Redbooks Paper

Mainframes and GIS: A New ESRI Solution on IBM zSeries using Linux

IBM and ESRI are leading the development of Geographical Information Systems (GIS) solutions away from a specialized departmental technology into a more comprehensive enterprise-wide platform. Enterprise GIS now contains all the reliability, manageability, and performance requirements demanded by Enterprise Information Technology solutions.

Customer demand encouraged IBM and ESRI to expand GIS enterprise solutions to use mainframe database repositories for spatial information and analysis. Therefore, IBM undertook a study at the IBM @server zSeries Benchmark Center in Poughkeepsie, NY to determine how to utilize the zSeries platform in the ESRI GIS solution. WIth the successful outcome of this project, there is more to come from IBM and ESRI.

ibm.com/redbooks   **1**

GIS overview

GIS is a set of information technologies that assemble, store, manipulate, and display data by geographic location. ESRI is the leading software provider in this rapidly expanding field. ESRI GIS software helps thousands of organizations transform statistical and geographical data into meaningful information and provides visual answers to questions about customers, sites, supplies, logistics, delivery, and strategic operations. Organizations can use this software to present information in the form of an electronic map, positioning them to organize, share, and visualize data so that communication is improved and decision making is enhanced. GIS plays a role in the process of solving critical business problems for customers in industries such as government, utilities, retail, banking, insurance, oil and gas, transportation, and health.

Until now, the GIS solution was implemented solely on IBM @server xSeries and pSeries systems, as well as IBM BladeCenter servers. However, 60% of the world's data is processed on zSeries mainframes. Many customers have the capacity and want the reliability that zSeries offers them. Thus, they came to IBM and ESRI with a business challenge.

Business challenge

Many commercial and public sector users have chosen the zSeries platform for business-critical applications and as the basis for On Demand Business because it provides a scalable and highly available application environment. See "Why Linux on zSeries?" on page 4. A number of users have expressed an interest in using their current zSeries systems to manage spatial data with their traditional data in an IBM DB2 database. Such data management is especially valuable when it is done in conjunction with the GIS tools provided by ESRI. The interest is based largely on the following factors:

► Their existing zSeries systems have hardware capacity available.

► They would like to move GIS data from separate departmental systems over to a central location for ease of application management.

► They prefer to avoid installing and maintaining a separate system for spatial data.

The customer interest in GIS support on zSeries provided IBM and ESRI with an opportunity to investigate the configuration and suitability of the mainframe platform for spatial applications. Under Linux® on zSeries, the DB2 Spatial Extender offers functionality identical to Windows®, UNIX®-based, and Linux platforms. This IBM Redpaper reports the testing methodology and results of the investigation of the Linux for zSeries platform.

ESRI typical configurations

ESRI products currently support DB2 with the Spatial Extender as a repository for spatial data in a wide variety of Windows and UNIX-based platforms. IBM and ESRI have published a paper called *Infrastructure recommendations for ESRI and IBM Geographic Information Systems,* REDP-9128, that describes configurations and characteristics for the xSeries and pSeries platforms. This paper can be found at:

> http://www.redbooks.ibm.com/abstracts/redp9128.html/

Software components

A number of IBM and ESRI software components are available for composing a solution:

► **DB2 Connect**

   Provides a client application interface to the DB2 Universal Database (UDB) server

► **DB2 Client Support**

   Part of DB2 UDB that provides the interface for local applications and client applications with DB2 Connect

► **DB2 Spatial Extender**

   A feature of DB2 UDB that provides support for spatial types, spatial functions, and spatial indexes based on the DB2 UDB object relational architecture

► **DB2 UDB**

   Server for relational database management system (RDBMS)

► **ESRI ArcGIS desktop and developer products**

   User software for developing and running GIS applications

► **ESRI ArcSDE**

   Middleware that provides an RDBMS-independent interface to ArcGIS desktop and developer products and manages multiple client connections

Common configurations

There are a number of ways that you can arrange the components in a solution. The common configurations have either two or three tiers.

