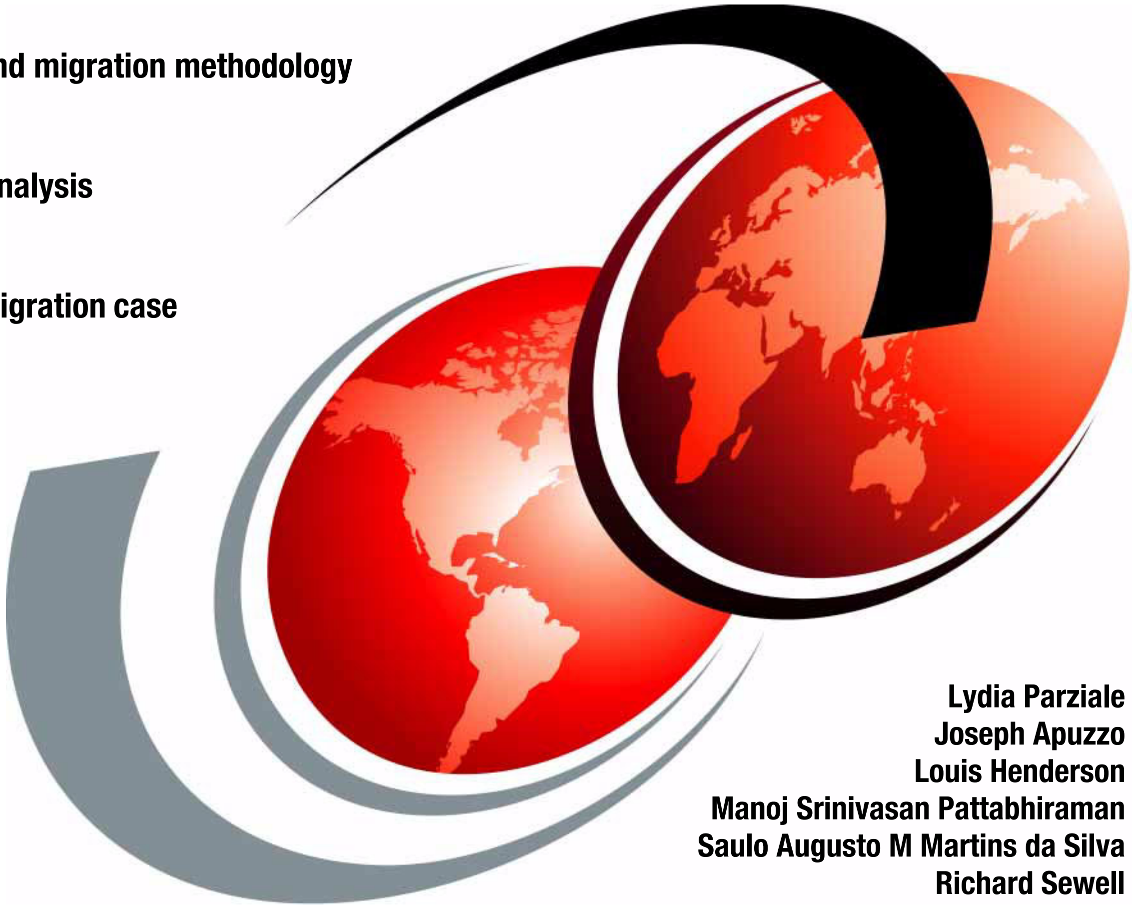
Draft Document for Review September 2, 2009 5:38 pm
SG24-7727-00

Practical Migration to Linux on System z

Overview and migration methodology

Migration analysis

Hands on migration case study



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

Redbooks

ibm.com/redbooks

International Technical Support Organization

Practical Migration to Linux on System z

May 2009

SG24-7727-00

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

First Edition (May 2009)

This edition applies to z/VM Versions 5.3 and 5.4. Novell Suse Linux Enterprise Server versions 10 and 11 and Red Hat Enterprise Linux version 5.3.

This document created or updated on September 2, 2009.

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

Preface

There are many reasons why you would want to to optimize your servers through virtualization using Linux® on System z®, some of them are:

- Too many distributed physical servers with low utilization
- A lengthy provisioning process that delays the implementation of new applications
- Limitations in data center power and floor space
- High Total Cost of Ownership (TCO)
- Difficulty 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 attempt to 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 that 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 and has been employed by IBM for 24 years in various technology areas.

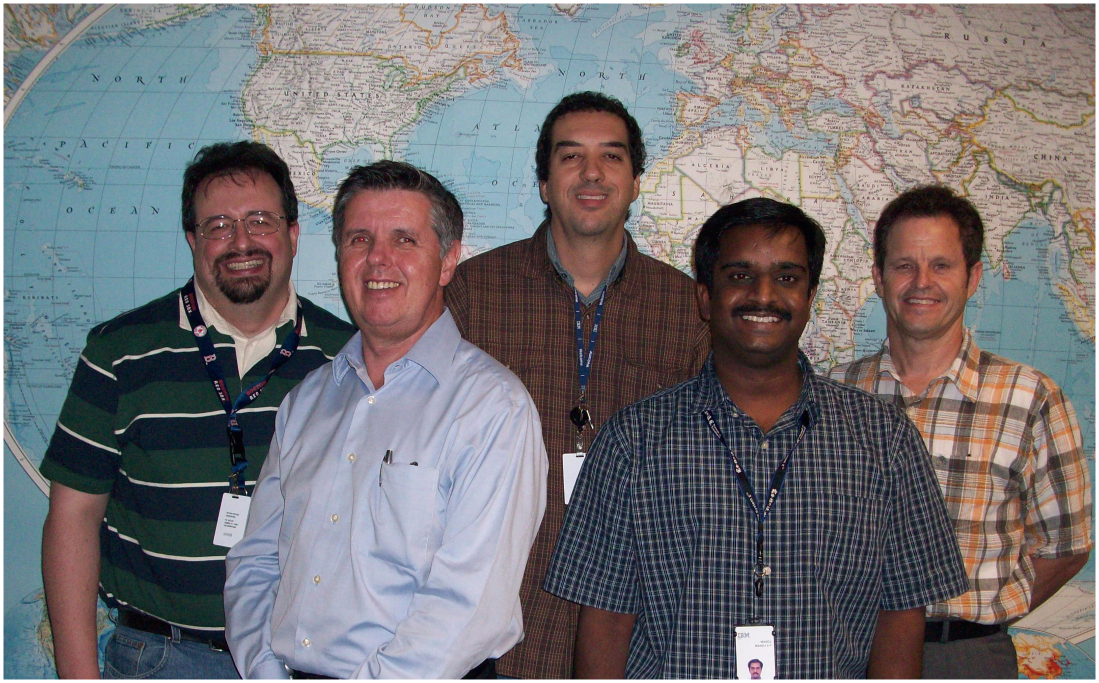


Figure 0-1 From left: Joseph Apuzzo, Richard Sewell, Saulo Augusto M Martins da Silva, Manoj Srinivasan Pattabhiraman and Louis Henderson.

Joseph Apuzzo is an Advisory Programmer from Poughkeepsie New York, supporting IBM's 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. He holds a Master of Computer Science from Pace University with an undergraduate degree in Electrical Engineering Technology from SUNY Binghamton. He is a subject matter expert on IBM's GPFS™ file system which runs on AIX®, Linux and MS® Windows®. In his personal time he supports state and local Linux users groups (www.mhvlug.org)

Louis Henderson is a systems programmer/consultant living on the west coast of the U.S. He started his career with EDS World Corp. as a COBOL and Assembler programmer. He has worked as a system programmer and consultant in many locations around the world. He has over 30 years experience with various mainframe technologies. He holds an undergraduate degree in Chinese Language and Literature and a Masters Degree in Business. His current interests are 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 that 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 INDIA/ASEAN Region. Manoj has contributed to several IBM Redbook publications and has been a presenter at various international technical conferences. His areas of expertise include z/VM, Linux on System z developement, Migration and Porting, Performance & Tuning, and z/VM REXX programming.

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

Richard Sewell is a System z Architect working with financial services customers for IBM in Sydney, Australia. He has over 30 years 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 expertise is in operating systems including z/OS®, z/VM, Linux, capacity planning, training and design of technology solutions based on IBM system z. He has 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

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

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

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

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

Thanks to Priya Sekaran, Senior Web Developer who provided us with support and valuable inputs on 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

A decision to undertake the migration of a software application is never taken lightly. Who hasn't heard of migrations that have gone amiss 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 sucessfull 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

Introduction

In this chapter we discuss the reasons why migrations happen 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 of a Linux on IBM System z environment.

1.1 Why 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 migration of a complex heavily integrated application running across multiple servers, architectures and possibly sites. No matter how big or small the migration the main reasons why migrations are undertaken is usually one or a combination of the following:

- Reduce server sprawl through consolidation
- Reduce power and cooling requirements
- Regain data center space
- Contain or reduce costs
- Re-engineer applications
- Reduce complexity
- Replace un-supported 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, include:

- Expiry of distributed server leases
- Expiry of software contracts

In all of these cases the driving imperative to migrate 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 solve many of the problems associated with the growth of distributed servers through consolidation of multiple physical servers to a virtualized environment on servers with a greater number of cores. While this reduces the number of physical servers it may not substantially reduce the power and cooling requirements nor 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 floorspace of their two data centers. With over 5,000 distributed servers the remaining space is take up by racks of distributed servers, distributed storage, System z storage and networking equipment.

In Figure 1-1 we see a photograph of a small to medium size server farm.

Figure 1-1 A medium sized server farm

As 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 very fragile as Figure 1-2 indicates.

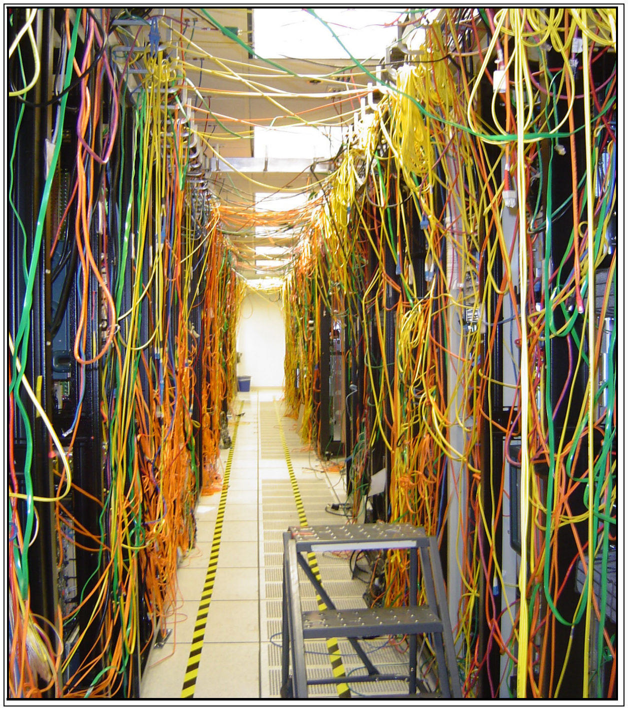


Figure 1-2 Network complexity

Whilst there is nothing inherently wrong with large server farms they do have a number of major problems. Perhaps the most critical of these is their complexity which requires very skilled and experienced people to manage them. Other factors are their power and cooling requirements and the very complex issues of managing the life cycle of so many assets. For example, when old servers are retired do they get switched off and removed from the data center? Or are they left continully powered-up with nothing running on them since 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 and 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 didn't build one of their data centers on the Columbia River near a hydro-electric power station in Oregon for only the view[1].

The health of our environment is crucial to us all and a not-for-profit organisation, The Carbon Disclosure Project (CDP), has been established to seek information on greenhouse gas emissions from the world's largest companies. They have

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

conducted annual surveys over the last 8 years and the CDP has become the gold standard for carbon disclosure methdology and process.

In a survey response to The Carbon Disclosure Project in 2007, a major Australian bank stated that their largest data center represents 15% of their total stationary energy expenditure. In the same survey another major Australian bank stated that their data and transaction processing centers account for almost 40% of their energy use. These are significant numbers and provide an opportunity for organisations to rein in their distributed server sprawl and substantially reduce power and cooling requirements by consolidation to a virtualised platform such as that provided by IBM System z.

1.1.2 IBM's Big Green Server Consolidation

In 2007, IBM anounced that it will consolidate about 3,900 servers onto about 30 IBM System z's running Linux. The new server environment will consume 80% less energy and occupy 85% less floor space than the original 3,900 servers. Estimated savings are more than $250 million over 5 years in energy, software and systems support costs. While most companies will not achieve this level of savings there are still significant savings to be gained with a consolidation of distributed servers to Linux on System z.

Note: For more details of 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 Why Migrate to Linux?

One of the main benefits of a migration to Linux is that it allows organizations to break the link between the operating system and specific hardware platforms. What this means is that once 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. The following diagram describes the commercial IT platforms and IBM products that Linux supports.

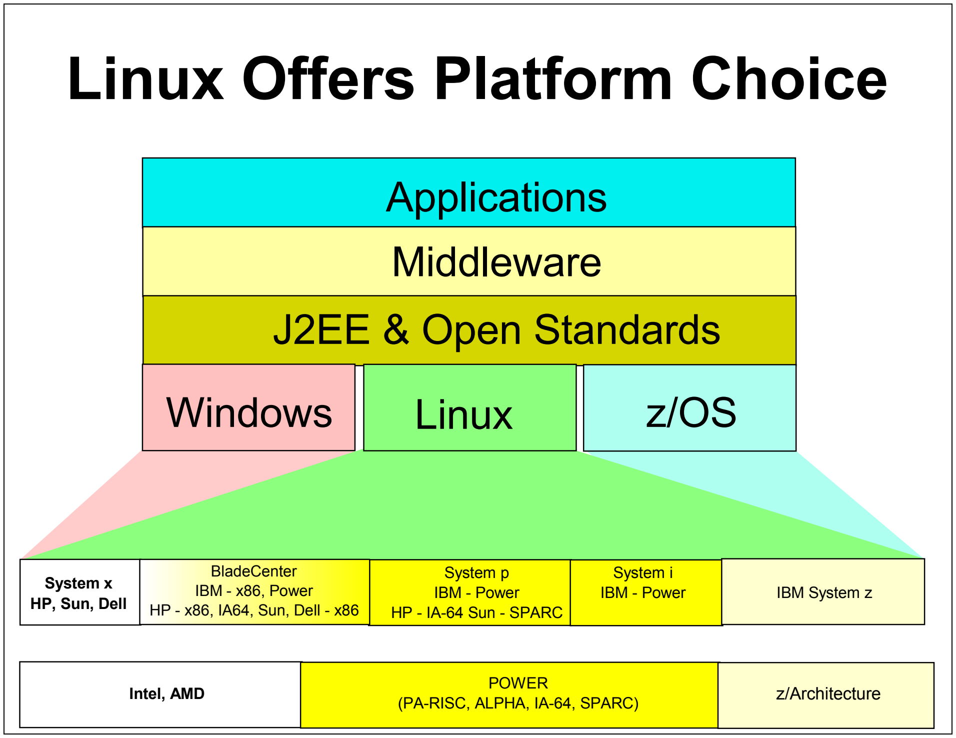


Figure 1-3 14Wide Variety of Commercial IT Platforms Supported by Linux

One of the major benefits of Linux is that it is Open Source. What this means is that 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 ever invented. Two enterprise distributions[2] of Linux are from:

- Red Hat® - Red Hat Enterprise Linux

http://www.redhat.com

- Novell® - SUSE Linux Enterprise Server

http://www.novell.com

What both Red Hat and Novell provide for customers using Linux is a variety of support options including 24 x 7 support with 1 hour response time worldwide for 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 in order 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 reason for the increased interest and uptake of Linux are its rich set of features, including virtualization, security, Microsoft® Windows interoperability, development tools, a growing list of Independent Software Vendor's applications, performance and, most importantly, its multi-platform support.

This multi-platform 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.

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

[3] MIPS (Millions of Instructions Per Second) 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.

1.3 Why 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 over 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 is the ideal platform for consolidation of 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 with the Enterprise Class being able to scale to 64 processors and the Business Class to 10 processors.

IBM System z servers are the cornerstone of a dynamic architecture that helps you transform IT to take advantage of IBM's 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, both small and large businesses can wrap System z's enterprise benefits and advantages around a common open platform. Developers can produce applications that deploy on cell phones, laptops and also Linux vitual 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]
- 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 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 of Common Criteria, Evaluation Assurance Levels, Protection Profiles and a list of certified products go to:
>
> `http://www.commoncriteriaportal.org`
>
> The certified evaluation levels for System z Operating Systems, as at 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+

- Industry leading virtualization hypervisor, z/VM, 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™: up to 16 virtual LANs allowing memory to memory TCP/IP communication between LPARS

- Scalability
  - System z10 EC scales to 64 application processors and up to 1.5TB of memory
  - System z10 BC scales to 10 application processors and up to 248 GB of memory
- Security

[5] PR/SM is standard component of all IBM System z models . It is a hypervisor that enables logical partitions (LPAR) 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.

  - Clear key integrated cryptographic functions providing 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 the 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 DR 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 will reduce 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 will allow tracking of trends which will provide 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 on migrating to Linux on System z:

- Reduced Total Cost of Ownership (TCO)
  - Environmental savings - single footprint vs. hundreds of servers

  - Consolidation savings - less storage, less servers, less 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

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

For the first migration it is recommended to choose an application that is not overly complex and the Linux on System z platform is supported by the ISV or in the case of a home grown application the source code is available. Many ISVs support Linux on System z and these include but are not limited to IBM, Oracle®, SAP®, IBI, Red Hat & Open Source. Applications that require close proximity to corporate data stored on the mainframe are also ideal candidates as well as applications that have high I/O rates since I/O workloads are off-loaded 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). As 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 the like.

In Chapter 8, “Application analysis” on page 91 we will provide an in depth analysis of the process of determining the most appropriate application to migrate to a Linux on System z environment.

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

1.4 z/VM Virtualization for Linux on IBM System z

Virtualization of computing systems, contrary to popular belief, is not a new 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 shows the development history of z/VM.

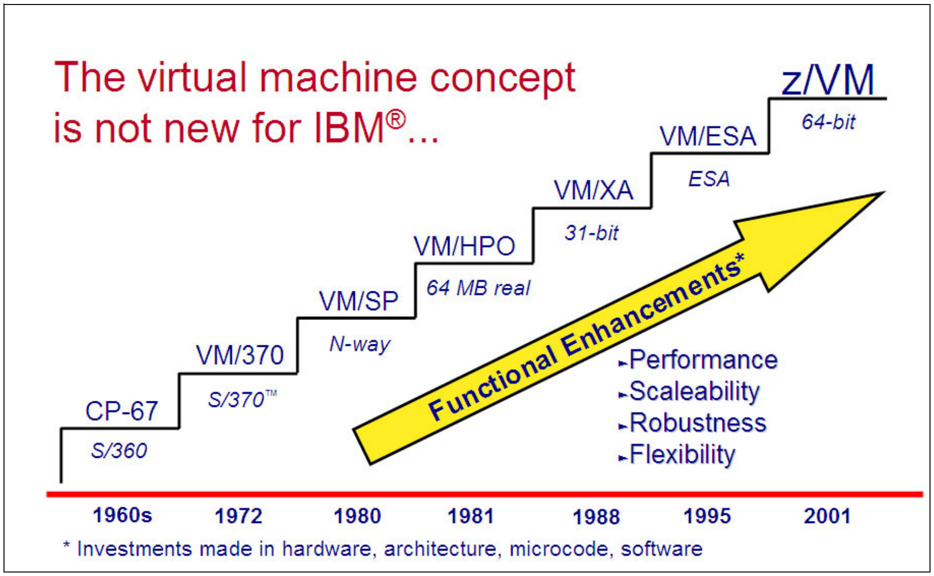


Figure 1-4 20Family History of z/VM

The role of a virtualization hypervisor is to share physical hardware resources so that a single physical server can act a 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 they are the only server using the hardware. As some servers are more important than others, z/VM also manages the scheduling[7] of virtual machines according to its defined priority.

[7] Scheduling is the process where 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.

By virtualizing the IBM System z hardware with z/VM, it is possible to run the system at a very high utilization rate in order to get 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 machines. More technical information on z/VM is provided in Chapter 5, “Technical concepts” on page 55

1.4.1 Leveraging the Power of z/OS for Distributed Platforms

One of the major benefits 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 data bases such as DB2® and IMS and having distributed applications in the same System z footprint can eliminate many of the physical network bottlenecks and turbo-charge the applications.

In a multi-tier 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 bottle necks.

1.4.2 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 illustrates the various components of Linux on System z.

.

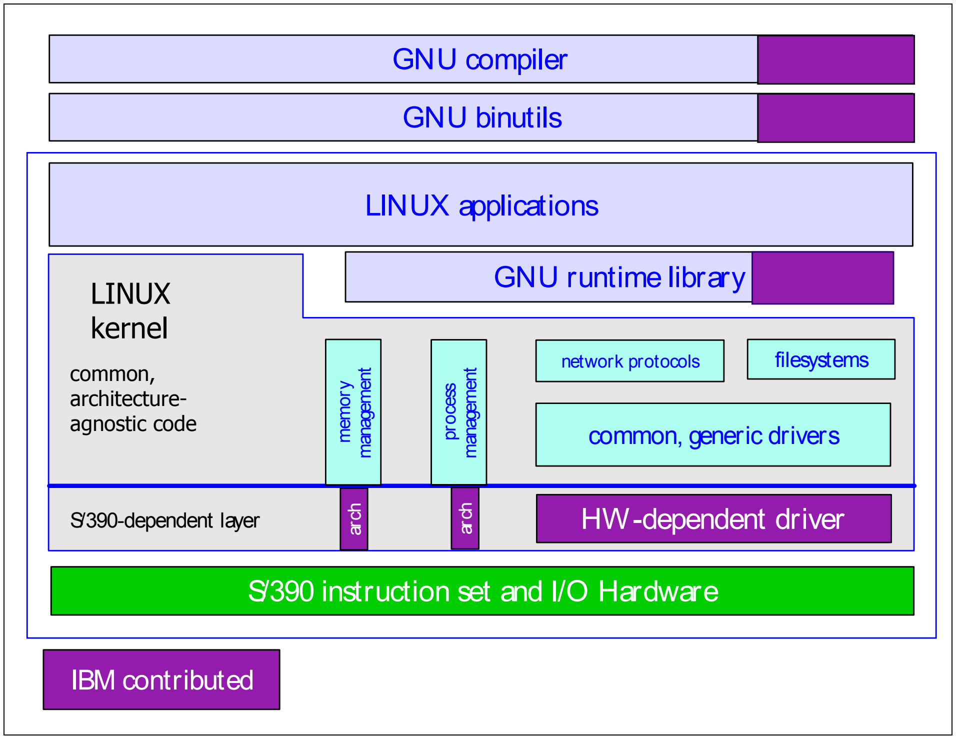


Figure 1-5 Linux on IBM System z

While 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 or CP is a general purpose processor that can run z/OS, z/VM, Linux, z/VSE, z/TPF. On IBM System z10 CPs are available in a large number of capacity ratings in order to ensure that customer s pay only for what they need

1.5 Sizing Workloads for Migration

We have discussed the benefits of the System z platform in its role as an enterprise Linux server but 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(s) of the servers to be consolidated
- CPU utilization of these servers
- Workload characteristics of applications to be consolidated

While 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 IBM’s 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, i.e. database management system, Java™, I/O bound, compute bound etc.
  - Costs
    - Software license and maintenance costs
    - Hardware purchase & maintenance costs
    - Staff costs
    - etc.

The outputs from the tool are:

- Number of IFLs required to support distributed workload
- Amount of memory required

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

- Total Cost of Ownership (TCO) analysis of the various configuration options. This is based on cost inputs in the model.

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 has 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 servers 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 bechmark in terms of response time and throughput appear in Figure 1-6 and Figure 1-7 on page 25.

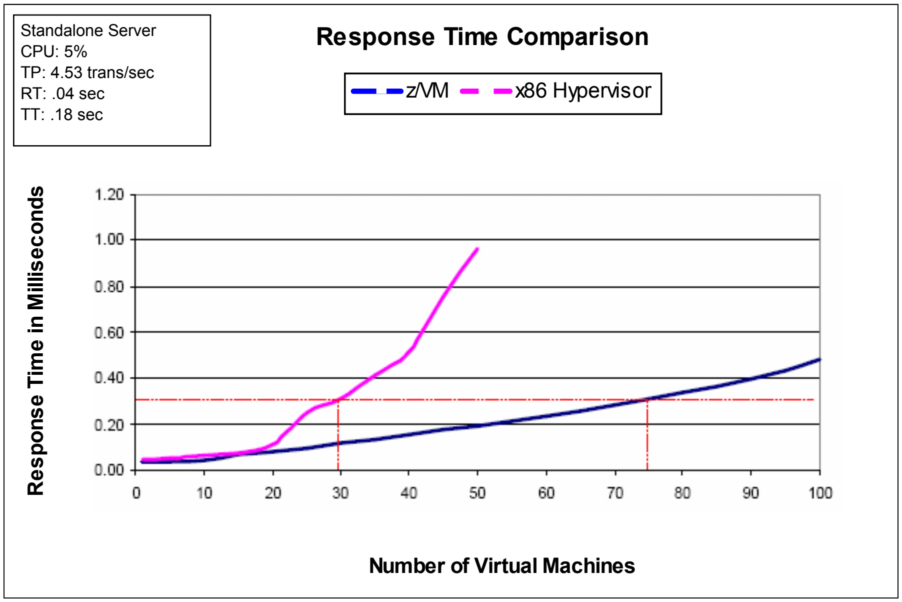


Figure 1-6 Response Time Comparison

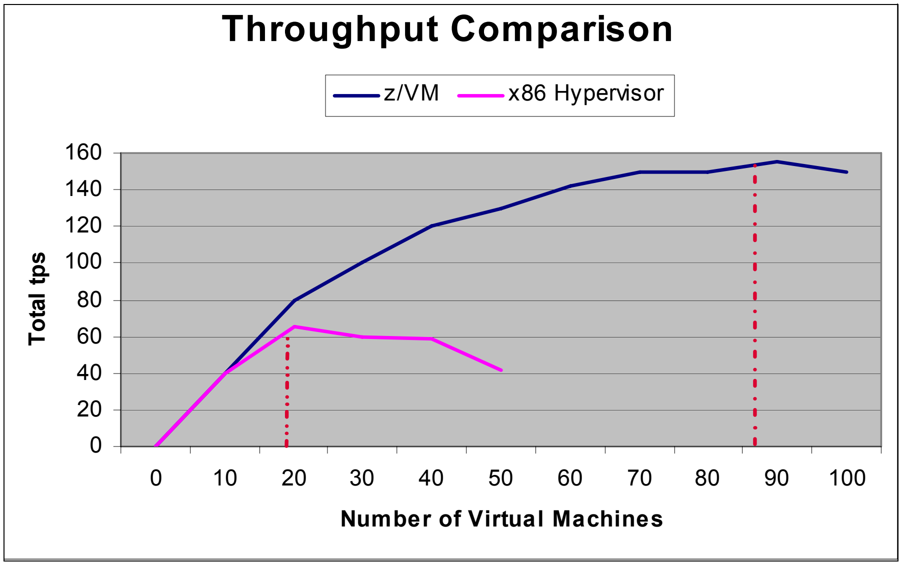


Figure 1-7 25Throughput Comparison

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:

1. Buy stand-alone x86 servers running one image on each
2. Rent an Amazon EC2 instance running one image each
3. Buy large x86 servers and provision virtual servers using VMware
4. 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.

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 1-8 275 Year TCO Comparison

As you can see virtualization on your own equipment proves to be a compelling argument 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 as it can scale to 64 processors of which all can be IFLs. Depending on the workloads this could potentially support up to 900 servers.

The cost per image of a brand 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 have not been discussed here but are discussed in the Whitepaper. 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 some 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 beween 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 o f the migration project, not just when they are needed to perform tasks for you in the execution stages of migration project.

In this chapter, a definition of the stakeholder and their roles and responsibilities will be discussed. Not every organization will use the exact same titles that we list for the stakeholder but the titles that you use should have matching functions that are described here. This list is meant to be comprehensive. Not all stakeholders that we discuss will be involved in your project. And there are usually some versitile and skilled people who t will participate in or be responsible for multiple roles over several functional areas.The size of your organization and the complexity of the migration effort will determine the number and breath of stakeholders involved.

2.1 Stakeholder Definitions

In this section, stakeholders will be categorized as belonging to the comparitively non-techincal business stakeholders or the more technically oriented information technologies stakeholders.

2.1.1 Business Stakeholders:

- Business Owner or Business Manager

  The business manager is the head of a business line such as marketing, sales, finance, etc. His concerns are with the business and perceives information technology as a tool to accomplish buisness tasks effeciently and effectively. He may have a member of his staff reporting to him that deals with technical issues. Proposals for migration may come from him or his staff. These proposals will have to be evaluated by the technology stakeholders. Conversely, proposals for migration may originate with the technology stakeholders and they must provide sufficient justification to the business owner. We have discussed some justifications for migration in Chapter 1, “Introduction” on page 9.

  Large and complex consolidation projects will require participation and buy-in from several business owners and business lines. It is clear that the business owners and IT management must be closely aligned and co-operate openly .

- Business Managers and Supervisors

  These stakeholders have concerns for the workflow within their deparments. They understand the importance of the application and how their employees utilitze it. They will select the users that are the most qualified and motivated to participate in the migration project.

- Users

  The users are the end customers. They use the application or consume the services provided by the application. They are the ones that can do testing and assure that the application is working the same after the successful implementation of the migrated system. In a migration without enhancements, the user should not see any changes. Availability and response times should meet the Service Level Objectives agreed to management and communicated to the users. Their perspecive 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 person'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

  The project manager has the responsiblity 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 of progress against plans, escalating any issues or risks where appropriate.

- IT Managers and Supervisors

  Some stakeholders will be managers and/or supervisors for the front-line staff that we describe next. Managers at this level will have various influences 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. They 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 aquire required skills. These skills can be hardware or software related.

- Mainframe System Administrator/Systems Programmer

  The systems programmer will set up the LPAR(s) for z/VM to run in. (Or a single instance of Linux could run in an LPAR.) The tools used are Hardware Configuration Management (HCM) and the Hardware Configuration Definition (HCD). HCM is the graphical user interface to HCD. With HCD the physical hardware environment is defined.

  The mainframe systems programmer will be responsible for setting up hardware definitions. The hardware components defined are CHPIDS(channels), control units, and devices. A channel is a generic term for extenal I/O communication paths to Open System Adapters (OSA) for ethernet networks, FICON® or Fibre Channel Protocol (FCP) for attached disk, printers, tapes, and consoles. The systems programmer will install and maintain the z/VM ~~operating system~~ including the definition of user directories and resources for CMS users and Linux guests. Also configured are 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

  The Linux Adminstrator may assist in installing Linux on System z, or take over administration tasks once the Linux guest is installed for him. They will work closely with the system programmer when major configuration changes or additions are made, such as the increase in memory, disk space, or CPUs. 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 will partially depend upon where the source sysem is hosted, i.e. the platform where the source application resides. Obviously, the administrator of the source system will be heavily involved because it is the application that is being migrated.

  There will be other services such as DNS, mail servers, security, etc. that are running on Unix or Window servers. These and other services will usually be required by the application that is being migrated. The adminisrtators of these services will be required to make adjustments for the migrated application.

- Network Engineer

  This stakeholder designs, installs, and maintains data communication equipment such as routers, switches, local area networks (LAN), wide area network (WAN) and other network appliances. They monitor the network for performance and errors. The network engineer needs to know the communications components that are unique to Linux on System z. Recommended reading for the Network Engineer to learn more about IBM System z networking can be found in Appendix 6, “Network analysis” on page 63. Other network concepts and tools will be the same for the Network Engineer. The role of the Network Engineer during the migration will be to help design the new network and deploy any changes to the existing network.

- Database Administrator

  The tasks that are performed by a database administrator may be separated into two or more different but related job functions such as a database analyst, database administrator and systems administrator. This person or people are responsible for the installation and maintenance of the database management system (DBMS) code base. They will design and implement the corporate data bases. They will maintain the performance and integrity of the data bases. They 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 due to the similarities of administration, including GUI interfaces and tooling.

- Security Administrator

  The functional area of security has become more visable and critical as company assets are exposed to the internet and available on mobile and wireless devices. The secrity officer and architect are concerned about data protection, including the authentication and authorization of users who access the companies applications. The target application must adhere to existent security policies or demonstate heightened security methods and standards. For more details on Linux on System z security see Chapter 11, “Security analysis” on page 127.

- Application Architect and Developer

  Applications developed in-house will require porting and testing on the target Linux system. The effort involved can vary greatly depending on the 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 applicaions. IBM Global Services, as part of their 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 that will be responsible for this porting effort. See Appendix 8, “Application analysis” on page 91 for more information on issues to be considered.

- Operator

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

- Service Desk

  These people are on the front lines of support to the customer. They are usually the first ones to get a call when there is a real or percieved probelm with the application. They need to be the first staff to get trained on the new environment and be heavily involved in the migration testing in order to be ready to provide support from day one.

- End Users

  Perhaps the most important people involved in a migration are those who will use the application every day. They need to be involved from the start as the success of the project will depend in a large case on how easy the system is to use for them. Ideally, it should have the same look and feel to what they are accustomed to, but in many cases a migration is often an opportunity for firms to ‘improve’ the application and this often results in additional functions and procedures that they will need to learn.

- Vendors

Various vendors will be involved in the migration project. There may be vendors for hardware, middleware, and applications. They have many resources that should be exploited. They are ready to help if you make your needs known to them. They can respond quickly and are often the most cost effective source of information and solutions to problems.

If you are running 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 have skills that your staff does not yet have or skill 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 tranfer takes place for persisent, recurring tasks.

2.2 Assemble the stakeholders

A meeting of stakeholders (or representatives of larger groups of stakeholders) is a good way to set expectations and to perform other planning considerations. A meeting of this group of people will be a good way to identify if there any additional administrator, manager, or user skills enhancements that are needed. These will also be the people to whom status and milestone results are reported. Some of these people may have never met. A cohesive, efficient and successful project requires personal relationships.

In order to make sure that all interests are taken into account, it is a good idea to request a meeting of the key people asking for and affected by the migration. Smaller subsets of stakeholders with related tasks and responsibilities should meet to enhance communications and teamwork.

Communicating the change

Such meetings are good ways to open communication channels. The effect of a good communications plan will be to flatten out the negative aspects of the acceptance curve, as shown in Figure 2-1 on page 35.

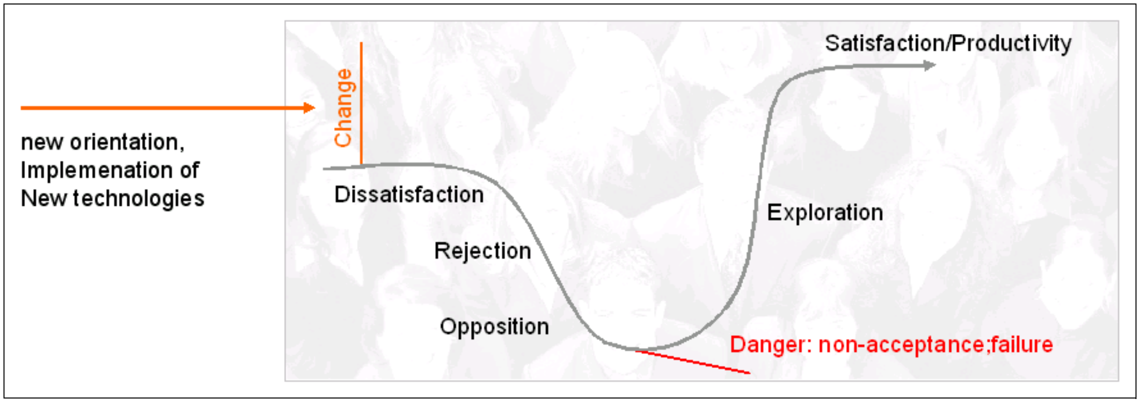


Figure 2-1 Acceptance of new technologies

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

Regarding the IT support team, these same issues have even more importance. A strategic decision to switch the operating system or platform can cause the impression of disapproval of the work they have done so far. It might give staff the impression that their current skills are being devalued. It probably will not be easy to convince an administrator of Solaris™ systems that migrating to Linux is a good 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) up-front during 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 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. Identify metrics for other criterion of success.

3

Migration Methodology

In information technology, migration is 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 some information regarding approaches involved in planning a migration.

3.1 Migration Approach

After the business value and need for moving on to Linux on System z has been accepted by various organizational stakeholders, it's 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. See Figure 3-1 for an approach to planning.

