IBM

Practical Migration to Linux on System z

- Overview and migration methodology
- Migration analysis
- Hands-on migration case study

Lydia Parziale
Joseph Apuzzo
Saulo Augusto M Martins da Silva
Louis Henderson
Manoj Srinivasan Pattabhiraman
Richard Sewell

Redbooks

ibm.com/redbooks

International Technical Support Organization

Practical Migration to Linux on System z

October 2009

SG24-7727-00

Note: Before using this information and the product it supports, read the information in "Notices" on page xi.

First Edition (October 2009)

This edition applies to z/VM Version 5.3 and Version 5.4, Novell SUSE Linux Enterprise Server Version 10 and Version 11, and Red Hat Enterprise Linux Version 5.3.

Contents

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 illustrate 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.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

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

AIX®
BladeCenter®
DB2 Connect™
DB2 Universal Database™
DB2®
DS6000™
DS8000®
Dynamic Infrastructure®
ECKD™
eServer™
FICON®
FlashCopy®
GDPS®
GPFS™
HiperSockets™
HyperSwap®
IBM Systems Director Active Energy Manager™
IBM®
IMS™
Informix®
Lotus®
Parallel Sysplex®
POWER®
PR/SM™
Processor Resource/Systems Manager™
RACF®
Rational®
Redbooks®
Redbooks (logo) ®
S/390®
Smarter Planet™
System Storage™
System z10™
System z9®
System z®
Tivoli®
WebSphere®
z/OS®
z/VM®
z/VSE™
z9®
zSeries®

The following terms are trademarks of other companies:

AppArmor, Novell, SUSE, the Novell logo, and the N logo are registered trademarks of Novell, Inc. in the United States and other countries.

Oracle, JD Edwards, PeopleSoft, Siebel, and TopLink are registered trademarks of Oracle Corporation and/or its affiliates.

JBoss, Red Hat, RHCE, and the Shadowman logo are trademarks or registered trademarks of Red Hat, Inc. in the U.S. and other countries.

SAP, and SAP logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries.

VMware, the VMware "boxes" logo and design are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions.

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

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

Intel, Itanium, Intel logo, Intel Inside logo, and Intel Centrino logo 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.

Preface

There are many reasons why you would want to optimize your servers through virtualization using Linux® on System z®.

- Your enterprise may have too many distributed physical servers with low utilization.
- There may be a lengthy provisioning process that delays the implementation of new applications.
- There may be limitations in data center power and floor space.
- The enterprise may experience a high Total Cost of Ownership (TCO).
- There may be difficulty in allocating processing power for a dynamic environment.

This IBM® Redbooks® publication provides a technical planning reference for IT organizations that are considering a migration to Linux on System z. The overall focus of the content in this book is to walk the reader through some of the important considerations and planning issues that you could encounter during a migration project. Within the context of a pre-existing UNIX®-based or x86 environment, we present an end-to-end view of the technical challenges and methods necessary to complete a successful migration to Linux on System z.

The team who wrote this book

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

Lydia Parziale is a Project Leader for the ITSO team in Poughkeepsie, New York, with domestic and international experience in technology management including software development, project leadership, and strategic planning. Her areas of expertise include e-business development and database management technologies. Lydia is a certified PMP and an IBM Certified IT Specialist with an MBA in Technology Management. She has been employed by IBM for 24 years in various technology areas.

Figure 1 From left: Joseph Apuzzo, Richard Sewell, Saulo Augusto M Martins da Silva, Manoj Srinivasan Pattabhiraman and Louis Henderson

Joseph Apuzzo is an Advisory Programmer from Poughkeepsie, New York who supports IBM High Performance Computing customers. He has been with IBM since 1998, but has been working with Linux since 1993 (starting with Yggdrasil Linux with kernel 0.98.1) and is Linux Professional Institute L1 certified. Joseph holds a Master's degree in Computer Science from Pace University with an undergraduate degree in Electrical Engineering Technology from SUNY Binghamton. He is a subject matter expert on the IBM GPFS™ file system, which runs on AIX®, Linux and MS Windows®. In his personal time he supports state and local Linux user groups.

Saulo Augusto M Martins da Silva is a Linux on System z I/T Specialist with IBM in Brazil. He has 12 years of experience in Linux Architecture and has specialized in Linux/390 for five years. Since joining IBM two and half years ago, he has worked with Linux on System z performance and monitoring. Saulo holds a Computer Information Systems degree from Faculdade de Ciencias Gerencias - UNA. He has RHCE certifications and LPI certification. He has co-authored other IBM Redbooks publications on subjects such as SystemTap Programming.

Louis Henderson is a System Programmer and Consultant who has worked in many locations around the world. He has more than 30 years of experience with

various mainframe technologies. He began his career with EDS World Corp. as a COBOL and Assembler programmer. Louis holds an undergraduate degree in Chinese Language and Literature and a Master's degree in Business. His current interests include working with Linux on System z and z/VM®.

Manoj Srinivasan Pattabhiraman is a Senior System z Consultant with IBM STG Lab Services, India. He holds a Master's degree in Computers from the University of Madras. He has more than nine years of experience in z/VM Application Development and Systems Programming for Linux on zSeries®. In his current role, he provides consultation and implementation services for various Linux on System z customers in the INDIA/ASEAN Region. Manoj has contributed to several IBM Redbooks publications and has presented at various international technical conferences. His areas of expertise include z/VM, Linux on System z development, Migration and Porting, Performance and Tuning, and z/VM REXX programming.

Richard Sewell is a System z Architect who works with financial services customers for IBM in Sydney, Australia. He has more than 30 years of experience both locally and internationally in operations, software support, systems programming, data center management, education and training, sales, marketing and project management working for both vendors and customers. His areas of expertise include the operating systems z/OS®, z/VM, Linux; capacity planning; training; and designing technology solutions based on IBM System z. Richard holds an MBA from Southern Cross University, Australia, and has worked for IBM since 2007.

Thanks to the following people for their contributions to this project:

Roy P. Costa
International Technical Support Organization, Poughkeepsie Center

Don Smith, Vaughn Burton, Joe Consorti, Scott Loveland
IBM USA

Mark Ebeling, Edward McCarthy, Chris Moss, Simon Williams
IBM Australia

Juari Alves D. dos Santos, Evandro Tadeu S. Vargas
IBM Brazil

Mark Brown, Chuck Davis, William Dy, Paul Ionescu, Jeff Richardson, Kurt Taylor, Robbie Williamson, the authors of *Solaris to Linux Migration: A Guide for System Administrators*, SG24-7186

Bill Reeder, author of *Determining which Solutions are the Best Fit for Linux on System z Workloads,* ZSW03037-USEN-00

Scott A. Bain, Innes Read, John J Thomas, Femina Merchant, the authors of *Advantages of a Dynamic Infrastructure®: A Closer Look at Private Cloud TCO*

William J. Clarke, Luiz C. Alves, Timothy J. Dell, Herwig Elfering, Jeffrey P. Kubala, Chun Ming Lin, Michael J. Mueller, Klaus Werner, the authors of *IBM System z10™ Design for RAS,* IBM Journal of Research and Development, Volume 53, Number 1, Paper 11, 2009

Priya Sekaran, Senior Web Developer, who provided support and valuable input about MS Windows application migration to Linux on System z

Become a published author

Join us for a two- to six-week residency program! Help write a book dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an e-mail to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Part 1

Overview and migration methodology

The decision to undertake the migration of a software application is never taken lightly. Everyone has heard of migrations that have gone awry, thereby costing organizations significant sums of money. So why do organizations still continue to migrate applications and systems?

The goal of this part of the book is to provide information that can help organizations undertake a successful platform migration to Linux on IBM System z with a minimum amount of risk.

Here, we describe the planning process and migration methodology with practical examples of migrations from both Sun Microsystems SPARC servers and x86 servers to Linux on IBM System z.

1

Migration considerations

In this chapter we discuss the reasons why migrations are undertaken and why Linux running on IBM System z provides the most secure and reliable platform for a consolidation of distributed servers. We also provide an overview of z/VM and investigate the Total Cost of Ownership (TCO) of a Linux on IBM System z environment.

1.1 Reason to migrate systems

Migrations can take many forms, from the relatively simple task of moving a server's workload to a new and more powerful server of the same architecture to the migration of a complex and heavily integrated application running across multiple servers, architectures, and possibly sites. No matter how large or small, migrations are usually undertaken for one or a combination of the following reasons:

- Reduce server sprawl through consolidation
- Reduce power and cooling requirements
- Regain data center space
- Contain or reduce costs
- Reengineer applications
- Reduce complexity
- Replace unsupported operating systems and hardware platforms
- Leverage new technology
- Accommodate new systems acquired through merger or acquisition
- Lack of available skills
- Reduce overall management costs

Two events that could provide the opportunity to consolidate distributed servers to a more cost-effective virtualization platform, such as IBM System z, are:

- Expiry of distributed server leases
- Expiry of software contracts

The driving imperative for migration is to provide a positive outcome for the business in areas of cost reduction, systems availability, and improved productivity. Another factor that is often not considered is whether it is possible to move to a platform that gives organizations a greater degree of vendor independence.

There is a general trend by organizations to resolve many of the problems associated with the growth of distributed servers by consolidating multiple physical servers to a virtualized environment on servers with a greater number of cores. Although this strategy reduces the number of physical servers, it may not substantially reduce the power and cooling requirements or provide the vendor independence that many organizations seek.

1.1.1 How green is your data center

Over the past decade, distributed servers have taken over data centers. A large financial organization reported that their total System z footprint, which runs all of their mission-critical applications, now accounts for only 0.5% of the floor space

of their two data centers. With more than 5000 distributed servers, the remaining space is take up by racks of distributed servers, distributed storage, System z storage, and networking equipment.

Figure 1-1 displays a small- to medium-sized server farm.

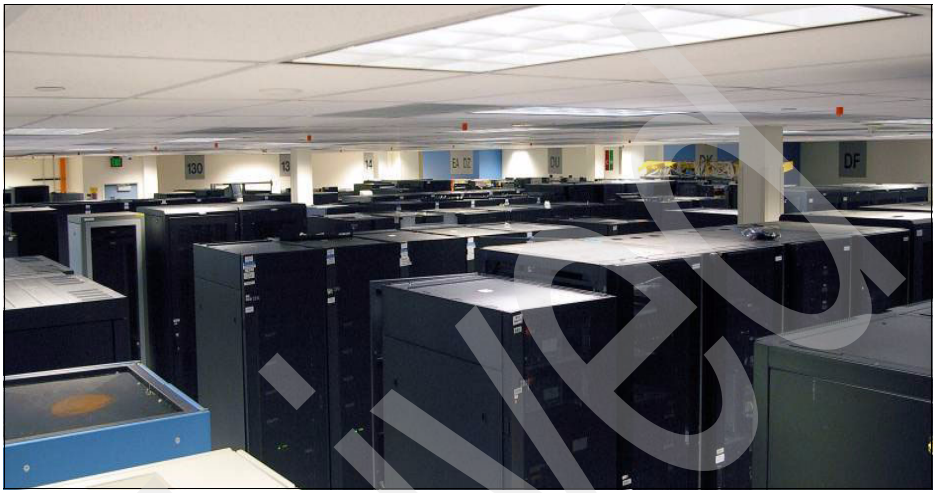

Figure 1-1 A small- to medium-sized server farm

Because all of these servers need to be networked with each other at least twice (for redundancy), and attached to one or more storage area networks (SAN) with multiple connections, the cabling between systems can become quite complex and fragile as Figure 1-2 on page 6 indicates.

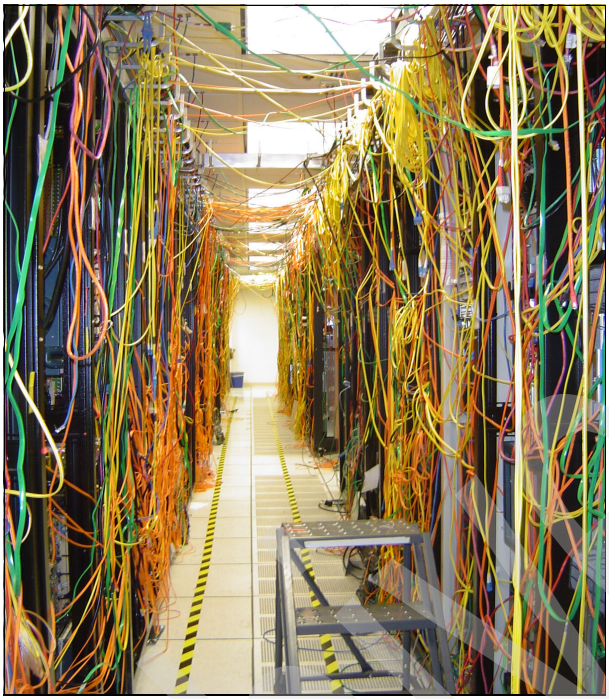

Figure 1-2 Network complexity

Large server farms present a number of major issues. Perhaps the most critical is their complexity, which requires a very skilled and experienced staff to manage. Other considerations include their power and cooling requirements, as well as the challenges of managing the life cycle of so many assets. For example, when old servers are retired are they switched off and removed from the data center? Or are they left continually powered-up with nothing running on them because no one is quite sure if any workloads are running on them?

In many instances, new data centers are being built to accommodate the ever-growing server farms. Now a critical factor is making sure enough power is available on the local electricity grid to meet the anticipated needs of the data center. Google built one of its data centers on the Columbia River in part because of the enormous power generating potential of the massive, mountain-fed river.[1]

The health of the environment is crucial to everyone. The Carbon Disclosure Project (CDP), which is a not-for-profit organization, has been established to seek information about greenhouse gas emissions from the world's largest

[1] http://www.nytimes.com/2006/06/14/technology/14search.html

companies. CDP has conducted annual surveys over the last eight years and has become the “gold standard” for carbon disclosure methodology and process.

In a survey response to CDP in 2007, a major Australian bank stated that its largest data center represents 15% of their total stationary energy expenditure. In the same survey another major Australian bank stated that its data and transaction processing centers account for almost 40% of its energy use. These are significant numbers and they provide an opportunity for organizations to rein in their distributed server sprawl and substantially reduce power and cooling requirements by consolidating to a virtualized platform like that provided by IBM System z.

1.1.2 The IBM Big Green server consolidation

In 2007, IBM announced that it would consolidate about 3900 servers onto about 30 IBM System z machines running Linux. The new server environment was projected to consume 80% less energy and occupy 85% less floor space than the original 3900 servers. Estimated savings amounted to more than $250 million over five years in energy, software, and systems support costs. Although most companies will not achieve this level of cost reduction, there are still significant savings to be realized with a consolidation of distributed servers to Linux on System z.

Note: For more details about environmentally-aware computing, see:

The Green Data Center: Steps for the Journey, REDP-4413

The Green Data Center: An Idea for Today, REDP-4523

Implementing the Poughkeepsie Green Data Center--Showcasing a Dynamic Infrastructure, REDP-4534

1.2 Benefits of migrating to Linux on System z

A significant benefit of migrating to Linux on System z is that it allows organizations to break the link between the operating system and specific hardware platforms. This means that after your applications are running on Linux, you are no longer tied to a specific hardware platform. For the first time you have control over the choice of hardware platform that will support your application.

Linux is available on a large variety of computing platforms, from set top boxes and handheld devices to the largest mainframes. Figure 1-3 illustrates the commercial IT platforms and IBM products that Linux supports.

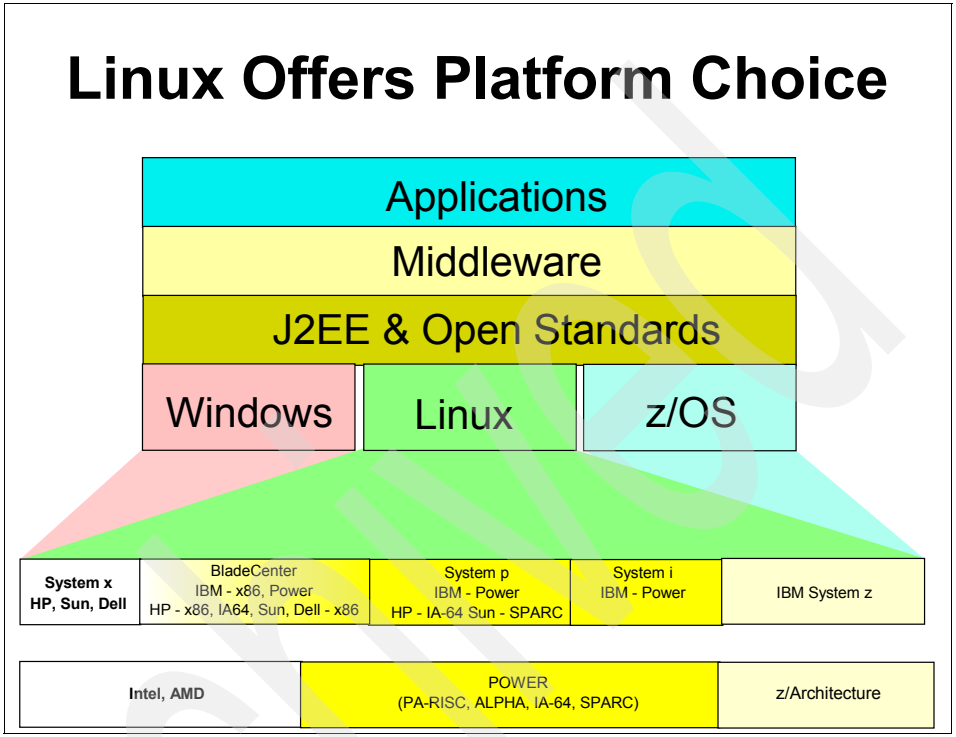

Figure 1-3 Eight-wide variety of commercial IT platforms supported by Linux

A major benefit of Linux is that it is open source; the software is unencumbered by licensing fees and its source code is freely available. There are hundreds of Linux distributions available for almost every computing platform. Two enterprise distributions[2] of Linux are:

- Red Hat - Red Hat Enterprise Linux

  http://www.redhat.com

- Novell - SUSE Linux Enterprise Server

  http://www.novell.com

Both Red Hat and Novell provide customers using Linux with a variety of support options, including 24 x 7 support with one-hour response time worldwide for

[2] A Linux distribution is a complete operating system and environment including compilers, file systems, and applications such as Apache (Web server), SAMBA (file and print), Sendmail (mail server), Tomcat (Java™ application server), MySQL (database) and many others.

customers running production systems. As well as the Linux operating system, both Novell and Red Hat offer a number of other open source products that they also fully support.

To simplify problem determination, IBM customers can contact IBM in the first instance and, if it is a new problem with Linux, IBM will work with Red Hat or Novell to resolve the problem.

Since 2000, there has been a increasing uptake of Linux by thousands of organizations worldwide. This has been primarily on x86 servers but there has been in recent years a very significant increase in Linux on IBM System z. This is reflected in a 77% increase in System z Linux MIPS[3] in 2008 as well as around 1,300 IBM System z customers now running Linux on Integrated Facility for Linux (IFL)[4].

The increased interest and uptake of Linux resulted from its rich set of features, including virtualization, security, Microsoft® Windows interoperability, development tools, a growing list of Independent Software Vendor (ISV) applications, performance and, most importantly, its multiplatform support.

This multiplatform support allows customers to run a common operating system across all computing platforms, which will mean significantly lower support costs and, in the case of Linux, no incremental licence charges. It also offers customers the flexibility of easily moving applications to the most appropriate platform.

1.3 Reasons to select Linux on System z

First announced in 1964, the IBM mainframe is the only computing system that has provided customers with a common architecture for more than 45 years. A program that was written and compiled in 1964 will still run on the latest mainframe, the IBM System z10. Today, as it has been for the last 45 years, the IBM System z is the most reliable and scalable computing platform available and it is the ideal platform for consolidating many hundreds of distributed servers.

There are two models of the IBM System z10: the Enterprise Class (EC) and the Business Class (BC). Both models share all of the same characteristics. The

[3] Millions of Instructions Per Second (MIPS) is a measure of System z performance and is analogous to GHz in the distributed server world.

[4] The Integrated Facility for Linux (IFL) is a specialty IBM system z processor that is designed to run only z/VM and Linux. It offers customers the opportunity to purchase additional capacity exclusively for Linux workloads, without increasing software charges for System z software running on general purpose processors in the server.

Enterprise Class can scale to 64 processors. The Business Class can scale to 10 processors.

IBM System z servers are the cornerstone of a dynamic architecture that helps you transform IT to take advantage of the IBM Smarter Planet™ initiative, where systems are becoming increasingly interconnected, instrumented and intelligent. System z delivers on the promise of a flexible, secure, and smart IT architecture that can be managed seamlessly to meet the requirements of today's fast-changing business climate.

By running Linux on IBM System z, businesses large and small can wrap the System z enterprise benefits and advantages around a common open platform. Developers can produce applications that deploy on cell phones, laptops and Linux virtual machines that all deliver the same flexibility and functionality. This allows the business to create solutions for the modern marketplace.

1.3.1 System z strengths

The strengths of the IBM System z are:

- Reliability
  - Redundant processors, I/O and memory.
  - Error correction and detection.
  - Remote Support Facility.
- Availability
  - Fault tolerance.
  - Automated failure detection.
  - Non-disruptive hardware and software changes.
- Virtualization
  - High-performance logical partitioning via Processor Resource/Systems Manager™ (PR/SM™).
  - Up to 60 (z10 EC) or 30 (z10 BC) logical partitions (LPAR)[5].
  - The most secure logical partitioning system available, having achieved Common Criteria EAL5 (evaluation assurance level 5) for LPAR isolation. EAL5 is the highest level of Common Criteria certification that can be achieved by commercially available hardware. Of commercial IT platforms

[5] PR/SM is standard component of all IBM System z models. It is a hypervisor that enables logical partitions (LPARs) to share system resources. PR/SM divides physical system resources, both dedicated and shared, into isolated logical partitions. Each LPAR is like an independent system running its own operating environment. It is possible to add and delete resources like processors, I/O, and memory across LPARs while they are actively in use.

only PR/SM on IBM System z10 and z9® has achieved EAL5 certification. EAL5 provides a high assurance level that logical partitions provide the same isolation as air-gapped systems.

> **Note:** For more details about Common Criteria, Evaluation Assurance Levels, Protection Profiles and a list of certified products, refer to:
>
> http://www.commoncriteriaportal.org
>
> The certified evaluation levels for System z operating systems, as of June 2009, are:
>
> - IBM z/VM 5.3: certified at EAL4+
> - IBM z/OS 1.9: certified at EAL4+
> - Red Hat Enterprise Linux Version 5.1: certified at EAL4+
> - SUSE Linux Enterprise Server 10 SP1: certified at EAL4+

  - The industry-leading virtualization hypervisor z/VM is supported on all IBM System z models.
  - Both PR/SM and z/VM employ hardware and firmware innovations that make virtualization part of the basic fabric of the IBM System z platform.
  - HiperSockets™ allows up to 16 virtual LANs, thus allowing memory-to-memory TCP/IP communication between LPARS.
- Scalability
  - System z10 EC scales to 64 application processors and up to 1.5 TB of memory.
  - System z10 BC scales to 10 application processors and up to 248 GB of memory.
- Security
  - Clear key integrated cryptographic functions provide high speed cryptography for data in memory.
    - Supports DES, TDES, Secure Hash Algorithms (SHA) for up to 512 bits, Advanced Encryption Standards (AES) for up to 256 bits and Pseudo Random Number Generation (PRNG).
  - Optional cryptography accelerators provide improved performance for specialized functions.
    - Can be configured as a secure key coprocessor or for Secure Sockets Layer (SSL) acceleration.
    - Certified at FIPS 140-2 level 4.

- Just-in-time deployment of resources
  - On/Off Capacity on Demand provides temporary processing capacity to meet short-term requirements or for testing new applications.
  - Capacity Backup Upgrade (CBU) provides temporary access to dormant processing units. This is intended to replace capacity lost due to a disaster. CBU gives customers the peace of mind knowing they can access additional capacity in the event of a disaster recovery situation without having to purchase additional capacity. Typically this would allow customers to sign up for CBU on a IBM System z10 at another site and use this capacity for up to 5 disaster recovery tests or for an extended period of time in the event of a declared disaster at the customer site.
- Power and cooling savings
  - With its low power and cooling requirements, the IBM System z10 is an ideal platform for the consolidation of distributed servers.
  - Consolidating hundreds of distributed servers to IBM System z10 reduces the power and cooling load in the data center.
  - The IBM Systems Director Active Energy Manager™ (AEM) for Linux on System z provides a single view of actual energy usage across heterogeneous IBM platforms within a data center. AEM allows tracking of trends, which provides accurate data to help properly estimate power inputs and more accurately plan data center consolidation or modification projects.

1.3.2 Value of Linux on System z

There is great value in migrating to Linux on System z:

- Reduced Total Cost of Ownership (TCO)
  - Environmental savings - single footprint versus hundreds of servers
  - Consolidation savings - less storage, fewer servers, fewer software licences, less server management/support
- Improved service levels
  - Systems management (single point of control)
  - Reliability, availability, security of System z
  - High performance integration with z/OS, z/VSE™, z/TPF
- Speed to Market
  - Capacity-on-demand capability of System z
  - Virtual Server provisioning in minutes instead of days or weeks

1.3.3 Choosing workloads to migrate to IBM System z

IBM System z offers a solid platform for the consolidation of distributed servers, but how do you choose which workloads to migrate?

For the first migration, we recommend that you choose an application that is not overly complex and that has a Linux on System z platform that is supported by the ISV or (in the case of a home-grown application), that has the source code available. Many ISVs support Linux on System z and these include but are not limited to IBM, Oracle, SAP, IBI, Red Hat and Open Source. Applications that require close proximity to corporate data stored on the mainframe are also ideal candidates, as are applications that have high I/O rates because I/O workloads are offloaded from the IFL by the z10 Service Assist Processor (SAP)[6].

The IBM System z10 has a very fast processor with a clock speed of 3.5 GHz (z10 BC) or 4.4 GHz (z 10 EC). Because the System z is designed to concurrently run disparate workloads, it is important to remember that some workloads may not be ideal candidates for consolidation on Linux on System z. Typically the workloads to avoid are those that require their own dedicated physical processors, are designed to run at very high sustained CPU utilization rates, or have very high memory needs. Examples include animation rendering, seismic analysis, Monte Carlo simulation, weather forecasting and so forth.

Chapter 8, “Application analysis” on page 85 provides an in-depth analysis of the process of determining the most appropriate application to migrate to a Linux on System z environment.

1.4 z/VM virtualization for Linux on IBM System z

Virtualization of computing systems, contrary to popular belief, is not a recent phenomenon. Virtualization has been in existence since 1967 when IBM introduced Control Program (CP)-67. Over the past 40-plus years, IBM has extended and refined these initial developments into today’s z/VM, the virtualization hypervisor for the IBM System z. z/VM supports Linux, z/OS, z/VM, z/TPF, and z/VSE, enabling a mixed workload operating environment.

Figure 1-4 on page 14 illustrates the developmental history of z/VM.

[6] The Service Assist Processor (SAP) runs I/O microcode.

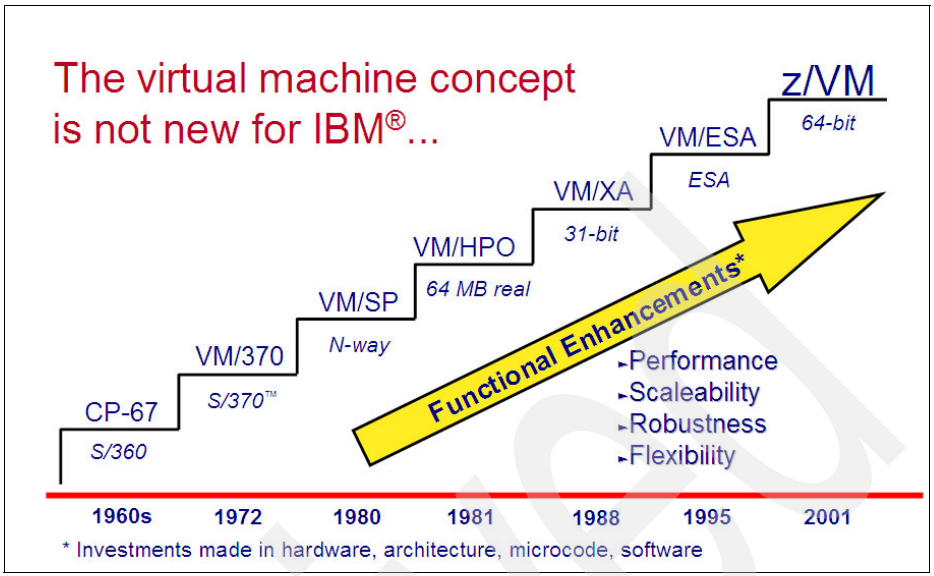

Figure 1-4 Family history of z/VM

The role of a virtualization hypervisor is to share physical hardware resources so that a single physical server can act as many virtual machines or servers. The IBM System z acts as a base to consolidate multiple distributed servers on a single hardware platform.

z/VM shares hardware resources with many virtual machines, each believing that it is the only server using the hardware. Because some servers are more important than others, z/VM also manages the scheduling[7] of virtual machines according to its defined priority.

By virtualizing the IBM System z hardware with z/VM, it is possible to run the system at a very high utilization rate to obtain the most throughput and maximize the benefit of consolidating distributed servers.

What makes z/VM different from any other hypervisor is that it has been designed to work closely with the System z hardware to minimize the overhead usually associated with virtualization. It uses a combination of software and hardware mechanisms to insure full system integrity. This allows the z/VM Control Program to isolate and protect virtual machines from each other and also operate without interference or harm, intentional or not, from guest virtual

[7] Scheduling is the process whereby CP allocates and gives control of physical resources such as CPU cycles, memory and I/O to a virtual machine. This is managed by z/VM based on the relative priority of a virtual machine.

machines. More technical information about z/VM is provided in Chapter 5, “z/VM and Linux on System z technical concepts” on page 49.

1.4.1 Leveraging the power of z/OS for distributed platforms

A major benefit of consolidating distributed servers to Linux running under z/VM is closer proximity to data. For many organizations, z/OS is the data server for corporate databases such as DB2® and IMS™, and having distributed applications in the same System z footprint can eliminate many of the physical network bottlenecks and accelerate application processing.

In a multitier environment such as SAP, there is significant network traffic between the application servers running on distributed servers and the database residing on z/OS. By consolidating, where appropriate, these distributed application servers on Linux, the physical network between the application servers and the data can be replaced by HiperSockets, which provide a TCP/IP environment where data is moved at memory transfer speeds. This eliminates many bottlenecks.

1.4.2 What is Linux on System z

Linux on IBM System z is not a special version of Linux. The only difference between the System z version of Linux and other platform versions of Linux are portions of the compilers, runtime libraries, and a hardware-dependent component of the Linux kernel. All of these changes are necessary to allow Linux to run on the base platform, and this is only about 1% of the total Linux code.

This requirement to have platform-dependent code is also the case for other platforms that Linux supports such as x86, Power and Intel® Itanium® architecture (formerly called IA-64). Figure 1-5 on page 16 illustrates the various components of Linux on System z.

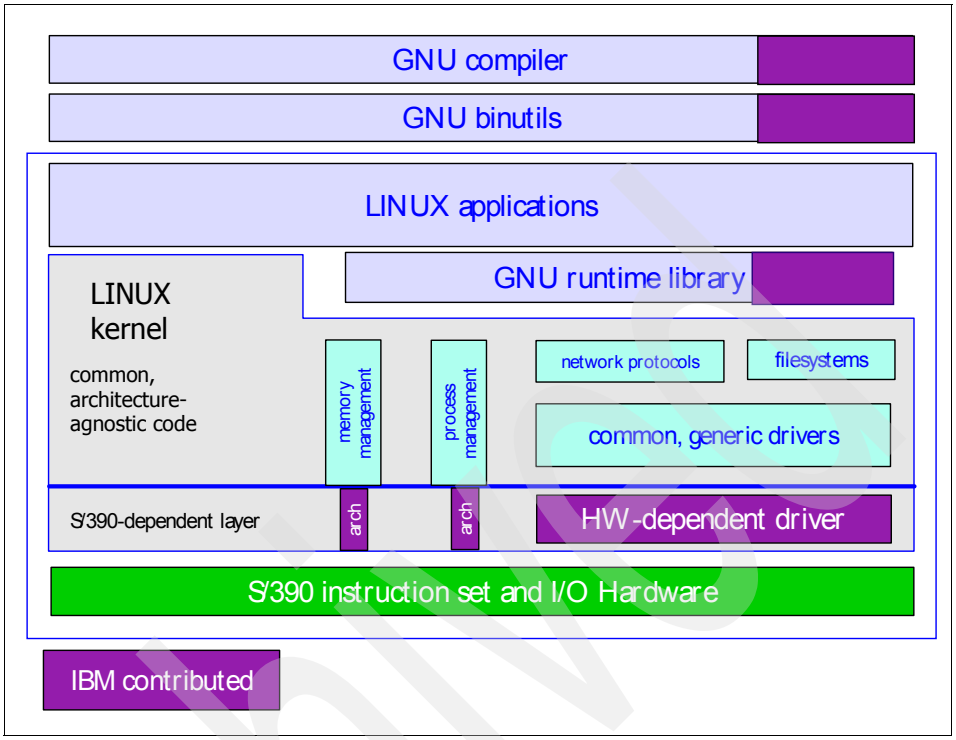

Figure 1-5 Linux on IBM System z illustrated

Although the recommended approach is to run Linux under z/VM, Linux can run in an IBM System z logical partition (LPAR) without virtualization. The Linux environment on IBM System z, whether in an LPAR or under z/VM, is pure ASCII. There is no need to convert data to EBCDIC format when moving to Linux on System z.

Linux on System z can execute on either a Central Processor (CP)[8] or an Integrated Facility for Linux (IFL). The major benefit of an IFL for customers running z/OS is that an IFL has no impact on IBM software pricing for z/OS. z/VM can also run on an IFL or CP.

In a Linux on System z environment, the recommended approach is to run z/VM and Linux on IFLs. z/VM should only run on a CP if z/OS is to be run under z/VM.

[8] The System z Central Processor (CP) is a general purpose processor that can run z/OS, z/VM, Linux, z/VSE, and z/TPF. On IBM System z10, CPs are available in a large number of capacity ratings to ensure that customers pay only for what they need.

1.5 Sizing workloads for migration

Previously we discussed the benefits of the System z platform in its role as an enterprise Linux server. However, one of the biggest challenges is to determine the IBM System z resources required to accommodate the distributed workload.

The first step is to determine the expected consolidation ratio for a given workload type. This allows us to answer the question “What is the theoretical maximum number of servers that can be consolidated?”

The answer to this question is a function of three factors:

- Processor speed (or speeds) of the servers to be consolidated
- CPU utilization of these servers
- Workload characteristics of applications to be consolidated

Although this may set limits on the upper boundaries for virtualization, the efficiency of the target platform and platform hypervisor may reduce consolidation ratios. In practice, service levels are often the determining factor.

As a first step in determining the System z resources required to consolidate distributed workloads, IBM offers a study using the IBM Rehosting Applications from Competitive Environments (RACE) tool.[9]

The inputs for the RACE tool are:

- Distributed server details
  - Vendor, model, CPU speed, memory capacity
  - Average peak CPU utilization
  - Workload type (that is, database management system, Java, I/O bound, compute bound, and so on)
  - Costs
    - Software license and maintenance costs
    - Hardware purchase and maintenance costs
    - Staff costs
    - And so on

The outputs from the tool are:

- Number of IFLs required to support distributed workload
- Amount of memory required
- Total Cost of Ownership (TCO) analysis of the various configuration options (based on cost inputs in the model)

[9] To arrange a RACE study, contact your IBM representative.

1.6 Total Cost of Ownership analysis

The business case to support a migration to Linux on System z is invariably focused on cost savings brought about by server consolidation to System z and an overall simplification of the distributed environment.

IBM recently performed a TCO benchmark of a sample banking application to determine the consolidation ratio using the IBM WebSphere® Application Server and DB2 running on Linux.

Note: More details of the TCO study are available at:

http://www-01.ibm.com/software/se/collaboration/pdf/CloudTCOWhitepaper_04212009.pdf

The existing server environment was a 4-way (single core) IBM x366 server using 3.66 GHz Intel processors and 1 GB of memory. Average utilization was 5%, with throughput of 4.5 transactions per second (tps) and average response time of 40 milliseconds.

A VM image of this workload was created and placed on an 8-core IBM x3950 server (4 x 3.5 GHz dual core processors) with 64 GB of total physical memory running VMware as a hypervisor. Multiple running instances of this VM image were added to the server until it could no longer handle any additional throughput.

The same test was then applied to a single frame IBM z10 EC (8 IFL cores at 4.4 GHz) running z/VM as a hypervisor. The results of the benchmark in terms of response time and throughput appear in the following figures.

Figure 1-6 on page 19 displays the response time comparison.

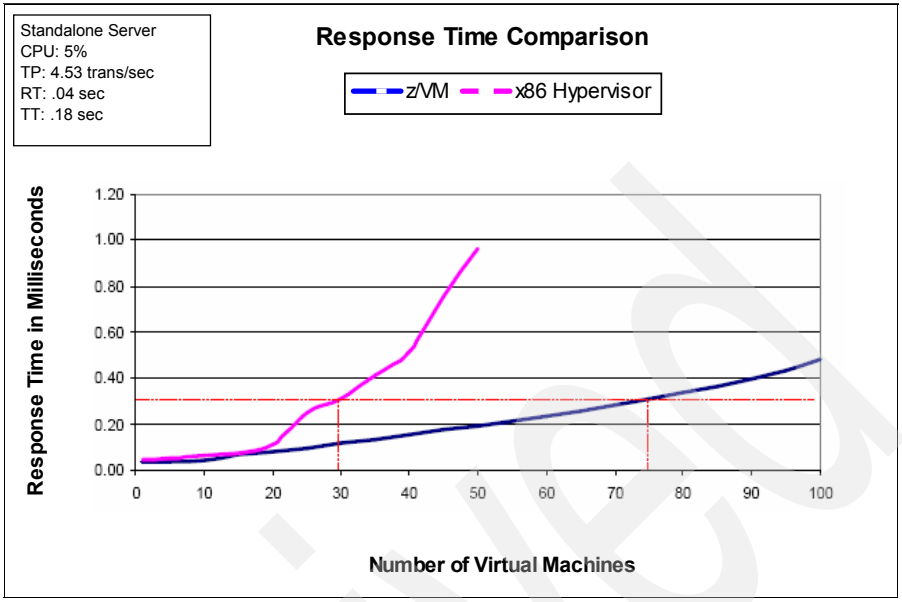

Figure 1-6 Response Time Comparison

Figure 1-7 displays the throughput comparison.

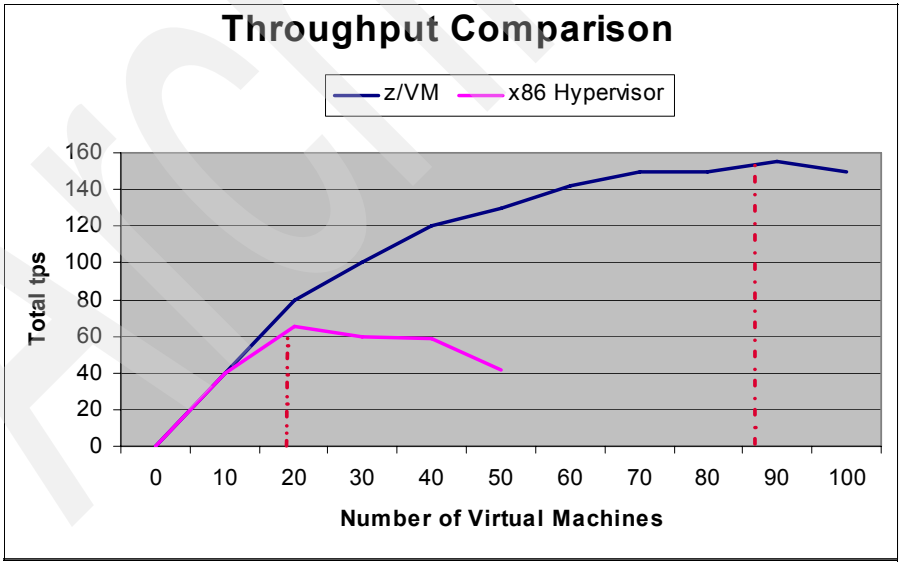

Figure 1-7 Throughput Comparison

The maximum number of VM images that could be supported with acceptable response time was approximately 30. The z/VM image showed the maximum number of images of about 75.

The number of images for z/VM was increased to 85 versus less than 20 VM images for the x86-based hypervisor when taking maximum throughput into account.

The TCO analysis was calculated over 5 years for 100 Linux images using four different platforms to see which one delivered the lowest cost per image or workload:

- Buy stand-alone x86 servers running one image on each.
- Rent an Amazon EC2 instance running one image each.
- Buy large x86 servers and provision virtual servers using VMware.
- Upgrade an existing z10 EC and provision virtual servers using z/VM.

TCO components included hardware, software, maintenance, facilities (power/cooling) and administration, and assumed 24 x 7 operation. Administrative costs were derived from the IBM RACEv tool and other internal studies. The results of this comparison are listed in Figure 1-8 on page 21.

Detailed Cost Breakdown for Linux Workloads (5 Year TCO)

	Buy Another Server	Rent a Virtual Server	Provision Your Own (x86 hypervisor)	Provision Your Own (z/Linux)
Runtime Platform	100 IBM x3250 with 4 cores each	100 Amazon Extra Large EC2 instances	Eight IBM x3950 with 8 cores each	12 IFLs added to existing z10 EC
Hardware Costs ▪ Server ▪ Storage ▪ Networking	$5,000,000	$2,880,000	$1,390,000	$3,000,000
Software Costs ▪ OS (Linux) ▪ Hypervisor x86, Linux on System z) ▪ App Server (IBM WAS) ▪ Database (IBM DB2)	$22,660,000	$22,010,000	$3,987,000	$2,644,000
Facilities & Admin ▪ Power ▪ Floor space ▪ Maintenance ▪ Sys Admin	$11,020,000	$4,020,000 (admin only)	$3,395,0000	$1,240,000
Total Cost	$38,680,000	$28,910,000	$8,772,000	$6,884,000
Number of workloads/images supported	100	100	100	100
Total cost per image	**$386,800**	**$289,100**	**$87,720**	**$68,840**

Figure 1-8 5-year TCO comparison

Figure 1-8 demonstrates that virtualization on your own equipment makes a compelling statement compared to renting virtual server instances. The IBM System z provides significant savings due to its much higher virtualization factor and its ability to continuously run at very high CPU utilization rates. The System z10 EC also has significant headroom because it can scale to 64 processors, which all can be IFLs. Depending on the workloads, this could potentially support up to 900 servers.

The cost per image of a new IBM System z10 Business Class (BC) was also investigated for customers who do not currently use IBM System z hardware. This system is slightly less expensive than x86-based servers running popular VM imaging software. However, it also provides the added features of the IBM System z platform.

z/VM also offers a number of benefits that are not discussed here but are discussed in the white paper. These were in the area of service management and automated self-service provisioning for a true private cloud environment.

2

Stakeholder considerations

This chapter provides useful information regarding the technical and non-technical stakeholders of the migration project. A stakeholder is anyone who is affected by the activities of the project. Conversely, it could also be stated that a stakeholder is anyone who affects the migration project. A stakeholder analysis is essential to facilitate the communication and cooperation between the project participants and to assure successful outcomes, whether the outcomes are individual milestones or the entire completed project. Make sure stakeholders are involved during the planning stages of the migration project, rather than simply when they are needed to perform tasks for you in the execution stages of migration project.

This chapter defines various types of stakeholders and discusses their roles and responsibilities. Not every organization will use the same titles for stakeholders as those listed here, but the titles that you use should match the functions described.

This list of stakeholders is meant to be comprehensive, so not all the stakeholders discussed here will necessarily be involved in your project. As well, there are often versatile and skilled people who participate in or are responsible for multiple roles over several functional areas.The size of your organization and the complexity of the effort will determine the number and breadth of stakeholders involved in your migration project.

2.1 Stakeholder definitions

This section categorizes stakeholders as comparatively non-technical business stakeholders, or as more technically-oriented information technology stakeholders.

2.1.1 Business stakeholders

- Business owners or business managers

  These stakeholders lead business lines such as marketing, sales, finance, and so on. They are concerned with the business and often view information technology as a tool to accomplish business tasks efficiently and effectively. These stakeholders may have a staff member reporting on technical issues, including migration proposals. Migration proposals must be evaluated by the technology stakeholders. Conversely, proposals for migration may originate with the technology stakeholders, who must provide sufficient justification to the business owner. Migration justifications are discussed in Chapter 1, “Migration considerations” on page 3.

  Large and complex consolidation projects will require participation and buy-in from several business owners and business lines. The business owners and IT management must be closely aligned and cooperate openly to achieve a successful migration.

- Business managers and supervisors

  These stakeholders are concerned with the workflow within their departments. They understand the importance of the application and how their employees utilize it. They select users who are the most qualified and motivated to participate in the migration project.

- Users

  These stakeholders are the end customers. They use the application or consume the services provided by the application. They are the ones who can perform testing and assure that the application is working the same after the successful implementation of the migrated system. In a migration without enhancements, users should not see any changes. Availability and response times should meet the Service Level Objectivesagreed to by management and communicated to the users. Their perspective and input to the conversion project is valuable. Their approval and satisfaction should be criteria for the success of the migration project.

2.1.2 Information Technology stakeholders

- IT management

  The highest level of IT management is usually the CIO. In some companies, the highest level of IT management may be a director or a manager. This stakeholder's role is to provide vision and leadership for information technology initiatives. The main concerns are to support business operations and services as well as to improve cost effectiveness, improve service quality, and develop new business process services. These stakeholders should clearly understand the benefits and risks of the migration project.

- Project manager

  This stakeholder has the responsibility of managing the plans, interdependencies, schedule, budget, and required personnel for the migration effort.

  Other responsibilities include defining and obtaining agreement on the approach. The project manager tracks and reports to all key stakeholders on progress against plans, escalating any issues or risks where appropriate.

- IT managers and supervisors

  Some stakeholders will be managers or supervisors of mainframe system administrators and system programmers. Managers at this level will have various types of influence on the migration project. Some projects may be originated and championed by these stakeholders. They usually have a high level of technical competence and understanding of the technologies that will be utilized in the migration project. These stakeholders should be intimately aware of the staffing and training considerations of the migration project. They should work closely with their staff to assess current skills and map out a training plan to acquire the required hardware- and software-related skills.

- Mainframe system administrator, system programmer

  The system programmer is responsible for setting up the logical partitions (LPARs) for z/VM to run in. (Or, a single instance of Linux could run in an LPAR.) The tools used for this task are Hardware Configuration Management (HCM) and the Hardware Configuration Definition (HCD). HCM is the graphical user interface to HCD. The physical hardware environment is defined with HCD.

  The mainframe system administratoris responsible for setting up hardware definitions. The hardware components defined are CHPIDS (channels), control units, and devices. A *channel* is a generic term for external I/O communication paths to Open Systems Adapter (OSA) for Ethernet networks, FICON® or Fibre Channel Protocol (FCP) for attached disk, printers, tapes, and consoles. System programmers install and maintain z/VM including defining user directories and resources for CMS users and Linux guests.

They also configure the network connections, virtual switches, and installation of additional products and services such as the IBM Performance Toolkit for VM.

- UNIX, Linux, and Windows system administrators

  Linux administrators may assist in installing Linux on System z, or take over administration tasks after the Linux guest has been installed. These stakeholders work closely with the system programmers when major configuration changes or additions are made (such as increasing the memory, disk space, or CPU). All other Linux administration duties will be the same as on other platforms, such as Linux on Intel.

  Various other Windows and UNIX administrators will be involved in the migration project. This is partially dependent upon where the source system is hosted (that is, the platform where the source application resides). The administrator of the source system will be heavily involved because that is the application that is being migrated.

  Other services such as DNS, mail servers, security, and so on will be running on UNIX or Window servers. These and other services will usually be required by the application that is being migrated. The administrators of these services will be required to make adjustments for the migrated application.

- Network engineers

  These stakeholders design, install, and maintain data communication equipment such as routers, switches, local area networks (LANS), wide area networks (WANs), and other network appliances. They monitor the network for performance and errors. During migration, network engineers help to design the new network and deploy any changes to the existing network.

  Network engineers must be familiar with the communications components that are unique to Linux on System z. For more information about IBM System z networking, refer to Chapter 6, “Network analysis” on page 57. Other network concepts and tools will be the same for these stakeholders.

- Database administrators

  The tasks performed by these stakeholders can be separated into two or more different but related job functions such as database analyst, database administrator, and system administrator. These stakeholders are responsible for installing and maintaining the database management system (DBMS) code base. They design and implement the corporate databases. Further, they maintain the performance and integrity of the databases and work closely with the application development group to ensure the application is running efficiently.

  If the migration does not include the conversion to another DBMS, these skills are usually readily transferable to Linux on System z because of the similarities of administration, including GUI interfaces and tooling.

- Security administrators

  The functional area of security has become more visible and critical as company assets become more exposed to the Internet and available on mobile and wireless devices. The security officer and architect stakeholders are responsible for data protection, including the authentication and authorization of users who access company applications. The target application must adhere to existent security policies or demonstrate heightened security methods and standards. For more details about Linux on System z security, refer to Chapter 11, “Security analysis” on page 121.

- Application architects and developers

  Applications developed in-house require porting and testing on the target Linux system. The effort involved can vary greatly, depending on what language the application is written in and how hardware-dependent the code is. Open source and commercial tools are available to help with tasks such as assessing the portability of your applications. IBM Global Services, as part of its migration services offerings, uses tools developed in cooperation with IBM Research to help with code assessment and conversion. The application architect and developers are the stakeholders who are responsible for this porting effort. Refer to Chapter 8, “Application analysis” on page 85 for more information about the issues that need to be considered.

- Operators

  These stakeholders control and monitor the application, the operating system, and the physical environment. They monitor consoles, logs, alerts, and graphical displays. They create problem tickets, notify support teams, and escalate issues to management. They require training on any new tools and procedures that result from the migration project.

- Service Desk staff

  These stakeholders are on the front line of support to the customer. They are usually the first ones to get a call when there is a real or perceived problem with the application. They need to be the first staff trained on the new environment, and should be heavily involved in the migration testing so they can provide meaningful support after the migration.

- Users

  Perhaps the most important stakeholders involved in a migration are those who will use the application every day. They need to be involved from the beginning because the success of the project will depend in large measure on how easy the system is for them to use. Ideally, it should have the same “look and feel” to which they are accustomed. However, in many cases a migration is often an opportunity for firms to improve the application, which often results in additional functions and procedures that they need to learn.

- Vendors

  Various vendors will be involved in the migration project; there may be vendors for hardware, middleware, and applications. Vendors have many resources that you can utilize, and they are often ready to help if you make your needs known. Vendors can respond quickly and are often the most cost-effective source of information and solutions.

  For ISV applications that you are targeting for migration, you need to determine if the vendors provide compatible versions that support the distribution of Linux that you plan to use. Many ISV applications have other third-party dependencies. Vendors should be able to help you map out all ISV dependencies, including middleware. Most leading middleware products are available on Linux, and there are often open source alternatives.

- Contractors

  Specialists can be called on to assist with transient needs. They may provide skills that your staff does not yet have, or skills that will not be needed after the migration project is completed. Contractors can be used to enhance the skills of your staff as they simultaneously perform tasks on the migration project. Make sure that skills transfer takes place for persistent, recurring tasks.

2.2 Assembling the stakeholders

Holding a meeting of stakeholders (or representatives of larger groups of stakeholders) is a useful way to set expectations and to address other planning considerations. Such a meeting will help to uncover whether additional administrator, manager, or user skill enhancements are needed. The participants will also be the people to whom status and milestone results are reported. Some of these people may have never met, and a cohesive, efficient, and successful project requires personal relationships.

To make sure that all interests are taken into account, it is useful to request a meeting of the key people who requested the migration and who are affected by it. Subsets of stakeholders with related tasks and responsibilities should also meet to enhance communications and encourage teamwork.

Communicating the change

Stakeholder meetings can be an efficient way to open communication channels. Effective communications plans will help to “flatten out” the negative aspects of the acceptance curve, as illustrated in Figure 2-1 on page 29.

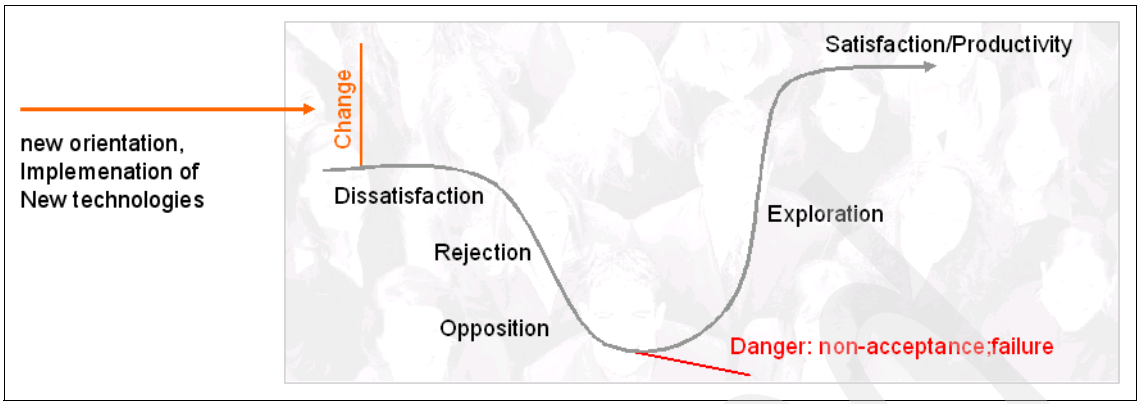

Figure 2-1 Acceptance of new technologies

A communications plan, coupled with proper training on the new system, should minimize the number of users who fall into rejection or opposition mode. It will encourage users to start out with acceptance instead of dissatisfaction as the initial response, and lead to a quick transition into exploration and productive use.