Figure 1 uses colored arrows to illustrate three ways that the ESRI software can be configured with DB2 UDB. The traditional approach is a three-tier solution with the ArcSDE middleware between the ArcGIS client and the RDBMS server. In the last few years, a two-tier solution has become more popular, with ArcGIS using what is termed "direct connect" to access the data directly. This approach incorporates ArcSDE functionality in the ArcGIS client.
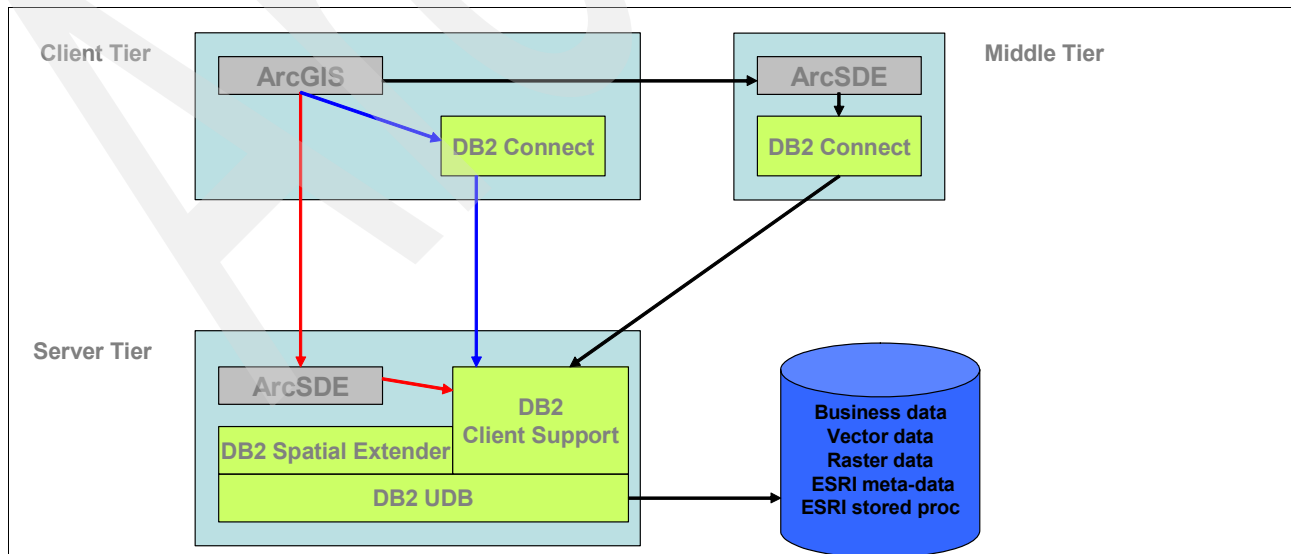

Figure 1   Tier and platform options

Three-tier two-platform (red arrows)

This is historically the most common approach. The ArcSDE middleware and RDBMS server are on the same platform (UNIX, IBM AIX, Solaris, HP-UX, Linux, or Windows) with the ArcGIS client on Windows. The advantage of this approach is that it provides the fastest communication between ArcSDE and the RDBMS. A potential disadvantage is that ArcSDE and the RDBMS must share cycles and memory on the server.

Three-tier three-platform (black arrows)

In this approach, each tier resides on its own hardware and software platform. This has the advantage of processing cycles on the platform dedicated to their respective software. You can scale ArcSDE by installing more middle-tier systems that share the same RDBMS tier. A disadvantage is the longer communication path between ArcSDE and the RDBMS. This solution is also more complicated to install and administer.

Three-tier two-platform, direct connect (blue arrows)

This is an increasingly popular approach that eliminates the middle tier. In this approach, there is no need to administer an ArcSDE platform because it uses client cycles for the ArcSDE functionality. However, it does require the installation and configuration of the RDBMS client software (DB2 Connect).

Proposed solution

The objective of this study was to determine how a DB2 Spatial Extender running with Linux on a zSeries server can be part of a spatial solution.

Why Linux on zSeries?

zSeries servers are among the most reliable and scalable servers for mission-critical applications in the marketplace. That is one reason why mainframe systems process more than 60% of the world's data each day. Other reasons to choose a mainframe include:

► A user has an existing zSeries system with unused capacity that could run DB2 cost-effectively

► There is an advantage of consolidating servers on a single complex

► The zSeries provides needed security, reliability and robustness

zSeries logical partition (LPAR) technology leads the industry. zSeries LPAR technology can reduce costs because it can share processors between LPARs, which makes it possible to deliver unused processing cycles in one LPAR to an LPAR that requires additional processing cycles. A single zSeries server can host many servers spread across multiple images of Linux, so that a single server can support multiple users and applications. You can add capacity to the server while the system is running; this is called Capacity Upgrade on Demand. This allows you to start with a small zSeries server and add capacity as the number of users increases.

With this scalability, zSeries can provide a lower total cost of ownership. With fewer hardware servers, you benefit from less complexity and a lower cost of administration and management. In addition, IBM offers Integrated Facilities for Linux (IFLs) at a significantly lower cost. For a given solution, hardware costs are likely to be less than one-third of the total cost over five years.

Scalability

Linux on zSeries provides a scalable platform for running applications. It effectively exploits the zSeries multiprocessor architecture so that application throughput can increase linearly when more processors are added to an image. When multiple Linux images are required, for the purposes of growing application capacity requirement or consolidating multiple applications, they can all run in a single zSeries server with excellent scalability. Capacity that is not used by one Linux image can be made available to other images, which reduces the "white space" problem commonly exhibited in distributed platforms.