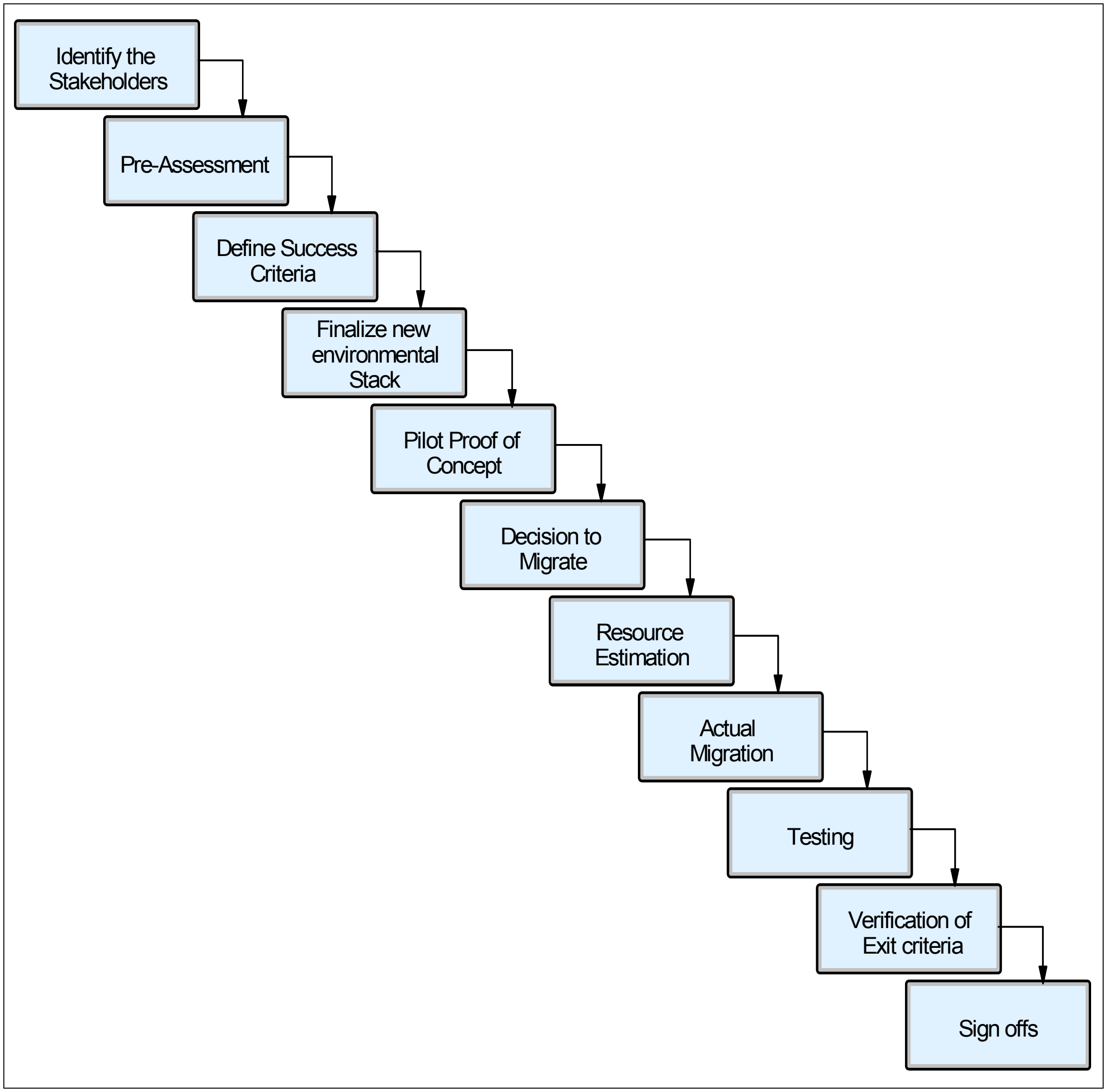


Figure 3-1 Process diagram of planning approach

3.1.1 Identify the Stakeholders

As already discussed in the previous chapter, the first factor is indentifying the stakeholders. Here, the stakeholders would identify the business and operational requirements that impact the migration process. All the stakeholders within the

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

- Business owners define the business and operational success criterias.
- System Administrators would provide information about the application requirements, database requirements and available network bandwidth and CPU load and allowable downtime.
- Security and compliance teams define compliance requirements for the entire migration activity.

3.1.2 Pre-Assessment

During the Pre-Assessment step, a high level analysis and initial feasibility study of the application architecture, source code dependencies, database compatibility and build environment would be performed and would provide a high level scope for the migration to the target operating system. Also, the applications running on current servers are assessed as to on whether or not they are available and certified to run on Linux on System z and an evaluation of the risks related to migration is performed. By doing this we identify the major risk areas at the earliest possible stage.

As a added step, a careful analysis of present and anticipated business needs would also need to 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 Sucess Criteria

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

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

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

We recommend doing 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 we need to list other parameters available in the tool which would provide the same functionality or feature. See Chapter 4, “Migration Planning checklist” on page 45 for examples of forms that can be used to help document the software and hardware requirements.

During this phase most of the technical incompatibilities and difference in the environmental options would be identified.

Custom built applications

If the custom built application is 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 indepth analysis should be carried out on the various build options specified, ensuring 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 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

Now that we have a clear understanding of the target environment and the areas with possible issues and risks, we can move on to a pilot proof of concept phase. This phase is a subset of the actual migration with a reduced scope and duration. Here, we would implement a small module or stand alone code snippet from the application onto the target environment. The proof of concept (PoC) phase will follow all of the same tasks and activities of the full migration. The main objective of the PoC is to focus on the identified areas of risk, empirically test the

recommended approaches and to prove that the full migration can be completed successfully.

In this way, the major potential migration risks identified during the assessment can be addressed in a controlled environment and the optimum solution 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

Once the pilot is complete, we would have a complete analysis on the target operating system environment as well a roadmap detailing the resources, time, and costs required to migrate to Linux on System z.

Analyze and discuss all the key requirements, such as timing, resource needs, and business commitments such as service level agreements (SLAs) with the stakeholders. Also, be sure to include related aspects of the migration, such as new workloads, infrastructure, and consolidation. The decision regarding the migration must be acceptable to all the stakeholders involved in this activity.

3.1.7 Resource Estimation

Understanding the migration objectives and developing metrics with stakeholder involvement and agreement helps to provide a good 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 (SLAs), for each of the stakeholders.

The 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 early involvement of affected teams are good ways to manage change issues. For example, support staff for hardware might be comfortable with UNIX-related hardware support, where to go for help, and so on. The “experts” in the prior environment might be less open to change if they feel threatened by new ways of doing things where they are not the “expert.”

Consider the following areas:

Resources — What hardware will be required? Software? Identify what housing issues are required (are the electrical and cooling inputs equal?). What staff will be required to help with the crossover?

Education
: Does the staff have adequate Linux education? Are there special skills required for the hardware or Linux/hardware combination?

Service level agreements
: While the installation, configuration, and testing of the change is taking place,what are the support mechanisms (both yours and those of any vendors)? What are your commitments to current stakeholders while you are performing the migration?

Related aspects to project
: 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 step is to do the actual migration of the customer applications and the infrastructure to the Linux on System z environment, resulting in an environment that is ready for hand-over 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 that 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 pre-defined set of test objectives is verified and the customer test requirements are validated on the target operational environment. This is an important step before verififcation 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 in mind for validation:

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

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

3.1.10 Verification of Success Criteria

After the successful migration of the environment, re-assess the original acceptance criteria with all the stakeholders and 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 needs to be done. Once this review is complete the testing phase has to 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 basic and generic information and templates needed to quickly assess the source and the target operating environment during the initial planning stages of an migration project.

4.1 What is a Planning Checklist ?

The planning checklist is used to identify requirements (hardware and software), migration tasks and the 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 organization to organization, the basic foundation of an effective planning checklist is probably quite similar to the generic checklists which we will be discussing in this chapter. The checklists are created for the target platform, Linux on System z.

4.2 Hardware Checklist

This template lists various hardware resourcses that need to be taken into consideration during a migration project. In the checklist template, 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. Example 4-1shows a sample hardware planning checklist.

Example 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		file system size		
		LVM Type		*Linear, Striped, not used
		Device Type		
		Number of Devices		
		ECKD model 3 DASD		
		ECKD model 9 DASD		
		ECKD model 27 DASD		
		zFCP device		
		Size of disk in GB		
		M-DISK		
		Size of disk in Cly		
Name		**Name**		
mount point		mount point		
size		file system size		
		LVM Type 3		*Linear, Striped, not used
		Device Type		
		Number of Devices		
		ECKD model 3 DASD		
		ECKD model 9 DASD		
		ECKD model 27 DASD		
		zFCP device		
		Size of disk in GB		
		M-DISK		
		Size of disk in Cly		
Name		Name		
mount point		mount point		
size		size		

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

[2] Custom File System : This area should be copied to attendent the number of file system that exist at source server or sould exist at destination server .

4.3 Product Checklist

In the software checklist template (see Table 4-1), we list all the products and tools that are used in the source operating environment and then chart out whether the same or similar products and tools are available on the target Linux on System z operating environment.

Table 4-1 Product and Tools Checklist

Application Name : **Description :**					
Product / Tool	**Version**	**Vendor / Source Website**	**Used Environment (Dev/Test/Prod)**	**Licensed(L)/Freeware(F)**	**Linux on System z (Availability)**

4.3.1 Application Implementation Checklist

The Application Implementation Checklist takes us one step further into the product checklist, where each product or tool is further drilled down to their features level. There are scenarios where the same product would not offer the same features on all the platforms. These details would be noted in this check list (see Table 4-2).

Table 4-2 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		

As in Table 4-2, “Application Implementation checklist” on page 49 each product or tool listed in the product checklist must be analyzed. All the parameters, dependencies and optimization options have to taken into account in the source

operating environment and then the planning team needs to be 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 similar feature by linking other product or tools in the target operating environment.

– Make note of the other paramaters that are available in the same tool of the target operating environment which can be combined to give the same characteristics as in the source environment.

– If the products or product options are fully incompatible or not available , replacing the part of the application stack would also be a good approach, which would minimize the effort involved in migration. But care has to be taken so 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.

– Most of the times optimization feaures or performance options for a product would be only available for that specific platform. In this case the optimization and performance options need to changed to offer the same charectertics in the target environment.

While filling out this application implementation checklist, we 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, the tools, products and their implementation differences would be accounted for in the actual migration. This would in turn ruduce the risks and the migration can be executed smoothly.

4.3.2 Training Checklist

One of the most crucial elements for a successful migration is to ensure 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 section 2.1.2, “Information Technology Stakeholders:” on page 31 for ideas on staff that may need additional training.

4.3.3 Application Environment Checklist

The source application that is to be migrated could be in the center of a very complex process. The source application could be inter-connected with many other applications, inputs, outputs and interfaces. Prepare a planning document that lists the resources that the source application needs to provide and all the services that it is currently providing. We have provided some examples of the resouces that are required of some applications in Table 4-3.

Make the descriptions as detailed as possible, provide physical location, hostname, IP address, network information, software product used, service owner, etc. Actions required for the migration effort could also be documented.

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 make sure that assumptions made by one group of stakeholders are not different from the assumptions made by other stakeholders.

Table 4-3 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 hostname		
DNS Services	Single or Multiple DNS Servers		
Firewalls	Firewall Locations and Functions		
Internet Connectivity	Router Location		
Intranet Connectivity	Webserver location and ports		

Source Resource	Source Location	Target Resource	Target Location
Email 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 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

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, and
- Networks

5.1 Virtualization Concepts

A standard feature of IBM System z is Processor Resource/System Manager (PR/SM) which 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 and also an associated weighting which defines its relative priority against other LPARs. This allows us 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 the physical resources. Depending upon the LPAR's configuration, the real resources avaliable could be multiplied as virtual resources to be used by the guests . At the z/VM level we can also have as guest operating systems like z/OS , Linux on System z , z/VSE, CMS and also 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.

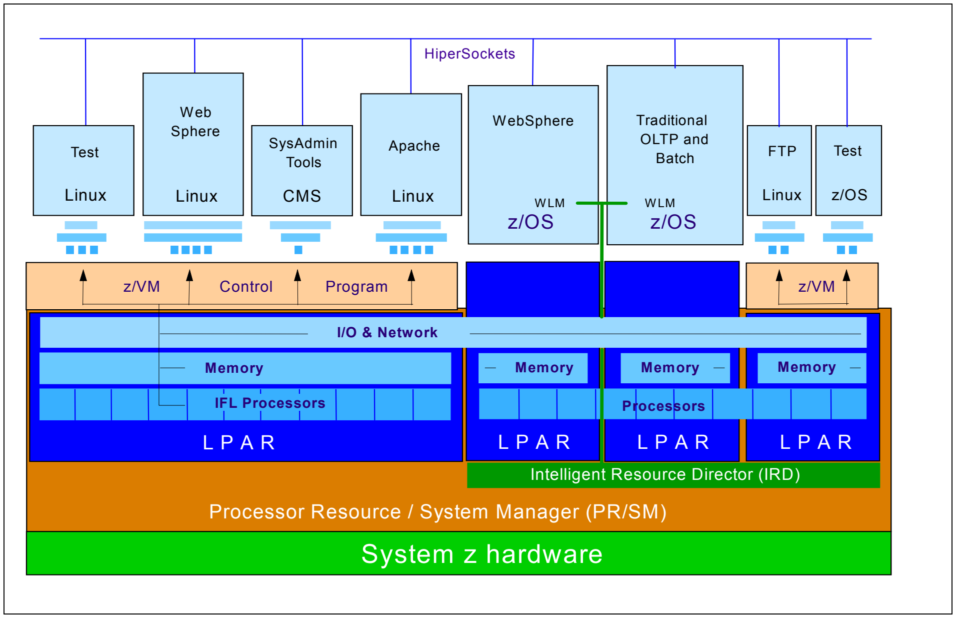


Figure 5-1 Virtualization environment

5.1.1 z/VM Operating System components

This section outlines the z/VM operating system components.

CP (Control Program)

The CP provides an end-user, in our specific case, the Linux operating system, a complete System z environment in 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, Linux adminsitradors for management, query and to 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 on 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 dispatched. Communication to the Linux system can be either via the Linux virtual machine console which must be a 3270 type terminal emulator or more commonly via a Telnet and SSH client terminal. Note that if an administrator logs off the Linux virtual console, this will cause the virtual machine to shutdown and terminate all running work. The administrator must use the **DISCONNECT** and not **LOGOFF** to ensure that this does not happen.

See the Appendix , “3270 Emulation” on page 250 for more information about 3270 session.

CMS (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 sytem components and 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 etc..

For more information about z/VM read:

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 where one is defined for Linux development and test and a second LPAR is defined for Linux production.

When working in a virtualized enviroment it should always be kept in mind that a performance problem that affects one guest in a z/VM host could affect the performance off all the guests in the same LPAR. The use of at least two different LPARs, configured correctly, could avoid problems in some common procedures such as application and operating system updates and upgrades and at the time of new application deployment, test and evaluation.

As it is necessary to assign logical CPUs to each LPAR, the number of the 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 over-allocate memory on the Linux guests than is avaliable to the LPAR.

5.1.3 CPU

The number of IFLs on the machine will reflect directly at 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 4 IFLs do not allocate 5 virtual CPUs to a single Linux guest machine. If any situation occurs where the Linux guest uses all 100% of the CPUs, that will adversely affect the entire LPAR.

However in an LPAR with four IFLs, you can assign 3 Virtual CPUs for LinuxA guest and another 2 virtual CPUs to LinuxB guest and also another 2 virtual CPUs to LinuxC guest . All request for CPU cycles will be managed by z/VM according to the Linux guests relative priorities.

The best practices of CPU configuration is to maintain the number of active virtual CPUs ratio of 4 to 1 of logical CPUs allocated to the LPAR .

5.1.4 Memory

System memory (Linux term) or storage (z/VM term) is a resource that is shared across all z/VM guests. Each virtual guest is asigned 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 one time and the amount of real memory (storage) that is allocated to the z/VM LPAR. z/VM does allow you to over-commit memory but the general recommendation for production workloads is to keep the over committment ratio of total amount of vitual memory likely to be active to total amount of vitual memory to around 2:1. For test or development workloads, the ratio should be no more than 3:1.

The key to determining the ‘right’ virtual memory size is to understand the working set for each virtual machine and also ensure that the Linux images do not have any unneeded processes installed. Other recommendations are to use VDISKs for Swap. This is described in “SWAP device consideration” on page 61.

Memory Management Features

There are some memory management features for Linux and z/VM that can be used to reduce the amount of memory required by virtual guests. These are:

- Cooperative Memory Management (CMM),
- Collaborative Memory Management Assist (CMMA),
- Named Saved Segments (NSS)
- Discontiguous Saved Segments (DCSS).

CMM is used to reduce double paging that may happen between Linux and the control program(CP). CMM requires IBM’s 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 is loaded with the **`modprobe`** command.

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

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

DCSS allow virtual machines to share re-entrant code for applications, such as Oracle, which also reduces overall storage requirements. In Figure 5-2 on page 60 we show how both NSS & 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.

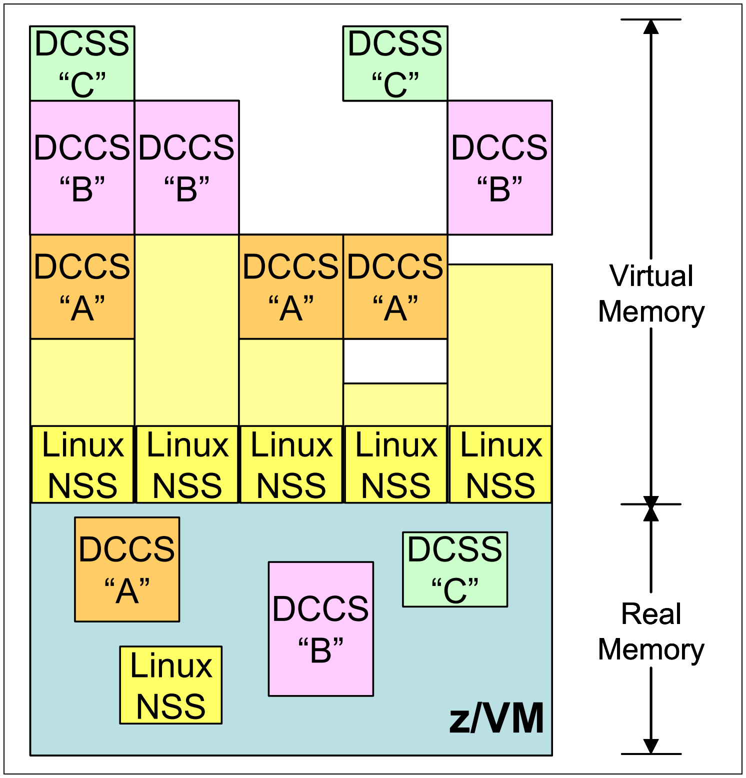


Figure 5-2 DCCS & NSS shared by multiple Linux virtual guest

> **Note:** For more information on setting up a Discontiguous Saved Segment and using Execute-In-Place (XIP) filesystem see:
>
> *Using Discontiguous Shared Segments and XIP2 Filesystems With Oracle Database 10g on Linux for IBM System z*, SG24-7285

One thing to be aware of when defining memory requirements for virtual Linux guests is that the *Linux kernel* will use all of the extra available memory allocated to it as a file system cache. While this makes sense on a standalone 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 can not get enought memory resource. Thus when paging occurs, this is an indication that more memory needs to either be assigned or the application 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 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.

> **Note:** For more information on optimizing memory on z/VM and Linux see:
>
> *Linux on IBM System z: Performance Measurement and Tuning*, SG24-6926

5.1.5 Disk

System z disk storage is commonly referred to as DASD (Direct Access Storage Device). 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, the disk device support is expanded to Fixed Block Architecture (FBA) DASD and also 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 measured in cylinders and megabytes is listed inTable 5-1.

Table 5-1 DASD table

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

The connection to SCSI devices is managed by the zFCP Linux module driver and the SCSI devices are usually dedicated to the Linux guests. For a more thorough discussion on Storage please read “Storage analysis” on page 77.

> **Note:** For more information on z/VM and disks, read Chapter 1 of:
>
> *Introduction to the New Mainframe: z/VM Basics*, SG24-7316

5.1.6 Network

For a Linux on System z environment, the physical network attach device is the Open System Adapter Express 2 or OSA Express 3. This is capable of handling up to 640 TCP/IP stacks simultaneuosly including Hipersockets for inter LPAR communcation. An IBM System z10 EC can support 48 OSA-Express2 or 96 OSA-Express3 ports. The Open System Adapter supports both copper and fibre Ethernet connections at speeds of up to 10Gb.

The z/VM operating sytem feature to access the TCP/IP network is TCP/IP for z/VM. The OSA-Express2 can be virtualized through a virtual switch (VSWITCH) device to many Linux guests and is avaliable using special z/VM machines called VSWITCH controllers . Each Linux guest connects using a virtual devices controlled by the **qeth** module to a virtual switch system in a z/VM LPAR.

One of the most important benefits of the VSWITCH system is that it can be setup with redundant OSA devices that provide a failover network system on z/VM.

HiperSockets provide a high-speed connectivity between servers running on a System z. This technology doesn’ t require any special physical device configurations or cabling.

Both OSA-Express and Hipersockets use the QDIO (Queue Direct I/O) 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 avaliable in Chapter 6, “Network analysis” on page 63.

> **Note:** For more information about network in Linux and z/VM, see Chapter 4 in *Linux for IBM System z9 and IBM zSeries*, SG24-6694

6

Network analysis

This chapter provides information about network migration configuration issues and 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 10km, 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 4 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 as z/VM provides a complete virtual network system that includes the possiblilty 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 distribuited machines from the same physical subnet to a single System z LPAR atached to the same network segment. See Figure 6-1 for an example depicting a single distributed network.

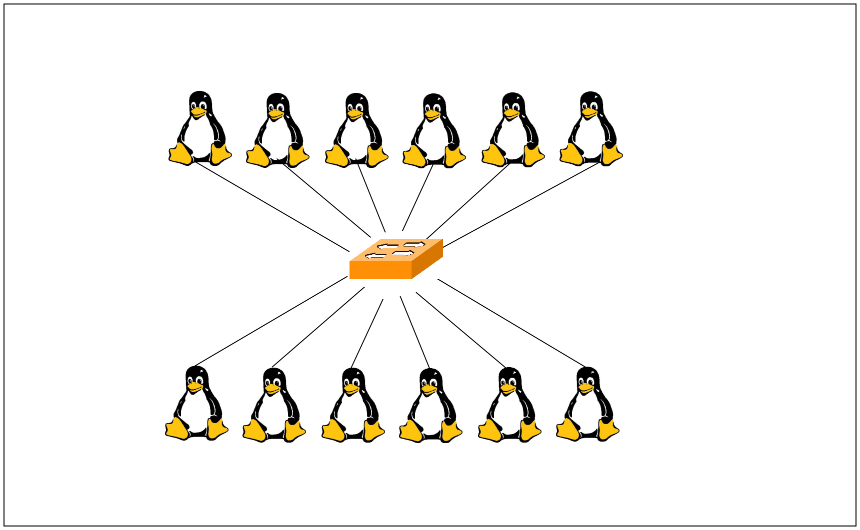


Figure 6-1 Single Distributed Network

Within this scenario, all physical machines can be migrated to a single System z box 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 describes this type of configuration .

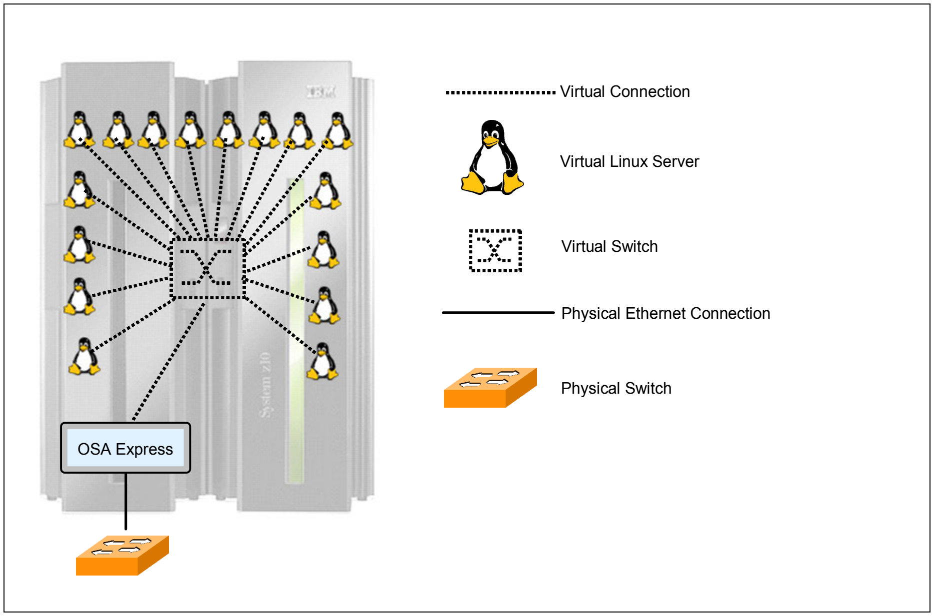


Figure 6-2 Single Virtualized Network

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 on the network . This will provide a network failover capability for the entire system as shown in the Figure 6-3.

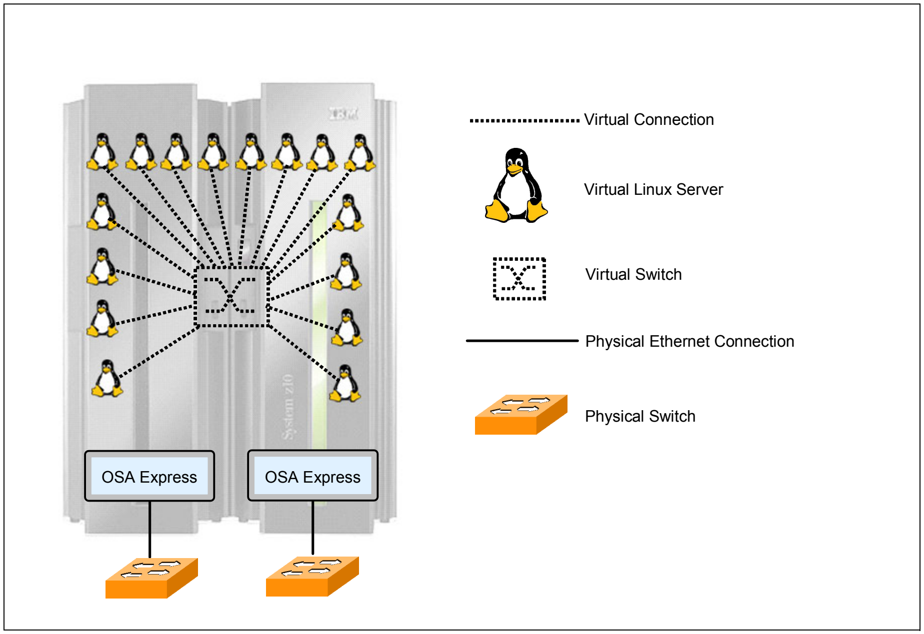


Figure 6-3 Single Virtualized network with failover solution

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.

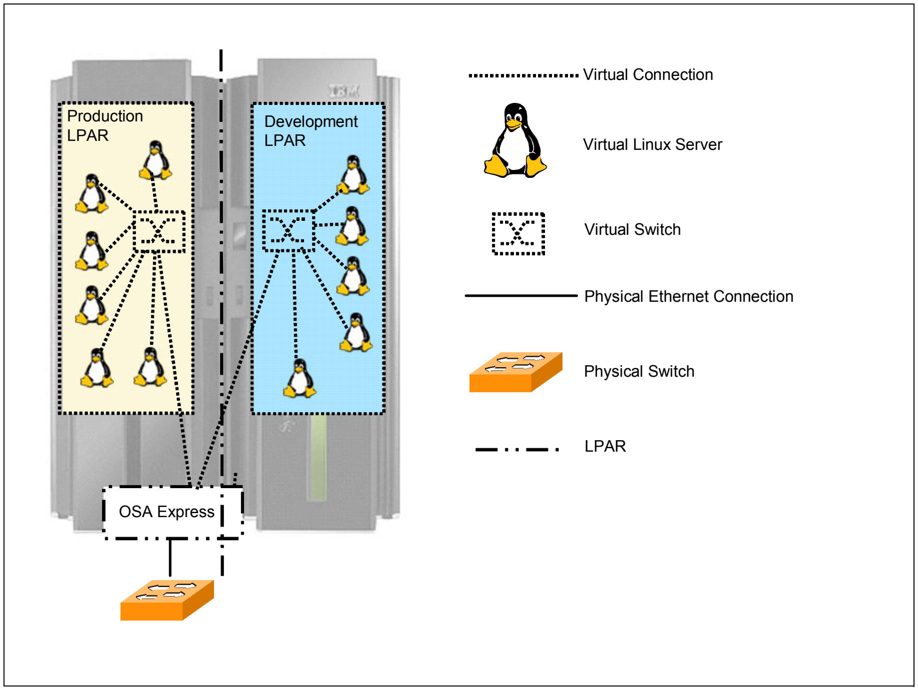


Figure 6-4 Single virtualized network with multiples LPARs

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.

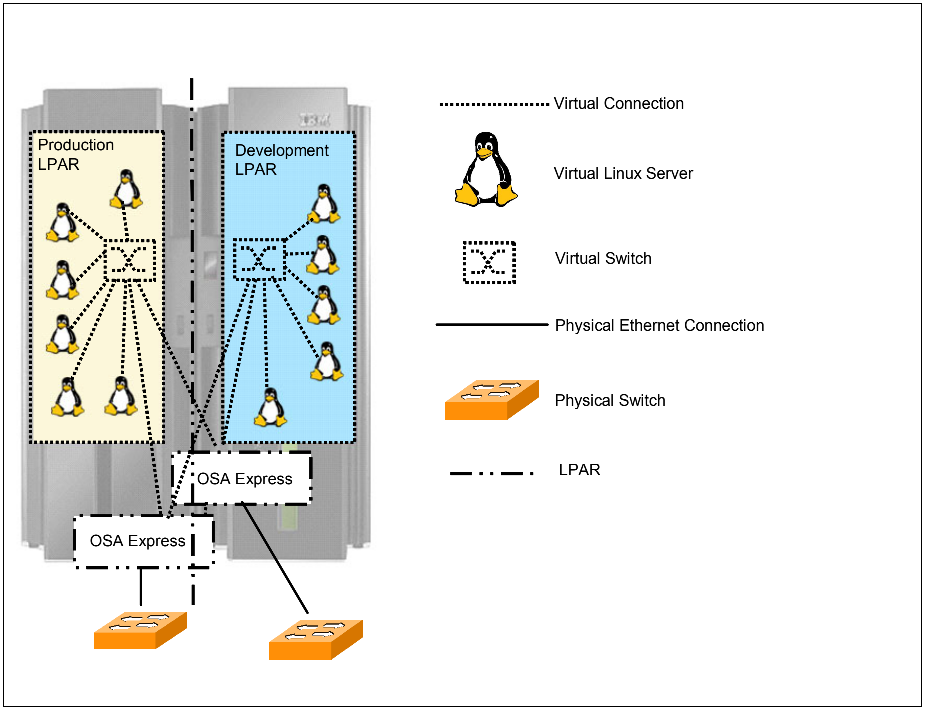


Figure 6-5 Single virtualized network with multiples LPARs with failover

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. Take note that we are analyzing the DMZ and a secure network scenario . In our first example in Figure 6-6, we show that we have a DMZ network where our Web Application Server is placed and a secure network where our database server is located.

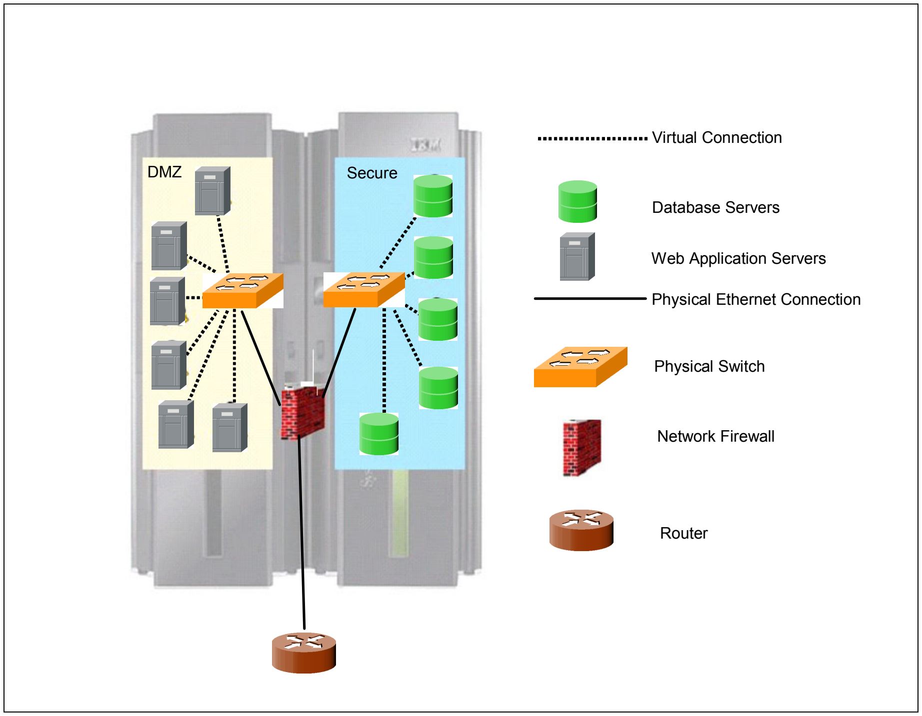


Figure 6-6 Multiple Distributed Network Scenario - DMZ segmented network

In Figure 6-7 and the Figure 6-8, we show the network architecture used in a Linux on System z scenario. 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.

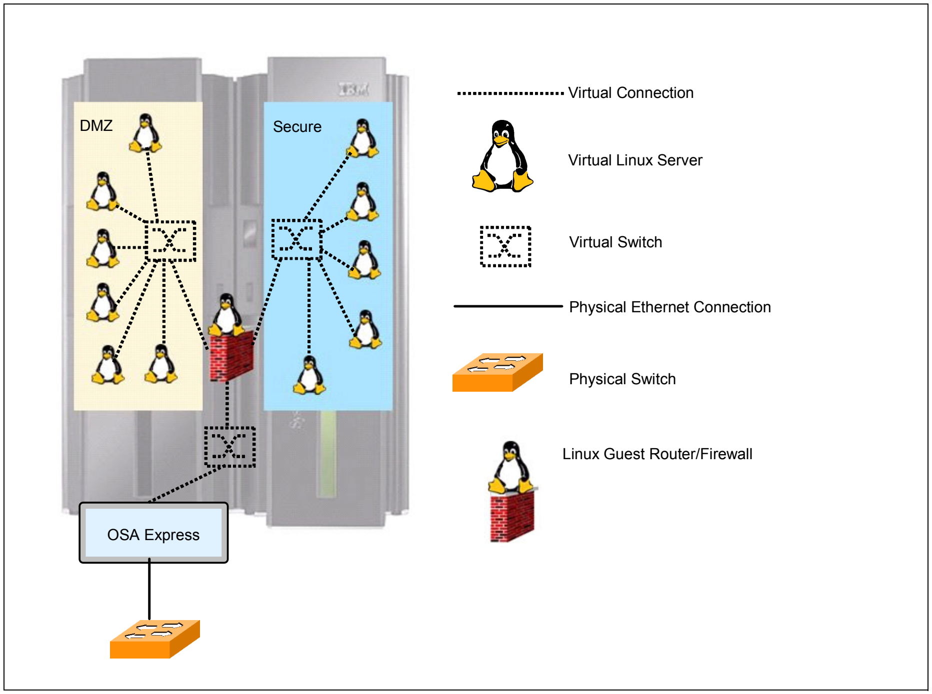


Figure 6-7 Multiple Virtualized Network Scenario - DMZ and Secure network

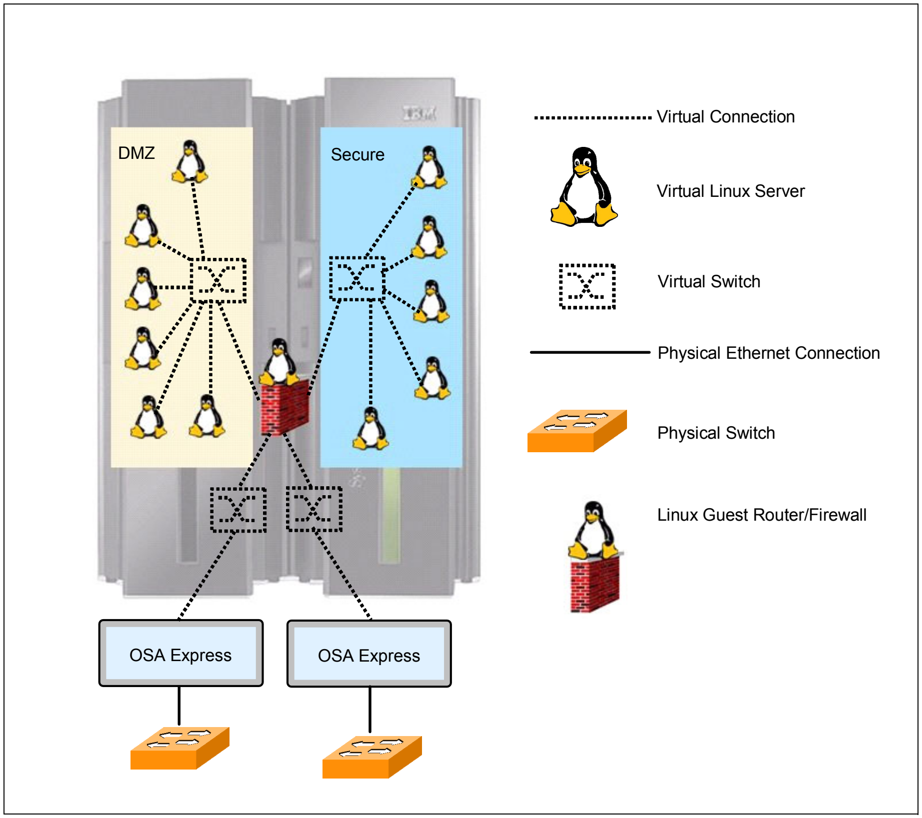


Figure 6-8 Multiple Virtualized Network Scenario with failover- DMZ and Secure network

In a second scenario that uses multiple logical partitions, we 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 a 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). See Figure 6-9 as an example.