These issues are even more important with regard to the IT support team. A strategic decision to switch an operating system or platform can inadvertently create an impression of disapproval of the work the team has have done so far, and might cause staff to think their current skills are being devalued. It can be challenging to convince administrators of Solaris systems that migrating to Linux is a worthwhile strategy, unless you can also convince them that the organization is prepared to make an investment in upgrading their skills as well.

You should be able to articulate the objectives for your Linux migration and relate them to your key business drivers. Whether you are trying to gain efficiencies by reducing costs, increasing your flexibility, improving your ability to support and roll out new application workloads, or some other key business drivers, be sure to set up objectives that line up with these. Even the smallest of migrations should be able to do this, and it will help guide your planning.

Defining metrics (increased performance, more uptime, open standards, enterprise qualities) early in the project will help your team stay focused. Be sure that you will have a means of tracking the metrics. Getting stakeholder agreement on your metrics early in the project will help ensure support ranging from executives to users.

Often, the migration to Linux will be accompanied by other objectives. For instance, some customers upgrade their database at the same time to get the latest features and performance enhancements and to obtain support that lines

up well with the latest distributions of Linux. As with any project, the scope must be well defined to prevent project overrun, but it is also important that you have a means to manage additions to the plan as business needs dictate.

Because cost is often a key motivator for migrating to Linux, give careful consideration to identifying where cost reduction is targeted. Identify metrics for defining return on investment prior to beginning migration activities, and identify metrics for other success criteria.

3

Migration methodology

In the field of information technology, the term *migration* refers to the process of moving from the use of one operating environment to another operating environment. In many cases, the move to a new platform involves various organizational and strategic changes. This chapter provides you with information regarding the approaches involved in planning your migration.

3.1 Migration approach

After the business value and need for moving to Linux on System z has been accepted by the various organizational stakeholders, it is time for the actual migration planning.

In a typical migration scenario an entire environment must be identified, rationalized, and tested for compatibility with the new host operating environment. Figure 3-1 on page 33 illustrates an approach to planning.

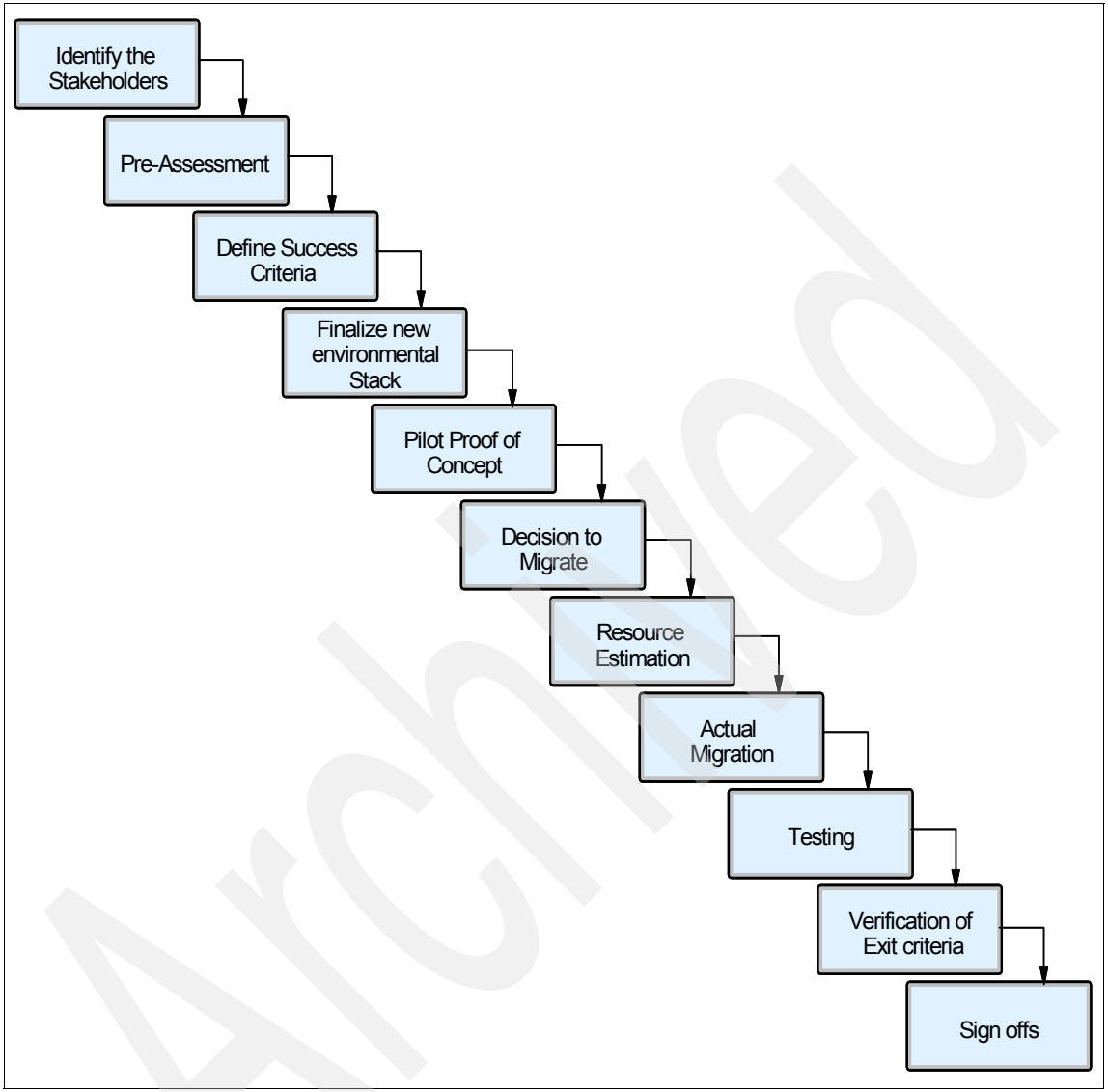

Figure 3-1 Process diagram of planning approach

3.1.1 Identify the stakeholders

As previously discussed, the first phase involves identifying your stakeholders. In turn, the stakeholders identify the business and operational requirements that impact the migration process. All stakeholders within the company must be

consulted to ensure that their requirements are factored into the migration planning.

- Business owners define the business and operational success criteria.
- System administrators provide information about the application requirements, database requirements, and available network bandwidth, as well as CPU load and allowable downtime.
- Security and compliance teams define compliance requirements for the entire migration effort.

3.1.2 Pre-assessment

During the pre-assessment phase, a high level analysis and initial feasibility study of the application architecture, source code dependencies, database compatibility, and build environment is performed. This task defines an overall scope for the migration to the target operating system. The applications running on current servers are assessed to determine whether they are available and certified to run on Linux on System z, and an evaluation of the risks related to migration is performed. This helps to identify major risk areas at the earliest stage.

Additionally, a careful analysis of present and anticipated business needs should also be carried out and weighed against the pros and cons inherent in each option of migration.The outcome of this phase is a recommended migration approach, as well as a high level risk assessment and analysis report identifying potential issues that can occur during the migration.

3.1.3 Define success criteria

In this phase, a consensus must be reached by all stakeholders regarding the porting project success criteria. Migration success may mean, for example, passing a percentage of system tests on the Linux on System z platform or passing a level of performance criteria set out by the system performance group.

Regardless of how the project success is defined, all stakeholders must understand and agree on the criteria before the porting effort starts. Any changes to the criteria during the course of the porting cycle must be communicated to all stakeholders and approved before replacing the existing criteria.

3.1.4 Finalize the new environmental stack

Usually a migration involves moving custom-built or third-party applications to another operating environment. This task involves careful analysis of different

tiers of the hierarchy based on a best fit between the database, the application requirements, and other environmental attributes.

We recommend that you perform a one-to-one mapping of the various middleware, compilers, third party tools, and their respective build parameters. If any of the one-to-one mapping for any parameters is missing, then you we need to list other parameters available in the tool that would provide the same functionality or feature. Chapter 4, “Migration Planning checklist” on page 39, provides examples of forms that can be used to help document your software and hardware requirements.

During this phase, most of the technical incompatibilities and differences in the environmental options are identified.

Custom-built applications

If custom-built applications are written in one or more programming languages, several tools may need to be validated on the target environment, such as compilers, the source code management system, the build environment, and potentially third party add-on tools.

Additionally, an in-depth analysis should be carried out on the various build options specified to ensure that the tools on the Linux on System z platform provide the expected functionality after the migration (for example, static linking, library compatibilities and other techniques). The effort involved can vary greatly depending on how portable the application code is.

ISV applications

If you are running Independent Software Vendor (ISV) applications on systems that you are targeting for migration, you need to determine if the vendors provide compatible versions that support the distribution of Linux that you plan to use. Many ISV applications have other third party dependencies. Be sure to map out all ISV dependencies, including middleware. Most leading middleware products are available on Linux on System z, and there are many open source alternatives.

3.1.5 Pilot proof of concept

After you have a clear understanding of the target environment and the areas with possible issues and risks, you can proceed to a pilot proof of concept (PoC) phase. This phase is a subset of the actual migration, but with a reduced scope and duration. In this phase you implement a small module or stand-alone code snippet from the application onto the target environment.

The PoC phase should involve all of the same tasks and activities of the full migration. The main objectives of the PoC are to focus on the identified areas of risk, empirically test the recommended approaches, and prove that the full migration can be completed successfully.

In this way, the major potential migration risks identified during the pre-assessment can be addressed in a controlled environment and the optimum solution can be selected and proven. This service targets the areas of issue and risk, proves that the optimal resolution methods have been selected, and provides a “recipe” to be applied during the full migration project.

3.1.6 Decision to migrate

After the pilot is complete, you should have a complete analysis of the target operating system environment as well a roadmap detailing the resources, time, and cost required to migrate to Linux on System z.

During this phase, you analyze and discuss all key requirements with the stakeholders including timing, resource needs, and business commitments such as Service Level Agreements (SLAs). Also discuss any related aspects of the migration, such as new workloads, infrastructure, and consolidation; the decision to implement the migration must be acceptable to all stakeholders involved in such activity.

3.1.7 Resource estimation

Understanding the migration objectives and developing metrics with stakeholder involvement and agreement helps to provide a useful base from which to build a plan. Be sure to build in all key requirements (such as resource needs) and business commitments (such as service level agreements) for each stakeholder.

Migration activities rely heavily on having ready access to the personnel responsible for the development, deployment, and production support of the applications and infrastructure in question. Anticipating change and assuring the early involvement of affected teams are efficient ways to handle change issues. For example, support staff for hardware might be comfortable with UNIX-related hardware support and know where to go for help. However, practitioners who are expert in the previous environment might be less open to change if they feel threatened by new ways of doing things where they do not have expertise.

Consider the following areas in performing your resource estimation:

- Resources

  Determine what hardware and software will be required. Identify the housing aspects required (for example, whether the electrical and cooling inputs are equal). Decide what staff is needed to help with the crossover.

- Education

  Determine whether the staff has adequate Linux education. Decide whether there are special skills needed for the hardware or Linux and hardware combination?

- Service Level Agreements

  While installing, configuring, and testing the change is occurring, determine what the support mechanisms are for both you and any vendors. Determine what your commitments are to current stakeholders while you are performing the migration.

- Related project aspects

  Be sure to set out what other things are happening in addition to the basic system changeover.

- Identify skills-related requirements

3.1.8 Actual migration

The scope of this phase is performing the actual migration of the customer applications and the infrastructure to the Linux on System z environment, thus producing an environment that is ready for handover to the testing phase.

The team follows the planned approach and methodology during their migration activities. If there is a need, modifications are made to the application source code and build environment.The new application binaries are generated and checked for compliance with the target version of the operating system.

3.1.9 Verification testing

The purpose of performing a formal test is to provide objective evidence that the predefined set of test objectives is verified and the customer test requirements are validated on the target operational environment. This is an important step before verification of a successful migration. The ultimate goal is to validate the postmigration environment and confirm that all expectations have been met prior to committing or moving to production.

Keep the following questions in mind for validation:

- Does it interoperate correctly?
- Can it handle the expected load?

Also during this stage, if any performance issues are encountered, the target environment can be tuned for maximum performance.

3.1.10 Verification of success criteria

After you successfully migrate the environment, reassess the original acceptance criteria with all the stakeholders. If the criteria is met, move the environment to production and obtain a sign-off for the migration activities.

If the success criteria are not met, a full review of the migration implementation must be performed. When this review is complete, the testing phase must be redone to ensure that the application being migrated meets the acceptance criteria and is ready to go into production.

4

Migration Planning checklist

This chapter provides you with the basic and generic information and templates you need to quickly assess the source and the target operating environment during the initial planning stages of a migration project.

4.1 The Planning checklist

The planning checklist is used to identify the hardware and software requirements, migration tasks, and project deliverables during a migration project. Although the approach and parameters for a migration planning checklist may vary somewhat from project to project or between organizations, the foundation of an effective planning checklist is similar to the generic checklists we discuss in this chapter. The checklists shown in this chapter are created for the target platform, Linux on System z.

4.2 Hardware Planning checklist

The Hardware Planning checklist template lists the various hardware resources that you need to take into consideration during a migration project. In the checklist template used in this project, the source environment's hardware resources are examined and we need to acquire similar or more advanced technology that is available for Linux on System z. Table 4-1 illustrates a sample Hardware Planning checklist.

Table 4-1 Hardware Planning checklist

Server Name :				
Source		**Destination**		
DEVICE	**Value**	**DEVICE**	**Value**	**Observations**
Number Of CPUs		***Number Of Virtual CPUs***		
Server Memory		***Server Memory***		
Real Memory		Virtual Memory		
SWAP Memory		**SWAP Memory**		
		V-DISK		
		M-DISK		
		ECKD model 3 DASD		
Network Connections		***Network Connections :*** [1]		
Connection Description		**Connection Description**		
Connection Type		Connection Type		
IP Address		IP Address		
		Device Connection Name/Address		
Connection Description		**Connection Description**		
Connection Type		Connection Type		
IP Address		IP Address		
		Device Connection Name/Address		
Connection Description		**Connection Description**		
Connection Type		Connection Type		
IP Address		IP Address		
		Device Connection Name/Address		

OS File System		*OS File System*	
		/ (root file system)	
		mount point	/
		size	
		/usr	
		mount point	/usr
		size	
		/var	
		mount point	/var
		size	
		/tmp	
		mount point	/tmp
		size	
		/home	
		mount point	/home
		size	
		/opt	
		mount point	/opt
		size	
[2] *Custom File System*		*Custom File System*	
Name		Name	
mount point		mount point	
size		size	
Name		Name	
mount point		mount point	
size		size	

[1] The Device Connection Type are :
QETH
Hipersocket
Direct OSA-Express2 connection

[2] Custom File System : This area should be copied so that the number of file systems that exist at source server also exist at destination server .

4.3 Product and Tools checklist

The software Product and Tools checklist template shown in Table 4-2 on page 42 lists all the products and tools that are used in the source operating environment.

It provides space where you can record whether the same or similar products and tools are available on the target Linux on System z operating environment.

Table 4-2 Product and Tools checklist

Application name: **Description:**					
Product/tool	**Version**	**Vendor/Source Web site**	**Used environment (Dev/Test/Prod)**	**Licensed (L) or Freeware (F)**	**Linux on System z (Availability)**

4.3.1 Application Implementation checklist

The Application Implementation checklist delves one level further into the product checklist, where each product or tool is drilled down to their features level. Note that there are scenarios where the same product would not offer the same features on all platforms. These details would be noted in this checklist, as shown in Table 4-3 on page 43.

Table 4-3 Application Implementation checklist

Application Checklist		
Java Application Name : Database Connectivity :		
	Sun- Solaris	Linux on System z
JVM Version		
Compilation Options		
JIT / Optimization parameters		
Native dependencies		
Third party jar's dependencies		
Custom Application Name : Language Used : Database Connectivity :		
	SUN-Solaris	Linux on System z
Application Arch Model	32-bit	64 bit, 31 bit
Compiler Used		gcc
OS Version		SLES 10, RHEL 5
Compiler Version		
	Compiler Performance Options	
Compilation		
Linking		
Shared Library		
System Libraries Used		
For Debug Build		
	Compiler Build Options	
Compilation		
Linking		
Shared Object Creation		

Each product or tool listed in the product checklist must be analyzed. All the parameters, dependencies, and optimization options must be taken into account in the source operating environment, and then the planning team must assess

whether the same kind of features or build options are available in the target operating environment.

If the same feature is not available with the same tools or product in the target environment, the team can assess other options:

- Obtain a similar feature by linking other product or tools in the target operating environment.
- Make note of the parameters available in the same tool in the target operating environment that can be combined to give the same characteristics as in the source environment.
- If the products or product options are fully incompatible or unavailable, replacing that part of the application stack would be a useful approach to minimize the effort involved in migration. But care must be taken to ensure that all the features and parameters offered by the product in the source environment are also available in the assessing product for the target environment.
- Often the optimization features or performance options for a product are only available for that specific platform. In such cases, the optimization features and performance options must be changed to offer the same characteristics in the target environment.

When filling out the Application Implementation checklist, you need to verify whether changing parameters or options in the target operating environment has any side effects on the application or other tools used for application implementation.

If all the checklists are properly analyzed and applied, then the tools, products, and their implementation differences would be accounted for in the actual migration. This would in turn reduce the risks and the migration can be executed smoothly.

4.3.2 Training checklist

A critical element in achieving successful migrations is ensuring that a Training checklist is put in place during the planning process. You will need to identify the people to be trained, the skills that need to be imparted, and a timetable of when the training needs to be done to ensure that staff are trained at the right time. Refer to the Stakeholder analysis in 2.1.2, “Information Technology stakeholders” on page 25 for more information about this topic.

4.3.3 Application Environment checklist

The source application to be migrated could be in the center of a very complex process. The application could be interconnected with many other applications, inputs, outputs, and interfaces. For this reason, you need to prepare a planning document that lists the resources that the source application needs to provide and all the services that it is currently providing. Table 4-4 lists examples of the resources that are required of some applications.

Make the descriptions as detailed as possible by providing the physical location, host name, IP address, network information, software product used, service owner, and so on. You can also document the tasks required for the migration effort.

The target resource and target location may be the same as the source resource and source location. By listing the resources and their location, you will ensure that assumptions made by one group of stakeholders are not different from the assumptions made by other stakeholders.

Table 4-4 Application Environment checklist

Source resource	Source location	Target resource	Target location
Internal FTP	FTP server on source application server		
External FTP	Batch process through central and secure FTP server		
Local print	Local print on local LAN		
Remote print	Vendor product secured by host name		
DNS services	Single or multiple DNS servers		
Firewalls	Firewall locations and functions		
Internet connectivity	Router location		
Intranet connectivity	Web server location and ports		

Source resource	Source location	Target resource	Target location
E-mail services	Mail transfer agent co-located on source application server		
Messaging services	MQ on source server		
Client software	User's desktop User's laptops Mobile appliance		
File services	Type, location, and security		
Log server	Central server location		
SNMP	Agent and server location		
Fax server	Secure server on network		

Part 2

Migration analysis

In this part we discuss the planning detail of a migration by covering a number of key elements involved in the overall migration. These elements are:

- Technical concepts of the System z platform, z/VM hypervisor, and Linux
- Networks
- Storage
- Applications
- Databases
- Backups and Archives
- Security
- Operations
- Disaster Recovery and Availability

5

z/VM and Linux on System z technical concepts

This chapter provides technical information about z/VM and Linux on System z. Topics discussed are:

- Virtualization concepts
- z/VM Operating System components
- Logical partitions
- CPU
- Memory
- Disk
- Networks

5.1 Virtualization concepts

A standard feature of IBM System z is Processor Resource/System Manager (PR/SM). PR/SM is a Type 1 virtualization hypervisor where a single IBM z10 EC can be defined with up to 60 logical partitions (LPARs). An IBM System z10 BC can de defined with up to 30 LPARs.

Each LPAR has access to a defined set of CPU, Memory and I/O resources, as well as an associated weighting that defines its relative priority against other LPARs. This allows you to run several operating systems such as z/OS, Linux on System z, z/VSE, z/TPF, and z/VM in their own LPARs. CPUs defined to an LPAR are known as *logical CPUs*.

The z/VM Operating System or hypervisor is the base of the second virtualization level on System z. At this level all hardware requests are handled by the z/VM hypervisor and correctly dispatched to the hardware. That allows multiple guest operating systems to share physical resources. Depending upon the LPAR's configuration, the real resources available could be multiplied as virtual resources to be used by the guests. At the z/VM level you can also have, as guest operating systems, z/OS, Linux on System z, z/VSE, CMS, and z/VM.

Figure 5-1 shows the relationship between the IBM System z hardware, PR/SM controlled LPARs, logical CPUs, shared I/O, and operating systems.

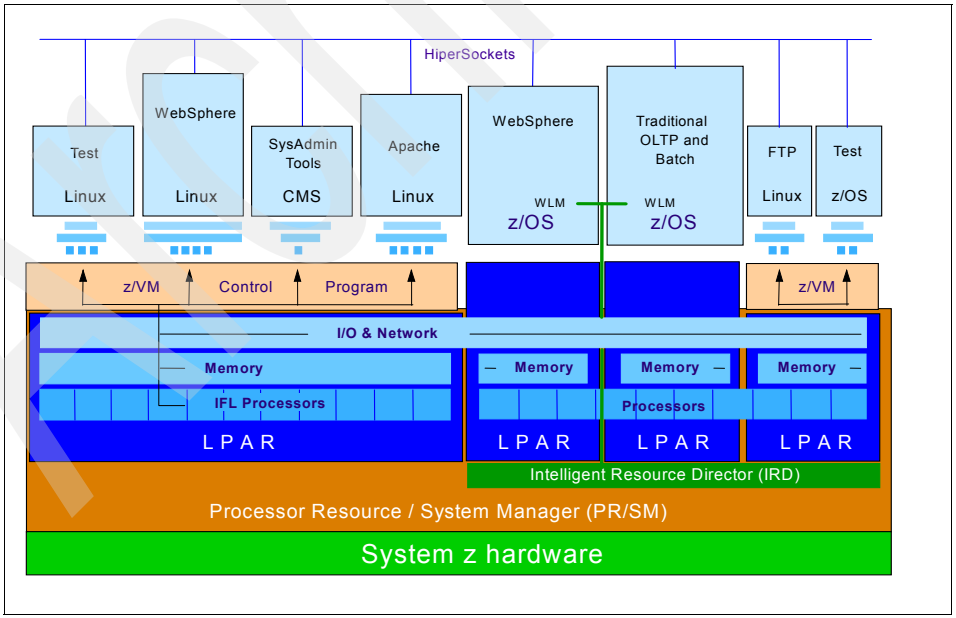

Figure 5-1 Virtualization environment

5.1.1 z/VM Operating System components

This section describes and explains the components of the z/VM Operating System.

Control Program

The Control Program (CP) provides a user (in our case, the Linux operating system) with a complete System z environment on a virtual machine with virtual resources that are equivalent to the real hardware resources. Communication with the control program is via CP commands that are used by operations staff and Linux administrators to manage, query, and allow the definition of additional resources.

When a Linux guest logs onto a z/VM session, it is starting its own CP session. For production systems this is usually done automatically when the z/VM system is initially loaded (IPLed) or booted. There is an entry in the z/VM directory for each virtual machine that can be started.

Each entry contains information about the virtual resources that are required by the guest operating system, as well as details of the relative priority of the virtual machine. This is used by CP in determining which virtual machine is to be dispatched. Communication to the Linux system can be either via the Linux virtual machine console (which must be a 327-type terminal emulator), or more commonly via a Telnet and SSH client terminal.

Note: If an administrator logs off the Linux virtual console, the virtual machine will shut down and terminate all running work. The administrator must use the `DISCONNECT` command (not the `LOGOFF` command) to ensure that this does not occur.

Refer to “3270 Emulation” on page 246 for more information about 3270 sessions.

Conversational Monitor System

The Conversational Monitor System (CMS) is an operating system that runs only as a z/VM guest. CMS is used by the z/VM system administrator to manage the system components and to create and edit virtual machine user profile entries in the z/VM environment. CMS is the operating system for many service machines such as TCP/IP, Print Services, Directory Maintenance, Accounting, Error Recording, and so on.

For more information about z/VM, refer to *Introduction to the New Mainframe: z/VM Basics*, SG24-7316.

5.1.2 Logical partition

At most Linux on System z installations, a single System z is configured into at least two different logical partitions (LPARs). One LPAR is defined for Linux development and test, and a second LPAR is defined for Linux production.

When you are working in a virtualized environment, keep in mind that a performance problem that affects one guest in a z/VM host could affect the performance of all guests in the same LPAR. Using at least two different LPARs, configured correctly, can avoid problems in some common procedures such as application and operating system updates and upgrades, and in new application deployment, test, and evaluation.

Because it is necessary to assign logical CPUs to each LPAR, the number of dedicated and shared logical CPUs will depend on the number of physical IFL processors that the machine has.

The unique resource that cannot be shared by the LPARs is the memory. However, when you are defining Linux guests on z/VM, it is possible to allocate more memory on the Linux guests than is available to the LPAR.

5.1.3 CPU

The number of IFLs on the machine will reflect directly on the performance of the Linux guest running in an LPAR. The number of virtual CPUs allocated to a single Linux guest should be not greater that the number of logical CPUs allocated to the machine. For example, if the LPAR has four IFLs, then do not allocate five virtual CPUs to a single Linux guest machine. If a situation occurs where the Linux guest uses 100% of the CPUs, that will adversely affect the entire LPAR.

However, in an LPAR with four IFLs, you can assign three virtual CPUs to a LinuxA guest and two virtual CPUs to a LinuxB guest, as well as another two virtual CPUs to a LinuxC guest. All requests for CPU cycles will be managed by z/VM according to the relative priorities of the Linux guests. CPU configuration best practice is to maintain the ratio of 4 active virtual CPUs to 1 logical CPU allocated to the LPAR.

5.1.4 Memory

System memory (to use the Linux term) or storage (to use the z/VM term) is a resource that is shared across all z/VM guests. Each virtual guest is assigned a defined amount of virtual storage during logon.

The key to efficient memory management is to be aware of the total amount of virtual memory that is likely to be active at any time, and also be aware of the amount of real memory (storage) that is allocated to the z/VM LPAR.

z/VM allows you to overcommit memory, but keep the overcommitment ratio of the total amount of virtual memory likely to be active to total amount of virtual memory to around 2:1. For test or development workloads, the ratio should be no more than 3:1.

The keys to determining the appropriate virtual memory size are to understand the working set for each virtual machine, and to ensure that the Linux images do not have any unneeded processes installed. Another recommendation is to use VDISKs for swap, as described in “Swap device consideration” on page 55.

Memory Management features

There are memory management features for Linux and z/VM that you can use to reduce the amount of memory required by virtual guests:

- Cooperative Memory Management (CMM)
- Collaborative Memory Management Assist (CMMA)
- Named Saved Segment (NSS)
- Discontiguous Saved Segment (DCSS)

CMM

CMM is used to reduce double paging that may happen between Linux and CP. CMM requires the IBM Virtual Machine Resource Manager (VMRM) running on z/VM to collect performance data and notify the Linux guest about the constraints when they occur. On Linux servers the `cmm` kernel extension is required, and it is loaded with the **`modprobe`** command.

CMMA

CMMA enables CP and Linux to share the page status of all 4 KB pages of guest memory. Linux does this by marking the status of each page; this allows CP to preferentially steal unused and volatile pages and thus reduce paging.

NSS

NSS allows virtual guests to share a read-only copy of a single operating system such as CMS or Linux. The benefit of this feature is that only one copy of the operating system resides in storage accessible by all virtual machines. This decreases storage requirements and simplifies maintenance.

DCSS

DCSS allows virtual machines to share reentrant code for applications, such as Oracle, which also reduces overall storage requirements. Figure 5-2 on page 54 illustrates how both NSS and DCSS work. There is one copy of the application in

real storage and Linux guests use this single copy. The NSS copy of Linux is also shared by all virtual guests.

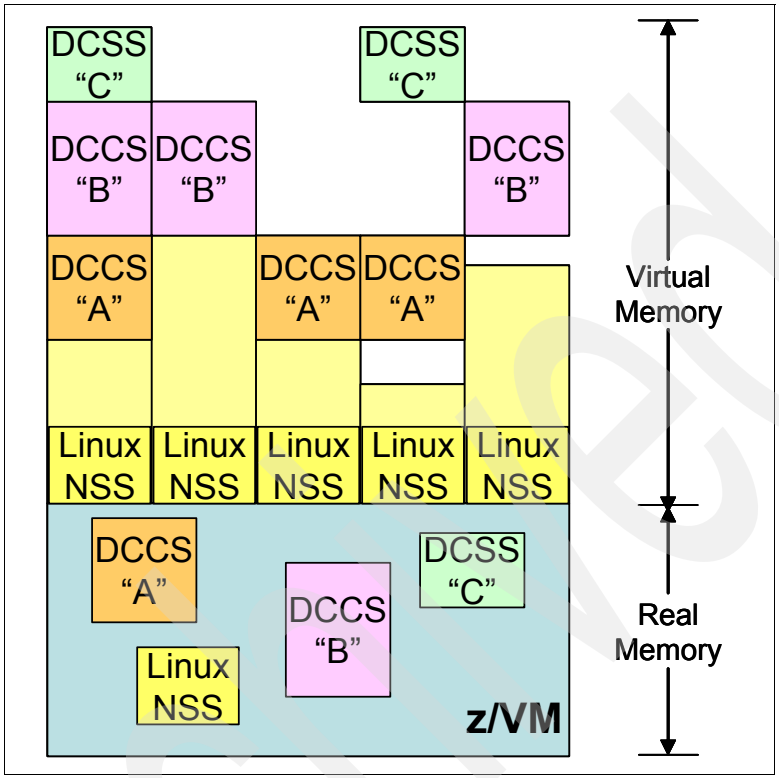

Figure 5-2 DCCS and NSS shared by multiple Linux virtual guests

For more information about setting up a Discontiguous Saved Segment and using the Execute-In-Place (XIP) file system, refer to *Using Discontiguous Shared Segments and XIP2 Filesystems With Oracle Database 10g on Linux for IBM System z*, SG24-7285.

Note: When defining memory requirements for virtual Linux guests, remember that the Linux kernel will use all the extra available memory allocated to it as a file system cache. Although this is useful on a stand-alone system (where that memory would otherwise go unused), in a shared resource environment such as z/VM this causes the memory resource to be consumed in the LPAR.

Therefore, it is important to assign only the memory needed for the running applications when they are at peak load.

Linux swap should be thought of as an overflow when an application cannot get enough memory resource. Thus, when paging occurs, this is an indication that either more memory needs to be assigned or the application needs to be analyzed to understand why more memory is needed.

Swap device consideration

The Linux on System z swap device must be configured if possible as a VDISK device. VDISKs are virtual disks allocated in memory, and they become a fast swap device for Linux. However, the Linux administrator should configure at Linux boot time to format the VDISK as a swap device. This must be done at every start or recycle of Linux.

For more information about optimizing memory on z/VM and Linux, refer to *Linux on IBM System z: Performance Measurement and Tuning*, SG24-6926.

5.1.5 Disk

System z disk storage is commonly referred to as Direct Access Storage Device (DASD). Traditionally, IBM System z has supported only Extended Count key Data (ECKD™) DASD, which was a developed from Count Key Data (CKD) devices to provide improved performance for Fibre Channel-connected DASD.

With z/VM and Linux on System z, disk device support is expanded to Fixed Block Architecture (FBA) DASD and also to Small Computer Systems Interface (SCSI). FBA and SCSI disks are connected to System z via the Fibre Channel Protocol (FCP).

The ECKD devices are defined as one of three 3390 DASD models, each of different sizes. The models, and their capacity as measured in cylinders and in megabytes, are listed in Table 5-1.

Table 5-1 3390 DASD models

Model	Cylinders	Megabytes
Model-3	3339	2354
Model-9	10017	7036
Model-27	30051	21123

It is good practice to use ECKD DASD to boot Linux and static data, and use SCSI FCP for data applications, because the performance of FCP disks is better.

The connection to SCSI devices is managed by the zFCP Linux module driver. The SCSI devices are usually dedicated to the Linux guests. For a more detailed

discussion about storage, refer to Chapter 7, “Storage analysis” on page 71. For more information about z/VM and disks, refer to *Introduction to the New Mainframe*, SG24-7316.

5.1.6 Network

For a Linux on System z environment, the physical network attach devices are the Open Systems Adapter Express 2 (OSA-Express2) or Open Systems Adapter Express 3 (OSA-Express3). These are capable of handling up to 640 TCP/IP stacks simultaneously, including Hipersockets for inter-LPAR communication.

An IBM System z10 EC can support 48 OSA-Express2 or 96 OSA-Express3 ports. The Open Systems Adapter supports both copper and fibre Ethernet connections at speeds of up to 10 Gb.

The z/VM Operating System feature to access the TCP/IP network is TCP/IP for z/VM. OSA-Express2 can be virtualized through a virtual switch (VSWITCH) device to many Linux guests. It is available using special z/VM machines known as VSWITCH controllers. Each Linux guest connects using a virtual device controlled by the qeth module to a virtual switch system in a z/VM LPAR.

An important benefit of the VSWITCH system is that it can be set up with redundant OSA devices that provide a failover network system on z/VM.

HiperSockets provides high-speed connectivity between servers running on a System z. This technology does not require any special physical device configurations or cabling.

Both OSA-Express and HiperSockets use the Queue Direct I/O (QDIO) mechanism to transfer data. This mechanism improves the response time using system memory queues to manage the data queue and transfer between z/VM and the network device. Various examples are available in Chapter 6, “Network analysis” on page 57.

For more information about network in Linux and z/VM, refer to *Linux for IBM System z9 and IBM zSeries*, SG24-6694.

6

Network analysis

This chapter provides information about network migration configuration issues and explains how the virtual network can be configured.

In this chapter the term OSA-Express2 is used when discussing the System z Open Systems Adapter. The IBM System z10 also supports OSA-Express3.

OSA-Express3 provides 10 Gb Ethernet long-range (up to 10 km, fiber optic only) and 10 GB Ethernet short-range (up to 550 m, fiber optic only) along with 1 GB Ethernet long and short wavelength (fiber optic only) and 1000Base-T Ethernet feature. OSA-Express3 also adds four ports per card, to a maximum of 96 ports for an IBM System z10 EC and BC.

6.1 Network migration overview

There are several different levels of the network migration that should be considered because z/VM provides a complete virtual network system which includes the possibility to create multiple virtual switches in the same LPAR. Virtual switches (or VSWITCH) allow, among other features, the use of Virtual LANs (VLANs).

The VSWITCH operates at either Layer 2 or Layer 3 of the OSI Reference Model, and is virtually attached to the same network segment where the OSA-Express card is physically connected.

6.1.1 Single network scenario

One of the most common scenarios is the migration of several distributed machines from the same physical subnet to a single System z LPAR attached to the same network segment. Figure 6-1 shows an example depicting a single distributed network.[1]

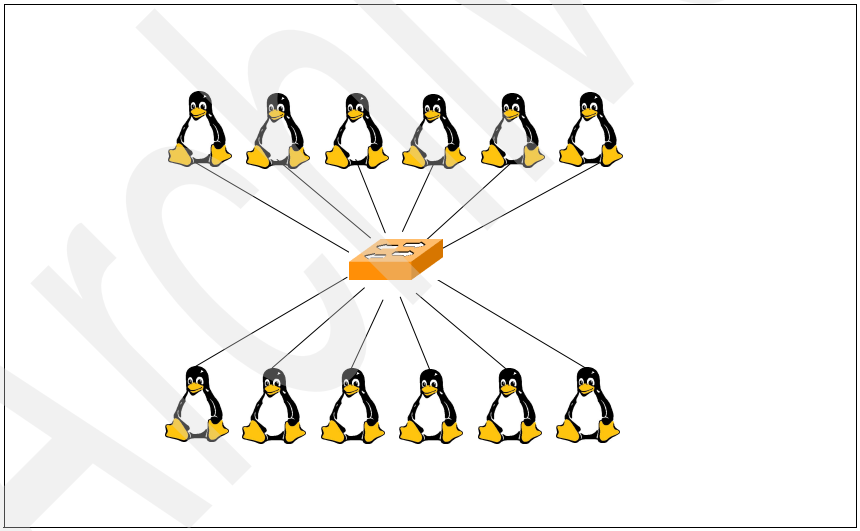

Figure 6-1 Single distributed network

Within this scenario, all physical machines can be migrated to a single System z machine running Linux sharing the same VSWITCH which is attached to an OSA-Express2 card. The OSA-Express2 card is then connected to the physical network. Figure 6-2 on page 59 illustrates this type of configuration.

[1] Tux symbol created by L. Ewing, lewing@isc.tamu.edu

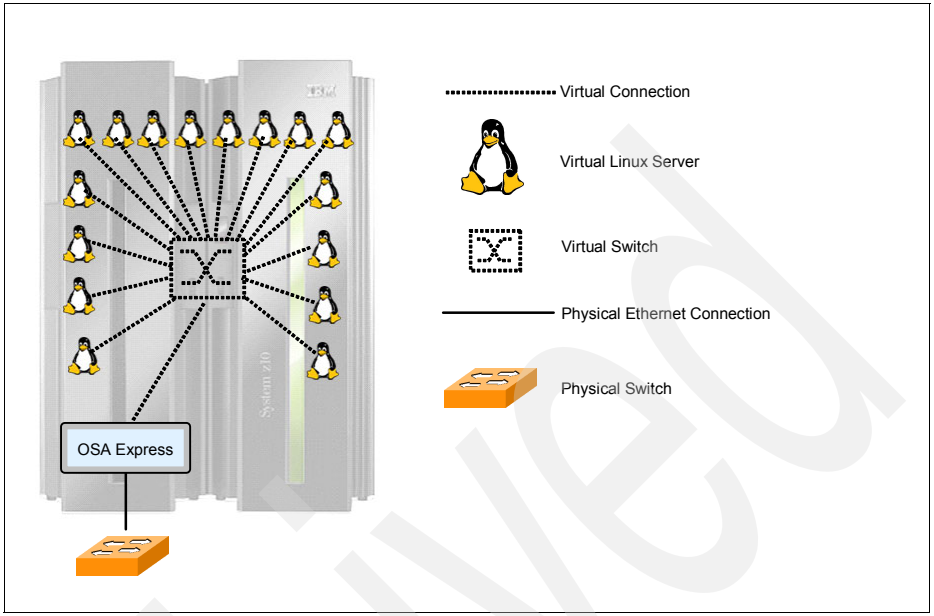

Figure 6-2 Single virtualized network

To increase the availability of each Linux guest, the recommended solution is to configure two or three OSA-Express2 cards attached to different physical switches in the network. This will provide a network failover capability for the entire system, as illustrated in Figure 6-3 on page 60.

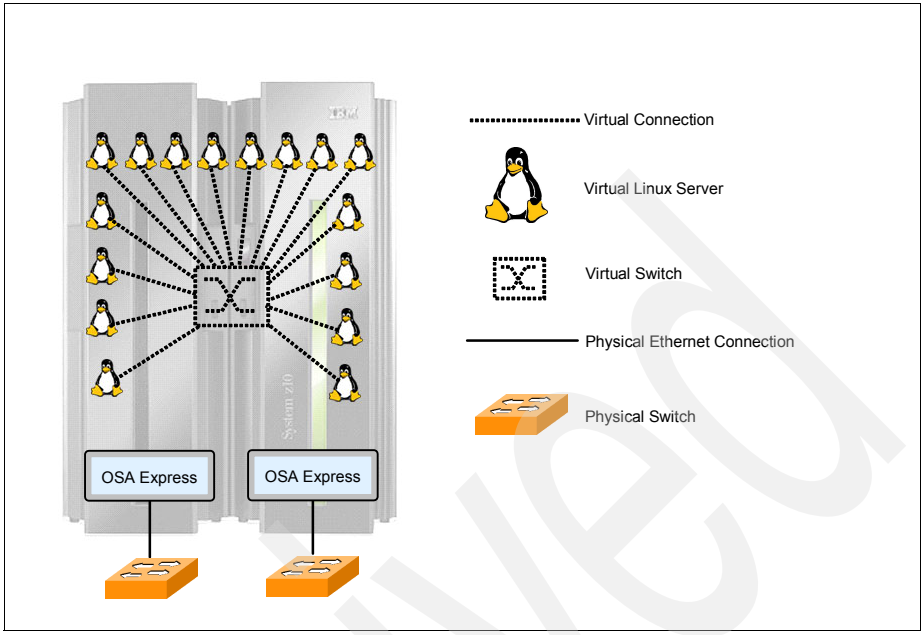

Figure 6-3 Single virtualized network with failover solution

In a Layer 2 VSWITCH configuration, all Linux guests have their own MAC address. Otherwise, in a Layer 3 VSWITCH configuration, the Linux guests respond with the OSA-Express2 card's MAC address to requests from outside the System z LAN segment.

In a multiple LPAR scenario where a single network segment is used, the recommended solution is to share the OSA-Express2 card between LPARs. Each LPAR's VSWITCH is connected to the OSA-Express2 card and this is directly connected to the physical network segment.

This is a common scenario where the development and production server are in separate LPARs. This configuration is illustrated in Figure 6-4 on page 61.

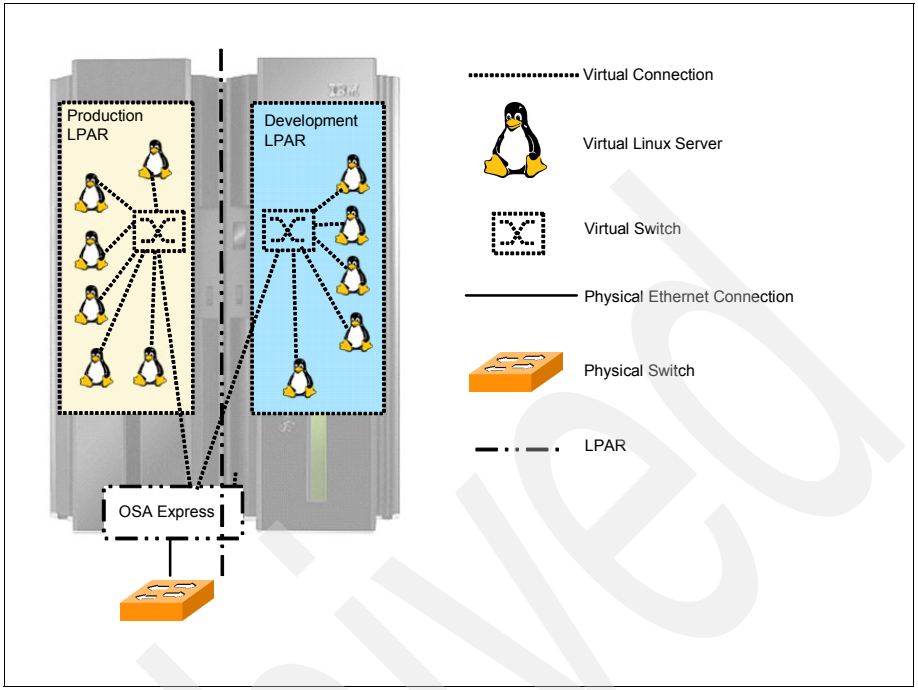

Figure 6-4 Single virtualized network with multiple LPARs

Similarly, the failover solution described previously can also be applied in this case. Sharing the two OSA-Express2 cards between LPARs as shown in Figure 6-5 on page 62.

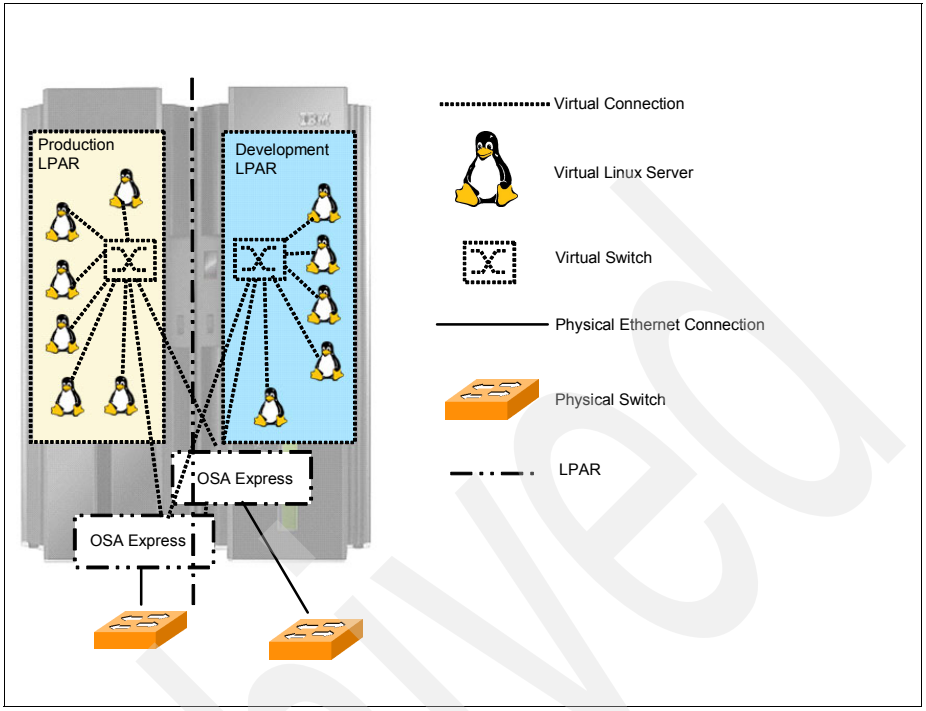

Figure 6-5 Single virtualized network with multiple LPARs with failover

6.1.2 Multiple network scenario

There are several types of network solutions that require multiple network segments. Some of these demand package routing or the use of multiple LPARs. This section provides suggestions for each type of network design.

DMZ and secure network

In some scenarios, different network segments are migrated to Linux on System z and share the same System z. Notice that we are analyzing the DMZ and a secure network scenario. Figure 6-6 on page 63 shows a DMZ network where the Web Application Server is placed, and a secure network where the database server is located.

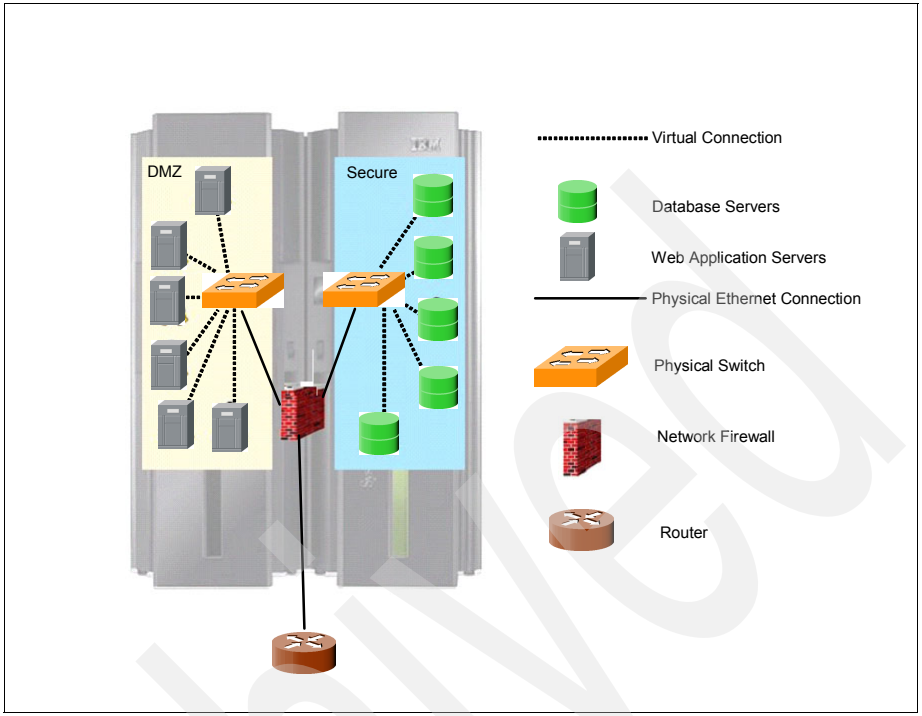

Figure 6-6 Multiple distributed network scenario - DMZ segmented network

Figure 6-7 on page 64 and Figure 6-8 on page 65 illustrate the network architecture used in a Linux on System z scenario.

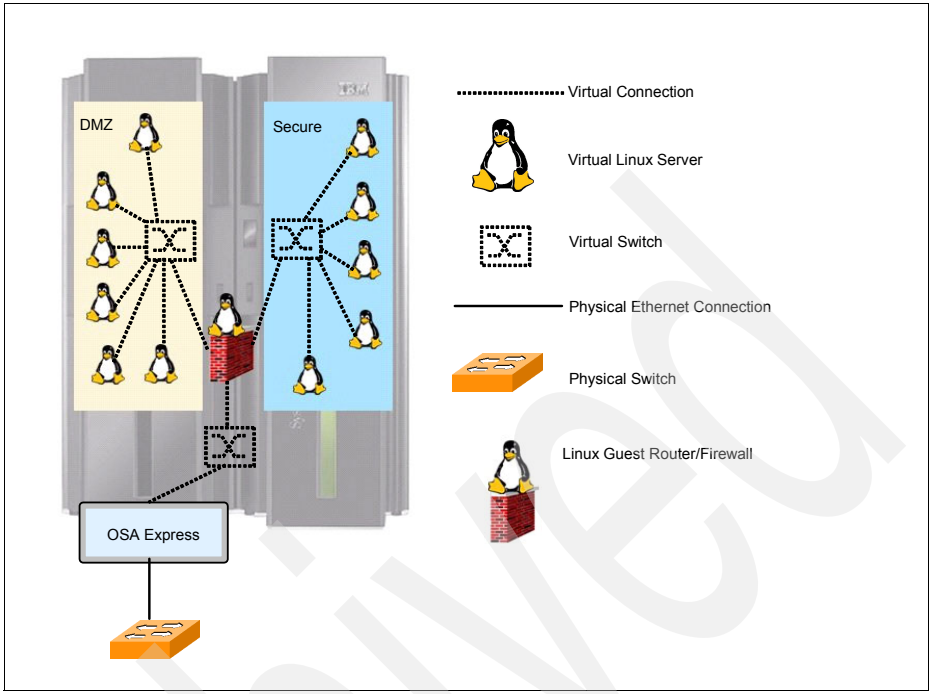

Figure 6-7 Multiple virtualized network scenario - DMZ and secure network

The OSA-Express2 card is connected to one physical switch (or two OSA cards, when the failover solution is configured). The physical firewall will be replaced by a Linux guest that will act as a router and firewall. All virtual Linux guests (servers) will be connected to two VSWITCHs in the Linux system.

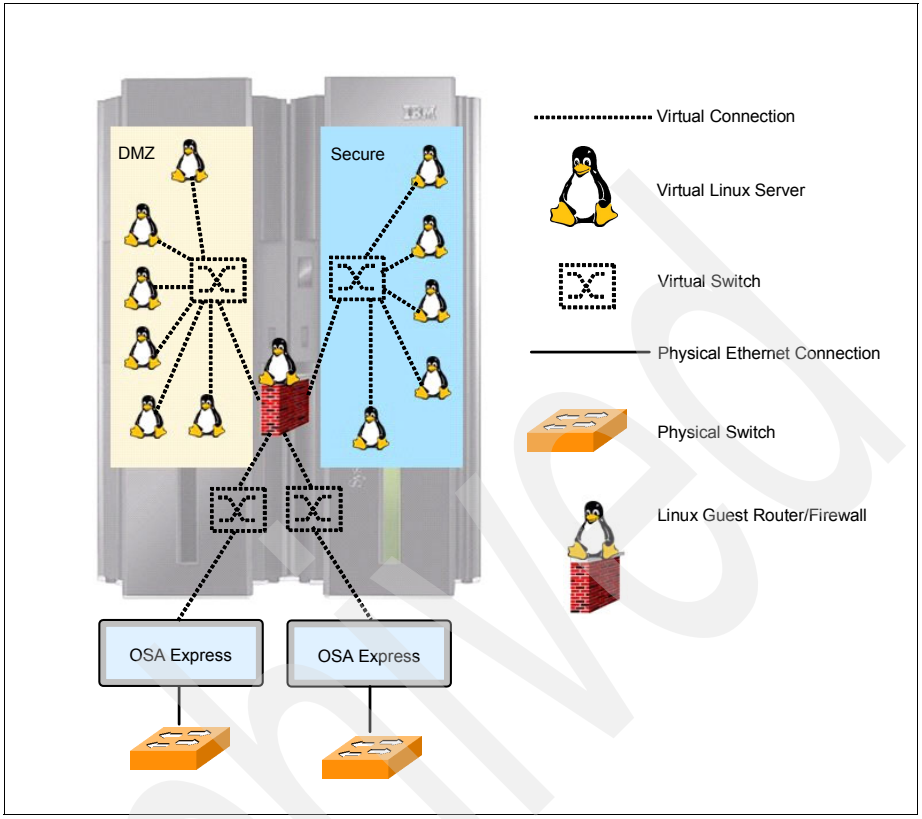

Figure 6-8 Multiple Virtualized Network Scenario with failover- DMZ and Secure network

In a second scenario that uses multiple logical partitions, you could isolate the entire secure network from the physical network segment. The communication between the LPARs is managed by HiperSockets devices.

