI) on an IBM zSeries server running Linux. A complete description of the IBI products and their uses, can be found at the IBI Web site at: www.informationbuilders.com/.

The applications that the customer was using were running on an older release of IBI software, and they were hosted on older non-IBM RISC architecture hardware. The customer was looking for alternative methods of deploying these applications that would provide better and more effective resiliency in the event of a disaster. In addition to strengthening the disaster recovery options of the solution, the performance, reliability, scalability and maintainability aspects of the solution were also examined with the goal of improving them as well.

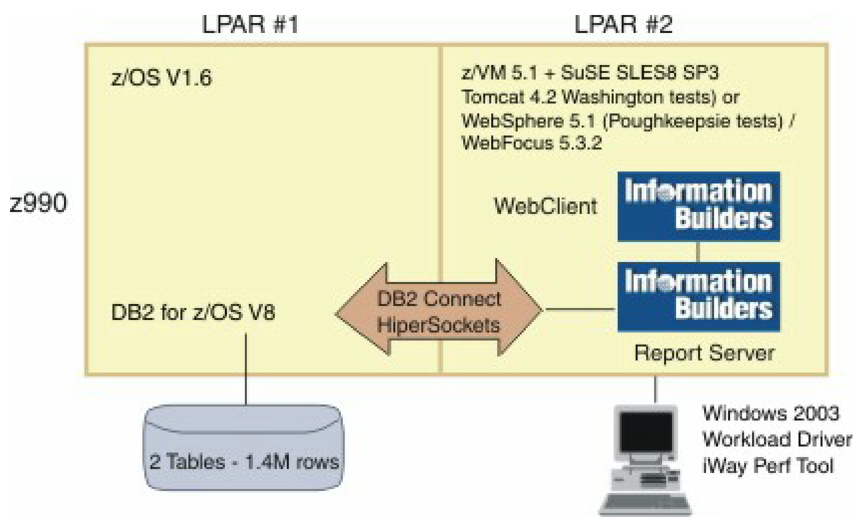
A plan was developed by the account team led by Jeff Sklar (jsklar@us.ibm.com) to model the performance and scalability of current IBI WebFOCUS software on IBM @server® zSeries® servers running Linux®. With the help of Information Builders Inc., initial testing began at the IBM Laboratories in Poughkeepsie, New York, in October 2004.

Initial Test Configuration

Initial testing was done using an IBM @server zSeries 990 (z990) server at the IBM zSeries Benchmark Center in Poughkeepsie, New York. A z990 server was made available to support the test. The processor was logically partitioned, with z/OS® Version 1 Release 6 running DB2® for z/OS Version 8 in one logical partition (LPAR) with four processors . The DB2 database was populated with data representative of the data used by the customer. In the second LPAR, z/VM® Version 5.1, SUSE SLES 8 Service Pack 3, WebSphere® Application Server 5.1.1, and IBI WebFOCUS Version 5.3.2 were installed to provide a WebClient and Report Server. The Linux LPAR was configured as a middle tier guest consisting of the Apache, WebSphere, and WebFocus Web Client software, as well as a WebFocus Report Server to issue SQL and format results. Four Integrated Facility for Linux (IFL) processors were made available for LPAR 2. The illustration (Figure 1) shows the structure of the logical partitions. Communication between the two logical partitions was done with the zSeries HiperSockets™ function.

The IBI product includes a tool for performance evaluation, the iWay performance tool. The tool simulates 100 concurrent users reading two DB2 database tables. It is designed to issue a continuous stream of SQL data requests, and it records a wide variety of performance data (e.g., response time, data transfer time, processing time).

Figure 1



The results of the initial testing showed linear scalabilty at one to four Integrated Facility for Linux (IFL) processor engines for the Linux environment. Based upon the results of that test, more resource was sought to determine if excellent performance could continue beyond the four IFL engines.,

Testing activities were then moved to the IBM Washington System Center where a z990 server with more processors was available for use.

The IBM Washington System Center Tests

The IBM Washington System Center was able to offer a z990 server with more available resources than the servers available in Poughkeepsie at that time. The goal of using more resource was to examine the impact upon performance of the zSeries/Linux/DB2/IBI configuration as more processing resource was added.