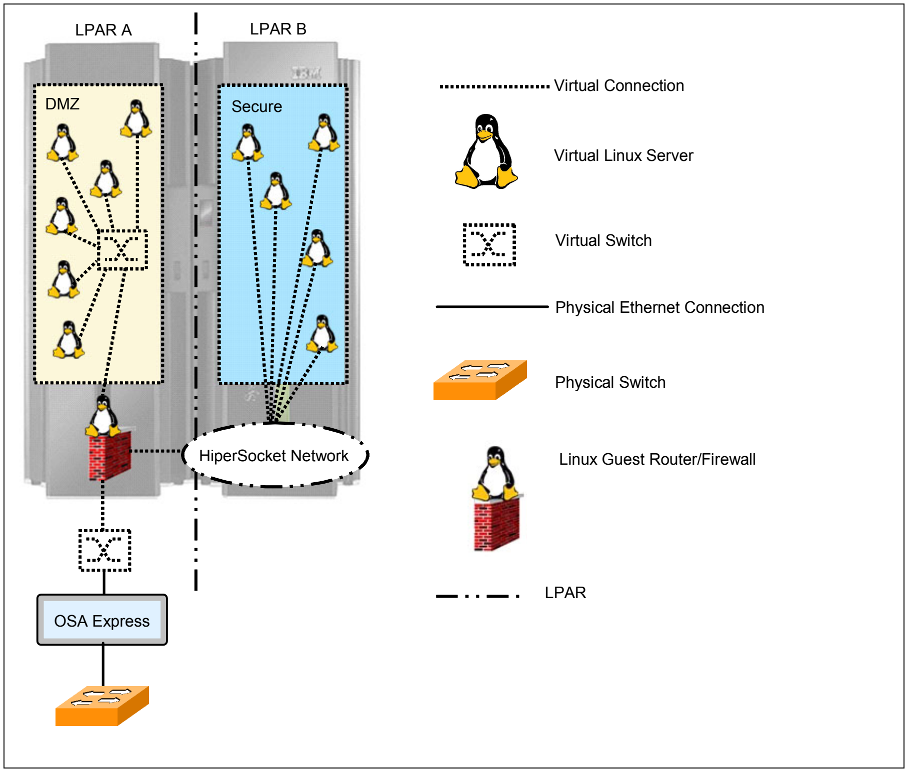


Figure 6-9 Multiple Virtualized Network Scenario with multiples LPARs

The use of hipersockets for this scenario is possible but that may not be the recomended solution. If one of the LPARs are CPU constrained, then 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 to have an external firewall to manage the network filters . The example shown in Figure 6-10 demonstrates the solution that would be described as a network segmented LPAR .

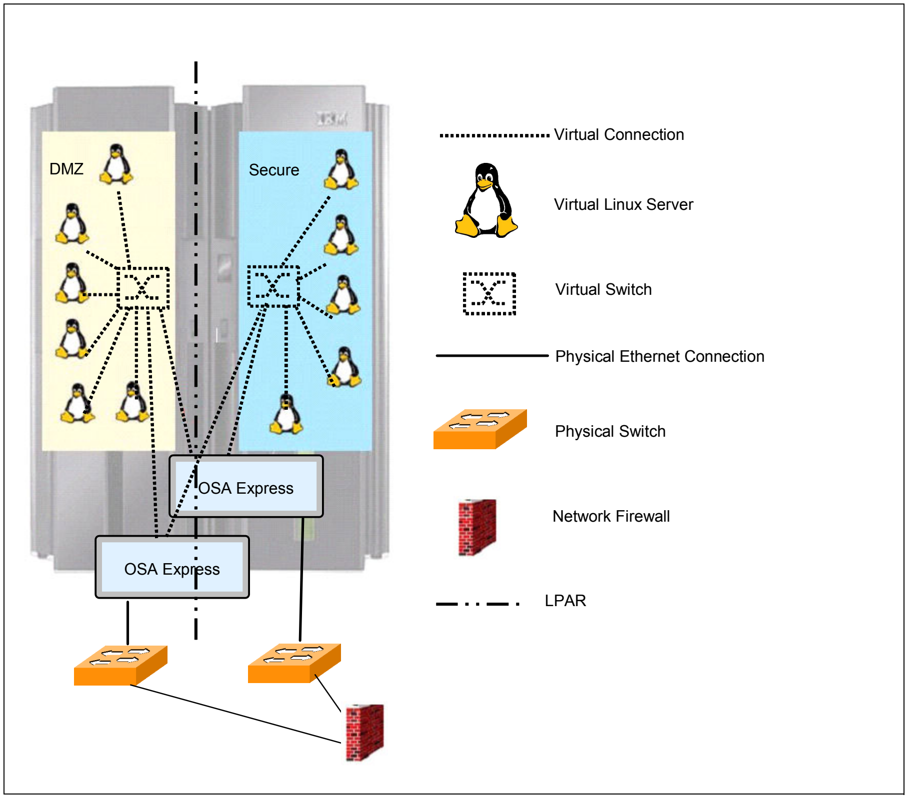


Figure 6-10 Multiple Virtualized Network Scenario with external Firewall

VLAN segmented Network

In the high security network scenario the solution is to use the LPAR enviroment mixed with the multiple network segmented solution. That solution includes the VLAN configuration to help in the segmentation of network packages.

In a distribuited enviroment multiple switches can share the same VLAN and because of switch port limitations, that could be a painful and complex solution. The Figure 6-11 illustrates this configuration enviroment.

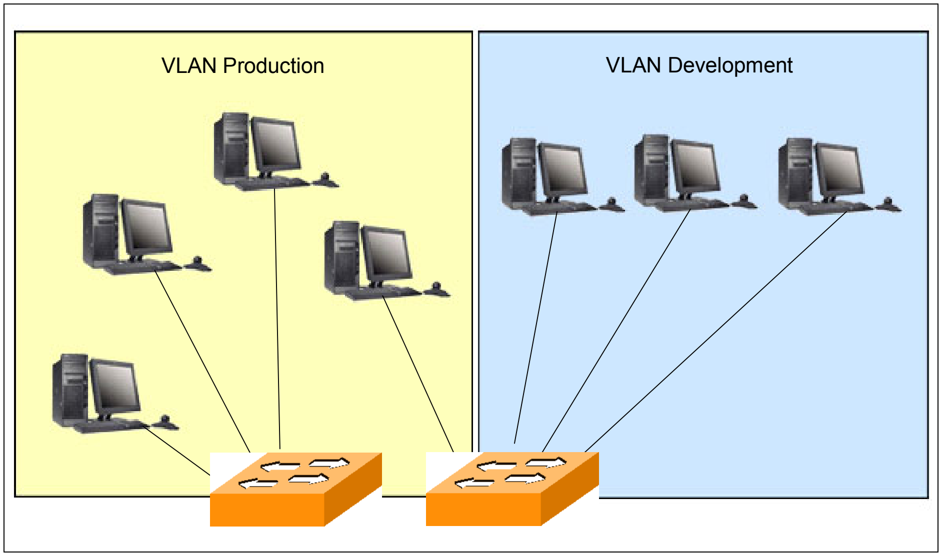


Figure 6-11 Distribuite Enviroment with multiples VLAN

As displayed in the Figure 6-12 the entire System z enviroment is virtualized and all configurations are made per virtual machine, which increases the security and reduces the complexity.

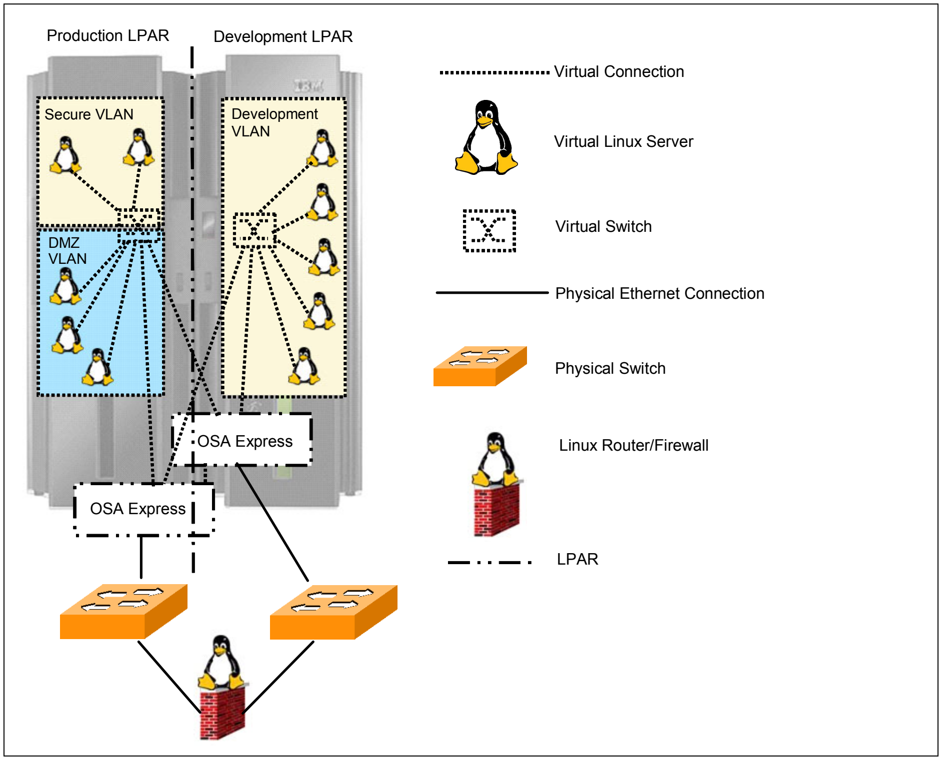


Figure 6-12 Multiple Virtualized Network Scenario with external Firewall and segmented VLANs

6.2 Steps for a sucessfull 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 basics steps :

- 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.

- Determine the VLAN IDs to Linux on System z servers.
- Configure the VSWITCH with listed VLAN IDs.
- Configure the Linux servers QDIO devices within the designated IP address

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

- Migrate the applications and files from the source server to the destination server.
- Shutdown the source servers.
- Change the Linux on System z server's hostname.
- Change 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 servers to the source IP address.

7

Storage analysis

This intent of this chapter is to explain to the reader some concepts and designs regarding the storage configuration possibilities for Linux on System z. Other storage migration issues will also be covered.

7.1 Data migration

There are two models of data migration that will be considered in this chapter. They are referred to as an online migration and an offline migration. The online migration consists of the case where the source server, target servers, and all services are up and running and a system outage is not required. The offline migration requires a service outage to switch over from the source server to the target servers.

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 also data loss. To prevent these kinds of problems, the data migration must always be executed during off-hours and always, have a data backup point just before the actual data migration activity begins.

7.1.1 Online Data Migration

There are some applications that are eligible for online migration. Basically, the application must provide multi-operating system clustering support and be avaliable on Linux on system z.

The steps for this kind of migration are :

- Install and configure the target Linux on system z server (refer to "Points to consider before an install of Linux on System z" on page 83 in this chapter for more details).
- Install the middleware application on the Linux on System z server .
- 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 migration where the application doesn' t 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.
- Include the Linux on System z server in a cluster as a cluster node .
- 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, see the chapter "Application analysis" on page 91 for more details.

- Shutdown 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 with several different approaches and functionality including :

- Using the network mount points NFS or SAMBA connections and the Linux command **CP** to copy the files over network.
- Using an FTP server on the source or target server.
- Using an SCP/SSH server between server and 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).

Rsync Application

For a better result using the `rsync` service, schedule service synchronization for an entire week before the outage. This can be accomplished by following these steps:

- On the first migration day, execute the first syncronization.

  This first syncronization should be issued in a period where the use of the server is low, the **rsync** application only copies files that are not locked, so there aren't any problems with files in use. During this time, the server can respond more slowly than normal because of the extra read I/O activity.

- During the migration week, a daily synchronizaiton can be executed at the server during off peak hours .

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

- The last synchronization day is the server outage day when access to the source server is denied to end-users.

  Since there are no open files, the **rsync** application will be able to copy all files to the target servers.

- The source servers are shutdown and all services are started on the target Linux on System z servers .

Transfer files over the network

Database migrations are the most common example of the requirement for files to be transfered 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 .See the chapter "Database analysis" on page 105 for more details.

Storage Volume Migration

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 (SVC)

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 81 shows where the SVC sits in the Storage Area Network and, as you can see, it has visibility to all supported storage on the SAN.

The benefits provided by the SVC are:

- Single point of control for heterogeneous storage resources
- Dynamic data migration between heterogeneous storage devices on a SAN
- The 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 non-disruptively migrate data to Linux on System z.

Note: For more information on the IBM System Storage SAN Volume Controller see:

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

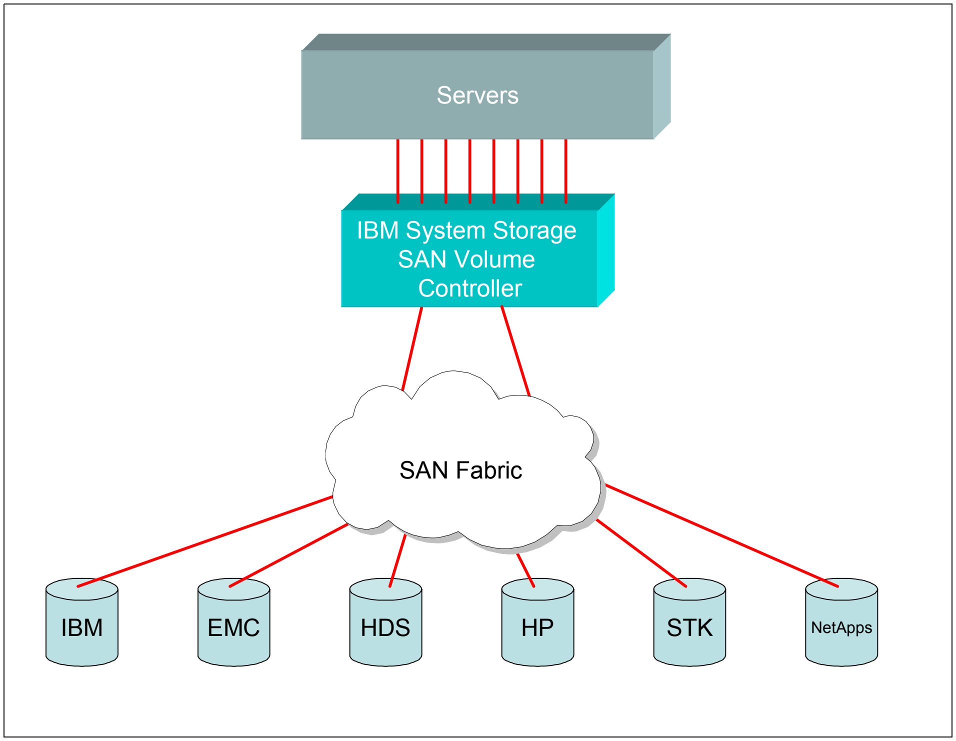


Figure 7-1 SAN Volume Controller

Note: There are also a number of Redbooks that have been written on 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 sucessful offline Storage Migration

The multiple possibilities provided by Linux on System Z to store and access files leads to many types of solutions. The solution you architect for the target system will dramaticlly 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).

- Create the new server file system with mount points for all data files (see the chapter “Points to consider before an install of Linux on System z” for more details).
- Create a temporary file system to be used in the file transfer process on the target server.
- Configure the network on the target server (“Steps for a sucessfull network migration” on page 75) .
- 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.
- 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 transfered using an FTP server configured on the target server.
- If there isn’t 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.
- 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 Points to consider before an install of Linux on System z

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

The best practices for Linux on System z determine that it only one version of a Linux OS distribution should be installed from scratch, so based on that, the basic Linux OS filesystem should be designed to allow the highest number of servers as possible. That is because all other Linux guests in the enviroment should be cloned from this source which is called the *golden image*. The file system that stores the application data is created after the cloning process.

> **Note:** For information on setting up a golden image for Red Hat Enterprise Linux read:
>
> *z/VM and Linux on IBM System z*, SG24-7492
>
> and for SUSE Linux read:
>
> *z/VM and Linux on IBM System z The Virtualization Cookbook for SLES 10 SP2*, SG24-7493

All file systems except the root (/) filesystem should be created as LVM devices. Filesystems 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 physicaly 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 this 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 filesystems:

- root (/) file system

- /usr file system
- /var file system
- /tmp file system
- /opt file system
- /home file system

The root (/) file system

This 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 must be formatted using the EXT2 file system format .

The /usr file system

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

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

This file system should be created on LVM devices (section 7.2.4, “Logical Volume Manager Devices”) that allows you to dynamically extend or reduce the file system size. The EXT3 file system is the best option here .

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

The /var file system

The /var file system is the system are where the all variables files such as spool files, cache files, and log files are written. This 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 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, 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.

As a dynamic file system it should be placed on an LVM device to allow it to be extended or reduced as needed . And the best option is to also use the EXT3 file system.

The /tmp file system

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

As a dynamic file system it should be placed on LVM devices to allow the capability to be extented or reduced as needed . Once again, the best option is to use the EXT3 file system .

The /opt file system

The /opt file system is where all third party applications should be deployed . Best practices teaches us that the /opt directory 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 and the IBM Websphere Application Server should be place 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. You should note that the /opt file system can also be a dynamic file system and it should be configured on an LVM device using the EXT3 file system .

The /home file system

The /home file system is design to allocate users 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 don't need a big /home file system because it isn't expected that development staff will store files on a

production server . On the other hand, on a developement application server it is expected that applications are developed there. Therefore, the developers will need file system space to create and transfer their files.

Depending upon the situation, the /home file system could be a dynamic file system, and if so, 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. Basiclly it is necessary to have at least one file system for data files and one for log files. So at least two file systems should be created plus 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 are in /var/lib/mysql directory, if the server is a MySql database server and you are using the Linux distribution from RedHat Linux or Novell Suse Linux please consider including a new file system at the /var/lib/mysql mount point. For each target database mangement server, make sure that you know where the binary files and the data files will be located. Only then can you plan to create the devices and filesystems for the target system.

It is possible that there are some 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, but 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, althought z/VM allows it at the hardware level.

In a shared DASD enviroment it must be remembered that the file system changes performed by the guest machine that has the read-write control will only be avaliable to all other guests that share the same file system after unmount and mount of the file system .

As an example, think of the enviroment of a web cluster service where the application servers only need read access to the web pages and don't need to write to the same file system where the files are allocated .

In the example in Figure 7-2, only the special file system and mount points relevant to the solution are represented. The data files 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 filesystem /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 .

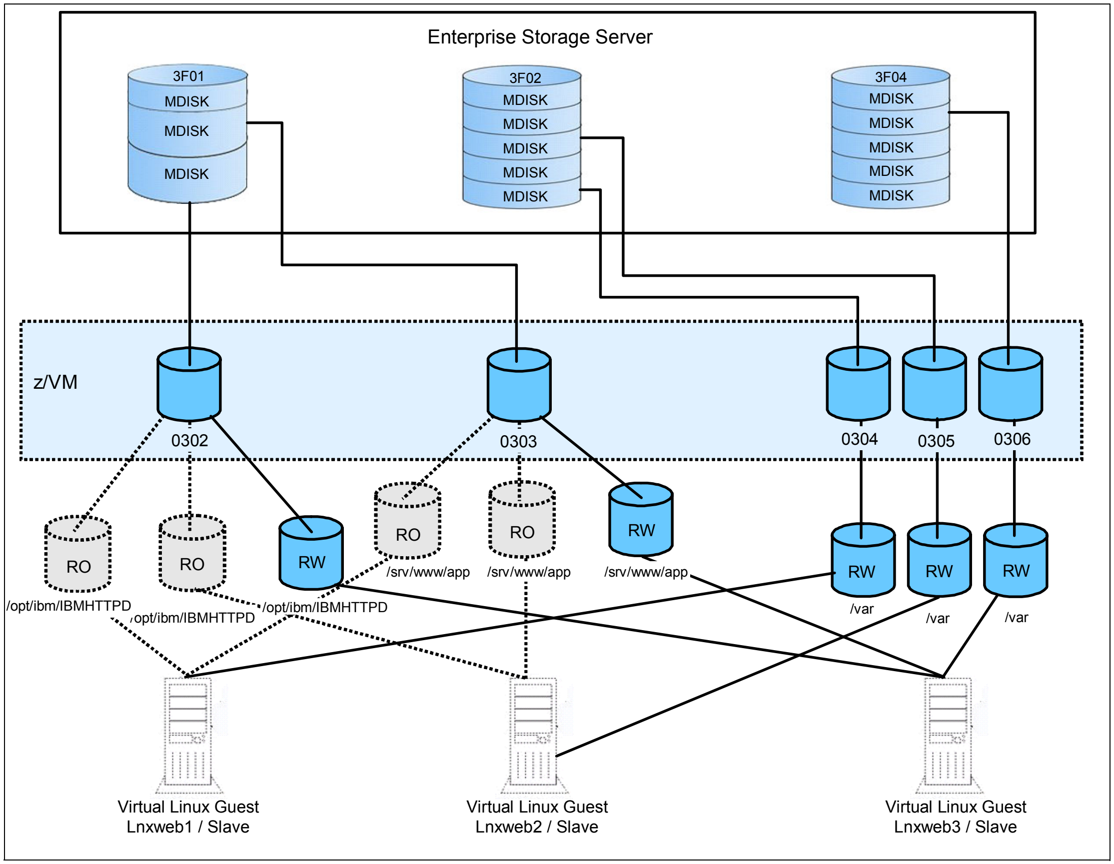


Figure 7-2 Shared Device for Webservers

The benefits of using a shared file sytem is based on economy of resources. You reduce application binaries space allocation and code updating efforts since you only have to update one master server and just remount it on the slave servers.

> **Important:** The system administrator must pay special attention to managing this kind of enviroment 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 you are using big 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. One of the limitations of zFCP devices is that the device size that only the guest OS can use is the size that was provide by the storage administrator. Once it is configured on Linux on System z, it is possible to split it into partitions just 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 with system performance and making the best use of storage resources.

7.2.4 Logical Volume Manager Devices

The Logical Volume Manager (LVM) is one of the most useful tools for Linux files system because its allows for dynamic management of the file system size and there are good tools to backup and restore the failing partitions.

Basicly LVM Volumes are composed of:

- Physical Volumes (PV): These are the storage devices that could 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 Groups (VG): That is the highest level of the LVM unit . It is created by one or more physical volumes and gathers together the Logical Volumes.
- Logical Volumes (LV): This is the disk partition of the LVM system. This is the area that is formatted and are accessed by users and applications. The logical volume is exposed through a mount point .

See the example in Figure 7-3. This shows five MDISK devices that are used by a Linux guest to create a unique Volume group. It is then further organized or allocated into into two Logical Volumes.

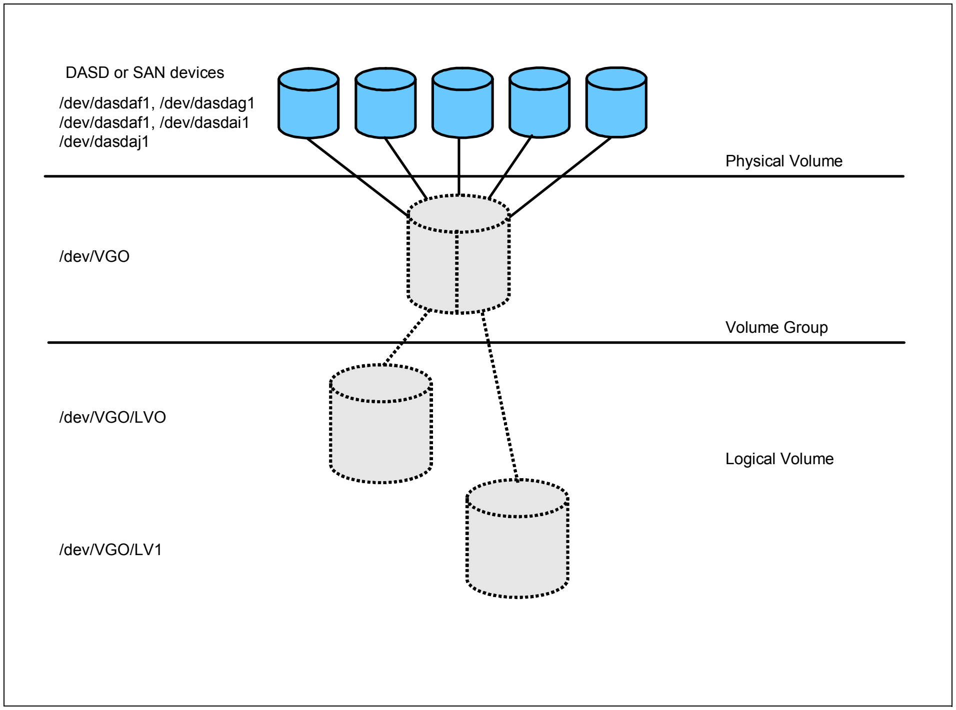


Figure 7-3 A simple LVM example

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 Logical Volume Manager is configured with striping.

Striped Logical Volume

The striped logical volume is a feature on LVM services where the number of blocks on a read/write request has the same number of the stripes that is set when the Logical Volume partition is created .

The simple practice when creating a striped logical volume is to define the same size for each physical device and use the same number of physical devices as number of stripes .

For the example in Figure 7-4, the file system needs 27Gb of space so we create three devices of 9GB 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 address for the different devices. This could provide a dramatic performance improvement for an application that is migrated to Linux on System z.

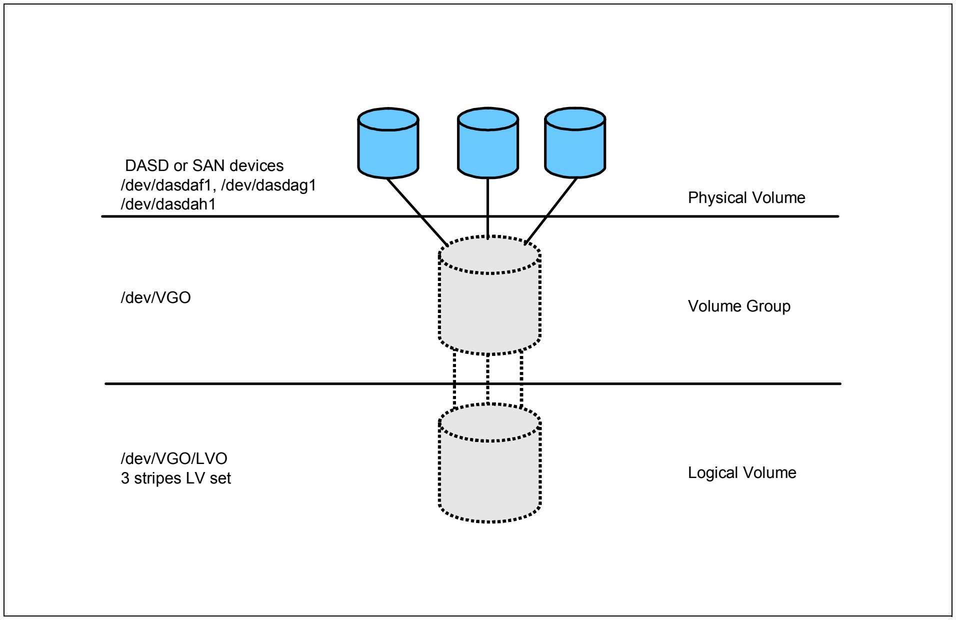


Figure 7-4 LVM Striped example

8

Application analysis

This chapter describes the analysis necessary to identify applications that would be good candidates for migration to Linux on System z.

We will cover 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 doesn't support Linux on System z
- How you can accomodate 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 migrate applications?

As discussed in Chapter 1, “Introduction” on page 9, an application migration is not something that is done without a great deal of planning. There also has to be a compelling reason to act. The following are a number of real world compelling reasons to act:

- An existing application is out growing its original platform and is close to reaching the architectural limits of the platform.
- Software licence costs that are rapidly growing as more and more servers are added to an application.
- Performance issues between distributed application servers and centralised data bases.
- Out of control distributed server growth leading to power and cooling issues in the data center.
- Complex distributed systems that are costly to maintain and are suffering from increasing unreliability.
- New application development following a merger or acquisition.
- Regulatory requirements that require a more secure environment.

The above situations are all good reasons to consider a migration to a more efficient platform like the IBM System z. In most cases a migration to Linux on System z will save an organization significant sums of money over a 3 to 5 year term. The question that needs to be answered is what applications can you migrate and what risk factors are associated with the migration?

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

8.2 What applications can be migrated

Like any computing platform, the IBM System z has its strengths and weaknesses. The aim of a migration should be to select an application that takes advantages of the strengths of the target platform. For System z, its classic strengths are 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 or not the application is supported on Linux on System z. This normally isn't a problem with home grown applications, depending on what language they were written in, but could be a show stopper for ISV supplied applications. There are almost 2,500 Linux on System z applications available.

Note: For more information on applications available for Linux on System z see: 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. As an IFL is usually classified as a single core, there could be significant savings by consolidating multiple distributed servers to an IFL.

To understand the potential software cost savings of a migration to IBM System z you should contact your software vendor in order to understand its policies and pricing with regard to application consolidation on System z.

8.3 Basic rules for selecting an application for migration to Linux on System z

This section discusses some of the basic rules used for selecting an application to migrate to Linux on System z. Here, we list applications that shouldn't or can't be migrated to Linux on System z as well as which applications are suited to migration.

Applications that shouldn't or can't be migrated:

- Applications that are available only on Intel or Unix platforms.
  - *Requesting that an ISV support their application on Linux on System z is a long process.*
- Servers that have already been virtualized.

  - *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 - 12 months of expiry, there may be a good business case to move the applications to Linux on System z with 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 on other platforms.*

Applications suited to migration:

- Applications or middleware (database, application servers etc.) 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 they are components of System z applications.
  - *You can supercharge your Linux on System z applications by putting them on the same physical server as their data source.*
- Applications with high and/or transactional I/O.
  - *Due to its design IBM System z excells at handling sustained high I/O rates.*
- Applications with lower sustained CPU peaks and average memory needs.
  - *These are the ideal workloads for IBM System z and 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 following applications leverage the System z platform classic strengths. The classic strengths of the System z platform that are of most importance for best fit applications are 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 legacy 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 painless. One of the reasons for this is that a many 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. You will find that this simplifies a migration considerably. IBM's software brands are:

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

> **Note:** For a complete up to date list of IBM applications and middleware supported on Linux on System z please go to:
>
> `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 realitively simple process. For more details on migration of WebSphere Application Server and DB2 database see Chapter 14, "MS Windows to Linux: WebSphere and DB2 Migration" on page 169

8.4.2 Oracle

As the Orcale 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), Oracle's high availability clustering soultion. 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 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.

Important: To get the latest information on 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.html

Note: For more details on how to install Oracle 10g, setup a Real Application Cluster and Install Oracle Application Server on IBM System z please see 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 details of an Oracle 10g migration from Solaris to Linux on System z see “Migration of archived Oracle databases across platforms” on page 124.

8.4.3 Other Software

The following is a list of some of the other non-IBM software that make good candidates for migration to Linux on System z.

SAP

- SAP application servers are also good candidates for consolidation to Linux on System z and is particularly appropriate if the SAP database is DB2 on z/OS. The benefits of HiperSockets and virtual networks reduce substantial amounts of network overhead that can occur on physical networks. Additional benefits are potentially reducing the systems 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 in many IBM System z accounts.

Infrastructure Services

- Network infrastructure, FTP, NFS, DNS etc are very well served on Linux on System z. These workloads are generally minimal, yet they are critical to the business. The main advantage of hosting these services on Linux on System z are the availability of the hardware's disaster recovery capabilities. Since a great deal 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 are used, not only is the 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 . There are both OPEN LDAP products as well as commercial products line Tivoli Directory Server, Tivoli Directory Integrator and Tivoli Access Manager. Using System z architecture customers can build a robust 24 x 7 LDAP infrastructure.

Application Development

- Whether it is Java, C/C++ or most other programming languages the vitrtualised Linux environment is an ideal platform for application development. While the developer generally develops on a stand-alone platform the testing and modifications are generally done in a server environment. Developers can be given multiple virtual servers to perform interactive testing while trouble shooting or enhancing the application. z/VM also provides a number of features that enhance application trouble shooting.
- Other major benefits are the ability to rapidly deploy virtual servers for user acceptance testing and integration testing and when that is finished the virtual servers are shutdown. If a developer inadvertently 'damages' a virtual server all that has to happen is to clone a new server. No need to spend a week formatting disks and re-installing the operating system and required applications.
- For new applications virtual servers are deployed in minutes and can be easily customized for a specific purpose. Many customers have standard server profiles that are pre-built so all that has to happen is the appropriate profile is cloned in order to create a new virtual server. This can be done in minutes. When an application is discarded for some reason the virtual servers can be discarded as well.

> **Note:** For more details on using the Linux on System z environment for application development please see *Linux on IBM eServer zSeries and S/390: Application Development*, SG24-6807

There is an extensive list of applications and solutions across all industries from over 60 ISVs that are certified and supported on Linux on System z. This information is available at:

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

8.4.4 Selecting an Application for a Proof of Concept

Once the business case has been done which shows that a Linux on System z migration will provide a positive Return on Investment, most customers will do two things at this point. One is to talk to other customers who have migrated applications to Linux on System z to understand how their migration went and get their recommendations on what they would or would not do next time. The other is to choose one of your own applications as a candidate for a proof of concept.

When choosing an application for a proof of concept it is best to keep it as simple as possible. The reason being that a proof of concept is done to show that an application can be migrated to a Linux on System z environment and that the application results are the same as the production system.

The general recommendation is to select an application that is reasonably self-contained and doesn't rely too much on inputs from multiple sources and other applications. The other factor is to choose an application that doesn't require a major re-write to run on Linux on System z.

The best applications to consider are Java based as 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 provided you have the source code as they will have to be re-compiled for the IBM System z platform.

Once the application has been chosen you also need to define the end objective or success factor. The minimum would be to produce the identical results as the production version.

For an example of a proof of concept for a customer migrating WebSphere Application Server and DB2 from Windows to Linux on System z please refer to Chapter 14, “MS Windows to Linux: WebSphere and DB2 Migration” on page 169

8.4.5 What if my application isn't supported on Linux on System z?

If the chosen application for migration to Linux on System z is not supported on Linux on System z by the software vendor the best approach is to contact the software vendor and ask them to support Linux on System z. However, this will not happen overnight so it is suggested that another application be chosen for 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 stays on its existing or modernised platform and communicates with Linux on System z.

One example of this would be a reporting program running on Solaris, that analyzes an Oracle database, also on Solaris. The Oracle database could be migrated to Linux on System z and the reporting program 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. The realtity is that an application migration to Linux on System z provides an opportunity to potentially simplify your application without having an impact on any interdependencies.

Many distributed applications have grown, in only a few years, from a single server to tens all the way up to hundreds of interconnected servers. These interconnected servers not only add network overhead but they 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. As System z supports all modern communication methods it is a quite straight forward process to continue to receive data inputs and transmit data outputs in the same way before the application was migrated. In this case there are no changes to external applications.

The main thing to remember during the migration planning is to map out completely 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 Sucessful Application migration

This section outlines some tips that should be followed to achieve a successful application migration for Java and C/C++ programs.

8.5.1 Special considerations for migrating Java application

We know that migration of 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 that needs to be taken in to account.

– Since 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). So if the application is written to conform to a particular Java distribution, then the helper classes referenced in the application need to be changed to refer to the new Java distribution classes.

– There are some special procedures that should be followed to get the best application migration. One critical point is to update the JVM to the current stable version. The compatibility with the old versions is significant and there are also performance improvements that benefits applications.

– Ensure that the Just-In-Time (JIT) compiler is enabled.

– Set the the minimal heap size (-Xms) equal to the maximal heap size (-Xmx). The size of the heap size should be always less the the total of memory configured to the server. In most cases large heap size implies at better performance.

8.5.2 Special considerations for migrating C++ application

When migrating C++ applications, there a few special considerations you need to keep in mind.

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 & return structure

Use care in the use of ptrace and the return structure as it is 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 isn't possible (for example, shared files) the application may have to be reworked to adjust for processing little endian data.

Stack frame layout and linkage is specific to S/390

See /usr/src/linux/Documentation/Debugging390.txt for details. Location of this file may vary depending on the distribution. In one instance, it was found in the following file: /usr/src/linux-2.2.16.SuSE/Documentation.

Changes to Build Scripts

There will be a need to make appropriate changes or updates to the Configuration/build/Makefile scripts or files and also there is a requirement to add support for the s390 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/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, 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 effort of the developer team, network team , middleware administrators team and Linux on System z team . Without the co-operation of all these groups, it is very difficult to have a successful migration.

We recommend the following steps for a successful migration:

1. Source application mapping

This first step is to analyze the source application with regard to its suitability to migration to System z keeping in mind the following:

   – Is the source code avaliable to be compiled/deployed on the target server?
   – Is there a version of the middleware avaliable for Linux on System z?
   – Are there performance reports of actual development test to be compared with after the migration ?

2. Design the network solution for the application (see Chapter 6, “Network analysis” on page 63 for more information).
3. Design the file system for the application and middleware (see Chapter 7, “Storage analysis” on page 77 for more information).
4. Clone the Linux on System z server(s) from golden image.
5. Configure the network at target server(s).
6. Create the custom file sytem at target server(s).
7. Install and configure middleware at target server.
8. Copy the application code from source to target server .

9. Compile and deploy the application code to target server .
10. Provide the first application test reports .
11. Start the performance test on the target server to understand 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. Shutdown the source server .
15. Change the IP address and hostname of the target server or change the DNS configuration to the target application server.

9

Database analysis

This chapter contains information about the configurations of the database application server on Linux on System z. Some best practices for different database software are also provide in this chapter.

9.1 Before the database migration

The database server is one of the most advocated services to be migrated to Linux on System z. But it is also the one that needs more attention to planning because there are also techinical configuration changes that must be considered.

During the migration planning discussion, the workload of the instances and the databases that are running at the source enviroment must be considered as well as the number of concurrent users and the number of instances and databases running in a unique source server.

9.1.1 Single instance migration

For single instance servers, the migration can be considered simple because the number of the variables from source enviroment to the new destination enviroment is relatively small. Consider the following steps in order to migrate when using the same database software vendor and version:

1. Configure the Linux on System z network (steps 1-4 from Appendix 6.2, “Steps for a sucessfull network migration” on page 75).
2. Configure the temporary storage area at source and destinantion 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. Shutdown the source servers.
7. Change the Linux on System z server hostname and IP address.
8. Perform import procedures at the destination server.
9. Perform the database and applications tests.