HiperSockets is a microcode implementation that emulates a Logical Link Control Layer of an OSA-Express interface. HiperSockets provides very fast TCP/IP communications between servers running in different LPARs. HiperSocket simulation is sometimes referred to as interface Queued Direct I/O (iQDIO). Figure 6-9 on page 66 illustrates an example.

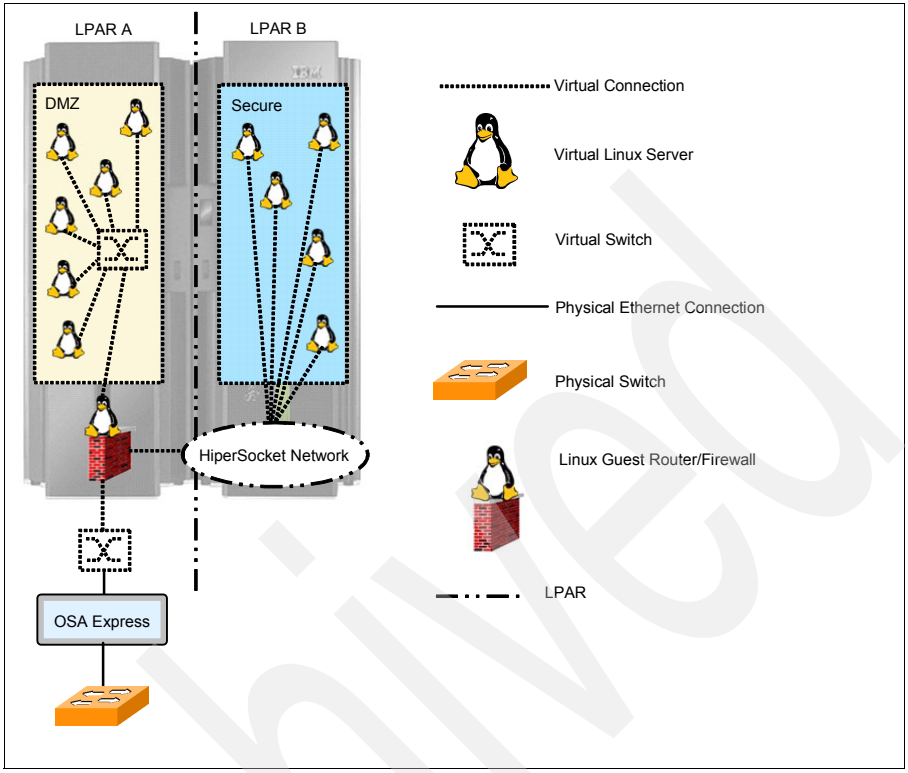

Figure 6-9 Multiple virtualized network scenario with multiples LPARs

Note, however, that although the use of HiperSockets for this scenario is possible, but it may not be the recommended solution. If one of the LPARs is CPU-constrained, that could cause a delay of network traffic.

A recommended solution for multiple networks

It is also possible to have the OSA-Express2 card shared between multiple LPARs on the same System z hardware. To create this kind of solution, it is recommended that you have an external firewall to manage the network filters. Figure 6-10 on page 67 illustrates the solution described as a network segmented LPAR.

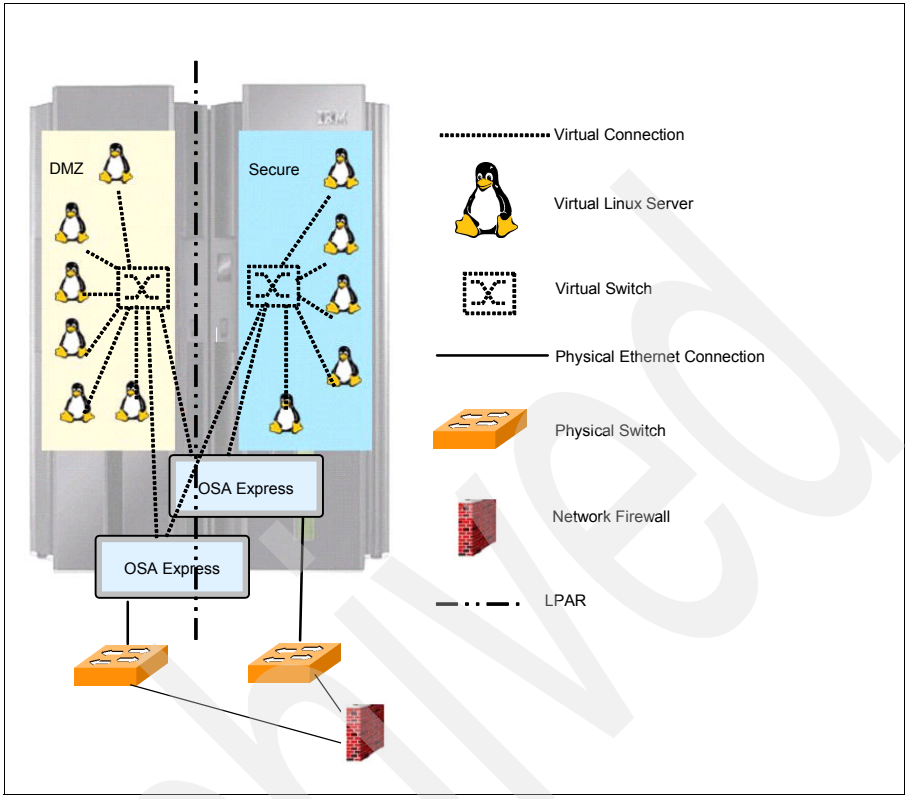

Figure 6-10 Multiple virtualized network scenario with external firewall

VLAN segmented network

For a high security network scenario, use the LPAR environment mixed with the multiple network segmented solution. This includes the VLAN configuration to help in the segmentation of network packages.

In a distributed environment, multiple switches can share the same VLAN. However, because of switch port limitations, that could be a painful and complex solution; see Figure 6-11 on page 68.

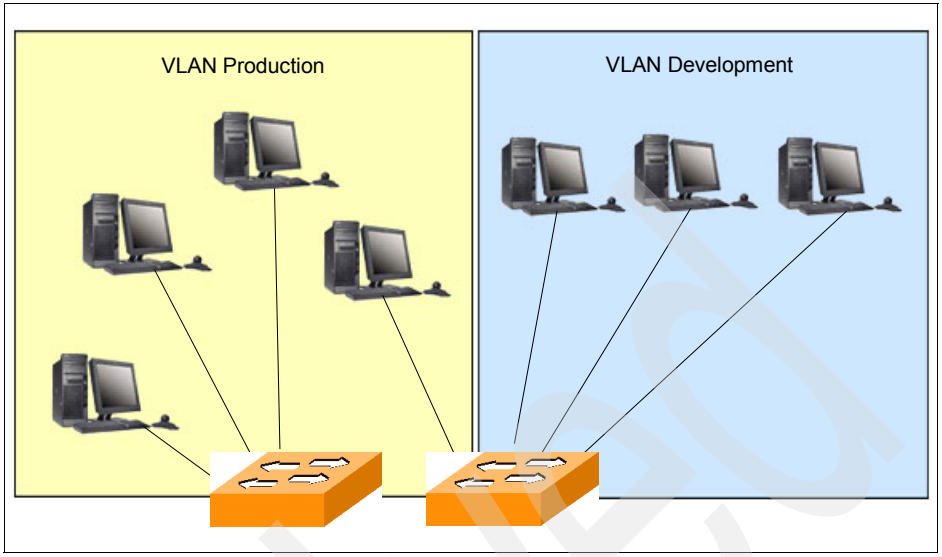

Figure 6-11 Distributed environment with multiple VLANs

As illustrated in Figure 6-12 on page 69, the entire System z environment is virtualized and all configurations are made per virtual machine, which increases the security and reduces the complexity.

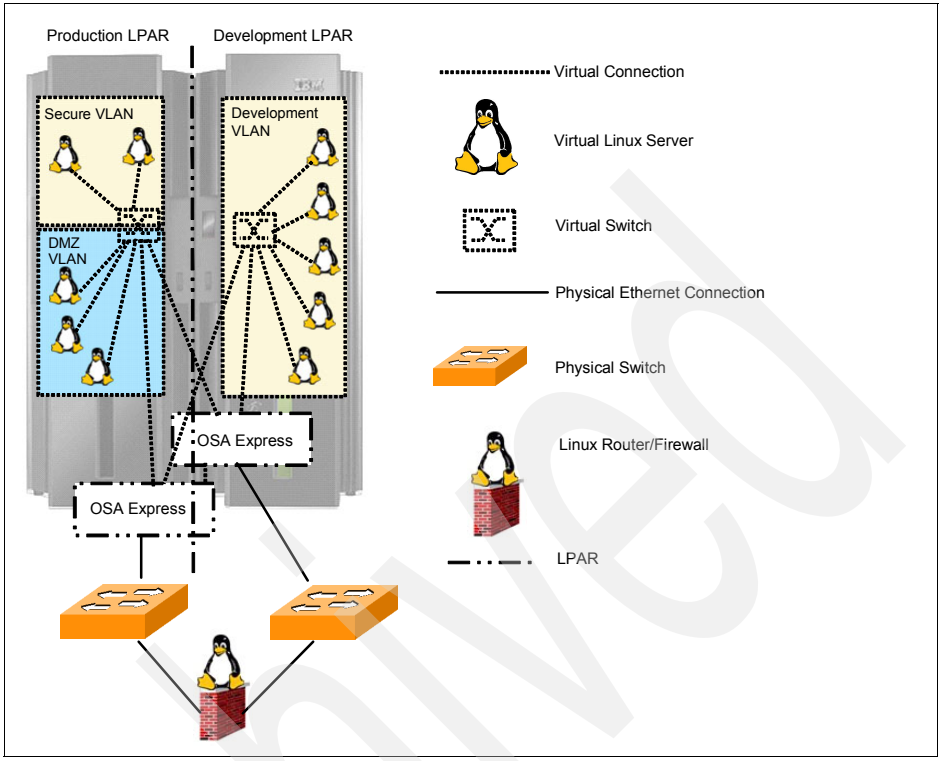

Figure 6-12 Multiple virtualized network scenario with external firewall and segmented VLANs

6.2 Steps for a successful network migration

The Linux on System z administrators and network administrators should work together to engineer the best solution for your environment. Here are the basic steps:

1. Determine the new IP address for the new servers. The IP address should be on the same IP network to minimize the number of variables of the entire migration.
2. Determine the VLAN IDs to the Linux on System z servers.
3. Configure the VSWITCH with the listed VLAN IDs.
4. Configure the Linux server's QDIO devices within the designated IP address.

At this point the destination server (Linux on System z server) must be designated with a different host name from the source server name.

1. Migrate the applications and files from the source server to the destination server.
2. Shut down the source server.
3. Change the Linux on System z server's host name.
4. Change the DNS registered name to the new Linux on System z IP address.

If the application running is an IP-based application, it is possible to change the IP address of the target Linux on System z server to the source IP address.

7

Storage analysis

This chapter is explains concepts and designs, such as online migration and offline migration, regarding the storage configuration possibilities for Linux on System z. Other storage migration issues are also covered.

7.1 Data migration

Two models of data migration are discussed in this chapter: online migration and offline migration.

- Online migration refers to the case where the source server, target servers, and all services are up and running and a system outage is not required.
- Offline migration requires a service outage to switch over from the source server to the target servers.

We examine both migration models in more detail in the following sections.

In both types of data migration, some unexpected issues must carefully be considered. The result of not doing so could lead to an extended outage or unexpected downtime, data corruption, missing data, or data loss.

To avoid such issues, online data migration must always be executed during off-hours, and you should always take a data backup just before the actual data migration activity begins.

7.1.1 Online data migration

Some applications are eligible for online migration. To be eligible, basically, an application must provide multi-operating system clustering support and be available on Linux on system z.

To perform an online migration, follow these steps:

1. Install and configure the target Linux on System z server (refer to “Linux on System z - preinstallation considerations” on page 76 for more details).
2. Install the middleware application on the Linux on System z server.
3. Copy the application data to the target Linux on System z server.

   The software application selection depends on the type of data that needs to be copied. Solutions like the Linux cp program can be used in online data migrations where the application does not change or the changes are totally controlled.

   Otherwise, the Rsync software application can be used to synchronize the application data between the server in a small period of time during the migration process.
4. Include the Linux on System z server in a cluster as a cluster node.

5. Monitor the Linux on System z server to verify that the application is responding to requests correctly.

   This step is not a test of the application on Linux on System z. The application must be tested on a development machine to guarantee that the application is a Linux on System z-compatible application (refer to Chapter 8, “Application analysis” on page 85 for more detail).

6. Shut down the source servers.

Always consider the content of the data that is migrated before choosing online migrations as a solution.

7.1.2 Offline data migration

Offline data migration can apply to all system migrations. This kind of data migration can be accomplished by using several different approaches and functionality including:

- Using the network mount points NFS or Samba connections and either the **DD** or **CP** Linux command.
- Using an FTP server on the source or target server.
- Using an SCP/SSH server between the server and the target server.
- Using the Rsync synchronization application between the source or target server.
- Attaching the storage volume to a Fiber Channel device (Linux-to-Linux migration).

Using the Rsync application

For a better result using the Rsync application, schedule service synchronization for an entire week before the outage by following these steps:

1. On the first migration day, execute the first synchronization.

   Execute the first synchronization during a time when the use of the server is low. (Rsync only copies files that are not locked, thereby avoiding any issues with files in use.) During this time, however, server response may be slower than normal because of the extra read I/O activity.

2. During the migration week, you can execute a daily synchronization at the server during off-peak hours.

   Only modified files will be copied from the source to the target server.

3. The last synchronization day is the server outage day, when access to the source server is denied to users.

   Because there are no open files, the Rsync application will be able to copy all files to the target servers.

4. Shut down the source servers and start all services on the target Linux on System z servers.

Transferring files over the network

Database migrations are the most common example of the requirement for files to be transferred over the network. That is because most database software needs an offline backup that includes a data export or data dump to a new file.

That exported/dumped file needs to be transferred across the network and the database import procedure must be executed at the target server; refer to Chapter 9, “Database analysis” on page 99 for more details.

Migrating storage volumes

When the source servers are Linux x86- or Linux on Power-connected to an external storage device using Fibre Channel, and if there is a zFCP device that is part of the same storage area network, it is possible to connect the source Linux volume to the target Linux server on IBM System z. However, both servers cannot share the same volume.

Storage SAN Volume Controller

One option available to simplify the storage and data migration for Fibre Channel disks involved in a migration to Linux on System z is to install the IBM System Storage™ SAN Volume Controller (SVC).

The SVC sits in the channel path and allows you to virtualize all FCP storage from multiple vendors that sit behind it. Figure 7-1 on page 75 shows where the SVC sits in the Storage Area Network (SAN). The SVC has visibility to all supported storage on the SAN.

The following benefits are provided by the SVC:

- Single point of control for heterogeneous storage resources
- Dynamic data migration between heterogeneous storage devices on a SAN
- Ability to pool the storage capacity of multiple storage systems on a SAN
- Scalability to support up to 1024 host servers
- Instant copies of data across multiple storage systems with FlashCopy®
- Copy data across metropolitan and global distances as needed to create high-availability storage solutions

When migrating Linux systems from x86, IBM Power, or IA-64 to Linux on System z, the SVC will allow you to nondisruptively migrate data to Linux on System z. For more information about the IBM System Storage SAN Volume Controller see:

http://www.ibm.com/systems/storage/software/virtualization/svc/index.html

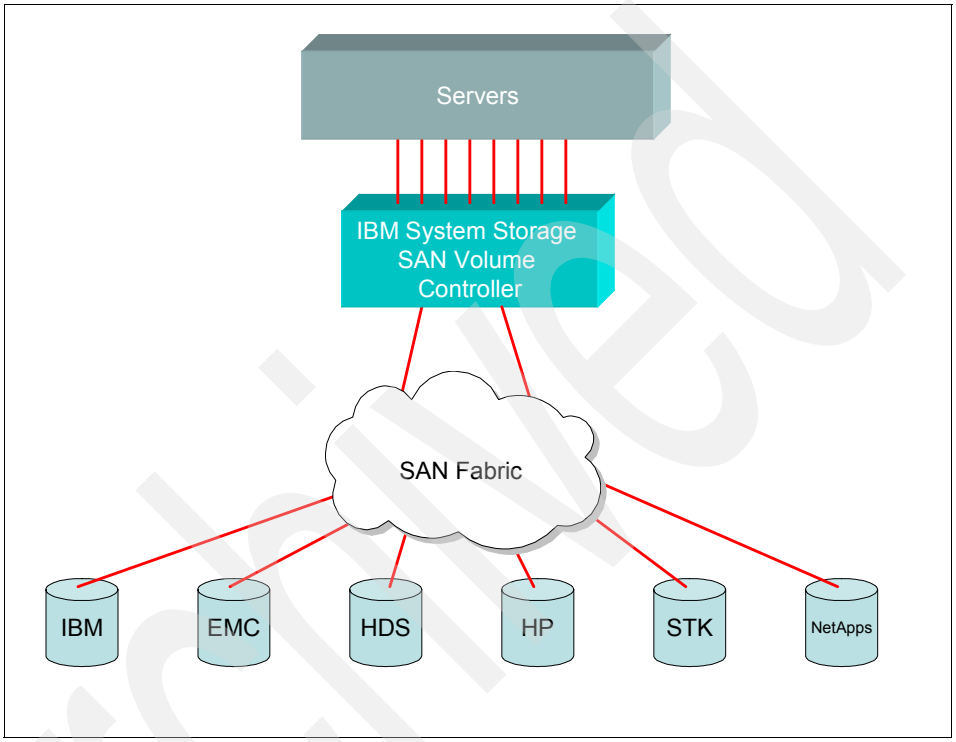

Figure 7-1 SAN Volume Controller

Note: There are also a number of Redbooks that have been written about the SVC. Some of these are:

- *SAN Volume Controller V4.3.0 Advanced Copy Services*, SG24-7574
- *Using the SVC for Business Continuity*, SG24-7371
- *Implementing the IBM System Storage SAN Volume Controller V4.3*, SG24-6423
- *SAN Volume Controller Best Practices and Performance Guidelines*, SG24-7521

7.1.3 Steps for a successful offline storage migration

The multiple possibilities provided by Linux on System z to store and access files lead to many types of solutions. The solution you architect for the target system will dramatically affect the flexibility, efficiency, and performance of the migrated application.

For source applications that reside on servers where storage is local or the external storage is not compatible with Fiber Channel data storage, all data must be copied using the network file system from the source server to the target server (Linux on System z).

1. Create the new server file system with mount points for all data files (refer to “Linux on System z - preinstallation considerations” for more details).
2. Create a temporary file system to be used in the file transfer process on the target server.
3. Configure the network on the target server (refer to “Steps for a successful network migration” on page 69).
4. Configure the target server as an NFS file server, a Samba file server, or an FTP File Server to upload the files from the source server.
5. Note the following points:

   - If there is enough space at the source server to compact all of the data, consider using data compression features such as zip, or tar with gzip and bzip formats. Both of these formats are compatible with Linux on System z. The data can be transferred using an FTP server configured on the target server.

   - If there is not enough space at the source server to compact the data, mount the NFS file system or map the Samba file system at the source machine, and copy the files across the network.
6. Verify the correct files permissions at the target directory. Adjust file permissions after the transfers for production work.

For file storage in an external storage system compatible with Fiber Channel, we can migrate to a Linux on System z server configured with zFCP adapters to connect directly to the volumes that should be migrated to Linux on System z servers.

7.2 Linux on System z - preinstallation considerations

The storage and file system design has a direct influence on system performance, system availability, and the capabilities for system expansion.

A best practice for Linux on System z is that only one version of a Linux OS distribution should be installed from scratch. Therefore, the basic Linux OS file system should be designed to allow the highest possible number of servers, because all other Linux guests in the environment should be cloned from this source (known as the *golden image*). The file system that stores the application data is created after the cloning process.

> **Note:** For information about setting up a golden image for Red Hat Enterprise Linux, refer to *z/VM and Linux on IBM System z*, SG24-7492.
>
> For SUSE Linux, refer to *SAN Volume Controller Best Practices and Performance Guidelines*, SG24-7521.

All file systems, except the root (/) file system, should be created as LVM devices. File systems created with LVM will make it possible to expand or reduce the file without a system outage (using SLES 10 SP2 or higher, or RHEL 5.0 or higher and LVM2).

The file systems could be shared physical between Linux guests from the same or different LPARs, but only one machine can get both read and write control at a time.

Consider using zFCP devices for databases and high I/O subsystems.

7.2.1 Linux file system

As mentioned previously, the basic Linux OS file system should be designed so that one single image (the golden image) can be cloned to be used on as many Linux servers as possible.

At the very least, the golden image should include the following file systems:

- root (/) file system
- /usr file system
- /var file system
- /tmp file system
- /opt file system
- /home file system

In the following sections, we discuss these file systems in more detail.

The root (/) file system

The `root` file system is the first file system to be created, and it is the file system that Linux will be booted (or IPLed) from. It is also the base for all other file systems in the hierarchical structures of the Linux operating system.

This file system should *not* be placed on an LVM device, and it must be formatted using the EXT2 file system format.

The /usr file system

The `/usr` file system is where all Linux standard base applications are installed. The binaries, libraries, and shared files are copied to this directory during the installation process. The file system size depends on the type of server you are running and on the distribution-based packages that need to be installed for the functions that the server provides.

The golden image `/usr` file system size should be the minimum to support the basic Linux distribution files. The ability to increase this file system is necessary because after cloning the server, the system administrator might need to increase the file system to install new packages or additional package dependencies.

This file system should be created on LVM devices (see 7.2.4, “Logical Volume Manager devices” on page 82) that allow you to dynamically extend or reduce the file system size. Using the EXT3 file system is the best option.

In a shared Linux on System z environment, this file system could be set as read-only because the system simply needs to read the application file into memory. This also offers an added security benefit because no one can delete or change any file in a directory mounted as read-only.

The /var file system

The `/var` file system is where all the variables files (such as spool files, cache files, and log files) are written. The `/var` file system has files that are constantly changing such as `/var/log/messages` and `/var/log/secure`.

The size of this file system depends on the number and type of applications that are running and how long the log files will be kept on the server. Also take into consideration whether the application is designed to write files here, as well as their sizes and frequencies.

The services control files are also placed on the /var file system so it could never be scaled to be a shared file system and it must be always be read-write.

Because it is a dynamic file system, it should be placed on an LVM device to allow it to be extended or reduced as needed. Using the EXT3 file system is the best option.

The /tmp file system

The `/tmp` file system was originally designed to store operating system and temporary application files that would be deleted every time that system is rebooted or deleted by the application right after the file is no longer in use. Some homemade applications use the `/tmp` file system as a dump area or an exchange file resource. In rare cases the size of the `/tmp` will need to be increased.

Because it is a dynamic file system, it should be placed on an LVM device to allow the capability to be extended or reduced as needed. Using the EXT3 file system is the best option.

The /opt file system

The `/opt` file system is where all third-party applications should be deployed. As a best practice, the `/opt` directory should be further organized by the company or organization that developed the application or software. The next directory level would be to specify the software package that is installed. For example, an IBM DB2 Universal Database™ server should be installed at `/opt/ibm/db2`. An IBM Web Sphere Application Server should be placed in the `/opt/ibm/WebSphere` directory.

The file system size will depend upon the size of the software packages that will be installed in it. It is easy to estimate the requirements for a single software package. But upgrades, maintenance and additional software packages are not so easy to plan for. Note that the `/opt` file system can also be a dynamic file system and should be configured on an LVM device using the EXT3 file system.

The /home file system

The `/home` file system is design to allocate user files. The size of the file system will depend upon the server function and the number of users defined on the server. For example, application production servers do not need a large `/home` file system because it is not expected that development staff will store files on a production server. On the other hand, it *is* expected that applications will be developed on a development application server, so developers will need sufficient file system space to create and transfer their files.

Depending upon the situation, the `/home` file system could be a dynamic file system. If it is dynamic, it should be configured on an LVM device using the EXT3 file system.

Other file systems

An example of additional file systems that could be created on a specific server during the migration process is the database server file system. Basically, you need to have at least one file system for data files and one for log files. So at a minimum two file systems should be created in addition to the file system where the application binary files would be installed. For an IBM DB2 database server, the default location for the binaries files is `/opt/ibm/DB2`.

Other database management systems put their data files in other directories. For example, the MySql database server default location for data files is the `/var/lib/mysql` directory. If the server is a MySql database server and you are using the Linux distribution from Red Hat Linux or Novell SUSE Linux, then consider including a new file system at the `/var/lib/mysql` mount point.

For each target database management server, make sure that you know where the binary files and the data files will be located, because only then can you plan to create the devices and file systems for the target system.

It is possible that there are file location differences depending upon the distribution of Linux that you install at your site. Make sure that you know these differences, if any, and plan for them.

7.2.2 Shared file system

The data storage in a Linux on System z environment can be shared physically by one or more Linux guests. However, because of limitations of the file system, it is not possible for two Linux guests to have read-write control to a device at the same time, although z/VM allows it at the hardware level.

In a shared DASD environment, keep in mind that the file system changes performed by the guest machine that has the read-write control will only be available to all other guests that share the same file system after unmount and mount of the file system. As an example, think of the environment of a Web cluster service where the application servers only need read access to the Web pages and do not need to write to the same file system where the files are allocated.

In the example shown in Figure 7-2 on page 81, only the special file system and mount points relevant to the solution are represented. The data file location is at mount point `/srv/www/app`. This is the file system that is shared between the Linux guests. There is also the shared file system `/opt/ibm/IBMHTTP`, where the Web server binaries are installed. For the IBMHTTP service, the log files are redirected to the local `/var/log/httpd` file system. All shared devices are the same DASD device type and managed by the z/VM operating system.

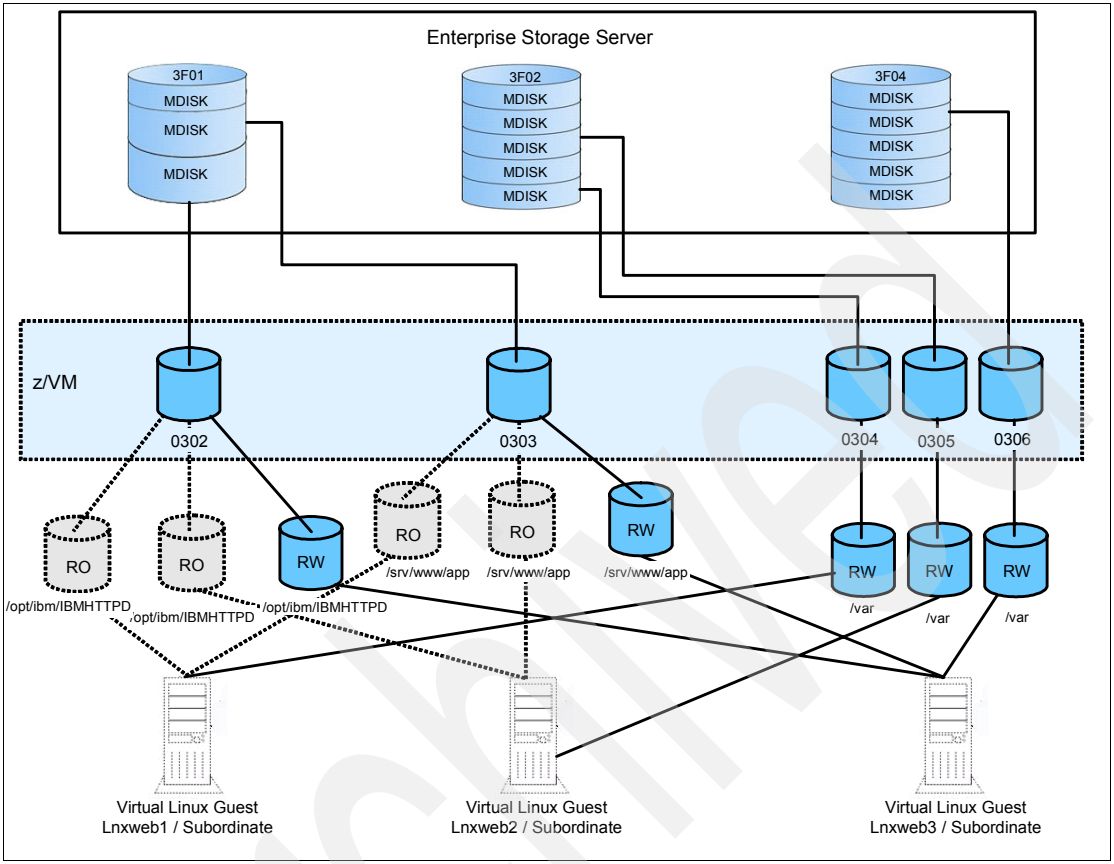

Figure 7-2 Shared device for Web servers

The benefits of using a shared file stem is based on economy of resource. You reduce application binary space allocation and code updating efforts because you only have to update one master server and just remount it on the subordinate servers.

> **Important:** System administrators must pay special attention to managing this kind of environment because if the same file system is mounted as read-write in two different servers, all data can be lost.

7.2.3 ECKD and zFCP devices

ECKD and zFCP devices can be shared by the same Linux guest. This is a common and helpful approach when using large file systems, as in the case of database servers.

The zFCP device, when configured with multiple access channels, provides a better I/O response than a single ECKD channel device. A limitation of zFCP devices is that the only device size that the guest OS can use is the size that was provided by the storage administrator.

After it is configured on Linux on System z, it is possible to split it into partitions like a simple SCSI device using the **FDISK** tool. Even though the sizes of the ECKD volume devices are determined at the storage hardware level, it is still possible to configure smaller volume sizes for the Linux guest when the z/VM system administrator formats the ECKD devices as MDISK devices.

A combination of both solutions can help you improve system performance and use storage resources efficiently.

7.2.4 Logical Volume Manager devices

The Logical Volume Manager (LVM) is very useful for Linux file systems because it allows you to dynamically manage file system size and has tools to help back up and restore failing partitions.

Basically, LVM volumes are composed of the following components:

- Physical volume

  A physical volume (PV) is a storage device, and it can be a DASD device or a SCSI device controlled by a zFCP channel. For Linux on System z, each DASD device is a physical volume. All the disk partitions that are configured under z/VM will be using mini-disk (MDISK) devices.

- Volume group

  A volume group (VG) is the highest level of the LVM unit. A volume group is created by one or more Physical Volumes and gathers together the logical volumes.

- Logical volume

  A logical volume (LV) is the disk partition of the LVM system. This is the area that is formatted and are accessed by users and applications. The LV is exposed through a mount point.

Figure 7-3 shows five MDISK devices that are used by a Linux guest to create a unique VG. It is then further organized or allocated into two LVs.

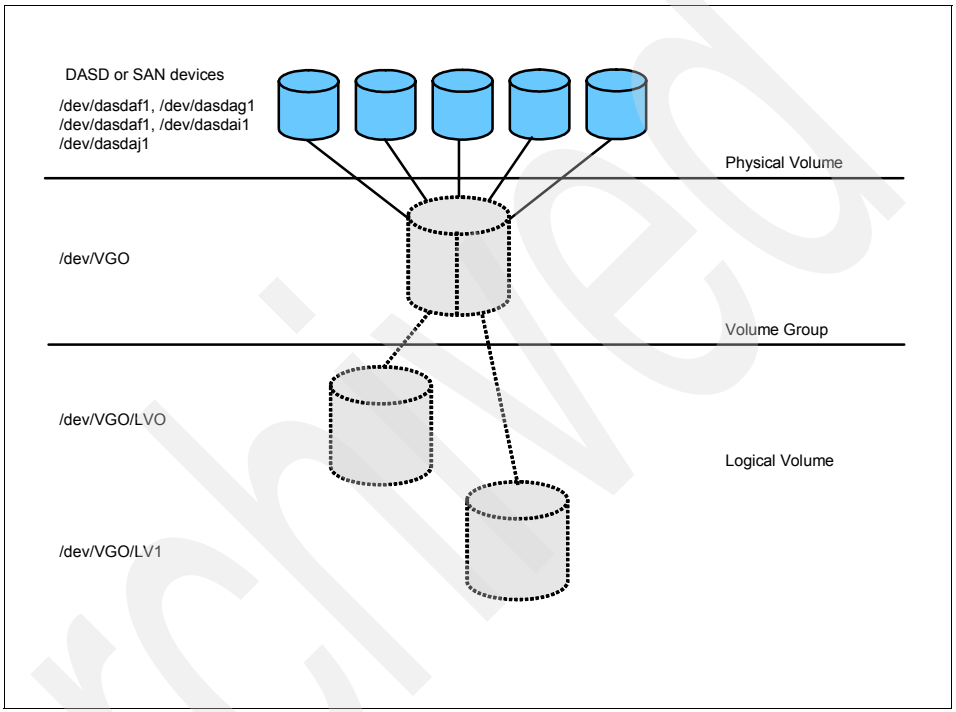

Figure 7-3 A simple LVM example

This facility is primarily used to extend or reduce the file system size of LVM devices. LVM also provides for multiple paths to the Linux guest to read or write the data. This is accomplished when the LVM is configured with striping.

Striped logical volume

The striped logical volume is a feature of LVM services in which the number of blocks on a read/write request is equal to the number of stripes set when the logical volume partition is created.

When creating a striped logical volume, define the same size for each physical device and use the same number of physical devices as the number of stripes.

For the example shown in Figure 7-4, the file system needs 27 GBof space so we create three devices of 9 GB each and use them to create a logical volume with three stripes. In this example, for this file system the Linux kernel can issue three different I/O requests for each application file request.

With LVM striping, we create multiple paths for the I/O process. In the case of z/VM, you should always use different channel addresses for the different devices. This could provide a dramatic performance improvement for an application migrated to Linux on System z.

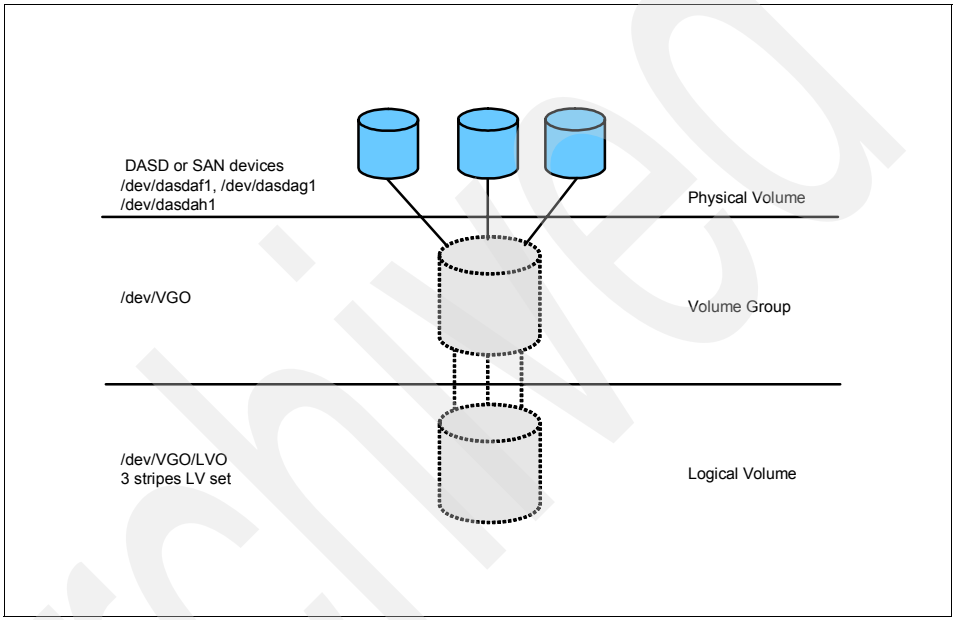

Figure 7-4 LVM - striped example

8

Application analysis

This chapter describes the analysis you need to perform to identify applications that would be good candidates for migrating to Linux on System z.

We discuss the following topics:

- How to identify the best candidates for a migration to Linux on System z
- How to select the appropriate application for a Linux on System z proof of concept
- What you can do if your ISV does not support Linux on System z
- How you can accommodate application interdependencies in a Linux on System z environment
- How you can redesign your application to take advantage of the strengths of the System z platform

8.1 Why to migrate applications

As discussed in Chapter 1, “Migration considerations” on page 3, application migration should only be undertaken after thorough planning. There also has to be a compelling reason to act, such as the following real world situations:

- An existing application has outgrown its original platform and is close to reaching the architectural limits of the platform.
- Software license costs are rapidly increasing as more and more servers are added to an application.
- Performance issues are arising between distributed application servers and centralized databases.
- Uncontrolled distributed server growth is leading to power and cooling issues in the data center.
- Complex distributed systems, which are costly to maintain, are suffering from increasing unreliability.
- New application development is required following a merger or acquisition.
- Regulatory requirements impose the need for a more secure environment.

Such situations present valid reasons for considering a migration to a more efficient platform like IBM System z. In most cases a migration to Linux on System z will help an organization realize significant cost savings over three to five years. The question is, which applications can you migrate and what risk factors are associated with the migration?

The output of this exercise will be a list of an organization’s applications ordered by complexity. The list is based on factors such as the number of servers or applications that make up the “IT systems”, and can generally be grouped as large, medium, or small applications or number of servers.

8.2 Which applications can be migrated

Every computing platform offers specific areas of strength, and the aim of a migration should be to select applications that take advantage of the strengths of the target platform. The classic strengths of IBM System z include high availability, high I/O bandwidth capabilities, the flexibility to run disparate workloads concurrently, and excellent disaster recovery capabilities.

Another key element in choosing the appropriate applications for migration is whether they are supported on Linux on System z. This is normally not a problem

with homegrown applications, depending on what language they were written in, but it could be a significant issue with ISV-supplied applications. There are almost 2,500 Linux on System z applications available.

> **Note:** For more information about applications available for Linux on System z, you can refer to the following URL:
>
> http://www-03.ibm.com/systems/z/os/linux/solutions

8.2.1 Financial benefits of a migration

In most organizations cost reduction is an ongoing challenge, and in many cases an application migration will reduce costs substantially. In a distributed environment there are typically many servers licensed for the same software product and this may provide an opportunity to save software licensing and maintenance costs. Many customers who have consolidated distributed servers to Linux on System z have reported significant software cost savings.

The cost savings arise because Linux on System z is treated by most software vendors as a distributed system and software is usually charged by the core. Because an IFL is usually classified as a single core, there could be significant savings by consolidating multiple distributed servers to an IFL.

To determine the potential software cost savings of a migration to IBM System z, contact your software vendor to understand its policies and pricing regarding application consolidation on System z.

8.3 Selecting an application for migration to Linux on System z

This section lists and describes the basic rules for selecting an application to migrate to Linux on System z.

The list includes applications that cannot or should not be migrated to Linux on System z, and explains why they are unsuitable. The list also includes applications that are suitable for migration.

8.3.1 Applications unsuitable for migration to Linux on System z

- Applications that are available only on Intel or UNIX platforms.
  - Requesting an ISV to support their application on Linux on System z is a long process.
- Servers that have already been virtualized.
  - In such cases, most of the TCO benefits of virtualization have already been realized and only minor benefits will be forthcoming. However, if the existing virtualized environment is reaching its limits or the server leases are within 9 to 12 months of expiry, then there may be a good business case for moving the applications to Linux on System z because of its higher virtualization capabilities.
- Applications with high sustained CPU utilization and high memory needs.
  - Single applications that tend to consume all computer resources are better suited to other platforms.

8.3.2 Applications suitable for migration to Linux on System z

- Applications or middleware (database, application servers, and so on) that are supported by a software vendor on multiple platforms including Linux on IBM System z.
  - There are no support issues and migration is much simpler.
- Applications that need close proximity to data on IBM System z, or that are components of System z applications.
  - You can boost the performance and speed of your Linux on System z applications by putting them on the same physical server as their data source.
- Applications with high I/O or transactional I/O.
  - Because of its design, IBM System z excels at handling sustained high I/O rates.
- Applications with lower sustained CPU peaks and average memory needs.
  - These are ideal workloads for IBM System z. The platform has been designed to run multiple workloads at a consistently high CPU and memory utilization.
- Application development environment for Linux on other platforms.
  - The virtualized Linux on System z platform provides an ideal environment to test applications before their deployment to Linux on other platforms.

8.4 Applications best suited for migration

The applications described in this section leverage the System z platform classic strengths, including high availability, high I/O bandwidth capabilities, the flexibility to run disparate workloads concurrently, and excellent disaster recovery characteristics.

Applications that are used to communicate directly with earlier mainframe applications are able to leverage architectural advantages of the System z platform

8.4.1 IBM software

IBM has ported many of its software products to Linux on System z. The benefit to customers is that a migration from one platform to another is in many cases quite effortless, because many of these products share the same code base across multiple platforms. This is particularly the case for WebSphere Application Server which, since Version 6, has had the same code base on Intel x86, POWER®, and System z; this simplifies migration considerably.

IBM software brands are:

- Information Management
- WebSphere
- Lotus®
- Rational®
- Tivoli®

Note: For a current list of IBM applications and middleware supported on Linux on System z, refer to the following site:

`ftp://ftp.software.ibm.com/software/linux/IBMSoftwareOnLinux.pdf`

Generally, migrating from IBM products on distributed servers to the same IBM products on Linux on System z is a relatively straightforward process. For more details about the migration of WebSphere Application Server and DB2 database see Chapter 14, “MS Windows to Linux - WebSphere and DB2 migration” on page 163.

8.4.2 Oracle

Because the Oracle database management software for Linux on System z is supported by the Oracle Corporation, it is a good candidate for migration to Linux

on System z. As of June 2009, the current GA versions are 10g and 11g. Oracle 9i, 10g, and 10gR2 are currently supported on Linux on System z.

Oracle databases on System z also support Real Application Clusters (RAC), the Oracle high availability clustering solution. The advantages for Oracle RAC on Linux on System z is a high availability cluster with very low latency within the System z platform combined with HiperSockets for inter-LPAR communication.

The Oracle Application Server 10g is also supported on Linux on System z, and it provides the ability to have a complete Oracle Java environment and high availability Oracle database within the same server.

In many cases Oracle supports mixed configuration mode where the database tier sits on Linux on System z and applications for Oracle E-Business Suite, Oracle PeopleSoft, Oracle Siebel, and Oracle Business Intelligence execute on distributed servers under Linux, Windows or UNIX. To obtain the latest information about which Oracle products are certified for Linux on System z, contact your Oracle representative or refer to:

http://www.oracle.com/technology/support/metalink/index.htm

Note: For more information about how to install Oracle 10g, set up a real application cluster, and install Oracle Application Server on IBM System z refer to the following Redbooks:

- *Experiences with Oracle 10gR2 Solutions on Linux for IBM System z*, SG24-7191
- *Experiences with Oracle Solutions on Linux for IBM System z*, SG24-7634
- *Using Oracle Solutions on Linux for System z*, SG24-7573
- *Experiences with Oracle 10g Database for Linux on zSeries*, SG24-6482

For more information about an Oracle 10g migration from Solaris to Linux on System z, refer to “Migrating archived Oracle databases across platforms” on page 118.

8.4.3 Other software

This section lists non-IBM software that are good candidates for migrating to Linux on System z.

SAP

- SAP application servers are candidates for consolidating to Linux on System z, particularly if the SAP database is DB2 on z/OS. The benefits

provided by HiperSockets and virtual networks help to reduce the substantial amounts of network overhead that can occur on physical networks.

Additional benefits include potentially reducing the system administration costs and limiting the environmental impacts from a reduction in distributed servers.

- SAP solutions using a System z framework of z/OS and Linux on System z have been successfully implemented at many IBM System z accounts.

Infrastructure services

- Network infrastructure, FTP, NFS, DNS, and so on are very well served on Linux on System z. These workloads are generally minimal, yet they are critical to the business. The main benefit of hosting these services on Linux on System z is the availability of the hardware's disaster recovery capabilities.

  Additionally, a significant amount of network traffic is generated between data on z/OS and FTP and NFS servers. When the servers are hosted on the same system as the data and HiperSockets is used, then not only is this network traffic greatly reduced, but the batch processing window for that data can also be reduced.

- LDAP security services fit very well running on Linux on System z, including both OPEN LDAP products as well as commercial products like Tivoli Directory Server, Tivoli Directory Integrator, and Tivoli Access Manager. Using System z architecture, clients can build a robust 24x7 LDAP infrastructure.

Application development

- Whether for Java, C/C++, or most other programming languages, a virtualized Linux environment is an ideal platform for application development. Although developers usually develop on a stand-alone platform, testing and modifying are generally performed in a server environment. Developers can be given multiple virtual servers to perform interactive testing while troubleshooting or enhancing the application. z/VM also provides a number of features that enhance application troubleshooting.

- Other major benefits include the ability to rapidly deploy virtual servers for user acceptance testing and integration testing and, when that is finished, the virtual servers are shut down. If a developer inadvertently "damages" a virtual server, then a new server simply has to be cloned; there is no need to spend a great deal of time formatting disks and reinstalling the operating system and required applications.

- For new applications, virtual servers are deployed quickly and can be easily customized for a specific purpose. Many customers have standard server profiles that are pre-built, so to create a new virtual server, the appropriate

profile simply has to be cloned, which can be done in minutes. When an application is discarded for some reason, the virtual servers can be discarded as well.

For more information about using the Linux on System z environment for application development, refer to *Linux on IBM eServer zSeries and S/390: Application Development*, SG24-6807.

To obtain an extensive list of applications and solutions across all industries from over 60 ISVs that are certified and supported on Linux on System z, refer to the following site:

http://www.ibm.com/systems/z/solutions/isv/linuxproduct.html

8.4.4 Selecting an application for a proof of concept

After a business case demonstrates that a Linux on System z migration will provide a positive return on investment (ROI), most clients follow this process:

1. Talk to other customers who have migrated applications to Linux on System z to understand how their migration went and to obtain their recommendations about how to proceed
2. Choose one of their own applications as a candidate for a proof of concept. (POC)

When choosing an application for a PoC it is good practice to keep it as simple as possible, because a proof of concept is performed to demonstrate that an application can be successfully migrated to a Linux on System z environment, and that the application results are the same as the production system.

Select an application that is reasonably self-contained and that does not rely too much on inputs from multiple sources and other applications. Also, choose an application that does not require a major rewrite to run on Linux on System z.

The best candidates are applications which are Java-based because they are generally platform-independent. However, if you are moving to a different J2EE specification and a different application server, you may have to make a number of code changes.

Applications written in C/C++ are also suitable if you have the source code, because they will have to be recompiled for the IBM System z platform.

After you select an application to migrate, you must also define the end objective or success factor. The minimum objective would be to produce results that are identical to the production version.

Chapter 14, “MS Windows to Linux - WebSphere and DB2 migration” on page 163, provides an example PoC for a customer migrating WebSphere Application Server and DB2 from Windows to Linux on System z.

8.4.5 Applications not supported on Linux on System z

If the application chosen for migration is not supported on Linux on System z by the software vendor, you can ask the vendor for this support. However, if the support is agreed to, it will not happen overnight so another application might be a better choice for the migration to Linux on System z.

If an unsupported application is required to work with another application that is supported, then the best option would be to use a hybrid environment where one application is on Linux on System z and the other application remains on its existing (or modernized) platform and communicates with Linux on System z. For example, suppose you have a reporting program running on Solaris that analyzes an Oracle database that is also on Solaris. In this case, the Oracle database could be migrated to Linux on System z and the reporting program could be migrated to IBM BladeCenter® running Solaris 10.

8.4.6 Application interdependencies

Not many applications are self-contained; in most cases an application obtains data from a number of other applications and its output is sent on to other applications. These applications can also be on different platforms and are often from entities outside your organization. An application migration to Linux on System z provides an opportunity to potentially simplify your application without impacting any interdependencies.

Many distributed applications have grown, in only a few years, from a single server to tens or even hundreds of interconnected servers. These interconnected servers not only add network overhead but also add complexity and built-in fragility. If such an application is being considered for migration, then its simplification should be at the core of what needs to be done. System z supports all modern communication methods, so it is a straightforward process to receive data inputs and transmit data outputs in the same way as before the application was migrated. In this case there are no changes to external applications.

The main thing to remember during migration planning is to completely map all application interdependencies. The aim here is to identify any obsolete networking technologies and interfaces, which may in turn require another application to be migrated to a current network technology.

8.5 Successful application migration

This section outlines the considerations to keep in mind as well as the steps to follow to achieve a successful application migration for Java and C/C++ programs.

8.5.1 Special considerations for migrating a Java application

Migrating Java applications from one platform to another is easy compared to the migration effort required for C or C++ applications. Even though Java applications are operating system-independent, there are implementation and distribution specifics to be taken into account, as explained here:

- Because most of the Java distributions have their own Java Virtual Machine (JVM) implementations, there will be differences in the JVM switches. These switches are used to make the JVM and the Java application run as optimally as possible on that platform. Each JVM switch used in the source Java environment needs to verified for a similar switch in the target Java environment.
- Even though Java Developer Kits (JDKs) are expected to conform to common Java specifications, each distribution will have slight differences in the helper classes that provide functionalities to implement specific Java application programming interfaces (APIs). If the application is written to conform to a particular Java distribution, then the helper classes referenced in the application must be changed to refer to the new Java distribution classes.
- There are special procedures to be followed to obtain the best application migration. One critical point is to update the JVM to the current stable version. The compatibility with earlier versions is significant and there are performance improvements that benefit applications.
- Ensure that the Just-In-Time (JIT) compiler is enabled.
- Set the minimal heap size (-Xms) equal to the maximal heap size (-Xmx). The size of the heap size should be always less the total of memory configured to the server. In most cases large heap size implies better performance.

8.5.2 Special considerations for migrating C++ applications

When migrating C++ applications, there a few special considerations to be aware of, as explained in this section.

Architecture-dependent code

Programs residing in directories (on non-S/390 systems) with names like `/sysdeps` or `/arch` contain architecture-dependent code. You will need to reimplement them for the System z architecture to port any of these programs to S/390®.

Assembler code

Any assembler code would need to be rewritten in S/390 Assembler. Opcodes would have to be changed to S/390 opcodes or, if the code uses assembler header files, you would need a S/390 version of the header. S/390 Assembler code for Linux uses the 390 opcodes but follows the syntax conventions of GNU assembler. The GNU assembler manual can be downloaded at:

http://www.gnu.org/manual/gas-2.9.1/as.html

ptrace and return structure

Exercise caution when using ptrace and the return structure because they are architecture-dependent.

Little endian to big endian

S/390 is a big endian system, storing multibyte numbers with the most significant byte at a greater (little endian) or lower (big endian) address. Any code that processes byte-oriented data that originated on a little endian system may need some byte-swapping. The data may have to be regenerated or, if that is not possible (for example, shared files), the application may have to be reworked to adjust for processing little endian data.

Stack frame layout and linkage specific to S/390

For details about stack frame layout and linkage specific to S/390, refer to `/usr/src/linux/Documentation/Debugging390.txt`. The location of this file may vary depending on the distribution. In one instance, it was found in the file `/usr/src/linux-2.2.16.SuSE/Documentation`.

Changes to build scripts

You will need to make appropriate changes or updates to the `Configuration/build/Makefile` scripts or files, and a requirement to add support for the S/390 platform.

/proc file system

The proc file system has some differences:

- `/proc/cpuinfo` format is different.
- `/proc/interrupts` is not implemented.
- `/proc/stat` does not contain INTR information.

Required languages and compilers

Currently C/C++, Perl, Tcl, Python, Scheme, Regina (REXX), the IBM Java JDK available in WebSphere, and other JDKs are available as open source.

Middleware, libraries and databases

Any middleware or libraries that are needed must be available on Linux for System z. Supported databases include examples of MySQL, Postgres, DB2 UDB, and DB2 Connect™. Middleware dependencies such as WebSphere 3.5, MQ Client, Tivoli, and Apache should be evaluated.

Shared objects

Linux currently does not support shared objects like mutexes, semaphores, and conditional variables across different processes.

8.5.3 Steps for an application migration

A successful application migration depends on the combined efforts of the developer team, the network team, the middleware administrator team, and the Linux on System z team. Without the cooperation of all these groups, it is very difficult to achieve a successful migration.

Follow these steps for a successful migration:

1. Perform source application mapping.

   Start by analyzing the source application, focusing on its suitability to migrate. Keep in mind the following points:

   a. Is the source code available to be compiled and deployed on the target server?

   b. Is there a version of the middleware available for Linux on System z?

   c. Are there performance reports of actual development tests to compare with after the migration?

2. Design the network solution for the application (see Chapter 6, “Network analysis” on page 57 for more information).

3. Design the file system for the application and middleware (see Chapter 7, “Storage analysis” on page 71 for more information).

4. Clone the Linux on System z server (or servers) from the golden image.

5. Configure the network at the target server (or servers).

6. Create the custom file system at the target server (or servers).

7. Install and configure the middleware at the target server.

8. Copy the application code from the source to the target server.

9. Compile and deploy the application code to the target server.
10. Provide the first application test reports.
11. Start the performance test on the target server to understand the performance of the migrated application.
12. Size the CPU and memory to fit the migration expectations.
13. Execute the application stress test.

After all tests have been completed and approvals granted:

14. Shut down the source server.
15. Change the IP address and host name of the target server, or change the DNS configuration to the target application server.

9

Database analysis

This chapter provides information about the configurations of the database application server on Linux on System z. Best practices for different database software are also presented.

