IBM

Practical Migration from x86 to Linux on IBM System z

- A guide to migrating popular applications and services from Linux on x86 to Linux on System z
- Practical guidance on planning, analysis, and TCO
- Comprehensive hands-on migration case study

Lydia Parziale
Eduardo Simoes Franco
Craig Gardner
Berthold Gunreben
Tito Ogando
Serkan Sahin

Redbooks

ibm.com/redbooks

International Technical Support Organization

Practical Migration from x86 to Linux on IBM System z

September 2014

SG24-8217-00

Note: Before using this information and the product it supports, read the information in “Notices” on page vii.

First Edition (September 2014)

This edition applies to z/VM Version 6.3, DB2 Version 10.5, SUSE Linux Enterprise Server Version 11, and Red Hat Enterprise Linux Version 6. Versions of other software components are incident to the versions available from the respective distributions referenced above.

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 grant 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites 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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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

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

COPYRIGHT LICENSE:

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

Trademarks

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

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

AIX®
DB2 Connect™
DB2®
DS6000™
DS8000®
ECKD™
FICON®
FlashCopy®
GDPS®
HiperSockets™
HyperSwap®
IBM SmartCloud®
IBM Systems Director Active Energy Manager™
IBM®
OMEGAMON®
Parallel Sysplex®
POWER®
PR/SM™
Processor Resource/Systems Manager™
RACF®
Redbooks®
Redbooks (logo)  ®
S/390®
Smarter Planet®
System Storage®
System z10®
System z®
Tivoli®
WebSphere®
Worklight®
z/OS®
z/VM®
z10™
zEnterprise®

The following terms are trademarks of other companies:

Worklight is trademark or registered trademark of Worklight, an IBM Company.

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

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

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

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

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

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

Preface

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

- 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 guide and example for IT organizations to migrate from their x86 environment to Linux on System z. It begins by examining the benefits of migrating workloads to Linux on System z. Here, we describe the workload centric method of information technology and then discuss the benefits of migrating workloads to Linux on System z.

Next, we describe total cost of ownership analyses and we guide you in understanding how to analyze your environment before beginning a migration project. We also assist you in determining the expected consolidation ratio for a given workload type.

We also describe virtualization concepts along with describing the benefits of migrating from the x86 environment to guests residing on an IBM z/VM® single system image with live guest relocation.

This IBM Redbooks publication walks you through a migration approach, includes planning worksheets, as well as a chapter to assist you in analyzing your own systems. We also discuss post migration considerations such as acceptance testing of functionality and performance measurements.

Authors

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

Figure 1 From left: Serkan Sahin, Craig Gardner, Tito Ogando, and Eduardo Simoes Franco

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 over 25 years in various technology areas.

Eduardo Simoes Franco is an IT Specialist and Technology Consultant in IBM Brazil. He has more than fifteen years of experience with IT Solutions and Linux, and has held many technical and management positions at a number of large corporations, where he held a variety positions in servers support, as a network analyst, security officer, IT coordinator, and consultant. He started at IBM as a Linux on IBM System z Specialist and currently is supporting and promoting IT solutions on IBM System z and IBM System Power platforms. He has a post-graduated degree in IT and teaches information management and technology courses in a Unifacs/Laureate university. He has CLA and LPI certifications.

Craig Gardner is a Senior Software Engineer at SUSE. He has 21 years of professional experience with Linux, starting first with installing, configuring, and running Slackware 2.1 throughout university departments as a systems administrator at Brigham Young University (BYU). While at BYU, Craig was a systems programmer for the university's administrative computing services organization on System/370. Craig has been working at SUSE since 2011, and worked for Novell prior to that for a total of 17 years, all the while involved in various projects being ported to and developed for Linux. His responsibilities at SUSE include providing expertise on System z, helping development teams port software to Linux on System z. In addition to his engineering responsibilities at SUSE, Craig is also an adjunct professor of computer science at Utah Valley University teaching classes on software engineering.

Berthold Gunreben is a Build Service Engineer at SUSE in Germany. He has 14 years of professional experience in Linux and is responsible for the administration of the Mainframe system at SUSE. Besides his expertise with Linux on System z, he is also a Mainframe System Specialist certified by the European Mainframe Academy (http://www.mainframe-academy.de). His areas of expertise include High Availability on Linux, Realtime Linux, Automatic Deployments, Storage administration on the IBM DS8000®, Virtualization Systems with Xen, KVM, and z/VM, as well as Documentation. He has written extensively on many of the SUSE Manuals.

Tito Ogando is an IT Specialist in IBM Brazil. He has six years of experience in Linux on IBM System z. He is currently working as a Linux Specialist for the Linux team in Brazil supporting more than 1800 Linux on System z servers for the IBM internal account. He is also an avid Python programmer responsible for maintaining the main tool for automating server builds and managing workloads of his team. He holds a degree in Computer Science from Universidade Federal da Bahia. His areas of expertise include Linux on IBM System z, troubleshooting, and automation.

Serkan Sahin is a Chief Architect for IBM Strategic Outsourcing Service Delivery at IBM in the Middle East and Africa. He has more than 18 years of professional experience in the IT industry. He is an experienced Architect, System Engineer, and IT Consultant who has been developing multi-component wall-to-wall complex IT infrastructure solutions. He has worked for IBM since 1999. He has been an IBM and Open Group Certified Architect as a Technology Architect since 2008. He is a certified IBM instructor for Architectural Thinking, Architecting for Performance Engineering and a Technical Leadership College professional for internal classes in IBM for professional give-back. He has both a Computer Science and Industrial Electronics degree.

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

Richard Conway and Robert Haimowitz, DST
IBM US

Willian Rampazzo
IBM Brazil

Jim Doran and Tamar Eilam
IBM US

Zdenka Spoljaric
IBM Australia

Thanks to the authors of:

- *Set up Linux on IBM System z for Production*, SG24-8137, published in November 2013:

  Lydia Parziale, Saulo Silva, David Borges De Sousa, Livio Sousa, Junius Mills, and Qi Ye
- *Practical Migration to Linux on System z*, SG24-7727, published in October 2009:

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

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

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 email 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

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  https://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

1

Benefits of migrating workloads to Linux on System z

In this section, we discuss the benefits and reasons to migrate to Linux on System z. Additionally, we discuss a new type of information technology that is called *workload centric* and explain how the workload-centric cloud benefits from Linux on System z with an enterprise cloud computing blueprint.

1.1 Benefits

A significant benefit of migrating to Linux on System z is that it allows organizations to break the link between the operating system and specific hardware platforms. This means that after your applications are running on Linux, you are no longer tied to a specific hardware platform. 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. Linux running on System z benefits from the hardware platform that includes a specialized processor, the Integrated Facility for Linux (IFL), cryptographic cards with dedicated RISC processors, and the bare metal hypervisor of IBM z/VM. Figure 1-1 illustrates the commercial IT platforms and IBM products that Linux supports.

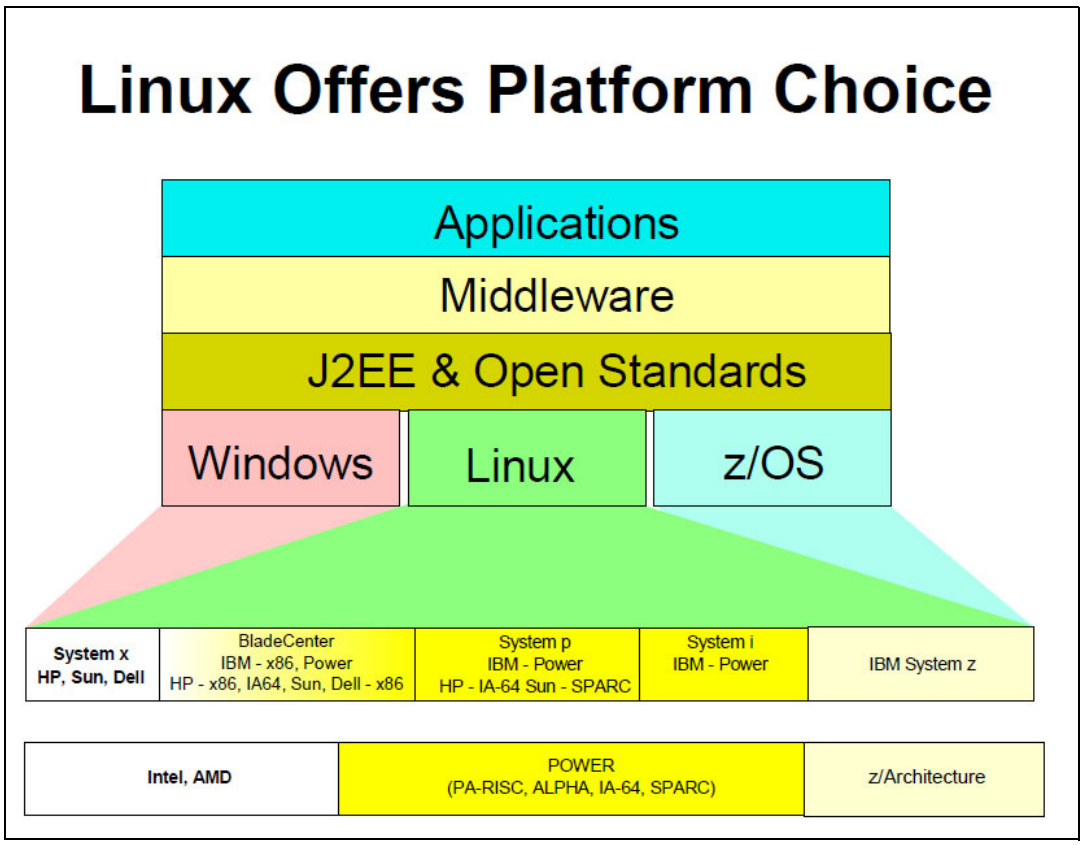


Figure 1-1 Variety of commercial IT platforms supported by Linux

A major benefit of Linux is that it is open source; the software is unencumbered by licensing fees and its source code is freely available. There are hundreds of Linux distributions available for almost every computing platform. Two enterprise distributions[1] of Linux are:

- Red Hat: Red Hat Enterprise Linux (RHEL)
  http://www.redhat.com
- SUSE: SUSE Linux Enterprise Server (SLES)
  http://www.suse.com

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

Both Red Hat and SUSE provide customers using Linux with various support options, including 24 x 7 support with one-hour response time worldwide for customers running production systems. As well as the Linux operating system, both SUSE 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 SUSE to resolve the problem.

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

This multiplatform support allows customers to run a common operating system across all computing platforms, which will mean significantly lower support costs and, in the case of Linux, no incremental license charges. It also offers customers the flexibility of easily moving applications to the most appropriate platform. For example, many IT organizations choose Linux on System z for the ability to scale databases across highly scalable hardware.

1.2 Reasons to select Linux on System z

First announced in 1964, the IBM mainframe is the only computing system that has provided customers with a common architecture for more than 50 years. Today, as it has been for the last 50 years, the IBM System z is the most reliable and scalable computing platform available and it is the ideal platform for consolidating many hundreds of distributed servers.

There are two current models of the IBM System zEnterprise®: the zEnterprise Class 12 (zEC12) and the zBusiness Class (zBC12). Both models share all of the characteristics that make the mainframe a uniquely powerful solution. The zEnterprise Class can scale to 101 configurable processors and 3 terabytes (TB) of memory while the zBusiness Class can scale to 13 configurable processors and 496 gigabytes (GB) of memory.

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

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

1.2.1 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 IBM Processor Resource/Systems Manager™ (IBM PR/SM™).
  - Up to 60 (zEC12) or 30 (zBC12) logical partitions (LPAR)[2] with independent virtual resources.
  - PR/SM is one of the most secure systems available, having achieved Common Criteria Evaluation Assurance Level 5+ (EAL5+) for LPAR isolation. This is one of the highest levels of certification offered that can be achieved by commercially available hardware.

    Note: For more information about Common Criteria, Evaluation Assurance Levels, Protection Profiles, and a list of certified products, refer to the following site:

    http://www.commoncriteriaportal.org

    The certified evaluation levels for System z Operating Systems, as of January 2014, are:

    - IBM z/VM version 6 release 1: certified at EAL4+
    - Red Hat Enterprise Linux 6.2: certified at EAL4+
    - SUSE Linux Enterprise Server 11 SP2: certified at EAL4+

  - The industry-leading virtualization hypervisor z/VM is supported on all IBM System z models.
  - Both PR/SM and z/VM employ hardware and firmware innovations that make virtualization part of the basic fabric of the IBM System z platform.
  - IBM HiperSockets™ allows up to 32 virtual LANs, thus allowing memory-to-memory TCP/IP communication between LPARs.
- Scalability
  - System zEC12 scales to 101 physical processors and up to 3 TB of memory.
  - System zBC12 scales to 13 physical processors and up to 496 GB of memory.
- Security
  - Clear key integrated cryptographic functions provide high-speed cryptography for data in memory.
    - Supports DES, TDES, Secure Hash Algorithms (SHA) for up to 512 bits, Advanced Encryption Standards (AES) for up to 256 bits, and Pseudo Random Number Generation (PRNG).
  - Optional cryptography accelerators provide improved performance for specialized functions.

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

- 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 backup capacity in addition to the installation that might be already available in numbers of assigned processors. This is intended to replace capacity lost due to a disaster. CBU gives customers the peace of mind knowing they can access additional capacity in the event of a disaster recovery situation without having to purchase additional capacity. Typically, this would allow customers to sign up for CBU on an IBM System zEC12 at another site and use this capacity for a number of contracted disaster recovery tests or for a contracted time in the event of a declared disaster at the customer site. For more information about CBU, check the *IBM zEnterprise EC12 Technical Guide*, SG24-8049.

► Power and cooling savings

– With its low power and cooling requirements, the IBM System zEC12 is an ideal platform for the consolidation of distributed servers.

– Consolidating hundreds of distributed servers to IBM System zEC12 reduces the power and cooling load in the data center.

– The IBM Systems Director Active Energy Manager™ (AEM) provides a single view of actual energy usage across heterogeneous IBM platforms within a data center. AEM allows tracking of trends, which provide accurate data to help properly estimate power inputs and more accurately plan data center consolidation or modification projects.

Note: For more detailed studies about IBM System z servers, consult the following IBM Redbooks technical guides:

– For zEnterprise Class 12 (zEC12): *IBM zEnterprise EC12 Technical Guide*, SG24-8049

– For zBusiness Class (zBC12):*IBM zEnterprise BC12 Technical Guide*, SG24-8138

1.3 A new type of information technology: Workload centric

An IT workload can be described as one or more system resources used by one system or a combination of systems to complete one or more jobs at the same time or time period using system resources such as CPU, Memory, I/O.

Perhaps this definition looks a little simple, but even if we try to create boundaries and focus only on one workload on one system it is not as simple to understand as we imagine because each workload has its own characteristics, own behaviors, and it is not easy to separate system functionality from infrastructure.

– A classic IT workload approach such as “workload silos” no longer meets the new business and IT requirements such as workload deployments for analytics, big data, or secure commerce solutions on multiple components, especially now that cloud has been positioned as an answer to all IT requirements.

Currently, when deploying a workload, multiple components need to be wired together: application code, middleware, management agents, preexisting services, virtual machines,

data, disks, and networks. Workload availability and performance depend on the correct wiring as shown in Figure 1-2. The question becomes how to provision and maintain the components used by a cloud management system in order to achieve both agility and optimization for a varying set of workloads.

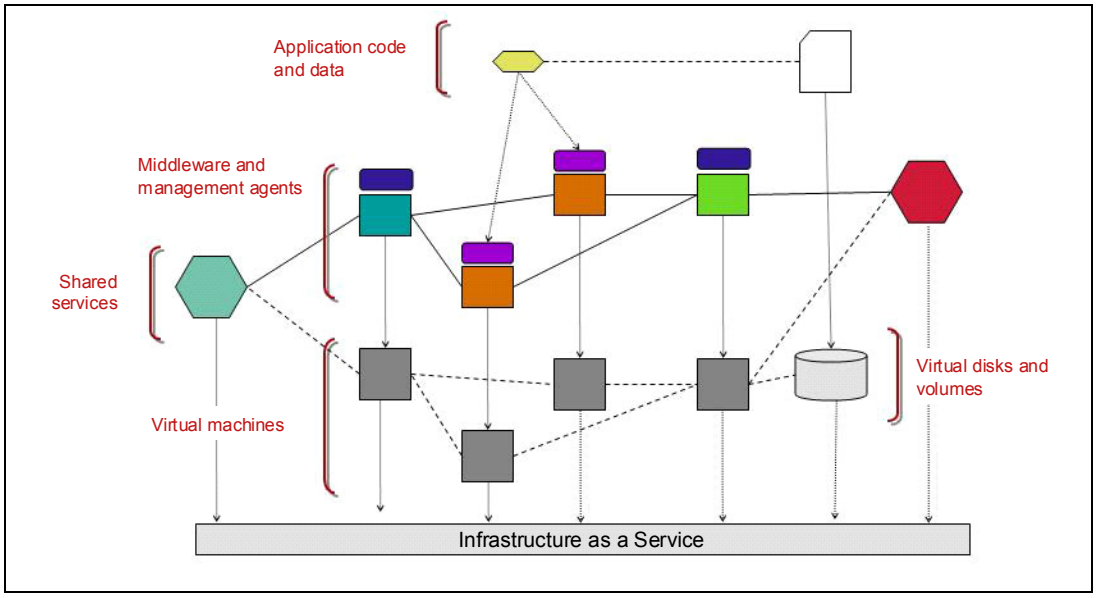


Figure 1-2 What is workload

The answer is to understand your dynamic workload patterns and dynamic automation composition for your heterogeneous system as well as autonomic and proactive management such as that shown in Figure 1-3.

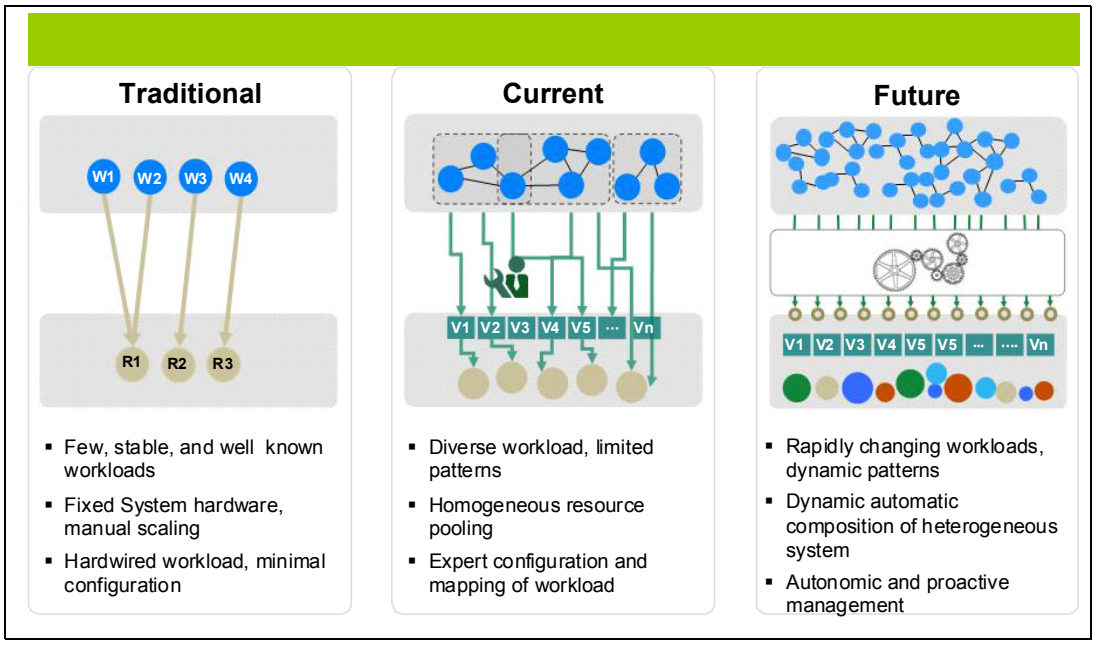


Figure 1-3 Future of Workload Management

The future, as shown in Figure 1-3, can be considered as the intersection of three main areas:

- Consumability

- Agility
- Efficiency

The intersection of these three main areas requires reconsideration of designing an IT workload, services, and infrastructure cloud solution. A Workload Centric Cloud provides abstraction and solution of workloads, services, and infrastructure and an end-to-end mapping of an intelligent software-defined environment ecosystem.

1.4 Workload-centric cloud

A workload-centric cloud can be simply described as a workload aware cloud solution. But this awareness is not covered by existing software as a service (SaaS), platform as a service (PaaS), or infrastructure as a service (IaaS) solutions because these simply run the “Cloud Enabled” workloads on their scalable, virtualized heterogeneous infrastructure and are not capable of being aware of the often varying workload definitions. They can only be aware of when system resources are busy or which system is running which workload. In this section, we discuss a different cloud architecture: a workload-centric cloud.

A workload-centric cloud offers a programmable and optimized way of continuously delivering and running workload.

As you can see in Figure 1-4 on page 8, this architecture is beyond what is commonly known as a *cloud architecture*. It is called a software-defined environment and it can help make the data center more customized and efficient by leveraging the diverse infrastructure according to workload types, business rules, and resource availability. The entire IT infrastructure is controlled not by hands and hardware but by software and workloads that are serviced automatically by the most appropriate resource.

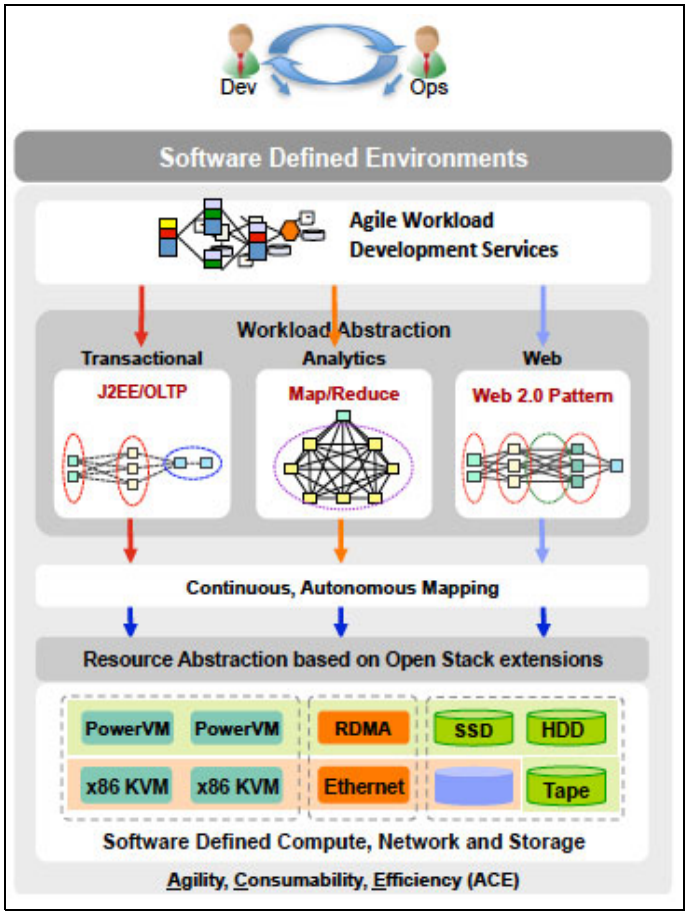


Figure 1-4 Workload-Centric Cloud Architecture

This architecture must be smarter and able to analyze and learn workloads and optimally manage everything.

One architecture to achieve a software defined environment is shown in Figure 1-5 on page 9. In this solution, workloads are defined and orchestrated using patterns and resources that are managed and deployed according to business rules and policies.

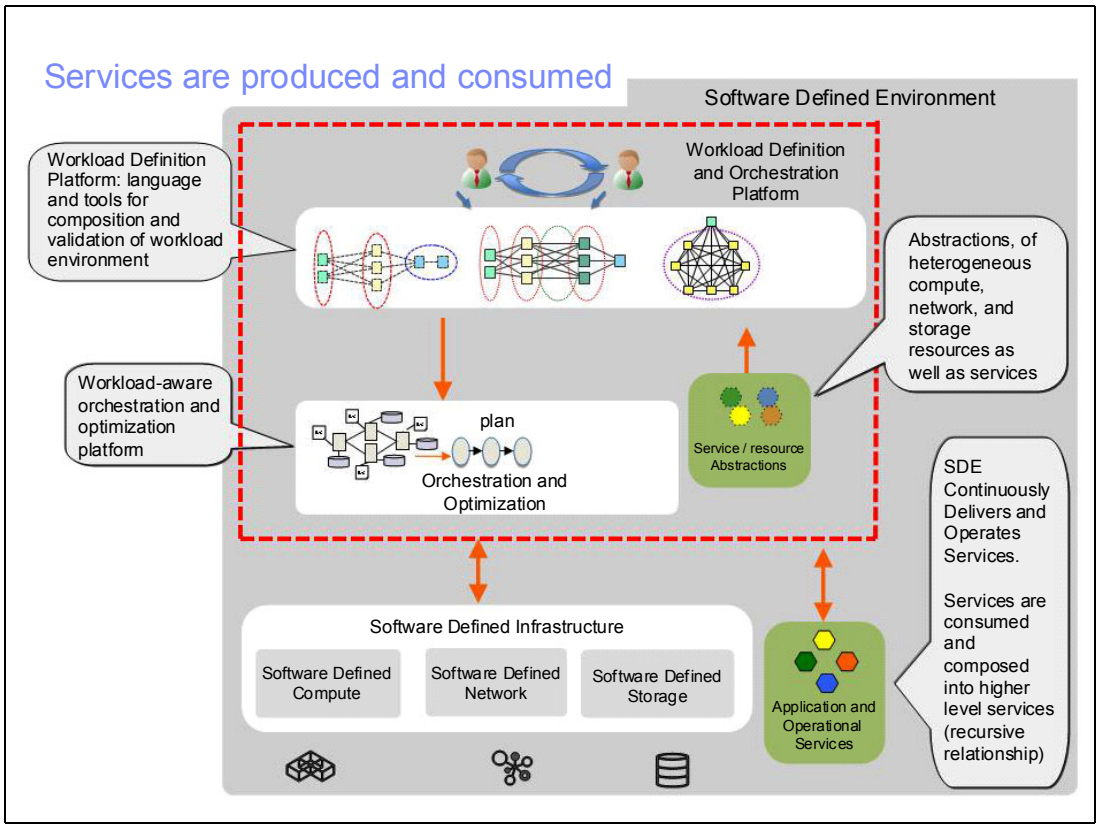


Figure 1-5 Software-Defined Environment for Workload Centric Cloud

In this architecture, first the workload definitions and resource abstraction is made known to the orchestration platform. Here, the business rules used to identify required service components are defined along with the necessary infrastructure resources.

Next, the infrastructure resources are managed by software-defined compute, network, and storage. Software-defined compute provides workload-aware infrastructure management and optimization. Software-defined networks for virtual environments, such as Linux on System z, create a virtual network for virtual machines and is decoupled and isolated from the physical network, which often can become a bottleneck. Software-defined storage allows for the management of huge surges in data driven by mobile and social technologies by pooling physical storage resources, regardless of the vendor, to improve utilization and do it in a cost-effective manner.