9.1.2 Multiple instance migration

For a multiple instance on a single server or multiple instances on multiple servers, the analysis of the migration is more detailed and complicated . In constrast, the results of the migration are lower license cost, less data center space and energy savings, better performance and other administration benefits.

Multiple servers to Linux on System z

One of the most important indicators for the migration relates to the distribution of servers peak load. The peak workload periods, how long it takes and how much server resources are used are the values that should be written down and compared. Table 9-1 on page 107 can be used to map the servers' workload to create the migration configurations.

Server Information			Peak Load Measure		Peak Load Time		
Server Name	Total of CPU	Total of Memory	%CPU used	%Memory Used	Week Day	Start Hour	Execution Time

Figure 9-1 Workload Table

As explained in Chapter 5., "Technical concepts" on page 55 the CPU and memory constraints in an LPAR is possible and desirable, but the server should maintain the same peak load for a long period of time if there aren't real CPUs to proccess each virtual CPU request.

To exemplify, think about the 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 are four and the number of real CPUs are three. The z/VM share algorithm will handle all the processor requests and the server still would be available but the performance of the application would probably be very good and would also affect LinuxC's response time. However, if the server peak load of LinuxA and LinuxB happens at different times, the entire LPAR will not be affected.

This kind of constraint is fine if it happens in miliseconds to second intervals but it' s not acceptable in interval lasting in minutes. This can be acceptable in 5 minute intervals depending on how critical the server is to the business purpose but never more than that.

The correct workload capacity plan is key for a sucesseful migration in case of multiple database servers in a single LPAR on System z.

Another point to consider regarding the CPU capacity is the relationship between the source server and migration server. This is not 1:1, in other words one distribuited server with 4 CPUs must not necessarily have 4 CPUs in the destination virtual server, best practices shows that the actual number is less than that . For more information see 9.2.1, "CPU" on page 109 in the "Technical Considerations" section of this chapter .

Multiple instance servers to Linux on System z

In a migration of a unique server running several database instances, the greater advantage is the simplicity of the database management allied with cost reduction. To get a good result 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 overcommitement of CPUs and memory.

In an environment where the instances are divided among various virtual servers, a software problem that occurs on a specific instance will affect only the database server where the instance is running. In this case, an entire server reboot would be necessary.

License savings can be shown, for example, where multiple instances are running on an 8 CPU server. The entire system will be charged by the eight CPUs whether it is used or not. When the software is migrated to Linux on System z, the LPAR can have 6 IFLs and the entire system will be charged for six CPUs and if the entire LPAR is not using all 6 CPUs, it is possible to reduce the number of CPUs allocated to this LPAR and reduce the price paid for a database license and create new Linux servers for new instances without any cost.

In order to minimize the work related to database software fixes and security updates, it is possible to use shared devices for database binaries and libraries

(see 7.2.2, “Shared file system” on page 86 in Chapter 7., “Storage analysis” on page 77 for more details).

Things to consider when migrating 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 ?
- In 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 ?

The same table from Figure 9-1 on page 107 can be used to map the instances used by changing the server name column to instance name .

With this information, it is possible to configure multiple servers in an LPAR to respond to all user requests without loss of performance and with a gain in 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 from an entire month instead of one single day of information . The more days you have, the more accurate your information will be.

9.2 Technical Considerations

Database managment software is the type of application that demands a better analysis when the migration is the option chosen. 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 and storage bound and table access should be considered at server migration.

9.2.1 CPU

The number of virtual CPU resources is very important in a database server. To set the most CPU to use that is possible isn’t a guarantee of better performance. The number of CPUs should be enough to avoid the processor queue. The

processor queue value is consider a high value when it is greater than 5 per processor in a 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 separated resource. Memory constraints or I/O constraints affect the processor queue number directly, so before determining if the server doesn't 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. In the same way, if the CPU time is dedicated to I/O WAIT, the file system should be reorganized.

The minimum number of virtual CPU 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 good value to start the database validation.

When validating the server migration, the number of CPUs can be increased with a server reboot. The best practice is start with the minimum number of CPUs as described in the previous paragraph and increasing according to need.

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 obtain good performance, but with Linux on System z, allocating more resources isn' t related to getting more performance. The machine should be sized as needed and one of the considerations should be 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 source server and increase or decrease as needed.

SWAP memory

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

A best practices with System z is to use the V-DISK devices as swap devices. As swap configured at V-DISK devices provides good response time, the eventual memory paging (the process that moves memory blocks from/to real memory from/to swap memory) is not consider a real problem and also there isn' t a problem fi 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 Linux on System z monitoring please refer to Appendix C, "Performance measurement" on page 267.

If the servers shows a very high paging value in a period of higher than 5 minutes, it is recommended that you 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 aren't used in real memory as its default configuration but that isnt the most desirable solution when considering database servers. In fact, it's best to avoid this type of situation. There is a configurable kernel parameter called *swappiness* that determines if more or fewer pages will be swapped (Example 9-1, "/proc/sys/vm/swappiness").

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

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

The configuration above will not avoid Linux swapping but will decrease the swap out process .

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. 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 2G of SWAP memory . For better performance, monitor the swap when the stress test occurs and increase or decrease it accordingly.

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 communicated: 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 would perform less I/O.

So to get the best memory allocation, it is necessary to set some Linux kernel parameters and these parameters depend on what the DBA allocated in the migration . In Example 9-3, “Shared Memory Configuration” , the server was set to use 60% (4.8G) of the 8G 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, “Messages Queue Configuration” shows 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

The data storage access ona database server is intensive and there is some consideration that should be taken 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 on 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 (7.2.4, “Logical Volume Manager Devices” on page 88) , this configuration will provide better I/O bandwidth on the system where there are intensive writes and sequential reads.

Chapter 5., "Technical concepts" on page 55 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 For sucessful migration

Almost all database servers use buffer pools in the shared memory area to manage the database memory context. Avoid any automatic memory management systems to allocate shared memory. If there is 6GB of shared memory to be allocated to the database application, force the database application to allocate all memory at the system start.

If the database server isn't 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, add more virtual CPU to the server. But the entire LPAR workload should be monitored in order to avoid the performance of a Linux guest interfering withl another Linux guest.

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

The Linux Administrator and Database Adminsitrator must work together in the Linux guest sizing process as changes may need to be done 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 IBM Informix® Dynamic Server Versions 10 or 11.10.

In Figure 9-2 the source and target databases supported by the IBM Migration Toolkit are listed.

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 9-2 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

For more details see:

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 will cover data protection or backup, record retention or archiving, storage management and security.

10.1.1 Backup concepts

Backup, means 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. Usually old backup versions are automatically deleted as new versions are created, if the number of versions stored exceeds the defined limit, or may be deleted after a certain 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, and therefore, special consideration should be given 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 once 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 & 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 Backup and Archived Data Migration

When moving to an newer or more modern environment, the archived data in the existing environment may no longer be supported depending of 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 migrate archived data?

Some of the factors which force the migration of archived data are:

- The preservation of data on the same medium where it was originally acquired would be facing the dual problem of the lifetime of the medium itself and of the long-term availability of the technology for reading it.
- Usually as time passes by, technologies age and eventually turn un-economic, or not competitive versus emerging ones, and that point alone in some cases may render the technology change very attractive.
- Some older storage technologies have a direct impact over the volume of data that can be stored and 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, most often, the disappearance of a technology is driven by the market and the difficulty in procurement of the related media and getting support of the withdrawn operating environments (servers) by the manufactures.

10.3 General Archival Migration Considerations

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

- Change in 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, in

order to allow efficient data retrieval the data inventory structures may need to be re-converted.

Since the operating system and the backup/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 straight forward as the storage backup/archival manager product would be able to access the existing archived data.

Usually any backup and archival manager would have built in migration tools that would migrate the archived data from the source operating environment to the target new environment. At this point it is recommened to re-organize the archives and purge the unwanted data. This would 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 with new technologies that are either proprietary or open source.

In this case the migration of the data from the existing archives would be bit time consuming because of the following reasons :

- Incompatible filesystem in the current and the target environment.

  Even though most of the Unix distributions use similar filesystem structures, there are differences in the way the filesystems are organized and how data is stored.

- Software Vendor Incompatibility.

  Meaning, the data archived using one vendor’s backup/migration manager software can only be accessed or restored by the same vendor software. Other software vendors’ backup and archival software may not be able to access or restore the archived data.

- The format and other charectertics of the archive would be only understandable to that software, which means that the data needs to be re-archived and relabled for restoraion.
- In case of database raw file backups, the database itself is required to restore the database and re-back it up using the new backup/archival software.
- In most of the cases the archived data would be accessed by a variety of applications for a number of purposes. If there is a change in the archival

format, these interconnected applications would need to be re-designed to access the new archival data.

- Ensuring the data integrity (no data loss or degradation) implies verification of the data after conversion. This will require some basic control in screening and a comparison of the new data being archived with the same data in the old archives.

10.4 Archived data migration aproaches

In this section, we will discuss the migration approaches that can be employed when there is a change in the target environment's software stack.

10.4.1 Restore to a temporary staging server and re-archive

In this method, the entire archived data needs to be restored to the staging server with the same operating system environment and if the new backup/archival manager is compatible with the existing operating system, then it would be used to backup the restored data from the staging server to Linux on System z.

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

10.4.2 Restore to target file system and re-archive

In this approach the archived data is restored from the old system to the new Linux on System z server, which means the old backup/archive managing software would restore and export the archived data to new filesystem partitions. Then exported archive data can be re-archived into the target archiving system. Here the current or old archival software needs to assesed for features that allows the product to export archived data to other filesystem. Basically the need for maintianing staging servers is reduced.

Also by incorporating this step, it may be possibile to perform the automation of the data archiving, retrieval and re-transcription, by using native scripting as well as features in the software products.

10.4.3 Leaving the archive data on the secondary server

In this approach, we would 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 however requires that your new target operating implementation doesn't require your old storage unit. But care can be taken so that whenever an archived data restore happens, it would be rearchived through the new software and at the same time deleted from the old archival environment. This way the space requirements would also gradually reduce within the old archival environment and reallocated to the new archival environment.

10.5 Restore - rearchive Scenario

In Figure 10-1 on page 122 we see an example of the information from one software product being restored to a staging file system and then archived using another software product, such as the IBM Tivoli Storgage Manager, accordingly.

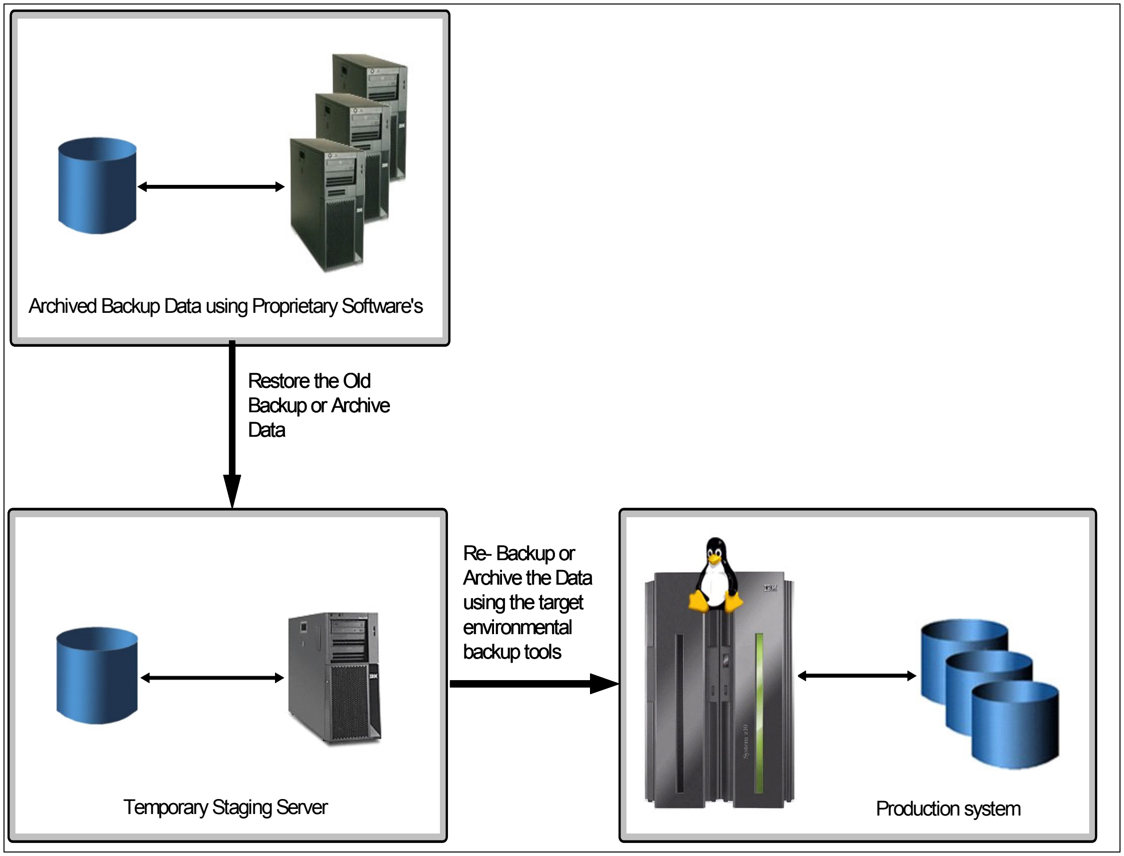


Figure 10-1 Using a staging server to migrate existing backups/archives to new technology

10.5.1 Application logs and redo logs

In order 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 as per the defined organisational attributes and backup policies.

10.6 Database raw file archives

Since here there is one more entity (Database) involved, the migration approach would depend on the following:

- Retention of database software in the target environment
- Change of database software in the target environment

10.6.1 Retention of database software in the target environment

While retaining the same database software in the target environment, the migration would only rearchive the data into database files using the target backup/archival software.

The following generic steps could be used to do this:

1. Using the source backup/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 values or CSV).
3. The staging server, accessed by the target backup/archive software then re-achives the restored file (step 1).

Migration of 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 would be via 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 hierachy 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 utiliy 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:** The actual steps on the DB2 Migration commands can be found at“DB2 database migration” on page 173

Migration of 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. And using the DDL Scripts create the respective tables, indexes.,etc.

> **Important:** Before importing the dump file, you must first create your tablespaces, using the information obtained in Step 1. 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/archival of your new database.

11

Security analysis

This chapter provides, a high level discussion of what needs to be included in analyzing programs and functions that are going to be part of the migration in the context of security. Next, we will discuss what enterprise wide authentication options are available and how they may play a role in migration. Since SSL/SSH is probably going to be used, we will discuss the use of the cryptography hardware that is available.

This chapter includes discussions on 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

One might assume that simply migrating an application from it's existing server to the target Linux on z/VM server would mean that the security would be kept the same. While that certainly could be the case, it probably is not going to be the case. One of the main advantages of migrating to z/VM is the access to enterprise class security. Thus the best time to plan and to take advantage of it would be the migration process.

The security analysis will center around

- 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 since, if a compromise occurs, its exposure is only to the effected 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.

So, like a fort or a castle, you are only as good as your weakest point. That is why security is very time consuming and it is difficult to predict the amount of time analysis will take. If this is the first time doing a security analysis, do not underestimate the time or scope of the task.

It is a general belief that "security through obscurity" is not a valid method. Use of open well vetted security methods implemented correctly will provide the best defense. As an example it is not recommended to develop your own cyptographic libraries, but to use open established ones that have been vetted for many years.

Hiding information creates more system administration work and any mistakes will allow an attack to surface in the system that can be exploited.

System logs as well as application logs need to be immutable. Logs need to be kept in such a way that they can not be altered by a user of the system. 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 assumptions made. Include all “what if” situations that are reasonable to expect to occur. Also document security plans such as in change control, audits and procedures for break-in in all domains.

11.1.2 Understanding the z/VM foundation

The control of the Linux virtual machine (VM) is done at the z/VM layer. Thus an understanding of its security and access is needed for a complete security survey to be done. 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 userid on the z/VM system is MAINT. This user has root authority and as such must be secured.

System z and existing security policies

In organizations there is often an existing security policy which dictates that the mainframe must not be internet facing. With the migration of a distributed environment to Linux on System z this often raises some questions concerning the role of System z within the existing security policy. The recommended approach regarding security policies is to conform with the existing policy as much as possible as it simplifies the migration process. While z/OS is usually 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 amongst 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 on Common Criteria please see Chapter 1.3.1, “System z Strengths” on page 16

To further ensure the isolation of the z/VM LPAR from the z/OS LPAR the Open System 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 there is 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. While 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 as the firewall software may not be used in the organization and as such would have to be certified. This could be a long and complicated process. The second is to continue to use the physical firewalls by having the inter-security level communication exit the virtual environment via an Open Systems Adapter (OSA), go through the physical firewall and return to the virtual environment via a different OSA. Figure 11-1 on page 131 describes the use of an external firewall.

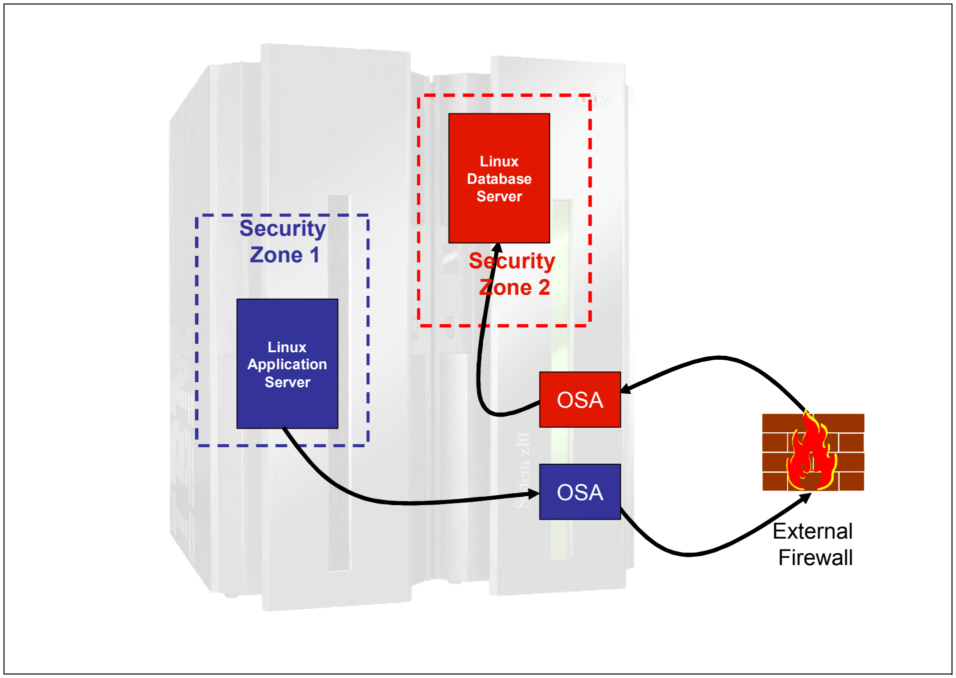


Figure 11-1 Using external firewalls between security zones

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

While the general recommendation during a migration is to conform to the existing security policy in order to not unnecessarily complicate the migration. The reality is that for applications within the System z footprint, as in the above example, there may not be a requirement for any firewalls provided that all incoming communications to System z are processed by external firewalls.

Who will control z/VM?

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

Also make sure to set aside a change window to allow for z/VM maintenance. You need to plan for z/VM maintenance which may require that some or all of the VM's be quiesced. So there should be a plan in place and a set schedule to allow for security and z/VM updates or maintance.

Further reading

Lucky for you there are some really good IBM Redbooks publications about z/VM and hosting Linux VMs. Here are some that we found applicable to security and networks:

- *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 of 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. Thus during migration, you may be given an already hardened Linux image, which you will just need to understand what is allowed or not with the image. But if one does not exist, then time should be given to creating one and maintaining it.

Creating a new hardened Linux VM

The basics of hardening a Linux VM are to remove all unnecessary applications and services. Then to secure the applications and services that are left. This process is outside the scope of this redbook but here are some suggested readings and references:

- 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

Migration to a hardened Linux VM

A hardned Linux VM should have most if not all applications and services removed and or disabled.There may be more then one hardened Linux VM to chose from, so make sure to pick the version that provides the most applications and or services needed to accomplish the migration.

Thus 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 as a set or individually. Thus completion of the security analysis can provide just such a case, once the

security analysis is completed. Make sure while documenting to include all applications and services as a delta from the base hardened Linux image.

Important: RHEL included the SElinux security method and SLES includes AppArmor® for it's enhanced security method. Determine if those environments are in use or required, and thus plan accordingly. It's important to note that these mechanisms are very complex and the time needs to be taken to identify code and or applications that have not been ported to work in these environments.

Maintaining a hardened Linux VM

It is nessary to maintain base hardened Linux VMs. Kernels change and security patches are issued, thus a plan for maintain the base image is needed and resources assigned 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 getting migrated. You need to know what the current security methods are. Next understand what the security methods will be in the target Linux environment. Are there to be enhancements? Finally you should then poll the stake holders to make sure that all migration security requirements will be fulfilled.

When moving an application to a Linux VM you should consider using as many VMs as you can. That is, separate as much as possible and thus use the Linux VM as a way to isolate applications from one another and their data. This is the opposite thinking when working on stand alone servers in which consolidation is the focus. Therefore, if many images are available, design the system so that as much separation exists between applications and data. The more isolation the more straight forward security will be.

11.2.1 Are there known security issues?

This section discusses how to look for potential security issues when migrating code and applications.

Migrating Code

Sounds silly to ask, but you need to, especially if you are taking over the code from others. If you are migrating the code to a Linux VM which 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. Thus any and all known issues need to be addressed, so plan for it.

Migrating Applications

If you know there is a security issue with an application, don't 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, spend the time to make those changes during migration. As an example, you should place a database on a different VM then the application using it. The more separation, the more straight forward security will be. Systems with easy to understand security tend to be easier to defend and maintain.

11.2.2 What are the dependences?

This section discusses how to look for dependencies before you migrate.

Code dependences

Almost all code uses API's and other libraries to carry out the tasks that it was designed for. Thus a review off these dependences needs to be done. If it is discovered that a dependency is on an item that has a known security issue, then a suitable replacement needs to be found and implemented.

Application dependences

A list of all application dependences 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. 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 this is invalid if any application or its dependency is on code that has known issues.

11.2.3 Are all user inputs being checked?

User input is the most used vector for attack of a system or program. All user interaction needs to be examined carefully. Check all input to make sure it is within the range of the data needed to be processed. A good policy is that raw input should never be passed to another application or system request. Also use of exceptions should be used, that is, always try to make sure that input conforms to the format that is expected and if the unexpected occurs it can be gracefully handled.

11.2.4 Plan 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 made while the application is still running. Make sure that there is a method to signal that configuration and or control files have been updated and need to be re-loaded. There may be a security issue that needs to be addressed by configuration changes. Thus in an Enterprise environment a program should not be stopped but signaled to take on changes. As an example you may need to change the TCP port that an application uses. Can the code handle the change gracefully?

Also make sure to carefully examine all configuration changes. Do not assume that the changes are valid, again check to make sure they are with in 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 most likely need to know:

- What ports will be used by the code for what purpose?
- What type of protocol will be used - TCP, UDP, ICMP, etc.?
- 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 of events

All logs need to be kept in a way so that they can not be changed. They need to be a permanent record of what occurred on the system. The Linux VM should be configured so that syslog (the Linux system log) has both a local record and is also forwarding it to a remote secure system. Also make sure that all critical applications are properly configured to use syslog.

> **Important:** SuSE has gone from using syslog to syslog-ng (ng stands for next generation). syslog-ng is improved and the preferred system logging facility. At the time of this writing, RHEL was still using syslog.

Implement syslog logging when migrating code

On the Linux VM either syslogd or syslog-ng will be running. Time should be taken to update the code as nessary to message this daemon. At the very least, all information that deals with security should be logged as well as critical state information. The advantage of implementing syslog functionality is that log maintenance will be done by the system (as in log rotation and archiving).

11.2.7 Understand all escalations of authority

Apply the principal of “principle of least privilege”. That is, the program should only operate with the authority it needs to accomplish it’s goal. So if the code accesses a data base, it should not access it as the administrator but a user who only has the access needed.

Migrating Code

Thus code should be analyzed to find all parts where there are escalations of authority. Then make sure that they account for execptions so that a de-escalation of authority exists. That is if the code can be broken it does not allow the user to operate at a different access level then is allowed.

Migrating Applications

Programs that run as “root”, the super user, needs to 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 then find out 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 thus 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 vet 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 is more then just who can access a system. Security is also about keeping the system available to the authorized users and unavailable to those that are not. The use of the DoS (Denial of Service) attack has become more frequent in recent years. Thus any Internet facing system must account for this possible issue.

To have executable system security, there needs to be an audit trail. There can not be any exceptions. That is, all access of the system needs to be logged in a secure fashion to ensure that if an authorized user commits an indiscretion, that it can not be covered up.

11.3.1 Availability analysis

Sometimes an attacker does not break in to a system, but instead brings down a service by overwhelming it with requests. Thus system or services availability needs to be understood and service level agreements maintained.

Is the Linux VM internet facing?

Think of the internet as a public place in which for the most part individuals are anonymous. Every step needs to be taken to mitigate malicious access. You will need to be able to identify individuals and their IP address so that, if nessary, you can work with the networking team to prevent malicious access while still allowing authorized users to have access.

How is availably communicated?

Establish a standard for communicating system availably. Include how to report issues and outages making sure they are communicated back to the correct individuals. A 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

All system logs as well as all application logs need to be immutable. If an attacker gains access one of their first tasks is to erase their presence so that they are not detected. Also if a user is attempting to do something they are not authorized to do, they will most likely attempt to cover their indiscretion by erasing log files or any incriminating evidence.

Are log files immutable?

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

Is there an audit trail that encompass all security domains?

Can you trace an individual's physical, network and application access to systems across the domains? Make sure that a security audit can be passed at all times. It is nessary to 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. Another method of attack is that the attacker might spoof or pretend to be a system or user that they are not. Three things must occur to protect against this:

- The user must be assured that they are connecting to the server they think they are.
- The server needs to be assured that the user is who they say they are.
- During this "Authentication" conversation it must be kept private so that eavesdropping can not occur

This is why it is good security practice to use Secure Shell (SSH) which will accomplish all three and disable telnet access. The use of Secure Sockets Layer (SSL) with web servers will also accomplish all three and is preferred over the default of no SSL.

11.4 Data integrity and confidentiality

The advantage of migrating to a Linux VM is that now data can be stored on an enterprise class system. Thus one must analyze what the current state of the data is and how it will fit in the new enterprise system.

11.4.1 Data integrity analysis

Data integrity is the assurance that the data is unchanged from creation to reception. Data integrity is also about understanding:

- Who can access what data and what is allowed
- Is there an audit trail in place to map who changed what, when.
- If the data is corrupted in some way, how is it to be restored.

- Is there a disaster recovery plan?

How is data at rest protected from unauthorized access?

Everyone knows how to protect access to a data base, but what about the raw data on the disk itself? We have seen how a laptop can get misplaced or stolen when it had a database full of accounts or data. Thus you need to protect data at rest. That is, the files themselves, making sure that the data is kept in a secure way. It is a good policy to prevent off-line copies of a data base to be kept on portable devices or drives. Control of data is key. Communicate data integrity policy to all individuals that have access and monitor the audit trail to make sure policy is being enforced.

Data backups are 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 were stolen and try and mitigate the risk.

11.4.2 Confidentiality analysis

Confidentiality needs to be communicated first then enforced. Thus before a user 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 violition is detected, then it will need to be communicated to the affected parties.

Understand 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) regarding the storage of credit card information?
- Is there a company policy, if so it needs to be followed.
- Document all expectations with how long you will keep the data, as in “we expect or are required to keep the data for up to 5 years”

Publish your confidentiality policy

You need to communicate the confidentiality policy in such a way as to assure that all users of the system are aware 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 in Example 11-1 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 the use of special charters to make the login screen attractive. The Linux VM is special, so it’s greeting should reflect that. It is also a good idea to communicate system information such as the Linux distribution with its version and release information.

On web pages, there should be a link from the main page so that the system policy can be easily accessed. A good example is Google, their privacy policy is easily found at `http://www.google.com/intl/en/privacypolicy.html`