9.1 Prior to database migration

The database server is one of the most highly recommended services to be migrated to Linux on System z. However, it also demands detailed planning because there are technical configuration changes to be considered.

During the migration planning discussions, the workload of the instances and the databases that are running at the source environment must be considered, along with the number of concurrent users and the number of instances and databases running in a unique source server.

9.1.1 Migrating a single instance

For single instance servers, migration is fairly simple because the number of the variables from the source environment to the new destination environment is relatively small. You can use the following steps to migrate when using the same database software vendor and version:

1. Configure the Linux on System z network (follow steps 1 to 4 as listed in 6.2, “Steps for a successful network migration” on page 69).
2. Configure the temporary storage area at the source server and at the destination server.
3. Stop the database services.
4. Issue the export/dump procedures at the source server.
5. Transfer the export/dump files through the network to the destination Linux on System z server.
6. Shut down the source server.
7. Change the Linux on System z server host name and IP address.
8. Perform import procedures at the destination server.
9. Perform the database and applications tests.

9.1.2 Migrating multiple instances

For a multiple instance on a single server, or multiple instances on multiple servers, migration is more detailed and complicated. However, among the benefits of the migration are lower license cost, less data center space needed, energy savings, and better performance.

Migrating multiple servers to Linux on System z

A significant factor in the migration of multiple servers to Linux on System z is the distribution of server peak load. Document and compare peak workload information, including how long the workloads take and how much server resource is used. You can use Table 9-1 to map server workloads when creating the migration configurations.

Table 9-1 Sample server workload map

Server information			Peak load measure		Peak load time		
Server name	**Total of CPU**	**Total of Memory**	**% CPU used**	**% Memory used**	**Weekday**	**Start hour**	**Execution time**

As explained in Chapter 5, “z/VM and Linux on System z technical concepts” on page 49, the CPU and memory constraints in an LPAR are possible and desirable, but the server should maintain the same peak load for a long period of time if there are not real CPUs to process each virtual CPU request.

For example, consider a configuration of one LPAR set with three real dedicated CPUs and running three Linux guests. LinuxA has two virtual CPUs, LinuxB has two virtual CPUs, and LinuxC has one virtual CPU.

If LinuxA and LinuxB servers have the same peak load time and period and during this peak load, both LinuxA and LinuxB use 100% of the CPU, that will cause a CPU constraint because the number of virtual CPUs is four and the number of real CPUs is three.

In this case, the z/VM share algorithm will handle all the processor requests and the server still would be available. However, the performance of the application would probably not be very good and would also affect LinuxC’s response time. However, if the server peak loads of LinuxA and LinuxB occur at different times, then the entire LPAR will not be affected.

This kind of constraint is acceptable if it happens in intervals of milliseconds to seconds, but it can become problematic in intervals that last for more than a few minutes, depending on how critical the server is to the business purpose.

Having the correct workload capacity plan is key to successfully migrating multiple database servers in a single LPAR on System z.

Another point to consider regarding CPU capacity is the relationship between the source server and the migration server; it is not 1:1. In other words, 1 distributed server with 4 CPUs must not necessarily have 4 CPUs in the destination virtual server; best practice shows that the actual number is less than that. For more information about this topic, refer to 9.2.1, “CPU” on page 103 in the 9.2, “Technical considerations” on page 103.

Migrating multiple instance servers to Linux on System z

The benefits resulting from migrating a unique server running several database instances include simplified database management and reduced cost. For best results from this type of migration, the workload analysis should be very detailed. Different instances have different workload types, times, and characteristics that might allow the overcommitment of CPUs and memory.

In an environment where the instances are divided among various virtual servers, a software problem occurring on a specific instance will affect only the database server where the instance is running, so only the database server where the instance is running would need to be rebooted.

It is possible to reduce the number of CPUs allocated to an LPAR by using IFLs. This would result in software licensing savings.

To minimize the work related to database software fixes and security updates, it is possible to use shared devices for database binaries and libraries. For more information about these topics, refer to 7.2.2, “Shared file system” on page 80 and to Chapter 7, “Storage analysis” on page 71.

Consider the following questions when you migrate from a multiple instance server to multiple Linux on System z virtual servers:

- Is the source server running at maximal CPU capacity?
- Is the use of the CPU balanced across all instances? Or is there a unique instance that is consuming all of the CPU?
- What is the average CPU cycles used by each instance?
- During which period does the instance use more CPU cycles?
- Does the instance write or read more data onto the disk devices?
- How much memory does each instance have allocated?

You can use Table 9-1 on page 101 also to map the instances used by simply changing the Server name column to Instance name and documenting the appropriate information.

With this information you can configure multiple servers in an LPAR to respond to all user requests, without degraded performance and with improved database management. It will be easy to define the number of virtual CPUs that each server needs and avoid the constraint of real CPU in peak usage hours.

Tip: If possible, gather data for an entire month instead for a single day. The more data you have, the more accurate your analysis will be.

9.2 Technical considerations

Database management software requires particularly careful analysis when you are planning a migration. Most database servers use shared memory segments and semaphores to process communications. The database application also uses buffer page configuration to speed up table access and the overall application. In other words, database servers are memory-bound and storage-bound and table access should be considered at server migration.

9.2.1 CPU

The number of virtual CPU resources in a database server is very important; setting the maximum possible does not guarantee better performance. The number of CPUs should be large enough to avoid the processor queue. The processor queue value is considered high when it is greater than 5 per processor in an interval of 30 seconds.

The number of processes in a processor queue is influenced by all the other resources of the server, and should not be analyzed as a separate resource. Memory constraints or I/O constraints affect the processor queue number directly, so before deciding that the server does not have enough CPU and adding a new CPU to the service, analyze the CPU schedule time. If the system is running in a high processor queue and most of the CPU time is dedicated to SYSTEM, it probably is associated with memory. The correct parameter to resize is the memory size. Similarly, if the CPU time is dedicated to I/O WAIT, the file system should be reorganized.

The minimum number of virtual CPUs that should be allocated to a database server is two virtual CPUs. In most cases, the number of CPUs of the source server minus two is a useful value to start the database validation. When

validating the server migration, the number of CPUs can be increased with a server reboot. Best practice is start with the minimum number of CPUs as described and increase as needed.

> **Note:** Linux on System z provides a daemon (`cpuplugd`) that automatically starts and stops virtual processors based on virtual processor utilization and workload characteristics, thereby exploiting z/VM V5.4 share redistribution. The `cpuplugd` daemon is available with SUSE Linux Enterprise Server (SLES) 10 SP2. IBM is working with its Linux distributor partners to provide this function in other Linux on System z distributions.

9.2.2 Memory

The database server uses a very large memory area to achieve acceptable performance, but with Linux on System z, allocating more resources is not related to improving performance. Instead, the machine should be sized as needed and one consideration involved is the server paging process.

Keep in mind that a huge memory setting in the server is not desirable, so at the start of the migration, start the Linux memory size with 60% of the total memory sized from the source server and then increase or decrease as needed.

Swap memory

Specifically in database servers, the swap areashould exist and count as part of the total usable memory. However, it should be used only at the peak size to avoid the Linux kernel killing the database process because of memory constraint.

A System z best practice is to use the V-disk devices as swap devices. Because swap configured at V-disk devices provides desirable response time, the eventual memory paging (the process that moves memory blocks to and from real memory to and from swap memory) is not considered a real problem. It is also not considered a problem if the server has more than 50% of the swap memory allocated. However, this points to variable paging and swapping allocation, which must be monitored to avoid database outages. For more information about Linux on System z monitoring, refer to Appendix C, “Performance measurement” on page 263.

If the server shows a very high paging value for more than 5 minutes, then increase memory at the server and continue monitoring the server to find the best memory size.

The Linux server uses the swap memory to allocate memory pages that are not used in real memory as its default configuration. However, that is not the most

desirable solution when considering database servers. In fact, it is best to avoid this type of situation. There is a configurable kernel parameter called `swappiness` that determines whether more or fewer pages will be swapped; see Example 9-1.

Example 9-1 /proc/sys/vm/swappiness

```
at /etc/sysctl.conf file include the line
vm.swappiness = 0
```

The configuration displayed in the example will not avoid Linux swapping, but it will reduce the amount of swapping.

The second configuration regarding the swap pages is the `page-cluster` kernel parameters that control the number of pages that will be written at the swap in a single attempt; see Example 9-2. The default value is 8 pages at a time. Changing this value to a smaller value will reduce the paging time.

Example 9-2 /proc/sys/vm/pge-cluster

```
at /etc/sysctl.conf file include the line
vm.page-cluster = 1
```

The correct swap size must be, at most, 20% of the total memory considering the limit is 2 G of swap memory. Monitor swap when the stress test occurs and increase or decrease it accordingly to improve performance.

Shared memory

Linux systems use the interprocessor communication (IPC) facility for efficient communication of process with no kernel intervention. The IPC uses three resources to communicate: messages queues, semaphores, and shared memory.

Shared memory is a memory segment that is shared by more than one process. The size of the shared memory directly influences database performance because if the database can allocate more objects in real memory, the system will perform less I/O.

To obtain the best memory allocation, you must to set some Linux kernel parameters and these parameters depend on what the DBA allocated in the migration. As shown in Example 9-3, the server was set to use 60% (4.8 G) of the 8 G of real memory allocated to the server as shared memory.

Example 9-3 Shared memory configuration

```
at /etc/sysctl.conf file include the follow lines :
kernel.shmmax = 5153960755
```

```
kernel.shmall = 1258291
kernel.shmmni = 4096
```

The IPC messages queue allow interprocess communication issuing message writing and reading according to process schedules. Example 9-4 illustrates setting the message queue values.

Example 9-4 Messages queue configuration

```
at /etc/sysctl.conf file include the follow lines :
kernel.msgmni = 1024
kernel.msgmax = 65536
kernel.msgmnb =65536
```

9.2.3 Storage

Data storage access on a database server is intensive and needs to be considered during server migration. To take advantage of the System z SAP I/O processor, the first consideration in design is to spread the I/O workload over as many paths as possible of the storage server.

In the FICON/ECKD devices, consider using the hyperPAV solution for Linux on System z and a path group with FICON channels. A zFCP solution provides multipath access to the storage device.

In a multipath system using ECKD or SCSI devices, the best solution is to use small devices grouped as a Striped Logical Volume (see 7.2.4, “Logical Volume Manager devices” on page 82 for more details). Such a configuration will provide better I/O bandwidth on a system where there are intensive writes and sequential reads.

Chapter 5, “z/VM and Linux on System z technical concepts” on page 49, describes how disk device accesses are made and explains how an external storage system provides its own disk page caching. If such functionality is not utilized, then the Linux OS will spend CPU cycles with disk page caching.

9.3 Tips for successful migration

Almost all database servers use buffer pools in the shared memory area to manage the database memory context. Avoid using any automatic memory management systems to allocate shared memory. For example, if there is 6 GB

of shared memory to be allocated to the database application, then force the database application to allocate all memory at the system start.

If the database server is not using all server memory, try to reduce the server memory until the paging process occurs. The first result that indicates insufficient memory size for the Linux servers is swap paging.

If the server for any reason is showing a processor queue, then add more virtual CPU to the server. However, monitor the entire LPAR workload to avoid having the performance of a Linux guest interfere with another Linux guest.

The data files and log files must be in different file systems and should be striped across the storage hardware. There should also be multiple paths to the data to ensure availability.

The Linux administrator and database administrator must work together in the Linux guest sizing process because changes may be need at both the Linux and database levels.

9.4 Database migration tools

If a database vendor will not support their product on Linux on IBM System z and the database must be run on Linux on System z, one option is to use the IBM Migration Toolkit to migrate the source database to IBM DB2 Enterprise Server Versions 8.1, 8.2, 9, and 9.5 or to IBM Informix® Dynamic Server Versions 10 or 11.10.

Figure 9-1 on page 108 displays the source and target databases supported by the IBM Migration Toolkit.

Target DBMS	Source DBMS				
	Oracle Versions 8i,9i & 10g	**Sybase Adaptive Server Enterprise (ASE) versions 11, 12, 12.5 and 15**	**Sybase SQL Anywhere (ASA) version 9**	**Microsoft SQL Server versions 7, 2000 and 2005**	**MySQL versions 4 and 5**
DB2 Database for Linux, UNIX & Windows: versions 8.1,8.2, 9 and 9.5	✓	✓	✓	✓	✓
DB2 for i5/OS, V5R2, V5R3 and V5R4	✓	✓		✓	✓
DB2 for z/OS Version 8	✓				
DB2 for z/OS Version 9	✓	✓			
Informix Dynamic Server Version 10	✓		✓		✓
Informix Dynamic Server Version 11.10	✓		✓	✓	✓

Figure 9-1 Database migration options

The IBM Migration Toolkit converts DDL statements, CREATE statements, ALTER TABLE statements, SQL statements, triggers, procedures and functions from supported source database platforms to DB2 format. The supported source database platforms are:

- Sybase Adaptive Server Enterprise (ASE) 11,12, 12.5 and 15
- Sybase SQL Anywhere Version 9
- Microsoft SQL Server 7, 2000 and 2005
- Oracle 8, 9 & 10g
- MySQL versions 4 & 5

More detailed information can be found at:

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

10

Backup analysis

This chapter provides a conceptual approach to migrating backed-up data from an existing operating environment to the target Linux on System z environment.

10.1 Introduction to backup and archival concepts

This section gives a high level introduction to the basic data and storage management paradigms used widely in the IT Industry. We cover data protection or backup, record retention or archiving, storage management, and security.

10.1.1 Backup concepts

The term *backup* refers to the creation of an additional copy of a data object to be used for operational recovery. As already mentioned, the selection of data objects to be backed up needs to be done carefully to ensure that, when restored, the data is still usable.

A data object can be a file, a part of a file, a directory, or a user-defined data object like a database table. Potentially, you can make several backup versions of the data, each version at a different point in time. These versions are closely tied together and related to the original object as a group of backups. The files are backed up via normal daily backup operations each day that it changes. The most recently backed-up file version is designated the “active” backup. All other versions are “inactive” backups.

If the original data object is corrupted or lost on the client system, *restore* is the process of recovering typically the most current version of the backed-up data. The number and retention period of backup versions is controlled by backup policy definitions.

Old versions are automatically deleted as new versions are created, either when:

- The number of versions stored exceeds the defined limit
- Or, after a defined period of time

10.1.2 Common backup types

There are several types of common backups:

- Normal
- Incremental
- Daily

A *normal* backup copies all selected files and marks each as having been backed up. With normal backups, you need only the most recent copy of the backup file to restore all of the files.

An *incremental* backup backs up only those files created or changed since the last normal or incremental backup. It marks files as having been backed up. If you use a combination of normal and incremental backups, you need the last normal backup set as well as all the incremental backup sets to restore your data.

A *daily* backup copies all selected files that have been modified on the day that the daily backup is performed. The backed-up files are not marked as having been backed up.

10.1.3 Archiving concepts

Archiving means creating a copy of a file as a separate object in the storage repository to be retained for a specific period of time. Typically you would use this function to create an additional copy of data to be saved for historical purposes; for this reason, give special consideration to this task to ensure that the data format is not dependent on anything. Vital records (data that must be kept due to government regulation, compliance, legal or other business reasons) are likely candidates for the archive process.

You can specify to delete the original copy of the data on the source system after the archive copy is created on the server. In this way, you can use an archive to make additional space available on the storage medium. However, archive should not be thought of as a complete space management function, because transparent automatic recall is not available.

Therefore, the difference between backup and archive software is that backup creates and controls multiple backup versions that are directly attached to the original client file, whereas archive creates an additional stored object that is normally kept for a specific period of time, as in the case of vital records.

10.1.4 Backup and archival tools

There are a variety of methods of performing backups with Linux on System z. These include command-line tools included with every Linux distribution, such as **dd**, **dump**, **cpio**, as well as **tar**. Also available are text-based utilities, such as **Amanda** which is designed to add a more user-friendly interface to the backup and restore procedures. Finally, commercial backup utilities are also available, such as the IBM Tivoli Storage Manager. Any one of these backup solutions can provide protection for your valuable data.

10.2 Migrating backed-up and archived data

When moving to an newer or more modern environment, the archived data in the existing environment may no longer be supported, depending on the storage technologies used. It becomes necessary to migrate archived data to newer, more current formats. This ensures compatibility with the production IT environment and maintains data availability.

10.2.1 Why to migrate archived data

Factors that force the migration of archived data include:

- Preserving data on the same medium where it was originally acquired would face the dual problem of the lifetime of the medium itself and of the long-term availability of the technology for reading it.
- Over time, technologies age and eventually become less competitive and less economical to run compared to emerging ones. In some cases, that point alone may render the technology change very attractive.
- Some older storage technologies have a direct impact on the volume of data that can be stored as well as the space requirements due to the low MBytes/cm3 and corresponding high Weight/MByte factors.
- Unlike the IBM mainframe (which has been supported for the last 40 years), the disappearance of a technology is often driven by market forces as well as by the difficulty in procuring related media and in obtaining support of the withdrawn operating environments (servers) by the manufacturers.

10.3 General archival migration considerations

There are multiple ways of migrating data from the existing operating environment to another operating environment:

- Change in the hardware environment
- Change in the hardware and software environment

10.3.1 Change in the hardware environment

This scenario applies when the hardware (servers and storage devices) is to be replaced by newer and more efficient hardware environments.

Sometimes change in the hardware environment leads to a change of storage technology, which means reorganizing the media data content. Therefore, to

allow efficient data retrieval the data inventory structures may need to be reconverted.

Because the operating system and the backup and archival management tools are going to be retained or upgraded, there would not be any incompatibility issues with the archived data. This also means that the migration would be relatively straightforward because the storage backup and archival manager product would be able to access the existing archived data.

Often backup and archival managers have built-in migration tools that will migrate the archived data from the source operating environment to the target new environment. This is a useful point at which to reorganize the archives and purge unwanted data, to efficiently reduce the storage needs of the archives.

10.3.2 Change in the hardware and software environment

This scenario applies when the IT department decides to move to a totally new operating environment (both hardware and software). In this case both the hardware and software technologies would be replaced. The hardware would have a highly efficient virtualization server and the software would have new technologies that are either proprietary or open source.

In this case, the migration of the data from the existing archives would be somewhat time-consuming due to the following factors:

- Incompatible file system in the current and the target environment.

  Even though most of the UNIX distributions use similar file system structures, there are differences in the way the file systems are organized and how data is stored.

- Software vendor incompatibility.

  Often data that is archived using one vendor's backup and migration manager software can only be accessed or restored by the same vendor software. The backup and archival software of other software vendors may not be able to access or restore the archived data.

- The format and other characteristics of the archive would be only understandable to that software.

  In such cases, the data needs to be re-archived and re-labeled for restoring.

- Database raw file backups.

  In such cases, the database itself is required to restore the database and re-back it up using the new backup and archival software.

- Archived data would be accessed by a variety of applications for a number of purposes.

  In such cases, if there is a change in the archival format, these interconnected applications would need to be redesigned to access the new archival data.

- Ensuring the data integrity (no data loss or degradation) implies verification of the data after conversion.

  This task will require basic control in screening and a comparison of the new data being archived with the same data in the old archives.

10.4 Approaches to migrating archived data

In this section we discuss the migration approaches you can employ when there is a change in the target environment's software stack.

10.4.1 Restoring to a temporary staging server and re-archiving

In this approach, all archived data must be restored to the staging server with the same operating system environment. Then, if the new backup and archival manager is compatible with the existing operating system, it would be used to back up the restored data from the staging server to Linux on System z.

This approach may be time-consuming because all archived information needs to be restored and then re-archived and labeled by the new backup and archival manager with similar policy attributes such as date, time, and so on.

10.4.2 Restoring to a target file system and re-archiving

In this approach, the archived data is restored from the old system to the new Linux on System z server. This means that the old backup and archive managing software would restore and export the archived data to new file system partitions. Then exported archive data can be re-archived into the target archiving system. Here the current or old archival software needs to assessed for features that allow the product to export archived data to other file systems. The need for maintaining staging servers is reduced.

By incorporating this step it may also be possible to automate data archiving, retrieval, and retranscription by using native scripting as well as features in the software products.

10.4.3 Leaving the archived data on the secondary server

In this approach, you use a scaled-down capacity old system for archival restores, and divert all backups to the new operating environment. Eventually the data from the old operating environment will expire and fall away.

This approach requires that your new target operating implementation does not need your old storage unit. But care can be taken so that whenever an archived data restore occurs, it would be re-archived through the new software and at the same time deleted from the old archival environment. In this way the space requirements would gradually reduce within the old archival environment and be reallocated to the new archival environment.

10.5 A restore and re-archive scenario

Figure 10-1 on page 116 illustrates an example of information from one software product being restored to a staging file system and then archived using another software product, such as the IBM Tivoli Storage Manager.

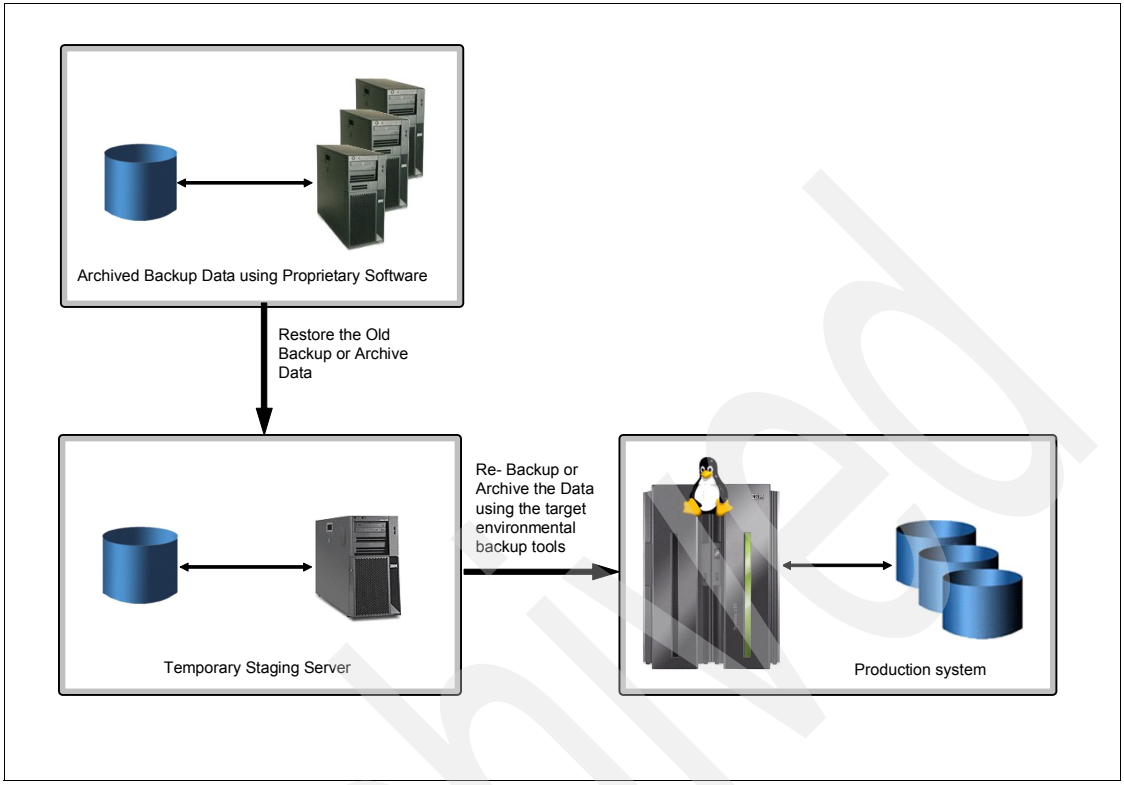

Figure 10-1 Using a staging server to migrate existing backups and archives to new technology

10.5.1 Application logs and redo logs

To migrate application and redo logs, the following steps are involved:

1. From the existing archival software, the archived data needs to be restored to a staging server with the same operating environment. This staging server can be a partition in the same operating environment as well.
2. The new server running Linux on system z, which is also being used for the current backups and archives, would be connected to the staging server for accessing the already restored logs (as per the previous steps).
3. The new backup/archival software connects to the staging server, accesses the restored data and re-archives it according to defined organizational attributes and backup policies.

10.6 Database raw file archives

In the case of a database being involved, the migration approach would depend on the following scenarios:

- Database software retained in target environment
- Database software changed in target environment

10.6.1 Database software retained in target environment

If the same database software is retained in the target environment, then the migration would only re-archive the data into database files using the target backup and archival software.

The following generic steps could be used for this task:

1. Using the source backup and archival software, restore the archived files to the staging server.
2. In the staging server, the restored archived data needs to be imported and then exported to a common format supported by the database, such as comma-separated value (CSV).
3. The staging server, accessed by the target backup and archive software then re-archives the restored file (step 1).

Migrating archived DB2 databases across platforms

Even though DB2 has many different ways of migrating the archived data from one operating environment to the target, the simplest and most flexible way of migrating the data is by using the **DB2MOVE command** with the **INSERT** or **LOAD** parameter.

There are four file formats supported for import and export. The format chosen usually reflects the source it comes from or the target tools to be used. Usually the extension of files such as `.ixf`, `.del` or `.asc` reveal the content format. For example, a file named `employee.ixf` will contain uneditable DB2 UDB interchange format data. Import has the ability to traverse the hierarchy of typed tables in `.ixf` format.

The following steps present a general overview of how to move an archived database between platforms.

1. Restore the database archive file into the source DB2 database.
2. Use the export utility to export the database in to any of the file formats supported by DB2
3. Import the exported file to the target environment.

4. Using the new backup and archival manager product in the target environment, perform a full backup/archival of your new database.

> **Note:** For the actual steps of the DB2 migration commands, refer to 14.3, “DB2 database migration” on page 167.

Migrating archived Oracle databases across platforms

The Export and Import utilities are the only methods that Oracle supports for moving an existing Oracle database from one hardware platform to another.

The following steps present a general overview of how to move a archived database between platforms.

1. Restore the database archive file into the source oracle database.
2. As a DBA user, issue the SQL query shown here to get the exact name of all tablespaces. You will need this information later in the process.

```
SELECT tablespace_name FROM dba_tablespaces;
```

3. As a DBA user, perform a full export from the source database, as shown:

```
exp <database name> FULL=y FILE=oradbtst.dmp
```

4. Move the dump file to the target database server. If you use FTP, be sure to copy it in binary format (by entering binary at the FTP prompt) to avoid file corruption.
5. Create a database on the target server. Then, using the DDL Scripts, create the respective tables, indexes and so on.

> **Important:** Before importing the dump file, you must first create your tablespaces, using the information obtained in step 1of this list.
>
> Otherwise, the import will create the corresponding datafiles in the same file structure as at the source database, which may not be compatible with the file structure on the target system.

6. As a DBA user, perform a full import with the **IGNORE** parameter enabled:

```
imp <database name> FULL=y IGNORE=y FILE=oradbtst.dmp
```

Using **IGNORE=y** instructs Oracle to ignore any creation errors during the import and permit the import to complete.

7. Using the new backup and archival manager product, perform a full backup and archival of your new database.

10.6.2 Database software changed in target environment

If the database software is going to change in the target environment, migration needs will be dependent on which database software you are migrating from and to. Contact your database management team so they can specify specific steps that must be taken for a successful migration.

11

Security analysis

This chapter provides an overview of the security considerations you need to include in analyzing programs and functions that are going to be part of the migration. Available enterprise-wide authentication options and their possible role in migration is also discussed. Finally, because SSL/SSH is probably going to be used, we explain the use of the cryptography hardware that is available.

This chapter discusses the following topics:

- Security migration overview
- Code and application analysis
- Availability and accountability
- Data integrity, assurance, and confidentiality
- Security change management
- Enterprise authentication options
- Integrated Cryptographic Service Facility (ICSF)

11.1 Security migration overview

You might assume that simply migrating an application from its existing server to the target Linux on System z server would mean that the security would remain the same. Although that could happen, it probably will not be the case. A major benefit of migrating to z/VM is access to enterprise-class security. Thus, the best time to plan for and take advantage of this benefit is during the migration process.

The security analysis will center around the following areas:

- Code and application analysis
- Availability and accountability analysis
- Data integrity and confidentiality analysis
- Change and recovery management

11.1.1 Basics of security

Overall security is comprised of three domains:

- Physical security
- System security
- Network security

In each domain the concept of “principle of least privilege” is applied which results in the security policy. That is where each individual is only granted the access that they need, no more. You will need to establish individuals and their roles and who is going to be allowed to do what. This is vital for overall system security because if a compromise occurs, its exposure will only be to the affected role.

Use mandatory access controls to not only ensure that privileged access is given to only what is needed, but to also ensure that authorization is withdrawn when privileges are revoked.

A basic premise underlying the concept of security is that you are only as strong as your weakest point. That is why security is very time-consuming, and it is difficult to predict the amount of time that analysis will take. If this is the first time that you are undertaking a security analysis, do not underestimate the time or scope involved in this task.

It is generally held that “security through obscurity” is not a valid method. Using open, well-established security methods implemented correctly provides the best defense. For example, instead of developing your own cryptographic libraries, you should instead use open, established ones that have been vetted for many

years. Hiding information creates more system administration work and any mistakes may fail to protect against attacks.

System logs, as well as application logs, need to be immutable. Logs must be kept in such a way that they cannot be altered by system users. If logs can be altered then overall system integrity will be in question if an impropriety is suspected. Thus it is paramount that all logs be kept in a way that makes them a permanent record of what occurred on the system.

Document the system security and all the assumptions made. Include all “what if” situations that may reasonably be expected to occur. Also, document security plans such as change control, audits, and procedures for break-ins in all domains.

11.1.2 Understanding the z/VM foundation

The Linux virtual machine (VM) is controlled at the z/VM layer. Thus, for a complete security survey to be done, you need both access and an understanding of its security.

The VM layer allows for many Linux images or other operating systems (like z/OS) to run on the same hardware at the same time. The z/VM layer allows for resources to be shared between each VM. It also allows for virtual devices to be created and consumed, like HiperSockets. The highest priority user ID on the z/VM system is MAINT. The MAINT user has root authority and as such must be secured.

System z and existing security policies

Most organizations have an existing security policy dictating that the mainframe must not be Internet-facing. With the migration of a distributed environment to Linux on System z, this often raises questions concerning the role of System z within the existing security policy. A useful approach regarding security policies is to conform with the existing policy as much as possible because it simplifies the migration process. Although usually z/OS is never directly connected to the Internet, this may be a requirement for a distributed environment running on Linux under z/VM in the same System z footprint.

Processor Resource/System Manager (PR/SM) has been certified through the Common Criteria at Evaluation Acceptance Level (EAL) 5. This is the highest rating for commercially available IT products and among virtualization hipervisors. Only PR/SM on System z has achieved this rating. EAL5 provides a high assurance level that logical partitions provide the same isolation as air-gapped systems. For more details about Common Criteria, refer to 1.3.1, “System z strengths” on page 10.

To further ensure the isolation of the z/VM LPAR from the z/OS LPAR, the Open Systems Adapters (OSA) used to connect to external networks by z/VM should be dedicated to the z/VM LPAR. These precautions will ensure that the z/OS environment remains isolated from the Internet. However, if the security policy states that nothing on the mainframe can be connected to the Internet, then you have the option of putting the Web servers on x86 servers with a physical firewall between the Web servers and z/VM.

Firewalls and existing security policies

In many cases an organization's existing security policy will identify specific firewalls that have been approved for use on the corporate network. Most often these are hardware firewall appliances. Although z/VM can provide a virtual network between the virtual Linux servers, there is often a requirement to have a firewall between distributed servers such as an application server talking to a database server. In a distributed environment, the firewall is in the communication path.

For z/VM there are two options. The first is to implement a software firewall on a virtual server within the virtual Linux environment.This has some challenges because the firewall software may not be used in the organization and as such would have to be certified, which could be a long and complicated process.

The second option is to continue to use the physical firewalls by having the intersecurity level communication exit the virtual environment via an Open Systems Adapter (OSA), go through the physical firewall, and then return to the virtual environment via a different OSA. Figure 11-1 on page 125 illustrates the use of an external firewall.

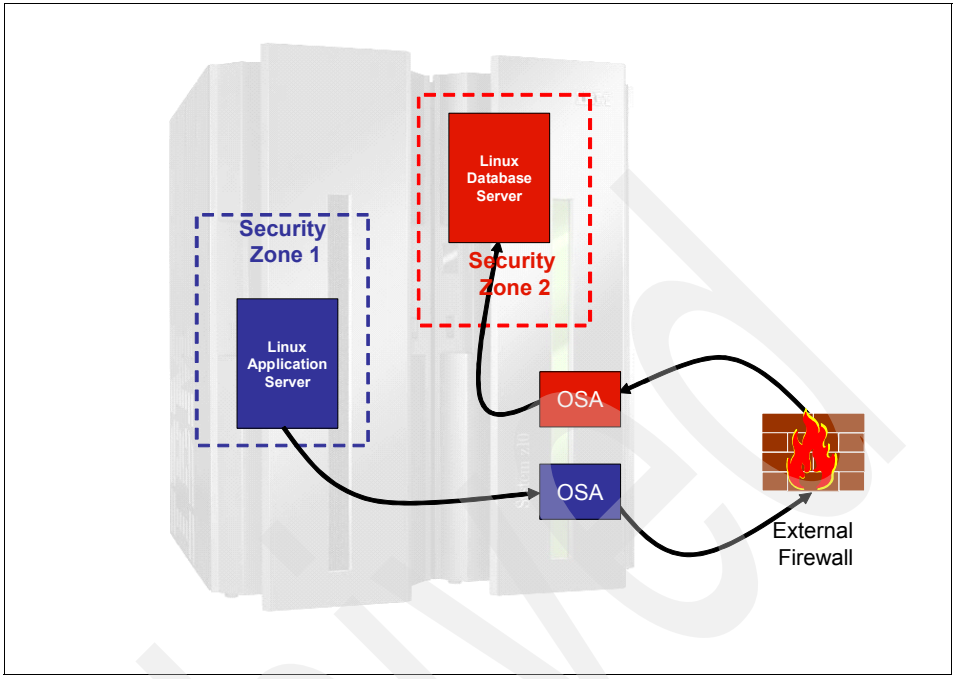

Figure 11-1 Using external firewalls between security zones

In Figure 11-1, the different security zones shown could be in separate LPARs or in the same LPAR. Customers have reported that there is minimal performance impact when using external firewalls.

As mentioned, conforming to the existing security policy can simplify a migration. However, the reality is that for applications within the System z footprint, as shown in Figure 11-1, there may be no requirement for firewalls if all incoming communications to System z are processed by external firewalls.

Control of z/VM

Who will own the z/VM, and what is the protocol for requesting changes or actions? If you will control the z/VM, then you need to fully understand z/VM because it is the basis for all the VMs. It must be secure and its access should be highly controlled. Also, a change request protocol should be documented and published to all stakeholders.

You also need to plan for z/VM maintenance, which may require that some or all of the VMs be quiesced. So ensure that a change window is set aside to allow for maintenance; put a plan in place and a set schedule to allow for security and z/VM updates or maintenance.

Security references

For further information about z/VM and hosting Linux VMs, as well as security and networks, refer to the following IBM Redbooks:

- *Security on z/VM*, SG24-7471
- *Introduction to the New Mainframe: z/VM Basics*, SG24-7316
- *IBM System z Connectivity Handbook*, SG24-5444

11.1.3 Hardening the base Linux VM

The term *hardening* is commonly used in server security to mean the process of taking a generic purpose operating system and changing it to only provide what is necessary for the production environment. This provides a baseline for security for the given operating system.

During migration you may be given an already hardened Linux image, and you will simply need to know what is allowed and not allowed with the image. However, if a hardened Linux image does not already exist, then you should create and maintain one.

Creating a new hardened Linux VM

The basics of hardening a Linux VM consist of removing all unnecessary applications and services, and then securing the applications and services that are left. Explaining this process is beyond the scope of this book, but the following references may prove helpful to your understanding of this topic:

- Hardening Linux
  http://www.freesoftwaremagazine.com/articles/hardening_linux
- Securing and Hardening Red Hat Linux Production Systems
  http://www.puschitz.com/SecuringLinux.shtml
- *Linux on IBM eServer zSeries and S/390: Best Security Practices*, SG24-7023

Migrating to a hardened Linux VM

A hardened Linux VM should have most if not all applications and services removed and or disabled. (Be aware that there may be more than one hardened Linux VM to chose from, so be sure to choose the version that provides the maximum number of applications and services that you need to perform the migration.)

You will need your migration analysis to determine what needs to be re-enabled. If any applications are to be installed and services enabled, you will need to provide credible business cases for each, individually or as a set. Completing the security analysis can provide just such business cases. Make sure the

documentation includes all applications and services as a delta from the base hardened Linux image.

> **Important:** RHEL includes the SElinux security method, and SLES includes AppArmor for its enhanced security method. Determine whether those environments are in use or required, and plan accordingly.
>
> Those mechanisms are very complex, so invest the time to identify code and applications that have not been ported to work in these environments.

Maintaining a hardened Linux VM

It is necessary to maintain base hardened Linux VMs. Kernels change and security patches are issued, so you need to develop a plan for maintaining the base image and assigning the resources to accomplish it. Thus, successive migrations will benefit from a properly maintained base hardened Linux VM.

11.2 Code and application analysis

Take the time to analyze all code and applications that are being migrated, because you will need to know what the current security methods are. You also need to understand what security methods will be used in the target Linux environment, and whether there will be enhancements. Finally, poll the stakeholders to ensure that all migration security requirements will be met.

When moving an application to a Linux VM, consider using as many VMs as you can. That is, separate as much as possible and use the Linux VM to isolate applications from one another and their data. If many images are available, design the system so that as much separation as possible exists between applications and data. The more isolation, the more straightforward the security will be.

11.2.1 Security issues

This section discusses determining potential security issues when migrating code and applications.

Migrating code

When migrating code, you need to ask whether any known security issues exist. If migrating the code to a Linux VM that is in an enterprise system, you do not want the application that will be generated from the code to be the weakest link in the system security. All known issues need to be addressed, so plan for it.

Migrating applications

If you know there is a security issue with an application, then do not use it. You will need to address all security issues before the system is placed in production. If there are more secure ways to configure an application, then invest the time to make those changes during migration; for example, place a database on a different VM than the application using it. Remember, the more separation, the more straightforward security will be. Systems with easy-to-understand security tend to be easier to defend and maintain.

11.2.2 Dependencies

This section discusses determining dependencies before you migrate.

Code dependencies

Almost all code uses APIs and other libraries to carry out the tasks that it was designed for. Thus, you need to review these dependencies *before* migrating. If you discover that a dependency exists on an item that has a known security issue, then you must find and implement a suitable replacement.

Application dependencies

A list of all application dependencies should be generated and reviewed for known security issues. Only fixed or known secure versions should be used. Then and only then should migration tests be done. Be aware that there will be a temptation to migrate the application over to the new Linux VM and test to prove that the migration is achievable, but such testing will be invalid if any application or its dependency is on code that has known security issues.

11.2.3 Checking user input

User input is the vector that is most commonly used to attack systems and programs, so all user interaction must be examined carefully. Check all input to make sure it is within the range of the data needed to be processed. Raw input should never be passed to another application or system request.

Exceptions should also be used. That is, try to ensure that input always conforms to the format that is expected and if the unexpected occurs, that it can be gracefully handled.

11.2.4 Planning for updates when migrating code

When code is migrated to an enterprise-class system, changes need to be addressed in a different manner. Unlike less critical code, changes must be

allowed to be executed while the application is still running. Thus, you must ensure that a method is in place to signal that configuration and control files have been updated and need to be reloaded.

There may be a security issue that needs to be addressed by configuration changes. In an enterprise environment, a program should not be stopped but only signaled to take on changes (for example, you may need to change the TCP port that an application uses). Ensure that the code can handle such changes gracefully.

Carefully examine all configuration changes. Do not assume that the changes are valid; verify that they are within the bounds of the setting. If they are not, handle the error gracefully.

11.2.5 Networking

If the code implements TCP sockets, make sure that its design and function are reviewed with the networking team that represents the firewall. That team will probably need to know the following information:

- What ports will be used by the code, and for what purpose?
- What type of protocol will be used: TCP, UDP, ICMP, and so on?
- Will special settings be used on the port, as in TCP keepalive?
- How long can the a connection tolerate a lack of response?
- How long will a connection be allowed to idle?

11.2.6 Logging and recording events

As previously mentioned, all logs must be kept in a way so that they cannot be changed. They need to be a permanent record of what occurred on the system. Configure the Linux VM so that syslog (the Linux system log) not only keeps a local record, but also forwards it to a remote secure system. Also make sure that all critical applications are properly configured to use syslog.

Important: SUSE has moved from using syslog to using syslog-ng (ng stands for next generation). syslog-ng is improved and is the preferred system logging facility.

At the time of writing, RHEL is still using syslog.

Implementing syslog logging when migrating code

On the Linux VM, either syslogd or syslog-ng will be running. Take time to update the code as needed to send messages to this daemon. At the very least, all

information that deals with security should be logged, as well as critical state information. The benefit of implementing syslog functionality is that log maintenance will be performed by the system (as in log rotation and archiving).

11.2.7 Escalations of authority

Apply the “principle of least privilege”; that is, programs should only operate with the authority needed to accomplish a goal. So if the code accesses a database, it should access it only as a user with the access needed, and not as an administrator.

Migrating code

Code should be analyzed to determine where there are escalations of authority. Also ensure that it accounts for exceptions, so that a de-escalation of authority exists. In other words, make sure that if the code is broken, it does not allow the user to operate at a different access level than is allowed.

Migrating applications

Programs that run as `root`, the super user, must be carefully examined and assurances given that they are operating as designed. Thus, it is best to not allow any code or program to run with such authority, if at all avoidable. Make sure that server applications are run at the suggested secure settings during all phases of the migration. You do not want to run applications as the administrator while developing, only to discover during testing that certain functions do not work.

11.2.8 Security test plan and peer review

All code and applications that are to be migrated should be in their secure mode during development straight through to test and deployment. It will also be necessary to validate the security assumptions made. This will determine the security test plan. Test everything that can be tested and document what was not tested and why. It is also worthwhile to test change control and verify the restore of backups. If an incident does occur, the only way to recover may be to patch the fault and restore data from the backups (assuming that they have not been compromised).

11.3 Availability and accountability

Security involves much more than simply who can access a system. It also involves keeping the system available to authorized users and unavailable to

unauthorized uses. Denial of Service (DoS) attacks have become more frequent in recent years, and Internet-facing systems must take the possibility of such threats into account.

To implement executable system security there needs to be an audit trail, without exceptions. All access to the system must be logged in a secure fashion to ensure that if an authorized user commits an indiscretion, that it cannot be covered up.

11.3.1 Availability analysis

Sometimes attackers do not break in to a system, but instead bring down a service by overwhelming it with requests. Thus system or services availability needs to be understood and Service Level Agreements maintained.

Internet-facing Linux VM considerations

The Internet is a public “space” where for the most part individuals are anonymous, so every effort must be made to mitigate malicious access if you have an Internet-facing Linux VM. You will need to be able to identify individuals and their IP addresses so that, if necessary, you can work with the networking team to prevent malicious access while still allowing authorized users to have access.

Communicating availability

Establish a standard for communicating system availability that explains how to report issues and outages, to ensure that they are communicated to the appropriate staff. An unexpected interruption in availability can be the first sign that there is a security issue that needs to be addressed.

11.3.2 Accountability analysis

As previously mentioned, all system logs and application logs must be immutable. If attackers gain access, they generally erase evidence of their presence to avoid detection. Also, if users attempt to perform unauthorized acts, they may try to cover their indiscretions by erasing log files or incriminating evidence.

Making log files immutable

Configure syslog or syslog-ng to store logs on a separate secure server. Optimally, the logs should be stored in a Write Once, Read Many (WORM) device. Do not delete logs, but keep a secure backup.

Audit trails encompassing all security domains

Make sure that security audits can be passed at all times by verifying that you can trace an individual's physical, network, and application access to systems across domains. You must be able to show a system access audit trail from all domains, not just from system access.

Authentication

Ensure that communication end-points are who they say they are. Attackers often "spoof" or pretend to be a system or user that they are not. To protect against such attacks, "authentication" conversations are used:

- Users must be assured that they are connecting to the server they think they are.
- Servers need to be assured that users are who they say they are.
- This authentication must be kept private so that eavesdropping cannot occur.

Disabling Telnet access and using Secure Shell (SSH) will accomplish this authentication. Using Secure Sockets Layer (SSL) with Web servers will also accomplish this and is preferred over the default of no SSL.

11.4 Data integrity and confidentiality

A benefit of migrating to a Linux VM is that data can be stored on an enterprise-class system. However, you need to analyze the current state of the data and then determine how it will fit in the new enterprise system.

11.4.1 Data integrity analysis

Data integrity refers to the assurance that data is unchanged from creation to reception. Data integrity also entails understanding the following items:

- Who can access what data and what is allowed
- Whether there an audit trail in place to map who changed what and when
- Whether the data is corrupted in some way and how is it to be restored
- Whether there a disaster recovery plan in place

Protecting data at rest from unauthorized access

Protecting access to a database is well understood, but what about protecting raw data on the disk itself? Mobile computers with databases full of accounts or data are sometimes misplaced or stolen. Thus, you need to protect data "at rest" (meaning the files themselves) and ensure that the data is kept in a secure way. You should prevent offline copies of a database from being kept on portable

devices or drives. Control of data is key. Be sure to communicate the data integrity policy to all individuals who have access, and monitor the audit trail to make sure that the policy is being enforced.

Data backups - part of security

Part of your security plan needs to include backups and how they are stored. They need to be kept in a secure way. When backups are kept separate from the system for disaster recovery purposes, use encryption to prevent unauthorized access. Understand the impact if the backups are stolen and mitigate the risk.

11.4.2 Confidentiality analysis

Confidentiality must first be communicated and then enforced. Thus, before users can access a system they need to be told what the confidentiality of a system is and how any data or information will be used or shared. Then a system needs to be in place to enforce the policy. This is normally done by auditing access logs. If a violation is detected, then it will need to be communicated to the affected parties.

Understanding laws and regulations before an incident occurs

Before you can create a confidentiality policy you need to understand what is legally expected:

- Are there national, regional, or state laws that need to be followed?
- Are there any industry compliance requirements (such as Payments Card Industry (PCI) requirements) regarding the storage of credit card information?
- Is there a company policy? If so, it needs to be followed.
- Document all expectations regarding how long to keep the data (for example, “We expect or are required to keep the data for up to 5 years.”).

Publishing your confidentiality policy

You need to communicate the confidentiality policy in such a way as to ensure that all users of the system are aware of it and thus can be held accountable. When a use logs in to a system, use the Message Of The Day (MOTD) found in `/etc/motd` as shown in Example 11-1 on page 134 to communicate with your system users.

Example 11-1 Use /etc/motd to communicate system policy

```
****************************************************
*      .--.     Welcome to the Linux s/390x VM     *
*     |o_o |    SUSE Linux Enterprise Server 10.2  *
*     |:_/ |    System Admin: John Doe              *
*    //   \ \                 jdoe@company.com     *
*   (|     | )  This system governed by corprate   *
*  /'\_   _/`\  Policy K49-r v21 please read       *
*  \___)=(___/  before accessing system            *
****************************************************
```

Tip: Use ANSI art or special characters to make the login window attractive. It is useful to display system information such as the Linux distribution with its version and release information, along with a greeting.

On Web pages, create a link from the main page so that the system policy can be easily accessed. If you are allowing VNC login, as discussed in Appendix B, “Remote access applications” on page 245, display the policy by updating `/etc/gdm/custom.conf` as shown in Example 11-2.

Example 11-2 Policy found in /etc/gdm/custom.conf

```
[greeter]
DefaultRemoteWelcome=false
RemoteWelcome=Connected to %n must read policy K49-R v21
```

Having a plan in place before an incident occurs

Have a plan in place in case confidentiality is violated. The plan should include:

- Who should be notified and what should be disclosed about the incident.
- If there is a requirement to notify the public, document how and what should be disclosed.
- Communicate actions that will be taken to prevent future incidents.

11.5 Security change management

No system is perfect so there will be changes, however infrequent. Because security fixes are important to keep current, there should be a plan to understand their impact on the system. If a Linux VM needs to be restarted, it must be done in an orderly and timely basis.

After the system is moved from test to production mode, it will remain that way. Outages are expensive for companies, but failing to plan change windows and downtime will also cause security problems. In the rare case that a VM needs to be restarted, you need the ability to allow for these types of changes.

Testing changes with a clone of the Linux VM

The advantage of migrating to a Linux VM is that you can clone a VM and test changes before applying them to the production images. Run through the complete change from start to finish, rather than assuming will work.

Record how long it takes to make changes and test worse case scenarios (also keeping track of the time). After testing the change on the clone is complete, you will be able to report to production stakeholders how long the change will take and how long the worst case will take.

11.6 Enterprise authentication options

Migrating to an enterprise system means that user and identification management can be consolidated. In this section we discuss enterprise authentication options and where to find the corresponding information explaining how to implement them.

11.6.1 A common centralized LDAP server

When migrating applications and code to a Linux VM, you can simplify user administration by storing user information in a Lightweight Directory Access Protocol (LDAP) server. Configuring the Linux VM to authenticate from a centralized LDAP server provides the following benefits:

- User management is simplified; users can be managed across the enterprise.
- Changes made to a user will be applied across all images.
- An offline VM could contain outdated user information. Using LDAP assures that bringing an old image online will not compromise current security.

11.6.2 LDAP server on z/OS means RACF integration

If RACF® is used to manage user information, then installing LDAP on a z/OS system will allow LDAP access to RACF. In turn, this allows a single, highly secure repository of user information in RACF and lets that information be exposed to Linux VMs via an LDAP server. For more information, refer to *Linux on IBM zSeries and S/390: Securing Linux for zSeries with a Central z/OS LDAP Server (RACF)*, REDP-0221.

You can also configure Samba to use LDAP as its user repository. Thus, you can have one security domain across MS Windows, AIX and Linux, with System z as the core. For more information about this topic, refer to *Open Your Windows with Samba on Linux*, REDP-3780.

11.7 Integrated Cryptographic Service Facility

When migrating to a Linux VM, the underlying hardware has the ability to accelerate cryptographic mathematics. The Integrated Cryptographic Service Facility (ICSF) allows for this work to be offloaded from the processor. Instead, the work is processed by the crypto-assist processor that is integrated into every processing unit (PU) of every System z9® and System z10U, or the Crypto Express card, if it is installed.

The supported APIs are listed here.

OpenCryptoki

An open source implementation of Public-Key Cryptography Standard #11 (PKCS#11), OpenCryptoki uses the libica shared library to access IBM cryptographic adapters through the z90crpyt device driver.

OpenSSL

An open source implementation of Secure Sockets Layer, OpenSSL can utilize the libica shared library for hardware encryption.

Global Security Kit

Provided as part of the IBM HTTP server, Global Security Kit (GSKit) manages SSL certificates. It utilizes OpenCryptoki for hardware encryption.

Using this approach will offload the cycles and allow for more concurrent access to a Web server that is using SSL or applications that use one of supported APIs. Refer to *Using Cryptographic Adapters for Web Servers with Linux on IBM System z9 and zSeries*, REDP-4131, to learn how to configure your system so that your Linux VM will take advantage of the installed hardware.

12

Operational analysis

The source application comes with a complete support structure. Part of that support structure performs daily operational tasks. Depending upon the application, this support could be 24 hours a day, 7 days a week, 365 days a year. The application will rely upon manual and automated intervention to start, stop, monitor, and maintain the services provided by the application.

This chapter discusses some of the operational issues which, if present in the source application, must be addressed in the target application. A careful and detailed analysis about how the source application is supported by operations staff is required for a successful migration effort.

An analysis of the operational functions may highlight characteristics of the application that were not clear from the analysis of other application interfaces or from the code itself. The application code may be successfully ported but it is just as important that the application's operational support structures be migrated successfully as well.

12.1 The operational environment

Operational environments present many tasks and challenges to the operations staff, who are often required to multitask when monitoring consoles and managing other physical equipment. For this reason, it is important to ensure that the migrated application fits in smoothly with the current operational environment.

Operational tasks that might be affected by the source application migrating to the target application running on Linux on System z are discussed in 12.2, “Operational migration tasks” on page 138.

12.2 Operational migration tasks

This section discusses operational issues that might change when migrating the source application to the target application in a new environment.

- Starting and stopping the application

  These processes can be automated or manual. The source application probably had certain methods for starting and stopping its processes, but the target application will probably have different commands and methods for starting and stopping the application.

  If the target application is a manual process, then the operators will have to be trained and the appropriate documentation will have to be written and published. If it is an automated process, then the automation scripts need to be written, tested, documented, and explained to the operators.

- Notification of problems

  Sometimes operators can receive automated messages or indicators that they are unfamiliar with and do not know how to respond to. Operators need to know who to turn to for guidance when this type of problem arises, so the application owner needs to be clearly identified. If the application owner is unavailable or unresponsive, then escalation procedures need to be in place. These details might change when the application is migrated to the target system.

- Normal intervention and monitoring

  Some applications need to be checked or modified during their life cycle throughout the day. Often this simply involves monitoring indicators or displays that show the health of the application. New procedures for the migrated target application must be communicated to the operators.

Hands-on training sessions are optimal for operators as they learn by observation and perform required tasks.

- Hardware manipulation

  Some migrations will include hardware consolidation or replacement. Operators will need to be trained on how to operate and manipulate the new hardware. Even if the operators are not required to manipulate the hardware, it is still useful to let them know what is running on the new server and to have the appropriate documentation, labels, and signs available for reference.