1.5 Enterprise cloud computing blueprint for System z

Enterprise computing encompasses all the various types of business software solutions including database management software such as IBM DB2®.

Enterprise cloud computing refers to the computer environment that delivers infrastructure, such as Web services, software, and platform services to your entire enterprise. Figure 1-6 on page 10 shows the IBM System z platform ready for each layer of cloud.

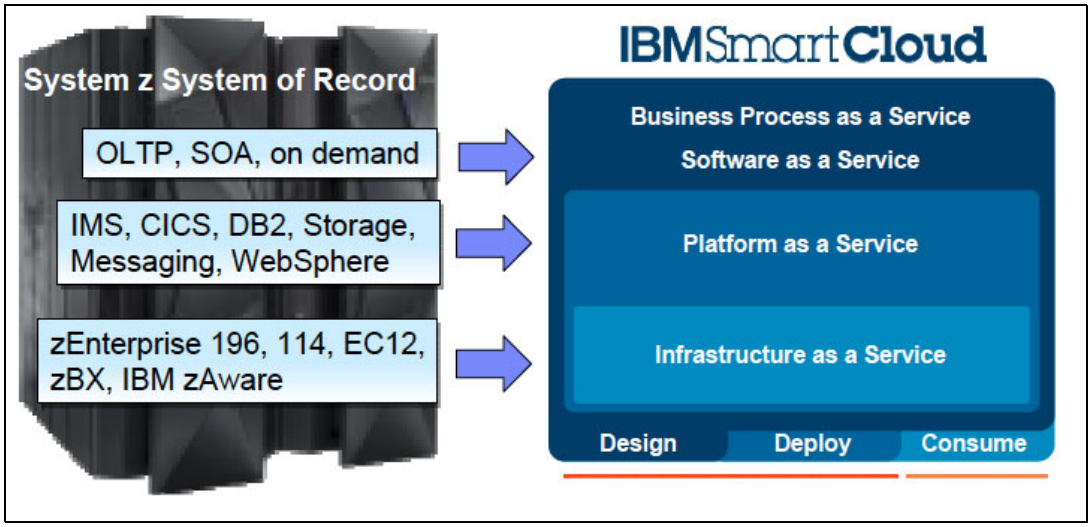


Figure 1-6 IBM System z is "cloud ready"

Virtualization and cloud blueprint for Linux on System z

The IBM Enterprise Cloud Solution for Linux on System z provides those delivering cloud services the ability to rapidly deploy a trusted, scalable OpenStack-based Linux cloud environment with the ability to start small and scale up to 6000 virtual machines in a single footprint on System z and can provide the following benefits:

Virtualization portfolio

The overall IBM virtualization portfolio includes:

- Infrastructure and Virtualization Management
  - zEnterprise: zEC12, zBC12
    - Massively scalable
    - Characterized by great economics and efficiencies
    - Highly secure and available
  - z/VM 6.3
    - Support more virtual servers than any other platform in a single footprint
    - Integrated OpenStack support
  - Linux on System z
    - Distributions available from Red Hat Enterprise Linux and SUSE Linux Enterprise Servers
  - IBM Wave for z/VM
    - A graphical interface tool that simplifies the management and administration of a z/VM and Linux environment

Entry-level cloud

At the entry level for cloud computing, Linux on System z can help simplify, automate, and manage your cloud better:

- Standardization and automation
  - Extreme Cloud Administration Toolkit (xCAT) on z/VM
    - Shipped with z/VM 6.3

- Allows customers to set up a rudimentary cloud environment, without acquiring any additional product
- Based on open source code
- Focused on upward integration to SmartCloud

– SmartCloud Entry
- A simple, entry-level cloud management stack
- Based on open source
- First tier in the SmartCloud suite of cloud management products

Advanced cloud

At the advanced level for cloud computing, Linux on System z assists in orchestrating your service lifecycle management and cloud optimization:

► Orchestration and Optimization

– Cloud Ready for Linux on System z
- Image-based cloud service delivery with integrated provisioning, monitoring, service catalog and service desk, storage management, and HA

– SmartCloud Provisioning
- Builds on functionality of SmartCloud Entry and adds middleware pattern support for workload deployment

– SmartCloud Orchestrator
- Builds on functionality of SmartCloud Provisioning and adds runbook automation

– Cloud Management Suite for System z
- Automated and simple orchestration with patterns for repeatable and controlled process, and an open flexible platform based on open standards.
- Immediate monitoring for IBM z/VM and provisioned Linux guest.
- Built in backup and recovery for private cloud storage.
- High security value, high I/O bandwidth, high availability, and greater scale with mainframe private cloud.

1.5.1 Empowered virtualization management: IBM Wave for z/VM

IBM Wave for z/VM is an intuitive virtualization management software product that provides management, administration, provisioning, and enables automation of Linux virtual servers in a z/VM environment. The graphical user interface home page is shown in Figure 1-7 on page 12.

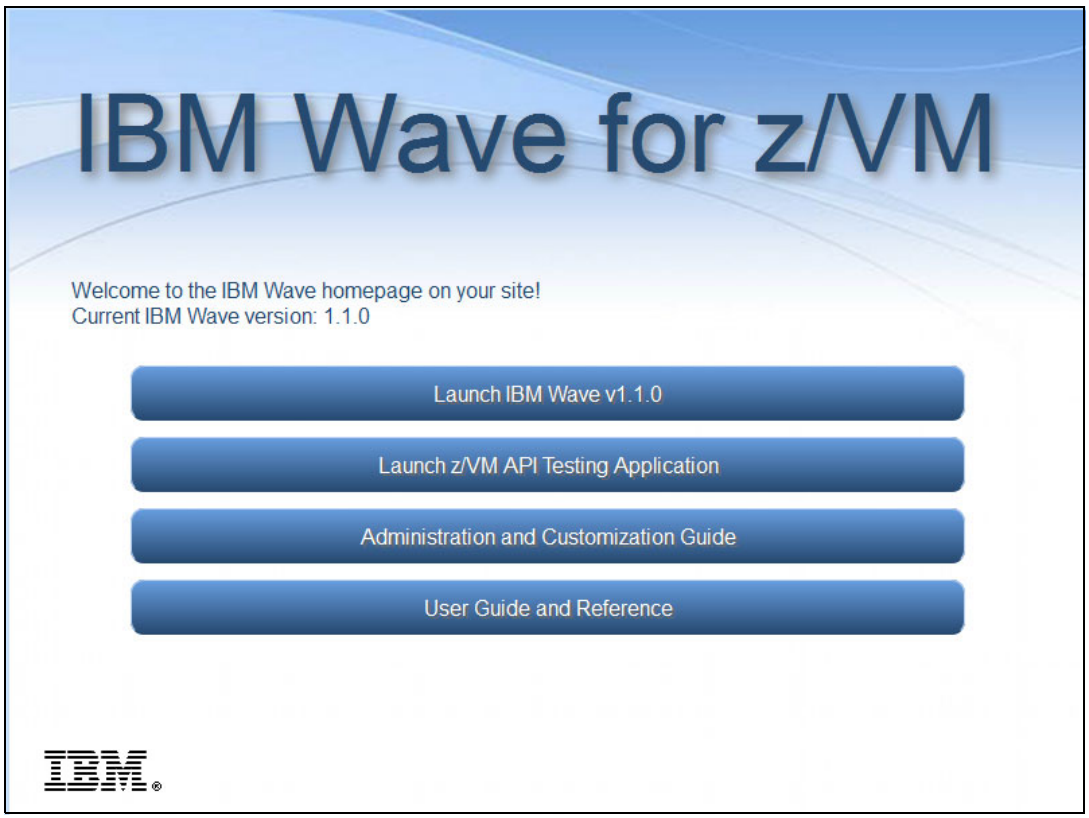


Figure 1-7 IBM Wave for z/VM home page

Organizations that want to expand their business using virtualization technology often find that they are limited by budget, skills, management complexity, and other challenges. With IBM Wave for z/VM (IBM Wave), IT organizations can unleash the power of z/VM virtualization, improve productivity, and simplify administration and management to jump-start their journey to a highly virtualized private cloud environment.

IBM Wave allows IT organizations and service providers to simplify and automate z/VM administration and management by using an intuitive graphical, content-rich interface to manage z/VM and Linux guests. IBM Wave is designed to simplify operations, drive productivity, reduce dependency on expert skills, and help accelerate the steps typically needed to transform a virtualized environment into a private cloud infrastructure. IBM Wave helps reduce the complexity of managing a scalable z/VM infrastructure, using a powerful content-rich user interface, to help manage even the most sophisticated virtualized computing environments.

IBM Wave is a virtualization management solution designed to help you intuitively manage both Linux virtual servers and z/VM. Using IBM Wave, you can automate repeatable tasks, such as provisioning new Linux servers, to help improve quality and reduce the potential for errors. IBM Wave helps you remove management obstacles associated with administering and maintaining Linux guests so you can direct your time to more business-critical tasks. It is designed to integrate seamlessly with z/VM and Linux environments. This integration helps you view and organize resources and manage your virtualized infrastructure more cost effectively by enabling your technical teams to be more self-sufficient.

Figure 1-8 on page 13 shows the IBM Wave main navigator.

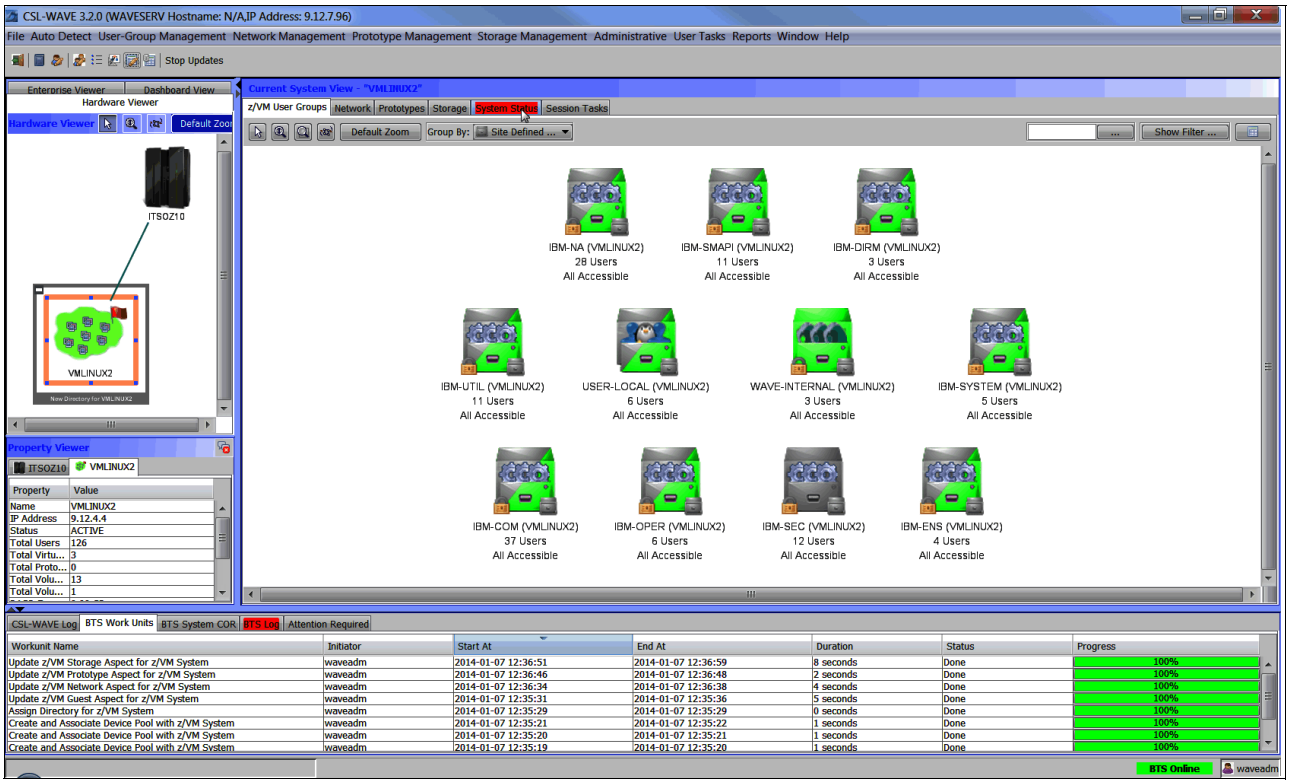


Figure 1-8 IBM Wave for z/VM main navigator

IBM Wave provides many features that can simplify and automate virtual server management:

- Dramatically reduces the learning curve to manage and administer z/VM and Linux guests to help improve administrator and system programmer productivity:
  - Readily manages large-scale implementations through a powerful, convenient content-rich, and task-intuitive user interface
  - Delivers enterprise-wide visualization and management of z/VM across LPARs and central processor complexes (CPCs)
  - Simplifies and automates day-to-day tasks and is ideal for organizations new to z/VM, organizations needing to better extend and take advantage of existing skills, or those that need to reduce the cost of management
- Reduces administration time and facilitates repeatable management of resources, drag-and-drop live guest relocation, and more:
  - Provides rapid cloning and provisioning of Linux guests on z/VM to help systems staff easily deploy Linux images
  - Easily provisions virtual resources including Linux guests, networks, and storage
  - Performs even complex tasks, such as cloning, using only a few mouse clicks
- Provides high-level performance monitoring and reporting by using the Performance Toolkit for z/VM:

  Delivers flexible reporting and comprehensive logging capabilities

Note: Additional performance monitoring of z/VM and Linux guests is available with IBM OMEGAMON® XE on z/VM and Linux.

- Helps identify, expose, and aggregate resources clearly, with intelligent content rich displays to depict resource use, performance, topology, and more
- Delivers an advanced security architecture with delegation of authority

Simplified approached improved virtualization management

IBM Wave provides a simplified approach to the management of IBM System z servers running z/VM and Linux. System z servers can be configured with z/VM instances that can run hundreds to thousands of virtual Linux servers with each one supporting individual workloads. IBM Wave's intelligent visualization of the virtual server environment and physical infrastructure provides intuitive management of physical servers, z/VM, Linux guests, and other resources. IBM Wave provides the necessary capabilities for complete virtual server provisioning, can readily scale to handle the most complex installations, and is an ideal solution to begin transitioning to a highly virtualized cloud infrastructure. With IBM Wave, you can rapidly gain insight into your entire virtualized infrastructure topology at a glance and also accelerate the path to using private clouds:

- Cloning and provisioning to accelerate the management of virtualized environments:
  - Provision Linux guests, network, and storage from a single user interface.
  - Capture and clone virtual servers across LPARs and CPCs.
  - Activate and deactivate z/VM guests in the current z/VM system.
  - Lock or unlock z/VM resources.
  - Create and configure virtual switches (VSWITCHes) and guest LANs.
  - Provide storage management and provisioning for z/VM and Linux.
  - Run shell scripts or REXX EXECs directly from the user interface for more customized management and provisioning.
- Intelligent visualization for effective management:
  - Use a rich user interface with active graphical views of all managed objects.
  - Attach digital reminder notes for display in a tooltip for managed objects such as:
    - z/VM guests, virtual networks, FCP-attached storage, disk storage components, scripts, and users. Context-aware technology integrates with these notes to help protect against unwanted actions.
  - Gain an informational view of objects with icons containing vital system information. An icon for a z/VM guest, for example, can indicate the type of operating system, the current status, and if the guest can be connected.
  - Use the innovative user interface to interact with the z/VM LPAR in a simple, intuitive way, such as connecting a virtual server to a virtual network or by dragging the connection between the virtual server and the virtual network.
  - Identify resource and relationship changes easily with automatic GUI updates.
- Management and configuration:
  - Display and manage virtual servers and resources, all from the convenience of a single graphical interface.
  - Provide convenient management of workloads across CPCs and z/VM LPARs designed to manage an unlimited number of z/VM instances.

- Support advanced z/VM capabilities such as SSI and LGR. Perform a live guest relocation of one or more z/VM guests.
- Help manage storage and networks, managing disk storage (ECKDTM and SCSI disks), Open Systems Adapters (OSAs), and HiperSockets connections, and use graphical management abilities. For example, IBM Wave can graphically display storage utilization across the z/VM system, and allow for further examination. Define and control network devices, such as VSWITCHes and guest LANs using a graphical view of the network topology.
- Use the intelligent user interface to connect various network objects within the network viewer and automatically detect, in advance, if that connection can be made.
- Discover z/VM resources and relationships across multiple LPARs, SSI clusters, and CPCs by using an agentless discovery facility that automatically refreshes updated views.
- Collect performance information from z/VM and Linux guests for monitoring of system behavior. If the Performance Toolkit for VM is installed, performance data can be viewed for every active z/VM guest, as well as globally for the entire z/VM system. Without the Performance Toolkit for VM, the z/VM **`INDICATE`** command is used, in which case CPU and performance data per z/VM guest will be unavailable. Additionally, the performance data will be less accurate in terms of sampling intervals because the **`INDICATE`** command uses a broader interval than the Performance Toolkit.

  Note: IBM OMEGAMON XE on z/VM and Linux can be used to provide more detailed performance monitoring and alerting.

- Administration:
  - Administer resources such as z/VM guests, storage, and networks. Incorporate flexible filtering capabilities to group and manage resources matching your criteria for maximum flexibility and to speed management tasks.
  - Delegate administrative capabilities to more precisely meet the needs of various administrative teams, all from the convenience of a single, powerful user interface.
  - Assign a z/VM account to one or more z/VM guests, assign one or more z/VM guests a value for a particular attribute, or assign z/VM guests to a project. Viewing by project helps you manage the way that you view resources.
  - Use role-based management to control the management of z/VM and also allow delegation of administrative responsibility.
  - Provide reporting through the built-in report writer to produce information about guests, users, LPARs, z/VM guests, disk storage, and more.
  - Deliver security management with scopes and permissions set for each user. Support is provided for IBM RACF® Security Server and other security products.

IBM Wave dramatically reduces administrative complexity

IBM Wave helps IT staff provision resources and images easily to dramatically reduce administrative complexity and reduce needed skills. It is designed to help IT reduce labor cost and improve the quality and efficiency of operations, with centralized control to manage even the most complex and large server environments, consistently and according to your defined policies. It can help you manage systems and resources, and improve the quality of management, reducing the chance of operator error and expensive corrections.

IBM Wave can provide the following benefits:

- Help you manage and organize resources at a glance, providing a compelling user interface that provides built-in management information designed to help reduce management steps
- Allow you to scale your management reach to manage a higher level of virtualization with the skills you already have
- Help simplify z/VM management and administrative tasks, shielding users from the complexity of z/VM commands
- Help administrators and operators visualize resources, storage, and networks, manage user accounts, and more
- Give you visibility into your z/VM environment with built-in reporting and monitoring
- Help shorten the time to deploy Linux servers as guests of z/VM
- Support LGR through drag and drop
- Help accelerate your steps to a private cloud model with simple cloning capabilities
- Help reduce the costs and effort of virtualization management, helping organizations optimize resources and focus on more business-critical tasks

2

Analyze and understand

This chapter outlines some of the points you need to consider before making the decision to migrate to the Linux on IBM System z platform. This chapter also provides a description of total cost of ownership (TCO) as well as assists in analyzing which workloads would make good candidates for migration. Additionally, we touch on the financial benefits of migration.

2.1 Total cost of ownership analysis

Many CIOs recognize the return on investment (ROI) in the information technology of their companies, but at the same time they are frustrated by an increasingly costly IT infrastructure. There are many reasons for these costs, some of which are the annual costs of software licensing, power, cooling, and ongoing support. The complexity of environments in most cases determinates the magnitude of these costs.

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

A credit union in Brazil migrated their complex x86 infrastructure to Linux on System z virtual servers on an IBM zEnterprise server. This credit union needed a flexible, secure, and scalable IT infrastructure that would support reliable 24 x 7 service and mobile access.

The credit union estimated that it would require more than 400 stand-alone Intel processor-based servers and 6 million kWh of additional electricity each year to support the workload managed by the Linux virtual servers on its z196 mainframes. With IBM System z, they are now spending 400 percent less in energy costs than if they had a distributed environment. The results of this migration saved them USD 1.5 million per year in energy costs alone.

The case study regarding this was published in May 2013 and can be found at:

http://ibm.co/1wbcHtq

2.2 Choosing workloads to migrate

When you have made the decision to migrate and consolidate, the next step is to examine which workloads would be good candidates to be migrated.

Several variables must be considered, such as:

- Associated costs
- Application complexity
- Service level agreements
- Skills and abilities of your support staff

Start with a not overly complex application that has a low service level agreement (SLA) and a staff that has the associated skills.

In the case of home-grown applications, ensure that you have the source code available. Regarding the operating system platform, even a workload from a different platform can be migrated but start with servers running Linux. This will substantially increase success. Applications that require close proximity to corporate data stored on IBM System z are also ideal candidates, as are applications that have high I/O rates because I/O workloads are offloaded from the IFL by the z12 Service Assist Processor (SAP)[1].

IBM System z has a very powerful processor with a clock speed of 4.2 GHz (zBC12) or 5.5 GHz (zEC12). Because System z is designed to concurrently run disparate workloads, it is important to remember that some workloads that required dedicated physical processors designed to run at very high sustained CPU utilization rates may not be optimal candidates for

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

migration to Linux on System z. This is because workloads that require dedicated processors will not take advantage of the virtualization and hardware sharing capabilities. An example of such an application might include video rendering, which requires specialized video hardware.

Chapter 6, “Migration analysis” on page 57, provides an in-depth analysis of the process of determining the most appropriate applications to migrate to a Linux on System z environment.

2.3 Analysis of how to size workloads for migration

One of the challenges of migration is to determine the resources required on the target platform 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 some factors:

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

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

Important: Others factors must be considered to get a complete TCO: floor space, energy savings, scalability, security, and outages. For a more accurate sizing study, contact your IBM representative.

One evaluation tool that IBM offers to the customers is IBM Rehosting Applications from Competitive Environments (RACE) tool.[2]

The inputs for the RACE tool are:

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

The outputs from the tool are:

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

[2] RACE is a tool that helps IBM to understand the customer environment. To arrange a RACE study, contact your IBM representative.

- Total cost of ownership (TCO) analysis of the various configuration options (based on cost inputs in the model)

IBM offers another tool to help Chief Information Officers (CIOs) in determining the System z resources required to consolidate distributed workloads. It is a self-help web tool named *IBM Smarter Computing Workload Simulator* that gives a fast and easy way to view areas of potential savings and efficiency through the lens of IBM Smarter Computing systems and technologies.

More details are available in the IBM Smarter Computing Workload Simulator URL:

http://www.ibm.com/smarter-computing/zz/zz/workload-simulator.html

2.4 Financial benefits of a migration

In addition, the benefits mentioned in the IBM System z allows costs reduction through the environment sharing. The analysis of these benefits includes:

- Reduced risk of downtime because the redundancy of the hardware and z/VM features like single system image (SSI) and Live Relocation.
- Save software licensing: Databases, operational systems, application server, and management software in a current distributed server farm can be licensed more cost effectively using the specialized System z Processor Integrated Facility for Linux (IFL).
- Save energy and be green: When you have hundreds or thousands of servers consolidated in a single box, the energy and cooling costs can be reduced until 75% in comparison to distributed environment.
- Save costs of on-going support: The complexity of maintenance of the environment is decreased since you have many virtual servers in a single box.

The cost savings arise because Linux on System z is treated by most software vendors as a distributed system, and software is usually charged by the core. Because an IFL is classified as a single core, and has high processing power, there could be significant savings by consolidating multiple distributed servers to an IFL. Figure 2-1 on page 21 shows an example company that has 45 virtual servers and uses only 14 licenses.

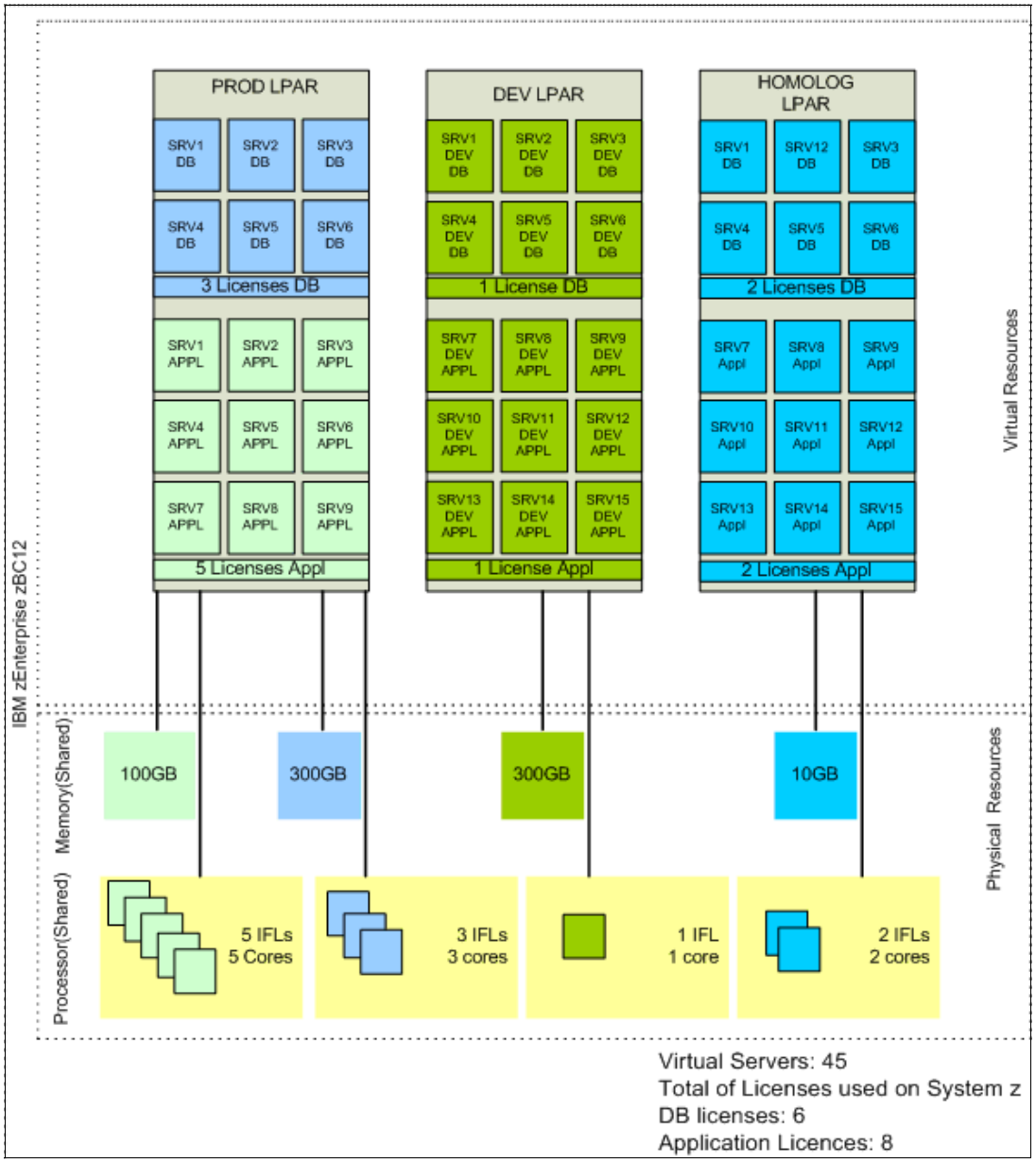


Figure 2-1 Example company saving licenses

To determine the potential software cost savings of a migration to IBM System z, you can initiate your study with the self-help web tool called *Alinean IBM System Consolidation Evaluation Tool*. Figure 2-2 on page 22 shows one panel of the tool.