High concurrency level

High numbers of concurrent users are handled with ease by Linux on the zSeries platform. When a system runs out of capacity, it is common to throttle the number of users that enter the system. This approach minimizes the probability of system crashes caused by a capacity overload. Users that try to sign on receive a "service not available" notification, which in turn degrades user satisfaction. However, Linux on zSeries provides a more stable environment for adding additional users even when processing capacity has been exhausted. To the extent that system resources are available to support the additional users, there is little or no degradation to application throughput. Not only does this increase user satisfaction, but it also reduces the frequency of system upgrades to accommodate a growing user base.

System configuration

To test the viability of the proposed solution, we put together a two-tier configuration with a zSeries server that supported four clients. The server performed the database storage and retrieval; the clients supported the ArcGIS function and provided query transactions to drive the workloads. Figure 2 shows the system configuration.
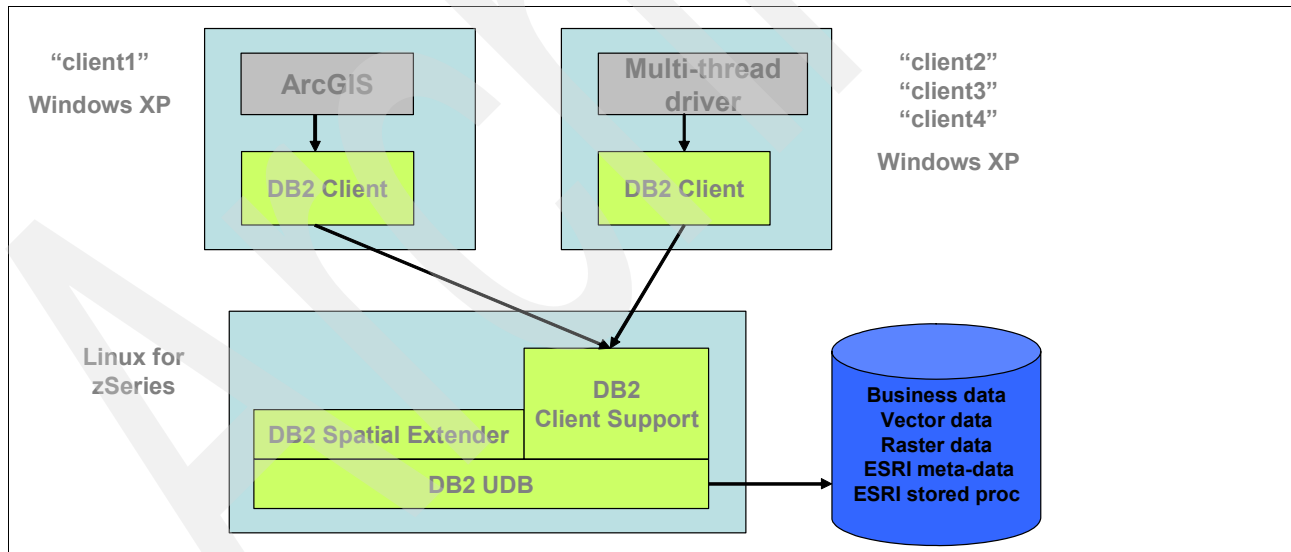

Figure 2   Configuration diagram

zSeries server

The zSeries server solution used the following hardware:

► z990 server model 2084-B16
► 3 CPs
► 4GB Central Storage, 2GB Expanded Storage
► 1 DS8000 model 2107-921
► 32GB cache, 6.4 TB total storage:

– Approximately 24 GB were for system use, such as file systems such as/, /opt, /home
– 195 GB were used for the spatial database.
– 8 FICON Express channel path IDs (CHPIDS)

The zSeries solution used the following operating system software:

► Linux SUSE SLES9 SP2, 31 bit distribution, 2.6 kernel
  – 2.6.5-7.191-s390 #1 SMP Tue Jun 28 14:58:56 UTC 2005 s390
► Direct Connected Open System Adapter (OSA)
► S/390 GNU/Linux)
► z/VM 5.1
► Performance Toolkit
► RACF

The zSeries solution used the following middleware:

► DB2 UDB Version 8.2.3 (i.e. v8.1 fixpack 10)
► DB2 Spatial Extender
► ESRI ArcSDE stored procedures for DB2 UDB

Windows client 1

Windows client 1 used the following hardware:

► IBM ThinkPad T41p Pentium® M mobile computer
► 1 GB storage
► 30 GB DASD
► 10/100 Ethernet

Windows client 1 used the following software:

► Windows XP SP 2
► ESRI ArcSDE V9.1
► ESRI ArcGIS V9.1
► ESRI ArcGIS performance measurement plug-in