If you are allowing VNC login as talked about in Appendix B, “Remote access applications” on page 249 you will want to display the policy by updating /etc/gdm/custom.conf as 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
```

Have a plan in place before an incident occurs

Don’t be caught off guard. Have a plan in place in case confidentiality was 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.
- Plan to communicate changes that will be taken to prevent future incidents.

11.5 Security change management

No system is perfect thus there will be changes, however infrequent. Since security fixes are important to keep up with, there should be a plan to understand their impact on the system and if a Linux VM needs to be re-started, it needs to be done in an orderly and timely basis.

Once the system is moved from test to production mode it will remain that way. Outages cost the company money, but failure to plan change windows and down time will also cause a security problem. In the rare case that a VM needs to be restarted you need the ability to allow for these types of changes.

Test 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, do not assume it will work. Also record how long it takes to make changes and test worse case scenarios (also keeping track of the time). Then when testing of the change has been completed, you will be able to report to production stakeholders how long it will take and how long the worst case will take. Production stakeholders hate surprises

11.6 Enterprise authentication options

An advantage of migrating to an enterprise system is that user and identification management can be consolidated. Here we will discuss some of the options available and where to find the corresponding information on how to implement them.

11.6.1 A common centralized LDAP server

When migrating applications and code to a Linux VM there is an opportunity to simplify user administration by storing user information in an LDAP (Lightweight Directory Access Protocol) server. By configuring the Linux VM to authenticate from a centralized LDAP server results in:

- Simplification of user management. 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, but using LDAP means that even bringing online an old image will not compromise current security.

11.6.2 LDAP server on z/OS means RACF integration.

If RACF® is used to manage user information installing LDAP on a z/OS system will allow LDAP access to RACF. This will allow one single highly secure repository of user information in RACF and thus allow that information to be exposed to Linux VM's via an LDAP server. Please see *Linux on IBM zSeries and S/390: Securing Linux for zSeries with a Central z/OS LDAP Server (RACF)*, REDP-0221for more information.

It is also be possible to configure SAMBA to use LDAP as it's user repository. Thus you can have one security domain across MS Windows, AIX, Linux, with System z as the core. Please see *Open Your Windows with Samba on Linux*, REDP-3780 for more information on SAMBA on Linux.

11.7 Integrated Cryptographic Service Facility (ICSF)

An advantage of migrating to a Linux VM is that the underlying hardware has the ability to accelerate cryptographic mathematics. The ICSF allows for this work to be off-loaded from the processor and done by the crypto-assist processor integrated into every processing unit (PU) of every System z9® and System z10U or the Crypto Express card if installed. The supported API's are

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 (GSKit)

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

Thus will off load the cycles and allow for more concurrent access to a web server that is using SSL or applications that use one of supported API's. Please read *Using Cryptographic Adapters for Web Servers with Linux on IBM System z9 and zSeries*, REDP-4131 to find out 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 will discuss some of the operational issues that, if present in the source application, need to be addrressed 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 funcions 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 successfuly ported but it is just as important that the application's operational support structures be migrated successfully as well.

12.1 Defining the Operational Environment

.There are many tasks and challenges that are present in the operational environment. An operator is often required to multi-task in many physcial and mental ways(Figure 12-1). The following picture is only a small depiction of the operational environment. This only shows consoles that need to be monitored. The operator still has a lot of other physical equipment to manage. Also not shown are the phones that can be a continual interrruption. It is very important to make sure that the migrated application fits in smoothly with the currrent operational environment. We will next discuss some operational tasks that might be affected by the source application migrating to the target application running on Linux on System z.

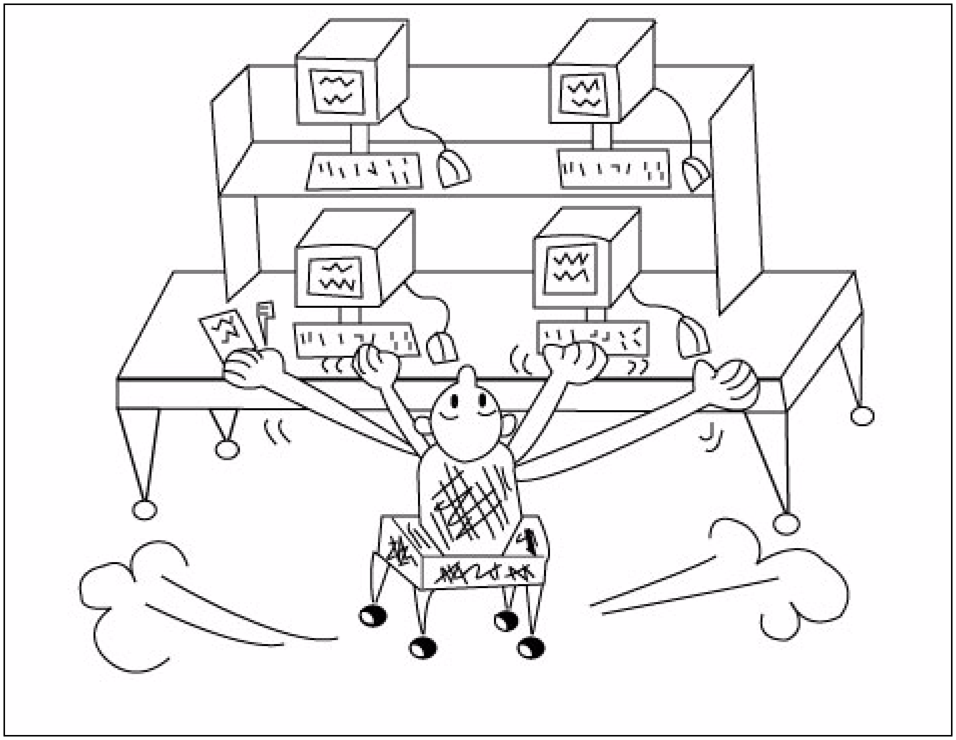


Figure 12-1 The Busy Operator

12.2 Operational Migration Tasks

We will discuss 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 had methods to start and stop its processes. The target application will have different commands and methods for starting and stopping the application. If the target application is a manual process, then the operators have to be trained and the appropriate documentation needs 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.

- Notificaton of Problems.

  Sometimes the operator will get an automated message or indicator that they are not familiar with and do not know how to respond. The operator needs to know who to turn to when this type of problem arises. The application owner needs to be clearly identified. If the application owner is not available or not responding, 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 lifecycle throughout the day. Sometimes this is just watching 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 best for operators as they learn by observation and performing the tasks required of them.

- Hardware Manipulation

  Some migrations will include hardware consolidation or hardware replacement. Operators need to be trained on the operation and manipulation of the new hardware. If the operator is not required to manipulate the hardware, it is still a good idea to let operators know what is running on the new server and have appropriate documentation, labels, and signs.

- Hardware Intervention and Problem Escalation

  There are fewer hardware interventions for the operator to deal with on System z. With the source application and server, an operator might be comfortable with and even required to reboot a server by using the power switch. The operator will be seriously out of place and making a grave error by using a power switch on System z when reacting to a server or application problem.

If there is a new hardware vendor in the migration project, then the method that the operator is to use to notify the vendor of an actionable message or event must 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 like serial numbers, machine type, location, etc.

- Batch proceedures 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 enviroment can be envisioned and described here. The intent of this chapter was 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 effected 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 & Availability Analysis

A migration of an application to Linux on System z gives an organization the opportunity to configure the application environment 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 will investigate the various levels of availability and illustrate this with examples of various high availability configurations.

13.1 Availability & Disaster Recovery: why do you need it?

IT systems are crucial to all organizations. When an IT outage occurs, for what ever reason, there can be a large impact on the business as crucial systems are unavailable. The key to ensuring this does not happen is to analyse your systems to determine a hierarchy of availability need. Remember, not everything has to have a remote hot site.

The starting point is to understand some of the terms used when talking about disaster recovery and high availability.

Disaster Recovery

The planning for and utilisation of redundant hardware, software, networks, facilities etc. to recover IT systems in the event of the unavailability of a data center or major components of an IT facility.

The following definitions of high availability, continuous operations and continuous availability come from the IBM High Availability Center of Competence:

High Availability (HA)

Provide service during defined periods, at acceptable or agreed upon levels, and masks unplanned outages from end-users. It employs Fault Tolerance, Automated Failure Detection, Recovery, Bypass Reconfiguration, Testing, Problem and Change Management.

Continuous Operations (CO)

Continuously operate and masked planned outages from end-users. It employs non-disruptive hardware and software changes, non disruptive configuration changes and software coexistance.

Continuous Availability (CA)

Deliver non-disruptive service to the end-user 7 days a week, 24 hours a day. There are no planned or un-planned outages.

The ultimate goal for mission critical systems should be Continuous Availability otherwise the systems shouldn’t be defined mission critical.

13.2 Availability analysis

The migration of an application to a virtualised 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 this isn't always straght forward. For example test and development workloads are generally not considered to be mission critical, but as they may be needed to correct an error in a production system then you should consider providing for some sort of test and development environment in your disaster recovery planning.

The challenge with disaster recovery is to get 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. What has to be included in this planning are the likely scenarios that could impact an application's availability plus unrelated events that could impact the ability of a business to function.

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

The aim of this chapter is to focus on the IT related issues but you should also have a plan for other non-IT related events.

13.3 Single Points of Failure

In determining the disaster recovery requirements of an application we need to look at the probability of failure of a component as well as the cost to eliminate a single point of failure.

The following table looks at 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
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 that requires a reboot
- Application maintenance

13.4 System z Features for High Availability

The IBM System z has been designed around providing high availability. Perhaps the most design 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 EC, the z10 BC will have zero 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 in order 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
- Ading a cryptographic processor to an LPAR

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

- 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 cater for the following events:

- CIU: Customer initiated upgrade for a permanent capacity upgrade
- CBU: Capacity BackUp : predefined capacity for disaster recovery. 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): 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: extra capacity in 24 hour increments is available to be turned on to satisfy peak demand in workloads.

Note: For more information of the IBM System z10 please see: *IBM System z10 Enterprise Class Technical Introduction*, SG24-7515

For an in depth document on IBM System z10 Reliability, Availibility & Serviceability Features see *IBM Journal of Research and Development Vol 53, No. 1 Paper 11 2009.*
http://www.research.ibm.com/journal/rd/531/clarke.pdf

13.5 Availability Scenarios

In the following scenarios we look at a number of different situations where a Linux on IBM System z enviornment is setup 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 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.

Don't forget 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 not be affected by natural disasters.

Another important factor in the availability of applications is security and access controls. For more information on this topic please read Chapter 11, "Security analysis" on page 127.

13.5.1 Single System z LPAR: Clustered WAS Environment

In Figure 13-2 we have a System z LPAR sharing system resources to all Linux virtual machines in the LPAR. The WebSphere Applications Servers (WAS) are in a 2 node cluster. If the Intergrated 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 WAS workload fails the other node in the cluster will take over provided you are running WAS Network Deployment. This is achieved because an application deployed to a cluster runs on all members concurrently. Additional availability is provided through the non-disruptive addition of new virtual machines to the cluster.

> **Note:** For more details on high availability solutions for WebSphere Application Server see: *WebSphere Application Server Network Deployment V6: High Availability Solutions*, SG24-6688

Note that z/OS is optional in the first 6 scenarios.

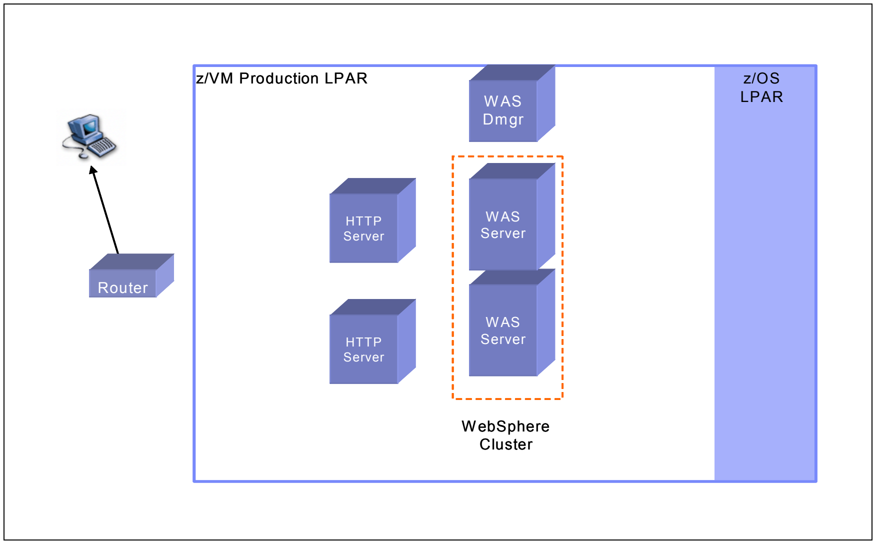


Figure 13-1 Single LPAR WAS Cluster

This environment also provides additional availability through redundant HTTP servers.

13.5.2 Multiple LPARs: HA Solution for Linux on System z

In Figure 13-3 we show 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 amongst all LPARs. The LPAR weight determines the relative priority of an LPAR against other LPARs.

In this case the production workload and WAS cluster is split across two LPARs which gives high availability to WAS as an LPAR or z/VM failure will not impact the availablity of WAS.

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.

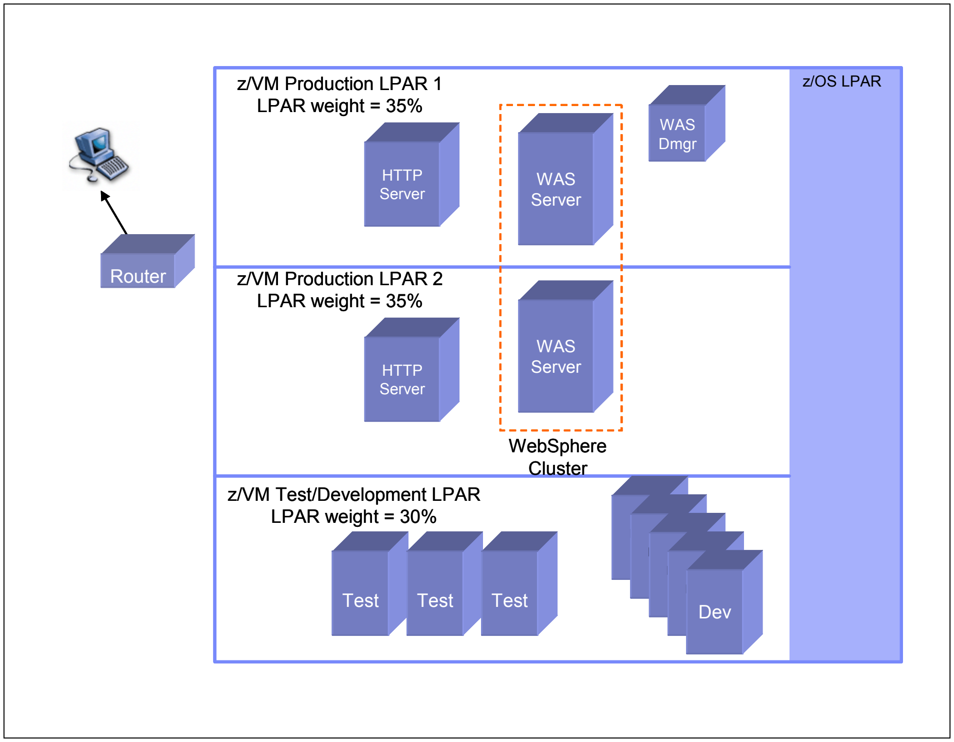


Figure 13-2 HA Linux Environment on IBM System z

13.5.3 Active/Cold Standby Cluster

Figure 13-4 describes another approach where 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 startes the HTTP Server on Node 1, the at user-defined intervals invokes a script to confirm it is still up and servibg pages. It also monitors the Linux node itself to ensure it remains active.

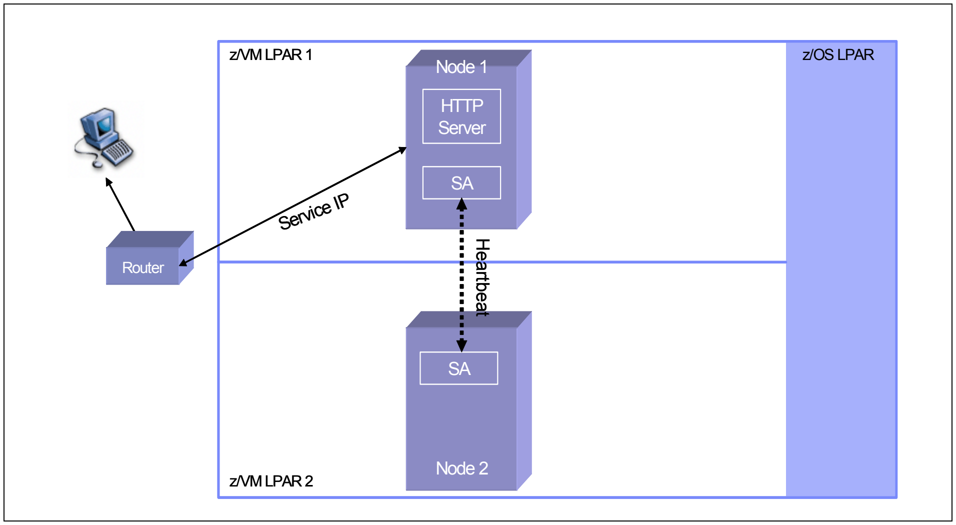


Figure 13-3 Normal Situation Tivoli System Automation monitors for outages

When a failure occurs RSCT determins that Node 1 is no longer responding. SA then moves the Service IP over to Node 2 and restarts the HTTP server there.

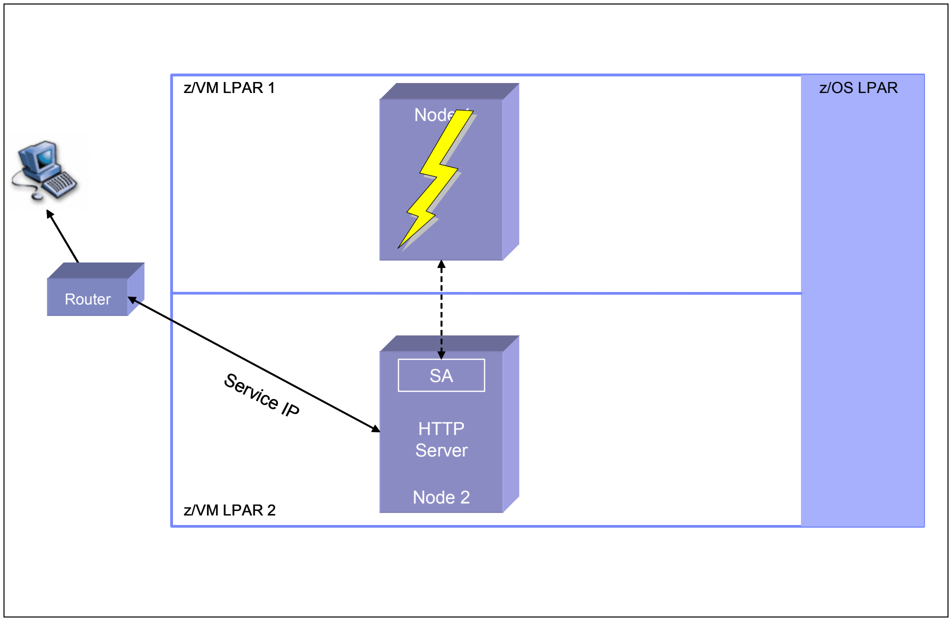


Figure 13-4 Outage occurs Tivoli System Automation fails over to cold standby

13.5.4 Active/Active Application Server Cluster

In Figure 13-6 the WAS environment is set-up in an active/active configuration where the WAS Cluster spans two Linux virtual machines in two LPARs. This set-up caters for the very rare occurance of the failure of an LPAR. ore importantly, it also allows z/VM maintenance to be done without an outage to the WebSphere applications. In this case the Linux servers and z/VM are shutdown 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. Obviously this would be scheduled for a time when the processing load is light.

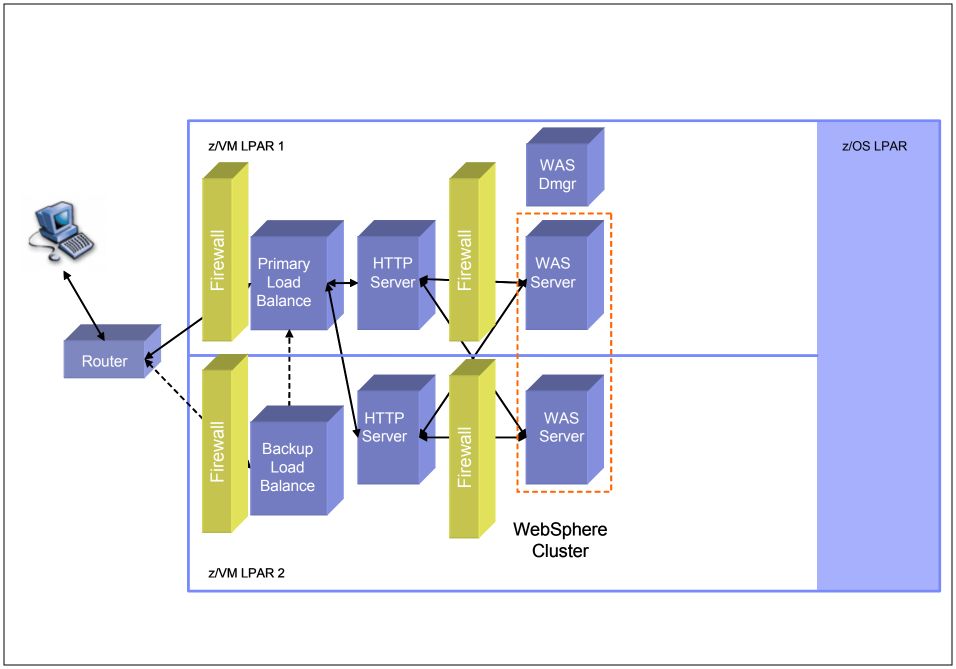


Figure 13-5 Active/active Websphere Application Server Cluster

13.5.5 Active/Active WAS Cluster with Database Replication

In Figure 13-7 a DB2 database is added to the active/active WebSphere Cluster. In order to provide high availability for the DB2 database the DB2 data replication feature, High Availability Disaster Recovery (HADR) is used. HADR protects against data failureby replication changes from the source database, called primary, to a target database, called standby. In then event of a z/VM or LPAR outage of the primary DB2 system the standby DB2 system will take over in seconds providing high availability. The communication between the DB2 primary and DB2 standby systems is via TCP/IP which in this case would be done using the System z's 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 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 high availability requirements of your DB2 database system.

Note: For more information of high availability DB2 features see *High Availability and Disaster Recovery Options for DB2 on Linux, UNIX, and Windows*, SG24-7363

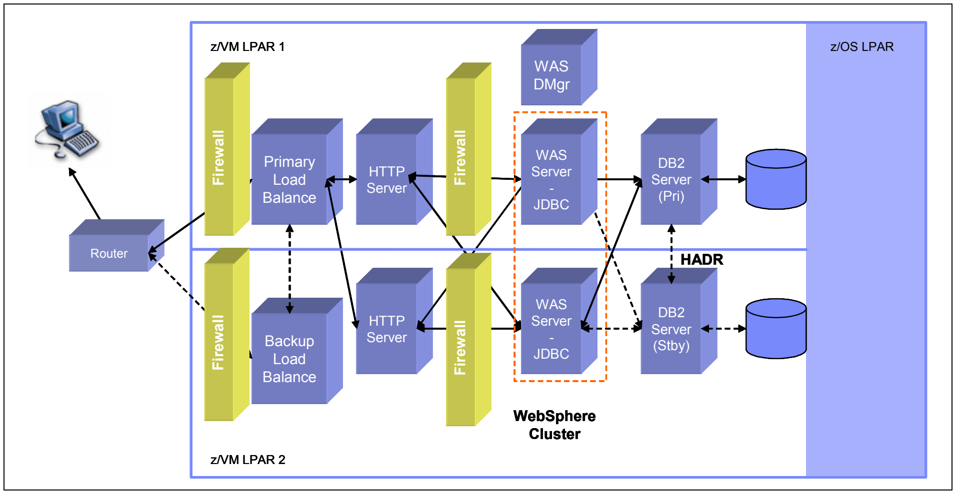


Figure 13-6 Active/active WAS Cluster and DB2 HADR

13.5.6 Active/Active WAS Cluster with Database Sharing

In Figure 13-8 database sharing was introduced using Oracle Real Application Clusters (RAC). Oracle RAC provides high availability for applications by having multiple RAC nodes sharing a single copy of the data. Should a cluster node fail 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 on Oracle RAC go to `http://www.oracle.com`

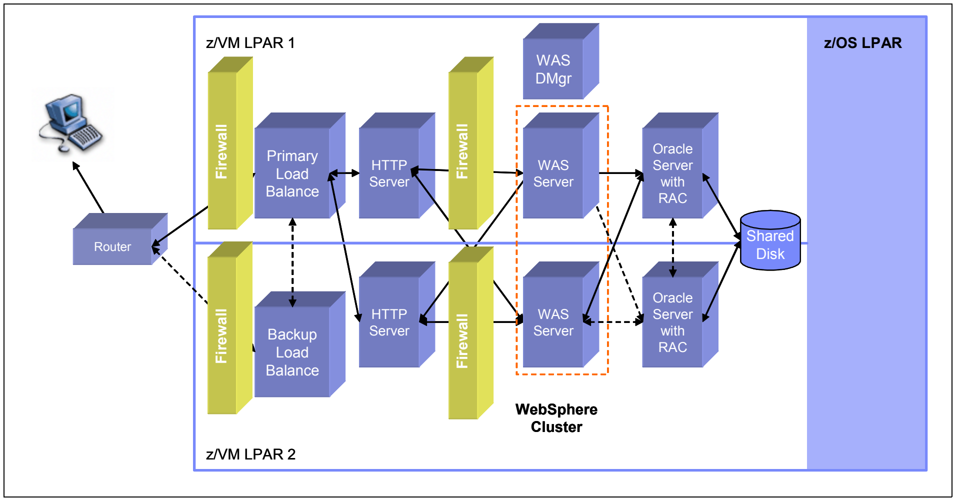


Figure 13-7 Database Sharing Using Oracle RAC

13.5.7 Active/active WAS Cluster with DB2 Sharing in z/OS Parallel Sysplex

In Figure 13-9 we introduce the additional benefits provided by the z/OS Parallel Sysplex®. Briefly, a Parallel Sysplex is a high availability configuration designed to provide continuous availability of systems and application. 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 centralised arbitrator known as the Coupling Facility.

Each WAS 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 co-operatively 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 see: *Clustering Solutions Overview: Parallel Sysplex and Other Platforms*, REDP-4072

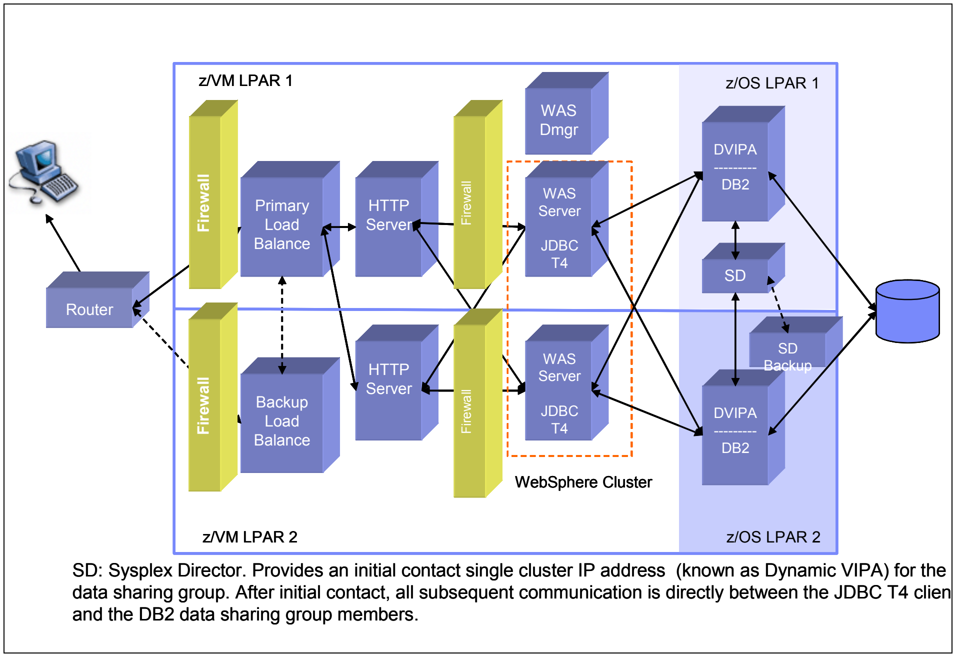


Figure 13-8 Database sharing using z/OS Parallel Sysplex

13.5.8 Active/Active WAS 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 WAS 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 storage subsystem(s) 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:

- Monitors the Parallel Sysplex cluster, coupling facilities, and storage subsystems and maintains GDPS status.
- 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).
- 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.
- Works with Tivoli System Automation Multiplatform across z/VM and Linux to understand their state and coordinate their restart during the recovery phase.
- Invokes network switching, based on user-defined automation scripts

[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.

In Figure 13-10 Site A and Site B are in a GDPS and share the same DB2 data. GDPS helps automate recovery procedures for planned and unplanned outages to provide near-continuous availability and disaster recovery capability.

.

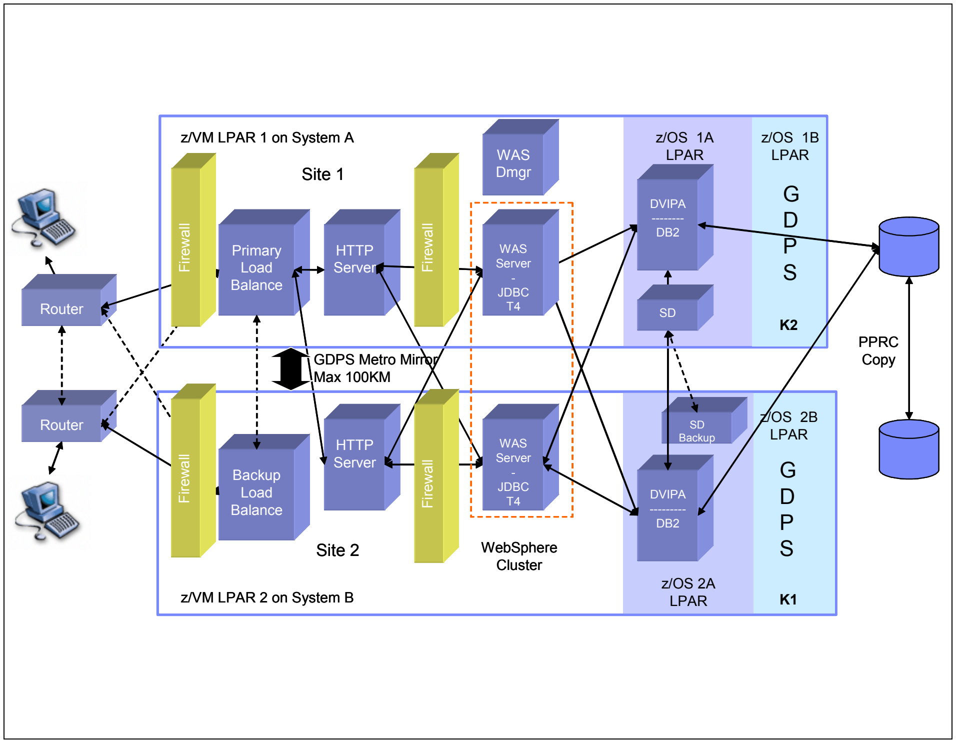


Figure 13-9 GDPS Solution for near-continuous availability

Distances greater than 100km requires 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 hence it is not synchronized with the primary copy. Further discussion of this topic is outside the scope of this manual.

Note: For more information about PPRC and other disk copy services see *DS8000 Copy Services for IBM System z*, SG24-6787.

13.6 Linux-HA Project

The Linux-HA project provides high availability solutions for Linux through an open development community. The majority of Linux-HA software is licensed

under the Free Software Foundation's GNU Public License (GPL) and the Free Software Foundation's GNU Lesser eneral 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/slave) 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 please see : *Achieving High Availability on Linux for System z with Linux-HA Release 2*, SG24-7711

13.7 Understanding the Availability Requirements of Your Applications

13.7.1 Service Level Agreements

In order to determine the availability requirements of applications you are wanting to migrate to Linux on System z you must take into account the needs of the business units who rely on these applications. A best case would be that there are service level agreements (SLA) in place that state requirements such as availability needs, response time, maximum system utilization, DR requirements etc. This should be the basis for the design of the target system on Linux.

If service level agreements do not exist then the first thing you need to do, before you startto design the solution, is to 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 of a service level agreement 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 and/or cover:

- 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 etc..
  - Definition of how availability is measured and who will do the measurement. For example: system availability, application availability, database availability, network availability etc..
- Minimum system response time
  - defined number and definition of where and how is it measured.

13.7.2 The Cost of Availability

As you have seen from the various examples in this chapter there is a great degree of difference in cost and complexity of the various availability options discussed. Providing continuous availability and a disaster recovery plan is not an insignificant expense but with the reliance on IT systems for almost all businesses this is an expense that can't be ignored.

If you have an web facing revenue generating application it is easy to calculate the cost of downtime by simply monitoring the average revenue generated over a period of time. This then gives you an idea of the lost revenue during an outage and how much you should spend to make the application more resiliant. Other businesses will have different ways of calculating the cost of downtime.

Don’t forget that for the success of any high availability configuration in a real disaster situation there needs to be a fully documented DR plan in place that is fully tested at least once per year.

Part 3

Hands on migration to Linux on System z

In this part of the book we will 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

There is also a chapter on technical procedures to introduce for better coexistance of z/VM and Linux.

14

MS Windows to Linux: WebSphere and DB2 Migration

This chapter describes a 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 customer as it could be fully tested without access to other systems or applications.

The aim of the migration 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 customer’s Intel server environment supported 15 Java applications running on an IBM WebSphere Application Server (WAS) 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.

As these applications were self-contained, they were chosen by the customer as an ideal and mission critical application that could be set up easily.

Note: Since the release of WebSphere Application Server Version 6, WAS has a common code base across all platforms. What this means is that migrating Java applications across platforms is much simpler and WebSphere skills are applicable across multiple platforms.

14.1.1 Existing Server (Source) Environment

The customer’s environment was over 60 rack mounted Intel servers running on the MS Windows 2003 Server. The Web Server was Microsoft’s 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. There were a very small amount of virtualized servers running under VMware ESX 2.5.

One of the reasons why the customer was contemplating a migration to Linux on System z was due to a number of performance issues related to their current set-up. 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 WAS applications during peak load.

The primary middleware running was WebSphere Application Server - Network Deployment V6.0.27 and DB2 Workgroup Edition Version 8.1.

In figure 14-1 we see a diagram of the existing production environment.

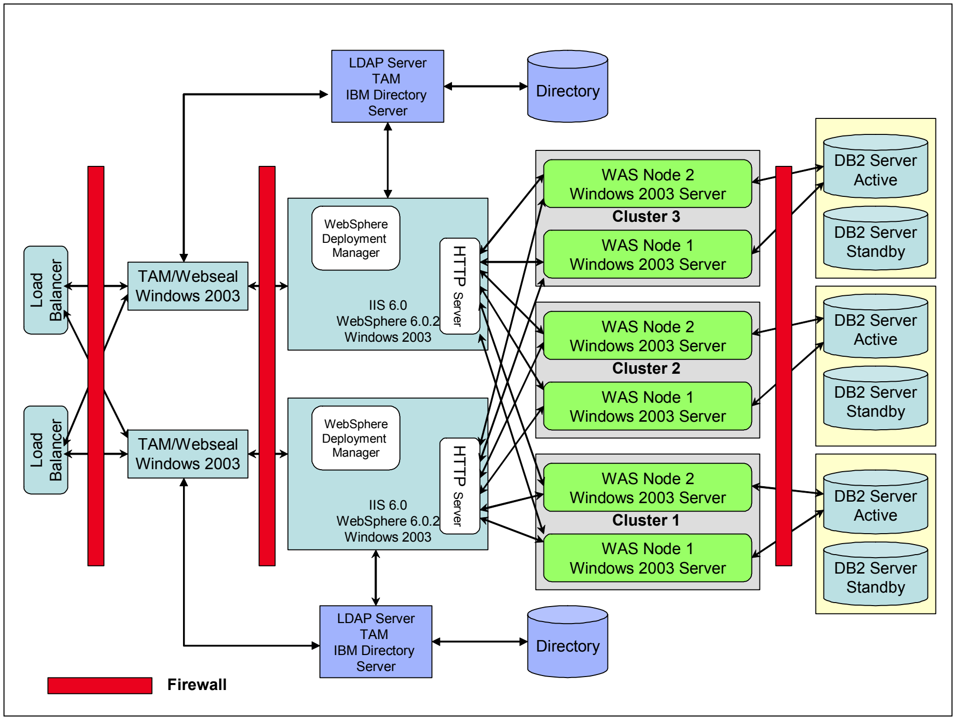


Figure 14-1 Customer WebSphere and DB2 Production Environment

14.1.2 Target Server Environment

As the customer wanted to consolidate their distributed MS Windows, WAS and DB2 environment to Linux on System z and as 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 DB2 data base 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 shows the simplified Linux, WAS and DB2 environment for the proof of concept.

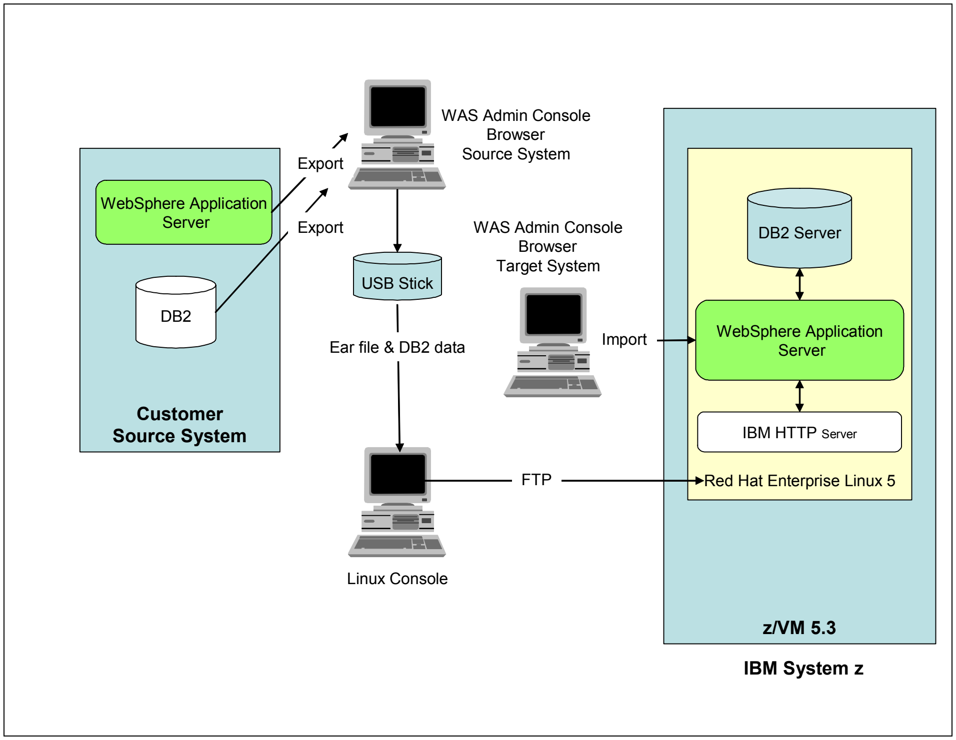


Figure 14-2 Linux proof of concept setup and application & data import/export

14.1.3 Tools Used in Proof of Concept

The source system was the customer's test and development system and was accessed by a customer owned MS Windows laptop. The target system was on a secure IBM network and was accessed by an IBM owned laptop. To move data between the two laptops a USB memory stick was used.

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 on remote access tools PuTTY and Filezilla see Appendix B, “Remote access applications” on page 249.

14.2 WebSphere Application Migration

The customer selected their 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 (WAS) Version 6.0.27 cluster.

The process of migrating from one WebSphere Application Server environment to another is quite straight forward. The application was migrated using the source system’s WebSphere Administrative Console’s **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 laptop 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’s **IMPORT** command.

Both export and import actions were completed with no problems.

14.3 DB2 database migration

IBM installed DB2 Enterprise Edition Version 8.1 on the RHEL5 system in the same virtual machine as WAS and the IBM HTTP Server. The tasks required to migrate the database from the MS Windows environment (source system) to the Linux on System z environment (target system) are as follows:

14.3.1 From the Source system:

The steps to migrate the database from the source server are as follows:

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. See 14.6.1, “Create Database Command” on page 177 for 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 target system. See 14.6.2, "DB2LOOK Output: Tablespace DDL Statements" for 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. ZIP all of the output from the **DB2MOVE command** in a single ZIP file. This 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 as extra characters will be added to the end of each line if transferred in ASCII. As 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 174 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 How to remove non-printable charactes from an ASCII transmitted file

```
sed 's/^M//g' INPUT_DDL.SQL > output.sql
```

To type the ^M character simultaneously hold down Ctrl and V keys while you type M. The command shown in Example 14-1stores the corrected file under the name *output.sql* in the current directory.

When the file created by the **DB2MOVE EXPORT** command has been transferred to Linux, unzip the contents.

For this migration, the DB2 database backup was appoximately 250 MB in size.

14.3.2 To the Target system:

The steps to migrate to the target system are:

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.See 14.6, "Commands & Output Messages" on page 177
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. Once 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. See example on previous page. Ensure 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
   ```

   The reason for this was that there were two large table spaces *ACE_BLOB_TS01* & *ACE_BLOB_TS02* , each of 320,000 pages or 10GB in size. This excedded the size of the file system. There are two ways to solve this problem, either increase the file system size or decrease the binary large object (BLOB) space allocation. We chose the latter and reduced the BLOBs from from 320,000 pages (10GB) to 32,000 pages (1GB) each. This reduced size had no impact on the proof of concept.
6. Once the DDL file has been successfully processed import the data into database using DB2MOVE LOAD command using the following command:

   ```
   db2 db2move ACEPRD01 load
   ```

For the ACEPRD01 functional test, the database load took 20 minutes.

14.3.3 Errors During Data Load

Some errors that we encountered and how we addressed them were:

- The **DB2LOOK** command on the MS Windows system did not extract the SYSTOOLS schema but the SYSTOOLS schema was exported by the **DB2MOVE EXPORT** command as 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
  ```
- The result was the successful creation and migration of the database ACEPRD01 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 of error messages received during the database load please refer to 14.6.3, "Output Messages" on page 179.

14.4 Testing Scenario & Results

Once 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 & /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 website.

In all cases the Linux system produced the same results as the production system. The only difference was that we could not save the quotes as it was not linked to the production Customer Management System.

The customer was very satisfied with the results of the test and this has reinforced their resolve to consolidate their MS Windows,WebSphere and DB2 environments to Linux on System z. The customer cut over their production to Linux on System z approximately 10 months after this test.

The net result for the customer was that 60 Intel servers were replaced with 3 IFLs on a IBM System z10 EC, giving the customer a more resilient system, with a much smaller footprint, both in floor space and power consumption as well as the potential for much 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 & 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, it is necessary to change all database locations from the MS Windows format to the Linux format. In our case we changed from:

`F:\db2data\’` to `‘/home/db2inst1/’`

for all location references

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

As the proof of concept was a success, the customer 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 the IT environment and the scalability and flexibility of the IBM System z10 solution.

Note: The existing IT environment of 6 WAS servers running in 3 clusters and 6 DB2 database servers was consolidated to 2 Linux virtual machines running WAS and 1 Linux virtual machine DB2 database server. This reulted in a significant simplification of the IT envionment with its resulting cost savings.

15

Technical Procedures

In this chapter we discuss some debugging tools that may come in handy as well as technical tips 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 to debug system problems during initial program load (IPL) or network configuration problems that prohibit adminsitrators from connecting to the Linux guest over the network.

The access to this console is only through 3270 console emulator terminal software (please see “3270 Emulation” on page 250 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 the VM userid is defined in the z/VM Directory. The line that starts with the USER string (02179) contains the VM userid (**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 bet 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 logon to the z/VM console within this Linux guest, you would just type the VM userid in the USERID field from a z/VM console *Welcome* screen as shown in Example 15-2 .

Example 15-2 z/VM Welcome Screen

```
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 screen and after the *ENTER* key is pressed, it will send the command to the Linux guest for processing. Remember that for security reasons the root and all other user password, when typed at a z/VM console, are not shown.

15.2 CP Commands

There are some tasks in the server configuration and/or during service trouble shooting where it is necessary to issue some z/VM CP commands to query the virtual devices or also 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 .

- Using a 3270 emulator.
- Using the *vmcp* Linux module.

3270 emulator

For sending CP commands from a 3270 emulator it is necessary to logon at the Linux virtual machine using the VM userid and issue the command in the command area of the 3270 terminal, but it must be issued using the **#CP** syntax, otherwise the command will be sent to the Linux TTY and will return a Linux error. Figure 15-1 shows the correct way to issue #CP commands. See more information about 3270 terminal at Appendix , "3270 Emulation" on page 250.