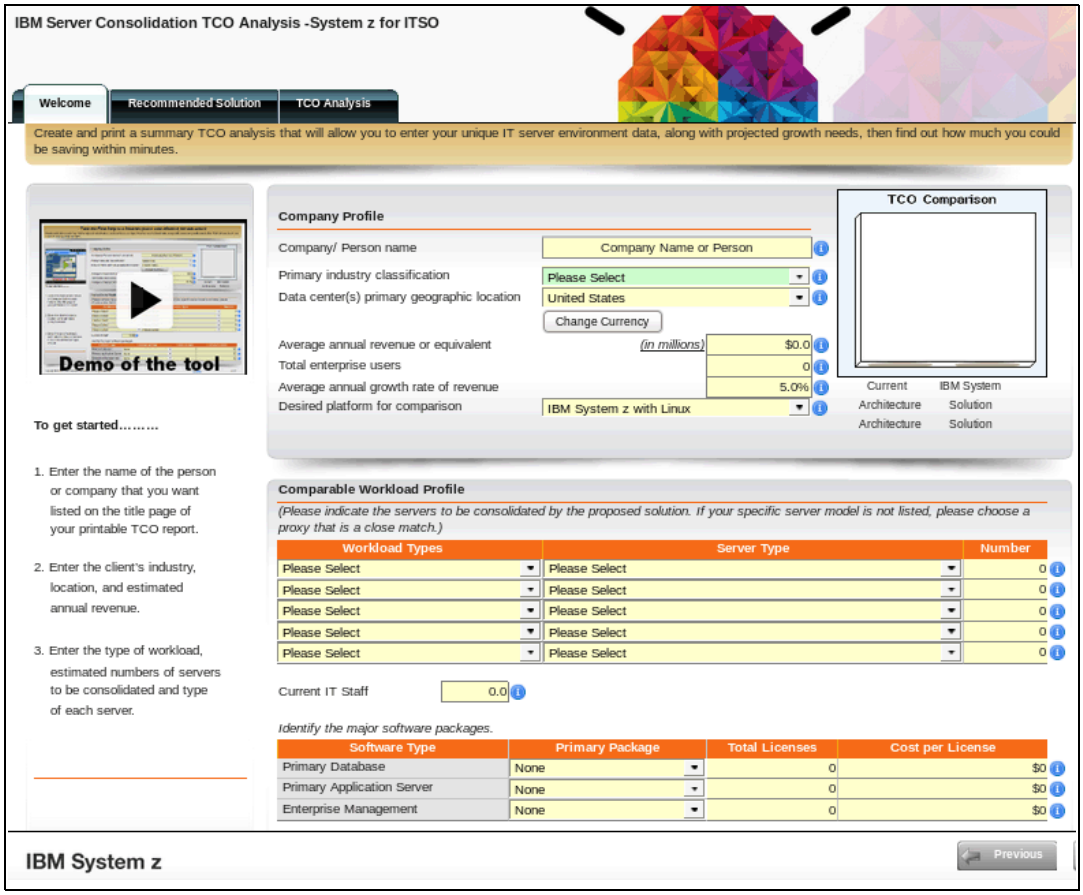


Figure 2-2 Alinean IBM System Consolidation Evaluation Tool: Original panel

More details about the tool can be found at the Alinean website:

https://roianalyst.alinean.com/stg

Note: For an accurate TCO study, contact your software vendor or IBM representative to understand its policies and pricing regarding application consolidation on System z.

3

Virtualization concepts

Virtualization is a highly prized capability in the modern computing environment. Virtualization on System z offers industry-leading and large-scale proven Cloud and IT optimization capabilities to drive down the costs of managing and maintaining the tremendous proliferation of servers in today's technology infrastructures.

This chapter provides helpful information about virtualization, particularly to compare and contrast the virtualization concepts of IBM mainframe computing with those commonly used by x86 distributed systems. Surely the two have many concepts in common, yet other concepts are very different. This brief comparison provides terminology, vocabulary, and diagrams that prove helpful in planning to migrate workloads to System z.

3.1 The demand for virtualization

As the computing environment grows in size and complexity, the sprawling infrastructure becomes more difficult to manage. As more physical servers are added to the environment, the resources such as CPU, RAM, and DISK are too easily wasted and cannot be efficiently used. Virtualization turns physical hardware into logical resources that can be shared, shifted, and reused. One of the most highly prized features of virtualization is the ability to dynamically dedicate more virtual resources, such as CPU, RAM, and DISK, to a virtual host while the virtual host is running, greatly easing the system administration tasks of scaling the supply of services to meet demand.

Systems administrators rely on virtualization to ease and facilitate the complex work of managing increasingly complex environments. IT managers look to virtualization to address the ever increasing demand for more computing power from customers while accommodating shrinking IT budgets.

The growing number of physical servers also increases the amount of power consumed in the data center. Virtualization helps to reduce the amount of electricity consumed, and hence reduces the cost to compute. Those with aspirations of a “green” data center are similarly more satisfied when virtualization is used.

Much has been made in recent years about virtualization, with research suggesting that more than half of all workloads in the data center are virtualized[3]. Despite its more recent hype, virtualization has existed in advanced computing systems for quite a long time. The conception of virtualization began in the late 1960s as IBM introduced Control Program (CP)-67. This innovation began humbly, but quickly grew to be a defining feature of IBM mainframes. (See more about IBM virtualization history in section 3.4, “Linux on z/VM” on page 26.)

3.2 IBM System z virtualization

Today, the IBM Process Resource/Systems Manager (PR/SM) facilitates virtualization of all physical resources to its logical guests. PR/SM is a standard feature on all System z servers. It is the hypervisor of System z, dividing the physical system resources into isolated logical partitions (LPARs). Each LPAR is effectively an independent system running its own operating environment, with its own CPU, RAM, and other resources. The zEC12 can be configured to run as many as 60 LPARs, each having its own set of logical resources. PR/SM in zEC12 has even more advanced features over its predecessors for better fine-tuning and optimization of guests’ sharing of system resources. (See *IBM zEnterprise EC12 Technical Guide*, SG24-8049 for much greater detail about the zEC12.)

Said another way and more briefly, System z is all about virtualization. System z is indeed the most versatile virtualization platform available.

A common practical application of virtualization on System z is to divide software development and applications deployment into two discrete activities: test, and production. With System z LPARs, it is routine to maintain all of the pre-release development of a system contained within one LPAR, such that anything that goes wrong in this test environment will not adversely affect anything that is running in the production environment. When development is completed, the workloads that have been developed in the test environment can be migrated to the production environment. One LPAR would be dedicated to

[3] Nemertes Research, “Data Center Dynamics”, Ted Ritter, 27 September 2011

development and testing activity, while a separate LPAR would be dedicated to the production environment.

Although the zEC12 is capable of running 60 LPARs, there is greater flexibility to virtualize even more of System z by using the features of z/VM. Section 3.4, “Linux on z/VM” on page 26 describes z/VM virtualization in more detail.

3.3 Typical x86 virtualization

Briefly stated, virtualization allows a single physical server to host numerous logical servers, sharing the physical resources in such a way as to allow all the logical servers to accomplish more than the single physical server could on its own, while maximizing the effective use of the physical resources. In such a virtual environment, the physical server is commonly called the “host” system while the several logical servers are known as “guests.” Although there are several software solutions in the industry that use variations of these terms, this publication will simply use the terms “host” and “guest” as defined above.

Figure 3-1 shows a very simple but typical way that systems administrators set up virtual services in a distributed x86 environment. A physical server employs virtualization software (such as KVM or XEN) to install and run a Linux guest. Figure 3-1 displays a physical server (host name “x86host1”) with three separate virtual Linux guest operating systems contained on this physical host. The physical server has a fixed amount of CPU, RAM, as well as physical access to Disk and Network resources. The virtual guests are allocated CPU, RAM and Disk resources as a subset of what is available from the physical server, and the Network resources are all equally shared by the guests and physical host.

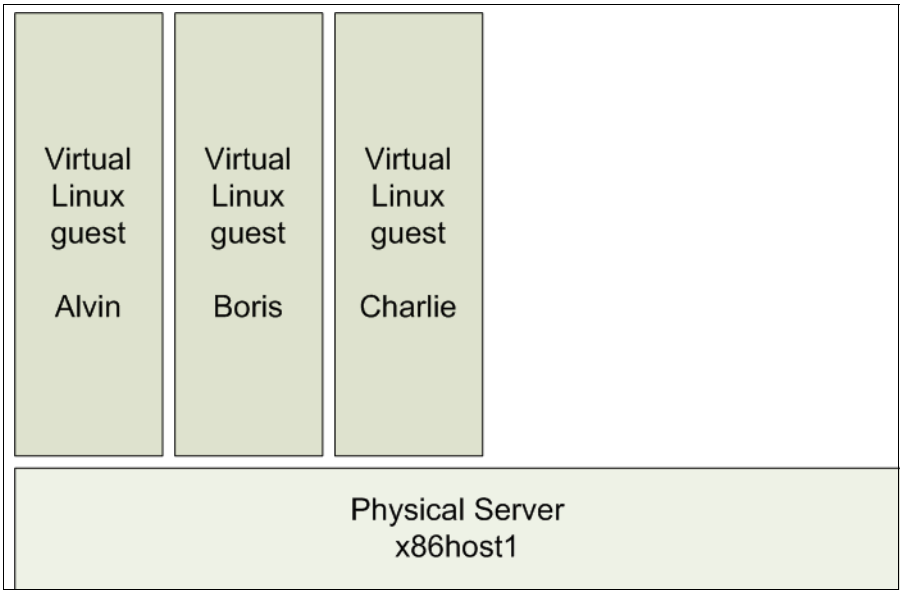


Figure 3-1 Typical x86 virtualization scheme

In a typical x86 deployment of virtual services, the physical servers are generally deployed in pairs or trios, often called *clusters of servers*. The clusters provide for some standard level of high availability such that if one of the physical servers were to fail, another would be able to take over the running workload with negligible interruption.

3.4 Linux on z/VM

Like virtualization systems deployed using x86 clusters, System z accomplishes these many virtualization functions. Unlike x86, System z does so in a very consolidated and comprehensive way. All of the extensive capabilities of the System z hardware are available for virtualization through the PR/SM hypervisor. But even more extensive capabilities exist when System z virtualization is facilitated by z/VM.

As mentioned in section 3.1, “The demand for virtualization” on page 24, virtualization on the IBM mainframe has evolved significantly since the 1960s, culminating in the development of z/VM. Figure 3-2 shows the developmental history of z/VM.

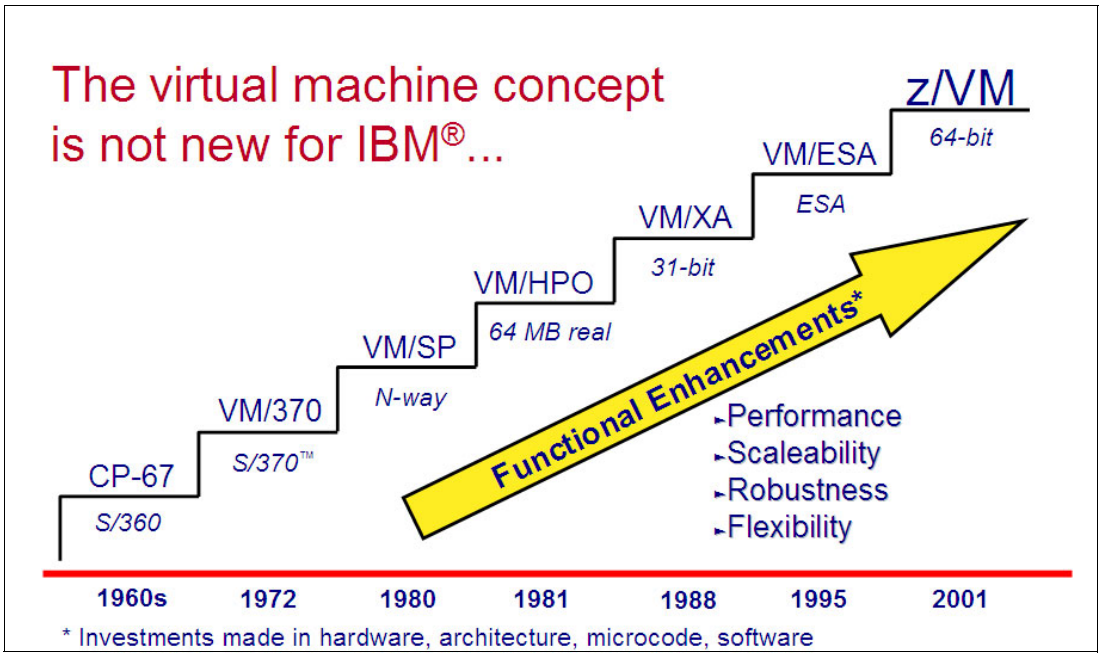


Figure 3-2 Developmental history of z/VM

z/VM is an operating system that facilitates and further enhances the PR/SM hypervisor. A systems administrator may likely know very little about the details of PR/SM. z/VM exposes all of the features and interfaces of the PR/SM hypervisor while further protecting and isolating each virtual machine (VM) from each other and from the physical resources. Make no mistake, Linux can easily operate in a logical partition (LPAR) afforded the general capabilities of PR/SM, and doing so may be an appropriate consideration. However, the virtualization capabilities of z/VM provide added isolation, resource sharing, and resource management features that many systems administrators require.

Running in its own LPAR, z/VM version 6.3 allows the system administrator to:

- Manage guests of various types, whether the guests are more traditional IBM z/OS® instances or Linux
- Maintain z/OS workloads, such as IBM DB2, relatively close to the Linux workloads, for better performance between the workloads and easier management
- Exceed the limit of 60 guests that would otherwise be limited by the number of LPARs allowed in zEC12

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

Using the earlier diagram of a typical x86 virtualization system as a model (Figure 3-1 on page 25), the picture in Figure 3-3 depicts a similar virtualization system as it relates to System z and z/VM.

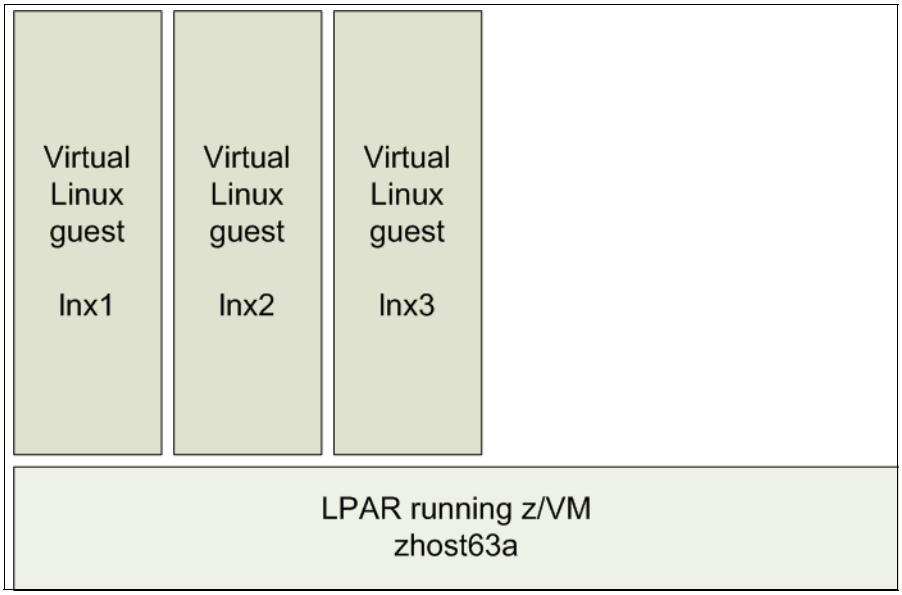


Figure 3-3 Simple z/VM with virtual Linux guests

Running Linux as a guest under z/VM is simple, and effectively no different from running z/OS under z/VM. SUSE and Red Hat both provide Linux distributions that run on IBM System z hardware. The work that IBM has done in collaboration with these major Linux distributions has provided code within the kernel and the core utilities tied to the kernel to facilitate the operation of the Linux kernel with System z hardware. Figure 3-4 on page 28 illustrates the work that IBM has contributed to the Linux kernel and the Linux operating system to allow Linux to run on System z.

Note: All recent Linux distributions that use GNU Linux kernel version 2.6 or later are technically capable of running on System z. Just keep in mind that the Linux kernel by itself does not make an operating system. To really have a Linux distribution that can run on System z, the distribution must also have binutils, glibc, and other core components built for System z.

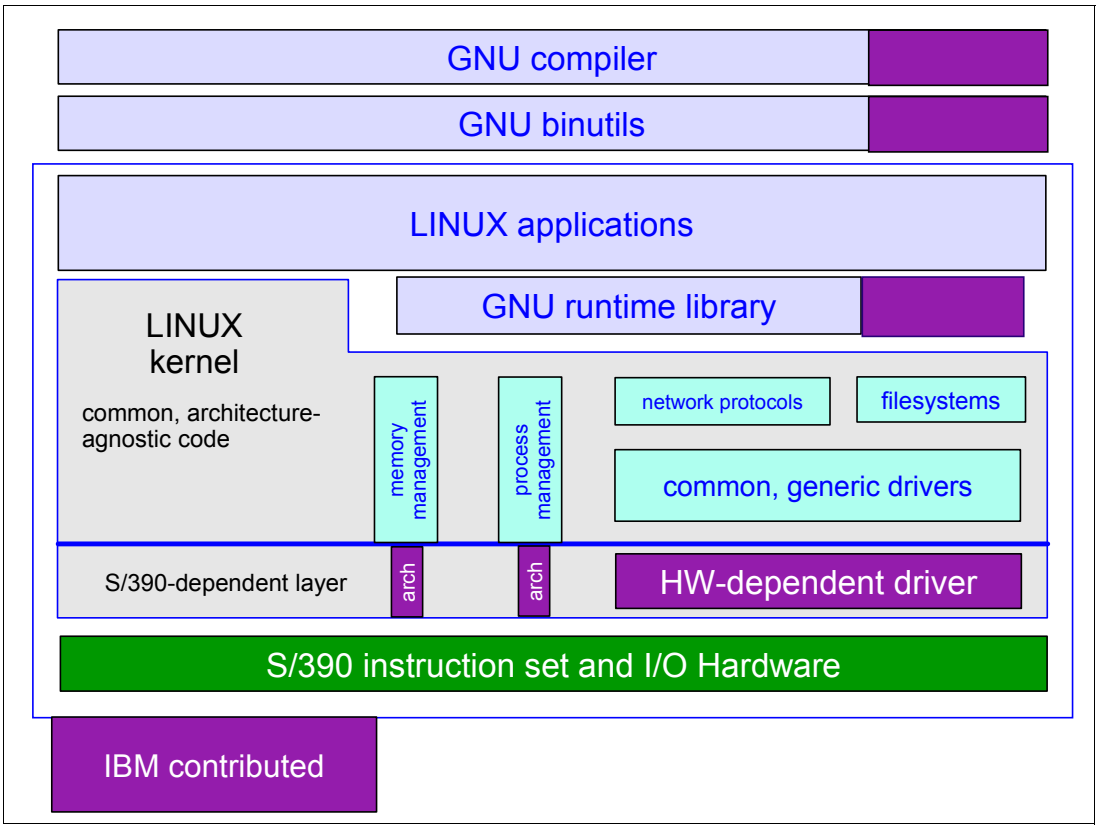


Figure 3-4 Linux kernel and core utilities characteristics on System z

Running Linux on z/VM makes deployment of services faster, often being able to spin up a running Linux server in a matter of minutes. Linux servers can be built, cloned, and deployed within the System z and z/VM infrastructure without the pain of requisitioning, purchasing, mounting, and wiring a new physical server. Development teams who need a new server for a proof of concept can set up and tear down a test environment over and over again with no impact to running production systems. New projects that have completed their development and are ready to be moved into a production environment can do so without the expense of moving or scaling physical resources. Production services can be effortlessly scaled to match the demand, and accommodate all manners of change management.

3.5 Single system image and live guest relocation

A critical responsibility of systems administrators is ensuring that all systems are running the latest operating systems software and that all maintenance and security fixes have been applied. Protecting the system from unexpected downtime and from security vulnerabilities, ensuring that applications are running at the latest patch release levels, and balancing loads across a diverse infrastructure are all tasks that keep systems administrators awake at night. This is a particularly troubling challenge in the data center, where downtime must be minimized and maintenance windows are scarce.

General virtualization does not accommodate the ability to take down the host system to apply maintenance updates. If you have a critical problem on the host system, all the guests running on the host will likewise need to be taken down in order to reboot the host. This is clearly not an ideal scenario.

The z/VM single system image (SSI) is a clustering technology that provides multiple, redundant host systems upon which virtualized guests run. Each member of the SSI cluster shares common pool DASD volumes, minidisks, network devices, and user data. Ideally the cluster members would be contained on separate CECs for optimum safety if a failure were to occur, although running the members on the same CEC is also feasible. The members of the SSI cluster are managed together.

Coupled with SSI is live guest relocation (LGR), which facilitates the relocation of a Linux guest from one member of the SSI cluster to another. This relocation happens nearly instantaneously, without the Linux guest having any knowledge of the relocation. Network processes and connections, disk operations, and user interactions on the Linux guest are completely unaware that the underlying infrastructure has moved to a different “physical” environment. Figure 3-5 depicts a very simple representation of an SSI cluster composed of two members, zhost63a and zhost63b. zhost63a is currently hosting three Linux guests while zhost63b hosts a single Linux guest.

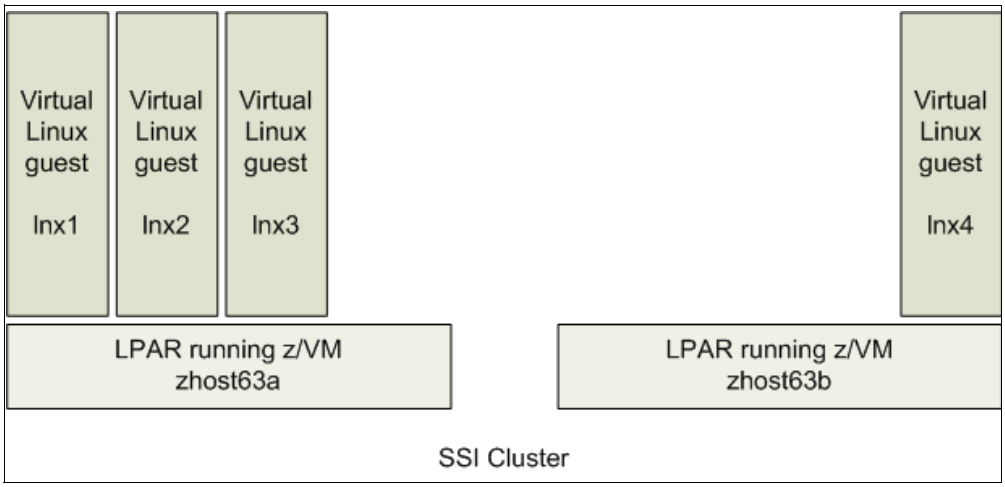


Figure 3-5 Simple representation of SSI cluster before live guest relocation

The relocation of Linux guests from one SSI member to another makes it possible to perform maintenance on the individual SSI cluster members without disrupting the services running on the Linux guests. With all Linux guests relocated away from an SSI member, that SSI member can now be updated and rebooted, with no impact at all to any running guests. When the maintenance on this SSI member is completed, Linux guests can be relocated back to their original host members. Perhaps all Linux guest systems could be relocated to this SSI member while similar maintenance is performed on other SSI members in the cluster.

An additional benefit of SSI and LGR is the ability to relocate workloads to accommodate a more balanced use of system resources. If an SSI cluster currently contains a configuration of multiple Linux guests that are overusing the network, a portion of the guests could be relocated to a different member of the SSI cluster where network utilization is lower.

Figure 3-6 on page 30 shows that a Linux guest has been relocated from zhost63b to zhost63a with no interruption in the services that are running from the Linux guest. Now that there are no guests running on zhost63b, the host can rebooted. After rebooting zhost63b, Linux guests can be relocated back onto zhost63b.

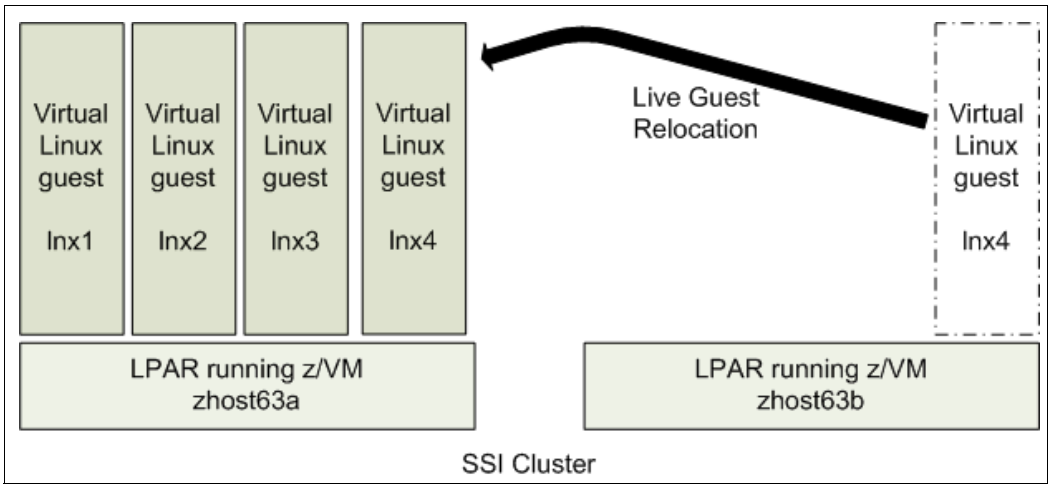


Figure 3-6 A simple representation of live guest relocation of a Linux guest

This convenient mechanism of relocating guests with LGR among the various SSI cluster members is precisely the flexibility that the systems administrator needs in order to keep systems up to date while minimizing downtime, while also giving the administrator the ability to move workloads more freely within the infrastructure to make the best use of resources.

More to the point, knowing that z/VM, SSI, and LGR can be used in this way makes the decision to migrate workloads to Linux on System z all the more compelling.

This section provides merely a brief overview of SSI and LGR. There are several IBM Redbooks publications that describe SSI and LGR in greater detail. Such titles include (but are not limited to):

- *An Introduction to z/VM Single System Image (SSI) and Live Guest Relocation (LGR)*, SG24-8006
- *The Virtualization Cookbook for IBM z/VM 6.3, RHEL 6.4, and SLES 11 SP3*, SG24-8147
- *Using z/VM v 6.2 Single System Image (SSI) and Live Guest Relocation (LGR)*, SG24-8039

3.6 z/VM operating system components

There are two separate components that are part of z/VM that aid PR/SM in managing the virtualization environments. These are command line operating environments that give the system administrator control over the hypervisor. This section describes and explains these components.