Windows clients 2 and 3

Windows clients 2 and 3 used the following hardware:

► IBM Intellistation – dual 450 MHz Xeon
► 512 MB storage
► 9.1 GB SCSI DASD
► 10/100 Ethernet

Windows clients 2 and 3 used the following software:

► Windows XP SP 2
► Java™ JRE 1.5
► Custom Java classes for DB2 stress testing

Windows client 4

Windows client 4 used the following hardware:

► Intellistation with 2 2.2GHz processors
► 3 GB storage
► 5 17 GB SCSI DASD
► 10/100 Ethernet

Windows client 4 used the following software:

- ► Windows XP SP 2
- ► ESRI ArcSDE V9.1
- ► ESRI ArcGIS V9.1
- ► ESRI ArcGIS performance measurement plug-in
- ► Java JRE 1.5
- ► Custom Java classes for DB2 stress testing

Connectivity

The connectivity was provided by:

- ► GB Ethernet between zSeries and mobile computer using Direct Connected OSA
- ► 10/100 zSeries with Gigabit backbone

Testing environment

All three tests use the same sample data from the San Diego, California, area. This data is nine vector layers that represent features such as land parcels, highways, and schools, plus a raster orthoimage background layer. We used monitoring tools from VM and Linux to measure the performance of the test queries.

Monitoring tools

During each measurement interval, batch monitors captured the key performance data. For z/VM, the performance toolkit was used to gather overall system performance reports. On the Linux guest server under z/VM, we used VMSTAT, SAR, and DB2BATCH to collect performance data. The data for each interval was saved to disk, providing historical snapshots of the measurement.

Performance monitor for z/VM

z/VM allows the user to define and run complete self-contained virtual machines, called guests. We run Linux as a guest operating system in its own virtual machine. VM Performance Toolkit captured and reported the following key system performance information for each measurement:

- ► Processor utilization
- ► Direct access storage device (DASD) activity
- ► Paging Activity
- ► Workload activity: resource consumption (processor, I/O, storage)

Performance monitor for Linux on zSeries

VMSTAT was used to monitor the CPU and memory activity. SAR was utilized to gather cumulative activity parameters in the operating system, such as processor utility, I/O transfer rate, paging and network activity, and load averages.

The DB2 SNAPSHOT monitor was used to capture information about the database and any applications connected at a specific time. Taken every 30 seconds, snapshots were useful for determining the status of a database system.

Test case scenarios and queries

The basis of the testing were queries that we sent to the host in increasing numbers. The three types of queries were "single-thread map draw," "multi-thread spatial query for a large-scale region," and "multi-thread spatial query for a small-scale region."

Single-thread map draw

The single-thread map draw had the following objectives, metrics, success criteria, and methodology:

► **Objective:** Show comparable performance with other platforms
► **Key metric:** Elapsed time to draw map
► **Success criterion:** Performance within 50% of other platforms
► **Methodology**: Measurement that should take approximately 5 minutes

Description

The single-thread map draw test uses ArcGIS to draw a map based on the vector and raster data. A typical map rendered at a scale of 1:10 000 corresponds to a 5 000 foot by 5 000 foot region that contains 1,500 to 2,000 vector features. The raster data occupied approximately 1MB in 64 database rows.

This test used an ArcGIS plug-in from ESRI. The plug-in reads a control file with the coordinates of the map region to be displayed. The map coordinates are set in ArcGIS, which turns these coordinates into SQL queries that are issued to DB2 Connect on Client 1. The query is then passed to and processed by DB2 UDB on the Linux server. The spatial data is returned to ArcGIS and drawn on the screen. The plug-in measures the elapsed time and repeats the operation the specified number of times (10 in our tests).

Spatial queries

The second and third types of queries (scalability tests) had the following objectives, metrics, success criteria, and methodology:

► **Objective:** Determine the number of queries that can be done per run interval (5 minutes)
► **Key metric:** Total number of queries issued in the test window
► **Success criterion:** Linear scaling of queries when adding CPs

We ran two types of multi-thread spatial queries: one for a "large-scale "region and one for a "small-scale" region.

Multi-thread spatial query: large region

This multi-thread spatial query corresponded to the same region and data as the ArcGIS map draw test and performed similar queries without drawing a map. We ran tests to simulate:

► 60 users (approximately the capacity of a Windows server)
► 120 users
► 240 users
► 480 users

Description

A Java multi-thread application issues a sequence of spatial queries that simulate those issued by a typical map application. The spatial data is returned to the DB2 client but is not drawn. At start up, the number of threads, the "think" time between queries, and the total run time can be specified. Each thread starts with a geographic region offset from previous threads and each "draw" is for a slightly different region.