IBI ran a series of tests on five distributed environments using industry standard data and the software described earlier in this document and in Figure 1. Every effort was made to keep as many aspects of the IBM Washington System Center benchmark similar to the other benchmarks that had been conducted by IBI on other platforms. The software and industry standard data used at the Washington System Center was chosen deliberately to be identical to the configuration used by IBI. In the IBI tests, the hardware used was Intel® and RISC-based hardware from various manufacturers. Database software used by IBI was, in some cases, not DB2 Universal Database™. However, the data loaded into DB2 on z/OS for the Washington System Center benchmark was identical to the data used in the IBI tests, as was the logical design of the database. The actual SQL statements representing the work being performed was identical with other IBI benchmarks as well. In addition, the same IBI product set that was used in the IBI benchmarks was used in the IBM Washington System Center benchmark. Finally, the number of concurrent users (100), and minimal think time used were also identical to previous IBI benchmarks. The IBI and IBM Washington System Center tests both used the IBI iWay tool for performance measurement.

The IBI Washington System Center benchmark differed from previous benchmarks by using the zSeries hardware platform, and by using DB2 runing on the IBM z/OS operating system for the back end database, and by using Linux on zSeries with Apache / Tomcat and WebSphere for the application servers.

Test Environments:

The benchmark environment was established to remove any constraints that might prevent the WebFocus product from fully utilizing all CPUs that were made available. Thus, much more memory than would have been needed was allocated, along with a large number of FICON® channels to make sure that I/O to DASD did not become a bottleneck (turns out very little disk I/O occurred). In addition, the number of active virtual machines was kept to a minimum to avoid any potential interference. Lastly, the number of physical CPUs available exceeded what would be needed by the test virtual machines in all of the tests except the ones above 8 virtual processors. In the case where each virtual machine had 16 or 24 virtual processors, the total number of virtual processors exceeded the number of physical processors available to the test LPAR.

The benchmark team was prepared to deploy additional WebFocus virtual machines if it appeared as though one virtual machine could not fully utilize the maximum number of CPUs available in the test. However, this was never needed since a single WebFocus virtual machine was always able to fully utilize the available CPUs.

The benchmark design consisted of a three-tier and sometimes a four-tier environment. The client tier was external to the zSeries server, and consisted of two IBM @server xSeries® boxes. The IBI iWay workload driver was installed on these boxes to simulate 100 active users submitting requests at a rate of 300 / minute with virtually no think time in-between. This was done to match the environment IBI had used in previous benchmarks, and to provide a significant load on the zSeries to drive up processor utilization.

The second tier was varied between an application server environment (Apache / Tomcat, or WebSphere), and communication directly with the http listener in the WebFocus product. This was done to compare utilization without an intervening application server (which was expected to be higher) against utilization going through an intervening application server. The WebFocus product made up the third tier, when tests went through an intervening application server, and the second tier when an application server was not used.

The final tier was a DB2 V8 database in a z/OS 1.6 LPAR. This database contained the rows being requested by the client, and formatted by the WebFocus reporting product. Each client request resulted in either 61 or 3000 rows being retrieved from the backend database. Communication between the database and WebFocus product was with DB2 Connect™ V8, running in the same virtual machine as the WebFocus product. Physical connectivity between the WebFocus virtual machine and DB2 in the z/OS LPAR used a real HiperSockets connection. The HiperSockets connection was configured to use the largest maximum frame size possible.

With each number of CPUs being allocated, 6 tests were conducted. These tests were named as follows:

Workload Name	Description
Backend Small	xSeries to WebFocus run without intervening application server, retrieve 61 rows - note non-web
Backend Large	xSeries to WebFocus run without intervening application server, retrieve 3000 rows - note non-web
Tomcat Small	Client to Apache/Tomcat application server. Request to WebFocus comes from application server, 61 rows. - note uses InformationBuilders Web Client
Tomcat Large	Client to Apache/Tomcat application server. Request to WebFocus from application server, 3000 rows. - note uses InformationBuilders Web Client
WebSphere Small	Client to WebSphere application server. Requests to WebFocus from application server, 61 rows. - note uses InformationBuilders Web Client
WebSphere Large	Client to WebSphere application server. Requests to WebFocus from application server, 3000 rows. - note uses InformationBuilders Web Client

Memory

When Linux is running in a shared memory environment, such as in a z/VM virtual machine, it is desirable that memory be used in as efficient a manner as possible. Unfortunately, Linux typically uses all memory not consumed by process requests for page cache and buffer space. Consuming a large number of pages for these uses is not really necessary in a zSeries environment, since I/O operations are already being cached at multiple levels.