Control program

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

When a Linux guest logs on to a z/VM session, it is starting its own CP session. For production systems, this is usually done automatically when the z/VM system is initially

loaded (IPLed) or booted. There is an entry in the z/VM directory for each virtual machine that can be started.

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

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

Conversational Monitor System

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

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

Both CP and CMS give the system administrator a more direct route to manipulating the available resources for the benefit of the Linux guest.

3.7 Virtualized resources

A key feature of System z is how resource utilization is optimized and maximized. In the current environment of distributed computing, the RAM, the CPU, or the disk is underutilized most of the time that the server is running, but is necessary to have available when the server is under peak load. With System z, a considerable amount of “overcommitting” is possible, such that RAM, CPU, and I/O can adequately accommodate the workload when the workload needs it, and the resources can be diverted elsewhere, without having to commit specific resources to any one workload. Although resources can be rigidly committed to a specific workload, it is the flexibility of the virtual resources that is so appealing. Overcommit is quite powerful for z/VM virtualization because typically not every guest will need all of its allocated resources all at the same time.

3.7.1 Virtualized CPU

Whether running Linux directly in an LPAR or in z/VM, the guest needs processing power. System z offers a few choices of processors. When Linux is the chosen operating system for the workloads, system administrators generally choose to employ an Integrated Facility for Linux (IFL) Processing Unit (PU) due to its attractive price point. Where a mix of Linux and other systems coexist, a standard central processor (CP) is the more appropriate choice. (Other specialty CPs exist, such as the zAAP for running dedicated Java workloads.)

The number of IFLs or CPs on the machine reflect directly on the performance of the Linux guest running in an LPAR. The number of virtual CPUs allocated to a single Linux guest should not exceed the number of logical CPUs allocated to the LPAR. For example, if the

LPAR has four IFLs, then do not allocate five virtual CPUs to a single Linux guest machine. If a situation occurs where the Linux guest uses 100% of the CPUs, that will adversely affect the entire LPAR.

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

3.7.2 Virtualized disk

System z disk storage is commonly referred to as a *direct access storage device* (DASD). Mainframe system administrators have a long and unique history with DASD. Traditionally, IBM System z has supported only IBM 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.

The ECKD devices are defined as one of three 3390 DASD models, each of different sizes. The models, and their capacity as measured in cylinders and in megabytes, are listed in Table 3-1.

Table 3-1 Some standard 3390 DASD models

Model	Cylinders	Storage Capacity
Model-3	3,339	2.83 GB
Model-9	10,017	8.51 GB
Model-27	32,760	27.84 GB
Model-54	65,520	55.68 GB

Although DASD is common in the mainframe world, it is not well known in the distributed x86 world. Many x86 computing environments will have disk storage maintained in Storage Attached Networks (SANs) and other similar, external storage arrays. But System z is fully capable of using disk storage from a SAN or network-attached storage (NAS). In many cases, the system administrator chooses to maintain the data of a particular application on the storage array while choosing to migrate the application workload to System z. Whether maintaining the data on a SAN or migrating the data to the System z storage, the virtualized disk can be readily accessed by the workloads in the virtual environment. In many cases, leaving the data intact on the SAN will ease and simplify the migration effort.

With z/VM and Linux on System z, disk device support is expanded to fixed-block architecture (FBA) DASD and also to Small Computer System Interface (SCSI). FBA and SCSI disks are connected to System z via the Fibre Channel Protocol (FCP). The connection to SCSI devices is managed by the zFCP Linux module driver. The SCSI devices are usually dedicated to the Linux guests.

It is good practice to use ECKD DASD to boot Linux and for static data, and use SCSI FCP for data applications. DASD is well suited to booting and hosting the operating system, but systems administrators may find that performance from the SCSI FCP disks is better adapted for data.

Disk storage, by itself, is not really a virtual resource. The bits and stripes on the disk do not have the same characteristics for virtualization that memory does. Disk is a more permanent

resource than memory. Nevertheless, allocating free disk space for a workload should be just as flexible and effortless as allocating virtual processing power or virtual memory. A competent hypervisor facilitates the management of disk storage.

For a more detailed description about disk storage, see 6.2, “Storage analysis” on page 69. For more information about z/VM and disks, refer to *Introduction to the New Mainframe*, SG24-7316.

3.7.3 Virtualized memory

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

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

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

The keys to determining the appropriate virtual memory size are to understand the working set for each virtual machine, and to ensure that the Linux images do not have any unneeded processes installed. Another recommendation is to use VDisks for swap, as described in “Swap device consideration” on page 35.

Memory management features

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

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

These are features that simply are not possible in a distributed x86 environment. Only z/VM can provide these versatile features, dramatically reducing the amount of physical memory required to maintain a similar set of workloads.

CMM

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

CMMA

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

NSS

NSS allows virtual guests to share a read-only copy of a single operating system such as CMS or Linux. The benefit of this feature is that only one copy of the operating system resides

in storage accessible by all virtual machines. This decreases storage requirements and simplifies maintenance.

DCSS

DCSS allows virtual machines to share reentrant code for applications, such as Oracle, which also reduces overall storage requirements. Figure 3-7 illustrates how both NSS and DCSS work. There is one copy of the application in real storage and Linux guests use this single copy. The NSS copy of Linux is also shared by all virtual guests.

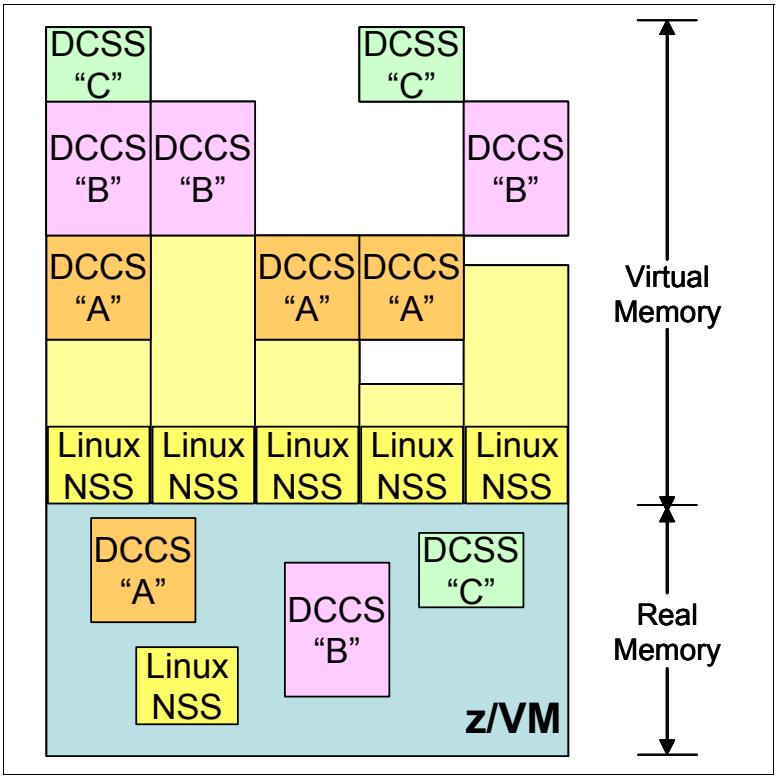


Figure 3-7 DCCS and NSS shared by multiple Linux virtual guests

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

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

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

Swap device consideration

Understand that the concept of “swapping” is different today than when it was invented, back when large amounts of RAM were ridiculously expensive. Modern operating system memory technology is more focused on paging than swapping. As suggested in the note a few paragraphs back, it is a best practice to commit a specific amount of virtual memory to each Linux guest to accommodate no more than its intended workload, and to fine tailor this amount of memory precisely so that paging does not normally occur. This may not be realistic, but it is a principle to seriously consider.

In the absence of the perfect memory configuration, and when workloads demand significant swapping, the ideal is to provide a VDisk device for this purpose. VDisks are virtual disks allocated in memory, and they become a fast swap device for Linux. Swapping to a VDisk in memory is far more efficient than swapping to DASD, and it is generally less expensive, too, considering all factors. The Linux administrator must take care during the initial installation of the Linux guest to ensure that the VDisk is formatted as a swap device. But more than that, the VDisk must also be formatted each time the Linux guest is booted.

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

3.7.4 Virtualized Network

The physical network in System z consists of devices known as *Open Systems Adapters* (OSAs). Several varieties are available, such as the OSA-Express4S and OSA-Express5S. These are capable of handling up to 640 TCP/IP stacks simultaneously, including HiperSockets for inter-LPAR communication. An IBM System zEC12 provides up to 96 OSA-Express5S ports for external network communications. The Open Systems Adapter supports both copper and fiber Ethernet connections at speeds of up to 10 Gb.

As might be expected, the z/VM feature to access the Internet Protocol network is TCP/IP for z/VM. OSA-Express devices can be virtualized through a virtual switch (VSWITCH) device to many Linux guests. It is available using special z/VM machines known as *VSWITCH controllers*. Each Linux guest connects using a virtual device controlled by the qeth module to a virtual switch system in a z/VM LPAR.

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

HiperSockets provide high-speed interconnectivity among guests running on a System z. This technology does not require any special physical device configurations or cabling. The guests simply communicate with one another internally via the in-memory capabilities of the PR/SM hypervisor. HiperSockets, however, are not intended to be used for sophisticated networking and should not be used for external traffic.

Both OSA-Express and HiperSockets use the Queue Direct I/O (QDIO) mechanism to transfer data. This mechanism improves the response time using system memory queues to manage the data queue and transfer between z/VM and the network device. Various examples are available in 6.1, “Network analysis” on page 58.

For more information about network in Linux and z/VM, refer to *Advanced Networking Concepts Applied Using Linux on IBM System z*, SG24-7995.

4

Migration process

In the field of information technology, the term *migration* refers to the process of moving from one operating environment to another. In many cases, the move to a new platform involves various organizational and strategic changes.

This chapter provides you with information regarding the approaches involved in planning your migration and defines various types of stakeholders along with their roles and responsibilities. Not every organization uses the same titles for stakeholders as those listed here, but the titles that you use should match the functions described.

Additionally, this chapter describes the process that should be used when undergoing a migration project from identifying the stakeholders, assembling them, and identifying success criteria through to verifying both the migration itself and its success.

4.1 Stakeholder definitions

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

4.1.1 Business stakeholders

Business stakeholders are those who are responsible for making the decisions about the business and provide direction for migration:

- Business owners or business managers

  These stakeholders lead business lines such as Chief Financial Officer (CFO), marketing, and sales. They are concerned with the business and financial resources used in the project. They often view information technology as a tool to accomplish business tasks efficiently and effectively. These stakeholders may have a staff member reporting on technical issues, including migration proposals, that must be evaluated by the technology stakeholders. Conversely, proposals for migration may originate with the technology stakeholders, who must provide sufficient justification to the business owner. Migration justifications are discussed in Chapter 2, “Analyze and understand” on page 17.

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

- Business managers and supervisors

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

- Quality Auditors

  Large and complex consolidation projects require participation from quality auditors to create the Quality Indicators (QI) and ensure that the QI get achieved post migration project.

- Users

  These stakeholders are the end customers. They use the application or consume the services provided by the application and perform testing to assure that the application is working at least as the same level after the successful implementation of the migrated system. In a migration without enhancements, users should not see any changes. Availability and response times should meet the service level objectives agreed to by management and communicated to the users. Their perspective and input to the conversion project is valuable. Their satisfaction must be criteria for the success of the migration project.

4.1.2 Operational stakeholders

Operational stakeholders are different from Business stakeholders in that these are the people who are responsible for implementing the systems and changes:

- Chief Information Officer (CIO)

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

- Project manager (PM)

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

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

- IT managers and supervisors

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

- Mainframe system administrator, system programmer

  The mainframe system administrator is responsible for setting up hardware definitions. The hardware components defined are CHPIDs (channels), control units, and devices. A *channel* is a generic term for external I/O communication paths to Open Systems Adapter (OSA) for Ethernet networks, IBM FICON® or Fibre Channel Protocol (FCP) for attached disk, printers, tapes, and consoles. System programmers install and maintain z/VM including defining LPARs, user directories, and resources for CMS users and Linux guests. They also configure the network connections, virtual switches, and installation of additional products and services such as the IBM Performance Toolkit for VM.

  To maintain and execute these tasks, they have an option to use IBM Wave for z/VM that simplifies the management of z/VM and Linux guests providing an intuitive graphical interface.

- UNIX, Linux, and Windows system administrators

  Linux administrators may assist in installing Linux on System z, or take over administration tasks after the Linux guest has been installed. These stakeholders work closely with the system programmers when major configuration changes or additions are made (such as increasing the memory, disk space, or CPU). All other Linux administration duties will be the same as on other platforms, such as Linux on x86.

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

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

- Network engineers

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

  Network engineers must be familiar with the communications components that are unique to Linux on System z like Vswitch, for example. For more information about IBM System z networking, refer to 6.1, “Network analysis” on page 58. The network concepts and tools outside of the System z box is the same for these stakeholders.

- Database administrators (DBA)

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

- Application architects and developers

  Applications developed in-house require porting and testing on the target Linux system. The effort involved can vary greatly, depending on what language the application is written in and how hardware-dependent the code is. Open source and commercial tools are available to help with tasks such as assessing the portability of your applications. IBM Global Services, as part of its migration services offerings, uses tools developed in cooperation with IBM Research to help with code assessment and conversion. The application architect and developers are the stakeholders who are responsible for this porting effort. Refer to 6.3, “Application analysis” on page 79 for more information about the issues that need to be considered.

- Operators

  The operators monitor the application, the operating, and physical environment by checking the monitor consoles, logs, alerts. They raise problem tickets, notify support teams, and escalate issues to management. New tools and procedures that result from the migration project are required to them.

- Service Desk staff

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

- Users

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

Note: Users are identified both as Business stakeholders and as Operational stakeholders.

- Vendors

  The third-party vendors have many resources that you can use, and they are often ready to help if you make your needs known. They can respond quickly and are often the most cost-effective source of information and solutions.

  For independent software vendor (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 on System z, and there are often open source alternatives.

- Contractors

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

4.1.3 Security stakeholders

The functional area of security has become more visible and critical as company assets become more exposed to the Internet and available on mobile and wireless devices. The security stakeholders include *security administrators*.

The security administrators are the team responsible for data protection, including the authentication and authorization of users who access company applications. The target application must adhere to existent security policies or demonstrate heightened security methods and standards. For more details about Linux on System z security, refer to 6.6, “Security analysis” on page 98.

4.2 Identify the stakeholders

The first phase of the migration involves identifying the stakeholders, as defined in section 4.1, “Stakeholder definitions” on page 38. In turn, the stakeholders identify the business and operational requirements that impact the migration process. All stakeholders within the company must be consulted to ensure that their requirements are factored into the migration planning.

As defined in section 4.1, “Stakeholder definitions” on page 38:

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

4.3 Assembling the stakeholders

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

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

Communicating the change

Stakeholder meetings can be an efficient way to open communication channels. Effective communications plans help to “flatten out” the negative aspects of the acceptance curve.

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

These issues are even more important regarding the IT support team. A strategic decision to switch an operating system or platform can inadvertently create an impression of disapproval of the work the team has done so far, and might cause staff to think their current skills are being devalued.

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

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

Often, the migration to Linux will be accompanied by other objectives. For instance, some customers upgrade their database at the same time to get the latest features and performance enhancements and to obtain support that lines up well with the latest distributions of Linux. As with any project, the scope must be well defined to prevent project overrun, but it is also important that you have a means to manage additions to the plan as business needs dictate.

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

4.4 Migration methodology

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

In a typical migration scenario, an entire environment must be identified, rationalized, and tested for compatibility with the new host operating environment. Figure 4-1 illustrates an approach to planning.

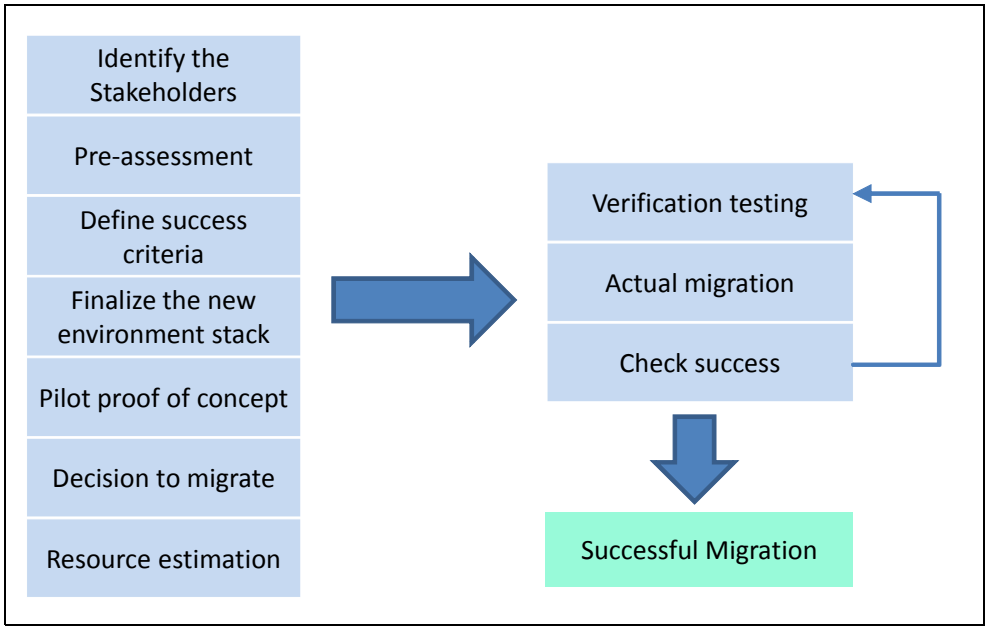


Figure 4-1 Typical migration approach

In section 4.2, “Identify the stakeholders” on page 41, we discussed identifying the stakeholders and in section 4.3, “Assembling the stakeholders” on page 42. In this section, we discuss each of the remaining elements in this approach.

4.4.1 Pre-assessment

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

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

4.4.2 Define success criteria

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

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.

4.4.3 Finalize the new environment

Usually a migration involves moving custom-built or third-party applications to another operating environment. This task involves careful analysis of different tiers of the hierarchy based on a best fit between the database, the application requirements, and other environmental attributes.

We recommend that you perform a one-to-one mapping of the various middleware, compilers, third-party tools, and their respective build parameters. If any of the one-to-one mapping for any parameters is missing, you need to list other parameters available in the tool that would provide the same functionality or feature. The 5.3, “Planning checklists” on page 51 provides examples of forms that can be used to help document your software and hardware requirements.

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

Custom-built applications

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

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

ISV applications

If you are running ISV applications on x86 that you are targeting for migration, you need to determine if the vendor provides compatible versions that support the distribution and version of the target Linux on System z. 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.

Note: There are many open source alternatives for many applications and services for Linux on System z.

4.4.4 Pilot proof of concept

After you have a clear understanding of the target environment and the areas with possible issues and risks, you can proceed to a pilot proof of concept (PoC). This phase is a subset of

the actual migration, but with a reduced scope and duration. In this phase, you implement a small module or stand-alone code snippet from the application onto the target environment.

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

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

Note: PoC projects may require additional funding and may lengthen the project schedule, but will likely contribute to the project's success.

4.4.5 Decision to migrate

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

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

4.4.6 Resource estimation

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

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

Consider the following areas in performing your resource estimation:

- Resources

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

- Education

  Identify skills-related requirements and determine whether the staff has adequate Linux and System z education. Decide whether there are special skills needed for the System z hardware or Linux and hardware combination.

- Service level agreements

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

- Related project aspects

  Be sure to check out what other projects are occurring in addition to the basic system changeover.

4.4.7 Actual migration

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

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

4.4.8 Verification testing

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

Keep the following questions in mind for validation:

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

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

4.4.9 Check against success criteria

After you successfully migrate the environment, reassess the original acceptance criteria with all of the stakeholders. If the criteria is achieved, move the environment to production and obtain a sign-off for the migration activities. Figure 4-2 on page 47 illustrates three important criteria of success from a user perspective.

If the success criteria are not achieved, the migration implementation must be reviewed and once complete, the testing phase must be redone to ensure that the application being migrated meets the acceptance criteria and is ready to go into production.

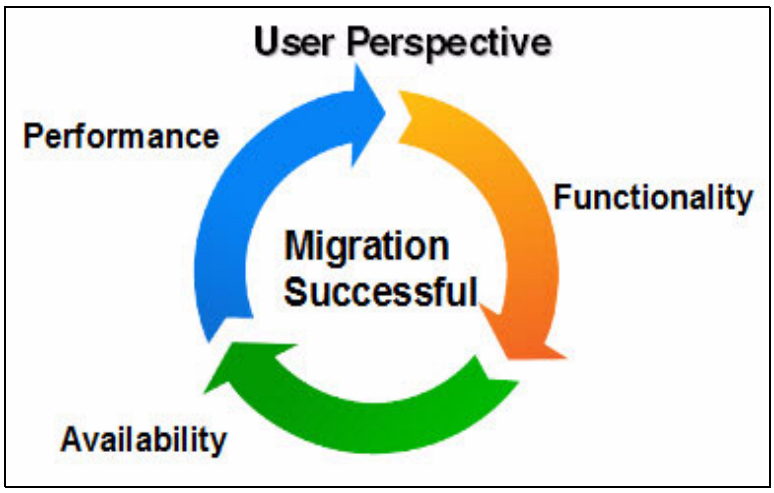


Figure 4-2 Criteria of success from the user perspective

5

Migration planning

This chapter provides an overview of the planning that should be done to successfully migrate workloads to Linux on System z. We provide you with basic, project management information that will be useful in the planning stages of a migration project.

Additionally, we provide you with basic and generic information and templates to aid in assessing the source and target operating environments during the initial planning stages of a migration project.

5.1 Migration project time commitments

There are many phases of the migration process, and anticipating how much time might need to be scheduled for each phase may be difficult. Some practical data assembled by one of the authors of this book suggests that time spent in each of the migration activities may be distributed according to the pie chart depicted in Figure 5-1. It shows that, in general, implementation (50%) and proof of concept and testing (20%) takes most of the time. Other projects may involve more planning than implementation. Your mileage might vary, but this information may help as you prepare the migration plan, as described throughout this chapter.

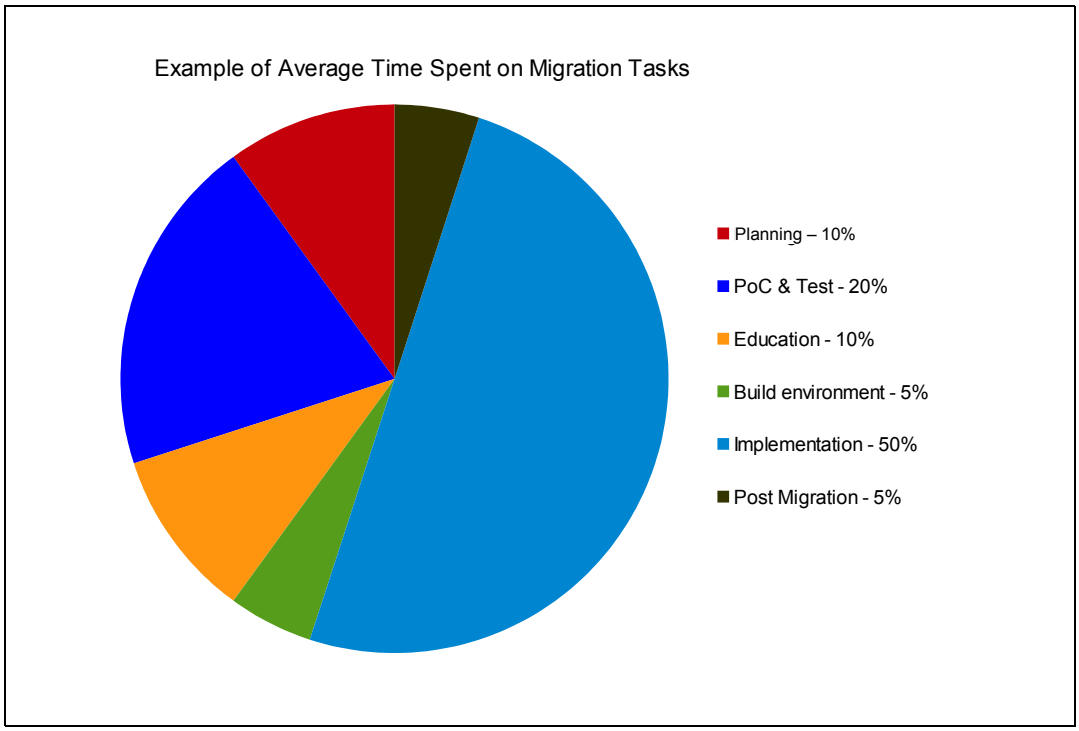


Figure 5-1 Typical time commitment averages for each aspect of migration

The phases of migration, at the highest level, can be described as:

- Planning: Conception of migration project plan. Once you have decided what will be migrated and how, the plan must specify the time, risks, and owner for each migration task.
- PoC and Test: Proof of Concept to check the compatibilities between the x86 and System z environment and give special focus to performance
- Education: It is important that the technical staff has the right skills to work on a System z migration and maintain the new environment
- Build Environment: In this phase, the new infrastructure will be readied for migration.
- Implementation: The actual migration. At this point, communication between stakeholders is important. All involved people must know and approve the migration and then have follow up reporting on the progress.
- Post Migration: After implementation, documentation must be created that further references the project as well as documenting all necessary maintenance and care procedures. Additionally, the project manager must have a signed acceptance agreement.

5.2 Project definition

Irrespective of the size or scope of the migration, the stakeholders must start with a detailed migration plan. The success of a server consolidation project depends on several factors: Clear project definition, preparation to organizational change, sponsorship, and a good strategy. The plan will give to stakeholders an understanding of risks, savings, deliverables, and the big picture of the migration.

The stakeholders will discuss the project plan and produce the principal goals of the migration plan. Documents must be created that represent the strategy that will be used to accomplish the migration goals.

5.3 Planning checklists

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

5.3.1 Product and tools checklist

The software product and tools checklist template shown in Table 5-1 lists all the products and tools that are used in the source operating environment.

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

Table 5-1 Software products and tools checklist

SOFTWARE PRODUCTS AND TOOLS CHECKLIST				
Name	**Version**	**Vendor/Source web site**	**License type**	**Linux on System z**

5.3.2 Application implementation checklist

The application implementation checklist delivers one level further into the product checklist, where each product or tool is drilled down to their features level. There are scenarios where the same product would not offer the same features on all platforms. These details would be noted in this checklist, as shown in Table 5-2

Table 5-2 The application implementation checklist

Application name: **Database connectivity:** **Technical application owner:**		
	Source (x86)	**Target (System z)**
OS Name and Version		
Architecture Model		
Compiler Name and Version		
Additional Software Packages		
Observation		
	Compiler Options for Performance	
Compilation		
Linking		
System Library with Version		
Shared Library		
For Debug		
	Compiler Options for Build	
Compilation		
Linking		
Shared Object Creation		

Each product or tool listed in the product checklist must be analyzed. All the parameters, dependencies, and optimization options must be taken into account in the source operating environment, and then the planning team must assess whether the same kind of features or build options are available in the target operating environment.

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

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

- Often, the optimization features or performance options for a product are only available for that specific platform. In such cases, the optimization features and performance options must be changed to offer the same characteristics to the target environment.