- Hardware intervention and problem escalation

  There are fewer hardware interventions for operators to deal with on System z.

  For example, with the source application and server, an operator might be comfortable with and even required to reboot a server by using the power switch. On System z, however, it is a serious error to use a power switch to react to a server or application problem.

  If there is a new hardware vendor in the migration project, then the method that the operators must use to notify the vendor of an actionable message or event needs to be communicated to the operators. A test of that procedure should be carried out and then documented. You should not wait for a critical situation to occur before learning how to contact vendors or other support personnel. The contact information should include day shift, off hours, and weekend names and numbers. The requirements for the vendor contact should be clear. The vendor often requires precise, detailed information such as serial numbers, machine type, location, and so on.

- Batch procedures and scheduling

  Most applications will have batch processes that support the application. Automatic scheduling software is common at most installations to schedule and track those batch processes. Schedulers within the operations department will be involved to create the necessary scheduling changes for the migrated application. The new schedules will then be communicated to the operators on each shift.

- Other considerations

  Not everything in your operating environment can be envisioned and described here. The intent of this chapter is to give you an idea of possible operational issues related to the migration project. Think of everything in your operating environment that may change or be affected by the migration of the source application to the target application. Then create a plan to perform the requisite operational migration tasks. And finally, execute your plan.

13

Disaster Recovery and Availability analysis

Migration an application to Linux on System z provides organizations with the opportunity to configure their application environments to an availability profile that matches its importance to the business.

This chapter discusses the various options available on the IBM System z platform for availability and disaster recovery. We investigate the various levels of availability and illustrate them using examples of various high availability configurations.

13.1 Availability and Disaster Recovery overview

IT system outages can significantly impact businesses by rendering critical systems unavailable. The key to ensuring that this does not occur is to analyze your systems and determine a hierarchy of availability need. Keep in mind that not everything needs a remote hot site.

For better understanding, the following terms and definitions are used when discussing Disaster Recovery, High Availability, and related concepts.

Disaster Recovery

Planning for and utilizing redundant hardware, software, networks, facilities, and so on to recover the IT systems of a data center or the major components of an IT facility if they become unavailable for some reason.

The definitions of High Availability, Continuous Operations and Continuous Availability are drawn from the IBM High Availability Center of Competence.

High Availability

Provide service during defined periods, at acceptable or agreed upon levels, and mask unplanned outages from users. High Availability (HA) employs Fault Tolerance, Automated Failure Detection, Recovery, Bypass Reconsideration, Testing, Problem and Change Management.

Continuous Operations

Continuously operate and masked planned outages from end-users. Continuous Operations (CO) employs nondisruptive hardware and software changes, nondisruptive configuration changes, and software coexistence.

Continuous Availability

Deliver nondisruptive service to users 7 days a week, 24 hours a day. With Continuous Availability (CA), there are no planned or unplanned outages.

The ultimate goal for mission-critical systems should be Continuous Availability; otherwise, the systems should not be defined mission-critical.

13.2 Availability analysis

Migrating an application to a virtualized Linux environment on IBM system z offers an opportunity to implement an availability profile in line with the impact of the unavailability of the application has on the organization's overall business.

Sometimes, however, such an analysis is not straightforward. For example, test and development workloads are generally not considered to be mission-critical. However, because they may be needed to correct an error in a production system, consider providing for some sort of test and development environment in your DR planning.

The challenge with DR is to achieve a balance between the impact of an unavailable system on the health of the business versus the cost of creating a resilient environment for the application. This planning should include the likely scenarios that could impact an application's availability, as well as unrelated events that could impact the ability of a business to function.

The usual IT issues such as server failure, network failure, power outage, disk failure, application failure, operator error, and so on can be planned for through duplication of resources and sites. Unrelated factors are rare and not directly related to IT, but they can have a huge impact on the ability of a business to function. These events include fire, natural disasters such as earthquake, severe weather and flood, as well as civil disturbances which can have a major impact on the ability of people to go to work.

Although this chapter focuses on the IT-related issues, you should also have a plan in place to deal with the other, non-IT related events.

13.3 Single points of failure

In determining the DR requirements of an application, you need to look at the probability of failure of a component as well as the cost to eliminate a single point of failure (SPOF).

Table 13-1 lists the components of an IBM System z virtualized environment running an application under z/VM and Linux and the relative costs of rectifying a single point of failure.

Table 13-1 Potential single points of failure that can impact availability

Single point of failure	Probability of failure	Cost to rectify
System z hardware	Very low	High
System z LPAR	Very low	Low
z/VM	Low	Low
Linux	Low	Very low
Disk system microcode	Low	Medium

Single point of failure	Probability of failure	Cost to rectify
Virtual network within z/VM system	Very low	Low
Physical network	Medium	Medium
Application	High	Very Low

Apart from hardware and software failures there are other outages that can impact an application's availability. These planned outages are:

- Hardware upgrades requiring a Power-on Reset
- LPAR configuration changes requiring a reboot of the LPAR
- z/VM maintenance
- Linux kernel maintenance requiring a reboot
- Application maintenance

13.4 System z features for High Availability

IBM System z has been designed around providing High Availability. Perhaps the most design effort has gone in to the transparent recovery of processor errors. In the event of a hard processor errors at an individual core level, the task is moved to a spare processor where processing continues transparently to the application. Two spare processors are available on all z10 ECs. The z10 BC will have zero (0) spares if all 10 CPs are in use and any unassigned processor can also be used as a spare. In the IBM System z10, a number of availability features have been introduced to reduce the number of planned system outages. For example, the following actions are now fully concurrent and require no system outage:

- Adding logical partitions (LPARs)
- Adding logical processors to a partition
- Adding logical channel sets (LCSSs) - I/O paths
- Adding subchannel sets
- Enabling dynamic I/O
- Adding a cryptographic processor to an LPAR

Additionally many services enhancements have been introduced to avoid planned outages:

- Concurrent firmware fixes
- Concurrent driver upgrades
- Concurrent parts replacement
- Concurrent hardware upgrades

The IBM System z10 offers a number of customer-initiated capacity on demand features. These billable features are designed to provide customers with additional capacity to handle the following events:

- Customer-Initiated Upgrade (CIU) is used for a permanent capacity upgrade.
- Capacity BackUp (CBU) is predefined capacity for DR. A system at a DR site does not need to have the same capacity as the primary site. In the event of a declared disaster, or for up to 5 DR tests, the customer can turn on the number of processors, including IFLs, required to handle the workload from the primary site.
- Capacity for a Planned Event (CPE) is used to replace capacity lost within the enterprise due to a planned event such as a facility upgrade or system relocation.
- On/Off Capacity on Demand provides extra capacity in 2-hour increments that is available to be turned on to satisfy peak demand in workloads.

Note: For more information about IBM System z10, refer to *IBM System z10 Enterprise Class Technical Introduction*, SG24-7515.

For an in-depth document about IBM System z10 Reliability, Availability and Serviceability (RAS) features, refer to *IBM Journal of Research and Development Vol 53, No. 1 Paper 11 2009*, which is available at:

http://www.research.ibm.com/journal/rd/531/clarke.pdf

13.5 Availability scenarios

In the following scenarios present a number of different situations where a Linux on IBM System z environment is set up with increasing degrees of availability and increasing levels of cost. The key to maximum availability is to eliminate single points of failure.

In all scenarios it is assumed the IBM System z10 is configured with redundant LPARs, redundant channel paths to disk (FICON and FCP), redundant Open System Adapters connected to the organization's network, redundant system consoles and redundant Hardware Management Consoles. This is the normal setup for an IBM System z10.

The application design should include redundant software servers. The storage infrastructure should also include redundant FICON directors, redundant Fibre Channel switches, mirrored disks, and data.

The communications network should be designed around redundancy with redundant network routers, switches, hubs and wireless access points.

Remember that for mission-critical systems, an uninterruptible power supply should also be provided as well as a second site far enough away from the primary site to avoid being affected by natural disasters.

Another important factor in the availability of applications is security and access controls. For more information about this topic, refer to Chapter 11, “Security analysis” on page 121.

13.5.1 Single System z LPAR: Clustered WebSphere Application Server environment

Figure 13-2 shows a System z LPAR sharing system resources to all Linux virtual machines in the LPAR. The WebSphere Applications Servers are in a two-node cluster. If the Integrated Facility for Linux (IFL) fails, System z10 will automatically switch the workloads to a spare or any unassigned processor without any disruption to the active task.

If a Linux Virtual Machine running the WebSphere Application Server workload fails, the other node in the cluster will take over if you are running WebSphere Application Server Network Deployment. This is achieved because an application deployed to a cluster runs on all members concurrently. Additional availability is provided through the nondisruptive addition of new virtual machines to the cluster.

Note: z/OS is optional in the first six scenarios.

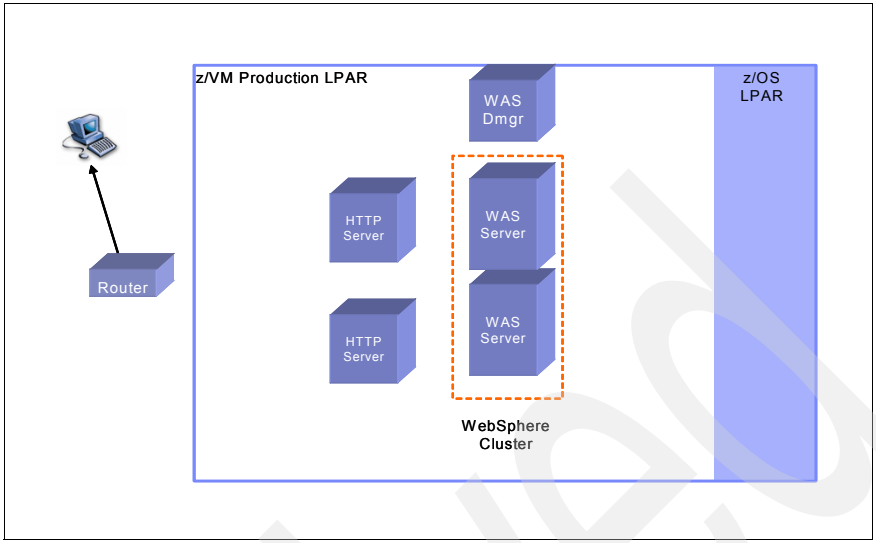

Figure 13-1 Single LPAR WebSphere Application Server cluster

This environment also provides additional availability through redundant HTTP servers.

13.5.2 Multiple LPARs: HA solution for Linux on System z

Figure 13-2 on page 148 shows a scenario where there are three LPARS defined. Each LPAR could have a dedicated IFL or a single IFL or multiple IFLs could be shared among all LPARs. The LPAR weight determines the relative priority of an LPAR against other LPARs.

In this case the production workload and WebSphere Application Server cluster is split across two LPARs, which gives HA to WebSphere Application Server because an LPAR or z/VM failure will not impact the availability of WebSphere Application Server.

Development and test workloads run in their own LPAR so any errant servers will have no impact on the production workloads. As in the first scenario, a failure of a System z10 IFL will be rectified automatically without any impact to the running application.

This configuration eliminates most failure points at a reasonable cost.

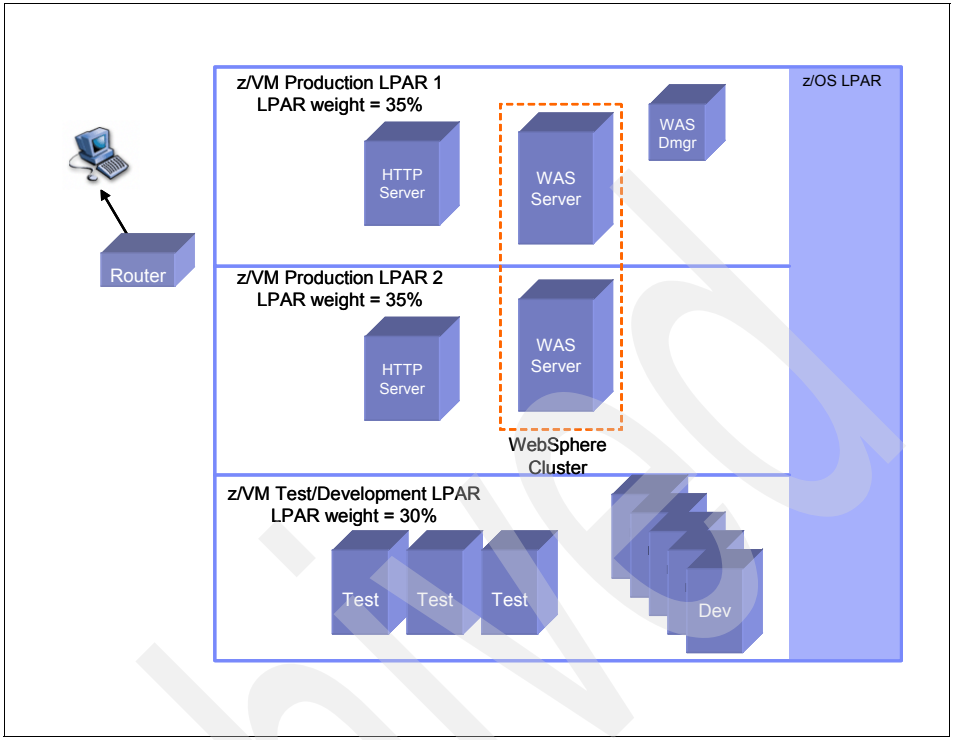

Figure 13-2 HA Linux environment on IBM System z

13.5.3 Active/Cold standby cluster

Figure 13-3 on page 149 describes another approach in which, instead of having redundant virtual servers, an active/cold standby cluster is established. In this case Tivoli System Automation Manager for Multiplatforms (SA) monitors the servers and in the event of an outage will automate failover to the cold standby server.

SA runs on each node in the cluster. It monitors cluster nodes and exchanges information through Reliable Scalable Cluster Technology (RSCT) services. SA also creates a Service IP address as an alias on an appropriate network adapter on Node 1 where the HTTP server will be started.

Only one instance of the HTTP Server is defined to SA to be able to run on either of the two nodes with a “depends on” relationship to a single IP address (the Service IP). SA starts the HTTP Server on Node 1 and at user-defined intervals invokes a script to confirm it is still up and serving pages. It also monitors the Linux node itself to ensure it remains active.

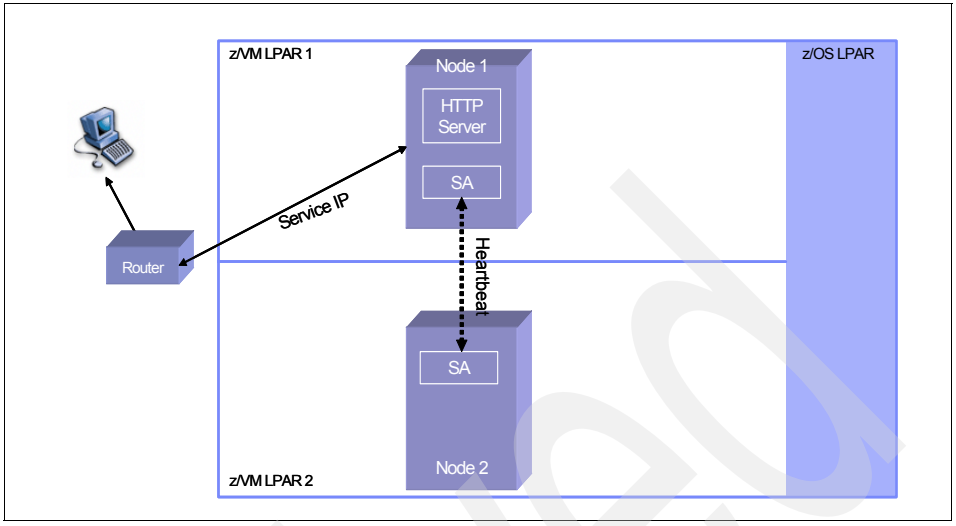

Figure 13-3 Normal situation - Tivoli System Automation monitors for outages

When a failure occurs, RSCT determines that Node 1 is no longer responding. SA then moves the Service IP over to Node 2 and restarts the HTTP server there, as illustrated in Figure 13-4.

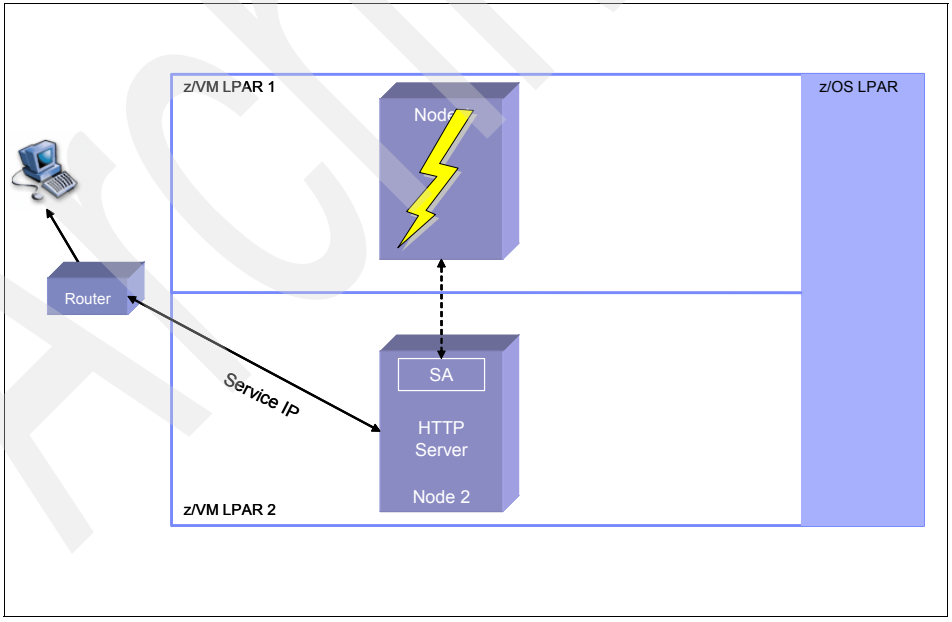

Figure 13-4 Outage occurs - Tivoli System Automation fails over to cold standby

13.5.4 Active/Active Application Server cluster

Figure 13-5 shows the WebSphere Application Server setup in an active/active configuration where the WebSphere Application Server Cluster spans two Linux virtual machines in two LPARs. This setup handles the very rare occurrence of the failure of an LPAR. More importantly, it also allows z/VM maintenance to be performed without an outage to the WebSphere applications. In this case the Linux servers and z/VM are shut down in LPAR 2. An Initial Program Load (IPL) is done of z/VM with new maintenance applied and the Linux virtual machines are restarted and the WebSphere cluster is restored. This task would be scheduled for a time when the processing load is light.

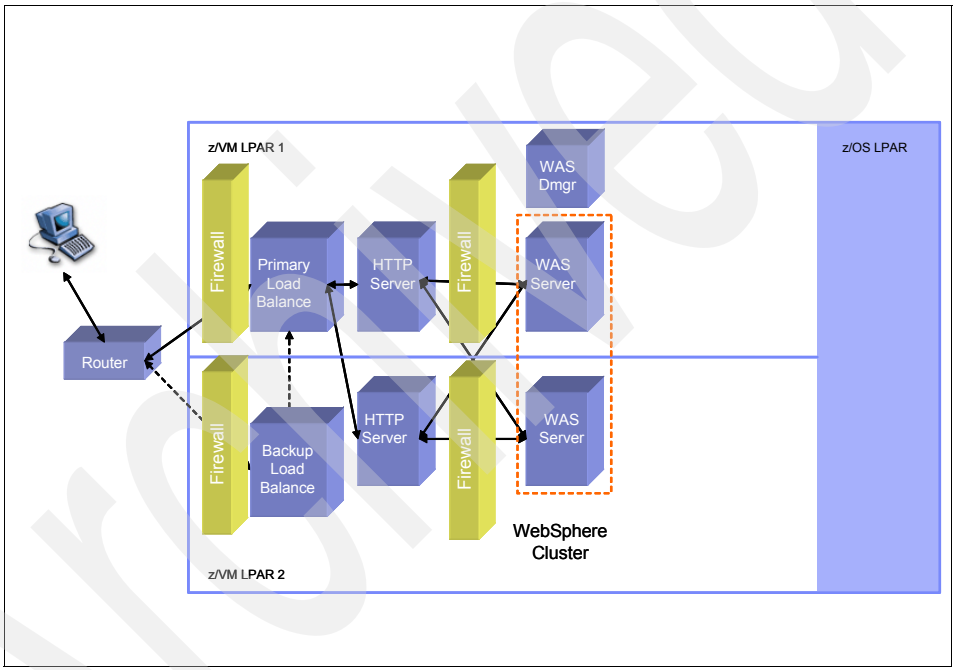

Figure 13-5 Active/active WebSphere Application Server cluster

13.5.5 Active/Active WebSphere Application Server cluster with database replication

Figure 13-6 on page 152 shows a DB2 database added to the active/active WebSphere cluster. To provide HA for the DB2 database, the DB2 data replication feature, High Availability Disaster Recovery (HADR) is used. HADR protects against data failure by replication changes from the source database (called primary) to a target database (called standby).

In the event of a z/VM or LPAR outage of the primary DB2 system, the standby DB2 system will take over in seconds, thus providing high availability. Communication between the DB2 primary and DB2 standby systems is via TCP/IP which in this case would be done using the System z high speed virtual network feature HiperSockets.

The Standby DB2 system can also be located at a remote site to provide enhanced availability in the event of a site failure.

IBM Tivoli System Automation (SA) running in both DB2 servers is designed to automatically detect a failure of the primary, and it issues commands on the standby for its DB2 to become the primary.

Other cluster management software could be used. However, Tivoli System Automation and sample automation scripts are included with DB2 to only manage the HA requirements of your DB2 database system.

Note: For more information about high availability DB2 features, refer to *High Availability and Disaster Recovery Options for DB2 on Linux, UNIX, and Windows*, SG24-7363.

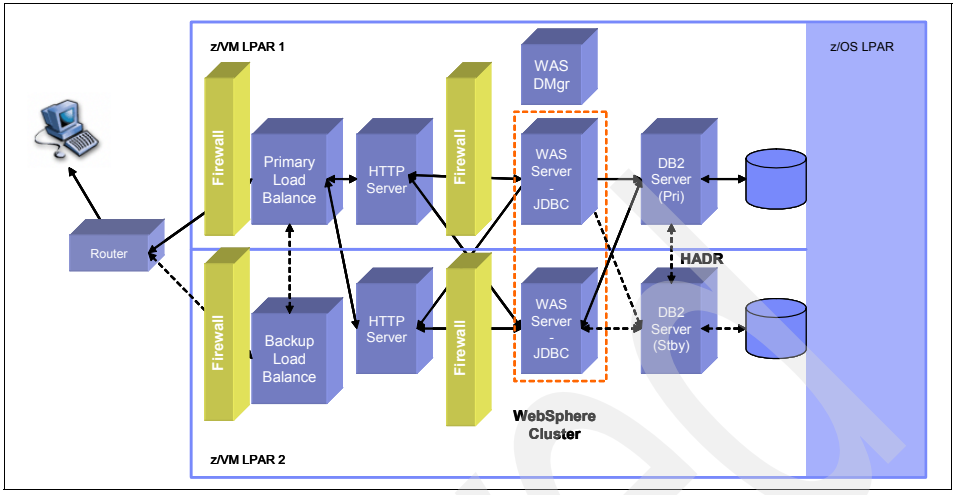

Figure 13-6 Active/active WebSphere Application Server cluster and DB2 HADR

13.5.6 Active/Active WebSphere Application Server cluster with database sharing

Figure 13-7 on page 153 shows that database sharing was introduced using Oracle Real Application Clusters (RAC). Oracle RAC provides HA for applications by having multiple RAC nodes sharing a single copy of the data. If a cluster node fails, the in-flight transaction is lost but the other server in the RAC can receive all Java Data Base Connector (JDBC) requests.

In a System z environment, communication between the database nodes would use a virtual LAN in the same LPAR or HiperSockets to other LPARs. Both methods are at memory-to-memory speeds with very low latency.

For more information about Oracle RAC, go to:

http://www.oracle.com

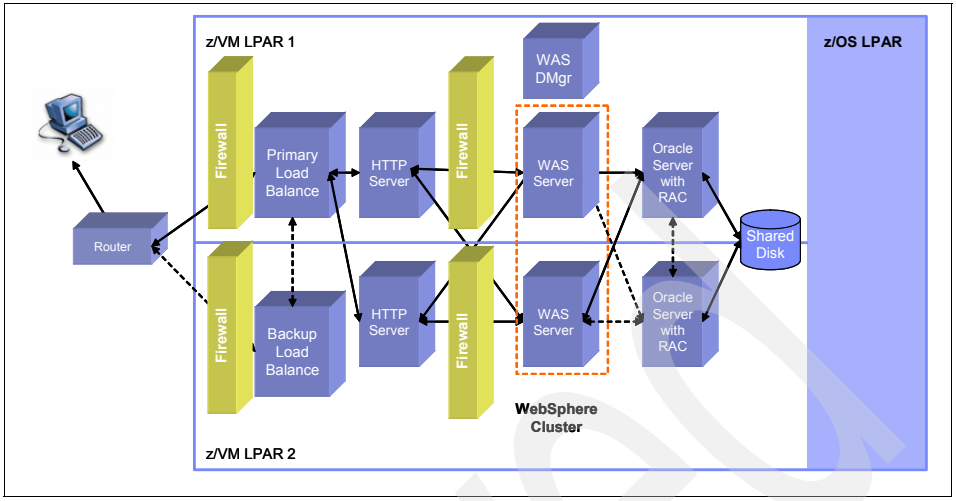

Figure 13-7 Database sharing using Oracle RAC

13.5.7 Active/Active WebSphere Application Server cluster with DB2 sharing in z/OS Parallel Sysplex

Figure 13-8 on page 154, we introduce the additional benefits provided by the z/OS Parallel Sysplex®. Briefly, a Parallel Sysplex is a high availability configuration designed to provide CA of systems and applications. In the case of DB2 data sharing, the Parallel Sysplex allows all members of the sysplex update access to shared data through the use of a centralized arbitrator known as the Coupling Facility (CF).

Each WebSphere Application Server is configured to use the JDBC Type 4 driver for communication with the DB2 z/OS data sharing members. It is sysplex-aware and works cooperatively with DB2 and the z/OS Workload Manager (WLM) on z/OS to balance workloads across the available members of the DB2 data sharing groups.

Note: For an introduction to z/OS Parallel Sysplex, refer to: *Clustering Solutions Overview: Parallel Sysplex and Other Platforms*, REDP-4072.

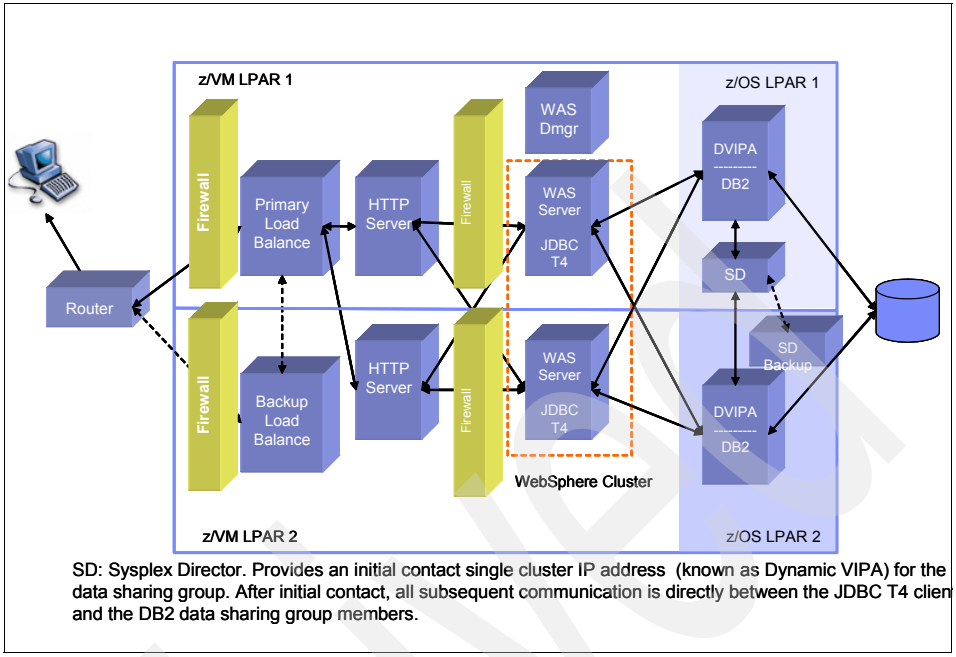

Figure 13-8 Database sharing using z/OS Parallel Sysplex

13.5.8 Active/Active WebSphere Application Server cluster with database sharing on z/OS across cities

For the ultimate availability solution, it is possible to have two sites up to 100 km (62 miles) apart and provide full DB2 data sharing between WebSphere Application Server clusters at each site. The key element in this solution is Globally Dispersed Parallel Sysplex (GDPS®) Metro Mirror. GDPS Metro Mirror uses a feature on the IBM ESS800 and DS6000™ and DS8000® family of storage systems called Peer to Peer Remote Copy (PPRC).

All critical data resides on the storage subsystem (or subsystems) in Site 1 (the primary copy of data) and is mirrored to Site 2 (the secondary copy of data) via Synchronous PPRC. With Synchronous PPRC, the write to the primary copy is not complete until it has been replicated to the secondary copy. PPRC is designed to make it possible for a site switch with no data loss.

The primary Controlling System (K1) running in Site 2 performs the following services:

- It monitors the Parallel Sysplex cluster, Coupling Facilities, and storage subsystems, and maintains GDPS status.
- It manages a controlled site switch for outages of z/OS and Parallel Sysplex, z/VM, and Linux on System z (as a guest under z/VM).
- It invokes HyperSwap®[1] on z/OS and z/VM for a site switch of disk subsystems, which can eliminate the need for an IPL at the recovery site to use the mirrored disks.
- It works with Tivoli System Automation Multiplatform across z/VM and Linux to understand their state and coordinate their restart during the recovery phase.
- It invokes network switching, based on user-defined automation scripts.

Figure 13-9 on page 156 shows Site A and Site B are in a GDPS and share the same DB2 data. GDPS helps to automate recovery procedures for planned and unplanned outages to provide near-Continuous Availability and Disaster Recovery capability.

[1] HyperSwap is a z/OS feature that provides for the continuous availability of storage devices by transparently switching all primary PPRC disk subsystems with the secondary PPRC disk subsystems for planned and unplanned outages.

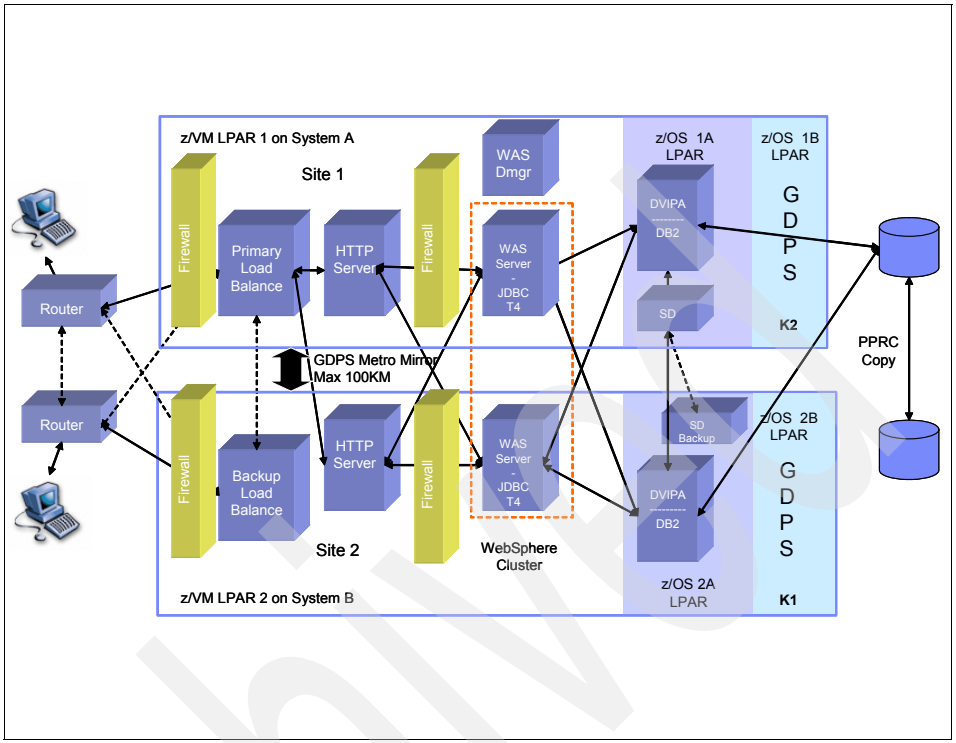

Figure 13-9 GDPS solution for near-continuous availability

Distances greater than 100 km require an asynchronous copy where the application resumes after the write is performed to the primary copy. The write to the remote mirror takes place at a later time, so it is not synchronized with the primary copy. More detailed discussion of this topic is beyond the scope of this book.

13.6 Linux-HA Project

The Linux-HA Project provides HA solutions for Linux through an open development community. The majority of Linux-HA software is licensed under the Free Software Foundation GNU Public License (GPL) and the Free Software Foundation GNU Lesser General Public License (LGPL).

The Linux-HA release 2 software package provides the following capabilities:

- Active/active and active/passive configurations
- Failover and failback on node IP address or resource failure

- Failover and fallback on customized resource
- Support for Open Cluster Framework (OCF) resource standard and Linux Standard Base (LSB) resource specification
- Command-line interface (CLI) and Graphical User Interface (GUI) for configuration and monitoring
- Support for a 16-node cluster
- Multi-state (master/subordinate) resource support
- Rich constraint support
- XML-based resource configuration
- No kernel or hardware dependencies
- Load balancing capabilities with Linux virtual server

\

Note: For more details of Linux-HA and examples of its use in a z/VM Linux on System z environment, refer to *Achieving High Availability on Linux for System z with Linux-HA Release 2*, SG24-7711

13.7 Understanding the availability requirements of your applications

This section describes how Service Level Agreements (SLAs) and the cost of providing availability can help you achieve a better understanding of the availability requirements of your applications.

13.7.1 Service Level Agreements

To determine the availability requirements of applications you want to migrate to Linux on System z, you must take into account the needs of the business units that rely on these applications. Ideally, Service Level Agreements (SLAs) are in place which state requirements such as availability needs, response time, maximum system utilization, DR requirements, and so on. This should be the basis for the design of the target system on Linux.

If SLAs do not exist, then before starting to design a solution, discuss with the business units what levels of service you can offer and what level of investment they are willing to make. The key to the success for an SLA is that it is both achievable and measurable with defined penalties for failure to deliver. You also need to ensure that there are regular reviews because things will change.

According to IT Service Management principles, a Service Level Agreement would typically define or cover the following topics:

- The services to be delivered
- Performance, tracking, and reporting mechanisms
- Problem and change management procedures
- Dispute resolution procedures
- The recipient's duties and responsibilities
- Security
- Legislative compliance
- Intellectual property and confidential information issues
- Agreement termination

Some of these components may not be relevant in an "in-house" SLA.

From an availability view point an SLA for an "in-house" business application should focus on the first two items, name what service is being delivered and how is it being measured:

- Application availability hours, for example:
  - 24 hours/day x 7 days a week
  - 6 am to 6 pm, weekdays
  - 9 am to 5 pm, weekdays, and so on
  - Definition of how availability is measured and who will do the measurement. For example, system availability, application availability, database availability, network availability, and so on
- Minimum system response time
  - Defined number and definition of where and how is it measured

13.7.2 The cost of Availability

As shown from the examples in this chapter, there is a great degree of difference in cost and complexity of the various Availability options discussed. Providing CA and a DR plan is not an insignificant expense but with the degree of reliance on IT systems by most businesses today, it is a cost that cannot be ignored.

If you have a Web-facing revenue-generating application, you can calculate the cost of downtime by simply monitoring the average revenue generated over a period of time. This provides an idea of amount of revenue that may be lost during an outage and how much you should spend to make the application more

resilient. Other businesses will have different ways of calculating the cost of downtime.

Keep in mind that for any HA configuration to be successful in a real DR situation, there needs to be a fully documented DR plan in place that is fully tested at least once every year.

Part 3

Hands-on migration to Linux on System z

In this part of the book we describe four migrations to Linux on IBM System z. These four migrations cover:

- A migration of WebSphere Application Server and DB2 from Windows 2003 Server to Red Hat Enterprise Linux 5 on System z
- A migration of open source application Sakai and MySQL from z86 Linux to SUSE Linux Enterprise Server 10 on IBM System z
- A LAMP migration of Media-Wiki to Linux on IBM System z
- A migration of .NET programs from Windows to SUSE Linux Enterprise Server on System z using MONO Extensions

This section also contains Chapter 15, “Technical procedures” on page 175, which discusses technical approaches you can implement for better coexistence of z/VM and Linux.

14

MS Windows to Linux - WebSphere and DB2 migration

This chapter describes the migration of a Java application and DB2 database from an existing Microsoft Windows, WebSphere, and DB2 environment to a Linux, WebSphere, and DB2 environment. The application was chosen by a client because it could be fully tested without access to other systems or applications.

The goal was to create a Linux environment running under z/VM for a proof of concept to prove the viability of a migration to Linux on System z.

14.1 Application migration background

The client's Intel server environment supported 15 Java applications running on an IBM WebSphere Application Server on a Microsoft Windows 2003 Server. The Java applications use DB2 as the database server, which also runs on the MS Windows 2003 Server. The self-contained application, ACE, was selected for the proof of concept. To conduct the proof of concept, IBM installed Red Hat Enterprise Linux 5 under z/VM on IBM System z. IBM HTTP server, WebSphere Application Server and DB2 were also installed on this Linux system.

Because these applications were self-contained, they were chosen by the client as an ideal and mission-critical application that could be set up easily.

Note: WebSphere Application Server, since the release of Version 6, has a common code base across all platforms. Thus, migrating Java applications across platforms is much simpler and WebSphere skills are applicable across multiple platforms.

14.1.1 Existing server (source) environment

The client's environment consisted of more than 60 rack-mounted Intel servers running on the MS Windows 2003 Server. The Web server was the Microsoft Internet Information Server. There were 22 servers in production, with the remainder allocated to an internal and external User Acceptance Test (UAT), System Test, development and disaster recover (DR).

The majority of the Intel servers were rack-mounted single core, two Xeon CPU systems running at 3.6 GHz. A very small number of virtualized servers were running under VMware ESX 2.5.

The client was contemplating a migration to Linux on System z in part because of a number of performance issues related to the current setup. WebSphere Application Server ran on 6 Intel servers in a 3 by 2 node MS Windows clustered configuration. There were also 6 database servers, 3 active and 3 standby. This clustered environment was unable to meet the heavy demands of the WebSphere Application Server applications during peak load.

The primary middleware running was WebSphere Application Server - Network Deployment V6.0.27 and DB2 Workgroup Edition Version 8.1.

Figure 14-1 on page 165 displays a diagram of the existing production environment.

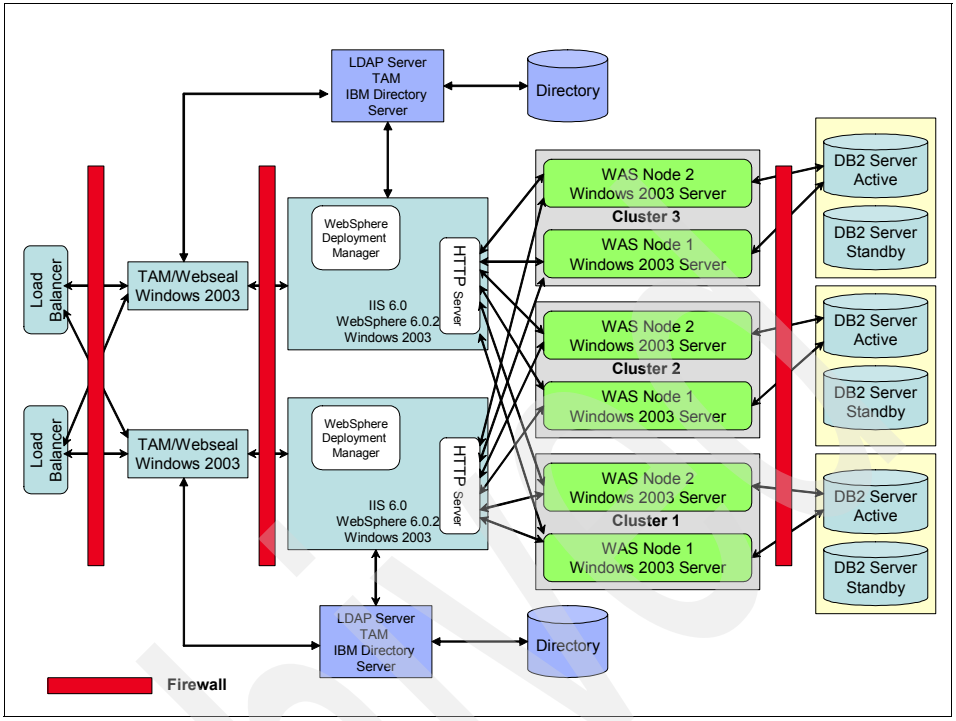

Figure 14-1 Client WebSphere and DB2 production environment

14.1.2 Target server environment

Because the customer wanted to consolidate its distributed MS Windows, WebSphere Application Server and DB2 environment to Linux on System z, and because this was a proof of concept, it was mutually agreed to set up a single Linux virtual machine that would contain the IBM HTTP Server (in place of Microsoft IIS), WebSphere Application Server ND and a DB2 database management system. The code levels for DB2 and WebSphere Application Server and DB2 were identical to the source environment.

> **Tip:** The customer was running DB2 Work Group Edition on an MS Windows 2003 Server. To run DB2 on Linux on System z it was necessary to install DB2 Enterprise Server Edition.

Figure 14-2 on page 166 shows the simplified Linux, WebSphere Application Server and DB2 environment for the proof of concept.

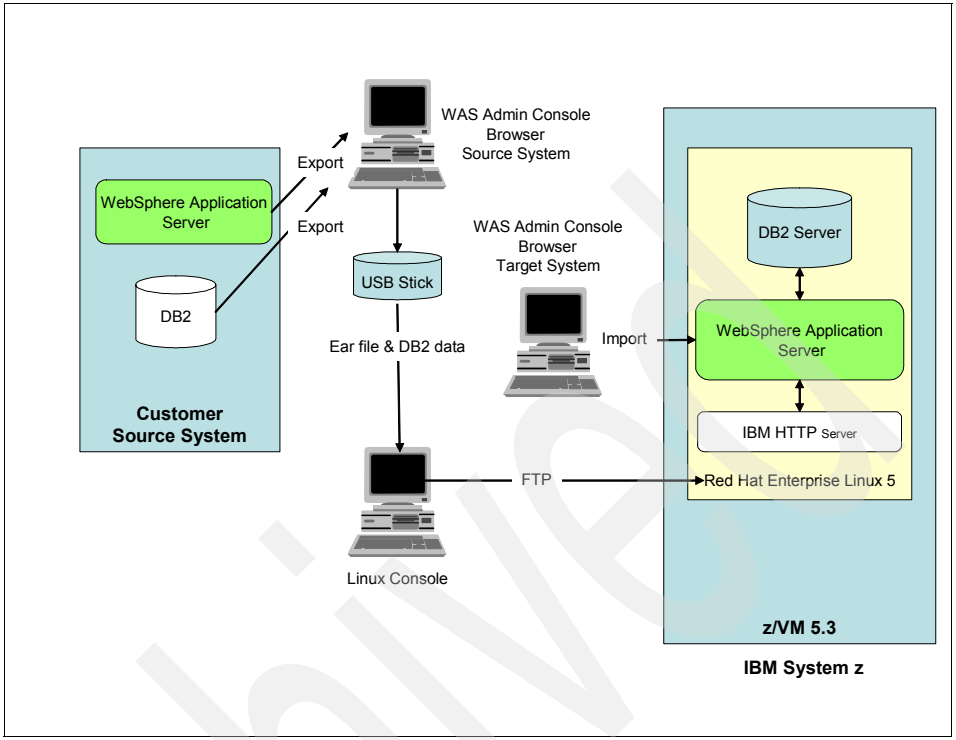

Figure 14-2 Linux proof of concept setup and application and data import/export

14.1.3 Tools used in proof of concept

The source system was the client's test and development system and was accessed by a client-owned MS Windows mobile computer. The target system was on a secure IBM network and was accessed by an IBM-owned mobile computer. A USB memory stick was used to move data between the two mobile computers.

For the Linux console we used PuTTY, a free Win32 Telnet and SSH client. This allowed access to the Linux system via its IP address.

To upload data, the free open source FTP Windows client Filezilla was used. Both the WebSphere Enterprise Archive (EAR) file and the DB2 data were transferred to the Linux system using Filezilla.

To access the WebSphere Administration Console on the target system, Internet Explorer was used by pointing to port 80 on the Linux server.

To run the proof of concept, Internet Explorer was used to access the customer's home page on the Linux system.

For more details about the remote access tools PuTTY and Filezilla, refer to Appendix B, "Remote access applications" on page 245.

14.2 WebSphere application migration

The client selected its Web-facing application ACE as the most appropriate application for a Linux on System z proof of concept. ACE was written in Java and executes in a WebSphere Application Server Version 6.0.27 cluster.

The process of migrating from one WebSphere Application Server environment to another is straightforward. The application was migrated using the source system's WebSphere Administrative Console **EXPORT** command. This created an EAR file on the client running the WebSphere Administrative Console under Internet Explorer. The EAR file was 66 MB in size and was written to a USB memory stick.

To load the EAR file to the target system, the USB memory stick was plugged into the IBM-owned mobile computer and the Filezilla FTP client was used to move the EAR file to the target system.

The installation of the application on the target WebSphere Application Server running on Red Hat Enterprise Linux 5 (RHEL5) was undertaken using the target system's WebSphere Administrative Console **IMPORT** command.

Both the export and import actions completed without incident.

14.3 DB2 database migration

IBM installed DB2 Enterprise Edition Version 8.1 on the RHEL5 system in the same virtual machine as WebSphere Application Server and the IBM HTTP Server. Migrating the database from the MS Windows environment (source system) to the Linux on System z environment (target system) consisted of the following tasks.

14.3.1 From the source system

Follow these to migrate the database from the source server:

1. Copy the **CREATE DATABASE** command used to create the database on the source system to the target system using Filezilla. Edit the file locations to values that are applicable to the target system. (14.6.1, “Create Database command” on page 172 provides details of the old and new file locations.)
2. Generate the data definition language (DDL) using the **DB2LOOK** command:

   ```
   db2look -d ACEPRD01 -e > ACEPRD01.ddl
   ```

3. From the MS Windows system, the target database locations require editing to specify the final destinations on the target system. (14.6.2, “DB2LOOK Output: Tablespace DDL statements” on page 172 provides more details.)
4. Export the data using the **DB2MOVE EXPORT** command.

   ```
   db2move ACEPRD01 export
   ```

   > **Note:** **DB2 Backup** cannot be used to move data between operating systems. You must use the **DB2MOVE EXPORT** command to export the data.

5. Ensure that output of **DB2MOVE** is directed to an empty directory.
6. Use zip to put all output from the **DB2MOVE command** into a single zipped file. In our case, this file was saved on a USB memory stick to allow upload to the Linux system.
7. FTP the zip file and the DDL output of **DB2LOOK** to the target system. FTP must be done in BINARY mode because extra characters will be added to the end of each line if transferred in ASCII. Because the DDL file is created on MS Windows, extra characters may be inserted when the file is created. If this does happen, it will be necessary to remove the extra characters using the technique displayed in Example 14-1 on page 169 on the target system.

   The **SED** command is shown in this example to remove control characters in a file transmitted in ASCII from Windows. This command is executed on the target system.

   When transmitted in ASCII from an MS Windows system, each line ends in x’0D (new line) and x’0A’ (carriage return) non-printable characters. The new line character causes the DB2 Command Line Processor to fail so this must be removed from the input file.

Example 14-1 Remove non-printable characters from an ASCII-transmitted file

```
sed 's/^M//g' INPUT_DDL.SQL > output.sql
```

To type the ^M character simultaneously, hold down the Ctrl and V keys while you type `M`. The command shown in Example 14-1 stores the corrected file under the name `output.sql` in the current directory.

After the file created by the **DB2MOVE EXPORT** command has been transferred to Linux, unzip the contents.

For this migration, the DB2 database backup was approximately 250 MB in size.

14.3.2 To the target system

Follow these steps to migrate to the target system:

1. The command examples listed in this chapter use the DB2 Command Line Processor. They could also have been submitted using the DB2 Command Processor graphical user interface (GUI).
2. Ensure all file locations have been updated to reflect final destinations; refer to 14.6, “Commands and output messages” on page 172 for details.
3. Under the DB2 administrator user ID, execute the **CREATE DATABASE** command with the same attributes as the source system. The command used in the DB2 Command Line Processor in this instance was:

   ```
   db2 -tvf create db CREATEDB_ACEPRD01.SQL -z createdb.log
   ```

4. After the database has been successfully created, create the tablespaces using the DDL file created by the **DB2LOOK command**. If the DDL file was transferred from Windows to Linux in ASCII mode, you will need to remove the new line characters from the end of each line in the file; refer to Example 14-1 for information.

   Ensure that all Windows file locations are updated to correct Linux file locations. Run DDL commands using the following command:

   ```
   db2 -tvf DDL.file.name -z ddl.log
   ```

   > **Note:** For the ACEPRD01 proof of concept, this process took approximately 40 minutes.

5. The first time the DDL SQL statements were run, the command failed with error message:

```
SQL0968C The file system is full
```

This occurred because there were two large table spaces `ACE_BLOB_TS01` and `ACE_BLOB_TS02`, each of 320,000 pages or 10 GB in size, which exceeded the size of the file system.

There are two ways to resolve this problem: either increase the file system size, or decrease the binary large object (BLOB) space allocation. We chose to decrease and reduced the BLOBs from 320,000 pages (10 GB) to 32,000 pages (1 GB) each. This reduced size had no impact on the proof of concept.

6. After the DDL file has been successfully processed, import the data into database by using DB2MOVE LOAD command as shown:

```
db2 db2move ACEPRD01 load
```

For the ACEPRD01 functional test, the database load took 20 minutes.

14.3.3 Errors encountered during data load

This section describes errors that we encountered and explains how we addressed them.

- The **DB2LOOK** command on the MS Windows system did not extract the SYSTOOLS schema. However, the SYSTOOLS schema was exported by the **DB2MOVE EXPORT** command because the SYSTOOLS table did exist on the MS Windows system. The missing SYSTOOLS schema resulted in the following message:

```
SQL3304N The table does not exist
```

  The SYSTOOLS schema is used for performance information and is created in response to a specific procedure such as GET_DBSIZE_INFO. This was not required for the proof of concept.

- The load for table ACE.CLNT rejected all 2324 rows due to the following messages:

```
SQL3550W  The field value in row "F0-2" and column "1" is not NULL,
but the target column has been defined as GENERATED ALWAYS.

SQL3185W  The previous error occurred while processing data from row
"F0-2" of the input file.
```

  These messages indicate that the row was not loaded because the contents of Column 1 had already been automatically generated by the system and were not NULL. As such all rows were rejected.

The fix for this was to load the data from the file created by the **EXPORT** command **tab137.ixf** with the following DB2 command:

```
db2 load from tab137.ixf of ixf modified by identifyoverride into
ACE.CLNT
```

As a result, the database ACEPRD01 was successfully created and migrated from the client's MS Windows system to the Red Hat Enterprise Linux 5 system running under z/VM Version 5.3.

For more details about the error messages received during the database load, refer to 14.6.3, "Output messages" on page 173.

14.4 Testing scenario and results

After the database had been migrated, minor modifications involving environmental variables were required to allow ACE to connect to the database and display the `/home/Version.jsp` and `/motor/Version.jsp` Web pages. The display of these two pages indicated that the application had been installed successfully.

The ACE application is the front-end quote engine for the client's insurance products. Testing involved lodging a number of requests for auto and home quotations and comparing these to similar requests on the production systems at the client's Web site.

The Linux system produced the same results as the production system in all cases. The only difference was that we could not save the quotes because the system was not linked to the production Customer Management System.

The client was very satisfied with the results of the test and this has reinforced the client's resolve to consolidate its MS Windows, WebSphere and DB2 environments to Linux on System z. The client cut over its production to Linux on System z approximately 10 months after this test.

The net result for the client was that 60 Intel servers were replaced with 3 IFLs on a IBM System z10 EC. This gave the client a more resilient system with a much smaller footprint, both in floor space and power consumption, as well as the potential for greatly reduced software costs.

14.5 Findings

The portability of the Java code meant that no changes were required to the application to execute successfully on IBM WebSphere running on Red Hat Enterprise Linux 5 on System z.

WebSphere applications from any platform would be ideal candidates for consolidation on Linux on System z. The main criteria are to ensure that the same versions of WebSphere and DB2 are used on both the source and target platforms.

14.6 Commands and output messages

This section shows the database commands use along with their resulting output.