Since the tests conducted in this benchmark did not involve a large amount of disk I/O, the Linux virtual machines did not use a large number of pages for buffer or page cache. Thus the Webserver virtual machine as well as the WebFocus virtual machine did not have z/VM Working Set Sizes (WSS) that were large percentages of the total virtual machine size. Thus for example, the Web server virtual machine's largest observed WSS during the 24 CPU test was about 55% of the virtual machine size.

The memory footprint of the WebFocus virtual machine when compared with the memory footprint of the Web server virtual machine during the 24 CPU run reveals that the WSS of the WebFocus virtual machine was on average less than ½ of the WSS of the Web server virtual machine. Both virtual machines had a 2 GB virtual machine size. In addition, as the tests progressed with 24 CPUs, the rate of change in WSS was less than half in the WebFocus virtual machine when compared with the Web server virtual machine.

The following chart provides a comparison of the memory footprint between the WebFocus virtual machine and the Web server virtual machine for the 24 CPU tests.

Virtual Machine	Minimum WSS	Maximum WSS	Range of WSS
WebFocus	450 MB	536 MB	86 MB
Web server	974 MB	1138 MB	165 MB

This comparison seems to indicate that the WebFocus product is a good candidate for running in a shared memory environment such as a z/VM virtual machine on zSeries. Even with a large number of CPUs, the WSS was only about 22 – 26% of the total virtual machine size. In addition, as processing became more intense with large row tests, the fluctuation of WSS was rather small considering the total virtual machine size.

It is important to mention here that the WSS would no doubt have been much larger due to Linux buffer and page cache if a large amount of disk I/O had occurred. However, what this data reveals is that it would be possible to trim the virtual machine size down significantly to reduce Linux buffer and page cache without substantially impacting the application. This conclusion is based upon the small footprint observed, and the small fluctuation that occurred during the tests.

Scalability

Initial testing at the IBM Washington System Center indicated similar scalability results to those observed during the Poughkeepsie tests using the data similar to the customers. The table below illustrates a decrease in response time as more processors are added to the logical partition with the IBI, Tomcat and Linux servers. Note that linear scalability is evident as processors are added, which illustrates a virtual absence of performance bottlenecks. Much of this can be attributed to the zSeries memory and caching structure and to HiperSockets, which are used for communication between DB2 UDB running under z/OS in LPAR 1, and the Linux based IBI software in LPAR 2.

	1 Processor	2 Processors	3 Processors	4 Processors
Response Time*	19.2	8.3	5.4	4.2
Scalability	baseline	116%	119%	114%

* average response time in seconds for 150 users

Performance

	2 IFL	4 IFL	8 IFL	16 IFL
z990	6.91	3.43	1.64	1.49

* Average response time in seconds for 3000 row data retrieval

All performance runs on the z990 server were done using 4 processor engines to support the DB2 UDB database running under z/OS in LPAR 1. LPAR 2 containing the IBI software running under Linux and z/VM was tested with 2 to 24 Integrated Facility for Linux processors. As the data in the above table illustrates, the z990 centralized environment was able to scale well up to 8 processors. It should be noted that the IBI Report Server is able to function without the IBI Web Client. Subsequent testing at the IBM Washington System Center showed that the IBI report server, in the tested configuration minus the Web serving software, could scale effectively using up to 24 Integrated Facility for Linux processors.

zSeries Performance Advantages

The zSeries server is the result of over 40 years of continuous innovation and refinement. Elements of the zSeries architecture are key in producing results like those illustrated in this document. Of special note are HiperSockets, context switching, and the zSeries hardware architecture as it relates to cache.