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

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

5.3.3 Application environment checklist

The source application to be migrated could be in the center of a very complex process. The application could be interconnected with many other applications, inputs, outputs, and interfaces. For this reason, you need to prepare a planning document that lists the resources that the source application needs to provide and all the services that it is currently providing. Table 5-3 lists examples of the resources that are required of some applications.

Make the descriptions as detailed as possible by providing the physical location, server host name, IP address, network information, software product used, focal point, and anything else you believe important to register about the services. The target environment must have the same infrastructure available to it as is available in the source environment.

Table 5-3 The application environment checklist

Source resource	Source location	Target resource	Target location
Internal FTP	FTP server on source application server		
External FTP	Batch process through central and secure FTP server		
Local print	Local print on local LAN		
Remote print	Vendor product secured by host name		
DNS services	Single or multiple DNS servers		
Firewalls	Firewall location and rules exported files		
Internet connectivity	Router location		
Intranet connectivity	Web server location and ports		
Email services	Mail transfer agent co-located on source application server		
Messaging services	IBM WebSphere® MQ on source server		
Client software	User's desktop User's notebooks Mobile appliance		
File services	Type, location, and security		
Log server	Central server location		
SNMP	Agent and server location		

5.3.4 Training checklist

A critical element in achieving successful migrations is ensuring that the migration team has skills in the new technology to be migrated. Ensure that a training checklist is put into 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.

5.3.5 Hardware planning checklist

The hardware planning checklist lists the hardware resources that you need to consider during a migration project. In the checklist used in this project, the source environment's hardware resources are examined and we needed to acquire similar or more advanced technology that is available for Linux on System z. Table 5-4 illustrates a sample of the hardware planning checklist that we completed for this IBM Redbooks project.

Table 5-4 Hardware planning checklist completed as an example

HARDWARE PLANING CHECKLIST			
SERVERNAME:			
RESOURCE	**SOURCE**	**DESTINATION**	**OBSERVATION**
Number of CPU	4	2	Real to Virtual
System memory(in GB)	8	8	
OS SWAP Memory(in GB)	4	4	
Network connection[a]			
Connection Description	Gigabit Ethernet	Gigabit Ethernet	
Connection Type	Gigabit Ethernet	Vswitch/GbE	
IP Address/Netmask	9.12.7.88/28	9.12.7.88/28	
VLan number : Vswitch	2	2 : Vswitch1	
Disk Resource[b]			
OS Filesystem	/ : 30 : Ext3	/ : 2 :Ext4	Root
Mount Point : Size(in GB) : Type		/opt : 3 :Ext4 LV OS	Logical Volume
Mount Point : Size(in GB) : Type		/var : 5 :Ext4 LV OS	
Mount Point : Size(in GB) : Type		/var : 5 :Ext4 LV OS	
Mount Point : Size(in GB) : Type		/tmp : 1 :BRTFS LV OS	
DATA Filesystem			
Mount Point : Size(in GB) : Type	/DB : 100 : Ext3	/DB:100:Ext4 LV DB	Logical Volume
Mount Point : Size(in GB) : Type	/WAS : 50 : Ext3	/WAS:50:Ext4 LV WAS	
Logical Volumes : Volume Group OS : 20GB Volume Group DB : 150GB Volume Group WAS: 80GB Volume Group MGM: 20GB			

HARDWARE PLANING CHECKLIST			
SERVERNAME:			
CUSTOM Filesystem			
Mount Point : Size(in GB) : Type		/MGM:10:Ext4 LV MGM	Logical Volume
Logical Volumes : Volume Group OS : 20GB Volume Group DB : 150GB Volume Group WAS: 80GB Volume Group MGM: 20GB			

a. For IBM System z, there are a number of available network connections:
- QETH
- HiperSockets
- Direct OSA-Express connection

b. It is recommended to use the Logical Volume Manager (LVM) for the Linux environment since it provides flexibility and reduces the downtime of the environment with online resize of the logical volumes.

6

Migration analysis

This chapter helps you to understand new features that you will find on Linux on IBM System z and provide a technical direction for your migration. Each section addresses a different part of your infrastructure using scenarios to exemplify how the migration will affect the environment.

The following main sections are available in this chapter:

6.1 Network analysis

This section provides information about network migration configuration issues, explains how the virtual network can be configured, and the facilities available on System z and IBM z/VM.

6.1.1 Network facilities available on System z and z/VM

On the Mainframe, quite a number of different network devices are available for use. Many of these come from a historical background, and should not be used for new implementations. They commonly stay, however, to continue the support of previous installations on newer hardware. Linux on System z can operate using all common network interfaces but for new installations, there are recommended methods for operation depending on the use case.

The following are some technologies that you will find in the System z world that are not used or even seen on x86 systems. This section clarifies some new facilities that you are going to find when you are migrating from x86 to System z. We provide some brief information that you can use to start your network planning. In each subsection, you can find a reference for more detailed information.

Open Systems Adapter

The Open Systems Adapter (OSA) is a hardware network controller. It is installed in a Mainframe I/O cage and provides connectivity to clients on local area networks (LANs) or wide area networks (WANs). It can be directly attached on Linux but will typically be attached to virtual switches (read more in the “Virtual switch” section below). You can find more technical information about OSA cards on *IBM zEnterprise EC12 Technical Guide*, SG24-8049.

OSA with Link Aggregation

You can aggregate multiple physical OSA cards into a single logical link, which is called a link aggregation group (LAG). This configuration increases the bandwidth and provides nondisruptive failover. How to configure it is well described on *Advanced Networking Concepts Applied Using Linux on IBM System z*, SG24-7995.

HiperSockets

HiperSockets is a microcode implementation that emulates a Logical Link Control Layer of an OSA interface. HiperSockets provides near zero latency at memory speed communications between servers running in different LPARs. When connecting a Linux guest to an IBM z/OS system on the Mainframe, the HiperSockets network in Layer 3 mode is the method to use. HiperSockets must be configured in the I/O configuration of the Mainframe. HiperSockets do not provide external connections. If an external connection is required, either a HiperSockets bridge must be implemented by using a VSWITCH, or a Linux guest must be set up as a router.

HiperSockets provide a very fast connection between LPARs. They provide an easy way to connect many Linux servers to a z/OS system in the same Mainframe. This direct connection without involving real hardware is an important factor to simplify setups with many Linux systems that must be connected to z/OS. Some benefits are explained in *Set up Linux on IBM System z for Production*, SG24-8137.

Virtual switch

A virtual switch (VSWITCH) is a software program that enables one virtual host to communicate with another virtual host within a computer system. Virtual switches typically emulate functions of a physical Ethernet switch. In Linux on System z, a VSWITCH provides direct attachment of z/VM guests to the local physical network segment. The VSWITCH allows IP network architects and network administrators to treat z/VM guests as a server in the network.

The switched network inside a z/VM Operating System commonly is implemented with a VSWITCH. When running the VSWITCH as Layer 2, it behaves similar to a real switch just between virtual machines.

The actual speed of a connection with a VSWITCH depends on a number of different variables. The type of traffic is as important as the real underlying hardware and the maximum transmission unit (MTU), which is the maximum size (in bytes) of one packet of data that can be transferred in a network. Common to all of those solutions is that the VSWITCH is faster than a real switch connected to the Mainframe would be.

VSWITCHes do not need a connection to an OSA card to operate. They can also provide purely virtual networks. This also simplifies the setup of private interconnects between guest systems. When creating private interconnects in an SSI with LGR enabled, the use of dedicated VLANs with external interface is recommended. This is necessary to accomplish the private connection between guests that run on different nodes in the SSI.

Implementing VLANs also helps if different guests run in different security zones of a network. It is easy to configure network interfaces to Linux guests that provide only selected VLANs to the guest. These can be configured either as tagged VLANs or as single untagged VLAN on an interface.

The VSWITCH infrastructure provides two basic configuration options. One configures user-based access, the other configures port-based access. From the possibilities, both are equivalent, just the configurations differs.

You can read more about VSWITCH benefits on *Set up Linux on IBM System z for Production*, SG24-8137, and technical information about *Advanced Networking Concepts Applied Using Linux on IBM System z*, SG24-7995.

6.1.2 Network migration overview

There are several different levels of network migration that should be considered because z/VM provides a complete virtual network system, which includes the ability to create multiple virtual switches in the same LPAR. VSWITCHes allow, among other features, the use of 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 card is physically connected.

In this section, we show some common scenarios and how they look on System z.

Single network scenario

One of the most common scenarios is the migration of several distributed machines from the same physical subnet to a single System z LPAR attached to the same network segment. Figure 6-1 on page 60 shows an example depicting a single distributed network.

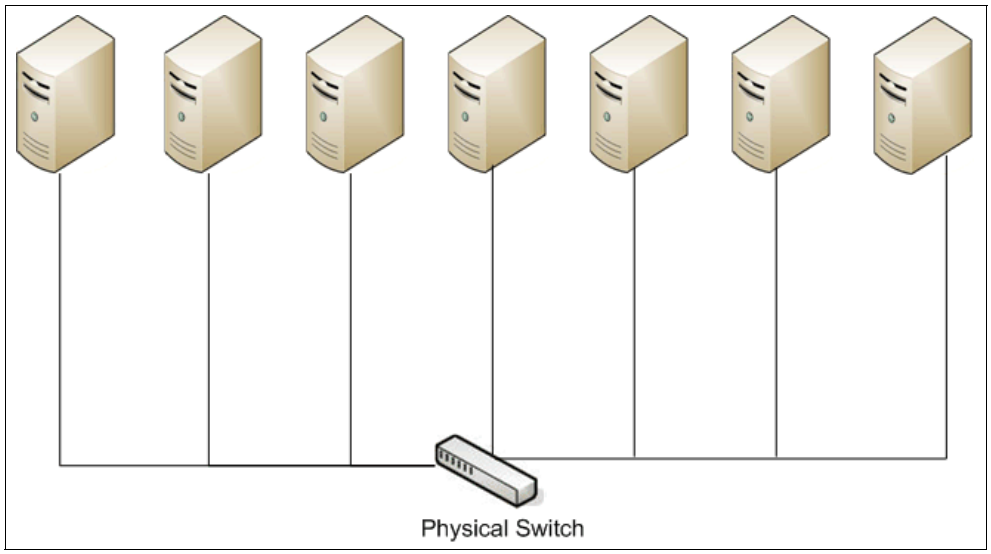


Figure 6-1 Single distributed network

Within this scenario, all physical machines can be migrated to a single System z machine running Linux and sharing the same VSWITCH, which is attached to an OSA card. The OSA card is then connected to the physical network. Figure 6-2 on page 61illustrates this type of configuration.

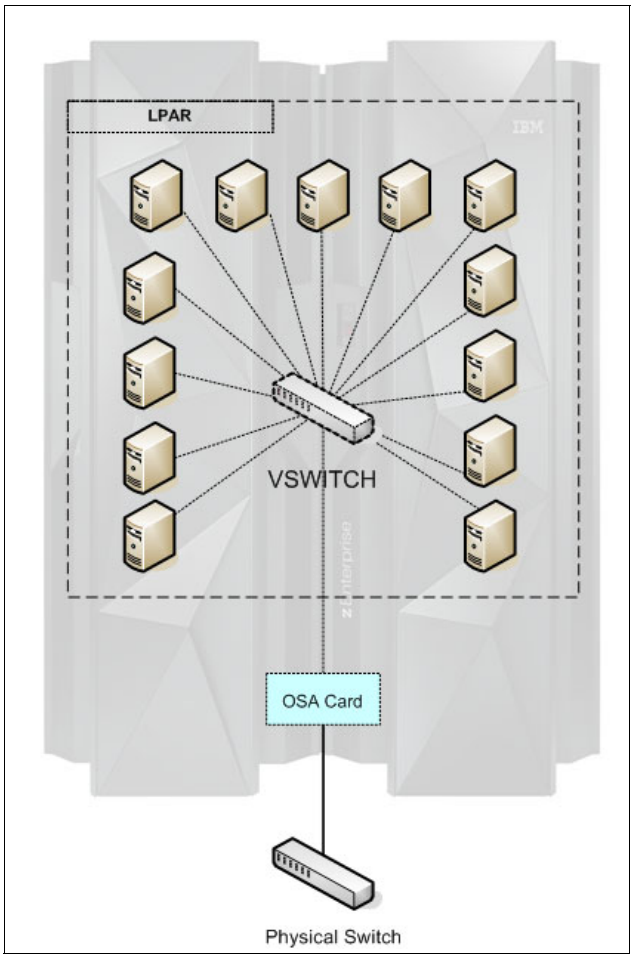


Figure 6-2 Single virtualized network

To increase the availability of each Linux guest, the recommended solution is to configure two or three OSA cards attached to different physical switches in the network. This provides a network failover capability, as illustrated in Figure 6-3 on page 62.

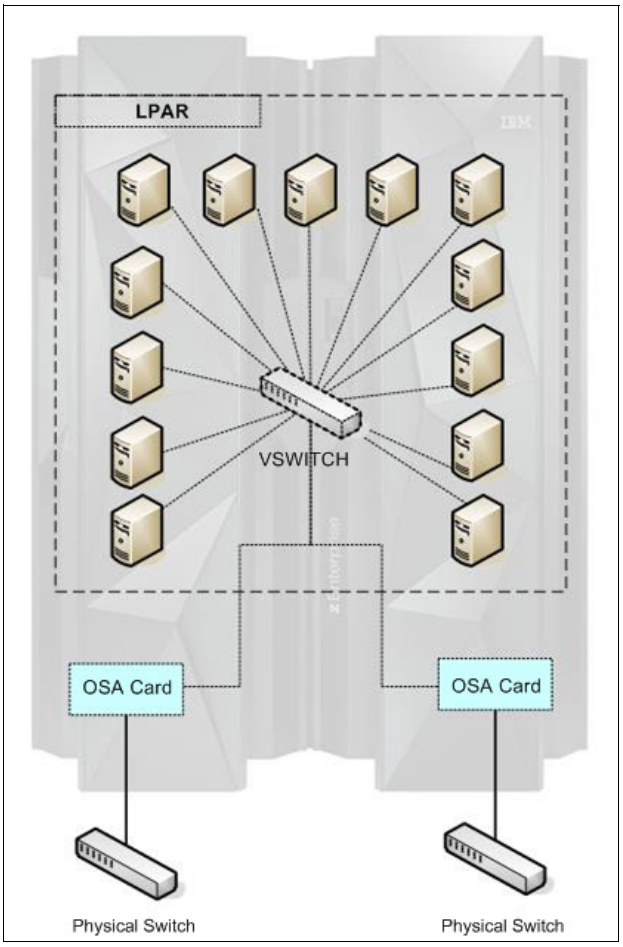


Figure 6-3 Single virtualized network with failover solution

In a Layer 2 VSWITCH configuration, all Linux guests have their own media access control (MAC) address. In a Layer 3 VSWITCH configuration, the Linux guests respond with the OSA 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 card between LPARs. Each LPAR's VSWITCH is connected to the OSA card and the OSA card is directly connected to the physical network segment. This is a common scenario where the development and production server are in separate LPARs. This configuration is illustrated in Figure 6-4 on page 63.

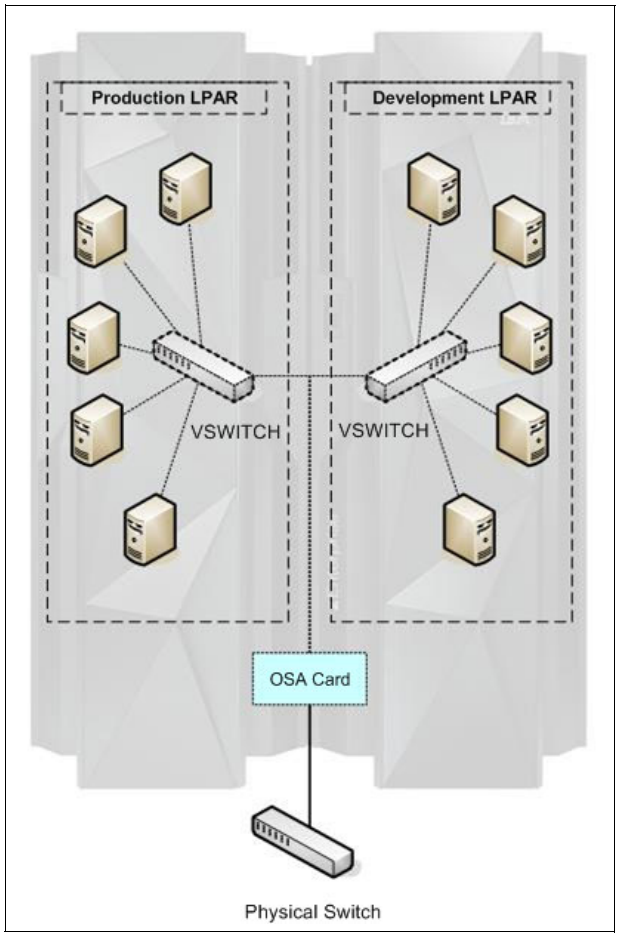


Figure 6-4 Single virtualized network with multiple LPARs

Similarly, the failover solution described previously can also be applied in this case. Sharing the two OSA cards between LPARs is shown in Figure 6-5 on page 64.

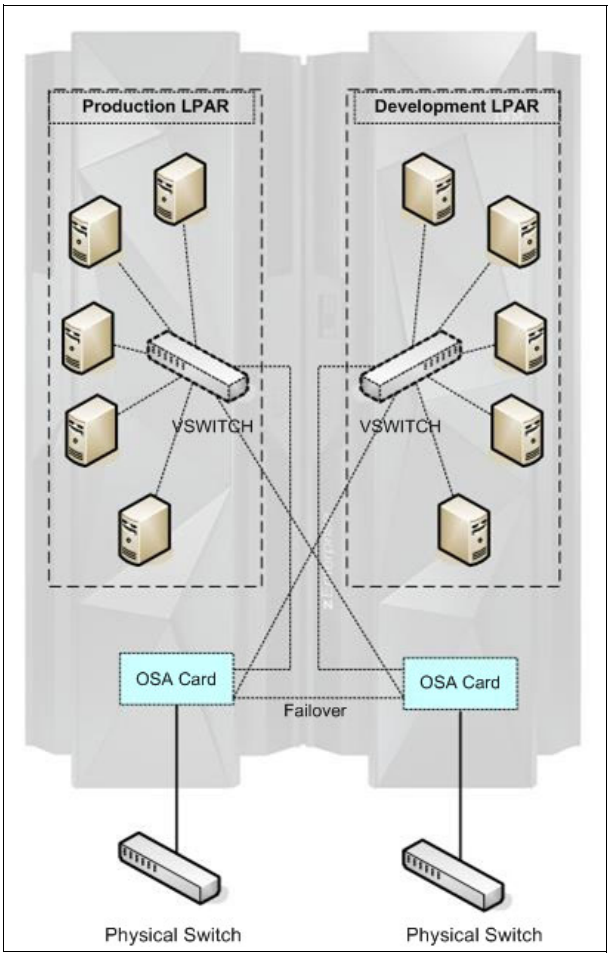


Figure 6-5 Single virtualized network with multiple LPARs and failover

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. We are analyzing the demilitarized zone, or DMZ and a secure network scenario. Figure 6-6 on page 65 shows a DMZ network where the Web Application Server is placed, and a secure network where the database server is located.

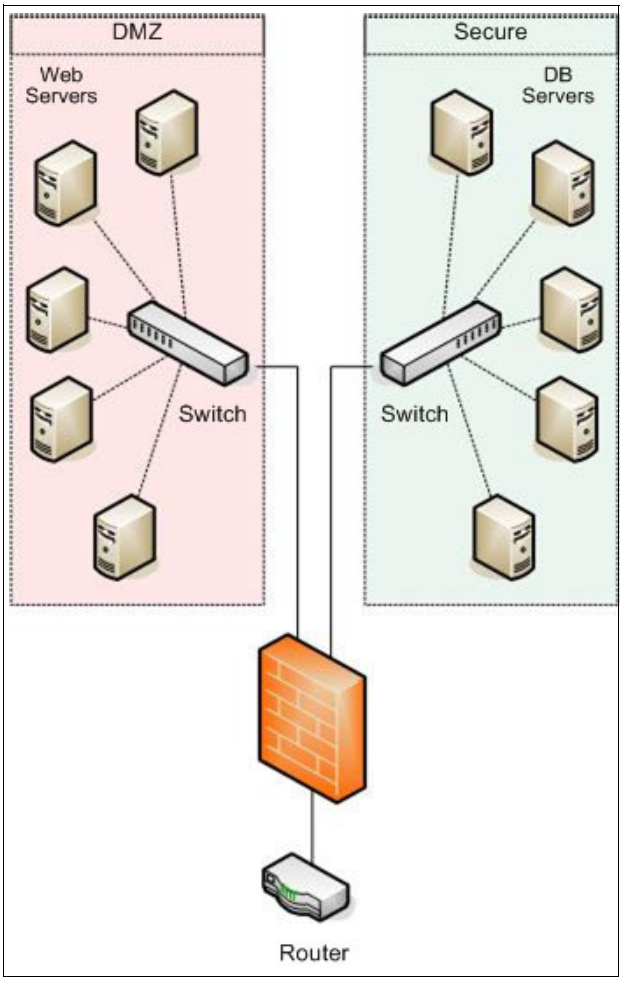


Figure 6-6 Multiple distributed network scenario: DMZ segmented network

You can set up the same scenario on System z. If you have in place a physical switch, a third-party firewall solution, and a router in your environment, you can reuse them as part of your network planning on System z. Otherwise, you can use some network facilities available on z/VM and System z.

The OSA card is connected to one physical switch (or two OSA cards, when the failover solution is configured). The physical firewall can be replaced by a Linux guest that can act as a router and firewall (if you do not have an appliance firewall solution). All virtual Linux guests will be connected to two VSWITCHs setting two different network segments. Figure 6-7 on page 66 shows a network using a Linux guest as a firewall.

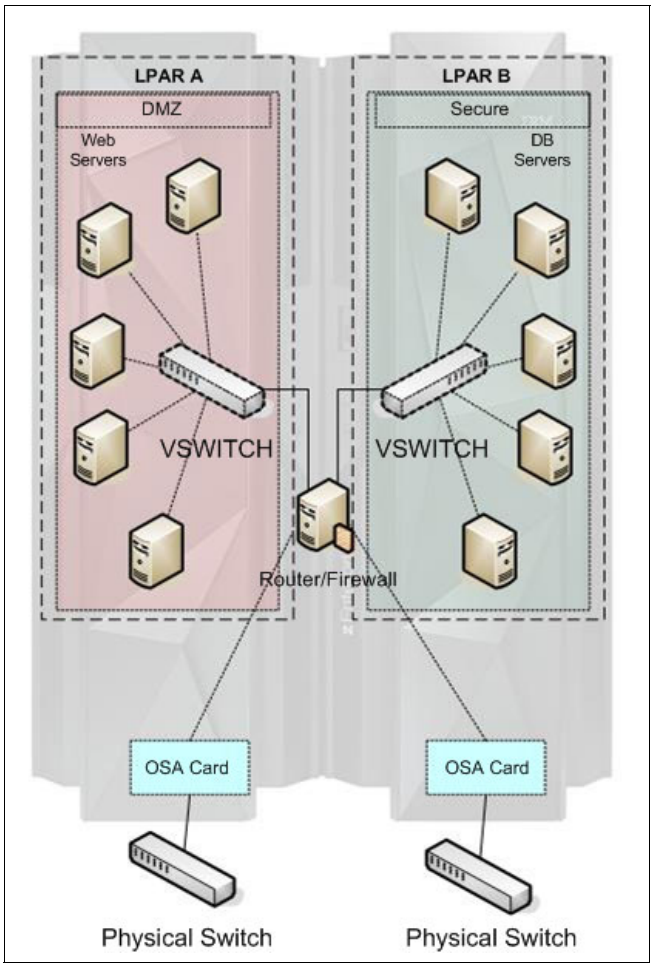


Figure 6-7 Multiple virtualized network scenario: DMZ and secure network

You might have noticed in Figure 6-7 that we are not sharing the OSA cards. It is possible to have the OSA card shared between multiple LPARs on the same System z hardware. To create this solution, it is recommended that you have an external firewall to manage the network filters. Figure 6-8 on page 67 illustrates the solution that is described as a network segmented LPAR.

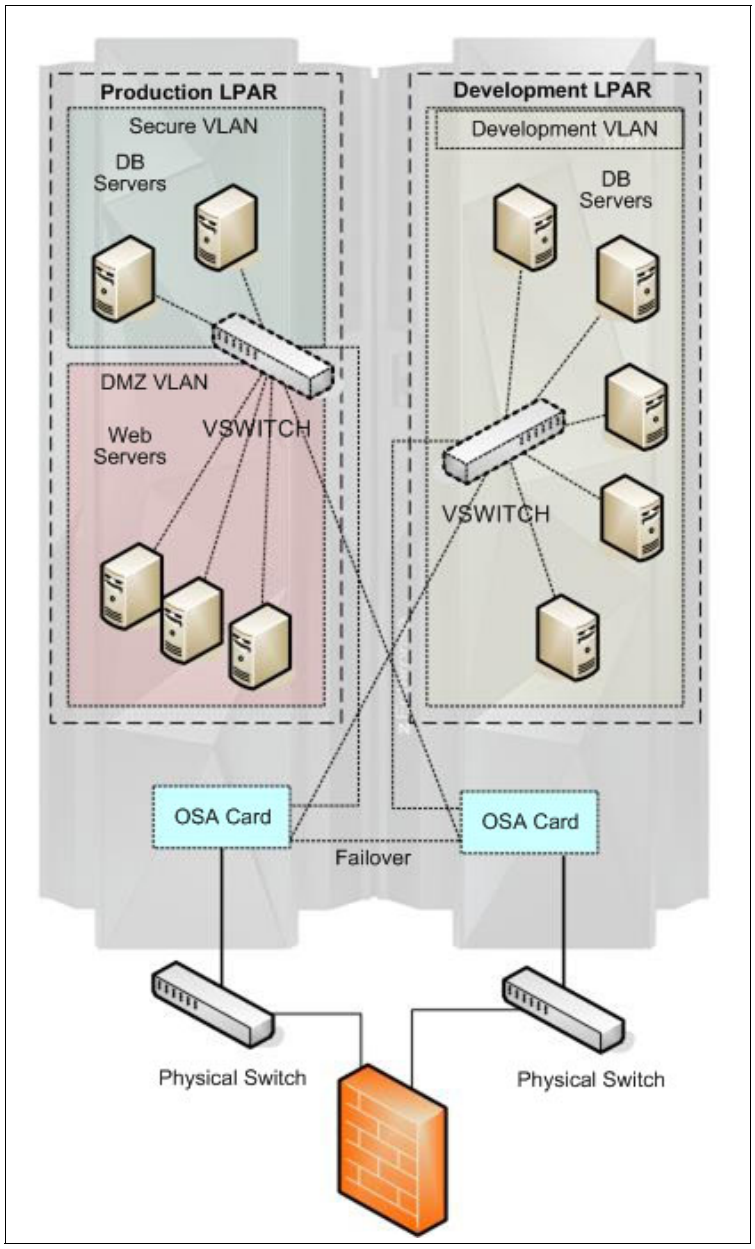


Figure 6-8 Multiple virtualized network scenario with failover: DMZ and secure network

You can isolate the entire secure network from the physical network segment using multiple LPARs. The communication between the LPARs is managed by HiperSockets devices.