A control file with the queries to be performed is read. The test file has one query against raster data and 12 queries against vector spatial tables. This corresponds to a map with a raster image background and nine feature layers, three of which have labeling (labeled layers must be fetched twice).

The application reports the total number of queries that were issued, the number of rows that were fetched, and the total number of bytes that were fetched.

Methodology

In all cases, the total run time was set to 5 minutes and the "think" time was set to 10 seconds. Each client window was used to start 30 threads. The application was started in command windows on each of the three workstations in suite B. The number of threads was increased until the system monitoring showed that it was close to 100% utilized. This was done for one, two, and three CPs.

Multi-thread spatial query: "small" region

This multi-thread spatial query used the same data as the other tests but queried a smaller region of 1,000 feet by 1,000 feet, corresponding to a map scale of 1:2000. Our tests simulated:

- ► 120 users
- ► 240 users
- ► 360 users

Description

This test is virtually the same as the large-scale region test. Because the geographic region is smaller, the number of rows and bytes returned by each query is smaller.

Methodology

The total run time was set to 5 minutes and the "think" time was set to 10 seconds. Each client was used to start 40 threads. The application was started in each of the three workstations. The numbers of threads were increased to the predetermined number of users. This was done for one, two and three CPs. It was not possible to start enough threads on the workstations to drive three CPs to saturation.

Data tables

The vector data consists of nine tables with spatial columns that represented features in the San Diego area. The data was loaded with the ArcSDE shp2sde command. Table 1 shows the statistics of the loaded features by table.

Table 1   Description of the nine tables

Rows	Pages (4K)	MB		Feature
275	57	0.2		RD_FWY
145027	24 756	99		RD_ALL
704	80	0.3		HYD_LAKE
5239	645	2.6		PARKS
1091	111	0.4		SCHOOL
387	39	0.2		TRAN_ROUTE_SD
23585	4720	18.9		W_HYDRANT

783159	160 446	641.8		PARCEL
106	11	0		ZIPCODE

The raster image data consists of one table that contains a mosaic of raster orthoimages, with each tile covering 128x128 pixels that correspond to a 2500x2500 meter region. A Perl script on the client issued the ArcSDE sdemosaic commands to import each of the raster image (tiff) files. Table 2 shows some of the table specifications.

Table 2   Description of the raster image table

# files	4538	
total file size	112.5	GB
table rows	6 926 307	
table pages (32K)	4 099 920	
LOB bytes	128.1	GB

Results

This section shows the results of the testing scenarios.

Single-thread map draw

Table 3 shows the results of performing 10 map draw operations.

Time in Seconds	Description	ArcGIS	DB2 Server
Average 3.37 per draw	Cold	Client1	pokz
Average 3.42 per draw	Warm	Client1	pokz

Table 3   Observed times

The terms *cold* and *warm* are used to describe the states where data to satisfy the query must be read from DASD and the states where the data to satisfy the query is most likely already in memory. Typically, it takes longer to access data from DASD. The cold state typically is observed the first time a map is drawn and the warm state when the map is redrawn.

The cold times were measured by performing the map draw operations in an environment where the system cache and DB2 buffer pools did not have any cached data. The Linux system was reset (figuratively) by copying a 32GB file from one file system to another, which should have been sufficient to clear out the DASD and operating system caches. The DB2 instance was stopped and restarted to flush out any buffer pool contents.

The warm times were measured by running the test several times and taking the best measurement. The measurements of different runs typically varied by +/- 1 second.

Note that the cold time for Linux on zSeries was faster than the warm time. This is probably because of the high-performance DASD and the relatively small amount of time that was consumed to perform I/O compared with other components, such as CPU activity and network transmission. There tends to be a 5% to 10% random variation in times between runs.

These results show comparable response times to Windows measurements that other studies have produced.

Large region scalability

The query returned 64 rows of image data that corresponded to about 1 MB and 1800 rows of vector spatial data that corresponded to 400 KB. Measurements were performed with three different configurations: one, two, and three central processors. The same Linux guest was used in all of the measurements, with the only difference being the number of CPs assigned to the guest.