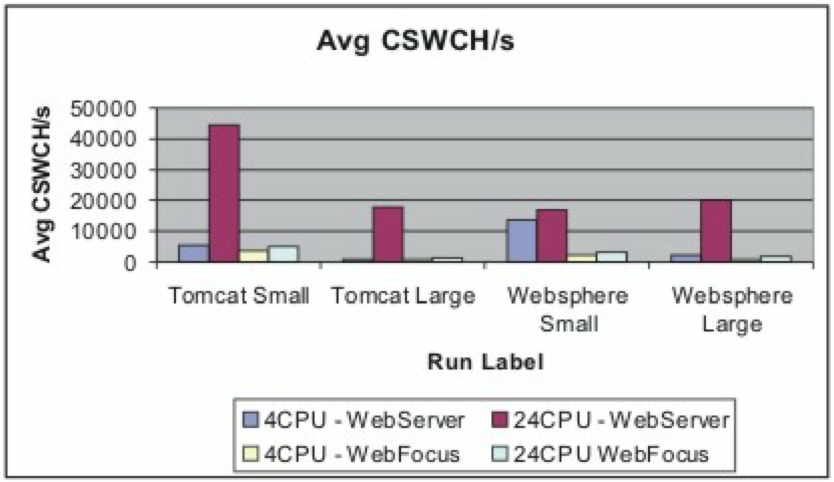
Context Switching

Since the WebFocus product as well as application servers are multi-threaded applications, it is helpful to look at context switching rates for a rough idea of application efficiency. In addition, this is an interesting metric to examine when considering how the application behaves as more CPUs become available.

The following chart (Figure 2) shows average rates of context switching per second for two of the tests (4 CPU and 24 CPU). As can be seen from the chart, the Web server virtual machine typically incurs a higher rate of context switching than the WebFocus virtual machine.

In addition, as the number of CPUs is increased from 4 to 24, there is a fairly large jump in context switching for the Web server virtual machine. There is a much smaller jump in context switching for the WebFocus virtual machine. For example, the average rate of context switching per second for the WebFocus virtual machine with 4 CPUs was 8120. When the number of CPUs was increased to 24, the average context switching rate increased to 12,407 per second, representing a 53% increase. The Web server incurred an average context switching rate of 5653 per second for the Tomcat Small test with 4 CPUs, and an average rate of 44,628 per second for Tomcat Small with 24 CPUs. This represents an increase of 689%!

Figure 2



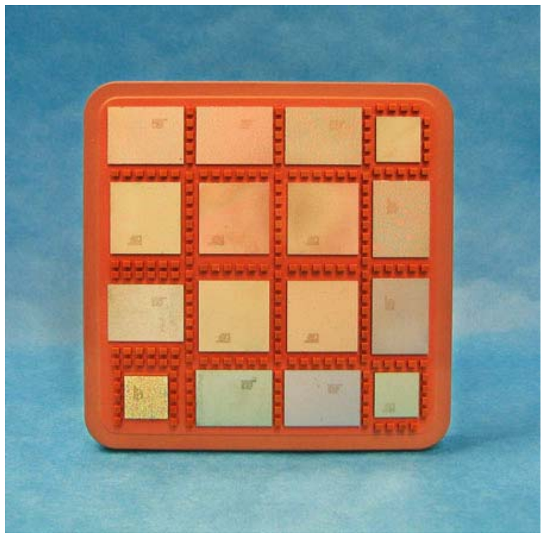
HiperSockets

Communication between servers can be a key element in end-to-end performance in a solution. HiperSockets is a unique zSeries function that allows communication between servers installed on the same zSeries server. Figure 1 shows how HiperSockets were used to provide communication between the DB2 UDB database running on z/OS, and the IBI Web Client and Report Server running on z/VM and Linux. Simply described, a HiperSockets is an in-memory implementation of a TCP/IP network. By moving the network into memory, several advantages can occur. A decreased need for cabling, switches and routers is usually possible since those functions are now provided internally. Second, the latency associated with physical networks, multiple server “hops” is greatly reduced. Third, security is improved since an eavesdropping device (commonly called a “sniffer”) that can be placed on external cabling, cannot be placed on a HiperSockets since inter-server communication now takes place within the zSeries server. Finally, network maintenance may be reduced since HiperSockets do not require complex cable routing and switching to link servers.