Figure 6-9 on page 68 illustrates an example.

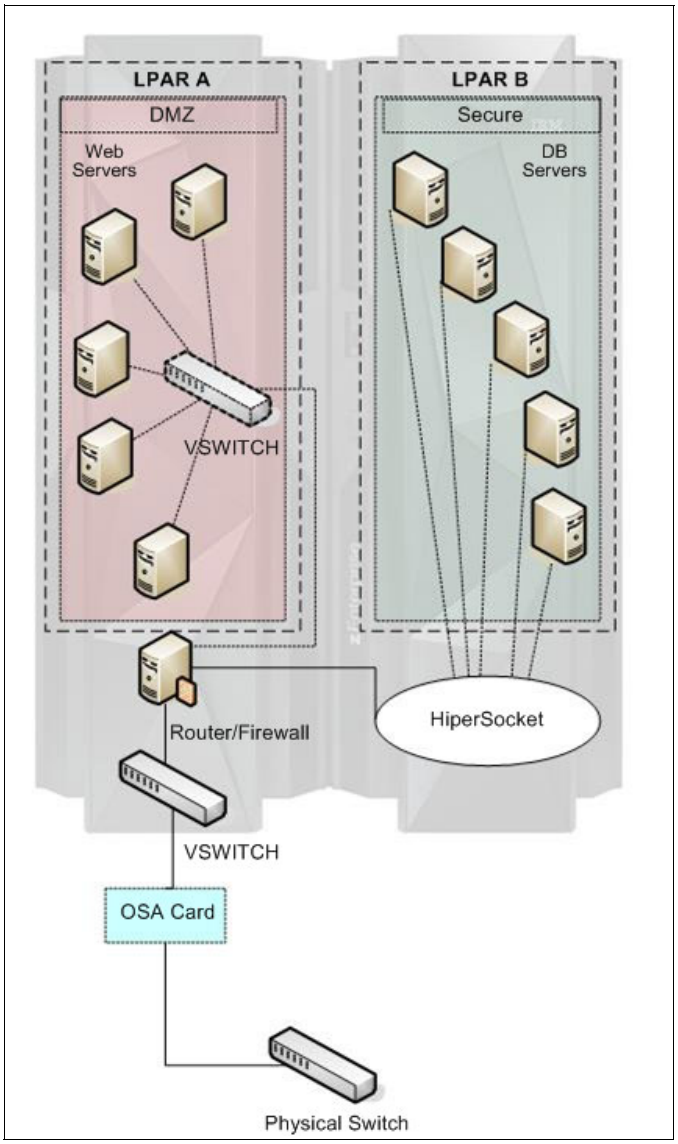


Figure 6-9 Multiple virtualized network scenario with multiples LPARs

Note: Although the use of HiperSockets for this scenario is possible, it might not be the recommended solution. If one of the LPARs is CPU-constrained, that could cause a delay of network traffic. You can read more about HiperSockets in *Set up Linux on IBM System z for Production*, SG24-8137.

VLAN segmented network

The use of the VLAN tag on z/VM SWITCHes is fully supported. The VLAN configuration will help in the segmentation of network packages, bringing security and organization to the environment. It will facilitate the administration of network grouping the guests with common requirements regardless of their physical location. The VSWITCH, like a physical switch, will provide full authorization on a per port basis for membership in a VLAN segment.

For a high security network scenario, use the LPAR environment mixed with the multiple network segmented solution. As illustrated in Figure 6-8 on page 67, the entire System z

environment is virtualized and all configurations are made per virtual machine, which increases the security, reduces the complexity, and simplifies the environment.

6.1.3 Helpful steps for a network migration

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

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

At this point, the target server (Linux on System z server) must be assigned a host name that is different from the source server name:

1. Migrate the applications (for more information, see section 6.3, “Application analysis” on page 79) and files from the source server to the target server.
2. Shut down the source server.
3. Change the Linux on System z server’s host name.
4. Change the DNS registered name to the new Linux on System z IP address.

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

6.2 Storage analysis

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

6.2.1 Data migration

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

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

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

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

Online data migration

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

To perform an online migration, follow these steps:

1. Install and configure the target Linux on System z server (refer to 6.2.2, “Linux on System z: pre-installation considerations” on page 73 for more details).
2. Install the middleware application on the Linux on System z server.
3. Copy the application data to the target Linux on System z server.

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

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

   This step is not a test of the application on Linux on System z. The application must be tested on a development machine to guarantee that the application is a Linux on System z compatible application (refer to 6.3, “Application analysis” on page 79 for more details).
6. Shut down the source servers.

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

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

Offline data migration

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

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

Using the Rsync application

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

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

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

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

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

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

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

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

Transferring files over the network

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

That exported/dumped file needs to be transferred across the network, and the database import procedure must be executed at the target server. Refer to 6.4, “Database analysis” on page 87 for more details.

Migrating storage volumes

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

Storage SAN Volume Controller

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

The SAN Volume Controller sits in the channel path and allows you to virtualize all FCP storage from multiple vendors that sit behind it. Figure 6-10 on page 72 shows where the SAN Volume Controller sits in the storage area network (SAN). The SAN Volume Controller has visibility to all supported storage on the SAN.

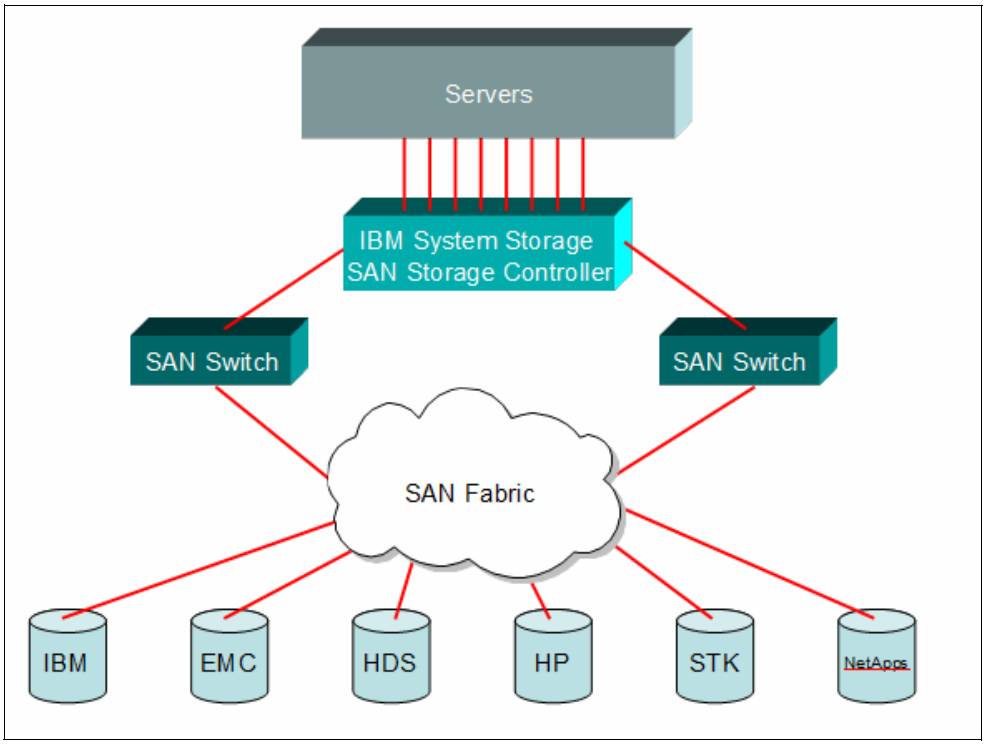


Figure 6-10 SAN Volume Controller

The following benefits are provided by the SVC:

- Single point of control for heterogeneous storage resources
- Dynamic data migration between heterogeneous storage devices on a SAN
- Ability to pool the storage capacity of multiple storage systems on a SAN
- Scalability to support up to 1024 host servers
- Instant copies of data across multiple storage systems with IBM FlashCopy®. More information about FlashCopy is available at the following site:

  `http://www.ibm.com/systems/storage/flash`
- Copy data across metropolitan and global distances as needed to create high-availability storage solutions (More details about *Implementing the IBM System Storage SAN Volume Controller V7.2*, SG24-7933)

When migrating Linux systems from x86 to Linux on System z, the SAN Volume Controller will allow you to non-disruptively migrate data to Linux on System z. For more information about the IBM System Storage SAN Volume Controller, see the following site:

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

- Additional information
  - *Introduction to Storage Area Networks and System Networking*, SG24-5470
  - *IBM System Storage SAN Volume Controller Best Practices and Performance Guidelines*, SG24-7521
  - *Implementing the IBM System Storage SAN Volume Controller V7.2*, SG24-7933
  - *Implementing FlashSystem 840 with SAN Volume Controller*, TIPS1137

Helpful steps for an auxiliary storage migration

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

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

1. Create the new server file system with mount points for all data files.
2. Create a temporary file system to be used in the file transfer process on the target server.
3. 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.
4. Note the following points:
   - If there is enough space at the source server to compact all of the data, consider using data compression features such as **`zip`**, or **`tar`** with **`gzip`** and **`bzip`** formats. Both of these formats are compatible with Linux on System z. The data can be transferred using an FTP server configured on the target server.
   - If there is not enough space at the source server to compact the data, mount the NFS file system or map the Samba file system at the source machine, and copy the files across the network.
5. 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 Fibre 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.

6.2.2 Linux on System z: pre-installation considerations

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

A best practice for Linux on System z is that only one version of a Linux OS distribution should be installed from scratch. Therefore, the basic Linux file system should be designed to allow the highest possible model of servers and then all other Linux guests in the environment should be cloned from this source (known as the *golden image*). On IBM Wave for z/VM, this golden image is called a *prototype*. The file system that stores the application data is created after the cloning process depending on the needs of the application that will reside on the server. If you want to know how to create an SLES 11 or RHEL 6.4 golden image, see *The Virtualization Cookbook for IBM z/VM 6.3, RHEL 6.4, and SLES 11 SP3*, SG24-8147.

Managing Linux with IBM Wave

IBM Wave for z/VM can help you through the process of creating and cloning your Linux servers. It is possible to create a prototype that will allow you to create a new server and expand your Linux farm easily. More details can be found in *IBM Wave for z/VM: Installation, implementation and exploitation*, SG24-8192.

Logical Volume Manager (LVM)

All file systems, except the root (/) file system, should be created as LVM devices. File systems created with an 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 Logical Volume Manager (LVM) is very useful for Linux file systems because it allows you to dynamically manage file system size and has tools to help back up and restore failing partitions.

Basically, LVM volumes are composed of the following components:

- Physical volume

  A physical volume (PV) is a storage device, and it can be a DASD device or a SCSI device controlled by a zFCP channel. For Linux on System z, each DASD device is a physical volume.

- Logical volume

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

- Volume group

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

Figure 6-11 shows five minidisk (MDisk) devices that are used by a Linux guest to create a unique VG. It is then further organized or allocated into two LVs.

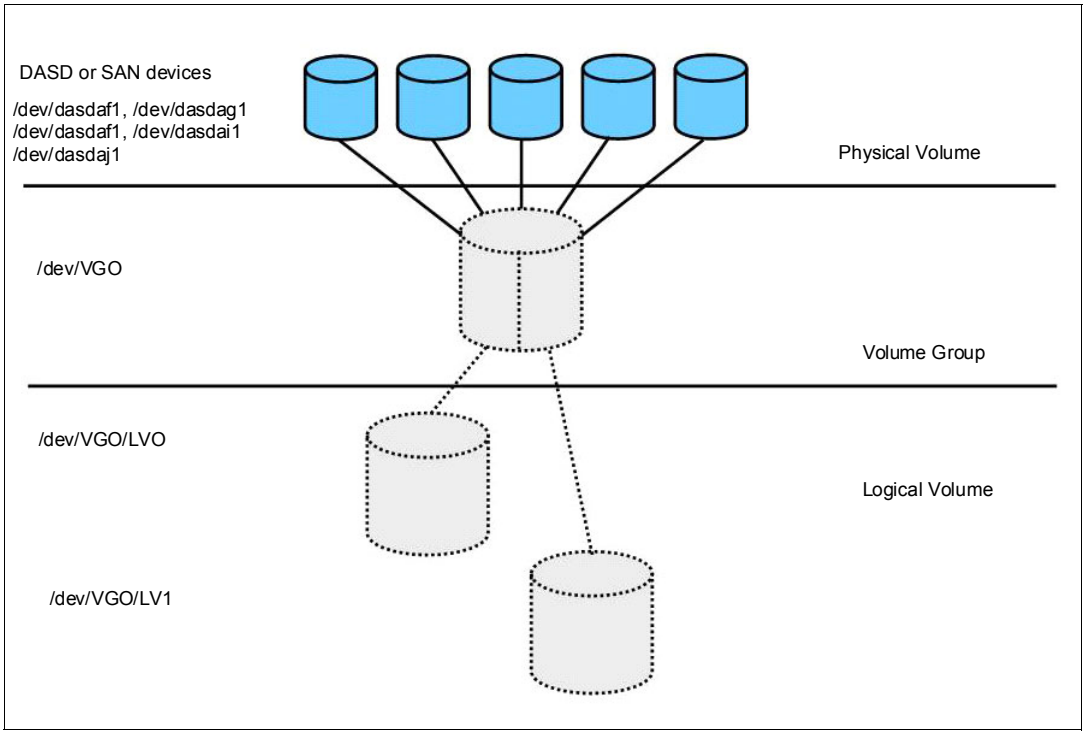


Figure 6-11 LVM example

There is, however, a small performance price that must be paid when using LVM. The flexibility of LVM often outweighs the cost of the performance hit.

For more information about the LVM setup during the installation, see *The Virtualization Cookbook for IBM z/VM 6.3, RHEL 6.4, and SLES 11 SP3*, SG24-8147.

Linux file system

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

The golden image should include the following file systems:

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

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

The root (/) file system

The `root` file system is the first file system to be created, and it is the base for all other file systems in the hierarchical structures of the Linux operating system. A size of 350 MB should be enough for the root file system.

Important: The `root (/)` file system should not be placed on an LVM device because in case of an LVM failure, you can recover the system using the single user mode.

The /boot file system

The `/boot` file system is often left simply as a subdirectory under root (/), but maintaining this directory structure as its own partition can be particularly useful. `/boot` contains the boot files, such as the kernel, the parm file, the initial ramdisk, and the system map. In SLES and RHEL, the /boot partition also contains the boot loader configurations, such as zIPL or GRUB. As it holds the kernel files, it may be considered to be the most important partition of all. Keeping it as its own partition helps preserve its important status and maintain its integrity.

Important: Like `root (/)`, the `/boot` file system should not be placed on an LVM device. The recommended file system type for `/boot` is EXT3.

The /usr file system

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

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

This file system should be created on LVM devices that allow you to dynamically extend or reduce the file system size.

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

The /var file system

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

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

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

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

The /tmp file system

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

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

The /opt file system

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

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

The /home file system

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

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

Other file systems

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

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

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

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

Additional resource

You can see additional recommendations, like volume group and disk naming convention in *Set up Linux on IBM System z for Production*, SG24-8137.

Shared file system

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

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

In the example shown in Figure 6-12 on page 78, only the special file system and mount points relevant to the solution are represented. The data file location is at mount point `/srv/www/app`. This is the file system that is shared between the Linux guests. There is also the shared file system `/opt/ibm/IBMHTTP`, where the web server binaries are installed. For the IBMHTTP service, the log files are redirected to the local `/var/log/httpd` file system. All shared devices are the same DASD device type and managed by the z/VM operating system.

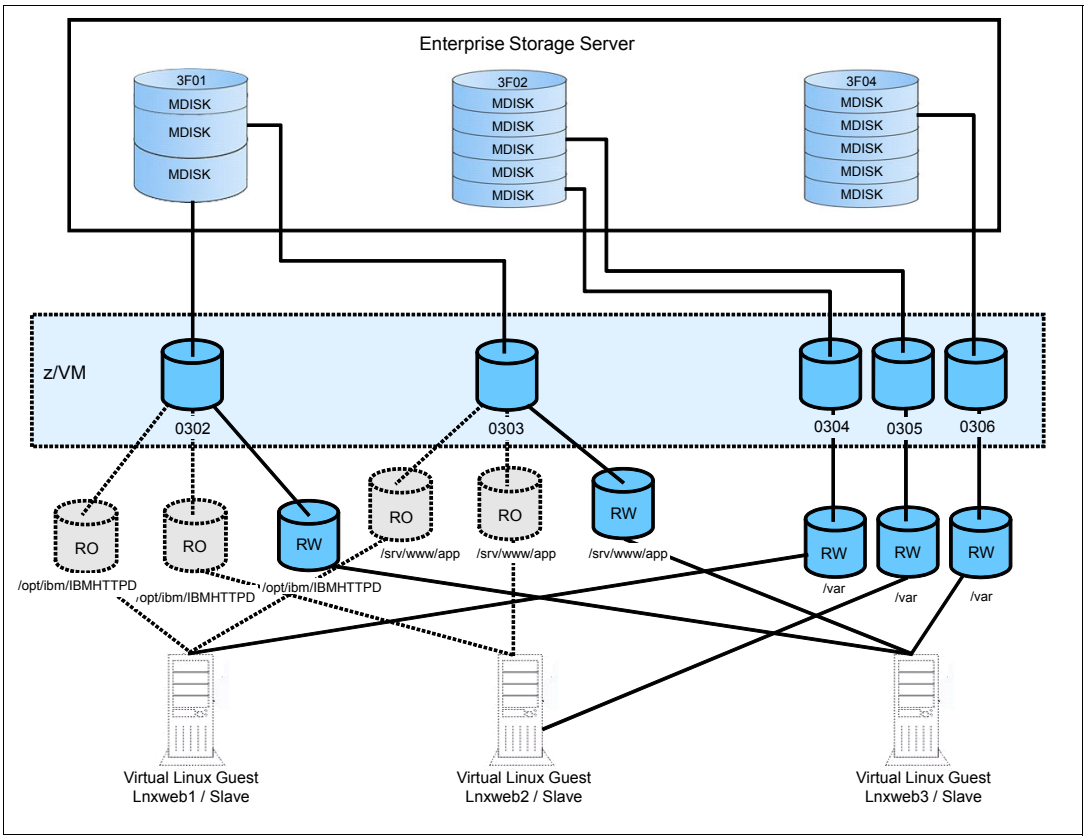


Figure 6-12 Shared devices example

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

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

ECKD and zFCP devices

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

The zFCP device, when configured with multiple access channels, provides a better I/O response than a single ECKD channel device. After it is configured on Linux on System z, it is possible to split it into partitions like a simple SCSI device using the **`FDISK`** tool. Even though the sizes of the ECKD volume devices are determined at the storage hardware level, it is still possible to configure smaller volume sizes for the Linux guest when the z/VM system administrator formats the ECKD devices as MDisk devices.

A combination of both solutions can help you improve system performance and use storage resources efficiently. For more information, see *Linux on IBM System z: Performance Measurement and Tuning*, SG24-6926.

6.3 Application analysis

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

We discuss the following topics:

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

6.3.1 Why migrate applications

As discussed in Chapter 5, “Migration planning” on page 49, application migration should only be undertaken after thorough planning. There also must be a compelling reason to act, such as the following real world situations:

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

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

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

6.3.2 Which applications can be migrated

Every computing platform offers specific areas of strength, and the aim of a migration should be to select applications that take advantage of the strengths of the target platform. The

classic strengths of IBM System z include high availability, high I/O bandwidth capabilities, the flexibility to run disparate workloads concurrently, and excellent disaster recovery capabilities.

Another key element in choosing the appropriate applications for migration is whether they are supported on Linux on System z. This is normally not a problem with homegrown applications, depending on what language they were written in, but it could be a significant issue with ISV-supplied applications.

6.3.3 Selecting an application for migration to Linux on System z

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

The following list includes applications that cannot or should not be migrated to Linux on System z, and explains why they are unsuitable:

- Applications that are available only on Intel or UNIX platforms.

  Requesting an ISV to support their application on Linux on System z is a long process.

- Servers that have already been virtualized.

  In such cases, most of the TCO benefits of virtualization have already been realized and only minor benefits will be forthcoming. However, if the existing virtualized environment is reaching its limits or the server leases are within 9 to 12 months of expiry, there may be a good business case for moving the applications to Linux on System z because of its higher virtualization capabilities.

The following list includes applications that are suitable for migration and explains why they are suitable:

- Applications or middleware (database, application servers, and so on) that are supported by a software vendor on multiple platforms, including Linux on IBM System z.

  There are no support issues and migration is much simpler.

- Applications that need close proximity to data on IBM System z, or that are components of System z applications.

  You can boost the performance and speed of your Linux on System z applications by putting them on the same physical server as their data source.

- Applications with high I/O or transactional I/O.

  Because of its design, IBM System z excels at handling sustained high I/O rates.

- Applications with lower sustained CPU peaks and average memory needs.

  These are ideal workloads for IBM System z. The platform has been designed to run multiple workloads at a consistently high CPU and memory utilization.

- Application development environment for Linux on other platforms.

  The virtualized Linux on System z platform provides an ideal environment to test applications before their deployment to Linux on other platforms.

6.3.4 Applications best suited for migration

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

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

IBM software

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

Linux on System z offers various solutions, which you can see here:

`http://www.ibm.com/systems/z/os/linux/solutions`

Generally, migrating from IBM products on distributed servers to the same IBM products on Linux on System z is a relatively straightforward process. You can see some examples in Chapter 8, “Hands-on migration” on page 155.

DB2

You can use the well known DB2 for Linux, UNIX, and Windows product on Linux on System z. It works seamlessly in the System z virtualization environment, straight out of the box. Also the autonomic features such as self-tuning memory management (STMM), and enhanced automatic storage, will help the database administrator to maintain and tune the DB2 server. Check section 8.2, “Migrating DB2 and its data” on page 157 to see a migration example from x86 to System z.

You can find more information and also use cases in *DB2 10 for Linux on System z Using z/VM v6.2, Single System Image Clusters and Live Guest Relocation*, SG24-8036

Oracle

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

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

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

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

`http://www.oracle.com/technetwork`

Additional information can be found in:

- *Experiences with Oracle 11gR2 on Linux on System z*, SG24-8104
- *Experiences with Oracle Solutions on Linux for IBM System z*, SG24-7634

6.3.5 Other software

This section lists non IBM software that is a good candidate for migrating to Linux on System z.

SAP

- SAP application servers are candidates for consolidating to Linux on System z, particularly if the SAP database is DB2 on z/OS. The benefits provided by HiperSockets and virtual networks help to reduce the substantial amounts of network overhead that can occur on physical networks.

  Additional benefits include potentially reducing the system administration costs and limiting the environmental impacts from a reduction in distributed servers.
- SAP solutions using a System z framework of z/OS and Linux on System z have been successfully implemented at many IBM System z accounts.
- A typical setup for deploying an SAP system uses DB2 for z/OS as the database backend, and Linux on System z for the SAP application server platform. DB2 client uses IBM DB2 Connect™ to establish the connection between the database backend and the SAP application server.

Infrastructure services

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

  Additionally, a significant amount of network traffic is generated between data on z/OS and FTP and NFS servers. When the servers are hosted on the same system as the data and HiperSockets is used, then not only is this network traffic greatly reduced, but the batch processing window for that data can also be reduced.
- LDAP security services fit very well running on Linux on System z, including both OpenLDAP products as well as commercial products like IBM Tivoli® Directory Server, Tivoli Directory Integrator, and Tivoli Access Manager. Using System z architecture, clients can build a robust LDAP infrastructure.

Application development

- Whether for Java, C/C++, or most other programming languages, a virtualized Linux environment is an ideal platform for application development. Although developers usually develop on a stand-alone platform, testing and modifying are generally performed in a server environment. Developers can be given multiple virtual servers to perform interactive testing while troubleshooting or enhancing the application. z/VM also provides a number of features that enhance application troubleshooting.
- Other major benefits include the ability to rapidly deploy virtual servers for user acceptance testing and integration testing and, when that is finished, the virtual servers are shut down. If a developer inadvertently "damages" a virtual server, a new server simply has to be cloned. There is no need to spend a great deal of time formatting disks and reinstalling the operating system and required applications.
- For new applications, virtual servers are deployed quickly and can be easily customized for a specific purpose. Many customers have standard server profiles that are pre-built, so to create a new virtual server, the appropriate profile simply has to be cloned, which can be done in minutes. When an application is discarded for some reason, the virtual servers can be discarded as well.

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

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

http://www.ibm.com/systems/z/solutions

6.3.6 Selecting an application for a proof of concept

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

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

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

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

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

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

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

6.3.7 Applications not supported on Linux on System z

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

If an unsupported application is required to work with another application that is supported, the best option would be to use a hybrid environment where one application is on Linux on System z and the other application remains on its existing (or modernized) platform and communicates with Linux on System z. For example, suppose that you have a reporting tool that only runs on x86 that analyzes an Oracle database that is also on x86. In this case, the Oracle database could be migrated to Linux on System z and the reporting tool could remain running on x86.

6.3.8 Application interdependencies

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

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

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

6.3.9 Successful application migration

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

6.3.10 Special considerations for migrating a Java application

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

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

6.3.11 Special considerations for migrating C++ applications

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

Architecture-dependent code

Programs residing in directories (on non IBM System z® 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 System z.

Assembler code

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

`http://www.gnu.org/software/binutils`

ptrace and return structure

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

Little endian to big endian

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

Stack frame layout and linkage specific to System z

For details about stack frame layout and linkage specific to System z, refer to `/usr/src/linux/Documentation/Debugging390.txt`. The location of this file may vary depending on the distribution. If you are not able to find it, try the kernel documentation:

`https://www.kernel.org/doc/Documentation/s390/Debugging390.txt`

Changes to build scripts

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

/proc filesystem

The proc filesystem has some differences:

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

Available languages and compilers

Additionally you have available some popular programming languages like: Ruby, Perl, Tcl, Python, Scheme, Regina (REXX), the IBM Java JDK.

Shared objects

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

6.3.12 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, Oracle, DB2 UDB, and DB2 Connect. As described on 6.3.4, “Applications best suited for migration” on page 80, there is a bunch of middleware available for System z architecture, like Apache, Tomcat, vsftp and more. You can check it from the package manager or the official website of your Linux distribution.

6.3.13 Helpful steps for an application migration

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

Here are some steps that you might find helpful during your migration:

1. Perform source application mapping.

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

   a. Is the source code available to be compiled and deployed on the target server?
   b. Is there a version of the middleware available for Linux on System z?
   c. Are there performance reports of actual development tests to compare with after the migration?

2. Design the network solution for the application (see 6.1, “Network analysis” on page 58 for more information).
3. Design the file system for the application and middleware (see 6.2, “Storage analysis” on page 69 for more information).
4. Clone the Linux on System z server (or servers) from the golden image.
5. Configure the network at the target server (or servers).
6. Create the custom file system at the target server (or servers).
7. Install and configure the middleware at the target server.
8. Copy the application code from the source to the target server.
9. Compile and deploy the application code to the target server.
10. Provide the first application test reports.
11. Start the performance test on the target server to understand the performance of the migrated application.
12. Size the CPU and memory to fit the migration expectations.
13. Execute the application stress test.