14.6.1 Create Database command

This command created the database on the target system:

```
CREATE DATABASE ACEPRD01 ON '/home/db2inst1'
USING CODESET 1252 TERRITORY AU COLLATE
USING SYSTEM CATALOG TABLESPACE  MANAGED BY SYSTEM USING (
'/home/db2inst1/SYSCATSPACE_01' )
USER TABLESPACE  MANAGED BY SYSTEM USING
( '/home/db2inst1/USERSPACE_01' ) TEMPORARY
TABLESPACE  MANAGED BY SYSTEM USING
( '/home/db2inst1/SYSTEMPSPACE_01' ) ;
```

14.6.2 DB2LOOK Output: Tablespace DDL statements

When migrating from Ms Windows to Linux on System z, you must change all database locations from the MS Windows format to the Linux format. In our case, for all location references we changed from:

```
F:\db2data\' to '/home/db2inst1/'
```

Here we show the before and after DDL statements after adjusting for the correct file locations:

Before:

```
------------------------------------
-- DDL Statements for TABLESPACES --
```

```
------------------------------------

CREATE REGULAR TABLESPACE SYSTOOLSPACE IN DATABASE PARTITION
GROUP IBMCATGROUP PAGESIZE 4096 MANAGED BY SYSTEM
   USING ('F:\db2data\ACEPRD01\SYSTOOLSPACE')
   EXTENTSIZE 32
      PREFETCHSIZE AUTOMATIC
      BUFFERPOOL IBMDEFAULTBP
OVERHEAD 12.670000
TRANSFERRATE 0.180000
DROPPED TABLE RECOVERY ON;
```

After:

```
------------------------------------
-- DDL Statements for TABLESPACES --
------------------------------------

CREATE REGULAR TABLESPACE SYSTOOLSPACE IN DATABASE PARTITION
GROUP IBMCATGROUP PAGESIZE 4096 MANAGED BY SYSTEM
   USING ('/home/db2inst1/ACEPRD01/SYSTOOLSPACE')
EXTENTSIZE 32
PREFETCHSIZE AUTOMATIC
BUFFERPOOL IBMDEFAULTBP
OVERHEAD 12.670000
TRANSFERRATE 0.180000
DROPPED TABLE RECOVERY ON;
```

14.6.3 Output messages

Normal Completion for each table showed the following output:

```
* LOAD:  table "ACE"."HOME_EXTRACT"
-Rows read:     322790
-Loaded:        322790
-Rejected:           0
-Deleted:            0
-Committed:     322790

Error Messages for ACE.CLNT
```

14.7 Final configuration and results

The proof of concept was a success, and the client installed a Linux-only IBM System z10 EC in late 2008.

The first environments migrated were the test and development systems. The production system was migrated in May 2009. All migrations were done on schedule.

The final result was that the migration was a resounding success with significant cost savings in infrastructure, software licenses, simplification of the IT environment and the scalability and flexibility of the IBM System z10 solution.

Note: The existing IT environment of 6 WebSphere Application Servers servers running in 3 clusters and 6 DB2 database servers was consolidated to 2 Linux virtual machines running WebSphere Application Server and 1 Linux virtual machine DB2 database server.

This produced a significant simplification of the IT environment, with associated cost savings.

15

Technical procedures

In this chapter we discuss debugging tools you may find useful, as well as technical tips about on how to integrate Linux on System z with z/VM so as to allow z/VM CP commands to be issued from Linux.

We also discuss how to add virtual CPUs to a Linux virtual server without requiring a reboot of Linux.

15.1 Linux on System z console

z/VM and Linux on System z provide a console so you can debug system problems during initial program load (IPL) or network configuration problems that prohibit administrators from connecting to the Linux guest over the network.

Access to this console is only through 3270 console emulator terminal software (refer to “3270 Emulation” on page 246 for more details). The system administrator should log on to the z/VM operating system using the user ID related to the Linux guest to access its console.

Example 15-1 shows that the VM user ID is defined in the z/VM Directory. Line 02169, which starts with the `USER` string, contains the VM user ID id `LNXRH2` in the second field and the VM password `LNXRH2` in the third field. The other fields on the line are related to memory size of the virtual machine.

Line 02181 defines the type of virtual machine and the maximum number of virtual CPUs that could be set. In this example, a maximum of six virtual CPUs can be set.

Example 15-1 z/VM directory entry for Linux guest machine

```
02177 ******************************
02178 *
02179 USER LNXRH2 LNXRH2 1G 2G G
02180    IPL CMS PARM AUTOCR
02181    MACHINE ESA 6
02182    NICDEF C200 TYPE QDIO LAN SYSTEM VSWITCH1
02183    CONSOLE 0009 3215
02184    SPOOL 000C 3505 A
02185    SPOOL 000D 3525 A
02186    SPOOL 000E 1403 A
02187    LINK MAINT 0190 0190 RR
02188    LINK MAINT 019E 019E RR
02189    LINK MAINT 019F 019F RR
02190    LINK MAINT 019D 019D RR
02191    LINK TCPMAINT 0592 0592 RR
02192    MDISK 0191 3390 0241 40 LX2U1R MR
02193    MDISK 0201 3390 0001 1000 LXDE1E MR
02194    MDISK 0202 3390 1001 9016 LXDE1E MR
02195 *
02196 ******************************
02197 *
```

To log on to the z/VM console within this Linux guest, enter the VM user ID in the `USERID` field from a z/VM console Welcome window, as shown in Example 15-2.

Example 15-2 z/VM Welcome window

```
z/VM ONLINE



                    / VV         VVV MM      MM
                   /  VV       VVV  MMM     MMM
          ZZZZZZ  /   VV     VVV   MMMM   MMMM
             ZZ  /    VV   VVV    MM MM MM MM
            ZZ  /     VV  VVV     MM  MMM  MM
           ZZ  /      VVVVV      MM   M   MM
          ZZ  /       VVV       MM       MM
         ZZZZZZ /      V        MM       MM

         built on IBM Virtualization Technology



 Fill in your USERID and PASSWORD and press ENTER
 (Your password will not appear when you type it)
 USERID   ===>  lnxrh2
 PASSWORD ===>

 COMMAND  ===>
                                                RUNNING   VMLINUX2
```

Inside the Linux guest, all commands will be placed at the bottom of the z/VM console window. After you press the Enter key, the command is sent to the Linux guest for processing. Remember that for security reasons, the root and all other user passwords, when typed at a z/VM console, are not shown.

15.2 CP commands

During server configuration or during service troubleshooting, there are some tasks for which you need to issue certain z/VM CP commands to query the virtual devices or to configure some devices online as virtual CPUs or DASD devices.

There are two different ways to send these commands to the z/VM operating system:

- By using a 3270 emulator
- By using the vmcp Linux module

We explain these methods in more detail in the following sections.

3270 emulator

To send CP commands from a 3270 emulator, you need to log on at the Linux virtual machine using the VM user ID. Then issue the command in the command area of the 3270 terminal. You must issue the command using the **#CP** syntax; otherwise, the command will be sent to the Linux TTY and will return a Linux error.

Figure 15-1 illustrates the correct way to issue #CP commands. For more detailed information about the 3270 terminal, refer to “3270 Emulation” on page 246.

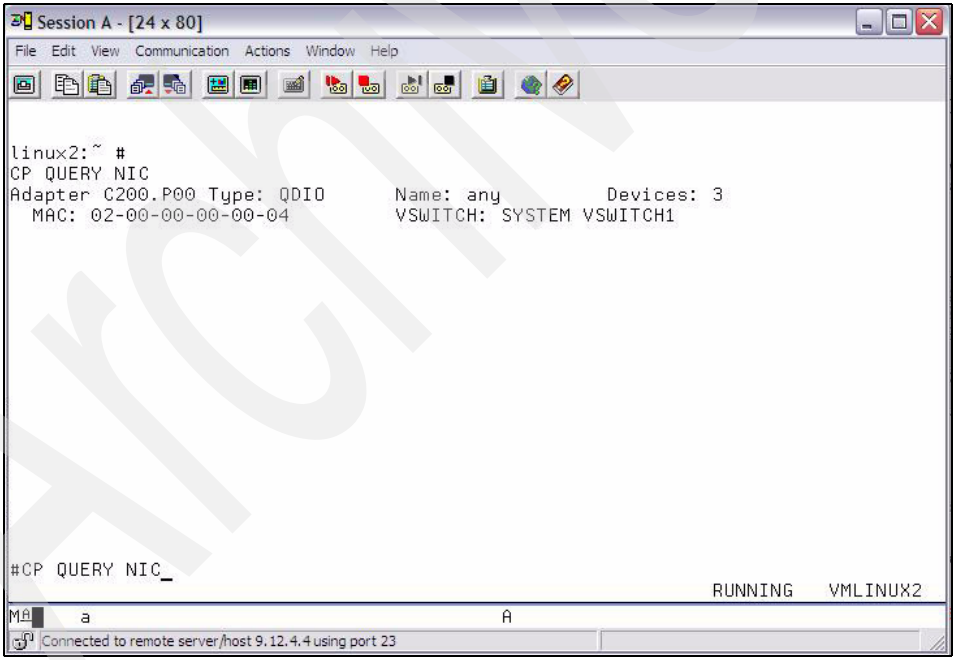

Figure 15-1 CP QUERY command issued at a 3270 terminal

vmcp module

Linux on System z runs a specific module called vmcp for virtual machine control program. This module provides a facility that allows you to issue CP commands

from a Linux ssh session. The vmcp module must be loaded for the vmcp commands to work; see Example 15-3.

Example 15-3 Loading the vmcp module

```
# modprobe vmcp
#lsmod
linux2:~ # lsmod
Module                  Size  Used by
vmcp                   24584  0
sg                     76608  0
sd_mod                 58408  0
sr_mod                 40748  0
scsi_mod              272776  3 sg,sd_mod,sr_mod
cdrom                  65072  1 sr_mod
apparmor               87352  0
loop                   39184  0
dm_mod                129824  0
qeth                  285360  0
ipv6                  494888  43
vmur                   34336  0
qdio                  125968  2 qeth
ccwgroup               27904  1 qeth
ext3                  245008  2
jbd                   117312  1 ext3
dasd_eckd_mod         107520  12
dasd_mod              135600  7 dasd_eckd_mod
```

After the vmcp module is loaded, you can issue the z/VM CP commands from the Linux session as shown in Example 15-4.

Example 15-4 CP QUERY command issued from Linux session

```
linux2:~ # vmcp query nic
Adapter C200.P00 Type: QDIO      Name: any         Devices: 3
  MAC: 02-00-00-00-00-04         VSWITCH: SYSTEM VSWITCH1
```

Tip: To understand how to configure Linux on System z to automatically log on, and for other useful configuration hints, refer to the following IBM Redbooks publications:

- For Red Hat Enterprise Linux

  z/VM and Linux on IBM System z, SG24-7492

- For SUSE Linux Enterprise Server

  z/VM and Linux on IBM System z - The Virtualization Cookbook for SLES 10 SP2, SG24-7493

15.3 Adding CPU dynamically

One strength of Linux on System z is that outage times are reduced because almost all configuration changes can be performed online. For example, you can add virtual CPUs to a Linux guest with all servers running.

However, you must perform the following configuration tasks to enable this facility.

1. Alter the default `/etc/zipl.conf` file configuration, shown in Example 15-5, to include the options that configure the kernel to allow new CPUs to come online without a server reboot.

Example 15-5 Default /etc/zipl.conf file

```
[root@linux6|~] cat /etc/zipl.conf
[defaultboot]
default=linux
target=/boot/
[linux]
        image=/boot/vmlinuz-2.6.18-92.el5
        ramdisk=/boot/initrd-2.6.18-92.el5.img
        parameters="root=LABEL=/1 "
```

2. Edit `/etc/zipl.conf` using vi or another editor to include the `additional_cpus=x` parameter, where `x` is the maximum value of additional virtual CPUs. The new `/etc/zipl.conf` file must look like Example 15-6.

Example 15-6 Changed /etc/zipl.conf file

```
[root@linux6|~] cat /etc/zipl.conf
```

```
[defaultboot]
default=linux
target=/boot/
[linux]
        image=/boot/vmlinuz-2.6.18-92.el5
        ramdisk=/boot/initrd-2.6.18-92.el5.img
        parameters="root=LABEL=/1 additional_cpus=6"
```

3. To put the changes you made in the configuration file `/etc/zipl.conf` into effect, issue the **`zipl`** command to write it on the IPL device and reboot the server, as shown in Example 15-7.

Example 15-7 The zipl and reboot steps

```
[root@linux6|~] zipl
Using config file '/etc/zipl.conf'
Building bootmap in '/boot/'
Building menu 'rh-automatic-menu'
Adding #1: IPL section 'linux' (default)
Preparing boot device: dasdb (0202).
Done.
[root@linux6|~] shutdown -r now

Broadcast message from root (pts/0) (Fri Jun 19 14:55:24 2009):

The system is going down for reboot NOW!
```

4. Issue the z/VM CP commands and Linux commands (after the server reboot) to dynamically add the CPU to the server (keeping in mind that you must set up the profile Guest Machine in z/VM with the correct MACHINE ESA parameters, as shown in Figure 15-1 on page 178).

   Example 15-8 shows the new server configurations available in Linux. The results from the first command (`cat /proc/cmdline`) return the information about the configuration when the kernel was initialized.

   The results from the second command (`ls /sys/devices/system/cpu/`) return the first level of the directory tree that handles the virtual CPUs device information in Linux.

Example 15-8 CPU information commands

```
[root@linux6|~] cat /proc/cmdline
root=LABEL=/1 additional_cpus=6 BOOT_IMAGE=0
[root@linux6|~] ls /sys/devices/system/cpu/
```

```
cpu0  cpu1  cpu2  cpu3  cpu4  cpu5  cpu6
```

5. Activate the CPU. All directories inside `/sys/devices/system/cpu/` have a file named `online` that receives the value zero (0) for CPUs not actived and the value 1 for an activated CPU .

   To activate the CPU, it must first be enabled on the VM side by using the commands shown in Example 15-9.

Example 15-9 Enabling additional virtual CPUs for a Linux guest on z/VM

```
[root@linux6|~] vmcp query cpus
CPU 00  ID  FF22DE5020978000 (BASE) CP   CPUAFF ON
[root@linux6|~] vmcp define cpu 01 <- Define the additional CPU
CPU 01 defined
[root@linux6|~] vmcp query cpus
CPU 00  ID  FF22DE5020978000 (BASE) CP   CPUAFF ON
CPU 01  ID  FF22DE5020978000 STOPPED CP   CPUAFF ON
```

6. After the CPU has been enabled in z/VM, you bring it online to the Linux server by using the commands shown in Example 15-10.

Example 15-10 Activating the CPU for a Linux guest

```
[root@linux6|~] cat /sys/devices/system/cpu/cpu1/online
0
[root@linux6|~] echo 1 >  /sys/devices/system/cpu/cpu1/online
[root@linux6|~] cat /sys/devices/system/cpu/cpu1/online
1
[root@linux6|~] vmcp query cpus
CPU 00  ID  FF22DE5020978000 (BASE) CP   CPUAFF ON
CPU 01  ID  FF22DE5020978000 CP   CPUAFF ON
```

15.4 Final considerations

The examples in this section explain how to access and use z/VM CP commands and Linux commands together to issue Linux on System z box configurations.

For more information about z/VM CP commands, refer to *CP Commands and Utilities Reference*, SC24-6081.

For more information about Linux on System z special application and device drivers, refer to *Linux on System z Device Drivers, Features, and Commands*, SC33-8411.

16

Example Sakai migration

In this chapter, we migrate an open source learning management system, Sakai, from a different operating environment to a target operating environment residing on Linux on System z. We follow the migration methodology discussed in Chapter 3, “Migration methodology” on page 31.

This sample migration demonstrates how preplanning helps in the proper execution of any migration effort. This chapter also explains the technical tasks involved in migrating and the methods used in solving issues when implementing Sakai from source on Linux on System z.

16.1 Our environment

In this scenario, we are looking at migrating the entire Sakai-based Learning Management System (LMS) from Sun Solaris to our Linux on System z.

Sun Solaris environment

This SPARC server was configured with three processors and three gigabytes of memory, as shown in Example 16-1.

Example 16-1 The hardware configuration

```
root@vclsun1:/:140: > prtdiag
System Configuration:  Sun Microsystems  sun4u Sun Enterprise 450 (3 X
UltraSPAR
C-II 400MHz)
System clock frequency: 100 MHz
Memory size: 3584 Megabytes

========================= CPUs =========================

                     Run   Ecache   CPU    CPU
Brd  CPU   Module   MHz     MB    Impl.   Mask
---  ---  -------  -----  ------  ------  ----
SYS    1     1      400     4.0   US-II    10.0
SYS    2     2      400     4.0   US-II    10.0
SYS    3     3      400     4.0   US-II    10.0
```

This server has Tomcat as the Web application server and Oracle 10.2.0.3 as the database management system. The version of Sun Solaris is displayed in Example 16-2.

Example 16-2 Sun Solaris version

```
root@vclsun1:/:133: > uname -na
SunOS vclsun1 5.10 Generic_127111-06 sun4u sparc SUNW,Ultra-4
```

Linux on System z

The target server was configured on z/VM as a guest image, with the hardware configuration shown in Example 16-3.

Example 16-3 Linux on System z hardware information

```
test19:~ # cat /proc/sysinfo
Manufacturer:         IBM
```

```
Type:                 2094
Model:                700
Sequence Code:        0000000000027F2D
Plant:                02
```

The memory information is displayed in Example 16-4.

Example 16-4 Linux on System z memory information

```
test19:~ # cat /proc/meminfo
MemTotal:      2047784 kB
```

The operating system was SLES 10 SP2, as shown in Example 16-5, with Tomcat as the application server and Oracle 10.2.0.3 as the database management system.

Example 16-5 Linux on System z version

```
ecmtest19:/opt/tomcat/bin # cat /etc/issue
Welcome to SUSE Linux Enterprise Server 10 (s390x)
Kernel 2.6.16.54-0.2.12-default running on an  s390x (under z/VM 5.3.0)
```

Note: This chapter discusses the Sakai migration from a Java application perspective. We discuss the various tool implementations used during migration and the actual Java problems we faced during this activity.

The chapter does *not* discuss the database installation or the database migration.

16.2 What is Sakai

Sakai is an open source-based, learning management system (LMS) or course management system. The Sakai application framework has a broader purpose and is designed to support the wide variety of collaboration and learning activities that take place within education and increasingly beyond. Sakai is a collaborative learning environment. It can be applied in both a teaching and learning environment, as well as in research collaboration.

Sakai is a framework with a collection of open source tools developed by a large number of universities to provide a supplementary learning environment suitable for on-campus courses, distance learning courses and internal training purposes.

Sakai is a powerful and flexible solution that supports not only teaching and learning but also research and administrative collaboration.

16.3 Using Sakai on Linux on System z

Using Linux on System z with Sakai offers the following benefits:

- Abilities (scalable, available, reliable)
- Mature virtualization platform
- Real time integration of transactions, information, and analytics
- Business-driven services
- Service-oriented perspective
- Centralized control and management
- Consolidation of hundreds to thousands of diverse workloads
- Most efficient cost per computing power

16.4 Migration planning

We assume that the stakeholders for this migration effort already have been identified and the business reasons for moving to Linux on System z have been accepted by the stakeholders. Even though the Sakai application is a Java-based open source project, there will be some dependencies and prerequisites that need to be considered.

16.4.1 Sakai application analysis

We started with a thorough review of the system perquisites for Sakai as derived from the Sakai project Web site. The findings are:

- The site was positioning Sun Microsystem's Java 2 Platform Standard Edition 5.0 as an platform for the compilation of Sakai.
- There were reported compilation issues with the latest release of Java Platform, Standard Edition 6 (Java SE 6), otherwise known as Java 1.6.
- Because Linux on System z does not support the Sun Java platform, we anticipated issues in the source codes referencing Sun Java-specific classes or features.
- Sakai requires an open source compilation tool called Maven.
- There were dependencies on the Tomcat versions that Sakai works with.
- Sakai can use the following databases: MySQL, Oracle 9i, and 10g.

- There was a need to download a compatible JDBC Driver for MySQL and Oracle.
- Irrespective of MySQL or Oracle, the database needs to be configured to use the UTF-8 character set.

Application Planning checklist

We referred to Chapter 4, “Migration Planning checklist” on page 39 for guidance and filled in the Application Planning checklist as shown in Table 16-1.

Table 16-1 Application Planning checklist

Item	Version	Source Available	Licence/Free	Linux on System z
SAKAI	2.5.	http://sakaiproject.org/portal	Freeware	Available / Recompile
Java 2 Platform Standard Edition 5.0	1.5	http://www.ibm.com/developerworks/java/jdk/linux/download.html	Freeware	IBM JAVA Available
Oracle	10.2.0.3	http://www.oracle.com	License	Available
Apache Maven	2.1.0	http://maven.apache.org/download.html	Freeware	Available
Linux Oracle JDBC		www.oracle.com	Freeware	Available
TOMCAT	5.5.26	http://archive.apache.org/dist/tomcat/tomcat-5/	Freeware	Available

From the checklist, it is evident that most of the tools used for Sakai migration are also available on Linux on System z. We also noted that there is a slight variation in the Java Virtual Machine (JVM) being used.

The severity of the migration would now depend on the application’s dependence on the Sun Java-specific classes and features. Usually, open source applications

take this difference into consideration and write code that does not depend on any specific Java environment. The migration effort can be slightly greater if the application is written to be more specific to a particular Java environment. Keeping this in mind, we decided to proceed with a proof of concept before starting the actual migration.

16.5 Our proof of concept

As discussed in 3.1.5, “Pilot proof of concept” on page 35, the main objective of this proof of concept (PoC) is to focus on the identified areas of risk and document these risks as discussed in 16.4.1, “Sakai application analysis” on page 188. Additionally, we needed to recommend approaches to mitigate the risks to prove that the full migration can be completed successfully.

We were looking for a useful Sakai component or module to be tested on the Linux on System z operating environment. The Sakai community has the Sakai Demo package, which is a prebuilt version of Sakai with Apache Tomcat and other simple configurations. This demo package was an ideal candidate for our proof of concept, so we decided to implement it.

Tip: The Sakai Demo package is an ideal candidate for testing Sakai on a new operating environment (operating system and hardware). It can be downloaded from the following address:

http://sakaiproject.org/portal/site/sakai-downloads

Because all the programs and databases were already packaged with the demo, we downloaded and simply decompressed the Sakai Demo file, as shown in Example 16-6.

Example 16-6 Uncompressing the Sakai Demo

```
test19:/root2/sakai-demo-2.5.4 # tar -zxvf sakai-demo-2.5.4.tar.gz
```

We followed the documentation that explained how to start a Sakai Demo project.

We found the documentation by examining the demo directory listing as shown in Example 16-7.

Example 16-7 Sakai Demo directory listing

```
test19:/root2/sakai-demo-2.5.4 # ls
```

```
ECLv1.txt  RELEASE-NOTES  common     demo_readme.txt  sakai
shared          stop-sakai.bat  webapps
LICENSE    RUNNING.txt    components  licenses        sakai_readme.txt
start-sakai.bat  stop-sakai.sh   work
NOTICE     bin            conf        logs            server
start-sakai.sh   temp
```

The first step was to set the environmental variables for the IBM Java, as shown in Example 16-8.

Example 16-8 Setting environmental variables for Java

```
test19:~> export JAVA_HOME=/opt/ibm/java2-s390x-50
test19:~> export PATH=$PATH:$JAVA_HOME/bin
```

In the demo package, under the master directory, there is a **start-sakai.sh** wrapper script written around the actual Tomcat startup script. This script sets up the respective environmental variables of Java, Tomcat, and Sakai before starting Tomcat to load Sakai; see Example 16-9.

Example 16-9 Sakai Demo startup

```
test19:/root2/sakai-demo-2.5.4 # ./start-sakai.sh &
Using CATALINA_BASE:   /root2/sakai-demo-2.5.4
Using CATALINA_HOME:   /root2/sakai-demo-2.5.4
Using CATALINA_TMPDIR: /root2/sakai-demo-2.5.4/temp
Using JRE_HOME:       /opt/ibm/java2-s390x-50
:
:
:
:
INFO: ContextListener: contextInitialized() (2009-06-17 16:36:40,765
main_org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/se
rvlets-examples])
INFO: SessionListener: contextInitialized() (2009-06-17 16:36:40,766
main_org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/se
rvlets-examples])
INFO: Starting Coyote HTTP/1.1 on http-8080 (2009-06-17 16:36:41,209
main_org.apache.coyote.http11.Http11BaseProtocol)
INFO: JK: ajp13 listening on /0.0.0.0:8009 (2009-06-17 16:36:42,091
main_org.apache.jk.common.ChannelSocket)
INFO: Jk running ID=0 time=0/23  config=null (2009-06-17 16:36:42,097
main_org.apache.jk.server.JkMain)
```

```
INFO: Find registry server-registry.xml at classpath resource
(2009-06-17 16:36:42,188
main_org.apache.catalina.storeconfig.StoreLoader)
INFO: Server startup in 312019 ms (2009-06-17 16:36:42,243
main_org.apache.catalina.startup.Catalina)
```

The Sakai Demo package started to load. We were able to access the Sakai portal, shown in Figure 16-1, by accessing the following URL:

http://<hostname>:8080/portal

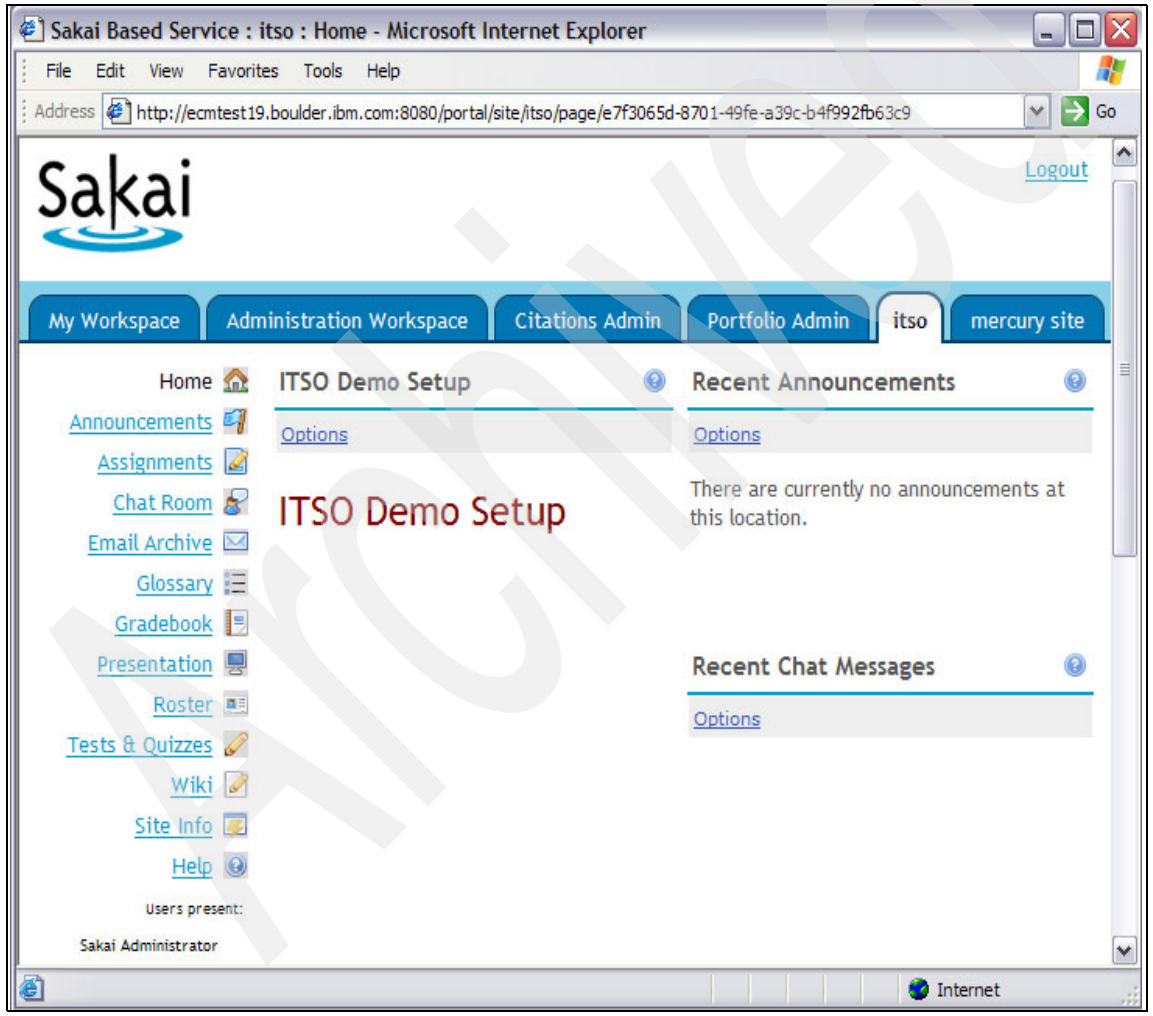

Figure 16-1 Sakai Demo portal

We analyzed the logs of Sakai to check that everything was working correctly. We verified the various learning management tools that are packaged with the Sakai Demo, and performed a parallel analysis of the Sakai application logs at the same time. While performing a verification of a particular tool known as the Gradebook, we encountered a significant number of errors; see Figure 16-2.

Examining these errors, our Java developers believed that they occurred because the Sakai Demo package would have been compiled in a Sun Java platform and we were trying to execute in an IBM-based Java Virtual Machine environment.

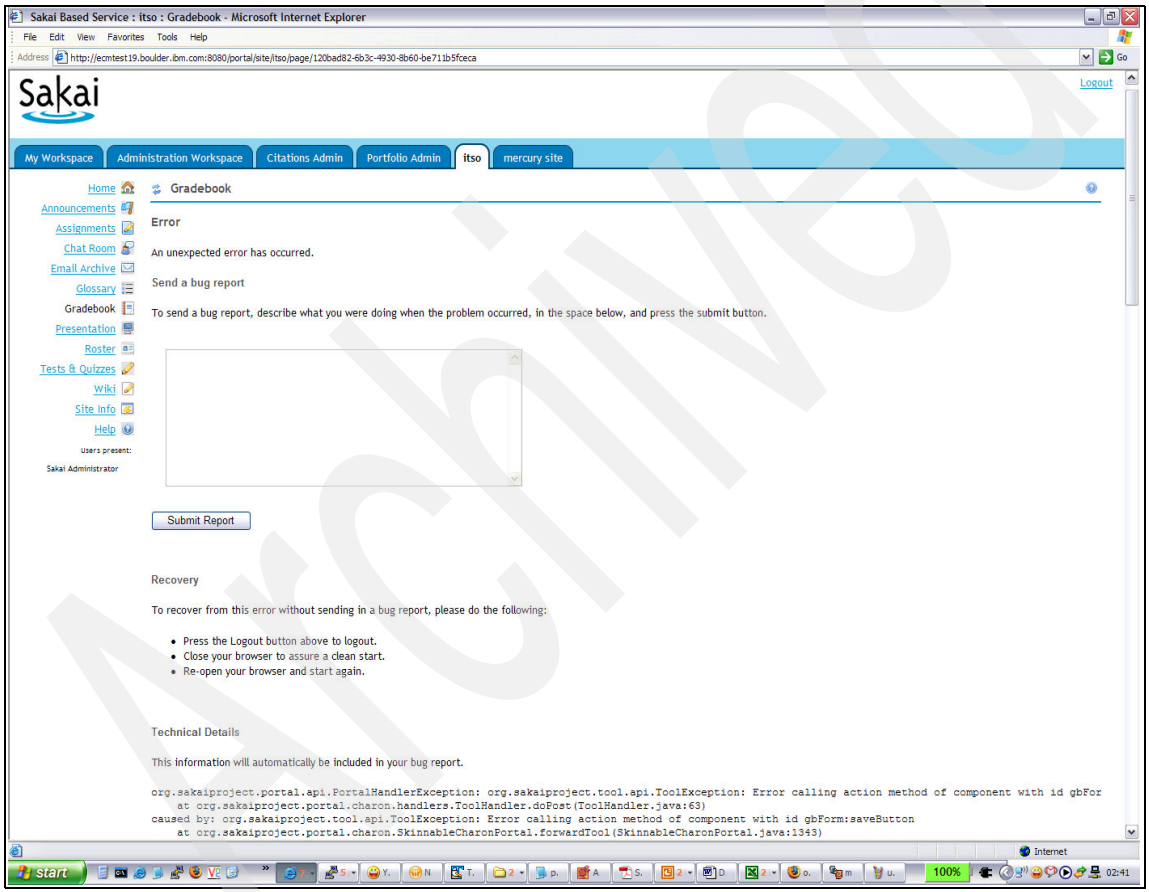

Figure 16-2 Our Sakai Demo Gradebook issue

16.5.1 Findings during the proof of concept

Our observations during this proof of concept included the following points:

- The Sakai Demo package was implemented easily without any operating environment changes (except the Java platform). The proof of concept was quick and easy.
- Most of the tools packaged in the Sakai Demo worked perfectly, which meant that the tools were not developed for any specific Java platform or environment.
- A few changes in the code might be required, where the application references Sun Java-specific classes or features.

16.5.2 Proof of concept outcome

The proof of concept addressed most of the issues outlined in 16.4.1, “Sakai application analysis” on page 188. It was also evident from the POC that the actual migration can be done with minimum code and operating environment changes.

16.6 Performing the migration

After we had become familiar with the Sakai application during the proof of concept, we began downloading the compressed Sakai source from the Sakai Web site. Referring to our “Application Planning checklist” on page 189, we started to download each of the tools listed in the checklist to the target Linux on System z environment.

> **Note:** There are a variety of Sakai implementation documents available on the Internet, many of which are specific to various operating environments. We recommend using this documentation, which is available from the Sakai Web site, for guidance.

We downloaded the IBM Java 1.5 (64-bit) version from the IBM site as per our Application Checklist and started our installation, as demonstrated in Example 16-10.

Example 16-10 Installing IBM Java 1.5

```
test19:~ # rpm -ivh java5_zlinux_s390x.rpm
```

We then set the Java environmental variables so that they were all pointing to correct Java installation directories, as shown in Example 16-11.

Example 16-11 Exporting Java environment variables

```
test19:~> export JAVA_HOME=/opt/ibm/java2-s390x-50
test19:~> export PATH=$PATH:$JAVA_HOME/bin
```

16.6.1 Installing Tomcat

Apache Tomcat provides an ideal environment for running Sakai as a Web application. Tomcat implements and conforms to both the Java Servlet and JavaServer Pages (JSP) specifications, and can be run in stand-alone mode or in conjunction with a Web application server such as the Apache HTTP server or JBoss.

As per the Sakai recommendations, Sakai works well with Tomcat version 5.5.26. Tomcat can be downloaded as an binary install archive from the following Web site:

http://archive.apache.org/dist/tomcat/tomcat-5/

Unpack the Tomcat archive into your installation directory of choice (for example, `/opt/`). Also create a symbolic link to the unpacked directory to simplify the path.

Example 16-12 Unpacking Tomcat binary install and creating a symbolic ink

```
test19:/opt # tar -zxvf apache-tomcat-5.5.26.tar.gz
ln -s apache-tomcat-5.5.26 tomcat
```

The base Tomcat directory (for example, `/opt/apache-tomcat-5.5.26`) is referred to as $CATALINA_HOME. For convenience, we can set $CATALINA_HOME as an environment variable, as shown in Example 16-13.

Example 16-13 Setting the Tomcat environmental variable

```
export CATALINA_HOME=/opt/tomcat
export PATH=$PATH:$CATALINA_HOME/bin
```

Note: Because most of the environmental variables have to be set up before the tools are accessed, it is advisable to modify startup files like `~/.bash_login ot .profile` to set and export shell variables.

Because Tomcat will be loading the Sakai application, we recommend that you set the memory for smooth operations. The standard way to control the JVM

options when Tomcat starts up is to have an environment variable `JAVA_OPTS` defined with JVM startup options.

After you set `JAVA_OPTS`, Tomcat will see this environment variable upon startup and use it; see Example 16-14.

Example 16-14 Export java options

```
export JAVA_OPTS="-server -XX:+UseParallelGC -Xmx768m
-XX:MaxPermSize=160m -Djava.awt.headless=true"
```

The examples perform two tasks:

- Instruct the virtual machine to use the Java HotSpot Virtual Machine
- Enable parallel garbage collection using the parameter `+UseParallelGC`

Java product version number 5.0 and developer version number 1.5.0 defines a class of machines referred to as *server-class* machines. These machines have two or more physical processors and 2 GB or more of physical memory, as well as adequate heap size. The parameter `MaxPermSize` sets the size of the Permanent Generation to accommodate more, longer-persisting objects.

16.6.2 Installing Maven

Apache-Maven is a popular open source build tool for Java projects, and it is designed to make the Java build process easy. Sakai uses the Maven build tool to compile and deploy its project modules. Maven can be downloaded from the following Web site:

http://maven.apache.org/download.html

Unpack the downloaded apache-maven distribution archive into your directory of choice (for example, `/opt/apache-maven-2.1.0`); see Example 16-15.

Example 16-15 Unpacking Maven and exporting the environmental values

```
test19:/root2 # tar -jxvf apache-maven-2.1.0-bin.tar.bz2
export MAVEN_HOME=/opt/apache-maven-2.1.0
export PATH=$PATH:$MAVEN_HOME/bin
```

To ensure sufficient memory allocation during builds, add a MAVEN_OPTS environment variable, as defined in Example 16-16 on page 197.

Example 16-16 Exporting Maven options

```
export MAVEN_OPTS='-Xms256m -Xmx512m -XX:PermSize=64m
-XX:MaxPermSize=128m'
```

Setting up Maven

To begin setting up Maven, create a local Maven repository (.m2) in your home directory, as shown in Example 16-17.

Example 16-17 Creating a Maven repository

```
cd $HOME
mkdir -p .m2/repository
```

Under the `.m2` directory, create the `settings.xml`. The settings element in the `settings.xml` file contains elements used to define values that configure Maven execution. These include values such as the local repository location, alternate remote repository servers, and authentication information. We changed the application server base location to `/opt/tomcat`; see Example 16-18.

Example 16-18 Created settings.xml under the .m2 directory

```
<settings xmlns="http://maven.apache.org/POM/4.0.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
                       http://maven.apache.org/xsd/settings-1.0.0.xsd">
  <profiles>
    <profile>
      <id>tomcat5x</id>
      <activation>
        <activeByDefault>true</activeByDefault>
      </activation>
      <properties>
        <appserver.id>tomcat5</appserver.id>
        <appserver.home>/opt/tomcat</appserver.home>
        <maven.tomcat.home>/opt/tomcat</maven.tomcat.home>
        <sakai.appserver.home>/opt/tomcat</sakai.appserver.home>
        <surefire.reportFormat>plain</surefire.reportFormat>
        <surefire.useFile>false</surefire.useFile>
      </properties>
    </profile>
  </profiles>
</settings>
```

In our case, we now verify that we can start Maven by executing the command **mvn --version** from the command prompt. This should start Maven and cause it to report its version; see Example 16-19.

Example 16-19 Verification of Maven execution

```
test19:/opt # mvn --version
Apache Maven 2.1.0 (r755702; 2009-03-18 15:10:27-0400)
Java version: 1.5.0
Java home: /opt/ibm/java2-s390x-50/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux" version: "2.6.16.54-0.2.12-default" arch: "s390x"
Family: "unix"
```

Now that we had the Maven environment ready, we downloaded the latest Sakai source code from the Sakai Web site and unpacked it in to a directory of choice; see Example 16-20.

Example 16-20 Decompressing the source code

```
test19:/opt/manoj # tar -zxvf sakai-src-2.5.4.tgz
```

16.7 The build process

To build Sakai from the source code, we needed to use the Apache-Maven tool. There are three steps in the process: clean, install, and deploy. These steps, otherwise known as Maven goals, perform the following actions:

- **clean** flushes the target for each Sakai project module prior to building.
- **install** installs the output of each Sakai project module into the repository.
- **sakai:deploy** deploys Sakai project modules from the repository to the target `$CATALINA_HOME/webapps` folder.

Sakai uses the Project Object Model (`POM.xml`) to provide an XML representation of basic project information covering dependency management, build targets, and so on. The POM contains all necessary information about a project, as well as configurations of plugins to be used during the build process.

The master `pom.xml` file is located in the base project directory, and it acts as a parent for other `pom.xml` files included in other Sakai subproject modules; see Example 16-21 on page 199.

Example 16-21 The pom.xml file

```
ecmtest19:/opt/manoj/sakai-src-2.5.4 # ls -la pom.xml
-rw-r--r-- 1 501 wheel 8368 Feb  5 12:21 pom.xml
```

During the initial build process, Maven will download any missing dependencies into your local repository, and then attempt to compile the Sakai code.

This process will take few minutes as all the missing dependencies for the compilation are downloaded; see Example 16-22.

Example 16-22 Locating dependencies

```
ecmtest19:/opt/manoj/sakai-src-2.5.4 # mvn clean install
[INFO] Scanning for projects...
```

16.7.1 Java symbol reference issue

In our case, during the initial build process, we encountered a build failure due to error relating to an referenced symbol in a particular Java source file; see Example 16-23.

Example 16-23 Build failure

```
[ERROR] BUILD FAILURE
[INFO]
------------------------------------------------------------------------
[INFO] Compilation failure

/opt/manoj/sakai-src-2.5.4/entitybroker/impl/src/java/org/sakaiproject/
entitybroker/impl/util/MapConverter.java:[20,55] package
com.sun.org.apache.xalan.internal.xsltc.runtime does not exist

/opt/manoj/sakai-src-2.5.4/entitybroker/impl/src/java/org/sakaiproject/
entitybroker/impl/util/MapConverter.java:[38,28] cannot find symbol
symbol  : class Hashtable
location: class org.sakaiproject.entitybroker.impl.util.MapConverter
```

We tried opening the Java file and discovered that there were references to Sun Java-specific classes in the source code. Because we used IBM Java, these classes would not be accessible; see Example 16-24 on page 200.

Example 16-24 Sun Java class imports build error

```
vi entitybroker/impl/src/java/org/sakaiproject/entitybroker /impl/util/MapConverter
.java

package org.sakaiproject.entitybroker.impl.util;

import java.util.HashMap;
import java.util.Map;

import com.sun.org.apache.xalan.internal.xsltc.runtime.Hashtable;
import com.thoughtworks.xstream.converters.Converter;
import com.thoughtworks.xstream.converters.MarshallingContext;
import com.thoughtworks.xstream.converters.UnmarshallingContext;
```

By searching the Internet, we determined that the same issue had been previously encountered and a patch was released by the Sakai community. We downloaded the patch from the Sakai Web site and applied it to the Sakai source.

In summary, we found out that the patch needs to applied on the Sakai source. This patch does the following:

- Replaces the references of internal Sun packages in entitybroker and the contrib tool Melete with an generic package reference.
- It also explicitly configures entitybroker to reference the xalan 2.6 library, instead of the implementation provided by the IBM JDK.

Example 16-25 displays applying the patch needed to resolve the symbol reference issue.

Example 16-25 Applying patch to solve the symbol reference issue

```
test19:/opt/manoj/sakai-src-2.5.4 # ls -la *.patch
-rw------- 1 root root 2108 Jun 18 13:40 ibm-no-internal-xalan.patch

ecmtest19:/opt/manoj/sakai-src-2.5.4 # patch -p0 < ibm-no-internal-xalan.patch
patching file entitybroker/impl/src/java/org/sakaiproject/entitybroker/impl/util/
MapConverter.java
```

Upon rebuilding the Sakai source, the symbol reference issue was solved and the building process continued, but it failed during automatic testing of the built Java modules.

XALAN/XERES

Apart from applying the patch, if you are building Sakai with the IBM JDK, the `JAVA_HOME/jre/lib/endorsed/` directory must be created and a copy of `xalan-2.6.0.jar` must be placed in it for all JUnit tests to pass. This tells the IBM JDK to use the XSL implementation provided by the Xalan library instead of its own implementation; see Example 16-26.

Example 16-26 Creating the directory structure for the xalan jar file

```
test19:~ # find . -name "xalan*.jar"

/root2/.m2/repository/xalan/xalan/2.6.0/xalan-2.6.0.jar

test19:~ # mkdir java2-s390x-50/jre/lib/endorsed

cp /root2/.m2/repository/xalan/xalan/2.6.0/xalan-2.6.0.jar /opt/ibm/java2-s390x-50/
jre/lib/endorsed/.
```

According to one of the Internet posts, test failures were occurring because of jcr unit testing, which has an xml namespace issue. We recommend that you skip these tests during initial build.

For this reason, we began to build the source in three stages:

1. Build Sakai without any unit tests.
2. Remove the unit test reference from `pom.xml`.
3. Deploy Sakai with unit tests.

Example 16-27 shows the commands used to build the Sakai source from scratch; however, we disabled the automatic unit testing of the project modules.

Example 16-27 Sakai build without unit testing

```
test19:/opt/manoj/sakai-src-2.5.4 # mvn -Dmaven.test.skip=true clean install
[INFO] Scanning for projects...
:
:
:
[INFO] Reports Tool ........................................ SUCCESS [3.115s]
[INFO] sakai-reports-util .................................. SUCCESS [1.248s]
[INFO] Reset pass .......................................... SUCCESS [6.028s]
[INFO] Sakai ............................................... SUCCESS [0.267s]
[INFO] Sakai Reset Pass Help ............................... SUCCESS [0.276s]
[INFO] ------------------------------------------------------------------------
[INFO] ------------------------------------------------------------------------
```

```
[INFO] BUILD SUCCESSFUL
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 24 minutes 19 seconds
[INFO] Finished at: Thu Jun 18 15:50:06 EDT 2009
[INFO] Final Memory: 185M/409M
[INFO] ------------------------------------------------------------------------
```

We removed the references of jcr unit tests from the master `pom.xml` file located at the Sakai master source directory.

We could then deploy the Sakai application using Maven. In this case we had Maven conduct the unit tests when it deployed the application; see Example 16-28. We also needed to specify the location where Maven was to deploy the application. Usually it would be Tomcat's base directory. The Maven command line parameter **-Dmaven.tomcat.home** specifies Tomcat's location; it can be omitted if Tomcat home is specified in Maven's `settings.xml` file.

Example 16-28 Rebuilding with tests as well as deploying the Sakai application

```
test19:/opt/manoj/sakai-src-2.5.4 # mvn clean install sakai:deploy -Dmaven.tomcat.
home=/opt/tomcat
[INFO] Scanning for projects...
[INFO] Reactor build order:
[INFO]   Sakai Master
[INFO]   Sakai Core Project
[INFO]   Sakai Velocity Project
:
:
[INFO] sakai-reports ..................................... SUCCESS [0.038s]
[INFO] Reports API ....................................... SUCCESS [2.140s]
[INFO] Reports Components Shared ......................... SUCCESS [3.155s]
[INFO] Reports Components ................................ SUCCESS [1.163s]
[INFO] Reports Tool ...................................... SUCCESS [7.859s]
[INFO] sakai-reports-util ................................ SUCCESS [1.308s]
[INFO] Reset pass ........................................ SUCCESS [3.092s]
[INFO] Sakai ............................................ SUCCESS [0.027s]
[INFO] Sakai Reset Pass Help ............................. SUCCESS [0.179s]
[INFO] ------------------------------------------------------------------------
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESSFUL
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 37 minutes 28 seconds
[INFO] Finished at: Thu Jun 18 16:51:53 EDT 2009
[INFO] Final Memory: 189M/369M
```

```
[INFO] ------------------------------------------------------------------------
```

16.7.2 Starting the Sakai application

Now that the build, testing, and deployment of the Sakai application was complete, we started Tomcat which in turn would load the deployed Sakai application; see Example 16-29.

Example 16-29 Starting Tomcat

```
test19:/opt/tomcat/bin # ./startup.sh &
```

During the startup of Tomcat, various Sakai application modules were loaded. Examine the Tomcat logs for any possible errors; see Example 16-30.

Example 16-30 Example Tomcat logs

```
test19:/opt/tomcat/logs # tail -f catalina.out
INFO: SessionListener: contextInitialized() (2009-06-18 17:16:45,329
main_org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/servlets-example
s])
INFO: Starting Coyote HTTP/1.1 on http-8080 (2009-06-18 17:16:45,863
main_org.apache.coyote.http11.Http11BaseProtocol)
INFO: JK: ajp13 listening on /0.0.0.0:8009 (2009-06-18 17:16:46,527
main_org.apache.jk.common.ChannelSocket)
INFO: Jk running ID=0 time=0/279  config=null (2009-06-18 17:16:46,529
main_org.apache.jk.server.JkMain)
INFO: Find registry server-registry.xml at classpath resource (2009-06-18
17:16:46,712 main_org.apache.catalina.storeconfig.StoreLoader)
INFO: Server startup in 197084 ms (2009-06-18 17:16:46,777
main_org.apache.catalina.startup.Catalina)
```

The server started without any error messages and was accessible through the URL:

http://<ipaddress>:8080/portal

Figure 16-3 on page 204 shows our successful load of the Sakai system on Linux on System z.

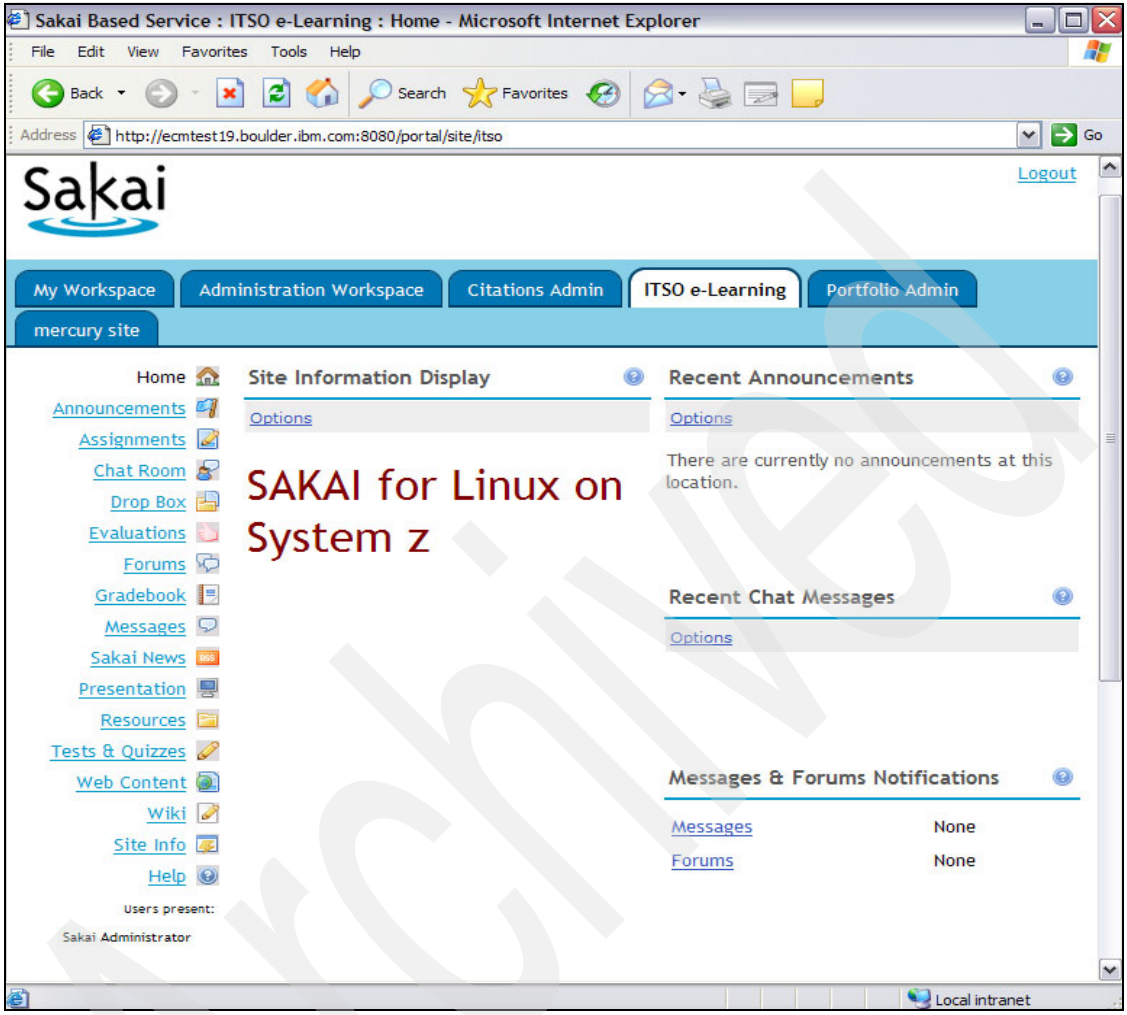

Figure 16-3 Successful load of Sakai on Linux on System z

16.8 Verification

As discussed in 3.1.9, “Verification testing” on page 37, this testing is designed to provide objective evidence that the testing requirements are met on the target operating environment.

First, we needed to ensure that the issue we encountered during the proof of concept phase has been rectified. Upon verification we found that the issue with

the Gradebook modules had been rectified. We performed extensive testing to ensure the proper functioning of all the Sakai application modules.

Because the verification testing proved that Skai was functioning as expected, we considered the migration to Linux on System z to have been a success.

16.9 Lessons learned

The key to a successful migration of Java across heterogeneous platforms is to conduct a proof of concept to identify errors associated with different Java implementations.

As shown in the Sakai proof of concept, we quickly identified issues with Java classes that are not common across different Java implementations.

17

Open source: A MediaWiki migration

This chapter demonstrates a migration of the open source application MediaWiki from two different source systems to the target system residing on Linux on System z.

In this example, an incomplete analysis of the requirements of the source and target system environments was done. As a result of this lack of effective planning, we expect to experience migration problems, even though this is a small and comparatively simple system to migrate.