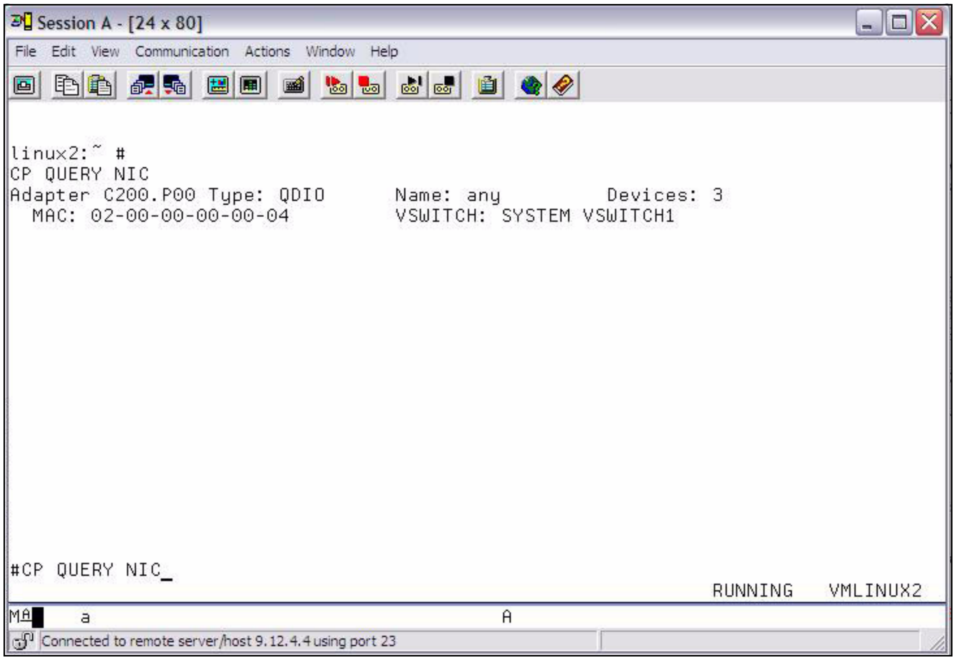


Figure 15-1 Example of CP QUERY command issue at 3270 terminal

vmcp module

Linux on System z runs a specific module called *vmcp* (virtual machine control program) and that provides a facility to issue CP commands from a Linux SSH session. This module must be loaded in order 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
```

Once the vmcp module is loaded, it is possible to issue the z/VM CP commands from the Linux session as shown in Example 15-4.

Example 15-4 Example of CP QUERY command issue 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 other useful configuration hints, please refer to the 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 of the strengths of Linux on System z is that outage times are reduced because almost all configuration changes can be done online. One example is the virtual CPU, where it is possible to add virtual CPUs to a Linux guest with all servers running.

There is some configuration that must be done in order to enable this facility.

1. Alter the default /etc/zipl.conf file configuration (see Example 15-5) to include the options that configure the kernel to allow that 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 the /etc/zipl.conf using vi or anotherr editor to include the additional_cpus=x parameter, where x is the maximum value of additional virtual CPUs. The new /etc/zipl.conf must look like the 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. Make the changes made in the configuration file /etc/zipl.conf take effect. To do this, issue the **zipl** command to write it on the IPL device and reboot the server as shown in Example 15-7 on page 187.

Example 15-7 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 . Example 15-8 shows the new server configurations avaliable in Linux . The results from the first command (`cat /proc/cmdline`) returns the information about the configuration when the kernel was initialized . The results from the second command (`ls /sys/devices/system/cpu/`) returns 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 the /sys/devices/system/cpu/ have a file named online that receives the value 0 for CPUs not actived and 1 for an activated CPU . To activate the CPU, first it must be enabled on the VM side with the commands shown in Example 15-9 on page 188.

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, the next step is to bring it online to the Linux server. The commands to do this is are 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 explained 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 Book*,SC24-6081-05 .

For more information about Linux on System z special applciation and devices drivers, refer to *Linux on System z Device Drivers, Features, and Commands*, SC33-8411-01

16

Example Sakai Migration

In this chapter, we migrate an open source learning management system, SAKAI, from a different operating environment to an target operating environment residing on Linux on System z. We follow the migration methodology discussed in “Migration Methodology” on page 37. The scope of this sample migration is to demonstrate how pre-planning helps in proper execution of any migration effort. Also this chapter gives an exact technical course of action in migration and methods used in solving issues in 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 is configured with three processors and three gigabyte of memory as shown in Example 16-1.

Example 16-1 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 shown 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 is 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
```

Memory Information is shown in Example 16-4:

Example 16-4 Linux on System Memory information

```
test19:~ # cat /proc/meminfo
MemTotal:      2047784 kB
```

The operating system is SLES 10 SP2 (see 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: The scope of this chapter is to discuss 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. This section does not discuss the database installation or the database migration.

16.2 What is SAKAI?

Sakai is open source based, learning management system (LMS) or course management system (CMS), 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 and can be applied in both a teaching and learning environment as well as 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 yet flexible solution that supports not only teaching and learning but also research and administrative collaboration.

16.3 Why SAKAI for Linux on System z

Choosing Linux on System z brings the following advantages:

- Able (scalable, available, reliable)
- Mature Virtualization Platform
- Real time integration of transactions, information & analytics
- Business-driven services
- Service oriented
- Centralized control & management
- Consolidation of hundreds to thousands of diverse workloads
- Most efficient cost per computing power

16.4 Migration Planning

We assume that the stakeholders are 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 kind of dependencies and prerequisties that need to be taken into account.

16.4.1 SAKAI application analysis

We started with a thorough review of the system pre-quisties for SAKAI from the SAKAI project website. 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
- Since Linux on System z doesnt support the Sun Java™ Platform, we anticipated some issues in the source codes referencing Sun JAVA specific classes or features.
- SAKAI requires an open-source compilation tool called Maven
- There were some dependencies on 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 Checklist

We referred to the the Chapter 4, “Migration Planning checklist” on page 45 and started to fill out the application analysis worksheet as shown in Figure 16-1 on page 193.

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

Figure 16-1 Application Planning Checklist

From the checklist, it is evident that most of the tools used for SAKAI Migration are also avaiable on Linux on System z . We also need to note that there is a slight variation in the Java (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 doesn't depend on any specific Java environment. The migration effort can be slightly greater if the application is written more specific to one particular Java environment. Keeping this in mind, we decided to go with doing a proof of concept before doing the actual migration.

16.5 Proof of Concept

As discussed in Chapter 3.1.5, "Pilot Proof of Concept" on page 41, The main objective of this proof of concept is to focus on the identified areas of risk and document these risks as discussed in Chapter 16.4.1, "SAKAI application analysis" on page 192 . Additionally, recommend approaches to mitigate the risks prove that the full migration can be completed successfully.

We were looking at the ideal SAKAI component or module that can be tested on the Linux on System z operating environment. The SAKAI community has the Sakai Demo package which is a pre-built version of Sakai with Apache Tomcat and other simple configurations. This demo package can be a good candidate for our proof of concept (PoC). So we decieded to implement the SAKAI Demo package.

> **Tip:** To test SAKAI on a new operating environment (OS and Hardware), the SAKAI demo package is an ideal candidate. It can be downloaded from the following location `http://sakaiproject.org/portal/site/sakai-downloads`

Since all the programs and databases are already packaged with the demo, we downloaded and just decompressed the SAKAI demo file (see Example 16-6).

Example 16-6 Uncompressing SAKAI Demo

```
test19:/root2/sakai-demo-2.5.4 # tar -zxvf sakai-demo-2.5.4.tar.gz
```

We followed the documentation on 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, this is 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 **"start-sakai.sh"** wrapper script written around the actual Tomcat startup script. This would set 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 and we were able to access the SAKAI portal by accessing a URL that looked like:

http:// <hostname>:8080/portal

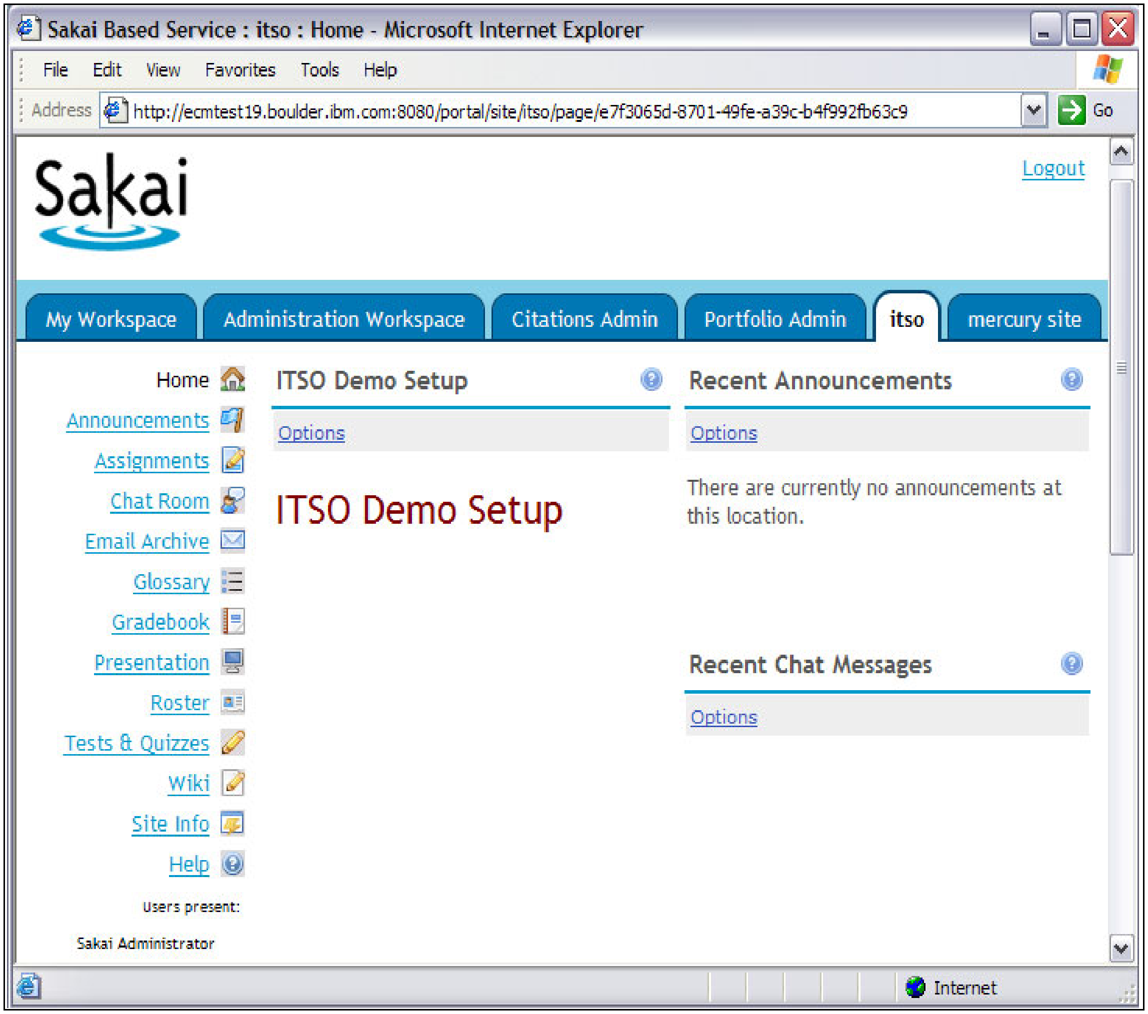


Figure 16-2 Sakai Demo portal

We analyzed the logs of SAKAI in order to check that everything was working correctly. We verified the various learning management tools that come packaged with the SAKAI demo and at the same time, did a parallel analysis of the SAKAI application logs. While performing a verification of a partiuclar tool, the “Gradebook”, we encountered a lot of errors.

Having a closer look at the error, our Java developers were saying that these errors were occuring because the SAKAI Demo package would have been compiled in a SUN Java Platform and we are trying to execute in an IBM based JAVA (VM) environment.

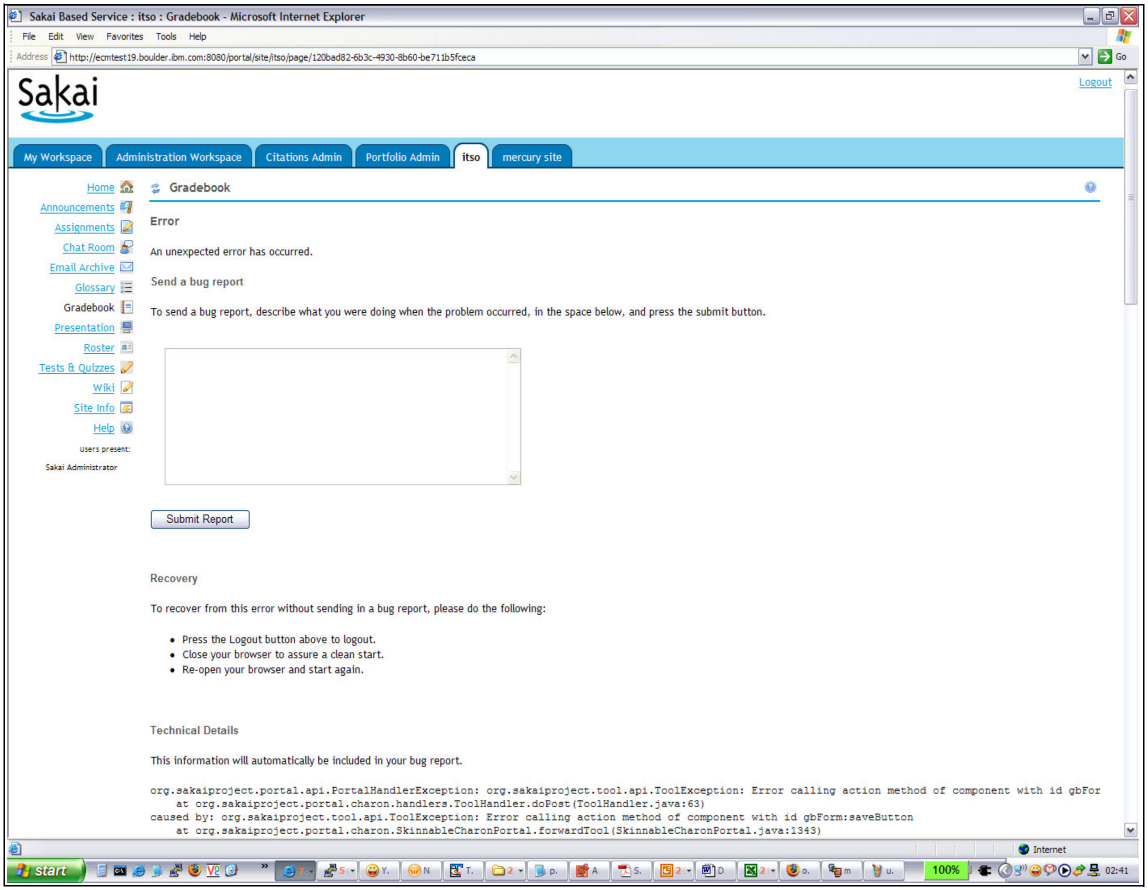


Figure 16-3 Encountered SAKAI Demo Gradebook Issue

16.5.1 Findings during the Proof of Concept

Some of our observations during this proof of concept:

- The SAKAI demo package was implemented easily without any operating environment changes (expect the Java Platform). The Proof of Concept was swift and easy.

- Most of the tools packged in the SAKAI Demo were working prefectly, which means the tools are not developed for any specific Java platform or environment.
- There might be few changes in the code 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 the chapter “SAKAI application analysis” on page 192. And it is also evident from the proof of concept that the actual migration can be done with minimum code and operating environment changes.

16.6 Actual Migration

Since we had a feel of the SAKAI application during the proof of concept, we started with downloading the compressed SAKAI Source from the Sakai website. Referring to our “Application Checklist” on page 193, 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 specific to a variety of operating environments. We recommend using this documentation that is available from the SAKAI website.

We downloaded IBM Java 1.5 (64 bit) version from the IBM site as per our application checklist and started our installation (Example 16-10).

Example 16-10 Installing IBM Java 1.5

```
test19:~ # rpm -ivh java5_zlinux_s390x.rpm
```

Then the Java environmental variables are set so that they are all pointing to correct Java installation directories(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 Tomcat Installation

Apache Tomcat provides an ideal environment for running Sakai as a web application. Tomcat implements/conforms to both the Java Servlet and JavaServer™ Pages (JSP™) specifications and can be run in standalone 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 the Tomcat version 5.5.26. Tomcat can be downloaded as an binary install archive from the following website `http://archive.apache.org/dist/tomcat/tomcat-5/`.

Unpack the Tomcat archive into your installation directory of choice, e.g. /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 (e.g. /opt/apache-tomcat-5.5.26) is referred to as $CATALINA_HOME. For convenience, we can set $CATALINA_HOME as an environment variable.

Example 16-13 Setting Tomcat environmental variable.

```
export CATALINA_HOME=/opt/tomcat
export PATH=$PATH:$CATALINA_HOME/bin
```

Note: Since most of the environmental variables has to be set up before the tools are accessed, it is advisable for one to typically modify startup files like ~/.bash_login ot .profile to set and export shell variables.

Since Tomcat will be loading the SAKAI application, it is recommened to 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.

Once you set JAVA_OPTS Tomcat will see this environment variable upon startup and use it.

Example 16-14 export java options

```
export JAVA_OPTS="-server -XX:+UseParallelGC -Xmx768m
-XX:MaxPermSize=160m -Djava.awt.headless=true"
```

The above examples instructs the VM to use the server HotSpot Java Virtual Machine (JVM) and then enables parallel garbage collection **"+UseParallelGC"**, Java 5.0 (1.5) defines a class of machines referred to as server-class machines. These are machines that have 2 or more physical processors and 2 or more GB of physical memory. Also adequate heap size, and sizes the permanent generation to accommodate more longer-persisting objects are also set.

16.6.2 Maven Installation

Apache-Maven is a popular open source build tool for Java projects, designed to make Java build process easy. SAKAI uses Maven build tool to compile and deploy its project modules. Maven can be downloaded from the following website http://maven.apache.org/download.html.

Unpack the downloaded apache-maven distribution archive into your directory of choice, e.g. /opt/apache-maven-2.1.0

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, we should add a MAVEN_OPTS environment variable as defined below.

Example 16-16 exporting Maven options

```
export MAVEN_OPTS='-Xms256m -Xmx512m -XX:PermSize=64m
-XX:MaxPermSize=128m'
```

Maven Setup

Create a local Maven repository (.m2) in your home directory.

Example 16-17 Creating 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 which 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`.

Example 16-18 Created settings.xml under

```
<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>
```

Now we verify that we can start Maven, execute the command mvn --version from command prompt. This should start Maven and cause it to report its version.

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 have Maven environment ready, We downloaded the latest SAKAI source code from the Sakai website and unpacked it in to a directory of choice.

Example 16-20 Decompressing the source code

```
test19:/opt/manoj # tar -zxvf sakai-src-2.5.4.tgz
```

16.7 Build Process

To build SAKAI from the source code, we need 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:

1. **`clean`** flushes the target for each Sakai project module prior to building
2. **`install`** installs the output of each Sakai project module into the repository
3. **`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 etc. 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 acts as a parent for other pom.xml files included in other Sakai sub-project modules.

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
```

For the first time during the build process, Maven will download any missing dependencies into your local repository, then attempt to compile the Sakai code . This process would take few minutes, since all the missing dependcies for the compilation would be downloaded.

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

During the intial building process we encountered a build failure due to error relating to an referenced symbol in a particular Java source file.

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 found out that there were references to Sun Java specific classes in the source code. Since we use IBM Java, these classes would not be accesible.

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;
```

Upon searching the internet, we found out that the same issue has been witnessed already and a patch was released by the Sakai community. So we downloaded the patch from the Sakai website and appliedit 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 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 is solved and the building process continued, but 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 it's own implementation.

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/.
```

As per one of the internet posts, test failures are happening because of jcr unit testing which has a xml namespace issue. It is recommended to skip the tests during the initial build.

So we started 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 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 can now deploy the Sakai application using Maven. In this case we had Maven conduct the unit tests when it deploys the application. We also need to specify the location where Maven has to deploy the application. Usually it would be Tomcat's base directory. The Maven command line parameter **-Dmaven.tomcat.home** specifies Tomcat's location and can be omitted if Tomcat home is specified in Maven's settings.xml file.

Example 16-28 Re-building 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 has completed, we can start Tomcat which in turn would load the deployed Sakai application (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 (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://<ip address>:8080/portal

Figure 16-4 on page 207 shows our successful load of the Sakai system on Linux on System z.

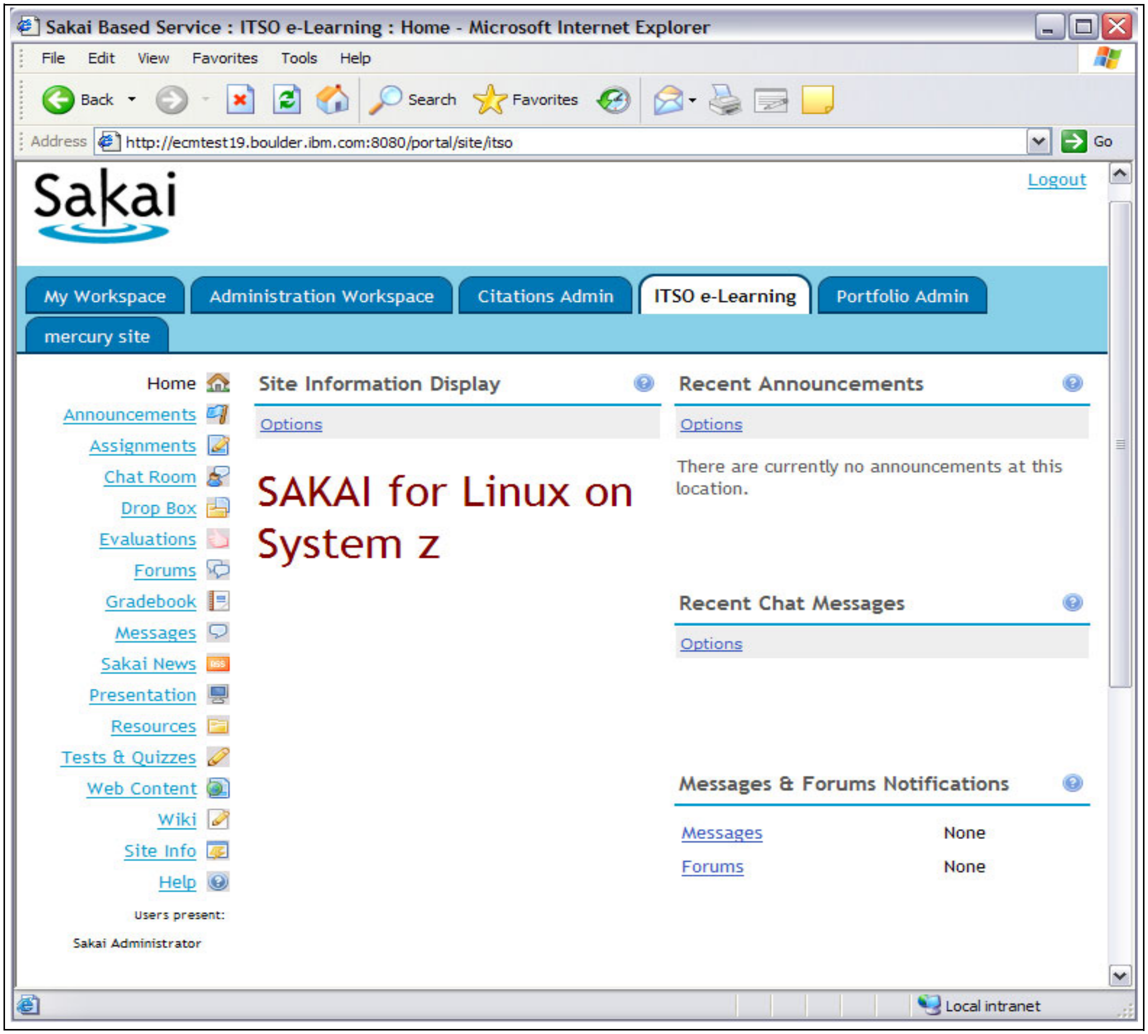


Figure 16-4 Sakai for Linux on System z

16.8 Verification Testing

As we discussed in the "Verification Testing" on page 43, this is to provide objective evidence that the testing requirements are met on the target operating environment.

First, we need 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 did extensive testing to ensure the proper functioning of all the Sakai application modules.

As 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 the 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 identifed issues with Java classes that are not common across different Java implementations.

17

Open Source: A Media-Wiki Migration

This chapter will show a migration of the open source application MediaWiki from two different souce systems to the target system residing on Linux on System z. This is an example where an incomplete analysis was done of the requirements of the source and target system environments. Even though this is a small and comparatively simple system to migrate, we expect to see problems in the migration due to the lack of planning.

In the first section, we will create the target system environment. We will describe the installation of MediaWiki on a z/VM guest running Linux on System z. Depending on the distribution you use, you can reference the IBM manual “Setting up a Web 2.0 stack on SUSE Linux Enterprise Server 10 SP2” or “Setting up a Web 2.0 stack on Red Hat Enterprise Linux 5.2” to install the components of the Web 2.0 stack. We will then show how to install MediaWiki and finally, how to migrate data from MediaWiki from an external platform to MediaWiki on System z.

The manuals listed above use the command line interface to perform the installation of the Web 2.0 stack. That method is a completly acceptable way to install software packages. We will show how to use the SLES installation tool Yast to install the Web 2.0 stack. We will only install those components required for the installation and use of MediaWiki.

17.1 YaST Setup of Installation Source

YaST requires that you first configure where the installation source is and how you are going to access it. The figure shown below is the first screen after invoking YaST from a PuTTY session. See Figure 17-1

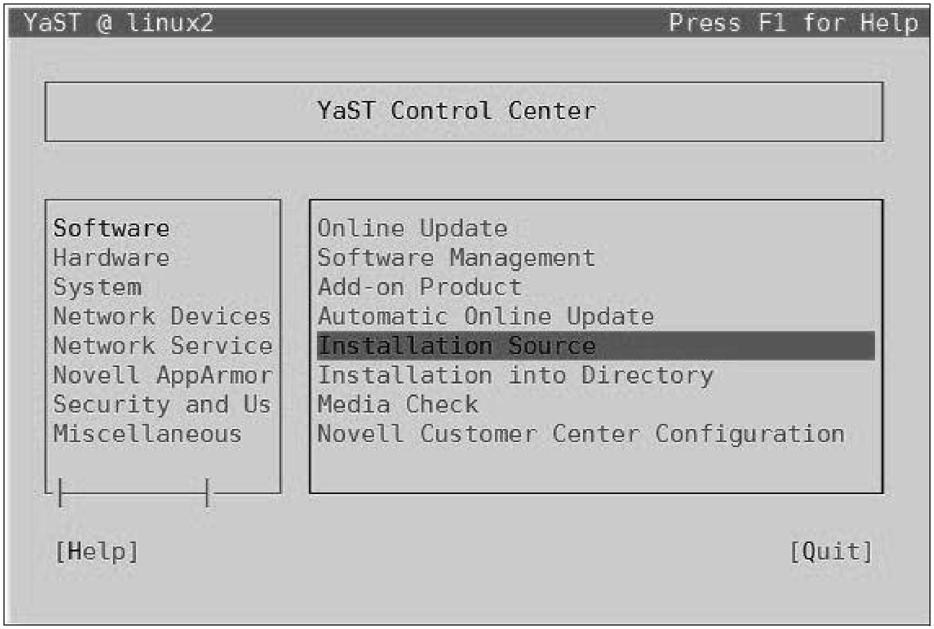


Figure 17-1 First panel to configure Installation Source

Use the Tab key to position highlighted text to the box on the right. (The Tab key will contine to circle throught the possible selections by highlighting them.) Use the arrow keys(up or down) to position highlighted text within the box on the right. Highlight “Installation Source” and press enter to get the screen shown in the Figure 17-2 on page 211.

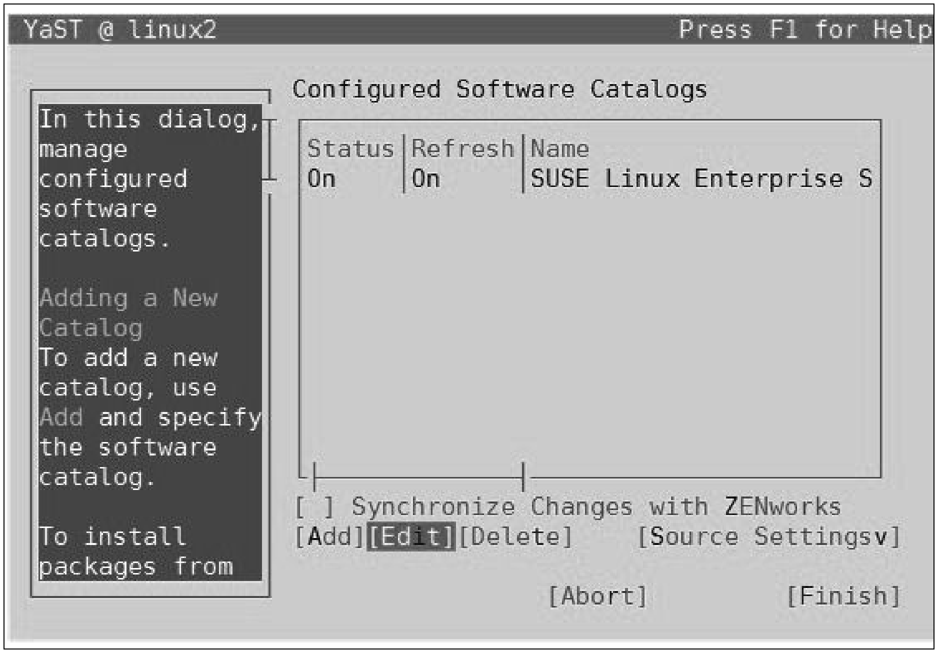


Figure 17-2 Select add or edit installation source

Use the tab key to position over the check box for "Synchronize Changes with ZENworks". Press the tab key to de-select this option. The x should disappear from the box after pressing the space bar. Use the tab key to select Add or Edit and press enter. The screen in Figure 17-3 on page 212 will be shown.

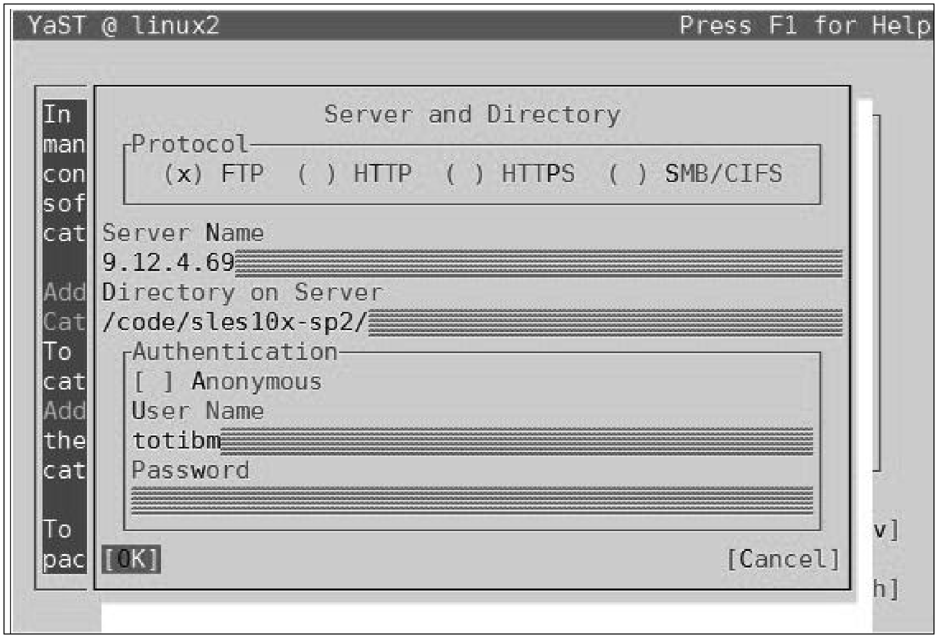


Figure 17-3 Software catalog server and directory

Use the tab key to position over the protocol you will use and press the space bar to select that option. An x will appear in the box you select. FTP is a common method to access distribution files. That is the protocol we have selected. Tab to the 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 to begin with a slash (/) and end with a slash (/). 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 Install the LAMP Stack

To begin the installation of additional sofware with YaST, return to the initial screen and tab to select “Software Management”. That screen is shown in Figure 17-4 on page 213.

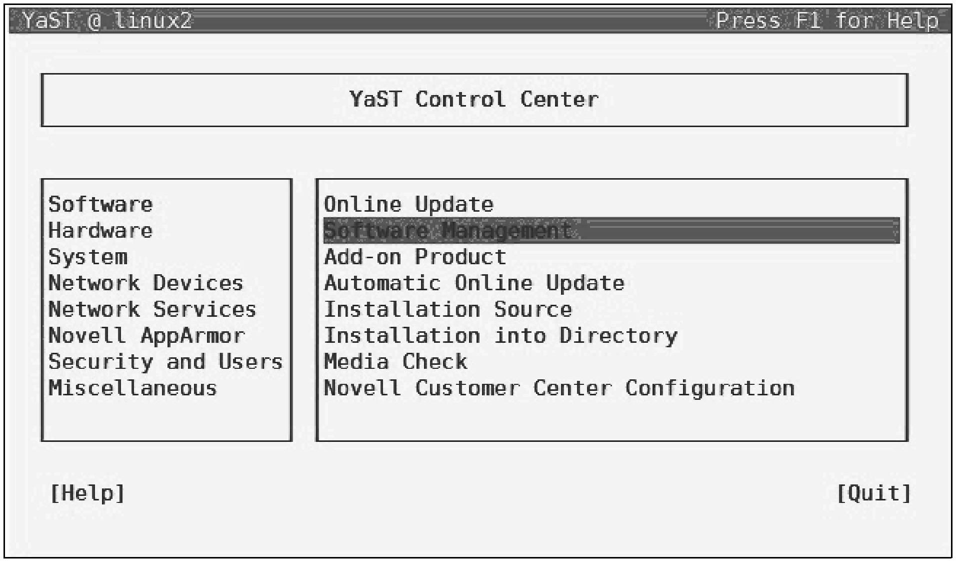


Figure 17-4 First panel to install software packages (Black and white image)

After Software Management is highlighted, press enter. The screen shown in Figure 17-5 on page 213 will be displayed.

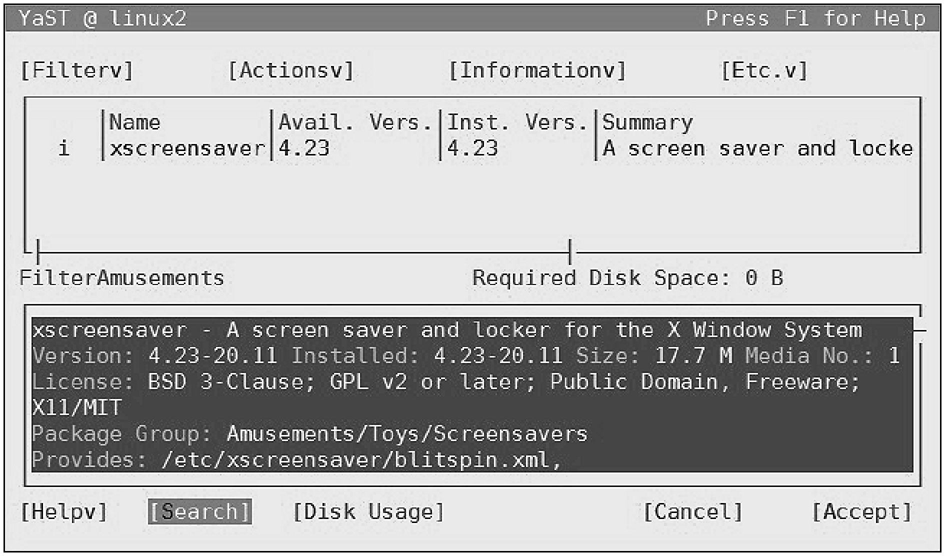


Figure 17-5 Go to the search panel

Use the tab key to highlight the search option and hit enter. Figure 17-6 on page 214 is now displayed.

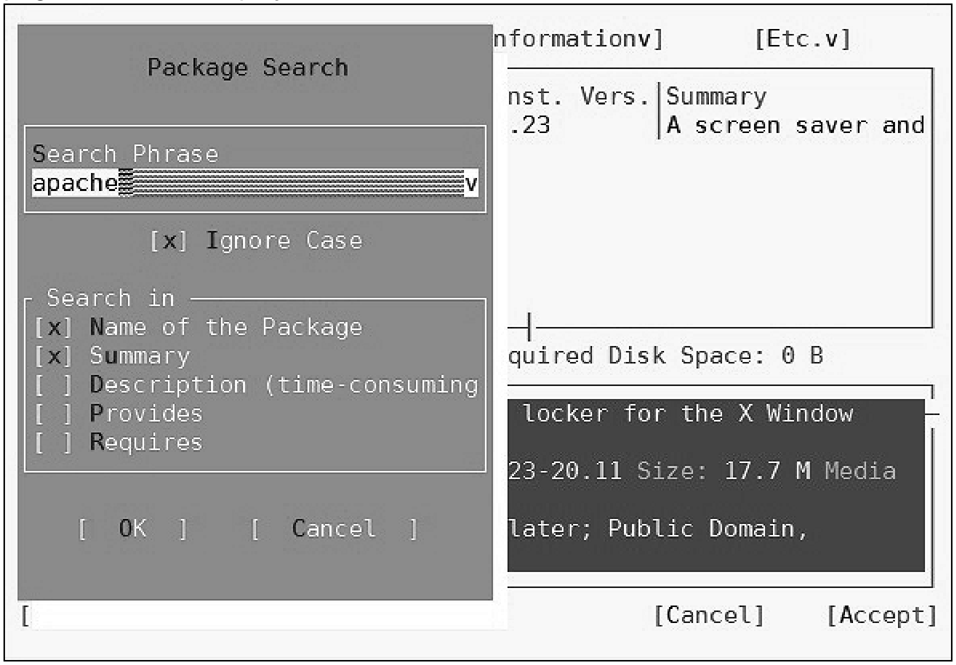


Figure 17-6 Search for sofware packages by name

The Search Phrase box should be highlighted. If not, use the tab key to select it, Type *apache* in the Search Phrase box. Leave "Ignore Case" checked. Also leave "Name" and "Summary" checked. The other boxes can be selected at your discretion for more information. Tab down to OK, and press enter to the screen shown in Figure 17-7 on page 215.

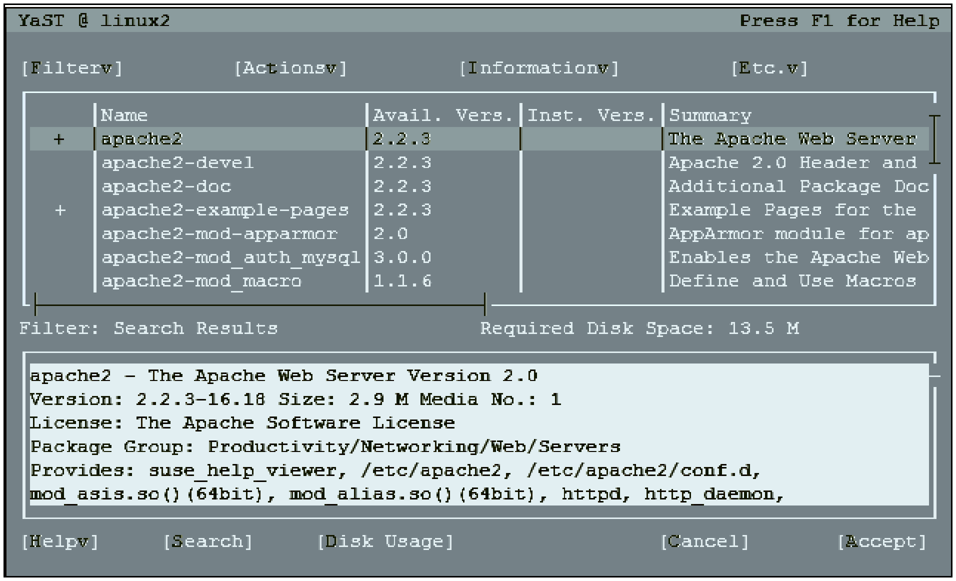


Figure 17-7 Select software packages to install

A list of all Apache packages are displayed. Highlight the packages you want and press enter to select it. A plus sign (+) will appear next to the selected packages. If there is a dependency for that package a plus sign will automaticly appear next to it also. 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 and use the space bar to select packages that you desire to install. Finally, tab down to highlight Accept and press enter to get to Figure 17-8 on page 216.

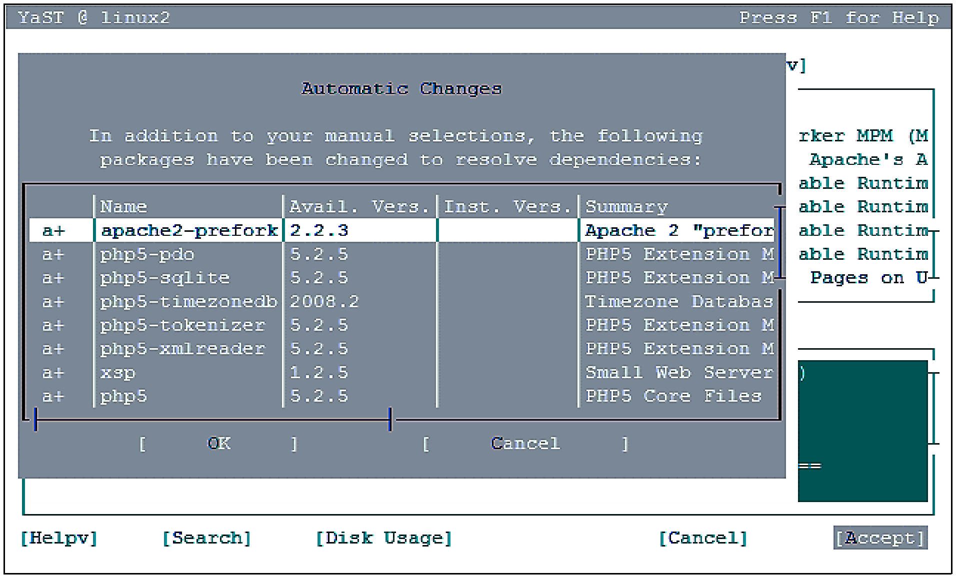


Figure 17-8 Additional requisite software packages selected

This screen lists all the additional dependencies outside of apache2 that will be installed. Tab down to highlight OK and hit 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 217 will be shown.

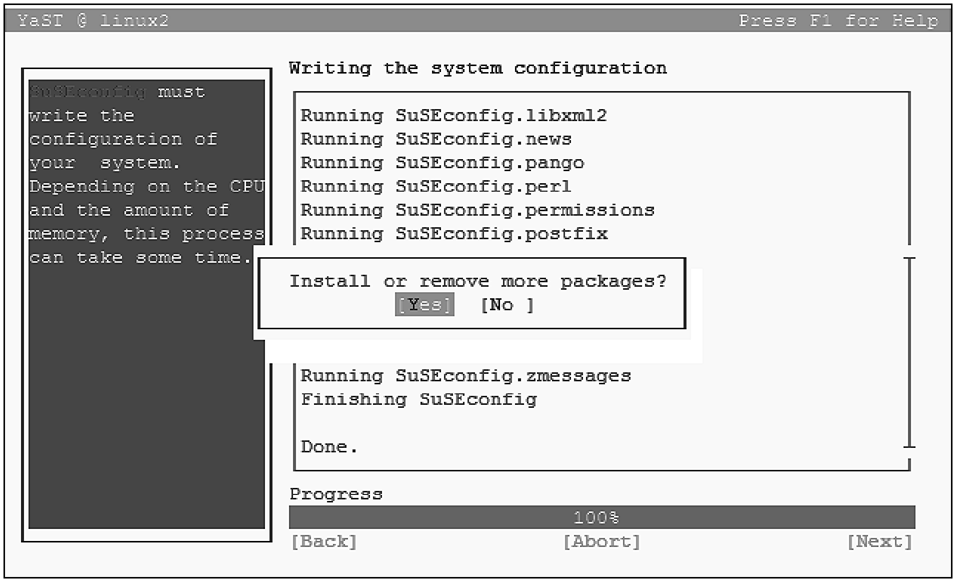


Figure 17-9 Continue to install software packages

Since we want to install PHP5 and Mysql next, we will leave *yes* highlighed and press enter. Tab down to highlght *search* and hit enter. You will be back at the panel to search for more packages.

In the search box type *mysql*, tab down to OK and hit enter. At the minimum, select with the space bar, *mysql, mysql-client*, and *php5-mysql*. Select “Accept” and press enter. You should get some more 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. You will then be asked if you want to install more software. Leave *Yes* highlighted and press enter.

Tab down to *Search* and hit enter. In the search box enter *php*, tab down to OK and hit enter. Notice in the results that you have ‘i‘ in the left hand column for many of the packages. This means that the package is already installed. In addition to the packages already installed, we selected *php5-devel* and *php5-pear*. Tab to *Accept* and hit enter. Tab to *OK* and hit enter. Software will be installed and again you have the choice to install 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 Start and Test LAMP Components

Before the installation of MediaWiki software we will configure and test Apache PHP and MySQL.

17.3.1 Apache test

You can display the version of Apache2 with the following command:

apache2ctl -v.

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 seen inFigure 17-10 on page 219. Your Linux on System z IP address or hostname is typed in as a URL like this:

```
http://9.12.2.91
```

or

```
http://your.server.name/.
```

Port 80 is the default and does not have to be specified.

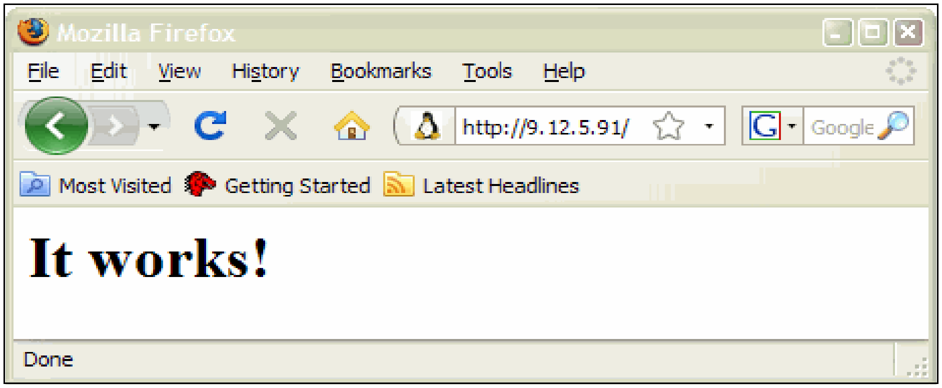


Figure 17-10 Successful test of Apache2 install

17.3.2 Verify 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 in this 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 informaiton as in this Example 17-3

Example 17-3 Finding the DocumentRoot on Red Hat

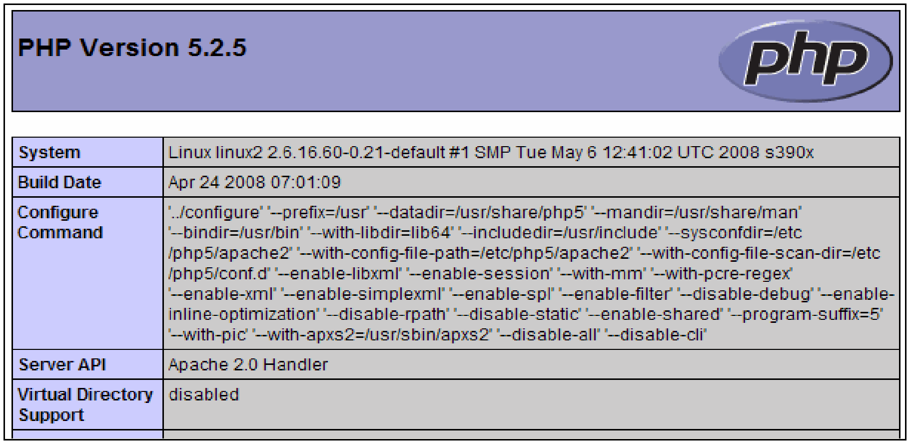
```
[root@linux6|~] grep 'DocumentRoot "' /etc/httpd/conf/httpd.conf
DocumentRoot "/var/www/html"
```

Then we can create a one line PHP script that will allow us to see all the PHP information about the install. So in /srv/www/htdocs in our example system we created the program phpinfo.php which consisted of this Example 17-4

Example 17-4 the phpinfo.php program

```
<?php phpinfo(); ?>
```

Then all we need to do is to run the php program by useing a web browser and viewing the phpinfo.php location as shown in Figure 17-11 on page 220.



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 By looking at the phpinfo.php file we can view the php configuration

17.3.3 MySQL Configuration

Display the version of MySQL with the following command from a PuTTY session: **mysql -V**. This will verify the base installation of MySQL. There are two additional configuration steps for MySQL before we can install MediaWiki.

1. The first step is to 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 Figure 17-12 on page 220.

```
cp /usr/share/mysql/my-medium.cnf /etc/my.cnf
chown root:root /etc/my.cnf
chmod 644 /etc/my.cnf
```

Figure 17-12 Configure MySQL Configuration File

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 Figure 17-13 on page 220 to set the password:

```
mysqladmin -u root password ‘yourpassword’
```

Figure 17-13 Set password for MySQL

17.4 Installation of MediaWiki software

Follow these steps to install MediaWiki on your Linux server running on System z:

1. Download the MediaWiki sofware as a TAR file from the following site: http://www.mediawiki.org/wiki/Download
2. FTP the the compressed TAR file to your Linux server.You can temporarily put it in a /tmp or your /home directory, later to be moved 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 . Simplify the name of the directory with the mv command. 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. Now 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 the components are available. You will get a report of the results of this check. You should get a web page with the following line included(usually in green):

```
Environment checked.
```

Below this message, there will be boxes and radio buttons to configure MediaWiki. Most of the options are your local names. At the point where information about the superuser account is requested, check the box, leave the userid as root and supply the MySQL password you set during the MySQL configuration steps in Chapter 17.3.3, “MySQL Configuration” on page 220, step 2.

After all the information is supplied, click on the “Install MediaWiki!” button at the bottom of the page. Fix any errors that are returned and click on the “Install MediaWiki!” button again. A successful install will show your environment settings and instructs 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. It 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 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 on how to include your own logo on your wiki. The home page of our wiki now looks like Figure 17-14 on page 222:

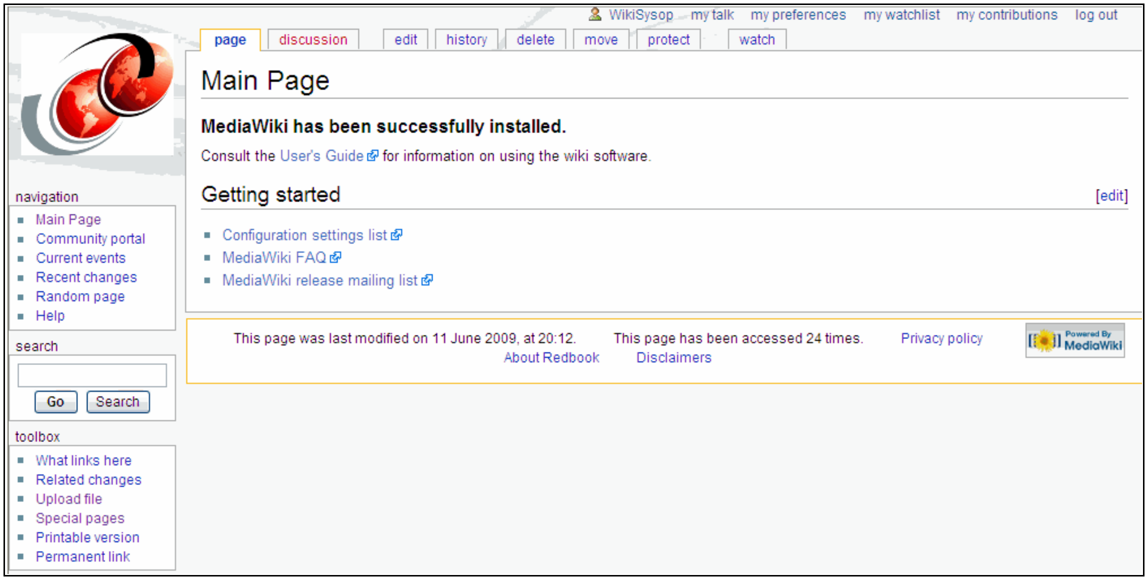


Figure 17-14 Home page with addition of Redbook logo

17.5 Export MediaWiki Pages Pilot

It would be ideal if you have a small part of the application that can serve as a pilot project for the migration, You can use the pilot project to learn a great deal about the readiness of the environment and your organization for the main migration project.

Our objective for the pilot is 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 uilities we used to dump the database are supplied by the application vendor. We attempted a full XML formated dump of the contents of the database to a large USB drive with 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 is clear that the source system has security requirements for the migration that were not discovered during the planning stage. The problem has been discovered during a pilot project so the impact to the overall migration project is limited.

A search of the vendor's documentation website was made to research the cause of the error. The following reference was found that 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 Import 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. See the chapter Remote Access Applications in this Redbook 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. The command used was:

```
php importDump.php < /tmp/louis/dump.xml
```

This import utility completed successfully. So now we were ready to test the success of the migrated data. Since the source data base had names and pictures of the contributors to the wiki, it was decided that the first test case was to search and display one of the contributors. The following is the results of that test.

We expected to see a picture of the wiki administrator. But no picture was displayed. Where the picture was supposed to display, there was the name of a file. SeeFigure 17-15 for the relevant portion of the web page that shows the file name and not the picture.

Figure 17-15 Failure to display picure of WikiSysop

We needed to question the wiki administrator to find out 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 data base itself. If you just migrate the main database you will not get the images. This was another failure in analysis and planning. All the supporting files for the application were not discovered and documented in early stages of the migration project. Again, since this is a simple project and we are only in the pilot stage of the migration, the implementation of a resolution will not be too time consuming. However, it could be envisioned that if it was a complex project with tight schedules, a mistake of this magnitude could jeopardize the success of the entire project. The importance of analysis and planning can not be over-estimated.

We returned to the source server and offloaded the /images file to a USB device. We transfered the /images file to the /images directory on the Linux on System z target server. Executing the search for the wiki adminstrator resulted in the images being displayed as expected.

17.7 Migration of Wiktionary Website

We have successfully installed the MediaWiki application on Linux on System z. We have also migrated the contents of a small data base to Linux on System z.

This has demonstrated that many organizations could easily benefit from using MediaWiki as a knowlege management system running on System z. Knowlege 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 hadware on System z is an ideal solution.

The small data base that we converted did not have all of the functions and facilities of the the MediaWiki software suite. We want to have a more complex and comprehensive test of Mediawiki running on Linux on System z. The first thought was to migrate all of Wikipedia to Linux on System z. Without a doubt, Wikipedia is the most popular application running on MediaWiki. We soon 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. We have no bias and make no judgement on the contents of Wiktionary. The data base for Wiktionary was available and suitable for our migration example.

17.8 Acquire English Wiktionary XML Dump

Wikimedia has a project that coordinates public dumps of wiki content. The sites listed in Figure 17-16 discuss the project, its use, limitations, tools, etc.

```
How to get the data: http://meta.wikimedia.org/wiki/Data_dumps
Where to get the data: http://download.wikimedia.org
Wikimedia servers: http://meta.wikimedia.org/wiki/Wikimedia_servers
```

Figure 17-16 Web sites for data base dumps

We followed the documentation to find the latest dump of the English Wiktionary. It was a simple process to download the compressed dump that 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 Import Wiktionary XML Dump

We then decompressed the file and started the import of the XML file. You can monitor the process of the import with the tail command. Those three steps are shown in Figure 17-17.

```
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
```

Figure 17-17 Import XML dump of database

The import procedure was done through an interpretive scripting process and no special tuning was for the data base 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 Wiktionary Test and Remediation

Our first test of the IBM Redbooks wiki with the Wiktionary data base uploaded had the result shown in Figure 17-18. Only one section of the screen is shown so that the problem is clearly displayed.

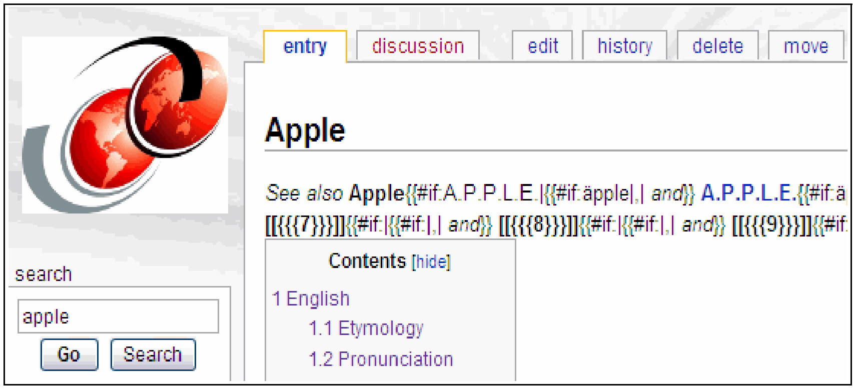


Figure 17-18 Import Error

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.

We knew that advanced users of MediaWiki software use functions that are installed through a facility called extenstions. 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 1) open source software is highly portable and 2) Linux on System z is a standard distribution, SLES10 in this case.

We downloaded the extension from the MediaWiki website 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 apple, pear and other favorite fruits were correct. We also used the IBM Redbooks Wiktionary while writing the documentation for this IBM Redbooks publication.

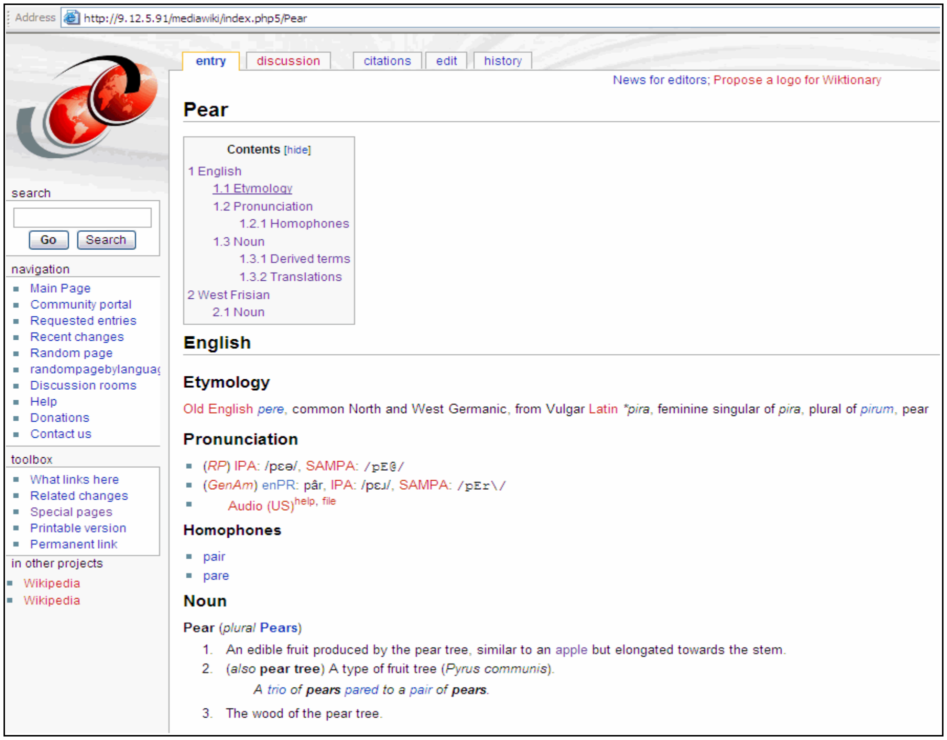


Figure 17-19 Successful display of wiktionary entry

17.11 MediaWiki Migration Conclusions

Looking back at our experience of migrating an open source application, we came to the conclusion that there were no errors or problems encountered with any of the sofrware components themselves. The Linux-Apache-MySQL-PHP (LAMP) software stack was installed easily on Linux on System z. MediaWiki software was also a clean and straighforward install. From these experiences, 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 problems that were 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 usefullness of a pilot project which uses a small portion of the entire application to be migrated. If that is not possibe in your migration project, then any pilot

project that uses the same middleware, tools, data bases, or other functions in your environment would be very valuable.

18

Mono Extensions and Microsoft .NET migration

This chapter provides a discussion of some early experiences of installing and using Mono software. Due to the wide use of .Net, there are millions of developers that have experience building applications in C#. There are some benefits of choosing Mono for application development. We will discuss some those benefits in this chapter.

This will be a very high level overview of this subject. Our team did not personally have in-depth experience with Mono or the .Net Framework. However, we felt that it was relevant to the topic of migration 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.

- C# Compiler

  As mentioned above, there are a lot of developers that have experience in building applications in C#. Along with the programmers comes support materials such as books, websites, education, and sample source code.

- Mono Runtime

  Runtimes provide functionality to high-level programming languages. The runtimes allows a programmer to concentrate on writing the application instead of redundant and complex system infrastructure code.

- Base Class Library

  Class libraries have always been a way to increase productivity. A programmer does not have to code his own solution for many programming acivities. These class libraries are compatible with Microsoft's .Net Framework.l

- Mono Class Library

  Mono supplies classes that supplement the base class library. These classes provide additional useful functionality

- Mono Migration Analyzer (MoMA)

  This is a tool that identifies issues when porting your .NET application to Mono. It helps identify specific calls (P/Invoke) and areas that are not supported by Mono. The results provides a guide to get started on porting .NET applications.

For the most recent information on the Mono Open Source Project, visit their website at:

http://www.mono-project.com

18.2 Installation of Mono Extensions from Novell

Novell provides a for fee extension to the common Mono code base. It's official name is SUSE Linux Enterprise Mono Extenstion. The supported platforms are x86, x86_64 and IBM System z 64 bit(s390x).

Note: An internal name of the 64 bit version of Linux on System is s390x. The **Redhat Package Manager**(RPM) names are known as s390x. The first hardware processor to run Linux was known as System 390. The internal name for RPM has not changed.

Even though we are going to use the Novell Mono Extension, we will not go into a detailed explanation of the product. Their website has more details and will have the most recent imformation. Their website is:

http://www.novell.com/products/mono

18.2.1 Download Mono Extension

The first step in this example is to download an evaluation copy of Mono Extension from the Novell website. You have to register to get a valid logon to the download site. After we got access to 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 to a loopback device so that YaST had access to it. The mount command we used is shown in Figure 18-1:

```
mount -o loopback -t iso9660 /home/louis/SLES11-MonoDVD.iso /mnt/iso
(We shortened the iso filename when moving it to /home directory.)
```

Figure 18-1 Mount the iso for Yast access

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 then led us through the install of the Mono Extension product. There was a pre-requiste of a few apache2-mod_mono packages. We just switched over to the Sotware Management function of YaST and did 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 was well-behaved. It installed the product in an /opt/novell/mono directory. These directories were created by the install process. The userid of the person who does the install will require update authority to the /opt direrctory.

18.2.2 Perform HelloWorld Test

Is there anything more basic yet universal than the HelloWorld test? We coded a simple C# program from an example on the Internet, compiled the program and then executed it. The commands and results are shown in Figure 18-2.

```
root@linux3|/home/louis mcs HelloWorld.cs
root@linux3|/home/louis mono HelloWorld.exe
Hello World form Louis Henderson
```

Figure 18-2 Hello World in C#

18.2.3 Running XSP ASP.NET Examples

There are examples that come with the Mono product. First we went to the directory where the test samples resided:

```
cd /opt/novell/mono/lib/xsp/test
```

Then we started the XSP server from the above directory, with the default settings except for the address setting where the server resides - our case, the address is 9.12.5.92.

```
/opt/novell/mono/bin/xsp -- address 9.12.5.92
```

The server is now running. In Figure 18-3, we show a complete example of starting the server, along with the messages that come from the server when it is started.

```
[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.
```

Figure 18-3 The server is now running

Then 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 as seen in Figure 18-4 was displayed.

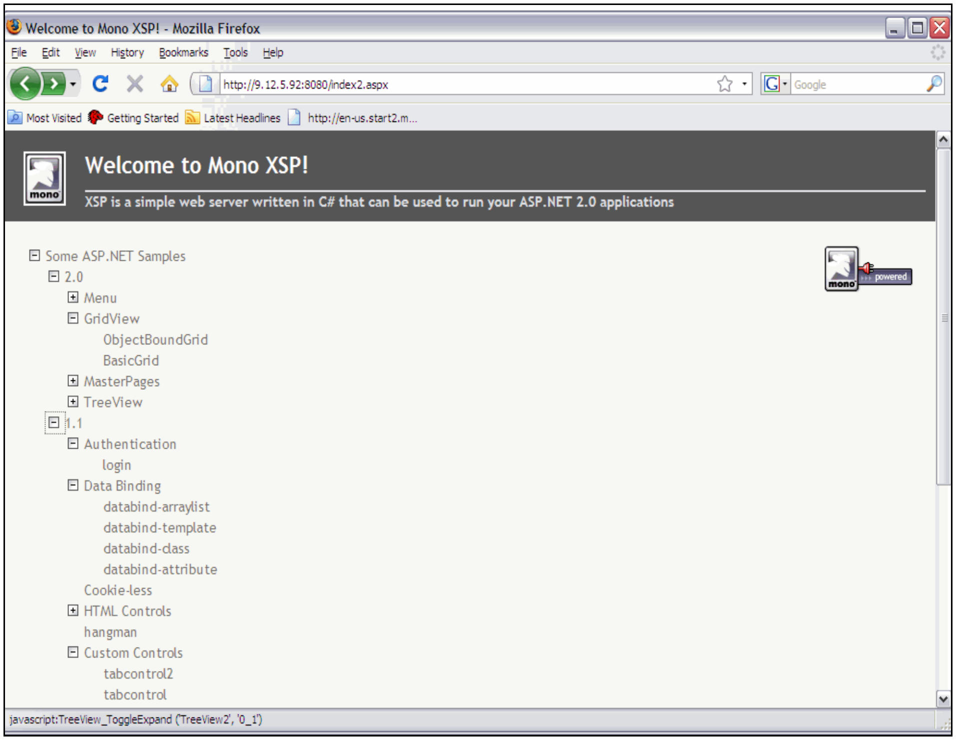


Figure 18-4 XSP web server showing ASP.NET examples

The samples were separated into ASP.NET 1.1 and ASP.NET 2.0. In the ASP.NET 1.1 section, there were nearly one hundred 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 as well as discuss installation of the Apache server and implementing Mono modules on the Apache server.

18.3.1 Example Application and Environment

Now we have our example MS .Net application running on Linux using XSP as in above "Running XSP ASP.NET Examples" on page 234. For our satisfaction, we wanted to test part of an already running application. So we selected the attendance register module for this case study since it had controls varying from basic to advanced MS .NET controls (like Gridview, Calender, Radio button list and drop down controls.).

> **Note:**This is not a full fledged case study on how to migrate an **MS.Net Application** on to Linux on System z with Mono extensions. Here we are just doing a feasibility study on running the MS.Net application or a module on Linux on System z.

Current environment

Currently the application was setup on an MS Windows 2000 Server, with 2GB of memory. The application was developed using MS.Net technologies like C# and ASP.Net. Currently the MS Windows server is running Internet Information Server (IIS) for hosting the Attendance application.

Target Environment

The target environment is installed with the latest GA version SLES 11 with all the necessary development tools and libraries. We show how we find the version of Linux in Figure 18-1.

Example 18-1 Linux on System z version

```
[root@linux5|~] cat /etc/issue
Welcome to SUSE Linux Enterprise Server 11 (s390x) - Kernel \r (\l).
```

The Kernel Version can be found by issuing the command shown in Figure 18-2

Example 18-2 Kernel Version

```
[root@linux5|~] uname -r
2.6.27.19-5-default
```

18.3.2 Installation of Apache

We decided to go ahead with the installation of the latest Apache release, in our example it is release 2.2.11, from the apache website. Figure 18-3 shows the steps in installing the Apache-http server on to our Linux on System z server.

Example 18-3 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
```

Once the installation is complete, the Apache version needs to be verified. We have installed the Apache server in the directory, **/usr/local/apache**, which is passed as a parameter to the **./configure** command. See Figure 18-4.

Example 18-4 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
```

Figure 18-5 shows how we can start the Apache-HTTP server just to check that everything is working properly.

Example 18-5 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 is working by accessing the webpage by just typing in the IP address of the server in the location bar of a Firefox web browser. The error logs should be checked for possible error and any errors need to be resolved before going on to the next steps.

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 namely *mod-mono-server*. This program actually handles the

requests. The communication between the Apache module and mod-mono-server is established using sockets.

Now that we have configured the Apache-HTTP server, we need to update the Apache configuration files in order 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. They are:

1. Include and use the default mono configuration file. This configuration file needs to be included in the httpd.conf. This is also known as Autohosting by the Mono community.
2. Manually update the httpd.conf to include the mono modules and also to make Apache recongize the MS.NET file extensions like .aspx..etc

Autohosting

Autohosting basically means that we 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 Figure 18-6.

Example 18-6 Configuring httpd.conf 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 would enable 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 the projects would be in the htdocs directory.

Manual Configuration

As discussed already, the Apache HTTP server's configuration file needs to be manually updated with the Mono modules and also the changes that would make the HTTP server recongnize the the MS.NET file extensions.

We started off with editing the http.conf file, by including the mod_mono.so modules (see Figure 18-7).

Example 18-7 Manual configuration of httpd.conf

```
[root@linux5|/usr/local/apache/conf] vi httpd.conf
LoadModule mono_module /usr/lib64/apache2/mod_mono.so
```

Now the httpd.conf file needs to be updated, so that the HTTP server can recognize the MS.NET file extensions. In Figure 18-8, we show the file extensions that we included.

Example 18-8 Updating httpd.conf to recognise 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
```

Also the Apache configuration file has to be updated so that it can find out which applications need to use the mono modules. Example 18-9 provides the Apache HTTP server with the applications that need to be handled by the Mono modules.

Example 18-9 Mono modules configuration in httpd.conf

```
Alias /test10 "/opt/atendmod"
AddMonoApplications  default "/test10:/opt/atendmod"

<Location /test10>
    SetHandler mono
</Location>
```

In Example 18-9 on page 239, we have 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. Also we are setting up the handler for the application as Mono using the **SetHandler** tag

Once the httpd.conf file is updated with all the configuration details as described, it’s time to start the Apache HTTP server with the Mono modules initialized.It is also recommended 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 website offers quite a good explanation on fixing commom problems. Example 18-10 on page 240 shows starting the server with Mono modules.

Example 18-10 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
```

From this example, the error log shows that the Apache HTTP server has been started sucessfully with the required Mono extensions and modules.

18.3.4 Verification

Now that we have setup Apache with the Mono extensions, it's time to place the actual MS.NET application in their respective directories as specified in the httpd.conf configuration file. In our case we are placing the MS.NET application under the /opt directory. We copy the applications over, as shown in Example 18-11 on page 240.

Example 18-11 Copying MS.NET application on to 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
```

We had small issues on the images not being displayed, but upon verification of the `Default.aspx web page, we found the image file directory l`ocations incorrectly pointing to some other directory as if it were still on the MS Windows platform.

Apart from that, the attendance module was working perfectly with all the MS.Net controls handled correctly by the Mono extensions. Figure 18-5 on page 241 shows our successful application launch from the web browser.

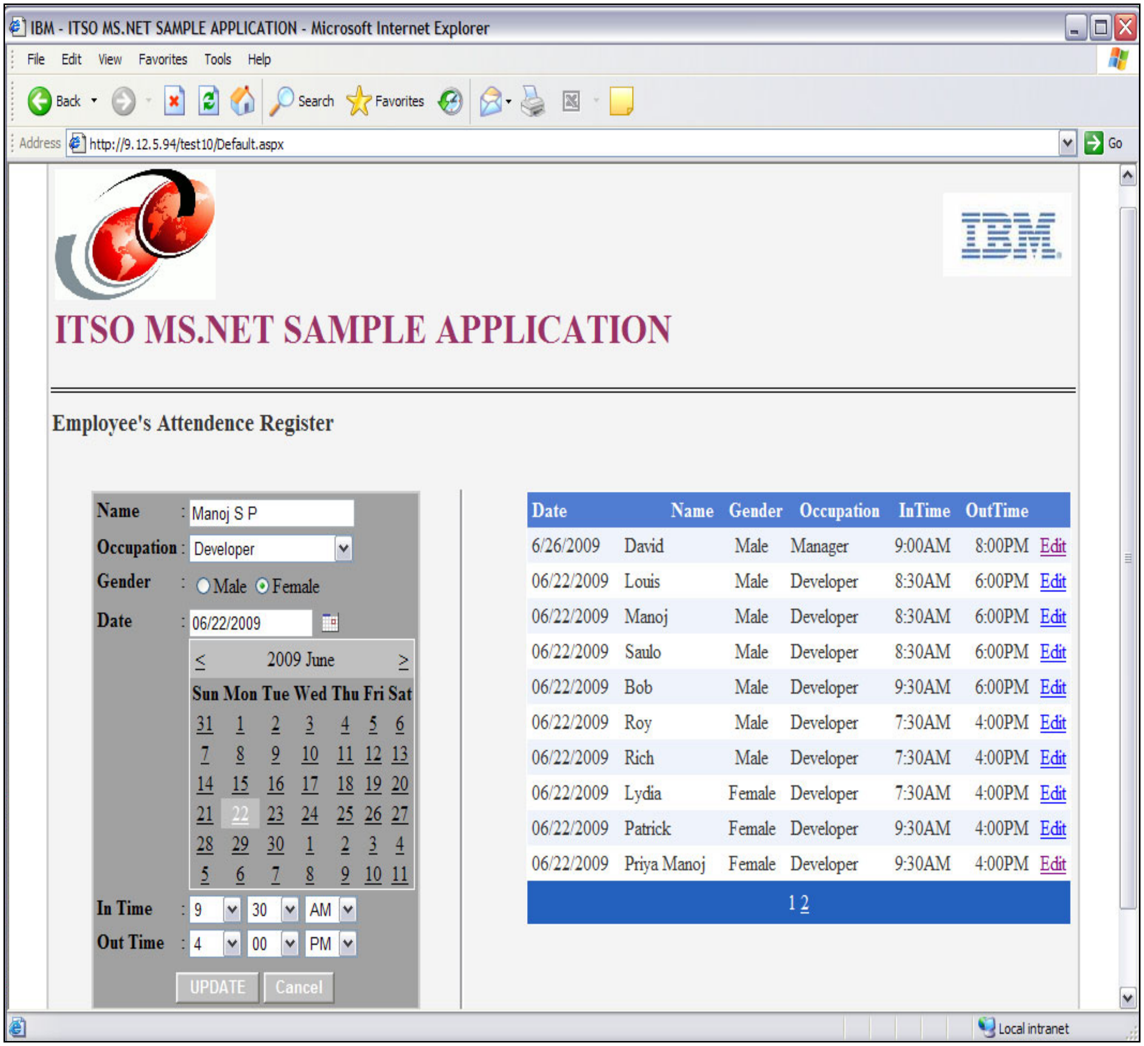


Figure 18-5 MS.Net Sample Module Running on Linux on System with Mono Extentions

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. 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.

Device Drivers, Features, and Commands, SC33-8411-02 (May 2009) can be found at:

http://www.ibm.com/developerworks/linux/linux390/development_development_documentation.html/

.Linux on System z Commands

Linux on System z	Usage
chccwdev	Set ccw devices online or offline
chchp	Change channel path status
chreipl	Modify re-IPL configuration

Linux on System z	Usage
chshut	Control system behavior
chzcrypt	Modify zcrypt configuration
cpuplugd	Activate CPUs and control memory
dasdview	Prints DASD information
dasdfmt	Format a DASD
fdasd	Partition a DASD
lschp	List channel paths
lscss	List subchannels - This command is used to gather & 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 behaiour during change of state
lstape	List tape devices
lszcrypt	List information about cryptographic adapers
lszfcp	List infromation 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
tape390_display	Display messages on tape devices and load tapes

Linux on System z	Usage
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 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 which are nessary to work with Linux on System z:

- 3270 Emulator for CP/CMS and Linux commands
- SSH, Secure Shell for a remote terminal
- SFTP, Secure File Transfer Protocol for file transfer
- The VNC remote desktop

Note: This appendix documents Linux and MS Windows tools and provides corresponding examples. Since 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 which is necessary to communicate with some components on a System z computer. Thus, a program that can communicate using this protocol are called 3270 emulators. In the past, a programmer would use a rather large dumb terminal that took up most of their desk. The physical connection was usually 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. It would not be a good idea 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 avable for all Linux distributions and can be installed from its repository of programs. It can be found at

`http://x3270.bgp.nu/download.html`

or

`http://sourceforge.net/projects/x3270/`

As you can see in Figure B-1, the x3270 is a full featured terminal emulator and is more than adequate for performing all the required administration tasks for z/VM.