After all tests have been completed and approvals granted:

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

6.4 Database analysis

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

6.4.1 Before database migration

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

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

6.4.2 Migrating a single instance

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

1. Configure the Linux on System z network (follow steps 1 - 4 as listed in 6.1.3, “Helpful steps for a network migration” on page 69).
2. Configure the temporary storage area at the source server and at the destination server.
3. Stop the database services.
4. Issue the export/dump procedures at the source server.
5. Transfer the export/dump files through the network to the destination Linux on System z server.
6. Shut down the source server.
7. Change the Linux on System z server host name and IP address.
8. Perform import procedures at the destination server.
9. Perform the database and applications tests.

6.4.3 Migrating multiple instances

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

Migrating multiple servers to Linux on System z

A significant factor in the migration of multiple servers to Linux on System z is the distribution of server peak load. Document and compare peak workload information, including how long the workloads take and how much server resource is used. You can use Table 6-1 to map server workloads when creating the migration configurations.

Table 6-1 Sample database server workload map

Server information			**Peak load measure**		**Peak load time**		
Server name	Total of CPU	Total of memory	% CPU used	% Mem. used	Week day	Start time	Stop time

As explained in section 3.7.1, “Virtualized CPU” on page 31, the CPU and memory constraints in an LPAR are possible and desirable, but the server should maintain the same peak load for a long period of time if there are not real CPUs to process each virtual CPU request.

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

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

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

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

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

Another point to consider regarding CPU capacity is the relationship between the source server and the migration server; it is not 1:1. In other words, one distributed server with four CPUs must not necessarily have four CPUs in the destination virtual server; best practice shows that the actual number is less than that. For more information about this topic, refer to 6.4.4, “Technical considerations” on page 89.

Migrating a multiple instance server to Linux on System z

Usually on your development environment you have one database server with multiple instances. This should be all right for a development environment, but when you are migrating your production environment you will want to isolate your instances and simplify the database management. For best results from this type of migration, the workload analysis should be very detailed. Different instances have different workload types, times, and characteristics that might allow the overcommitment of CPUs and memory.

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

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

To minimize the work related to database software fixes and security updates, it is possible to use shared devices for database binaries and libraries. For more information about these topics, refer to “Shared file system” on page 77.

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

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

You can use Table 6-1 on page 88 also to map the instances used by simply changing the Server name column to instance name and documenting the appropriate information.

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

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

6.4.4 Technical considerations

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

CPU

The number of virtual CPU resources in a database server is very important; setting the maximum possible does not guarantee better performance. The number of CPUs should be large enough to avoid the processor queue.

The number of processes in a processor queue is influenced by all the other resources of the server, and should not be analyzed as a separate resource. Memory constraints or I/O constraints affect the processor queue number directly, so before deciding that the server does not have enough CPU and adding a new CPU to the service, analyze the CPU schedule time. If the system is running in a high processor queue and most of the CPU time is

dedicated to SYSTEM, it probably is associated with memory. The correct parameter to resize is the memory size. Similarly, if the CPU time is dedicated to I/O WAIT, the file system should be reorganized.

In the beginning, you will not know how many virtual CPUs your database will need on System z. Start with a low number of CPUs and increase as needed.

You can read more about it in *Linux on IBM System z: Performance Measurement and Tuning*, SG24-6926.

Memory

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

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

Swap memory

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

A System z best practice is to use the VDisk devices as swap devices. Because swap configured at VDisk devices provides desirable response time, the eventual memory paging (the process that moves memory blocks to and from real memory and to and from swap memory) is not considered a real problem. It is also not considered a problem if the server has no 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.

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

The Linux server uses the swap memory to allocate memory pages that are not used in real memory as its default configuration. However, that is not the most desirable solution when considering database servers. In fact, it is best to avoid this type of situation. There is a configurable kernel parameter called `swappiness` that determines whether more or fewer pages will be swapped; see Example 6-1.

Example 6-1 /proc/sys/vm/swappiness

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

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

The second configuration regarding the swap pages is the `page-cluster` kernel parameters that control the number of pages that will be written at the swap in a single attempt; see Example 6-2. The default value is eight pages at a time. Changing this value to a smaller value will reduce the paging time.

Example 6-2 /proc/sys/vm/pge-cluster

```
at /etc/sysctl.conf file include the line
```

```
vm.page-cluster = 1
```

The correct swap size depends on your database and how much memory it uses. The swap memory should only be used in a usage peak, so your swap size should be a safe number that will hold this peak and avoid an outage due out of memory issues. Just for reference, you can use the amount of 20% of the total memory, but do not set more than 2 GB of swap memory at a first moment. Like the memory sizing, you should monitor swap when the usage peak occurs and increase or decrease it accordingly to improve performance.

Shared memory

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

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

To obtain the best memory allocation, you must to set some Linux kernel parameters and these parameters depend on what the DBA allocated in the migration. As shown in Table 6-2, some recommendations should be followed to avoid issues like memory starvation.

Table 6-2 Recommended kernel parameters

Parameter	Description	Recommended value
kernel.shmax	Defines the maximum size of one shared memory segment in bytes.	90% of the total memory, but if you have a large amount of storage you can leave 512 MB to 1 GB for the operating system instead.
kernel.shmall	Defines the available memory for shared memory in 4 K pages.	You should convert the shmmax value to 4 K (shmmax value x 1024 /4)
kernel.shmmni	Defines the maximum number of shared memory segments.	4096. This amount enables large segments to be created avoiding the need for thousands of small shared memory segments. This parameter may vary depending on your application.
kernel.sem	Four values must be set in this parameter. The first one is the number of semaphores, the second indicates the maximum number of semaphores. The third is the maximum number of semaphores operations within one semop call. And the fourth limits the number of allocatable semaphores.	250 256000 32 1024
kernel.msgmni	Maximum number of queues on the system.	1024
kernel.msgmax	Maximum size of a message in bytes.	65536
kernel.msgmnb	Default size of a queue in bytes.	65536

This table is from *DB2 10 for Linux on System z Using z/VM v6.2, Single System Image Clusters and Live Guest Relocation*, SG24-8036, and it was based on the IBM Knowledge Center website:

`http://publib.boulder.ibm.com/infocenter/db2luw/v10r1`

You can change it to fulfill your database needs.

Storage

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

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

Section 3.7.2, “Virtualized disk” on page 32, 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 used, the Linux OS will spend CPU cycles with disk page caching.

6.4.5 Migrating DB2 and Oracle from x86 to IBM System z

In the following sections, we provide an overview of the steps needed to migrate DB2 and Oracle from x86 to IBM System z. You can find a full example of migrating your DB2 data in Chapter 8, “Hands-on migration” on page 155.

Migrating DB2 databases across platforms

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

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

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

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

Migrating Oracle databases across platforms

Prior to Oracle 10g, one of the only supported ways to move an Oracle database across platforms was to export the data from the existing database and import it into a new database on the new server.

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

1. Connect to the source Oracle database.

2. As a DBA user, issue the SQL query shown here to get the exact name of all table spaces. You will need this information later in the process.

```
SELECT tablespace_name FROM dba_tablespaces;
```

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

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

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

Note: Before importing the dump file, you must first create your table spaces, using the information obtained in step 2 of this list.

Otherwise, the import will create the corresponding data files in the same file structure as at the source database, which might 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.

This method can require an excessive amount of down time if your database is large. Oracle has developed additional methods to migrate from one hardware platform to another:

- Transportable tablespaces - introduced in Oracle 8i to allow whole tablespaces to be copied between databases in the time it takes to copy the datafiles.
- Data Pump export/import - high performance replacements for the original Export and Import utilities.
- Recover manager (rman) - Oracle Database client that performs backup and recovery tasks on your databases and automates administration of your backup strategies.
- Oracle GoldenGate - a comprehensive software package for real-time data integration and replication in heterogeneous IT environments.
- Custom procedural approaches.

6.4.6 Tips for successful migration

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

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

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

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

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

6.5 Backup analysis

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

6.5.1 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. It covers data protection or backup, record retention or archiving, storage management, and security.

Backup concepts

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

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

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

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

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

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.

Archiving concepts

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

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.

6.5.2 z/VM backup

This section shows some options that you have to execute a backup from the z/VM level. These backups can also be used as snapshots of Linux, and can be used for recovering in case of disasters or data corruption. On the section 6.5.3, “Linux backup” on page 96 you can see some tools to back up data on Linux level.

z/VM offline backups

This kind of backup requires planning because it requires stopping the system. You can copy your files to another DASD disk or tape. This is similar to the **dd** tool from Linux. It will copy all the data from one disk to another. If you do not want to manually process your backups, you can check the “IBM Backup and Restore Manager for z/VM” section.

You can see a full example of z/VM offline backups in *Set up Linux on IBM System z for Production*, SG24-8137.

z/VM online backups

Online backups do not require the system to be shut down before performing the backup. Unfortunately, this method should not be applied to back up a running Linux machine because it can lead to data inconsistency. You might want to use it to back up spool files.

You can see a full example of z/VM online backups in *Set up Linux on IBM System z for Production*, SG24-8137.

IBM Backup and Restore Manager for z/VM

IBM Backup and Restore Manager for z/VM is a complete solution to back up and restore data for CMS or non-CMS systems (one file, a group of files, or an entire minidisk) in a VM environment. It is integrated with Tape Manager for z/VM, can compress data during the backup, and supports encryption exits.

You can find more information on the official website:

`http://www.ibm.com/software/products/en/backupvm`

6.5.3 Linux backup

There are various methods of performing backups with Linux on System z. These include command-line tools included with every Linux distribution, such as **dd**, **dump**, **cpio**, **rsync**, as well as, **tar**. These are very useful in the hands of a skilled administrator with experience using these tools. The tools have withstood the test of time. But they do require considerable skill to wield properly.

There are other utilities available that have customized the use of the command-line tools mentioned above. Amanda, for example, was designed to add a more user-friendly interface to the backup and restore procedures, making backup tasks a little easier to manage. It has both a client and server component to facilitate a central backup solution for various remote clients regardless of the platform. Amanda is typically included in all Linux distributions, and it is included with both SLES and RHEL.

Another handy feature of Linux is represented directly in the capabilities of the file system. File systems such as ZFS and BTRFS are capable of taking snapshots. These mechanisms can aid the backup process by allowing the backup software to concern itself with only backing up the static snapshot while allowing new changes to the data to continue unimpeded. This provides for much greater efficiency of the back up process.

Finally, commercial backup utilities, such as the IBM Tivoli Storage Manager, are recommended for an enterprise environment.

Read more about TSM at the following site:

http://www.ibm.com/software/products/en/tivostormana

6.5.4 Migrating backed-up and archived data

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

Why migrate archived data?

Factors that force the migration of archived data include:

- Preserving data on the same medium would face two problems:
  - The lifetime of the medium.
  - The long-term availability of the technology for reading it.
- Eventually, the technology change and your solution become less competitive compared to emerging ones.
- Some older storage technologies have a direct impact on the volume of data that can be stored as well as the space requirements due to the low MBytes/cm3 and Weight/MByte factors.
- End of support for your current solution.

6.5.5 General archival migration considerations

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

- Change in the hardware environment

- Change in the hardware and software environment

Change in the hardware environment

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

Sometimes change in the hardware environment leads to a change of storage technology, which means reorganizing the media data content. Therefore, to allow efficient data retrieval the data inventory structures might need to be reconverted.

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

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

Change in the hardware and software environment

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

6.5.6 Migrating to new backup software

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

Your old software is not compatible with Linux on System z

In this approach, because your new guest is not compatible with the old software, all archived data must be restored to a staging server compatible to the old backup tool. The staging server would be used to restore the archived data and share with the new Linux on System z server that is connected to the new backup software. An example is listed below:

1. From the existing archival software, the archived data needs to be restored to a staging server compatible with the old backup software.
2. The new server running Linux on System z that is used for the current backups and archives would be connected to the staging server (using a shared file system, for example) for accessing the already restored logs (as in the previous steps).
3. The new backup and archival software connects to Linux on System z, accesses the restored data, and rearchives it according to defined organizational attributes and backup policies.

Your old software is compatible with Linux on System z

In this approach, the archived data is restored from the old system to the new Linux on System z server. The exported archive data needs to be rearchived into the new archiving system.

You can either transfer all the data to the new backup software or transfer on demand.

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

6.6 Security analysis

This section discusses the following topics:

- Security migration overview
- Code and application analysis
- Availability and accountability
- Data integrity, assurance, and confidentiality
- Security change management
- Enterprise authentication options
- CP Assist for Cryptographic Function (CPACF)

6.6.1 Security migration overview

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

The security analysis will center around the following areas:

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

Basics of security

Overall security is composed of three domains:

- Physical security
- System security
- Network security

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

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

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

It is generally held that “security through obscurity” is not a valid method. Using open, well-established security methods implemented correctly provides the best defense. For example, instead of developing your own cryptographic libraries, you should instead use open, established ones that have been vetted for many years. Hiding information creates more system administration work and any mistakes may fail to protect against attacks.

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

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

6.6.2 Understanding the z/VM foundation

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

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

System z and existing security policies

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

Processor Resource/System Manager (PR/SM) has been certified through the Common Criteria at Evaluation Acceptance Level (EAL) 5+. For more details about Common Criteria, refer to section 1.2.1, “System z strengths” on page 3.

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

Firewalls and existing security policies

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

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

The second option is to continue to use the physical firewalls by having the inter-security level communication exit the virtual environment via an Open Systems Adapter (OSA), go through the physical firewall, and then return to the virtual environment via a different OSA. Figure 6-13 illustrates the use of an external firewall.

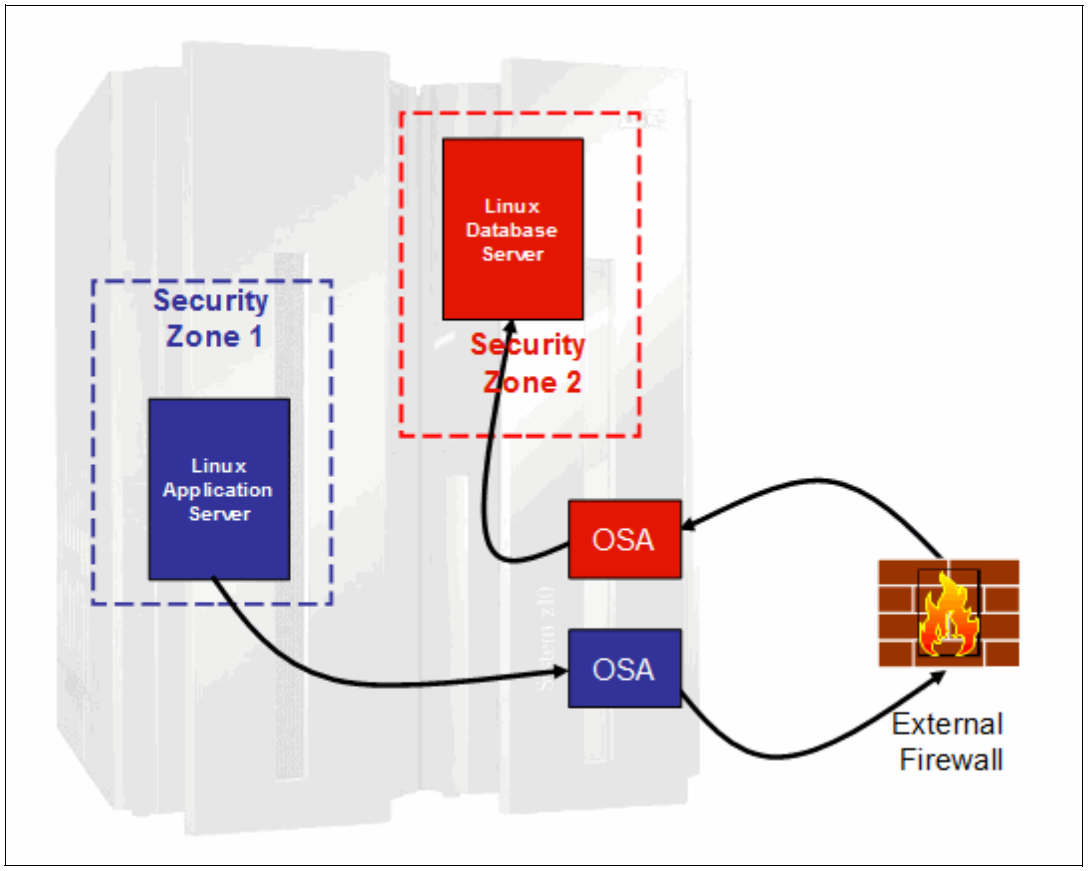


Figure 6-13 Using external firewalls between security zones

In Figure 6-13, the different security zones shown could be in separate LPARs or in the same LPAR. Customers have reported that there is minimal performance impact when using external firewalls.

As mentioned, conforming to the existing security policy can simplify a migration. However, the reality is that for applications within the System z footprint, as shown in Figure 6-13, there may be no requirement for firewalls if all incoming communications to System z are processed by external firewalls.

Control of z/VM

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

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

Security references

For more information about z/VM and hosting Linux on System z, as well as security and networks, refer to the following IBM Redbooks publications:

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

6.6.3 Hardening the base Linux on System z

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

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

Creating a new hardened Linux on System z

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

- *Security for Linux on System z*, SG24-7728

Migrating to a hardened Linux on System z

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

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

Important: RHEL includes the SELinux security method, and SLES includes AppArmor for its enhanced security method. Determine whether those environments are in use or required, and plan accordingly.

Those mechanisms are very complex, so invest the time to identify code and applications that have not been ported to work in these environments.

Maintaining a hardened Linux on System z

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

6.6.4 Code and application analysis

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

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

6.6.5 Security issues

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

Migrating code

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

Migrating applications

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

6.6.6 Dependencies

This section discusses determining dependencies before you migrate.

Code dependencies

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

Application dependencies

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

6.6.7 Checking user input

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

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

6.6.8 Planning for updates when migrating code

When code is migrated to an enterprise-class system, changes need to be addressed in a different manner. Unlike less critical code, changes must be allowed to be executed while the application is still running. Thus, you must ensure that a method is in place to signal that configuration and control files have been updated and need to be reloaded.

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

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

6.6.9 Networking

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

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

6.6.10 Logging and recording events

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

Implementing syslog logging when migrating code

On the Linux, syslog-ng will be running. Take time to update the code as needed to send messages to this daemon. At the very least, all information that deals with security should be logged, as well as critical state information. The benefit of implementing syslog functionality is that log maintenance will be performed by the system (as in log rotation and archiving).

6.6.11 Escalations of authority

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

Migrating code

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

Migrating applications

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

6.6.12 Security test plan and peer review

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

6.6.13 Availability and accountability

Security involves much more than simply who can access a system. It also involves keeping the system available to authorized users and unavailable to unauthorized uses. Denial-of-service attacks (DoSs) have become more frequent in recent years, and Internet-facing systems must take the possibility of such threats into account.

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

Availability analysis

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

Internet-facing Linux considerations

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

Communicating availability

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

6.6.14 Accountability analysis

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

Making log files immutable

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

Another approach to securing system logs is to use a remote log server, as supported by syslog-ng. See an example of this in section 7.4, “Deploying central log server” on page 146. The logs on the remote log server are not necessarily immutable, but they are not directly writeable from a system that has been compromised.

Audit trails encompassing all security domains

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

Authentication

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

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

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

6.6.15 Data integrity and confidentiality

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

Data integrity analysis

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

- Who can access what data and what is allowed
- Whether there is an audit trail in place to map who changed what and when

- Whether the data is corrupted in some way and how is it to be restored
- Whether there is a disaster recovery plan in place

Protecting data at rest from unauthorized access

Protecting access to a database is well understood, but what about protecting raw data on the disk itself? Mobile computers with databases full of accounts or data are sometimes misplaced or stolen. Thus, you need to protect data “at rest” (meaning the files themselves) and ensure that the data is kept in a secure way. You should prevent offline copies of a database from being kept on portable devices or drives. Control of data is key. Be sure to communicate the data integrity policy to all individuals who have access, and monitor the audit trail to make sure that the policy is being enforced.

Data backups: part of security

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

6.6.16 Confidentiality analysis

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

Understanding laws and regulations before an incident occurs

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

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

Publishing your confidentiality policy

You need to communicate the confidentiality policy in such a way as to ensure that all users of the system are aware of it and thus can be held accountable. When a user logs in to a system, use the Message Of The Day (MOTD) found in `/etc/motd` as shown in Example 6-3 on page 107 to communicate with your system users.

Example 6-3 Use /etc/motd to communicate system policy

```
************************************************
*      .--.     Welcome to the Linux s/390x VM     *
*     |o_o |    SUSE Linux Enterprise Server 11 SP3*
*     |:_/ |    System Admin: John Doe             *
*    //   \ \                 jdoe@company.com     *
*   (|     | )  This system governed by corprate   *
*  /'\_   _/`\  Policy K49-r v21 please read       *
*  \___)=(___/  before accessing system            *
************************************************
```

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

On web pages, create a link from the main page so that the system policy can be easily accessed. If you are allowing VNC login, display the policy by updating `/etc/gdm/custom.conf` as shown in Example 6-4.

Example 6-4 Policy found in /etc/gdm/custom.conf

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

Having a plan in place before an incident occurs

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

- Who should be notified and what should be disclosed about the incident.
- If there is a requirement to notify the public, document how and what should be disclosed.

Communicate actions that will be taken to prevent future incidents.

6.6.17 Security change management

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

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

Testing changes with a clone of the Linux on System z

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

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

6.6.18 Enterprise authentication options

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

A common centralized LDAP server

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

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

LDAP server on z/OS means RACF integration

If RACF is used to manage user information, then installing LDAP on a z/OS system will allow LDAP access to RACF. In turn, this allows a single, highly secure repository of user information in RACF and lets that information be exposed to Linux VMs via an LDAP server. For more information, refer to *Security for Linux on System z*, SG24-7728.

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

6.6.19 Integrated Cryptographic Service Facility

When migrating to Linux on System z, the underlying hardware has the ability to accelerate cryptographic mathematics. The CP Assist for Cryptographic Function (CPACF) supports synchronous cryptographic functions. The work is processed by the crypto-assist processor that is integrated into every processing unit (PU) of every IBM System z or the Crypto Express card, if it is installed.

The supported APIs are listed here.

OpenCryptoki

An open source implementation of Public-Key Cryptography Standard #11 (PKCS#11), OpenCryptoki uses the libica shared library to access IBM cryptographic adapters through the z90crypt device driver.

OpenSSL

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

Global Security Kit

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

Using this approach will offload the cycles and allow for more concurrent access to a web server that is using SSL or applications that use one of supported APIs. Refer to *The Virtualization Cookbook for IBM z/VM 6.3, RHEL 6.4, and SLES 11 SP3*, SG24-8147, to learn

how to configure your system so that your Linux on System z will take advantage of the installed hardware.

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

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

6.7.1 The operational environment

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

Operational tasks might be affected by the source application migrating to the target application running on Linux on System z.

6.7.2 Operational migration tasks

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

- Starting and stopping the application

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

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

- Notification of problems

  Sometimes operators can receive automated messages or indicators that they are unfamiliar with and do not know how to respond to. Operators need to know who to turn to for guidance when this type of problem arises, so the application owner needs to be clearly identified. If the application owner is unavailable or unresponsive, 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. Often this simply involves monitoring indicators or displays that show the health of the application. New procedures for the migrated target application must be communicated to the operators. Hands-on training sessions are optimal for operators as they learn by observation and perform required tasks.

- Hardware manipulation

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

- Hardware intervention and problem escalation

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

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

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

- Batch procedures and scheduling

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

- Other considerations

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

6.7.3 Single system image and live guest relocation

As seen in section 3.5, “Single system image and live guest relocation” on page 28, single system image (SSI) and live guest relocation (LGR) will simplify z/VM systems management. From the operational side, it will enable software or hardware maintenance and upgrades without disruption to the business. Operators or z/VM SysAdmins are able to move guests to other members that are on the same or separated System z servers. No disruption is necessary. You can move those guests across IBM System z10® EC, IBM System z10BC, IBM z196, IBM z114, IBM zEC12 and IBM zBC12.

Tip: The LGR can also be used for workload balancing.

6.7.4 IBM Wave for z/VM

To reduce the complexity of z/VM management, IBM Wave for z/VM is a perfect solution to help SysAdmins and Operators in their daily tasks. You can read more about IBM Wave for z/VM benefits in section 1.5.1, “Empowered virtualization management: IBM Wave for z/VM” on page 11. A list of features that may help on maintenance tasks is listed below:

- Display and manage virtual servers and resources, all from the convenience of a single graphical interface.
- Provision Linux guests, network, and storage from a single user interface.
- Capture and clone virtual servers across LPARs and CPCs.
- Activate and deactivate z/VM guests in the current z/VM system.
- Lock or unlock z/VM resources.
- Create and configure virtual switches (VSWITCHes) and guest LANs.
- Provide storage management and provisioning for z/VM and Linux.
- Run shell scripts or REXX EXECs directly from the user interface for more customized management and provisioning.
- Support advanced z/VM capabilities such as SSI and LGR. Perform a live guest relocation of one or more z/VM guests.

More information about IBM Wave can be found in *IBM Wave for z/VM: Installation, implementation and exploitation*, SG24-8192.

6.8 Disaster recovery and availability analysis

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

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

Disaster Recovery (DR)

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

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

`http://www-03.ibm.com/systems/services/labservices/solutions/hacoc.html`

High Availability

Provide service during defined periods, at acceptable or agreed upon levels, and mask unplanned outages from users. High Availability (HA) employs Fault Tolerance, Automated

Failure Detection, Recovery, Bypass Reconsideration, Testing, Problem, and Change Management.

Continuous Operations

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

Continuous Availability

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

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

6.8.1 Availability analysis

Migrating an application to a virtualized Linux environment on IBM System z offers an opportunity to implement an availability profile in line with the impact of the unavailability that the application has on the organization's overall business. Sometimes, however, such an analysis is not straightforward. For example, test and development workloads are generally not considered to be mission-critical. However, because they may be needed to correct an error in a production system, consider providing for some sort of test and development environment in your DR planning.

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

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

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

6.8.2 Single points of failure

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

Table 6-3 lists the components of an IBM System z virtualized environment running an application under z/VM and Linux and the relative costs of rectifying a single point of failure.

Table 6-3 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

Single point of failure	Probability of failure	Cost to rectify
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 requiring a reboot
- Application maintenance

6.8.3 System z features for High Availability

IBM System z has been designed around providing High Availability. Perhaps the most design effort has gone in to the transparent recovery of processor errors. In the event of a hard processor error at an individual core level, the task is moved to a spare processor where processing continues transparently to the application. In the IBM zEnterprise, a number of availability features have been introduced to reduce the number of planned system outages. For example, the following actions are now fully concurrent and require no system outage:

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

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

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

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

- Customer-Initiated Upgrade (CIU) is used for a permanent capacity upgrade.
- Capacity BackUp (CBU) is predefined capacity for DR. A system at a DR site does not need to have the same capacity as the primary site. In the event of a declared disaster, or for up to 5 DR tests, the customer can turn on the number of processors, including IFLs, required to handle the workload from the primary site.

- Capacity for a Planned Event (CPE) is used to replace capacity lost within the enterprise due to a planned event such as a facility upgrade or system relocation.

On/Off Capacity on Demand provides extra capacity in 2-hour increments that is available to be turned on to satisfy peak demand in workloads.