With a single CP, the guest was pushed to 100% busy to support a throughput of 12168 queries. With two CPs, CPU utilization was 93% and throughput was 24024 queries. For the three-CP configuration, CPU utilization achieved 87% while throughput reached 34125 queries. Figure 3 shows the scalability of the solution using small region data and indicates linear application throughput. Adding more capacity yields a linear increase in throughput up to the point that we measured.
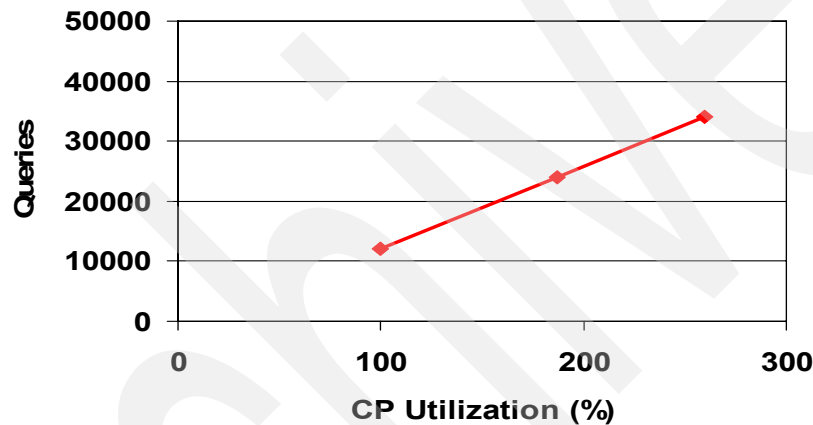

Figure 3   Scalability using large region data

Small region scalability

The same tests were performed for a smaller mapping scale, 1:2000. The query returns four rows of image data that correspond to about 65 KB and 135 rows of vector data that corresponds to 140 KB.

The linear application throughput (Table 4) was similar to that of the large-scale runs. Adding more capacity yields a linear increase in throughput up to the point we measured.

Table 4   Small region results

Number of CPs	CP utilization	Throughput in number of queries
One	96%	43 394
Two	94%	86 125
Three	88%	125 827

Figure 4 shows the scalability of our solution using small region data.
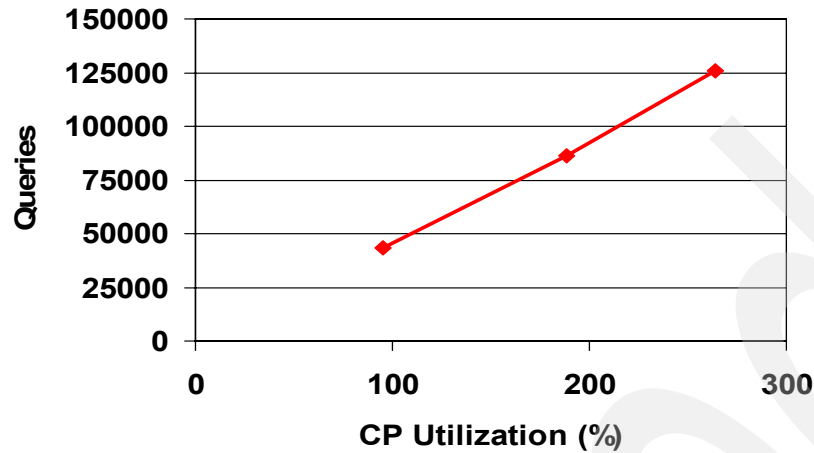

Figure 4   Scalability using small region data

Large region concurrency

Data from Figure 5 on page 12 indicates that a single CP Linux guest supports 60 concurrent users. These are not idle users. Instead, the users submitted work to the system throughout the measurement. With two CPs, the number of supported users increased to 120.

During the three CP measurement, some of the client machines (workload drivers) that were running the simulated users ran out of capacity. As a result, additional simulated users were required for the other machines to drive the Linux guest close to 100% utilization. This increased the total number of simulated users in this measurement.

Because there were more users than originally planned, each user ran a smaller number of queries. However, total system throughput, as measured by the total number of queries executed by Linux, did increase proportionally, as Figure 5 shows.
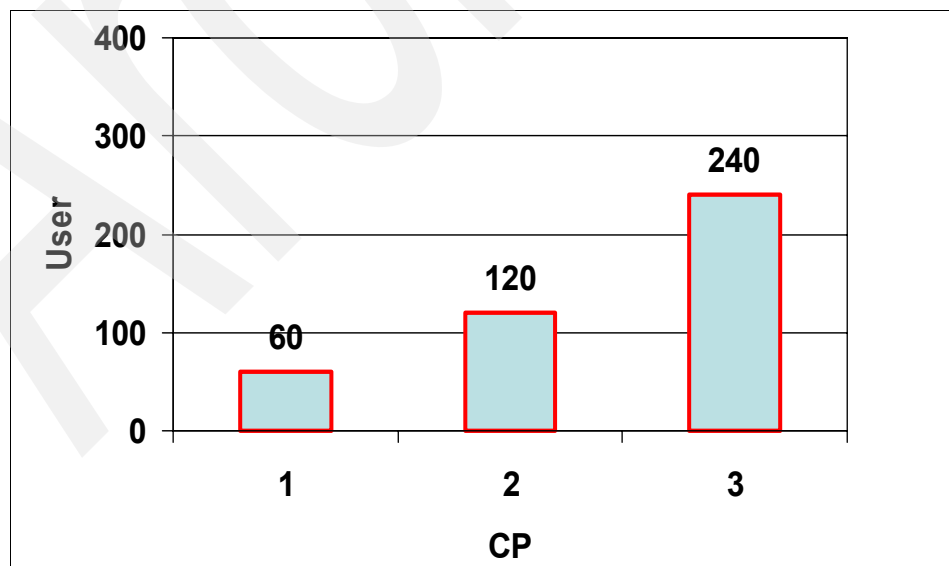