Cache Structure

The ability to access cache data rapidly contributes to several key aspects of performance. The ability to manage cache data efficiently can be seen at the very heart of the zSeries server, the z990 Muti-Chip-Module (MCM). The MCM consists of 101 layers of ceramic substrate. The MCM contains 400 meters of copper wire internally to provide connection between the elements on the MCM.

Figure 3
The z990 Multi-Chip Module



There are 12 processors on the MCM. They are dual core chips, which means that two processors can reside on the same chip. The processors are the rectangular chips on the outer edges of the MCM. Eight of the processors are for customer use and can be tailored to customer needs. For example they can be standard processors, Integrated Facility for Linux processors (IFLs), zSeries Application Assist Processors (zAAPs), or Coupling Facility. Two of the processor chips are System Assist Processors (SAP's), used for tasks like I/O operations, and two are spares, held in reserve in case of the failure of other processors. Each of the processors has 512K of Level One Cache. In the center of the MCM are four square chips, which house 32 Megabytes of Level 2 cache. The Level 2 cache is shared by all of the processors on the MCM. This design makes it possible to keep Level 2 cache repopulating to a minimum, and it also helps to reduce context switching latency since Level 2 cache hits can be serviced without having to leave the MCM.

Other zSeries advantages

Performance and scalability alone are not the only advantages delivered by a zSeries environment. Responsiveness, economy, flexibility, manageability, and the ability to grow on demand are other examples of the advantages provided by zSeries servers.

Resiliency

The zSeries product line is designed to offer layer upon layer of fault tolerance and error checking features. If a failure occurs, the built-in redundancy on the zSeries platform is intended to shift the work over from failing components to ones that work to prevent the end-user service from being interrupted. The failed components may be removed and replaced while the processor is still active, so service may continue.

There is more to availability than just the server being up – the application and the data must be available as well. zSeries platforms' availability features include the hardware, the operating system, application and database availability and the connection to disk. At the heart of zSeries platform availability is IBM Geographically Dispersed Parallel Sysplex™ (GDPS®) technology, which is positioned to provide a comprehensive business continuity solution for the z/OS platform. Based on geographical separation and automation, GDPS is a multi-site application availability solution designed to provide the capability to manage remote copy configuration and storage subsystem(s), automate Parallel Sysplex® operational tasks and perform failure recovery from a single point of control. GDPS provides the resource sharing, workload balancing and near continuous availability benefits of a Parallel Sysplex environment. It can also significantly enhance the capability of an enterprise to recover from disasters and other failures and to manage planned exception conditions, helping businesses to achieve their own near continuous availability and disaster recovery goals.

GDPS/PPRC has been enhanced to provide a new function called "GDPS/PPRC Coordinated Cross Platform Disaster Recovery for zSeries." This function is especially valuable for customers who share data and storage subsystems between z/OS and Linux on zSeries. An example of this type of application is the one illustrated in this document. Since the solution utilizes a mult-tiered acrhitecture, there is a need to provide a coordinated Disaster Recovery solution for both the z/OS and Linux on zSeries based components of the solution. GDPS/PPRC can now provide that.

Growth on demand

The zSeries server is able to scale up and scale out on demand. To scale up means to be able to add more processors for more capacity, as it is needed. zSeries servers provide a function called Capacity Upgrade on Demand. Capacity Upgrade on Demand gives a user the ability to log onto a secure Internet site with an IBM supplied password to order more compute power as it is needed. The process is accomplished quickly and efficiently electronically. No parts need to be shipped, and no engineer has to appear at the customer site.

Scaling out means adding additional virtual servers as needed. In the tests described in this document, z/VM is used to provide scale-out capability for the Linux based servers. Using the functions of z/VM, it is possible to define and provision a working Linux server in a matter of minutes, as opposed to hours or days typically required in distributed environments.

Management of the on demand IT infrastructure

Managing the complexity and the cost of an IT infrastructure has been an increasingly difficult task for IT managers over the years. As server farms have grown dramatically, and the demand for more applications with greater degrees of complexity have grown along with them, the desire to simplify the computing infrastructure is seen as a key requirement for IT executives. zSeries servers can be a key element in the simplification of an IT infrastructure.