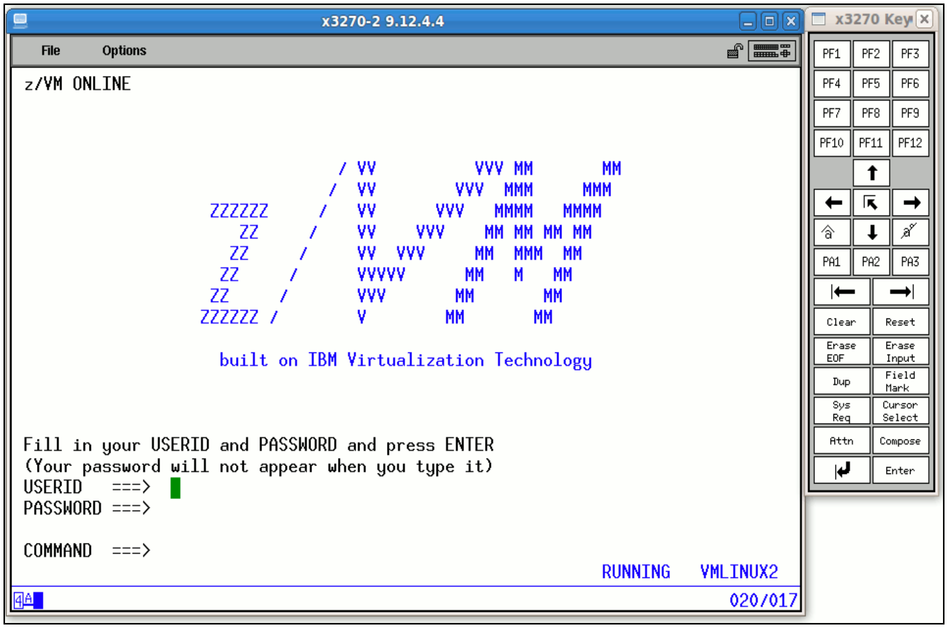


Figure B-1 x3270 running a session to our test VM.

> **Note:** The default background is the color black.Colors can be manually chnaged

MS Windows 3270 Emulation

For Microsoft Windows, there are a number of both commercial and FOSS(Free Open Source Software) 3270 emulator programs available. The x3270 program discussed above has been ported to MS Windows. More information can be found at:

`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, Figure B-2. You can get more information about this software at the following website:

`http://www-01.ibm.com/software/network/pcomm/`

> **Tip:** The enter key is the Ctlr key on the QWERTY pad or the Enter key on the numeric keypad, if available.

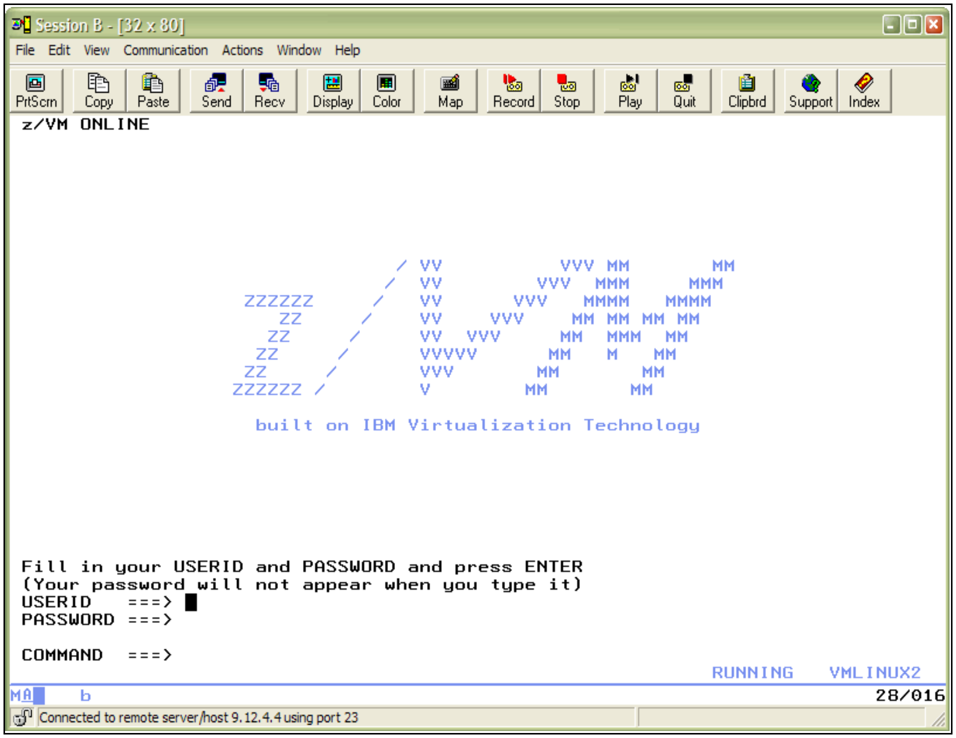


Figure B-2 IBM Personal Communicator showing the a 3270 session.

> **Note:** The default background is the color black.Colors can be manually chnaged

SSH, Secure shell

Secure shell is a network protocol that uses a secure channel between two network devices. It has improvements of confidentiality and integrity over older protocols such as Telnet. We will discuss and show examples of using components of the secure shell protocol suite

SSH Server

SSH is the most common way to access a remote Linux server. 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.

Please note 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 needs to 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. This can be easy done by running the command in Example B-1, "Red Hat" or Example B-2, "SuSE"

Example B-1 Red Hat

```
# sudo /etc/init.d/sshd reload
Reloading sshd: [  OK  ]
```

Example B-2 SuSE

```
# 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 it's predecessor but dropped from popularity since it sent all text in the clear including the password. SSH will encrypt all information sent over the network, this is the reason why both client and server have public and private keys.

Linux SSH Client

Most if not all Linux distributions come with the ssh client already installed. The most common version is OpenSSH, please see:

http://www.openssh.com

The command to connect to server "some.server.name as "user" is shown in Example B-3

Example B-3 Using ssh for the first timet

```
# 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. Please note that the first time you connect to a server its public key will be added to ~/.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.

SSH can also be used 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. Thus, if you wanted to test a connection to port 80 (the default port for the HTTP server) see Example B-4

Example B-4 Useing 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
```

Where the first 80 is the server side port then the localhost:80 is where it will be mapped to.

> **Note:** When mapping ports you are doing so on a system wide level. Thus 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 sill running. Thus one possible solution is to map to a local port that is not in use.

Another useful port forward is for VNC as 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 would move the VNC session on:1 to the local machine. This would allow you to connect to a VNC session on a server in which the 590x port range was blocked via a firewall. Thus 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.

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. So 5901 is the port that needs to be tunneled, but the viewer expects the relative number “1” since we move the port local all that is needed is localhost:1

MS Windows SSH Client

There are many SSH clients for Windows, both commercial and free of charge. We used PuTTY. It can be downloaded from the internet and easily installed. Please see:

http://ftp.chiark.greenend.org.uk/~sgtatham/putty/

Basic setup and configuration of all the PuTTY functions is outside the scope of this section.

Tip: Install the Bitstream Vera Font avable at http://www.gnome.org/fonts/ in MS Windows (use your favorite web search and lookup “installing fonts windows” if you do not know how) then use it as your font for PuTTY. Go to Window → Appearance then use the Change button as in Figure B-3 and set the font to Bitstream Vera Sans Mono. It will make it easier to identify zero (0) as apposed to capital “O” and lower case L (l) as apposed to the number one (1).

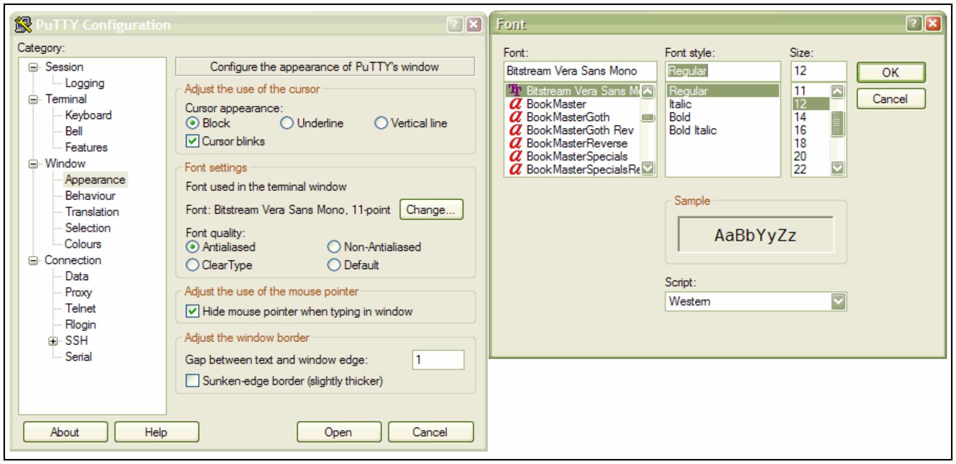


Figure B-3 PuTTY font configuration

Note: PuTTY can be used like OpenSSH to tunnel important ports from server to client. On the configuration window: Category → Connection → SSH → Tunnels add the VNC port used, say “5901” as an example to the Source port field then “localhost:5901” to the Destination field. You need to click the “Add” button to have it added to the list of forwarded ports. Note: you will need to go back to Session then click “Save” to permanently save the port forwarding information. See Figure B-4 on page 257

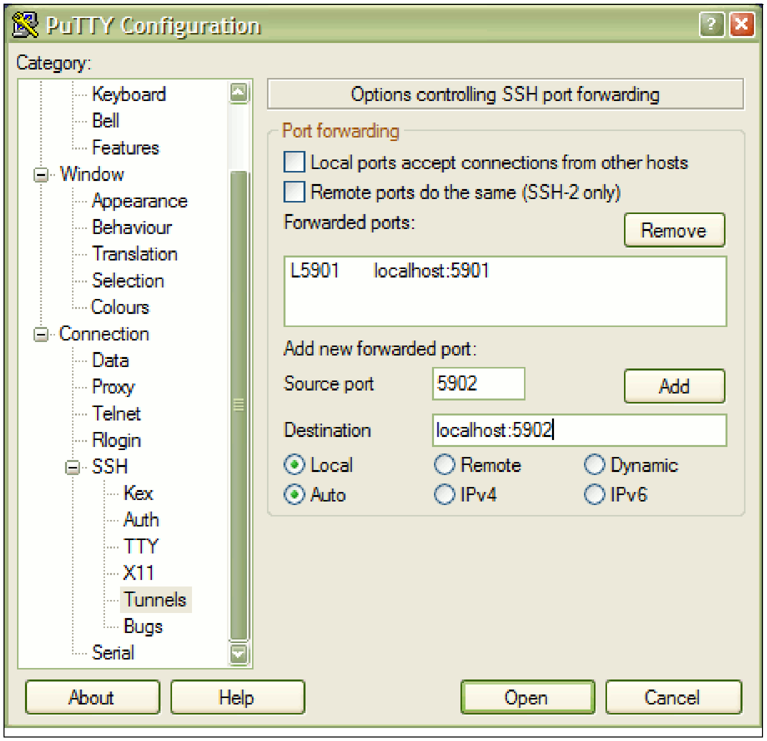


Figure B-4 Setting up PuTTY for port tunneling

Tip: Please note that if you have more then one **PuTTY session** running and mapping the same ports, the first session that was run will own those ports. Thus it is advisable to map different systems to different local ports to avoid this conflict.

SFTP, Secure File Transfer Protocol

SFTP is part of the SSH protocol, thus it does not take an extra service to configure and maintain. It is the easiest way to move files to and from your server, assuming that the SSH service is running on the server. Make sure the line 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
```

This is the preferred method to move 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 would be via the command line as in Example B-8

Example B-8 sftp options on Linux

```
sftp [[user@]host[:file [file]]]
sftp [[user@]host[:dir[/]]]
```

In this way you can move one or many file to or from the target Linux server. Also you can just use a directory to move a full directory

SFTP under MS Windows

Any good Microsoft Windows FTP program should include sFTP access both commercial and FOSS. The client used during the writing of this book was the FOSS client Filezilla and is shown in Figure B-5 on page 259:

http://filezilla-project.org

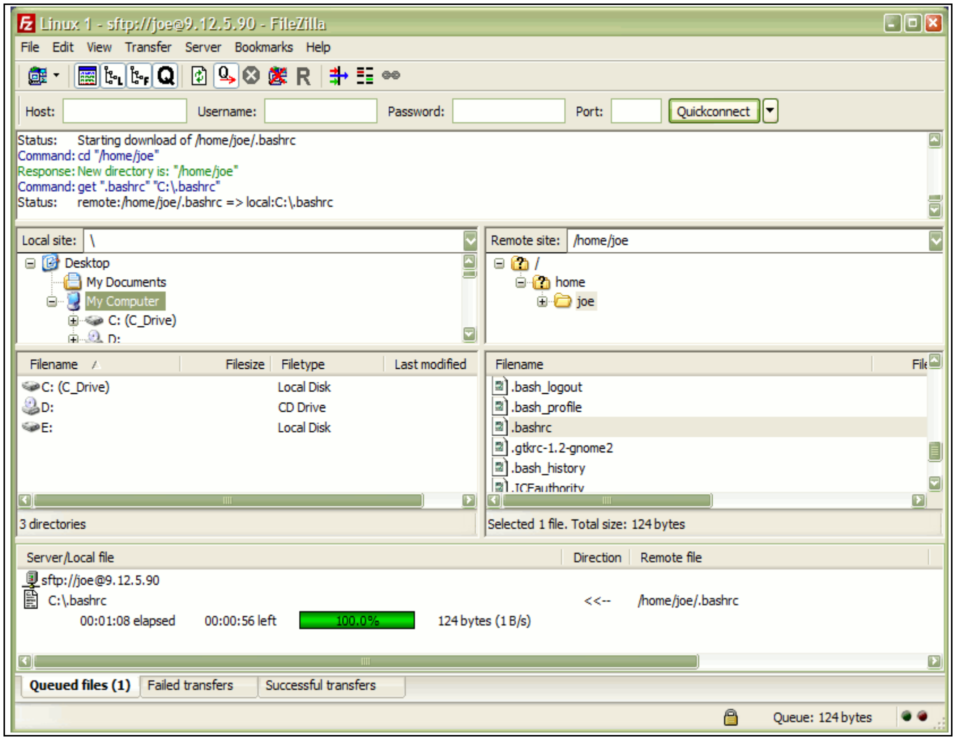


Figure B-5 Filezilla MS Windows example

VNC, Remote desktop

VNC is a remote graphical desktop that has two parts, a client and a server. Normally one 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 system.

Note: VNC is only the mechanism that moves the graphics from the server to the client. Xwindows is still run as the GUI environment and also a windows manager is needed by Xwindows to present the desktop metaphor. It is also interesting to note that the VNC server is avable 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 laptop. Again this is because VNC is moving the graphics from server to client.

Setting up VNC server on System z

There are two ways to configure VNC: a server per system or a server per user. When setting up a server per system the advantage is that an SSH session is not required to manage the server. The advantage of a per user VNC server is that all the configuration and use can be done via SSH and thus network security need not be altered.

Configuring a VNC server per system

If your Linux VM guest is installed with SuSE (all versions apply) then you are done! Just point your VNC client to “hostname:1” (where hostname is the address of the linux VM) and you will see the GDM login screen.

> **Important:** When logging into a Linux system via VNC please note that when the username and password are typed in they will be transmitted as VNC traffic over IP and thus in the clear (unencrypted). Thus this is NOT recommended for remote login over the internet.

If the Linux virtual machine is installed with Red Hat (all versions apply) then you will have to do some configuring to get it to work like a SuSE system. For further information please see the Red Hat web site “Red Hat knowledge base: How do I setup 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 re-boot of the system. Since we are implementing this on a System z s390 we thought it appropriate to document the steps.

1. Edit /etc/gdm/custom.conf and add what is listed in Example B-9 under there 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 Example B-11 to /etc/services

Example B-11 /etc/services

```
vnc01     5901/tcp     # GDM on VNC via xdmcp remote
```

4. Reload xinetd's config that was just updated so it takes on the changes as in Example B-12

Example B-12

```
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] then put a * in the “Trusted Devices” box. Then “Ok” and “Ok” to exit and activate the changes.
6. start gdm as in Example B-13

Example B-13

```
/usr/sbin/gdm
```

7. Put the system in runlevel 5 as 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

```
telinit q ; telinit 5
```

8. At this point VNC is now configured to serve GDM (the Gnome great program that will allow logon) on the VNC port one, you can test by using your VNC client as in Example B-15 where hostname is the address of the linux VM.

Example B-15

```
vncviewer hostname:1
```

Configuring a VNC server per user

Once a user is at a command prompt, to start a vnc server just run the vncserver command. Unfortunately the VNC server's default configuration is not usable at best, so this section explains how to configure the per user settings to make VNC server usable.

The configuration files are kept in the users home directory as in "~/.vnc" but that directory will not exist until a VNC command is run. I suggest first setting the required VNC password as 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". When we next run the vncserver command it will start the server and create a default xstartup file. The default startup file will default to primitive settings that are not likely to be useful. I suggest useing this Example B-17 for a Red Hat server and Example B-18 for a SuSE 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
```

Example B-18 /.vnc/xstartup file for SuSE

```
#!/bin/sh
xrdb $HOME/.Xresources
exec /usr/X11R6/bin/fvwm
```

Once "~/.vnc" contains a passwd and xstartup files you can start the server via Example B-19 Please read the manual page for a complete explanation of all the vncserver options.

Example B-19 Starting up a VNCserver

```
$ 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 shutdown the server when it is no longer needed, just run Example B-20 Note: 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 thing to do is to match the client with the server. Some VNC client will promote them selves as having automatic detection, which is interesting, but not necessary since there are only two common servers:

- Real vnc, see: http://www.realvnc.com
- Tight vnc, see: http://www.tightvnc.com

Thus just match the client with the server. We used standard VNC or vncserver for the writing of this book.

Under Linux

The VNC Viewer application is avable for all Linux distributions

Figure B-6 on page 264 is 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

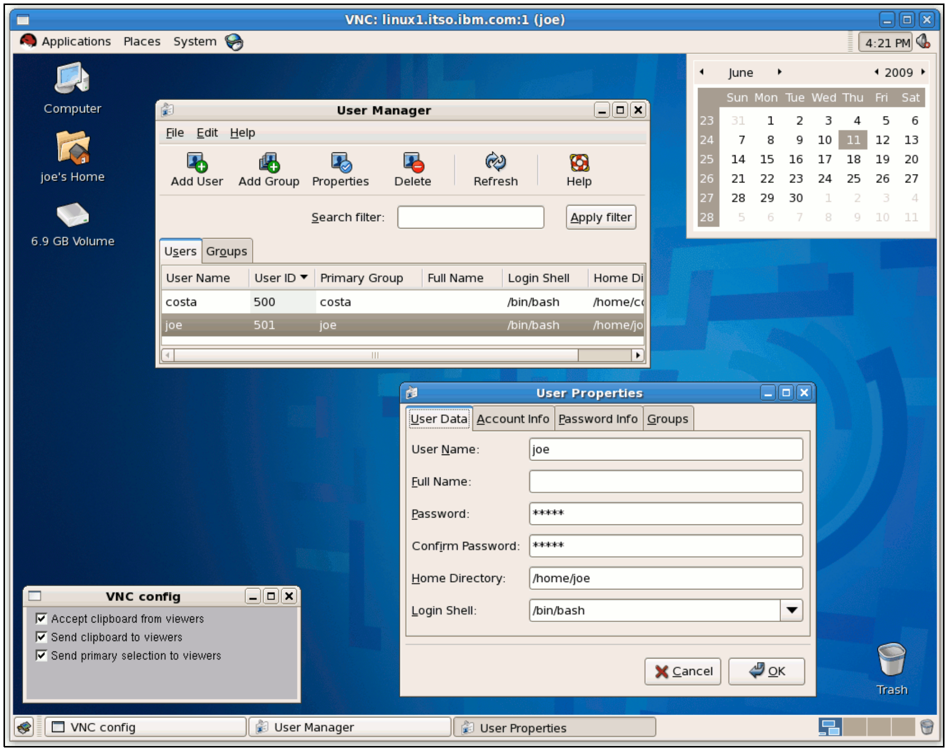


Figure B-6 Example VNC client running on a Linux Desktop from a Red Hat Serve

Under MS Windows

The MS Windows clients can be downloaded from the respective web sites and installed. Figure B-7 is an example of a remote desktop from a SUSE server running as a guest under z/VM with the optimized settings from Example B-17

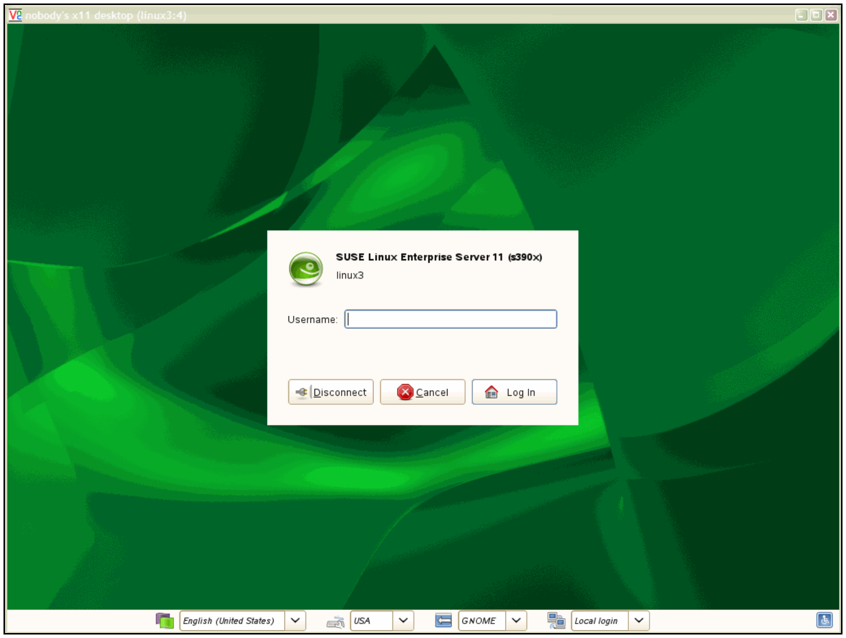


Figure B-7 Example VNC client on MS Windows from a SuSE Server login screen

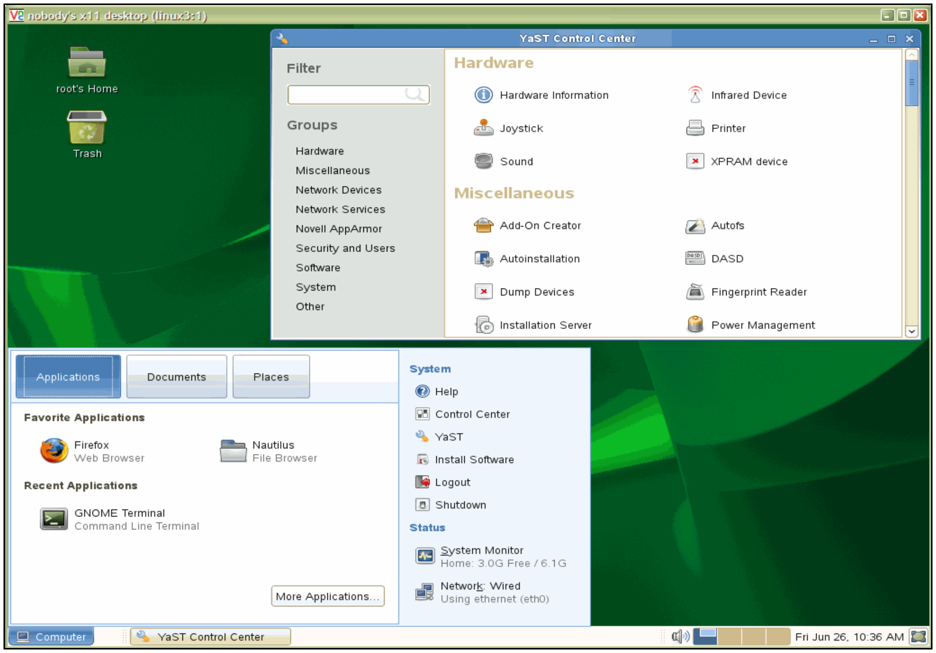


Figure B-8 Example VNC client on MS Windows with SuSE Desktop

C

Performance measurement

In this appendix we discuss performance measurement and its impact on the success of an application migration. The main point that is stressed is the need to measure the performace of the source application when running in production and compare that with the performance of the application on the target environment. The key is to choose measurable statistics in order to identify the performance improvement provided by the target ennvironment.

We also discuss some monitoring commands and tools that can assist in identifying and resolving performance inhibitors.

What is performance?

Performance with regards to 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 etc. However, when a migration is undertaken, it is very important to understand the performance metrics used on the source environment in order to understand the relative performance improvement on the target system.

Often the initial performance of the new system may not be what was expected and tuning must be done in order to improve the performance of the target system.Without proper metrics it is impossible to validate the performance of the new platform relative to the old. As such, one of the first things the migration project teams needs to do is to agree on what performance metrics from the source platform will be used in the migration project plan in order to measure the performance of the target platform.

Choosing what to measure

In order to determine the success of a migration, having the application on the target platform provide the same answers as the source platform isn't going to prove success. The natural expectation of the migration onto Linux on System z will be that the application is not only more resilient and available because of System z but that it also provides equal or better performance than the source platform. To ensure that the performance improvements are easy to show, it is important that the right metrics are chosen. 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 be 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 thing 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 269 shows the various components that make up user response time.

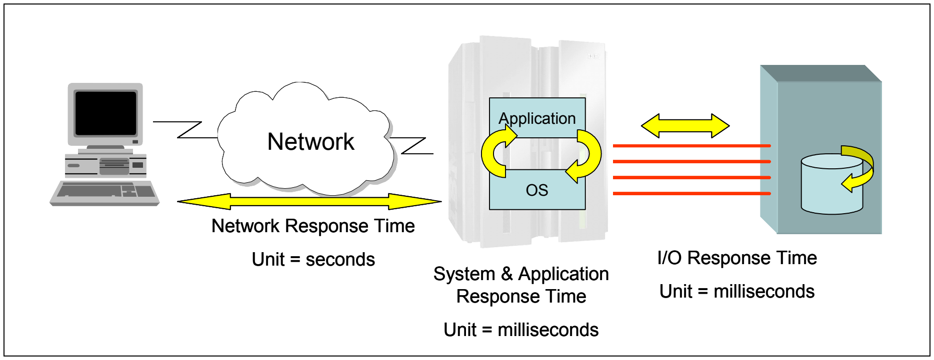


Figure C-1 Components that make up the reponse time of transaction

This figure shows that there are two points where response time could be measured. There is system response time and user response time. When trying to understand the relative performance improvement from a new system the only place to measure response time is from when a system receives the request and when it provides an response of some sort to the request. In the above case 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 as it can be influenced by network load.

In order 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

This performance metric perhaps provides a more meaningful measure of system performance as it is measures the number of transactions processed over a period of time. This is typically one second but could be any time period you wish.

In both cases the main point to remember is that you should have baseline performance metrics for the source system in order to properly compare both the new and old systems.

Performance and the proof of concept

Provided that the environment for the proof of concept is identical to the final target environment, you could use the proof of concept to get a good

understanding of the expected performance and throughput of the new system. However, the results of the proof of concept may not be duplicated on the final platform as 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 you may find that the target environment is not large enough to provide adequate performance.

If you are planning to modify the source application during the migration you should do some stress testing of 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 (LPAR). 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 amongst a number of LPARs.

Memory is also allocated to each LPAR and a thorough analysis of the intended workloads should be done 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 over committment ratio of 2:1 for production workloads or 3:1 for development/test workloads.

I/O devices are generally shared amongst 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 3 IFLs and 2 LPARs. Each LPAR should be asigned 2 Logical CPUs. Each virtual machine running JAVA should have two virtual CPUs as this improves JAVA dispatcher performance.

Size the memory allocated to each LPAR according to the 2:1 memory over committment 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 not satisfactory you will need to tune the various IT components in order to improve performance. To understand what needs to 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 some system tools such as:

- vmstat
- top
- Process Status (ps)

There are also a number of Open Source tools for Linux such as:

- **Nagios:** Comprehensive monitoring all mission-critical infrastructure components - including applications, services, operating system, network protocols, system metrics and network information.Figure C-2 on page 272 is a display of the Nagios tool that was installed on our Linux on System z server.

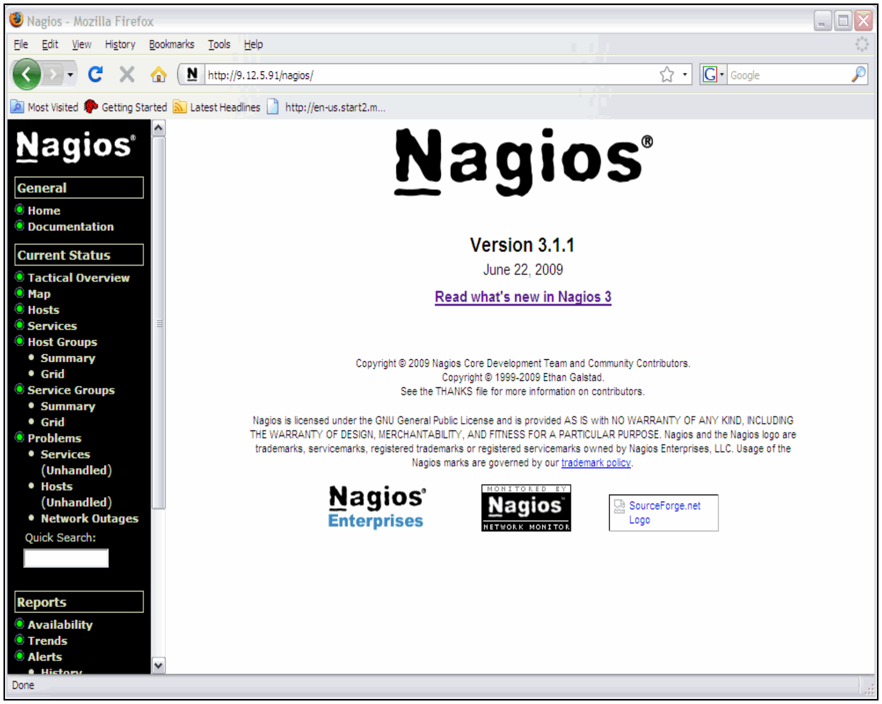


Figure C-2 Nagios home page

- **Xymon** (previously known as Hobbit): a system for real-time monitoring of hosts and networks we a web-based interface with availability reports and performance graphs.

The areas that typically need tuning in order to improve performance are:

- Processor sharing parameters
- Memory sizing
- Disk I/O
- Network I/O

The good news is that tools such as the IBM Tivoli Omegamon XE on z/VM and Linux will in many cases identify where any contention is occuring. This will allow you to more quickly solve the problem.

For a comprehensive analysis of performance tuning for a Linux on System z environment please refer to the following Redbook:

Linux on IBM System z: Performance Measurement and Tuning, SG24-6926

D

Additional material

This book refers to additional material that can be downloaded from the Internet as described below.

Note to Editor: This may be removed - awaiting whether or not we will included downloadable materials - 9/2/2009 - Lydia

Locating the Web material

The Web material associated with this book is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser at:

`ftp://www.redbooks.ibm.com/redbooks/SG247727`

Alternatively, you can go to the IBM Redbooks Web site at:

ibm.com/redbooks

Select the **Additional materials** and open the directory that corresponds with the IBM Redbooks form number, SG247727.

Using the Web material

The additional Web material that accompanies this book includes the following files:

File name	*Description*
????????.zip	????Zipped Code Samples????

System requirements for downloading the Web material

The following system configuration is recommended:

Hard disk space:	????MB minimum????
Operating System:	????Windows/Linux????
Processor:	???? or higher????
Memory:	????MB????

How to use the Web material

Create a subdirectory (folder) on your workstation, and unzip the contents of the Web material zip file into this folder.

Glossary

Your glossary term, acronym or abbreviation Term definition.
Sort terms: **highlight rows** → **Table** → **Sort** → **Column1** → **Sort**

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 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 Logical Partition

Metro Mirror Data replication technique designed to replicate data across distances up to 100km. Metro Mirror uses PPRC.

Minidisk A virtual disk occurying 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 virtualization layer built into IBM System z. Allows the creation of Logical Partitions (LPARs) for workloads to execute in.

ROI Return on Investment, the amount of time it takes to recover the TCA from savings generated by a project.

Source Server server being migrated from

Target Server server being mirated to

TCA Total Cost of Aquisition, 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 3 or 5 years.

Virtual CPU CPU defined to each Virtual Machine in the z/VM Directory

Virtual guest a virtual machine running under z/VM.

Vswitch Virtual LAN managed by z/VM

z/VM Virtualization hypervisor for IBM System z

Abbreviations and acronyms

This abbreviations and acronyms file is optional.	Use this file by adding names and descriptions to it. Sort these names: **highlight rows ∅ Table ∅ Sort ∅ Sort By: Column 1 ∅ Sort** or optionally add names and descriptions to the Index file instead of this file by indexing the first use of an abbreviation or acronym: **highlight text ∅ Special ∅ Marker ∅ Index ∅ New Marker**
abbreviation1	Description1
abbreviation2	Description2
IBM	International Business Machines Corporation
ITSO	International Technical Support Organization
abbreviation3	Description3
abbreviation4	Description4

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 284. Note that some of the documents referenced here may be available in softcopy only.

- IBM System z Strengths and Values, SG24-7333-01
- Linux on IBM System z: Performance Measurement and Tuning, SG24-6926
- Linux Performance and Tuning Guidelines, REDP-4285
- Problem Determination for Linux on System z, SG24-7599
- *Linux on IBM eServer zSeries and S/390: Best Security Practices*, SG24-7023
- Introduction to the New Mainframe: z/VM Basics, SG24-7316
- 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
- Linux on IBM eServer™ zSeries and S/390: Performance Toolkit for VM, SG24-6059
- IBM Lotus Domino 6.5 for Linux on zSeries Implementation, SG24-7021
- *Hipersockets Implementation Guide*,SG24-6816
- *IBM System z Connectivity Handbook*,SG24-5444-09
- *OSA-Express Implementation Guide*,SG24-5948-05
- Linux wih zSeries and ESS: Essentials, SG24-7025
- Solaris to Linux Migration: A Guide for System Administrators, SG24-7196
- Experiences with Oracle Solutions on Linux for IBM System z,SG24-7634
- Using Oracle Solutions on Linux for Sysem z, SG24-7573
- Fibre Channel Protocol for Linux and z/VM on IBM System z, SG24-7266

- z/VM and Linux Operations for z/OS System Programmers, SG24-7603
- SAP on DB2 9 for z/OS: Implementing Application Servers on Linux for System z, SG24-6847-01
- *Linux on IBM eServer zSeries and S/390: VSWITCH and VLAN Features of z/VM 4.4,* REDP-3719-00
- *Networking Overview for Linux on zSeries*, REDP-3901-00

  Hardware information is a little dated but network concepts are described well. Linux network options are for the older Linux 2.4 kernel.

Other publications

These publications are also relevant as further information sources:

- 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
- Getting Started with Linux on Sysem z Version 5 Release 4, SC24-6096-03

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:l

  `http://www.vm.ibm.com/pubs/l`
- z/VM performance tips:l

  `http://www.vm.ibm.com/perf/tips`
- z/VM performance publications:l

  `http://www.vm.ibm.com/perf/pubs/`

- Documentaion for Develpment Stream Site (Manuals 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 Homepagel

  http://www-03.ibm.com/developeworks/linux/linux390/index.html

- The Linux-S390 list server:

  http://www2.marist.edu/htbin/wlvlists

- IBM Midleware Available on Linux (Download PDF and click on System z)

  http://www-03.ibm.com/linux/matrix/

- A site where you can access relevant Share presentations

  http://www.linuxvm.org/

- Web 2.0 Install Manual for SUSE

  http://download.boulder.ibm.com/ibmdl/pub/software/dw/linux390/docu/lslsw200.pdf

- Web 2.0 Install Manual for Redhat

  http://download.boulder.ibm.com/ibmdl/pub/software/dw/linux390/docu/lrhlw201.pdf

- Linux man pages:l

  http://linuxcommand.org

- Rosetta Stone or Sysadmin Unixversal Translator:l

  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

E

F

G

H

I

J

K

L

T

To determine the spine width of a book, you divide the paper PPI into the number of pages in the book. An example is a 250 page book using Plainfield opaque 50# smooth which has a PPI of 526. Divided 250 by 526 which equals a spine width of .4752". In this case, you would use the .5" spine. Now select the Spine width for the book and hide the others: **Special>Conditional Text>Show/Hide>SpineSize(-->Hide:)>Set** . Move the changed Conditional text settings to all files in your book by opening the book file with the spine.fm still open and **File>Import>Formats** the Conditional Text Settings (ONLY!) to the book files.

IBM Redbooks

Practical Migration to Linux on System z

(1.5" spine)
1.5"<-> 1.998"
789 <->1051 pages

IBM Redbooks

Practical Migration to Linux on System z

(1.0" spine)
0.875"<->1.498"
460 <-> 788 pages

IBM Redbooks

Practical Migration to Linux on System z

(0.5" spine)
0.475"<->0.875"
250 <-> 459 pages

IBM Redbooks

Practical Migration to Linux on System z

(0.2"spine)
0.17"<->0.473"
90<->249 pages

(0.1"spine)
0.1"<->0.169"
53<->89 pages

To determine the spine width of a book, you divide the paper PPI into the number of pages in the book. An example is a 250 page book using Plainfield opaque 50# smooth which has a PPI of 526. Divided 250 by 526 which equals a spine width of .4752". In this case, you would use the .5” spine. Now select the Spine width for the book and hide the others: **Special>Conditional Text>Show/Hide>SpineSize(-->Hide:)>Set** . Move the changed Conditional text settings to all files in your book by opening the book file with the spine.fm still open and **File>Import>Formats** the Conditional Text Settings (ONLY!) to the book files.

IBM Redbooks

Practical Migration to Linux on System z

(2.5” spine)
2.5”<->nnn.n”
1315<-> nnnn pages

IBM Redbooks

Practical Migration to Linux on System z

(2.0” spine)
2.0” <-> 2.498”
1052 <-> 1314 pages

Draft Document for Review September 2, 2009 5:38 pm

Practical Migration to Linux on System z

Overview and migration methodology

Migration Analysis

Hands on migration case study

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 attempt to 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