Figure 5   Large region concurrent users

Small region concurrency

As shown in Figure 6, a single CP Linux guest supports 120 concurrent busy users. With two CPs, the number of supported users increased to 240, and using three CPs allowed us to run 360 users. Total system throughput, as measured by the total number of queries executed by Linux, increased proportionally (Figure 4).
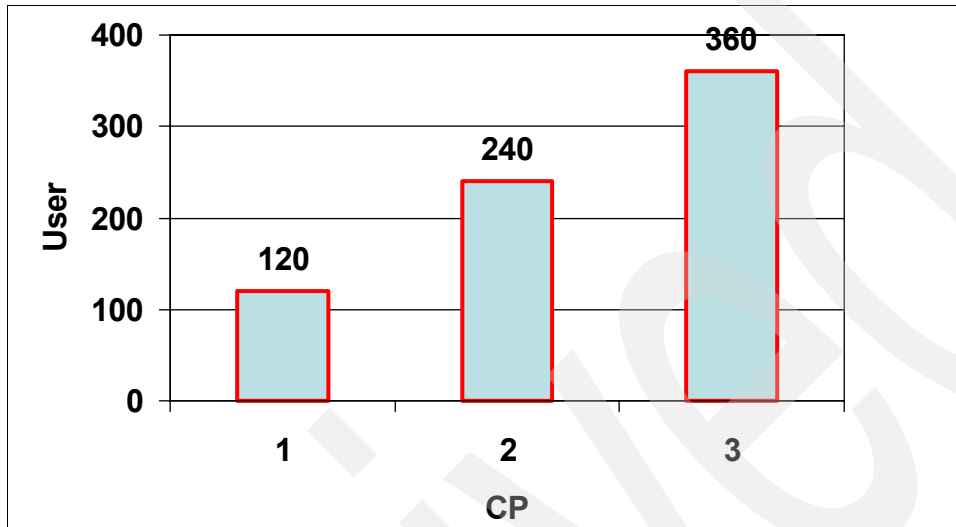

Figure 6   Small region concurrent users

Large region concurrency with processor saturation

The left portion of the chart in Figure 7 on page 13 shows a system that is running out of processor capacity. Processor utilization was 86.5% with 240 concurrent users.
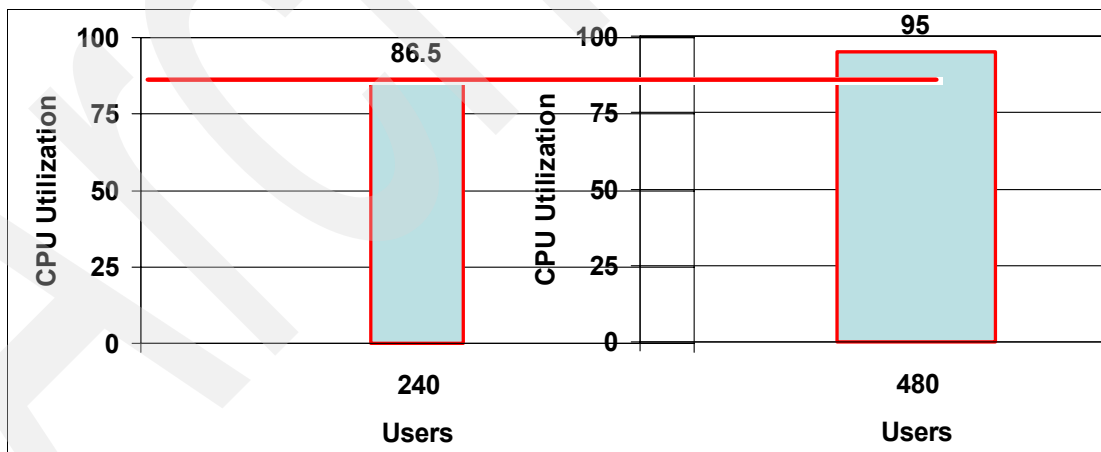

Figure 7   Large region concurrency with highly utilized processors—slightly higher throughput

Because the zSeries platform is known to perform well at high CPU utilization, we did not cap the number of users when the utilization approached full capacity. Instead, we ran a test to determine the effect of supporting additional users in a high utilization environment. The number of users was doubled to 480, driving processor utilization to 95%.