In the tests illustrated in this document, multiple servers were used that resided on a single physical zSeries server through the function of zSeries virtualization. However, zSeries virtualization goes beyond the mere creation and deployment of virtual servers. Management software to create, deploy, provision, and manage virtual servers is available from IBM Tivoli® and other vendors. There is also function throughout zSeries to help manage the workloads deployed on the zSeries server. Workload management functions are available for z/OS and z/VM based workloads to support policy driven resource assignment. The purpose of the workload management function is to automatically supply resource to key applications to support user defined service agreements.

Network management is also a key element in the operation of an on demand IT infrastructure. As noted earlier in this document, the zSeries HiperSockets function helps minimize the complexity, cost, and management of the network by reducing the external physical elements associated with a network while improving the responsiveness.

Economy

Economy can be measured in many ways; several have no doubt been obvious during the reading of this paper. As the need for virtual servers grew, they could be defined by z/VM. Using virtual servers, as opposed to physical servers, does not require purchasing a new physical server for each virtual server required, nor does it require additional floor space, and power costs remain constant. Additonally, a cost benefit in licensing charges may be seen, particularly in Linux environments. Typically Linux software is charged on a "per processor" basis. In a distributed environment, processor resource can mount up quickly with the addition of distributed servers. Each server may have more than a single processor in use, and therefore a software licensing charge will be incurred for each processor. The z990 model used in the IBM Washington System Center for the tests described in this document is a 32-way Symmetrical Multi-Processor which can support hundreds of virtual servers.

However the maximum number of processors that could be billed is 32 on a fully configured Linux only system.

However, Linux was not the only workload running in the tests. DB2 UDB was running under z/OS. Note that the Linux workloads were run in Integrated Facility for Linux processors, as opposed to standard processors. An Integrated Facility for Linux is a processor that runs at maximum speed, and supports a Linux workload exclusively. Other mainframe operating systems like z/OS and TPF will not run on an IFL. Therefore the IFL has no affect on the cost of the software running on those operating systems. The benefit of this technology is providing the flexibility of adding compute power to a zSeries processor for Linux without incurring a financial penalty in the cost of traditional mainframe software.

Another way that economy can be considered is the cost of not having application resource available. The cost of lost business or manufacturing delay can be formidable. To help reduce the incidence of unavailable resource, zSeries servers are designed to be highly reliable. An example of this was given in the section where the Multi-Chip Module was described. Spare processors are provided in the event of the failure of a processor. They can be varied on-line automatically and transparently to keep applications available. Spare memory is provided for the same reason. The result of these, and other design elements to enhance reliability are what has given the IBM mainframe its legendary reputation for reliability and a mean time between failure measured in decades.

Conclusion

The IBM Washington System Center benchmark demonstrated that WebFocus is able to increase the number of requests processed and reduce average response time when the number of CPUs available are increased up to 24 CPUs. This was the goal of the benchmark, and the data produced from the benchmark shows that the goal was met. The results of the testing illustrated zSeries servers' ability to scale and respond very well while running the IBI Business Intelligence workload. The tests also gave the zSeries server the opportunity to illustrate its responsiveness, manageability, and potential for growth in an on demand IT infrastructure. For these reasons, and the solutions for increasing resiliency, the zSeries based solution was the choice of the customer.

Copyright IBM Corporation 2005

IBM Corporation
Marketing Communications, Server Group
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
04/05
All Rights Reserved

IBM, IBM eServer, IBM @server, IBM logo, DB2, DB2 Connect, DB2 Universal Database, FICON, GDPS, Geographically Dispersed Parallel Sysplex, HiperSockets, Parallel Sysplex, Tivoli, WebSphere, xSeries, z/OS, z/VM, and zSeries are trademarks or registered trademarks of International Business Machines Corporation of the United States, other countries or both.

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

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

ON (LOGO) DEMAND BUSINESS is a trademark of International Business Machines Corporation.

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

Intel is a trademark of Intel Corporation in the United States, other countries or both.

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

Information concerning non-IBM products was obtained from the suppliers of their products or their published announcements. Questions on the capabilities of the non-IBM products should be addressed with the suppliers.

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

IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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

GM13-0700-00