In the first section, we create the target system environment. We describe the installation of MediaWiki on a z/VM guest running Linux on System z. Depending on your distribution, you can refer to *Setting up a Web 2.0 Stack on SUSE Linux Enterprise Server 10 SP2*, or to *Setting up a Web 2.0 stack on Red Hat Enterprise Linux 5.2*, for information about installing the components of the Web 2.0 stack. Next, we explain how to install MediaWiki and then how to migrate data from MediaWiki from an external platform to MediaWiki on System z.

The publication listed here use the command line interface (CLI) to perform the installation of the Web 2.0 stack, which is a completely acceptable way to install software packages. However, we also illustrate how to use the SLES installation tool YaST to install the Web 2.0 stack. We only install the components required for the installation and use of MediaWiki.

17.1 YaST setup of the installation source

YaST requires that you first configure where the installation source is and how you are going to access it. Figure 17-1 displays the first window shown after you invoke YaST from a PuTTY session.

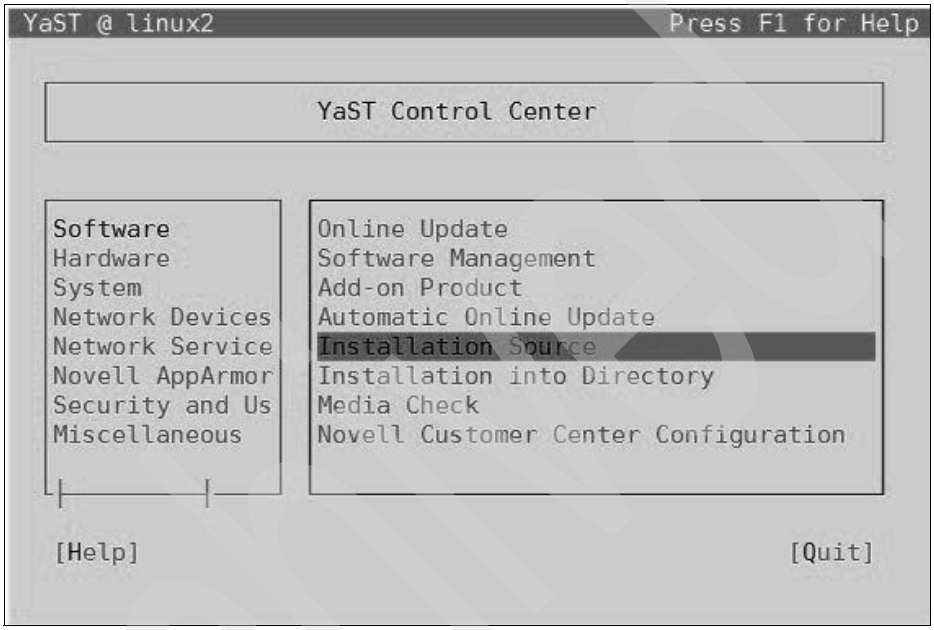

Figure 17-1 First panel to configure Installation Source

Use the Tab key to position highlighted text in the box on the right. (The Tab key will continue to scroll through the possible selections by highlighting them.) Use the up/down arrow keys to position the highlighted text within the box on the right. Highlight **Installation Source** and press the Enter key to reach the window shown in Figure 17-2 on page 209.

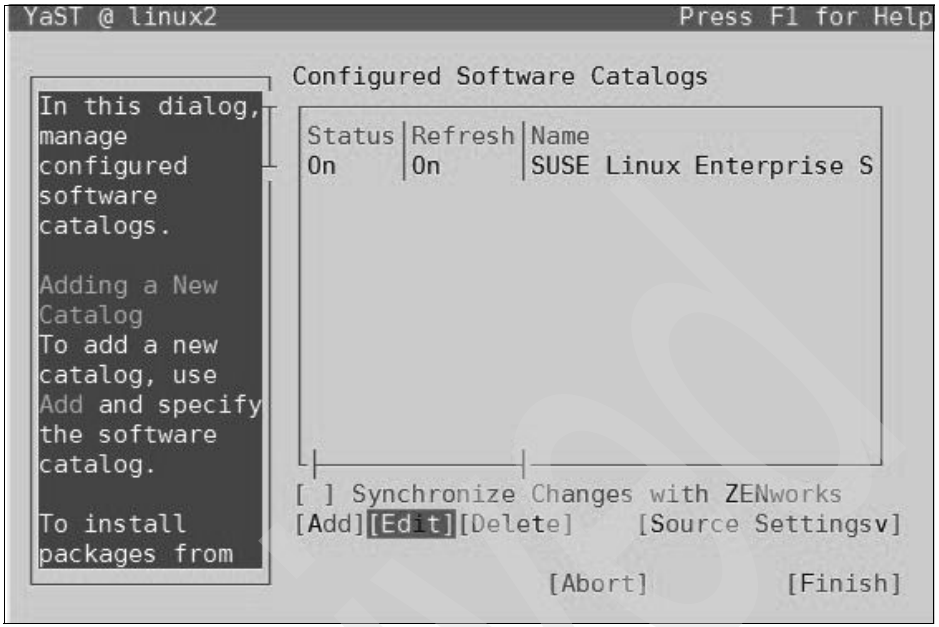

Figure 17-2 Select Add or Edit installation source

Use the Tab key to position over the check box for Synchronize Changes with ZENworks, then press the Tab key to de-select this option. The x should disappear from the box after you press the Spacebar.

Use the Tab key to select **Add** or **Edit**, then press Enter. The window shown in Figure 17-3 on page 210 will display.

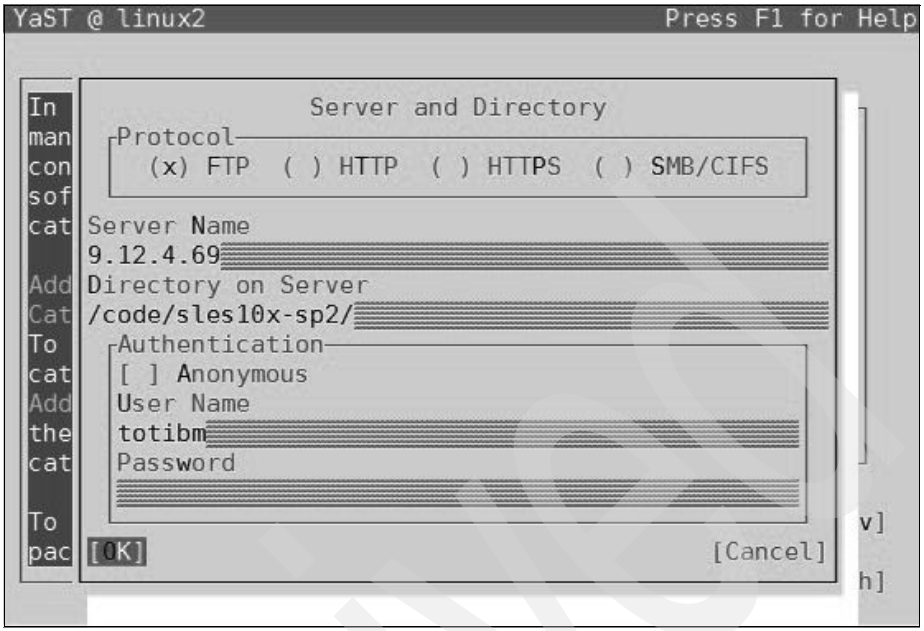

Figure 17-3 Software catalog server and directory

Use the Tab key to position over the protocol you will use and press the Spacebar to select that option. An `x` will appear in the box you select. FTP is a common method to access distribution files, and it is the protocol that we selected.

Tab to `Server Name` and enter an IP address or a host name that can be accessed through DNS. Next, tab down to `Directory on the Server`. This could be a directory for files or the location of a distribution iso. Make sure that you begin with a forward slashmark (/) and also end with a forward slashmark (/).

Finally, tab down to select or de-select the `Anonymous` option. If Anonymous is de-selected, then you have to supply a valid User Name and a Password. This is a user and password known on the FTP server.

17.2 Installing the LAMP stack

To begin the installation of additional software with YaST, return to the initial window and select **Software Management**, as shown in Figure 17-4 on page 211.

```
YaST @ linux2                                          Press F1 for Help

                          YaST Control Center

  Software                 Online Update
  Hardware                 Software Management
  System                   Add-on Product
  Network Devices          Automatic Online Update
  Network Services         Installation Source
  Novell AppArmor          Installation into Directory
  Security and Users       Media Check
  Miscellaneous            Novell Customer Center Configuration

  [Help]                                                    [Quit]
```

Figure 17-4 First panel to install software packages

After Software Management is highlighted, press Enter. The window shown in Figure 17-5 will be displayed.

```
YaST @ linux2                                          Press F1 for Help

[Filterv]        [Actionsv]        [Informationv]        [Etc.v]

     |Name         |Avail. Vers.|Inst. Vers.|Summary
  i  |xscreensaver |4.23        |4.23       |A screen saver and locke

FilterAmusements                    Required Disk Space: 0 B

xscreensaver - A screen saver and locker for the X Window System
Version: 4.23-20.11 Installed: 4.23-20.11 Size: 17.7 M Media No.: 1
License: BSD 3-Clause; GPL v2 or later; Public Domain, Freeware;
X11/MIT
Package Group: Amusements/Toys/Screensavers
Provides: /etc/xscreensaver/blitspin.xml,

[Helpv]   [Search]   [Disk Usage]                [Cancel]     [Accept]
```

Figure 17-5 Go to the search panel

Use the Tab key to highlight the search option and press Enter. Figure 17-6 is now displayed.

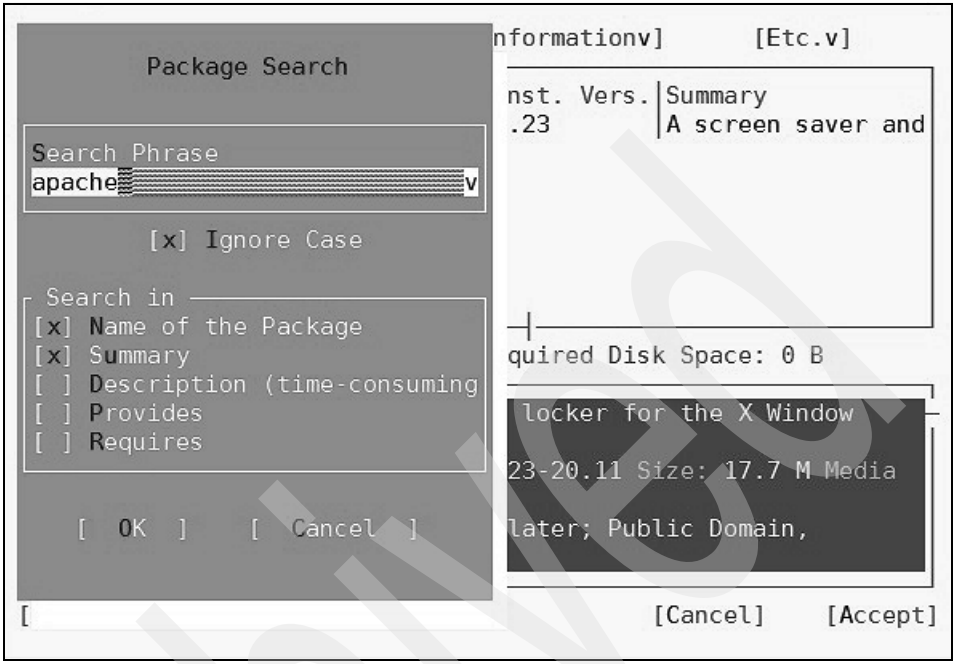

Figure 17-6 Search for software packages by name

The **Search Phrase** box should be highlighted. If it is not, use the Tab key to select it, and then type `apache` in the Search Phrase box. Keep `Ignore Case`, `Name`, and `Summary` checked. The other boxes can be selected at your discretion for more information. Tab down to **OK** and press Enter to reach the window shown in Figure 17-7 on page 213.

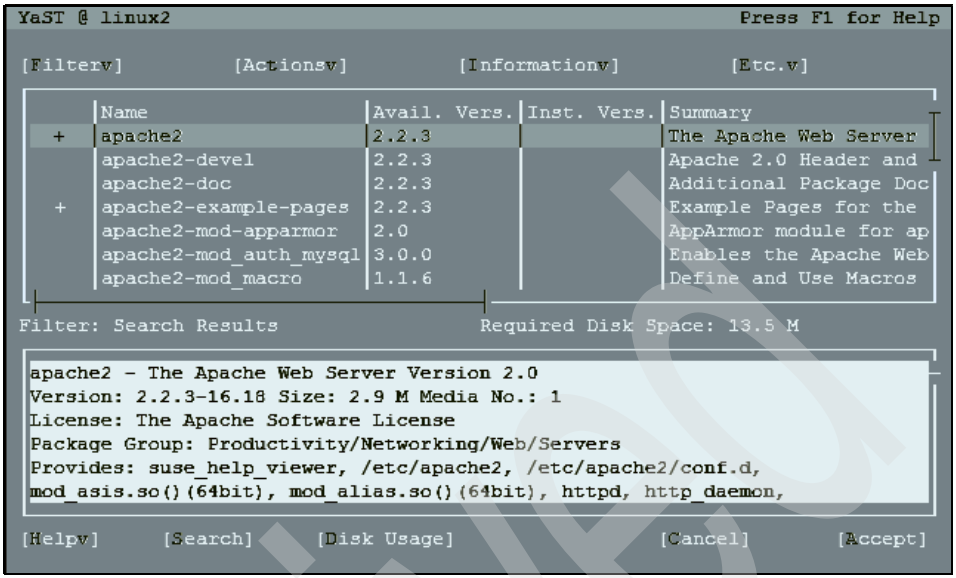

Figure 17-7 Select software packages to install

A list of all Apache packages is displayed. Highlight the packages you want and press Enter to select them. A plus (+) sign will appear next to the selected packages. If there is a dependency for that package, a plus sign will automatically appear next to it as well.

You must select packages **apache2** and **apache_mod_php5** for the required packages we will install. Use the arrow keys to search for other packages in the list. Use the Spacebar to select packages that you want to install. Finally, highlight **Accept** and press Enter to reach the window shown in Figure 17-8 on page 214.

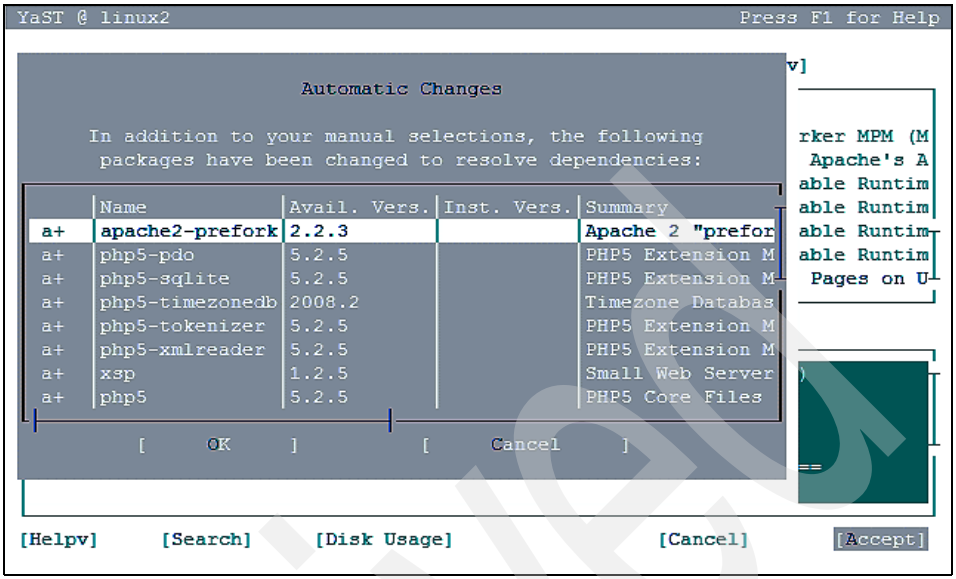

Figure 17-8 Additional requisite software packages selected

This window lists all the additional dependencies outside of apache2 that will be installed. Highlight **OK** and press Enter. The installation of the packages will now begin and the progress of the installation will be displayed. After all packages are installed, Figure 17-9 on page 215 will be shown.

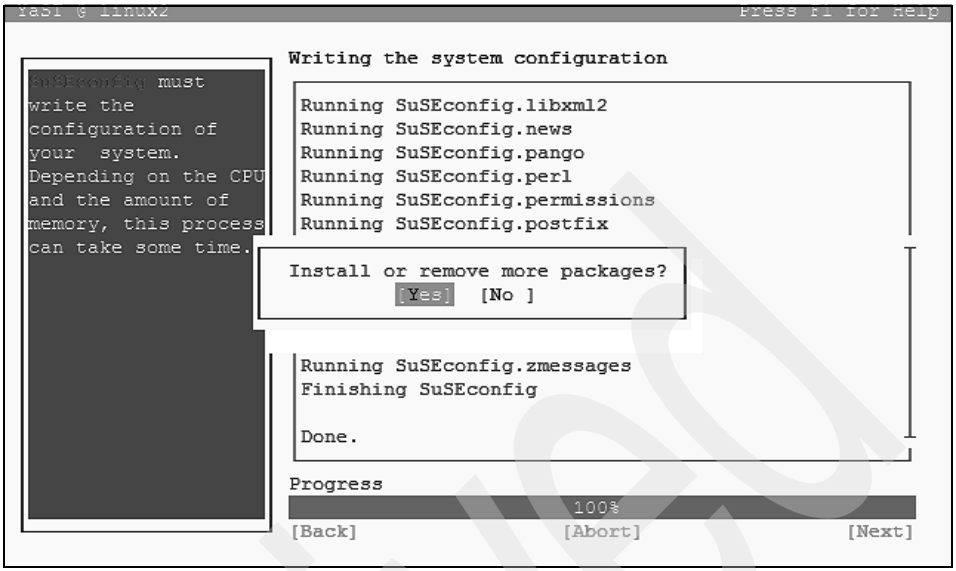

Figure 17-9 Continue to install software packages

In our case, because we want to install PHP5 and Mysql next, we kept `Yes` highlighted and pressed Enter.

Next, highlight **Search** and press Enter; you will be returned to the panel to search for more packages.

In the search box type `mysql`, tab to **OK** and press Enter. At the minimum, select with the Spacebar packages **mysql**, **mysql-client**, and **php5-mysql**. Select **Accept** and press Enter. You should receive some additional automatic changes (you may not get the same results if some of this software is already installed at your site).) Tab to **OK** and press Enter.

Again, the software you selected will be installed at this point. You will then be asked if you want to install more software. Keep `Yes` highlighted and press Enter.

Now tab to **Search** and press Enter. In the search box, type `php`, tab to **OK** and press Enter. Notice that you now have the letter "`i`" displayed in the left column for many of the packages. This means that the package is already installed.

In addition to the packages already installed, in our case we selected **php5-devel** and **php5-pear**. Tab to **Accept** and press Enter, then tab to **OK** and press Enter. The software will be installed and again you have the choice of installing additional software. At this point we selected **No**. You have now installed Apache, Mysql and PHP on top of Linux (also known as LAMP).

17.3 Starting and testing LAMP components

Before installing MediaWiki software, we configure and test Apache PHP and MySQL as described here.

17.3.1 Testing Apache

You can display the version of Apache2 by using the **apache2ctl -v** command.

Then start Apache2 as shown in the Example 17-1.

Example 17-1 Start Apache2

```
****************************************************
*      .--.     Welcome to the Linux s/390x VM     *
*     |o_o |    SUSE Linux Enterprise Server 10.2  *
*     |:_/ |    System Admin: John Doe             *
*    //   \ \                jdoe@company.com      *
*   (|     | )  This system governed by corprate   *
*  /'\_   _/`\  Policy K49-r v21 please read       *
*  \___)=(___/  before accessing system            *
****************************************************
[root@linux2|~] service apache2 start
Starting httpd2 (prefork)                                done
[root@linux2|~]
```

Next, test that your Linux on System z Apache Web server is communicating with a Web browser as shown in Figure 17-10 on page 217. Type in your Linux on System z IP address or host name as a URL:

http://9.12.2.91

or

http://your.server.name/

Port 80 is the default and does not have to be specified.

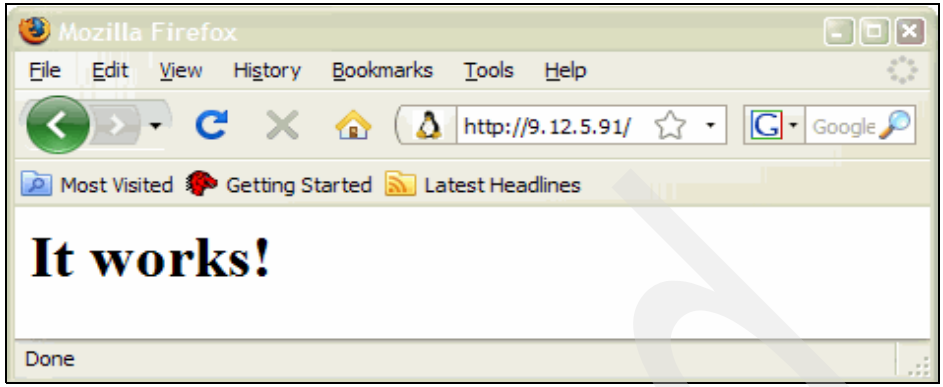

Figure 17-10 Successful test of Apache2 install

17.3.2 Verifying PHP is working

First we need to know where the root directory of the Apache server is. Under SUSE, we can pull it out of the configuration file for Apache as shown in Example 17-2.

Example 17-2 Finding the DocumentRoot on SUSE

```
[root@linux2|~] grep 'DocumentRoot "' /etc/apache2/default-server.conf
DocumentRoot "/srv/www/htdocs"
```

Under Red Hat, we can find the same information, as shown in Example 17-3.

Example 17-3 Finding the DocumentRoot on Red Hat

```
[root@linux6|~] grep 'DocumentRoot "' /etc/httpd/conf/httpd.conf
DocumentRoot "/var/www/html"
```

Now we can create a one-line PHP script that will allow us to see all the PHP information about the install. In `/srv/www/htdocs` in our example system, we created the program `phpinfo.php`, as shown in Example 17-4.

Example 17-4 The phpinfo.php program

```
<?php phpinfo(); ?>
```

Now we can run the php program by using a Web browser and view the `phpinfo.php` configuration, as shown in Figure 17-11 on page 218.

PHP Version 5.2.5	
System	Linux linux2 2.6.16.60-0.21-default #1 SMP Tue May 6 12:41:02 UTC 2008 s390x
Build Date	Apr 24 2008 07:01:09
Configure Command	'../configure' '--prefix=/usr' '--datadir=/usr/share/php5' '--mandir=/usr/share/man' '--bindir=/usr/bin' '--with-libdir=lib64' '--includedir=/usr/include' '--sysconfdir=/etc /php5/apache2' '--with-config-file-path=/etc/php5/apache2' '--with-config-file-scan-dir=/etc /php5/conf.d' '--enable-libxml' '--enable-session' '--with-mm' '--with-pcre-regex' '--enable-xml' '--enable-simplexml' '--enable-spl' '--enable-filter' '--disable-debug' '--enable-inline-optimization' '--disable-rpath' '--disable-static' '--enable-shared' '--program-suffix=5' '--with-pic' '--with-apxs2=/usr/sbin/apxs2' '--disable-all' '--disable-cli'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled

Figure 17-11 Looking at the phpinfo.php file to view the php configuration

17.3.3 Configuring MySQL

Before configuring MySQL, we first display and verify the version of MySQL with the following command from a PuTTY session: **mysql -V**. There are two additional configuration steps to complete for MySQL before we can install MediaWiki.

1. Copy a sample configuration file to the /etc directory. Then apply the appropriate ownership and access. Read through the configuration file to understand its contents. If you are going to use tables from the transactional storage engine InnoDB, be sure to uncomment that section in the my.cnf file, as shown in Example 17-5.

   Example 17-5 Configure MySQL configuration file

   ```
   cp /usr/share/mysql/my-medium.cnf /etc/my.cnf
   chown root:root /etc/my.cnf
   chmod 644 /etc/my.cnf
   ```

2. You must set the password for the database superuser. Remember this password because it is required for the installation of MediaWiki. Use the command shown in Example 17-6 to set the password.

   Example 17-6 Set password for MySQL

   ```
   mysqladmin -u root password ‘yourpassword’
   ```

17.4 Installing MediaWiki software

Follow these steps to install MediaWiki on your Linux server running on System z:

1. Download the MediaWiki software as a TAR file from the following site:

   http://www.mediawiki.org/wiki/Download

2. FTP the compressed TAR file to your Linux server.You can temporarily put it in a `/tmp` directory or in your `/home` directory, and move it later to its permanent location.

3. Extract the TAR file (your version number of MediaWiki may be different):

   ```
   tar xzf mediawiki-1.14.0.tar.gz
   ```

4. Move the extracted folder to the standard directory where Apache can access it. Use the **mv** command to simplify the name of the directory. Also grant access to the `config` folder for the installation script.

   ```
   mv mediawiki-1.14.0 /srv/www/htdocs/
   mv mediawiki-1.14.0/ mediawiki
   chmod a+w /srv/www/htdocs/mediawiki/config
   ```

5. From a Web browser, point to the following URL (`your.linux.hostname` can be an IP address):

   http://your.linux.hostname/mediawiki/config

`MediaWiki config` will check your environment to make sure that all components are available. You will get a report of the results of this check. You should receive a Web page showing the following message (usually displayed in green):

```
Environment checked.
```

Under this message boxes and radio buttons to configure MediaWiki are displayed. Most of the options are your local names. At the point where information about the superuser account is requested, check the box, keep the user ID as root, and supply the MySQL password you set in MySQL configuration step 2 in 17.3.3, “Configuring MySQL” on page 218.

After you supply all the requested information, click the **Install MediaWiki!** button at the bottom of the page. Fix any errors that are returned and click **Install MediaWiki!** again. A successful install will display your environment settings and instruct you to move the configuration file that was created for you. The message states that the install process created a configuration file called `LocalSettings.php`. This file must be moved to the MediaWiki main directory:

```
mv /srv/www/htdocs/mediawiki/config/LocalSettings.php
/srv/www/htdocs/mediawiki
```

Now you can access the new MediaWiki site by entering the following URL in a Web browser:

http://your.server.name/mediawiki

We modified our wiki to include the IBM Redbooks logo. See the MediaWiki documentation under $wgLogo for instructions about how to include your own logo on your wiki. The home page of our wiki then looked as shown in Figure 17-12.

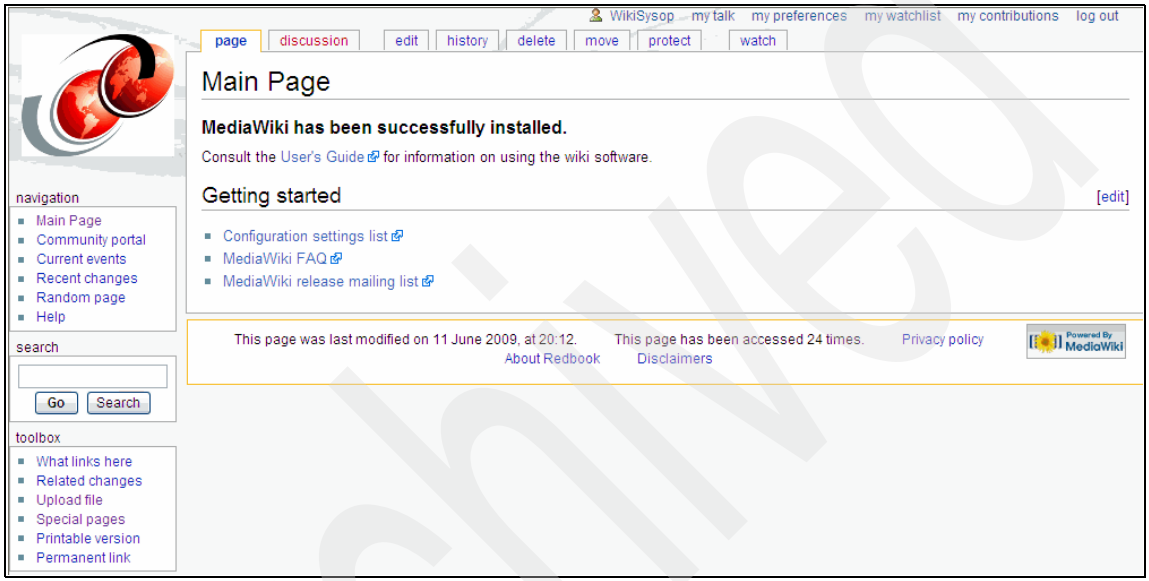

Figure 17-12 Home page with addition of Redbooks logo

17.5 Exporting MediaWiki Pages pilot

It is ideal if you have a small part of the application that can serve as a pilot project for the migration, because you can use the pilot project to learn a significant amount about the readiness of both the environment and your organization for the main migration project.

Our objective for the pilot was to export the contents of a MediaWiki database in XML format from the source system, and then import the data into the target system. In our MediaWiki migration example, we have an instance of MediaWiki running on Linux on Intel. The target system is the system we just installed on Linux on System z. The utilities we used to dump the database are supplied by the application vendor.

We attempted a full XML formatted dump of the contents of the database to a large USB drive by using the following command.

```
php dumpBackup.php --full > /media/Corsair/mw/dump.xml
```

This command was unsuccessful. We received a terse error message:

```
DB connection error: Access denied for user .....
```

It was clear that the source system had security requirements for the migration that were not discovered during the planning stage. However, because this problem was discovered during a pilot project, the impact to the overall migration project would have been limited.

We searched the vendor's documentation Web site to research the cause of the issuer. We found the following reference, which described how the backup utility works and how it determines the username and password to use for the dump:

`www.mediawiki.org/wiki/Manual:DumpBackkup.php`

After the appropriate configuration file (`AdminSettings.php`) was updated with the correct username and password, the database was dumped successfully in XML format to the USB drive.

17.6 Importing MediaWiki Pages pilot

The next step of the migration is to import the XML dump of the database to the target platform MediaWiki server. Again, this is the Linux on z server we installed previously. We first used common FTP to transmit the dumped database to Linux on z. It was uploaded to the `/tmp` directory. (You can also use SFTP; refer to Appendix B for a description of SFTP.)

The final migration step of the pilot project is to utilize the import utility and load the data to the Linux on z server. We used the following command for this task:

```
php importDump.php < /tmp/louis/dump.xml
```

This import utility completed successfully, so we were ready to test the success of the migrated data. Because the source database had names and pictures of the contributors to the wiki, we decided to make the first test case to search and display one of the contributors. The results of that test are described here.

We expected to see a picture of the wiki administrator, but no picture was displayed. The area where the picture was supposed to appear displayed the name of a file instead. Figure 17-13 on page 222 shows the relevant portion of the Web page displaying the file name rather than the expected picture.

Figure 17-13 Failure to display picture of WikiSysop

We questioned the wiki administrator to determine why the picture did not display. He informed us that *.jpg files were configured to be in a separate images directory on the system and not in the database itself. If you simply migrate the main database, you will not obtain the images.

This represented another failure in analysis and planning: not all the supporting files for the application were discovered and documented in early stages of the migration project. Again, because this was a simple project and we were only in the pilot stage of the migration, the implementation of a resolution would not be too time-consuming. However, if this were a complex project with tight schedules, a mistake of this magnitude could jeopardize the success of the entire project. The importance of analysis and thorough planning cannot be overestimated.

In our case, we returned to the source server and offloaded the `/images` file to a USB device. We transferred the `/images` file to the `/images` directory on the Linux on System z target server. Executing the search for the wiki administrator resulted in the images being displayed as expected.

17.7 Migrating the Wiktionary Web site

At this point, we had successfully installed the MediaWiki application on Linux on System z. We also migrated the contents of a small database to Linux on System z. This demonstrated that many organizations could easily benefit from using MediaWiki as a knowledge management system running on System z.

Knowledge in the form of documentation is often spread across many systems and data sources. Search capabilities on this far-flung documentation are limited to non-existent. A centralized wiki for knowledge-sharing with quick turnaround, ease of administration, low cost, and reliability of software and hardware on System z is an ideal solution.

The small database that we converted did not have all of the functions and facilities of the MediaWiki software suite. We wanted to execute a more complex and comprehensive test of MediaWiki running on Linux on System z. Initially, we thought about migrating all of Wikipedia to Linux on System z. Without a doubt, Wikipedia is the most popular application running on MediaWiki. However, we discovered that the disk space requirement was more than our test system was designed to handle.

Another wiki that is similar to Wikipedia is Wiktionary. It is a dictionary in wiki format. The database for Wiktionary was available and suitable for our migration example.

17.8 Acquiring an English Wiktionary XML dump

WikiMedia has a project that coordinates public dumps of wiki content. The following Web sites for database dumps discuss the project, its use, limitations, tools, and so on.

How to obtain the data:

http://meta.wikimedia.org/wiki/Data_dumps

Where to obtain the data:

http://download.wikimedia.org

WikiMedia servers:

http://meta.wikimedia.org/wiki/Wikimedia_servers

We followed the documentation to find the latest dump of the English Wiktionary. It was a simple process to download the compressed dump, which was in .bz2 format. We placed the file in the `/home` directory on our Linux server. It could be put in any directory that has sufficient space.

17.9 Importing a Wiktionary XML dump

We then decompressed the file and started the import of the XML file. You can monitor the process of the import by using the **tail** command. These three steps are illustrated in Example 17-7.

Example 17-7 Import XML dump of database

```
bunzip2 -dv enwiktionary.xml.bz2

php importDump.php < /home/louis/enwiktionary.xml >& progesss.log &
(Add nohup to the beginning of the command to keep the process going
even if your putty session is cancelled.)

tail -f progress.log
```

The import procedure was performed through an interpretive scripting process and no special tuning was needed for the database software or the disk subsystem. We made no prior estimate of how long the loading process would take but it was longer than we expected. These would be important analysis and planning points for a well-planned migration project. After the import of the dump was complete, it was time to test the results.

17.10 Testing and remediating the Wiktionary test

Our first test of the IBM Redbooks wiki with the Wiktionary database uploaded produced the result displayed in Figure 17-14 on page 225. (Only one section of the window is shown here so that the problem is clearly illustrated.) As can be seen, there is something wrong with the display. The same type of garbled data was scattered throughout this page and throughout other dictionary pages.

Figure 17-14 Import error

We knew that advanced users of MediaWiki software use functions that are installed through a facility called extensions. With a little research, we discovered that we needed to install the extension called Parser Functions. We were confident that the installation of a MediaWiki extension would be smooth and successful because open source software is highly portable and because Linux on System z is a standard distribution (SLES10, in this case).

We downloaded the extension from the MediaWiki Web site and installed it according to their directions. We did not have to recycle any processes. We only searched for another dictionary entry and found that our problem was solved. We did not test every page in the dictionary but we knew the definitions for apple, pear, and other fruits were displayed correct. We also used the IBM Redbooks Wiktionary while writing the documentation for this IBM Redbooks publication.

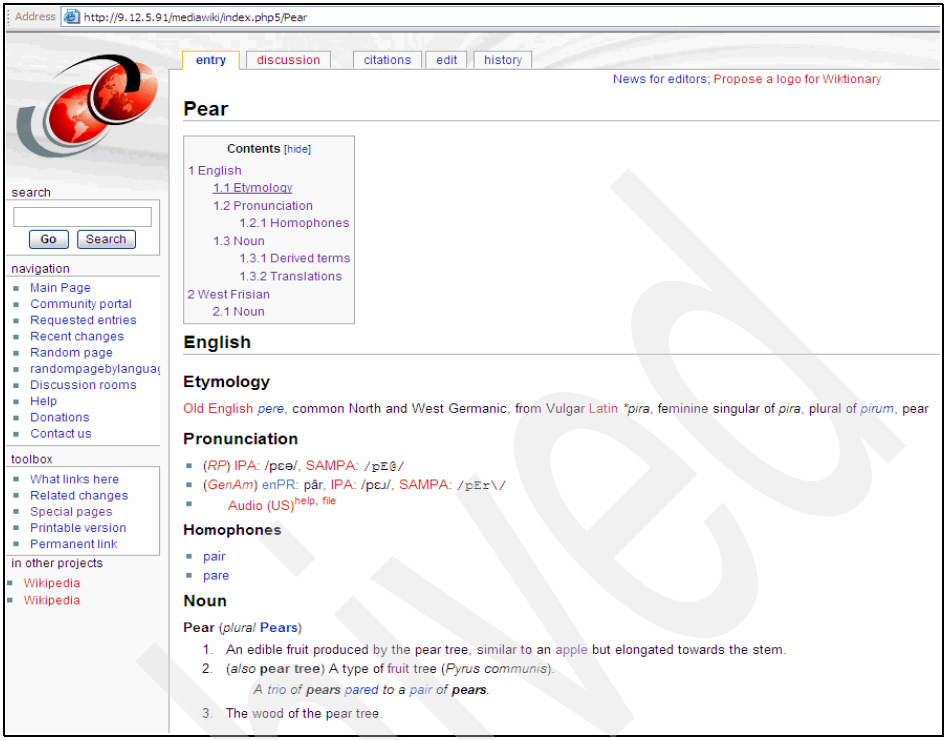

Figure 17-15 Successful display of Wiktionary entry

17.11 MediaWiki migration - conclusions

In our experience of migrating an open source application, we concluded that there were no errors or problems encountered with any of the software components themselves. The Linux-Apache-MySQL-PHP (LAMP) software stack was installed easily on Linux on System z. MediaWiki software was also a straightforward installation. From this, we conclude that other open source software would work well on Linux on System z. We also postulate that open source solutions on Linux on System z should be considered as viable business solutions. All the problems we encountered could have been discovered in a comprehensive analysis and planning stage of the project. The errors of missing permissions, missing image files, and requirements of additional add-on software could have been avoided. This small example has also shown the extreme usefulness of a pilot project that uses a small portion of the entire application to be migrated. If that is not possible in your migration project, then any pilot project that uses the same middleware, tools, databases, or other functions in your environment would be very valuable.

18

Mono Extensions and Microsoft .NET migration

This chapter discusses early experiences in installing and using Mono software. Because of the wide use of .NET, there are millions of developers who have experience building applications in C#. There are benefits to choosing Mono for application development, as explained here.

Note that this chapter simply presents an overview of this subject, because the authors did not personally have in-depth experience with Mono or with the .NET Framework. However, this subject is presented here because it is relevant to the topic of migrating to Linux on System z and to current developments within the industry.

18.1 Mono Open Source Project

Mono is often described as a software platform that is designed to allow developers to create cross-platform applications. It is an open source implementation of Microsoft's .NET Framework. There are many components included with Mono, as listed here:

- C# Compiler

  As mentioned, many developers have experience in building applications in C#, and support materials including books, Web sites, education, and sample source code are available.

- Mono Runtime

  Runtimes provide functionality to high level programming languages; they allow programmers to concentrate on writing an application instead of on writing redundant and complex system infrastructure code.

- Base Class Library

  Using class libraries increases productivity, because programmers do not have to code their own solutions for many programming activities. These class libraries are compatible with Microsoft's .NET Framework.

- Mono Class Library

  Mono supplies classes that supplement the base class library. These classes provide additional functionality.

- Mono Migration Analyzer

  The Mono Migration Analyzer tool identifies issues encountered when you are porting your .NET application to Mono. This tool helps to identify specific calls (P/Invoke) and areas that are not supported by Mono. The results provide a guide to getting started on porting .NET applications.

The most current information about the Mono Open Source Project is available at the following site:

http://www.mono-project.com

18.2 Installing Mono Extension from Novell

Novell provides a for-fee extension to the common Mono code base. Its official name is SUSE Linux Enterprise Mono Extension. The supported platforms are x86, x86_64, and IBM System z 64 bit(s390x).

Note: The Linux kernel architecture designation is “s390x” for 64-bit kernels. Therefore, the Red Hat Package Manager (RPM) that you would use is the s390x RPM.

The s390 name derives from the first hardware processor (System 390) to run Linux.

In the following sections we explain our use of the Novell Mono Extension, but do not provide a detailed explanation of the product. The following Web site provides more detailed and current information:

http://www.novell.com/products/mono

18.2.1 Downloading the Mono Extension

The first step in this example is to download an evaluation copy of Mono Extension from the Novell Web site. You must register to obtain a valid logon to the download site.

After accessing the download site, we downloaded an ISO image file of the product. In our case, the name of the downloaded file was SLES-11-DVD-s390x-GM-DVD1.iso.

The next step is to upload this file to the Linux on System z server. We used common FTP to transfer this .iso file to our Linux on System z server.

After the upload was complete, we mounted the .iso file to a loopback device so that YaST had access to it. The mount command we used is shown in Example 18-1.

Example 18-1 Mount the iso file for YaST access

```
mount -o loopback -t iso9660 /home/louis/SLES11-MonoDVD.iso /mnt/iso
(We shortened the iso filename when moving it to /home directory.)
```

Note: After completing the install, we investigated the improvements to YaST in SLES11. You can now directly use an iso image without mounting it as a loopback device.

Next, we invoked the YaST install tool. We selected **Software** and then tabbed over to **Add-On Products**. The screens led us through the install of the Mono Extension product. There was a prerequisite of a few apache2-mod_mono packages. For this task, we simply switched over to the Software Management

function of YaST and performed a search for all Apache packages. We installed all the apache2-mono packages. Then we returned to the Add-On Products function and the install completed successfully.

The installation process worked well. It installed the product in an `/opt/novell/mono` directory; these directories are created by the installation process. Note that the user ID of the person who performs the installation will require update authority to the `/opt` directory.

18.2.2 Performing a HelloWorld test

We performed a HelloWorld test by coding a simple C# program from an example on the Internet, and then compiling and executing it. The commands and results are shown in Example 18-2.

Example 18-2 HelloWorld in C#

```
root@linux3|/home/louis mcs HelloWorld.cs
root@linux3|/home/louis mono HelloWorld.exe
Hello World form Louis Henderson
```

18.2.3 Running XSP ASP.NET examples

The Mono product provides examples for your use. To use these examples, we first moved to the directory where the test samples resided:

```
cd/opt/novell/mono/lib/xsp/test
```

Then we started the XSP server from this directory using the default settings (except for the address setting where the server resides; in our case, the address was 9.12.5.92):

```
/opt/novell/mono/bin/xsp -- address 9.12.5.92
```

At this point, the server was now running. Example 18-3 shows starting the server, along with the messages received from the server when it is started.

Example 18-3 XSP server is running

```
[root@linux3|/opt/novell/mono/lib/xsp/test] /opt/novell/mono/bin/xsp --address
9.12.5.92
xsp2
Listening on address: 9.12.5.92
Root directory: /opt/novell/mono/lib/xsp/test
Listening on port: 8080 (non-secure)
Hit Return to stop the server.
```

Next, we brought up a Web browser and pointed to the server to access the test samples.The URL in our case was http://9.12.5.92:8080/. The test panel tree was displayed, as shown in Figure 18-1.

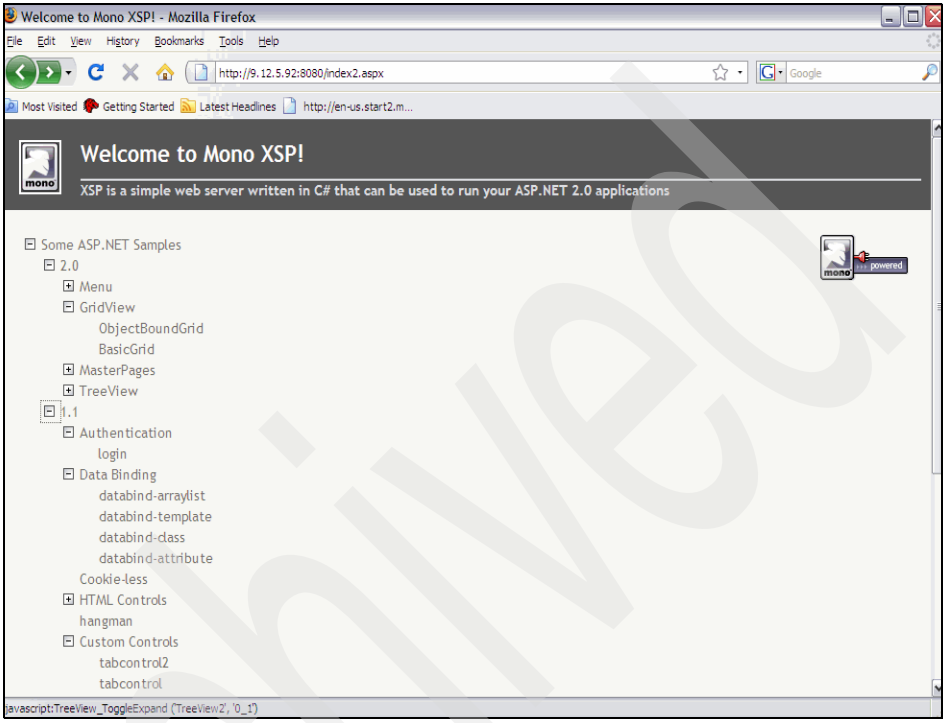

Figure 18-1 XSP Web server showing ASP.NET examples

The samples were separated into ASP.NET 1.1 and ASP.NET 2.0. In the ASP.NET 1.1 section, there were nearly 100 different test samples to execute. Only the hangman example was unsuccessful. Going back to the installed samples directory on the server, we could not find any reference to the hangman sample.

There were fewer test cases under the AP.NET 2.0 tree. We only found one anomaly in the Gridview test. Overall, we found the tests to be simple but successful.

18.3 Feasibility study: ASP.NET on Apache

In this section, we examine the example application and its environment. We also discuss installing the Apache server and implementing Mono modules on the Apache server.

18.3.1 Example application and environment

At this point, our example MS.NET application was running on Linux using XSP as described in 18.2.3, “Running XSP ASP.NET examples” on page 230. For our satisfaction, we wanted to test part of an already-running application. We selected the attendance register module for this case study because it had both basic and advanced MS.NET controls such as Gridview, Calender, a radio button list and drop-down controls.

Note: This is not a full-fledged case study about how to migrate an MS.Net Application on to Linux on System z with Mono extensions. This section simply describes a feasibility study we performed on running the MS.Net application or a module on Linux on System z.

Current environment

The application was currently set up on an MS Windows 2000 Server with 2 GB of memory. The application was developed using MS.Net technologies like C# and ASP.Net. The MS Windows server was running Internet Information Server (IIS) for hosting the Attendance application.

Target environment

The target environment was installed with the latest GA version SLES 11 with all the necessary development tools and libraries. Example 18-4 illustrates how to find this version of Linux.

Example 18-4 Linux on System z version

```
[root@linux5|~] cat /etc/issue
Welcome to SUSE Linux Enterprise Server 11 (s390x) - Kernel \r (\l).
```

Example 18-5 illustrates how to find the kernel version.

Example 18-5 Kernel version

```
[root@linux5|~] uname -r
2.6.27.19-5-default
```

18.3.2 Installing Apache

We decided to install the latest Apache release (in our example, release 2.2.11) from the Apache Web site. Example 18-6 displays the steps used to install the Apache-HTTP server onto our Linux on System z server.

Example 18-6 Apache HTTP server installation steps

```
[root@linux5|~] tar -zxvf httpd-2.2.11.tar.gz
:
:
[root@linux5|~/httpd-2.2.11] ./configure --with-expat=builtin
--prefix=/usr/local/apache --enable-module=so
:
:
[root@linux5|~/httpd-2.2.11] make

[root@linux5|~/httpd-2.2.11] make install
```

After the installation is complete, the Apache version needs to be verified. In our case, we installed the Apache server in the directory `/usr/local/apache`, which is passed as a parameter to the **`./configure`** command as shown in Example 18-7.

Example 18-7 Apache version

```
[root@linux5|/usr/local/apache/bin] ./apachectl -v
Server version: Apache/2.2.4 (Unix)
Server built:   Jun 19 2009 11:12:58
```

Example 18-8 shows how to start the Apache-HTTP server to verify that everything is working properly.

Example 18-8 Starting the Apache server

```
[root@linux5|/usr/local/apache/bin] ./apachectl start
[root@linux5|/usr/local/apache/bin] tail -f ../logs/error_log
[Wed Jun 24 15:55:47 2009] [notice] Apache/2.2.11 (Unix) configured --
resuming normal operations
```

We verified that the HTTP server was working by accessing the Web page and entering the IP address of the server in the location bar of a Firefox Web browser. Check the error logs for possible errors, and resolve any errors you find before proceeding.

18.3.3 Implementing Mono modules with Apache

Mod_Mono is an Apache 2.0/2.2 module that provides ASP.NET support for the Apache HTTP server. The module sends the requests for ASP.NET pages to an external program called mod-mono-server. This program actually handles the requests. The communication between the Apache module and mod-mono-server is established using sockets.

After configuring the Apache-HTTP server, we needed to update the Apache configuration files to make it send the requests that we want through the mod_mono module.

There are two ways by which you can configure your Apache server to run your MS.Net applications:

- Include and use the default Mono configuration file. This configuration file needs to be included in the `httpd.conf` file. (This is also known as Autohosting by the Mono community.)
- Or, manually update the `httpd.conf` file to include the Mono modules and to make Apache recognize the MS.NET file extensions like .aspx and so on.

Autohosting

Autohosting basically means that you need to load the `mod_mono.conf` file in the Apache configuration file. During the installation of the mono extensions, Mono places the configuration, by default, in the `/etc/apache2` directory. The mod_mono.conf file would be placed under the `/etc/apache2/conf.d` directory as shown in Example 18-9.

Example 18-9 Configuring httpd.conf file by using Autohosting

```
[root@linux5|/usr/local/apache/conf] vi httpd.conf
:
:
Include /etc/apache2/conf.d/mod_mono.conf
ServerRoot "/usr/local/apache"
```

Restarting with the HTTP server with these configuration changes enables it to start the server using any MS.NET projects placed under the directory configured in `http.conf`. (Usually, the default directory location for the applications or projects would be in the `htdocs` directory.)

Manual configuration

As previously mentioned, the Apache HTTP server's configuration file needs to be manually updated with the Mono modules as well as the changes that would make the HTTP server recognize the MS.NET file extensions.

We edited the `http.conf` file by including the mod_mono.so modules (see Example 18-10).

Example 18-10 Manual configuration of httpd.conf

```
[root@linux5|/usr/local/apache/conf] vi httpd.conf
LoadModule mono_module /usr/lib64/apache2/mod_mono.so
```

Next, the `httpd.conf` file must be updated so that the HTTP server can recognize the MS.NET file extensions. Example 18-11 displays the file extensions that we included.

Example 18-11 Updating httpd.conf to recognize the .NET file extensions

```
[root@linux5|/usr/local/apache/conf] vi httpd.conf
AddType application/x-asp-net .aspx
AddType application/x-asp-net .cs
AddType application/x-asp-net .config
AddType application/x-asp-net .dll
DirectoryIndex index.aspx
DirectoryIndex Default.aspx
DirectoryIndex default.aspx
```

The Apache configuration file also must be updated so that it can determine out which applications need to use the Mono modules. Example 18-12 shows providing the Apache HTTP server with the applications that need to be handled by the Mono modules.

Example 18-12 Mono modules configuration in httpd.conf

```
Alias /test10 "/opt/atendmod"
AddMonoApplications  default "/test10:/opt/atendmod"

<Location /test10>
    SetHandler mono
</Location>
```

Example 18-12 shows that we created an alias name for accessing the MS.Net application, as well as the **`AddMonoApplications`** configuration tag which informs the HTTP server that it is a Mono application. It also shows that we are setting up the handler for the application as Mono by using the **`SetHandler`** tag.

After the `httpd.conf` file is updated with all the configuration details as described, start the Apache HTTP server with the Mono modules initialized. We also recommend that you issue the command **`tail -f`** on the Apache error logs.

If there are any Mono modules-based issues in the logs, the Mono Web site offers useful explanations of how to resolve common problems. Example 18-13 shows starting the server with Mono modules.

Example 18-13 Restarting Apache with Mono modules

```
[root@linux5|/usr/local/apache/bin] ./apachectl start
[root@linux5|/usr/local/apache/bin] tail -f ../logs/error_log
[Wed Jun 24 17:43:16 2009] [notice] Apache/2.2.4 (Unix) mod_mono/2.4 configured --
resuming normal operations
```

In this example, the error log shows that the Apache HTTP server has been started successfully with the required Mono extensions and modules.

18.3.4 Verifying the configuration

After we set up Apache with the Mono extensions, we placed the MS.NET application in their respective directories as specified in the `httpd.conf` configuration file. In our case, we placed the MS.NET application under the `/opt` directory. We copied the applications over, as shown in Example 18-14.

Example 18-14 Copying MS.NET application to the Web access directory

```
[root@linux5|~] cp -r atendmod/ /opt/.
[root@linux5|~] chmod -R 777 atendmod/
[root@linux5|~] cd /opt/atendmod/
[root@linux5|/opt/atendmod] ls -l
total 96
-rwxrwxrwx 1 root root 28521 Jun 25 11:26 Blue hills.jpg
-rwxrwxrwx 1 root root 14624 Jun 25 11:26 Default.aspx
-rwxrwxrwx 1 root root  9062 Jun 25 11:26 Default.aspx.cs
-rwxrwxrwx 1 root root  2887 Jun 25 11:26 empstor.xml
-rwxrwxrwx 1 root root  1585 Jun 25 11:26 Web.Config.orgi
-rwxrwxrwx 1 root root   127 Jun 25 11:26 calendar.gif
-rwxrwxrwx 1 root root  1799 Jun 25 11:26 images.jpg
-rwxrwxrwx 1 root root 17685 Jun 25 11:26 redblogo.gif
-rwxrwxrwx 1 root root  2247 Jun 25 11:26 web.config
```

In our case, we experienced minor issues involving images failing to display. When we verified the Default.aspx Web page, we found the image file directory locations were incorrectly pointing to some other directory, as though still on the MS Windows platform.