As expected, the response time of each query was longer; however, throughput did not suffer (in fact, it increased by a small margin), and users were not denied service. These results

demonstrate that Linux on zSeries can support a very high number of concurrent users, even though the system reached processor saturation at a much lower user level.

Summary

This study demonstrated that the zSeries platform with Linux and DB2 Spatial Extender is a viable database server in an enterprise GIS solution that incorporates the ESRI GIS products. In addition to demonstrating comparable response times to other platforms, it confirmed the outstanding scalability and concurrency provided by this platform, even at high utilization levels.

The team that wrote this redpaper

This Redpaper was produced by a team of specialists working at the IBM zSeries Benchmark Center, Poughkeepsie, NY.

David Adler is a senior software engineer on the DB2 Spatial Extender development team. He has 18 years of experience in the design, implementation, and performance analysis of spatial database support.

Donald Corbett is a senior software engineer in the IBM System z9 Performance and Design department at IBM in Poughkeepsie. He has a B.A degree in Computer Science from the State University of New York at New Paltz. He is the project leader and lead performance analyst for the z9 Scalability Center for Linux. He has more than 35 years of experience with product design, development, performance analysis, z/OS, z/VM, and Linux on zSeries.

Leticia Cruz is a certified DB2 Solutions Expert at the zSeries Benchmark Center in Poughkeepsie. Her areas of expertise include installation, administration, monitoring, and tuning of DB2 UDB and DB2 for z/OS.

Mike Ebbers is a project leader at the International Technical Support Organization (ITSO) in Poughkeepsie, where he writes IBM Redbooks and Redpapers on large systems applications. He has been at IBM for 32 years.

Nin Lei is a certified senior consulting IT specialist at the IBM zSeries Benchmark Center. He drives performance analysis work for all projects and provides architectural guidance to clients that want to design applications to meet high performance scalability requirements. He also advises customers about new technology exploitation issues.

Stephen McGarril is a certified IT systems management specialist at the zSeries Benchmark Center in Poughkeepsie. As a systems programmer, he supports z/VM and z/OS platforms for client engagements, to help them maximize customer application performance.

Eugene Ong is an advisory software engineer who has been with IBM since 1987. He joined the zSeries Benchmark Center in 1998. His present areas of expertise include benchmarking and proofs of concepts on the Linux for zSeries operating system and associated middleware. He is also Linux Professional Institute Certified Level One.

Andrew Schmidt is an Independent Software Vendor (ISV) project manager for zSeries network computing and Linux on zSeries. He has a B.S. degree from Boston State College (now University of Massachusetts) and an M.S. degree from Marist College. He has more than 22 years of experience in product design, product development, product planning, product documentation, and relationship management. He was the MVS/JES3 Development Representative to US SHARE for 15 years.

John Theos is a DB system programmer and database administrator at the zSeries Benchmark Center. He has 20 years of experience with DB2 in mainframe environments.

Christopher Williams is a certified Project Management Institute project manager at the zSeries Benchmark Center in Poughkeepsie. He works with clients and ISV partners to assist them with growth opportunities and solution initiatives.

Contributions

Thanks to the people from IBM S&TG, Poughkeepsie, for their contributions to this project:

Bruce Ratcliff. Master Inventor, OSA TCP/IP Development
Steve Valley, Master Inventor, OSA TCP/IP Development
Thomas Doster, OSA Performance Analysis

Thanks to the following IBM reviewers for their contributions to this document:
David Beddoe, IBM Software Group, IBM US
Boris Bialek, IBM Software Group, IBM Canada
Forsyth Alexander, ITSO, Raleigh Center

Thanks to the following people from ESRI for their review of this document:
Julio Olimpio
Dave Peters
Jeff DeWees

Web sites and resources

For information about DB2 UDB, visit:

http://www.software.ibm.com/data/db2/

For information about DB2 Spatial Extender, visit:

http://www.software.ibm.com/data/spatial/

For information about ESRI, visit:

http://www.esri.com

ESRI tiered server configuration and system design strategies white papers are available at:

http://www.esri.com/systemsint/kbase/docs/tieredhardware.pdf
http://www.esri.com/systemsint/kbase/strategies.html

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

This document created or updated on January 17, 2006.

IBM ®

Send us your comments in one of the following ways:
- ► Use the online **Contact us** review redbook form found at:
  ibm.com/redbooks
- ► Send your comments in an email to:
  redbook@us.ibm.com
- ► Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. HYJ Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400 U.S.A.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

@server®	zSeries®	FICON®
@server®	z9™	IBM®
Redbooks (logo) ™	AIX®	Redbooks™
pSeries®	BladeCenter®	RACF®
xSeries®	DB2 Connect™	S/390®
z/OS®	DB2 Universal Database™	System z9™
z/VM®	DB2®	

The following terms are trademarks of other companies:

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

Microsoft, Windows, Windows NT, Windows Server, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

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

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