Apart from that, the Attendance module worked perfectly, with all the MS.NET controls handled correctly by the Mono extensions. Figure 18-2 shows our successful application launch from the Web browser.

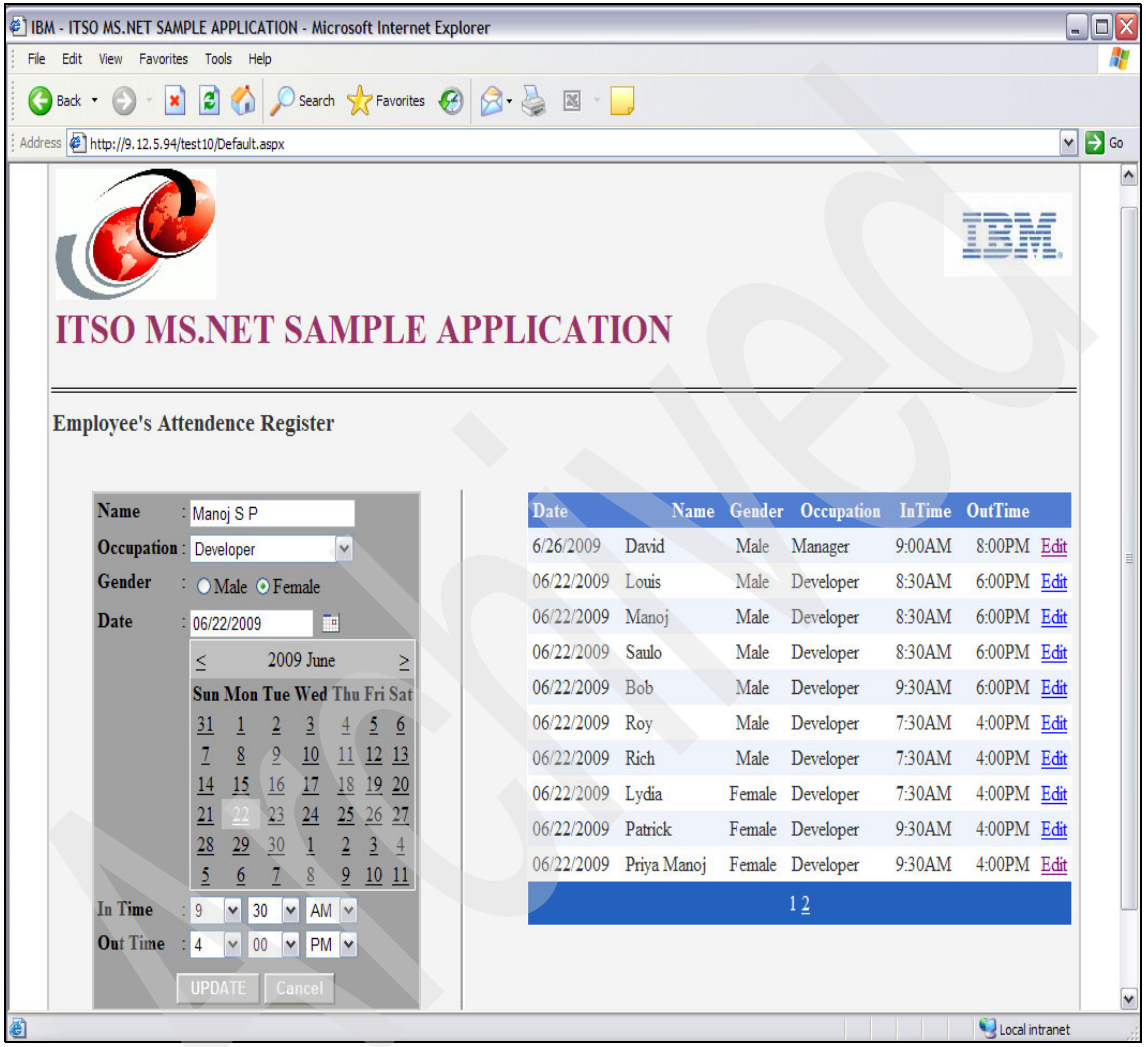

Figure 18-2 MS.NET sample module running on Linux on System z with Mono extensions

Part 4

Appendixes

A

Linux on System z commands

This appendix provides a table of commands that are used by Linux on System z administrators.

For more detailed information about these commands, refer to *Device Drivers, Features, and Commands,* SC33-8411, which can be found at:

http://www.ibm.com/developerworks/linux/linux390/development_development_documentation.html/

List of Linux on System z commands

These commands are unique to Linux on System z because of the specialized storage, network, and cryptographic devices available on System z. The commands prefixed with *vm* are used to communicate with z/VM. All commands have a man entry.

Table A-1 Commands - Linux on System z

Linux on System z	Usage
chccwdev	Set ccw devices online or offline
chchp	Change channel path status
chreipl	Modify reIPL configuration
chshut	Control system behavior
chzcrypt	Modify zcrypt configuration
cpuplugd	Activate CPUs and control memory
dasdview	Print DASD information
dasdfmt	Format a DASD
fdasd	Partition a DASD
lschp	List channel paths
lscss	List subchannels (used to gather and display information from sysfs)
lsdasd	List DASD devices
lsluns	Discover and scan LUNs in Fibre Channel SANs
lsqueth	List qeth-based network devices
lsreipl	List IPL settings
lsshut	System behavior during change of state
lstape	List tape devices
lszcrypt	List information about cryptographic adapters
lszfcp	List information on zfcp adapters, ports and units
qetharp	Query and purge OSA and hipersockets ARP data
qethconf	Configure qeth devices
tape390_crypt	Encryption support for z Series tape devices

Linux on System z	Usage
tape390_display	Display messages on tape devices and load tapes
tunedasd	Adjust DASD performance (configuration required)
vmcp	Send commands to z/VM hypervisor
vmur	Work with z/VM spool file queues

B

Remote access applications

This appendix provides information about and examples of remote access software, tools, methods, and enhancements to z/VM as well as Linux guest servers running under z/VM.

We discuss the following tools, which run on a Linux or MS Windows desktop; these tools are needed to work with Linux on System z:

- 3270 Emulator for CP/CMS and Linux commands
- Secure Shell (SSH) for a remote terminal
- Secure File Transfer Protocol (SFTP) for file transfer
- The VNC remote desktop

Note: This appendix documents Linux and MS Windows tools and provides corresponding examples. Because the tools used are common, other operating systems could be used and clients should be available. The Linux examples may also apply to Apple OS/X but have not been tested.

3270 Emulation

3270 is a protocol that is needed to communicate with some components on a System z computer. Thus, programs that can communicate using this protocol are called 3270 emulators. In the past, programmers would use a large “dumb” terminal that occupied most of a desk. The physical connection was usually done by coaxial cable. Today we use emulators that reproduce the terminal in software.

The 3270 emulator software is based on the Telnet protocol that emulates a 3270 connection. As a Telnet connection, all transmitted data including commands go through in the clear without any encryption. For this reason, it is not good practice to allow Internet connections to the z/VM console.

3270 terminal emulators that run under Linux or MS Windows are available and are described here.

Linux x3270 Emulation

The x3270 program is available for all Linux distributions, and it can be installed from its repository of programs. It can be found at the following sites:

http://x3270.bgp.nu/download.html

http://sourceforge.net/projects/x3270/

Figure B-1 on page 247 shows an x3270 display. The x3270 is a full-featured terminal emulator that is more than adequate for performing all the required administration tasks for z/VM.

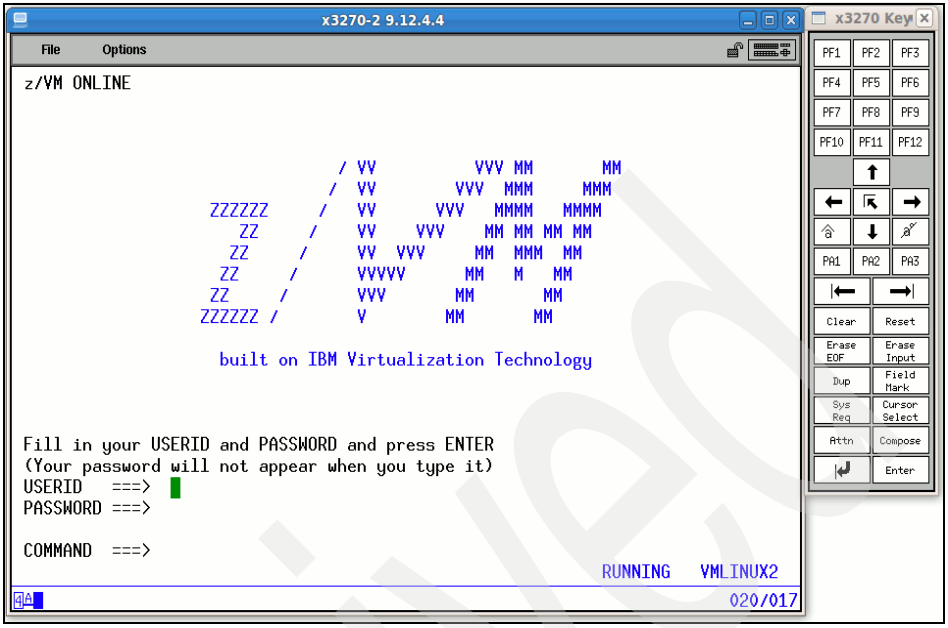

Figure B-1 x3270 running a session to our test VM

MS Windows 3270 emulation

For Microsoft Windows, there are a number of both commercial and Free Open Source Software (FOSS) 3270 emulator programs available. The x3270 program just discussed has been ported to MS Windows. For more information about this program, refer to the following address:

http://x3270.bgp.nu

When working under MS Windows, the team that created this book preferred to use a 3270 emulation package from the IBM Personal Communications Suite, which is shown in Figure B-2 on page 248. For more information about this software, refer to the following Web site:

http://www-01.ibm.com/software/network/pcomm/

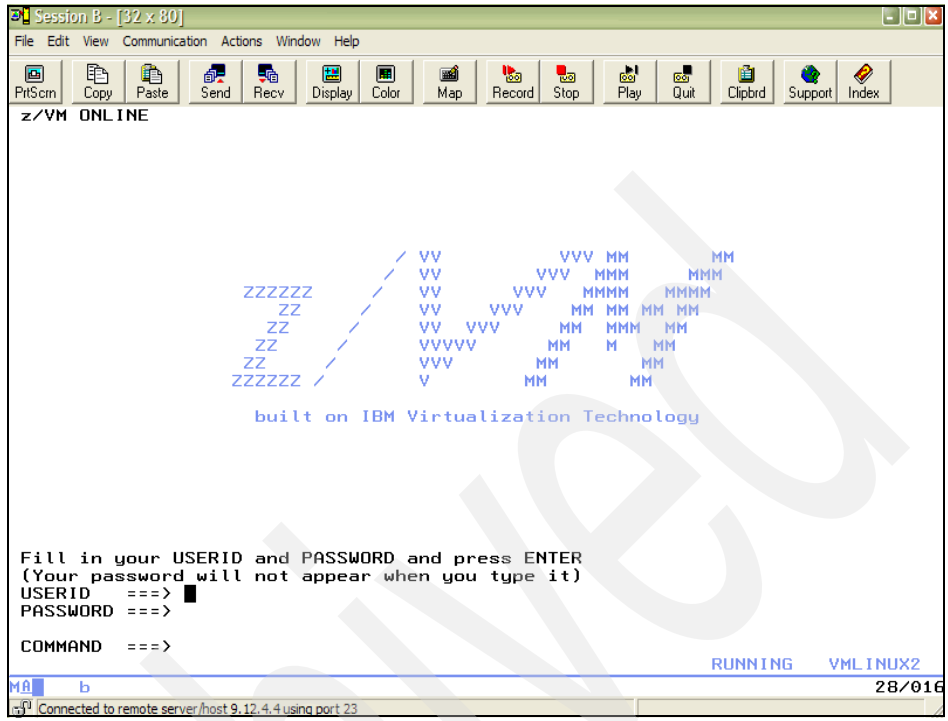

Figure B-2 IBM Personal Communicator showing a 3270 session

Tip: To input ENTER as indicated in Figure B-2, use the Ctrl key on the QWERTY keyboard or the Enter key on the numeric keypad, if available.

Secure Shell

Secure Shell (SSH) is a network protocol that uses a secure channel between two network devices. It has improved confidentiality and integrity over older protocols such as Telnet. In the following sections we explain and illustrate how to use components of the Secure Shell protocol suite.

ssh server

The most common way to access a remote Linux server is by using SSH. Make sure the process sshd is running on the Linux Server. The sshd configuration file is located in the `/etc/ssh` directory as filename `sshd_config`.

Note that the sshd process will default to the use of TCP port 22. Therefore, any firewalls or networking must allow for the remote system to connect to port 22. If another port must be used, it can be configured via the `sshd_config` file.

When changes are made to the `sshd_config` file, the sshd service will need to be informed to pick up the new changes. You can accomplish this by running the following commands. Example B-1 shows the Red Hat command.

Example B-1 Red Hat command

```
# sudo /etc/init.d/sshd reload
Reloading sshd: [  OK  ]
```

Example B-2 shows the SuSE command.

Example B-2 SuSE command

```
# sudo /etc/init.d/sshd reload
Reload service sshd done
```

ssh client

The ssh client is used to make a remote connection to a server and have a command line interface. Telnet was used earlier, but it declined in popularity because it sent all text in the clear, including the password. In contrast, ssh encrypts all information sent over the network, which is why both the client and the server have public and private keys.

Linux ssh client

Most if not all Linux distributions have the ssh client already installed when you obtain them. The most common version is OpenSSH; for more information, refer to the following site:

http://www.openssh.com

The command to use to connect to server `some.server.name` as `user` is shown in Example B-3.

Example B-3 Using ssh for the first time

```
# ssh user@some.server.name
The authenticity of host 'some.server.name (9.12.5.90)' can't be
established. RSA key fingerprint is
70:08:20:41:51:e4:a7:28:b1:44:0b:f8:8d:a3:14:c6.
Are you sure you want to continue connecting (yes/no)? yes
```

```
Warning: Permanently added 'some.server.name' (RSA) to the list of
known hosts.
user@some.server.name's password:
Last login: Wed Jun 10 13:38:26 2009 from 9.12.5.249
```

The server you are connecting to must be resolvable either by DNS or listed in `/etc/hosts`. Note that the first time you connect to a server, its public key will be added to the `~/.ssh/known_hosts` file. Also note that if the server's public key or address do not match after they have been added to the `known_hosts` file, that ssh may not allow the connection. The easiest way to resolve this problem is to remove the old reference in the `known_hosts` file and retry the **ssh** command.

You can also use ssh to tunnel ports from server to client. This is extremely useful when there are firewalls or other security methods between the client and the server. For instance, to test a connection to port 80 (the default port for the HTTP server), use the command shown in Example B-4.

Example B-4 Using ssh to tunnel httpd port 80

```
# ssh -L 80:localhost:80 user@some.server.name
user@some.server.name's password:
Last login: Wed Jun 10 13:53:42 2009 from 9.12.5.249
```

The first 80 is the server side port, and localhost:80 is where it will be mapped to.

Note: Keep in mind that when you are mapping ports, you are doing so on a system-wide level. Therefore, if there is a Web server on the local system, it will no longer be accessible while the ssh session that remapped port 80 is still running. A possible solution to this issue is to map to a local port that is not in use.

Another useful port is for VNC, as shown in Example B-5.

Example B-5 Using ssh to tunnel VNC port 5901

```
ssh -L 5901:localhost:5901user@some.server.name
user@some.server.name's password:
Last login: Wed Jun 10 13:53:42 2009 from 9.12.5.249
```

This command would move the VNC session on:1 to the local machine, which would then allow you to connect to a VNC session on a server in which the 590x port range was blocked via a firewall. Run the VNC client so that it points to the localhost, which is where the port is now mapped, as shown in Example B-6 on page 251.

Example B-6 Running VNC with the localhost

```
vncviewer localhost:1
```

Attention: When vncserver is run, it reports the session number as in some.server.name:1; the port that it is using is 5900+1 (or 5901). Thus, 5901 is the port that needs to be tunneled, but the viewer expects the relative number “1” because you moved the port local. Therefore, all that is needed is localhost:1.

MS Windows ssh client

There are many ssh clients available for Windows, both commercially and as no fee. In our case, we used PuTTY, which can be downloaded from the Internet and easily installed; see:

http://ftp.chiark.greenend.org.uk/~sgtatham/putty/

Describing the basic setup and configuration of all PuTTY functions is beyond the scope of this book.

Tip: You can install the Bitstream Vera Font, which is available at http://www.gnome.org/fonts/ in MS Windows, and then use it as your font for PuTTY. Go to **Window** → **Appearance**, and then use the **Change...** button as shown in Figure B-3. Set the font to Bitstream Vera Sans Mono. This will make it easier to identify zero (0) as opposed to an upper case O, and a lower case L (l) as opposed to the number one (1).

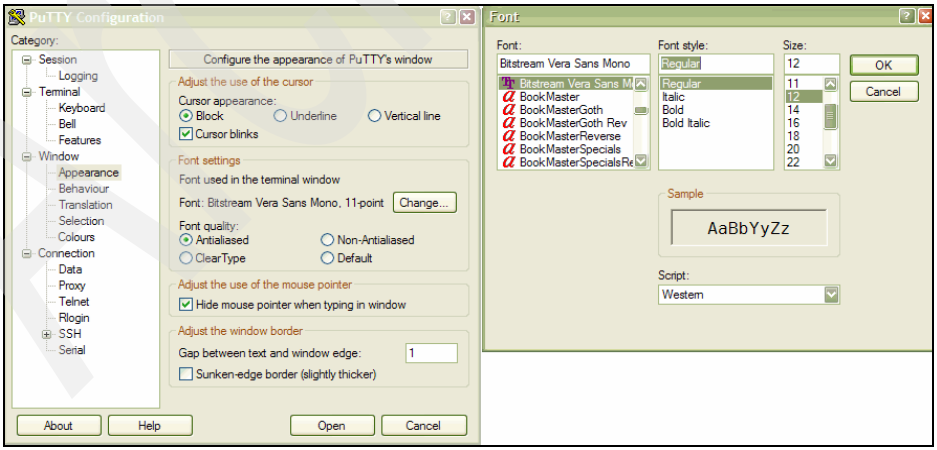

Figure B-3 PuTTY font configuration

Note: You can use PuTTY like OpenSSH to tunnel important ports from server to client. To do so, on the configuration window click **Category** → **Connection** → **SSH** → **Tunnels** and then add the VNC port used (for example, **5901**) to the Source port field. Add **localhost:5901** to the Destination field (click the **Add** button to have it added to the list of forwarded ports).

To permanently save the port forwarding information, return to Session and then click **Save**; see Figure B-4 on page 252.

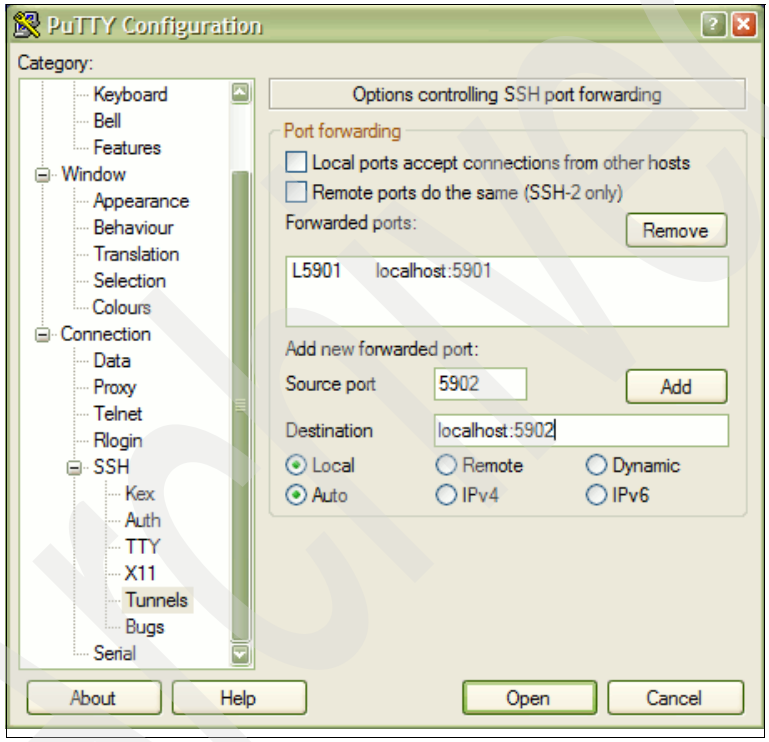

Figure B-4 Setting up PuTTY for port tunneling

Tip: If you have more than one PuTTY session running and mapping the same ports, the first session that was run will own those ports. To avoid this conflict, map different systems to different local ports.

Secure File Transfer Protocol

Secure File Transfer Protocol (SFTP) is part of the SSH protocol, so it does not take an extra service to configure and maintain. Using SFTP is the easiest way to move files to and from your server, assuming that the SSH service is running on the server (ensure that the line shown in Example B-7 is in the `/etc/ssh/sshs_config` file).

Example B-7 Config file /etc/ssh/sshd_config needs sftp line

```
Subsystem       sftp    /usr/libexec/openssh/sftp-server
```

Using SFTP is the preferred method for moving files from a Microsoft Windows system to and from the Linux on System z server.

SFTP under Linux

Although Filezilla is available for Linux, the most common way to move files is by using the command line as shown in Example B-8.

Example B-8 The sftp options on Linux

```
sftp [[user@]host[:file [file]]]
sftp [[user@]host[:dir[/]]]
```

In this way you can move a single file (or many files) to or from the target Linux server. Also, you can simply use a directory to move a full directory.

SFTP under MS Windows

Many Microsoft Windows FTP programs SFTP access, both commercial and FOSS. The client used during the creation of this book was the FOSS client Filezilla, as shown in Figure B-5 on page 254. Filezilla is available at the following site:

http://filezilla-project.org

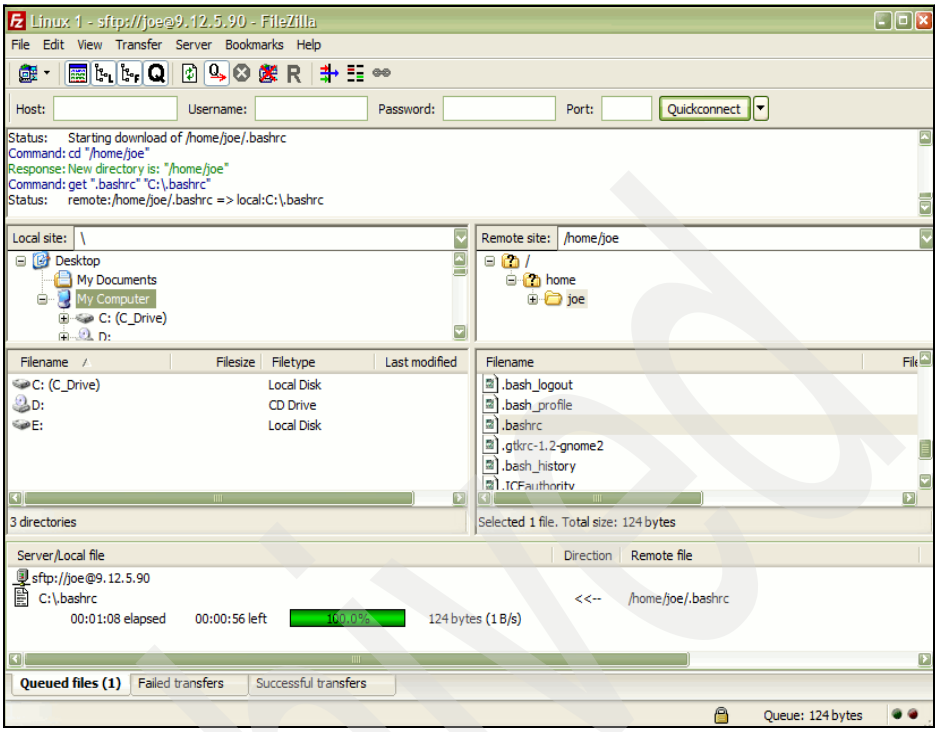

Figure B-5 Filezilla MS Windows example

VNC - remote desktop

VNC is a remote graphical desktop that has two parts, a client and a server. Normally, you would access Linux via an Xwindows desktop. When the system is accessed only via a network, VNC allows an Xwindows session to be sent over the network to VNC client software. VNC client versions are available for all popular operating systems.

> **Note:** VNC is only the mechanism that moves graphics from the server to the client. Xwindows is still run as the GUI environment; a Windows manager is also needed by Xwindows to present the desktop metaphor.
>
> The VNC server is available for MS Windows in that it can be used in the same way to allow an MS Windows XP desktop to be accessed on a remote Linux mobile computer. Again, this is because VNC moves the graphics from server to client.

Setting up the VNC server on System z

There are two ways to configure VNC: a server per system or a server per user. Note the following distinctions:

- If you set up a server per system, you do not need an SSH session to manage the server.
- If you set up a server per user, all the configuration and use can be performed via SSH; therefore, you do not need to alter network security.

Each configuration is explained in more detail in the following sections.

Configuring a VNC server per system

If your Linux VM guest is installed with SUSE (all versions apply), then there is no configuration to perform. Simply point your VNC client to `hostname:1` (where hostname is the address of the Linux VM) and you will see the GDM login window.

> **Important:** When you log into a Linux system via VNC and type in the username and password, they are transmitted as VNC traffic over IP and thus in the clear (unencrypted). For this reason, we do *not* recommend logging in remotely over the Internet.

If the Linux virtual machine is installed with Red Hat (all versions apply), then you will have to perform some configuration to get it to work like a SUSE system. For further information about this topic, refer to the site “Red Hat knowledge base: How do I set up VNC based on xinetd with XDMCP for GDM?” at:

http://kbase.redhat.com/faq/docs/DOC-2517

> **Note:** This procedure differs in that it does not include a reboot of the system. Because we implemented this on a System z s390, the steps are documented here.

1. Edit `/etc/gdm/custom.conf` and add the content listed in Example B-9 under the corresponding sections.

Example B-9 /etc/gdm/custom.conf

```
[daemon]
RemoteGreeter=/usr/libexec/gdmgreeter
[security]
AllowRoot=true          # only add if root needs to login
AllowRemoteRoot=true    # not recomended but documentation
```

```
[xdmcp]
Enable=true
```

2. Create the file `/etc/xinet.d/xvncserver` with the contents of Example B-10.

Example B-10 /etc/xinet.d/xvncserver

```
service vnc01 {
disable = no
protocol = tcp
socket_type = stream
wait = no
user = nobody
server = /usr/bin/Xvnc
server_args = -inetd -query localhost -once -geometry 1024x768
-depth 16 securitytypes=none }
```

3. Add the content shown in Example B-11 to the `/etc/services` file.

Example B-11 /etc/services

```
vnc01     5901/tcp     # GDM on VNC via xdmcp remote
```

4. Reload the `xinetd config` file that was just updated so it will reflect the changes, as shown in Example B-12.

Example B-12 Reload xinetd config

```
service xinetd reload
```

5. Change the firewall so that it will trust the Ethernet interface to allow for `xdmcp` to work. Run `system-config-securitylevel-tui` and chose **[Customize]**. Next, type an asterisk (*) in the Trusted Devices box. Click **Ok** and then click **Ok** again to exit and activate the changes.
6. Start `gdm`, as shown in Example B-13.

Example B-13 Activating gdm

```
/usr/sbin/gdm
```

7. Put the system in runlevel 5, as shown in Example B-14. You will also need to change the runlevel in `/etc/inittab` from 3 to 5 to make this change permanent.

Example B-14 Changing the runlevel to 5

```
telinit q ; telinit 5
```

8. At this point VNC is configured to serve GDM (which is the Gnome great program that will allow logon) on the VNC port one. You can test the status by using your VNC client as shown in Example B-15, where host name is the address of the Linux VM.

Example B-15 Testing VNC

```
vncviewer hostname:1
```

Configuring a VNC server per user

To start a VNC server, from a command prompt simply execute the **vncserver** command. Unfortunately, the VNC server default configuration is not particularly useful. This section explains how to configure the per user settings to make the VNC server usable. The configuration files are kept in the user's home directory as in ~/.vnc, but that directory will not exist until a VNC command is run. Therefore, first set the required VNC password as shown in Example B-16.

Example B-16 Setting VNC server password

```
$ vncpasswd
Password:
Verify:
```

Now you will find that the ~/.vnc directory exists and it contains the file passwd. The next time you run the **vncserver** command, it will start the server and create a default xstartup file. However, this default startup file will default to primitive settings that are not likely to be useful. Use the content shown in Example B-17 for a Red Hat server.

Example B-17 ~/.vnc/xstartup file for Red Hat

```
#!/bin/sh
[ -x /etc/vnc/xstartup ] && exec /etc/vnc/xstartup
[ -r $HOME/.Xresources ] && xrdb $HOME/.Xresources
vncconfig -iconic &
exec /usr/bin/gnome-session
```

Use the content shown in Example B-18 for a SUSE server.

Example B-18 /.vnc/xstartup file for SUSE

```
#!/bin/sh
xrdb $HOME/.Xresources
exec /usr/X11R6/bin/fvwm
```

After ~/.vnc contains a passwd and xstartup file, you can start the server via the content shown in Example B-19. For a complete explanation of all VNC server options, refer to the man page.

Example B-19 Starting up a VNC server

```
$ vncserver -geometry 1024x768 -depth 32

New 'some.server.name:1 (user)' desktop is some.server.name:1

Starting applications specified in /home/user/.vnc/xstartup
Log file is /home/user/.vnc/some.server.name:1.log
```

To shut down the server when it is no longer needed, run the content shown in Example B-20. (Notice that :1 was used to match the port started in Example B-19.)

Example B-20 Shutting down a VNC server

```
$ vncserver -kill :1
Killing Xvnc process ID 7422
```

VNC client

There are many commercial and FOSS versions of the VNC client. The important point to keep in mind is to match the client with the server. Some VNC client s are promoted as having automatic detection, but this function is not needed because there are only two common servers:

- Real vnc - for more information refer to:

  http://www.realvnc.com

- Tight vnc - for more information, refer to:

  http://www.tightvnc.com

Therefore, simply match the client with the server. In our case, we used standard VNC or vncserver.

Under Linux

The VNC Viewer application is available for all Linux distributions.

Figure B-6 on page 259 shows an example of a remote desktop from a Red Hat server running as a guest under z/VM (Linux on System z) with the optimized settings from Example B-17 on page 257.

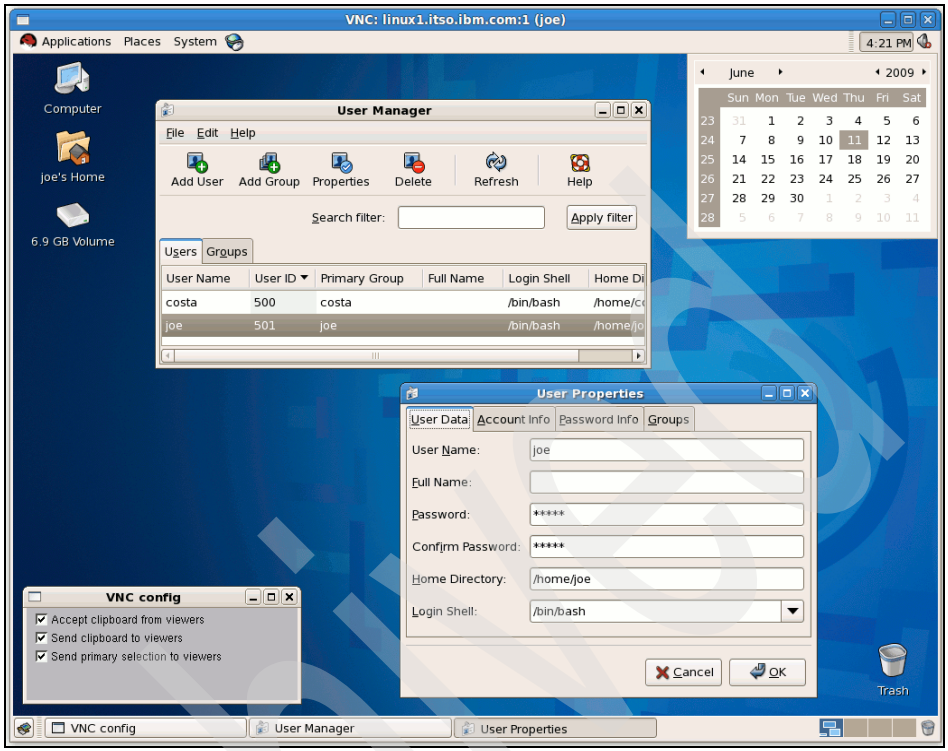

Figure B-6 VNC client running on a Linux Desktop from a Red Hat server

Under MS Windows

The MS Windows clients can be downloaded from the respective Web sites and installed.

Figure B-7 on page 260 shows an example of a remote desktop from a SUSE server running as a guest under z/VM with the optimized settings from Example B-18 on page 257.

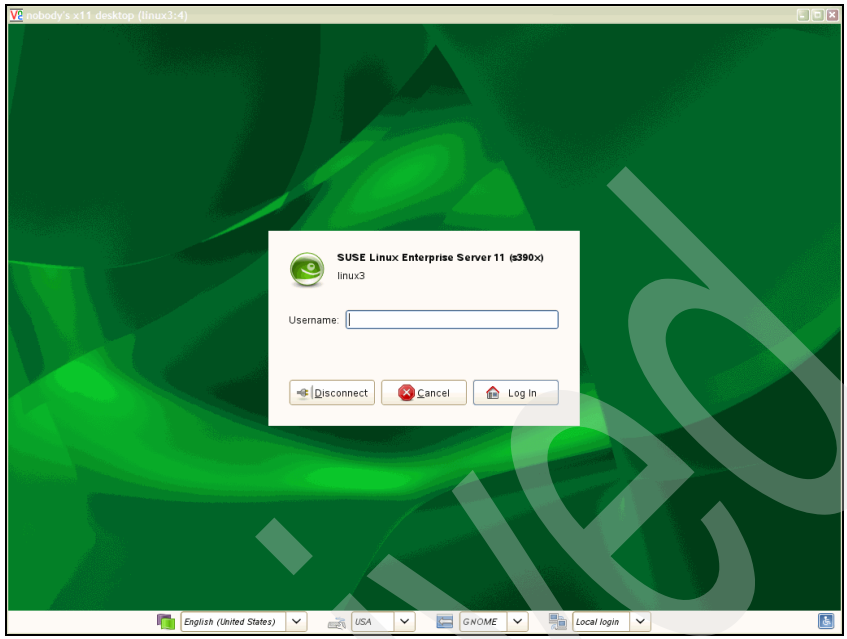

Figure B-7 VNC client from MS Windows connected to a SUSE server and receiving the login window

Figure B-8 on page 261 shows a VNC client on MS Windows with a SUSE desktop.

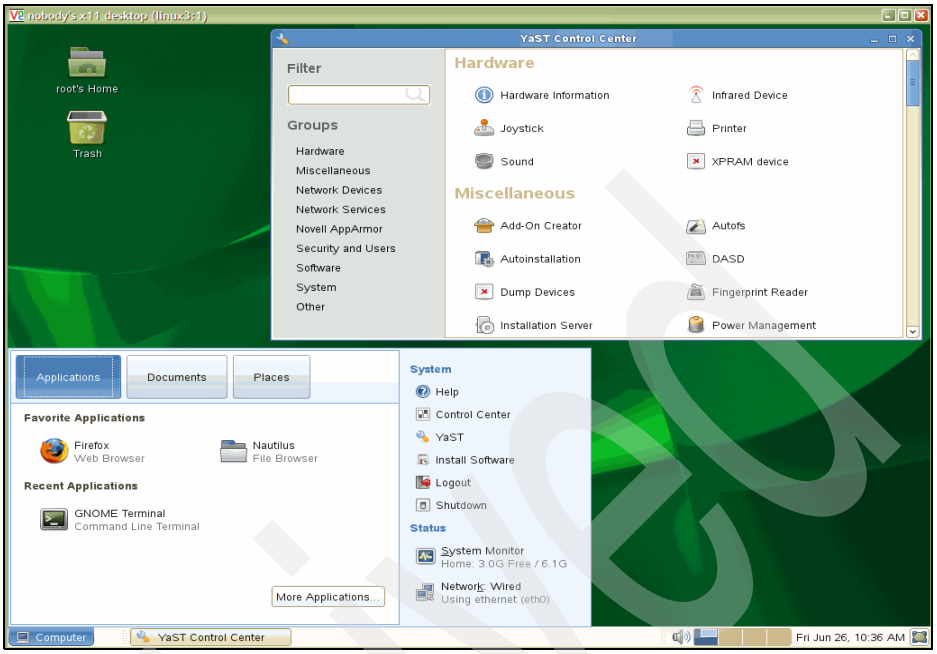

Figure B-8 VNC client from MS Windows connected to the SUSE Desktop

C

Performance measurement

In this appendix we discuss performance measurement and its impact on the success of an application migration. The most important point is that you need to measure the performance of the source application when running in production, and then to compare that with the performance of the application on the target environment. The key is to choose measurable statistics to identify the performance improvement provided by the target environment.

We also discuss monitoring commands and tools that can assist you in identifying and resolving performance inhibitors.

What is performance

"Performance" in computer systems is very much a relative term. Usually computer performance is discussed in measurable terms such as transactions per second, response time, time to process a booking or insurance sale, and so on. However, when a migration is undertaken, it is important to understand the performance metrics used on the source environment so that you can understand the relative performance improvement on the target system.

The initial performance of a new system may often not be as expected, so tuning must be undertaken to improve the performance of the target system. Without having proper metrics, it is impossible to validate the performance of the new platform relative to the former platform. For this reason, the migration project team first needs to agree on what performance metrics from the source platform will be used in the migration project plan to measure the performance of the target platform.

Choosing what to measure

To determine the success of a migration, simply having the application on the target platform provide the same answers as the source platform does not prove success. The natural expectation of a migration onto Linux on System z is that the application will not only be more resilient and available because of System z, but that it will also provide equal or better performance than the source platform. To ensure that the performance improvements are easy to show, it is important to choose the right metrics. But what are these metrics, and how should they be measured?

Response time

Response time is the measure of the time it takes for something to happen in a computer system. Generally we chose to measure the response time of a unit of work called a *transaction*. This could entail something as simple as checking an account balance, to something as complex as the time taken to issue a new insurance policy or open a new bank account.

The point to remember with computer systems is that the response time of a single transaction is the sum of a number of response times. Figure C-1 on page 265 shows the various components that make up user response time.

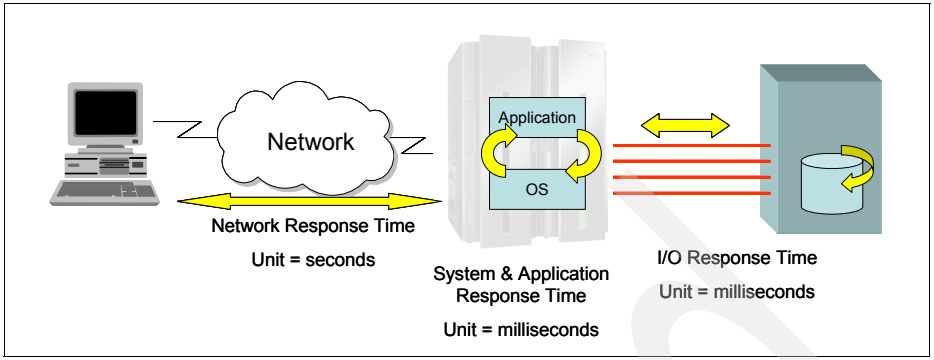

Figure C-1 Components that make up the response time of transactions

The figure shows that there are two points where response time could be measured: system response time and user response time. When you are trying to understand the relative performance improvement from a new system, the only point to measure response time is from when a system receives the request and when it provides a response of some sort to the request.

In the case illustrated in Figure C-1, the system response time will include application time and the I/O response time to access the data. If you choose the response time a user experiences at their terminal or over the Web, you will be adding in the network response time, which can vary greatly for the same transaction because it can be influenced by network load.

To compare the source and target systems directly, the recommended approach is to measure system response time on the source system and, assuming the application has not changed greatly, measure the system response time on the target platform.

Transaction throughput

The transaction throughput performance metric may provide a more meaningful measure of system performance because it measures the number of transactions processed over a period of time. This is typically one second, but could be any time period you prefer.

In both cases, you should have baseline performance metrics for the source system to properly compare both the new and old systems.

Performance and the proof of concept

Provided that the environment for the proof of concept (POC) is identical to the final target environment, you can use the proof of concept to achieve a solid understanding of the expected performance and throughput of the new system. However, the results of the POC may not be duplicated on the final platform because, in many cases, the migration is used as an opportunity to add additional functionality to the application. Depending on what is done, this could have a negative impact on the performance of the target system.

Another factor to take into consideration is that the sizing of the target environment was done with the assumption that the source application would not be changed. If there were additional functions added to the source application, however, then you may find that the target environment is not large enough to provide adequate performance.

For this reason, if you plan to modify the source application during the migration, stress test the application before going into production to ensure that the target configuration is large enough to accommodate the modified target application. One tool you can use to accomplish this is the IBM Rational Performance Tester. For more information, see:

http://ww-01.ibm.com/software/awdtools/tester/performance

Configuration tips

The z/VM and Linux environment will execute in one or more logical partitions (LPARs). Each LPAR has logical CPUs, memory, and I/O resources assigned to it. The logical CPUs can be dedicated to a single LPAR or shared among a number of LPARs.

Memory is also allocated to each LPAR. Perform a thorough analysis of the intended workloads to ensure that enough memory is allocated. Each virtual machine directory entry states a minimum and maximum (they can be the same) amount of virtual storage that the virtual machine can use. It is important to ensure that the total amount of virtual storage in use by concurrent virtual machines does not exceed the recommended overcommitment ratio of 2:1 for production workloads, or 3:1 for development/test workloads.

I/O devices are generally shared among virtual machines, and this is also defined in the z/VM directory. When required, physical disk volumes can also be dedicated to individual virtual machines. Network connections between virtual machines should use redundant Vswitches (Virtual LANs).

Network connections between LPARs to other virtual machines or to z/OS should use HiperSockets. Physical connections outside of the System z footprint use the Open Systems Adapter (OSA). Depending on the security policy in place in your organization, you may need dedicated OSAs if you need to keep the Linux network traffic separate from z/OS.

Configuration recommendations

When configuring for availability and performance you should have a minimum of three IFLs and two LPARs. Each LPAR should be assigned two logical CPUs. Each virtual machine running Java should have two virtual CPUs because this improves Java dispatcher performance.

Size the memory allocated to each LPAR according to the 2:1 memory overcommitment ratio for production and 3:1 for development.

Further configuration recommendations for z/VM and Linux are available at:

http://www.ibm.com/developerworks/linux/linux390/index.html

Performance tuning

If the performance of the target Linux on z/VM environment is unsatisfactory you need to tune the various IT components to improve performance. To understand what must be tuned you will need monitoring products such as:

- IBM Performance Toolkit for VM
- IBM Tivoli Omegamon XE on z/VM and Linux

Linux also includes system tools such as:

- vmstat
- top
- Process Status (ps)

Additionally, there are a number of Open Source tools for Linux such as:

- Nagios

  This tool provides comprehensive monitoring all mission-critical infrastructure components, including applications, services, operating system, network protocols, system metrics and network information. The Nagios tool was installed on our Linux on System z server.

- Xymon (previously known as Hobbit)

  This is a system for real-time monitoring of hosts and networks with a Web-based interface with availability reports and performance graphs.

The areas that typically need tuning to improve performance are:

- Processor sharing parameters
- Memory sizing
- Disk I/O
- Network I/O

Tools such as IBM Tivoli Omegamon XE on z/VM and Linux will in many cases identify where any contention is occurring. This will allow you to solve the problem more quickly.

For a comprehensive analysis of performance tuning for a Linux on System z environment, refer to *Linux on IBM System z: Performance Measurement and Tuning*, SG24-6926.

Glossary

Central Processor (CP). System z processor that can run all IBM System z operating systems.

FlashCopy. Instant copy of data created by the storage controller copying pointers to data rather than the data itself.

Globally Dispersed Parallel Sysplex (GDPS). A Parallel Sysplex that is distributed across separate sites. Metro Mirror is used to synchronize data across remote sites.

HiperSockets. High performance memory-to-memory virtual network for communications between LPARs on IBM System z.

Integrated Facility for Linux (IFL). System z specialty engine that can only run z/VM and Linux.

ISV. An independent software vendor.

Linux guest. A virtual machine under z/VM running the Linux Operating System.

logical CPU. CP or IFL defined to an LPAR. A logical CPU can be dedicated to an LPAR or shared between LPARs.

LPAR. A logical partition.

Metro Mirror. Data replication technique designed to replicate data across distances up to 100 km. Metro Mirror uses PPRC.

minidisk. A virtual disk occupying a portion or whole of a physical volume created by z/VM and allocated to virtual machines. Minidisks can be shared in read only mode to all virtual guests.

network segment. A portion of a network that is separated from the rest of the network by a device such as a repeater, hub, bridge, switch or router.

physical CPU. Actual physical CP or IFL activated on the IBM System z.

PPRC. Peer-to-peer remote copy, a synchronous data mirroring technique available on IBM DS8000 and DS6000 storage subsystems.

PR/SM. A virtualization layer built into IBM System z. Allows the creation of logical partitions (LPARs) for workloads to execute in.

ROI. Return on Investment, which is the amount of time it takes to recover the TCA from savings generated by a project.

source server. The server being migrated from.

target server. The server being migrated to.

TCA. Total Cost of Acquisition, which is the cost of purchasing hardware and software for a project. This should always be included in the TCO numbers.

TCO. Total Cost of Ownership, which includes all costs of a project over its expected lifetime, usually three or five years.

virtual CPU. CPU defined to each virtual machine in the z/VM Directory.

virtual guest. A virtual machine running under z/VM.

Vswitch. A virtual LAN managed by z/VM.

z/VM. A virtualization hypervisor for IBM System z.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

For information about ordering these publications, see “How to get Redbooks” on page 274. Note that some of the documents referenced here may be available in softcopy only.

- *IBM System z Connectivity Handbook*, SG24-5444
- *OSA-Express Implementation Guide*, SG24-5948
- *Linux on IBM eServer zSeries and S/390: Performance Toolkit for VM*, SG24-6059
- *Implementing the IBM System Storage SAN Volume Controller V4.3*, SG24-6423
- *Experiences with Oracle 10g Database for Linux on zSeries*, SG24-6482
- *WebSphere Application Server Network Deployment V6: High Availability Solutions*, SG24-6688
- *Linux on IBM eServer zSeries and S/390: Application Development*, SG24-6807
- *Hipersockets Implementation Guide*, SG24-6816
- *SAP on DB2 9 for z/OS: Implementing Application Servers on Linux for System z*, SG24-6847
- *Linux on IBM System z: Performance Measurement and Tuning*, SG24-6926
- *IBM Lotus Domino 6.5 for Linux on zSeries Implementation*, SG24-7021
- *Linux on IBM eServer zSeries and S/390: Best Security Practices*, SG24-7023
- *Linux with zSeries and ESS: Essentials*, SG24-7025
- Experiences with Oracle 10gR2 Solutions on Linux for IBM System z, SG24-7191
- *Solaris to Linux Migration: A Guide for System Administrators*, SG24-7196
- *Fibre Channel Protocol for Linux and z/VM on IBM System z*, SG24-7266

- *Introduction to the New Mainframe: z/VM Basics*, SG24-7316
- *IBM System z Strengths and Values*, SG24-7333
- *High Availability and Disaster Recovery Options for DB2 on Linux, UNIX, and Windows*, SG24-7363
- *Using the SVC for Business Continuity*, SG24-7371
- *Security on z/VM*, SG24-7471
- *z/VM and Linux on IBM System z The Virtualization Cookbook for Red Hat Enterprise Linux 5.2*, SG24-7492
- *z/VM and Linux on IBM System z The Virtualization Cookbook for SLES 10 SP2*, SG24-7493
- *IBM System z10 Enterprise Class Technical Introduction*, SG24-7515
- *SAN Volume Controller Best Practices and Performance Guidelines*, SG24-7521
- *Using Oracle Solutions on Linux for System z*, SG24-7573
- *SAN Volume Controller V4.3.0 Advanced Copy Services*, SG24-7574
- *Problem Determination for Linux on System z*, SG24-7599
- *z/VM and Linux Operations for z/OS System Programmers*, SG24-7603
- *Experiences with Oracle Solutions on Linux for IBM System z*, SG24-7634
- *Linux on IBM zSeries and S/390: Securing Linux for zSeries with a Central z/OS LDAP Server (RACF)*, REDP-0221
- *Linux on IBM eServer zSeries and S/390: VSWITCH and VLAN Features of z/VM 4.4,* REDP-3719
- *Open Your Windows with Samba on Linux*, REDP-3780
- *Networking Overview for Linux on zSeries*, REDP-3901

  Network concepts are described well. Linux network options are for the older Linux 2.4 kernel.
- *Clustering Solutions Overview: Parallel Sysplex and Other Platforms*, REDP-4072
- *Linux Performance and Tuning Guidelines*, REDP-4285
- *The Green Data Center: Steps for the Journey*, REDP-4413
- *The Green Data Center: An Idea for Today*, REDP-4523
- *Implementing the Poughkeepsie Green Data Center--Showcasing a Dynamic Infrastructure*, REDP-4534

Other publications

These publications are also relevant as further information sources:

- *CP Commands and Utilities Reference*, SC24-6081
- *Getting Started with Linux on System z Version 5 Release 4*, SC24-6096
- *Linux on System z Device Drivers, Features, and Commands*, SC33-8411
- *IBM Journal of Research and Development Vol 53, No. 1 Paper 11 2009*

  http://www.research.ibm.com/journal/rd/531/clarke.pdf
- Eilert, Eisenhaendler, Matthaeus, Salm, *Linux on the Mainframe*, Prentice Hall, 2003, ISBN 0-13-101415-3
- Mendoza, Skawratananond, Walker, *UNIX to Linux Porting*, Prentice Hall, 2006, ISBN 0-13-187109-9

Online resources

These Web sites are also relevant as further information sources:

- Linux on IBM System z home page:

  http://www-03.ibm.com/systems/z/os/linux/
- Linux on IBM System z library:

  http://www-03.ibm.com/systems/z/os/linux/library/index.html
- z/VM Resources for Linux on System z

  http://www.vm.ibm.com/linux/
- z/VM publications:

  http://www.vm.ibm.com/pubs/l
- z/VM performance tips:

  http://www.vm.ibm.com/perf/tips
- z/VM performance publications:

  http://www.vm.ibm.com/perf/pubs/
- Documentation for Development Stream Site (publications such as Device Drivers, Features and Commands, Using the Dump Tools, and Kernel Messages; also s390 tools download)

  http://www.ibm.com/developerworks/linux/linux390/development_recommended.html/

- Technical hints and tools to help simplify porting applications to System z

  http://www.ibm.com/developerworks/linux/library/l-systemz/
- IBM DeveloperWorks Tuning hints and tips

  http://www.ibm.com/developerworks/linux/linux390/perf/index.html/
- Porting Central

  http://www-03.ibm.com/servers/enable/site/porting/linux/zseries/l
- IBM DeveloperWorks Linux on System z home page

  http://www-03.ibm.com/developeworks/linux/linux390/index.html
- Linux-S390 list server

  http://www2.marist.edu/htbin/wlvlists
- IBM middleware available on Linux (download PDF and click System z)

  http://www-03.ibm.com/linux/matrix/
- Relevant SHARE presentations

  http://www.linuxvm.org/
- Web 2.0 installation manual for SUSE

  http://download.boulder.ibm.com/ibmdl/pub/software/dw/linux390/docu/lslsw200.pdf
- Web 2.0 installation manual for Red Hat

  http://download.boulder.ibm.com/ibmdl/pub/software/dw/linux390/docu/lrhlw201.pdf
- Linux man pages

  http://linuxcommand.org
- Rosetta Stone or Sysadmin Universal Translator

  http://www.bhami.com/rosetta.html

How to get Redbooks

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks publications, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

M

N

O

P

Q

R

S

T

Practical Migration to Linux on System z

(0.5” spine)
0.475”<->0.875”
250 <-> 459 pages

Practical Migration to Linux on System z

Overview and migration methodology

Migration analysis

Hands-on migration case study

There are many reasons why you would want to optimize your servers through virtualization using Linux on System z.

- Your enterprise may have too many distributed physical servers with low utilization.
- There may be a lengthy provisioning process that delays the implementation of new applications.
- There may be limitations in data center power and floor space.
- The enterprise may experience a high Total Cost of Ownership (TCO).
- There may be difficulty in allocating processing power for a dynamic environment.

This IBM Redbooks publication provides a technical planning reference for IT organizations that are considering a migration to Linux on System z. The overall focus of the content in this book is to walk the reader through some of the important considerations and planning issues that you could encounter during a migration project.

Within the context of a preexisting UNIX-based or x86 environment, we present an end-to-end view of the technical challenges and methods necessary to complete a successful migration to Linux on System z.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7727-00 ISBN 0738433411