Note: For more information about IBM zEnterprise System, refer to *IBM zEnterprise EC12 Technical Guide*, SG24-8049.

6.8.4 Availability scenarios

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

In all scenarios, it is assumed the IBM zEnterprise System 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 zEnterprise System.

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

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

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

Another important factor in the availability of applications is security and access controls. For more information about this topic, refer to 6.6, "Security analysis" on page 98.

Single System z LPAR: Clustered WebSphere Application Server

Figure 6-14 on page 115 shows a System z LPAR sharing system resources to all Linux virtual machines in the LPAR. The WebSphere Application Servers are in a two-node cluster. If the Integrated Facility for Linux (IFL) fails, IBM zEnterprise System will automatically switch the workloads to a spare or any unassigned processor without any disruption to the active task.

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

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

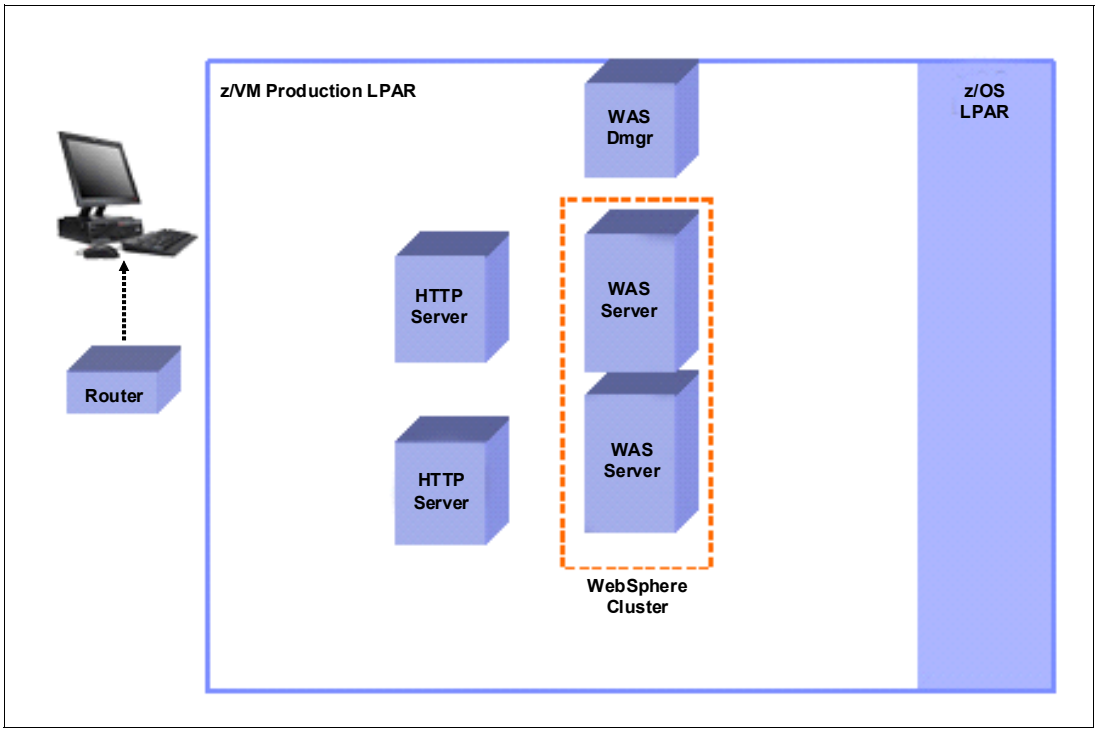


Figure 6-14 Single LPAR WebSphere Application Server cluster

This environment also provides additional availability through redundant HTTP servers.

Multiple LPARs: HA solution for Linux on System z

Figure 6-15 on page 116 shows a scenario where there are three LPARS defined. Each LPAR could have a dedicated IFL or a single IFL, or multiple IFLs could be shared among all LPARs. The LPAR weight determines the relative priority of an LPAR against other LPARs.

In this case, the production workload and WebSphere Application Server cluster is split across two LPARs, which give HA to WebSphere Application Server because an LPAR or z/VM failure will not impact the availability of WebSphere Application Server.

Development and test workloads run in their own LPAR so any errant servers will have no impact on the production workloads. As in the first scenario, a failure of a System z IFL will be rectified automatically without any impact to the running application.

This configuration eliminates most failure points at a reasonable cost.

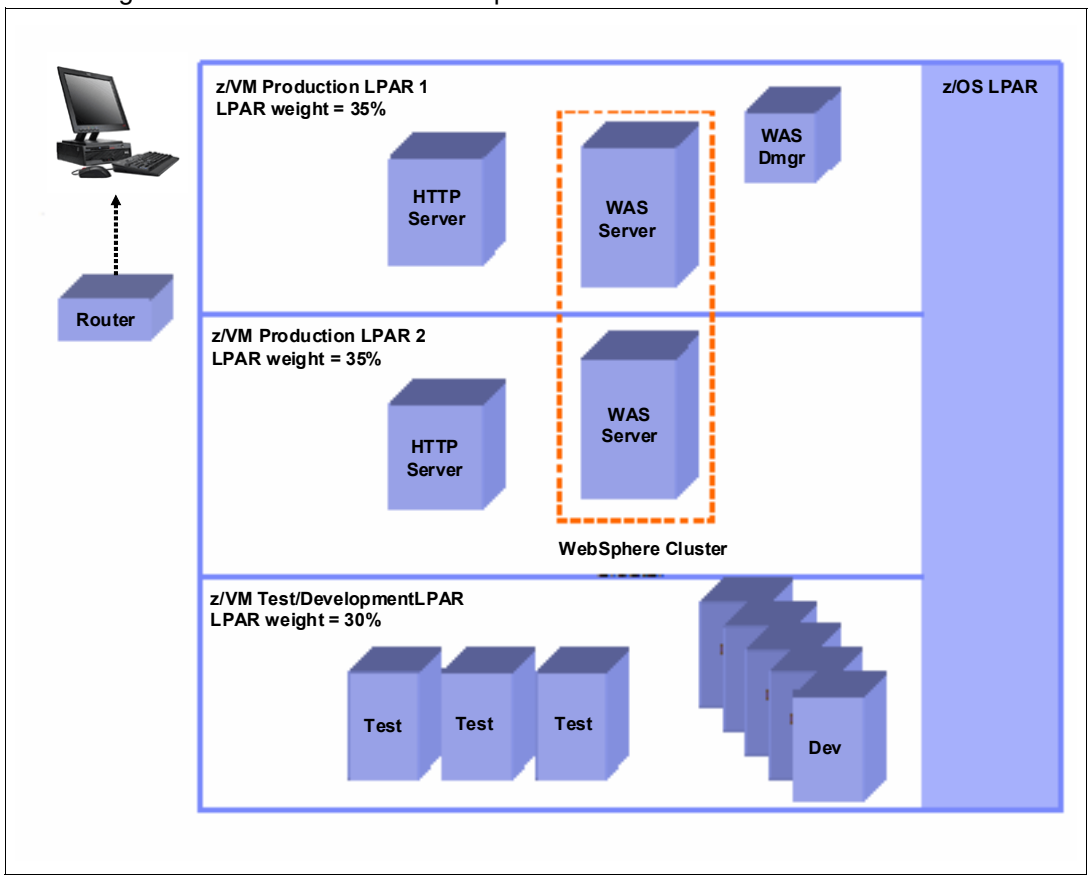


Figure 6-15 HA Linux environment on IBM System z

Active/cold standby cluster

Figure 6-16 on page 117 describes another approach in which, instead of having redundant virtual servers, an active/cold standby cluster is established. In this case, Tivoli System Automation for Multiplatforms (SA MP) monitors the servers and in the event of an outage will automate failover to the cold standby server.

SA MP runs on each node in the cluster. It monitors cluster nodes and exchanges information through Reliable Scalable Cluster Technology (RSCT) services. SA MP 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 MP 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 MP starts the HTTP Server on Node 1 and at user-defined intervals invokes a script to confirm that it is still up and serving pages. It also monitors the Linux node itself to ensure it remains active.

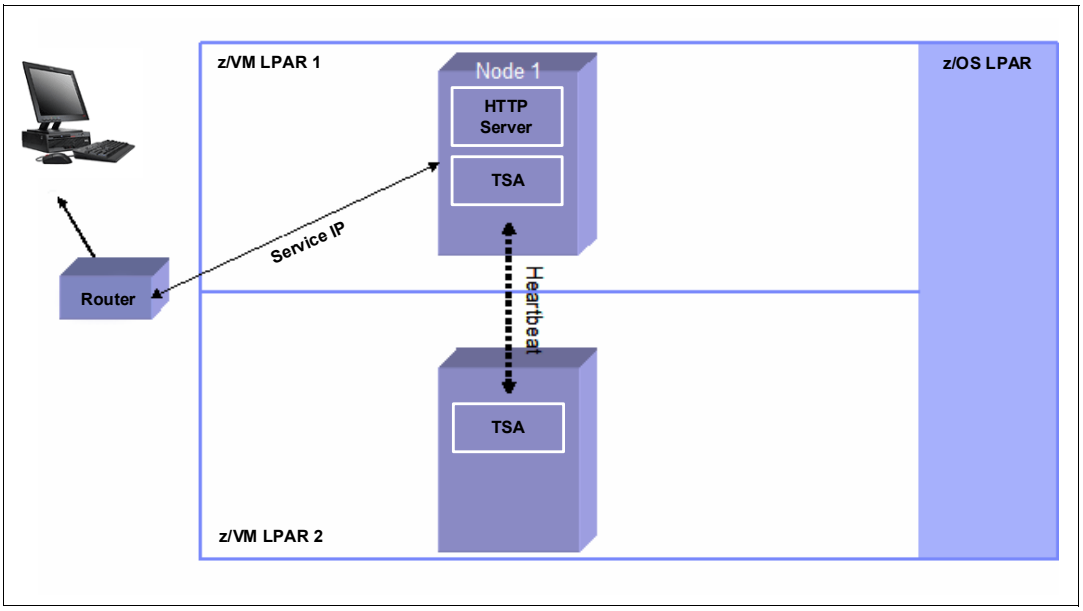


Figure 6-16 Normal situation: Tivoli System Automation monitors for outages

When a failure occurs, RSCT determines that Node 1 is no longer responding. SA MP then moves the Service IP over to Node 2 and restarts the HTTP server there, as illustrated in Figure 6-17.

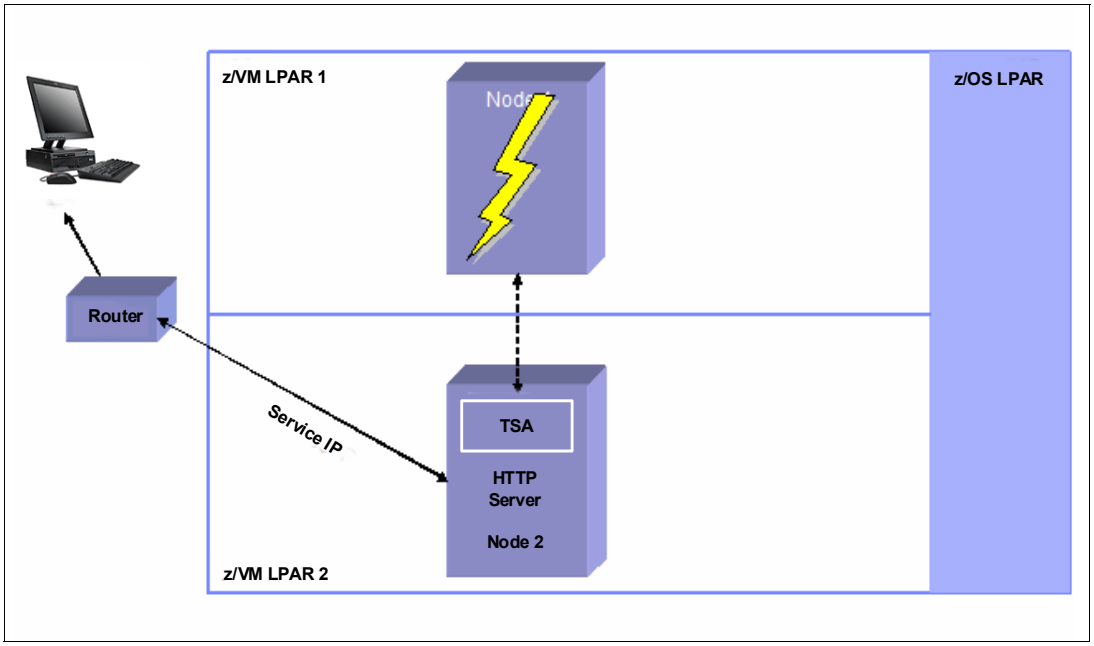


Figure 6-17 Outage occurs: Tivoli System Automation fails over to cold standby

Active/active application server cluster

Figure 6-18 on page 118 shows the WebSphere Application Server setup in an active/active configuration where the WebSphere Application Server Cluster spans two Linux virtual machines in two LPARs. This setup handles the very rare occurrence of the failure of an LPAR. More importantly, it also allows z/VM maintenance to be performed without an outage to the WebSphere applications. In this case, the Linux servers and z/VM are shut down in LPAR 2. An initial program load (IPL) is done of z/VM with new maintenance applied and the

Linux virtual machines are restarted and the WebSphere cluster is restored. This task would be scheduled for a time when the processing load is light. Live Guest Relocation could also be used to avoid an outage due to z/VM maintenance.

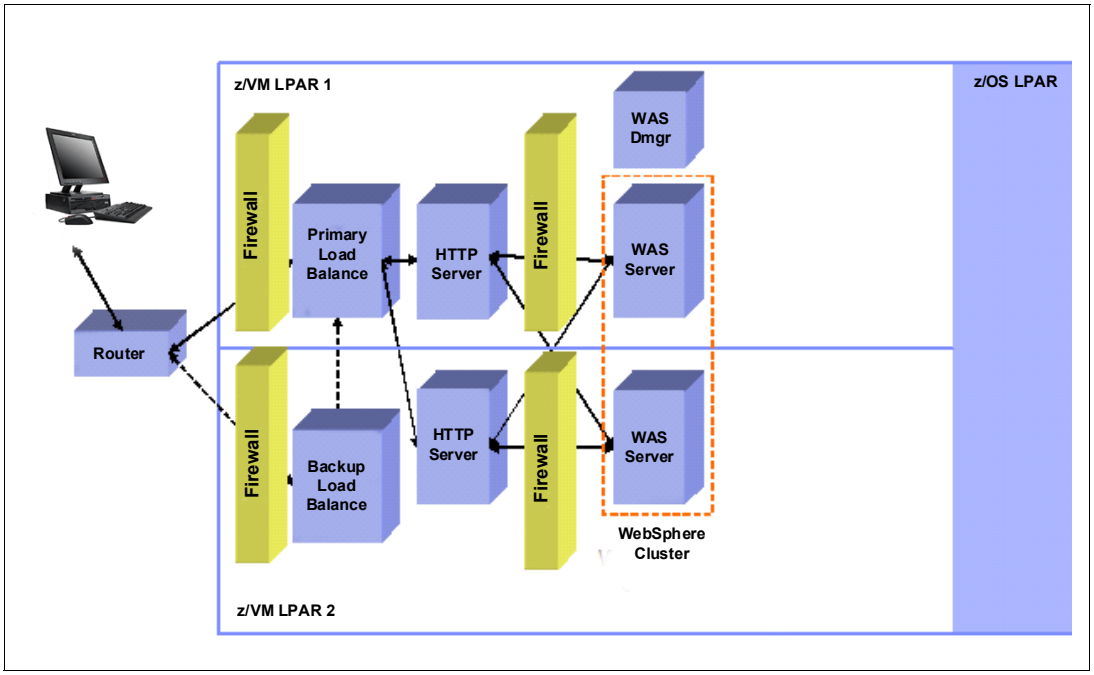


Figure 6-18 Active/active WebSphere Application Server cluster

Active/active WebSphere Application Server cluster with database replication

Figure 6-19 on page 119 shows a DB2 database added to the active/active WebSphere cluster. To provide HA for the DB2 database, the DB2 data replication feature, High Availability Disaster Recovery (HADR) is used. HADR protects against data failure by replication changes from the source database (called *primary*) to a target database (called *standby*).

In the event of a z/VM or LPAR outage of the primary DB2 system, the standby DB2 system will take over in seconds, thus providing high availability. Communication between the DB2 primary and DB2 standby systems is via TCP/IP, which in this case would be done using the System z high speed virtual network feature HiperSockets.

The Standby DB2 system can also be at a remote site to provide enhanced availability in the event of a site failure.

IBM Tivoli System Automation for Multiplatforms (SA MP) running in both DB2 servers is designed to automatically detect a failure of the primary, and it issues commands on the standby for its DB2 to become the primary.

Other cluster management software could be used. However, SA MP and sample automation scripts are included with DB2 to only manage the HA requirements of your DB2 database system.

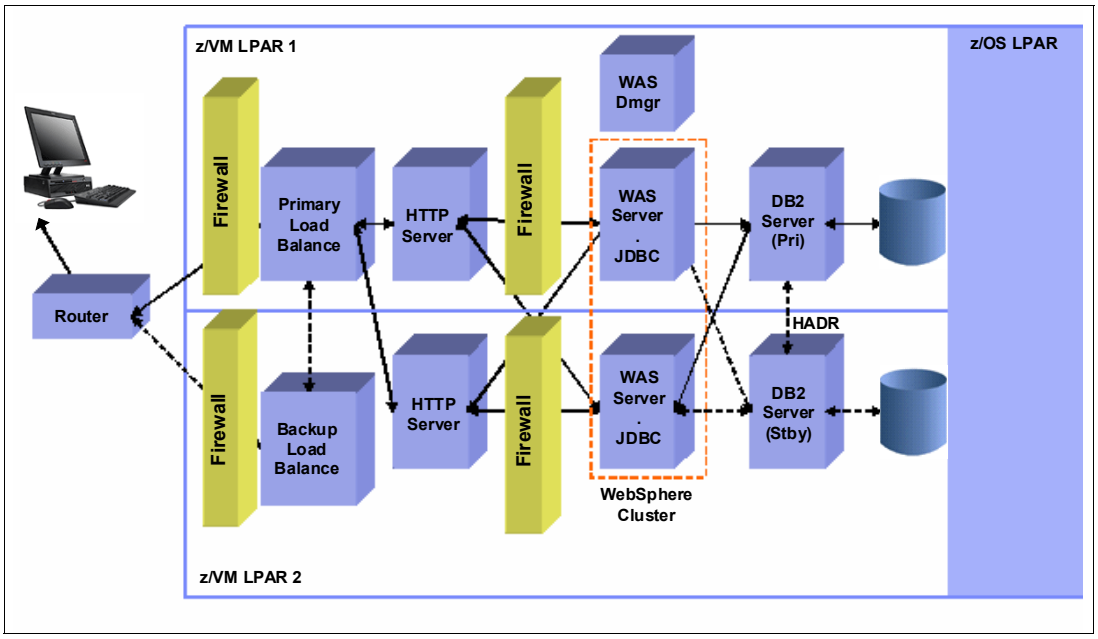


Figure 6-19 Active/active WebSphere Application Server cluster and DB2 HADR

Active/active WebSphere Application Server cluster with database sharing

Figure 6-20 on page 120 shows that database sharing was introduced using Oracle Real Application Clusters (RAC). Oracle RAC provides HA for applications by having multiple RAC nodes sharing a single copy of the data. If a cluster node fails, the in-flight transaction is lost but the other server in the RAC can receive all Java Database Connectivity (JDBC) requests.

In a System z environment, communication between the database nodes would use a virtual LAN in the same LPAR or HiperSockets to other LPARs. Both methods are at memory-to-memory speeds with very low latency.

For more information about Oracle RAC, go to the following website:

`http://www.oracle.com`

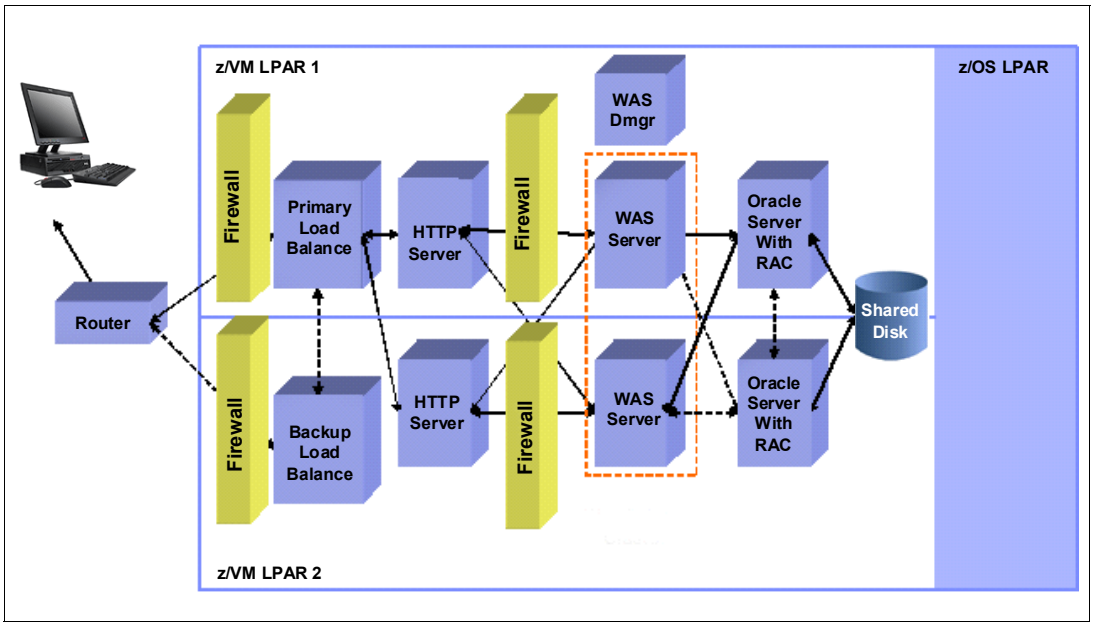


Figure 6-20 Database sharing using Oracle RAC

Active/active WebSphere Application Server cluster with DB2 sharing in z/OS Parallel Sysplex

In Figure 6-21 on page 121, we introduce the additional benefits provided by the z/OS IBM Parallel Sysplex®. Briefly, a Parallel Sysplex is a High Availability configuration designed to provide CA of systems and applications. In the case of DB2 data sharing, the Parallel Sysplex allows all members of the sysplex update access to shared data by using a centralized arbitrator known as the coupling facility (CF).

Each WebSphere Application Server is configured to use the JDBC Type 4 driver for communication with the DB2 z/OS data sharing members. It is sysplex-aware and works cooperatively with DB2 and the z/OS Workload Manager (WLM) on z/OS to balance workloads across the available members of the DB2 data sharing groups.

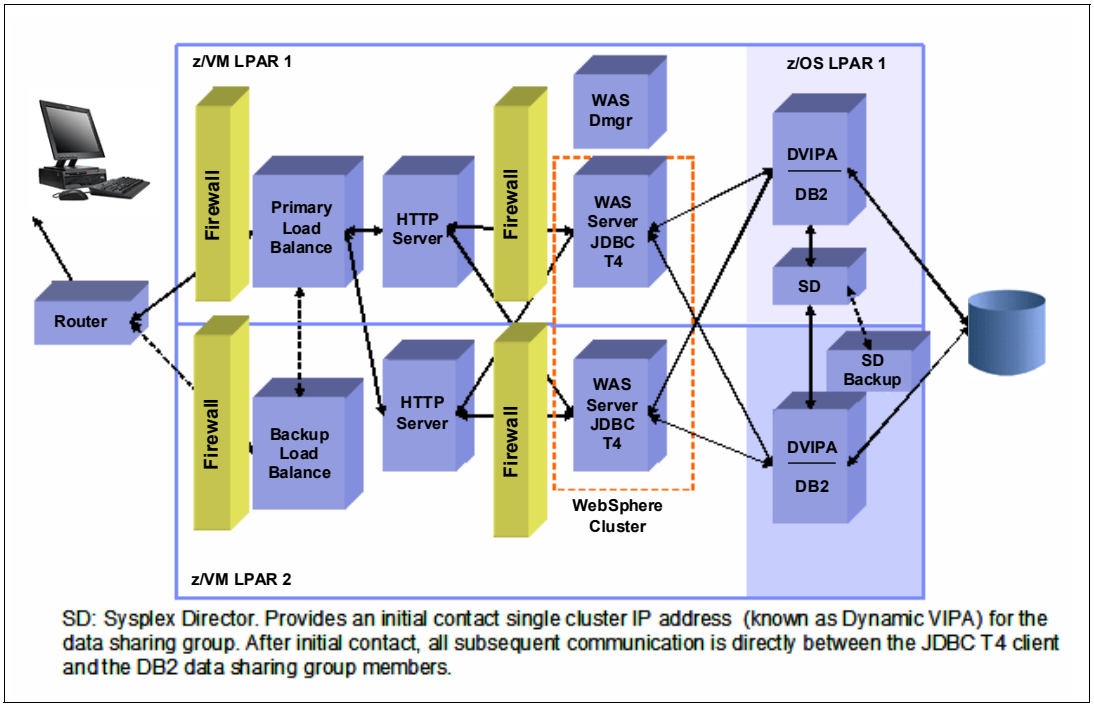


Figure 6-21 Database sharing using z/OS Parallel Sysplex

Active/active WebSphere Application Server cluster with database sharing on z/OS across cities

For the ultimate availability solution, it is possible to have two sites up to 100 km (62 miles) apart and provide full DB2 data sharing between WebSphere Application Server clusters at each site. The key element in this solution is Globally Dispersed Parallel Sysplex (IBM GDPS®) Metro Mirror. GDPS Metro Mirror uses a feature on the IBM ESS800 and IBM DS6000™ and DS8000 family of storage systems called Peer-to-Peer Remote Copy (PPRC).

All critical data resides on the storage subsystem (or subsystems) in Site 1 (the primary copy of data) and is mirrored to Site 2 (the secondary copy of data) via Synchronous PPRC. With Synchronous PPRC, the write to the primary copy is not complete until it has been replicated to the secondary copy. PPRC is designed to make it possible for a site switch with no data loss.

The primary Controlling System (K1) running in Site 2 performs the following services:

- It monitors the Parallel Sysplex cluster, Coupling Facilities, and storage subsystems, and maintains GDPS status.
- It manages a controlled site switch for outages of z/OS and Parallel Sysplex, z/VM, and Linux on System z (as a guest under z/VM).
- It invokes IBM HyperSwap®[1] on z/OS and z/VM for a site switch of disk subsystems, which can eliminate the need for an IPL at the recovery site to use the mirrored disks.
- It works with Tivoli System Automation Multiplatform across z/VM and Linux to understand their state and coordinate their restart during the recovery phase.
- It invokes network switching, based on user-defined automation scripts.

[1] HyperSwap is a z/OS feature that provides for the continuous availability of storage devices by transparently switching all primary PPRC disk subsystems with the secondary PPRC disk subsystems for planned and unplanned outages.

Figure 6-22 shows that Site A and Site B are in a GDPS and share the same DB2 data. GDPS helps to automate recovery procedures for planned and unplanned outages to provide near-Continuous Availability and Disaster Recovery capability.

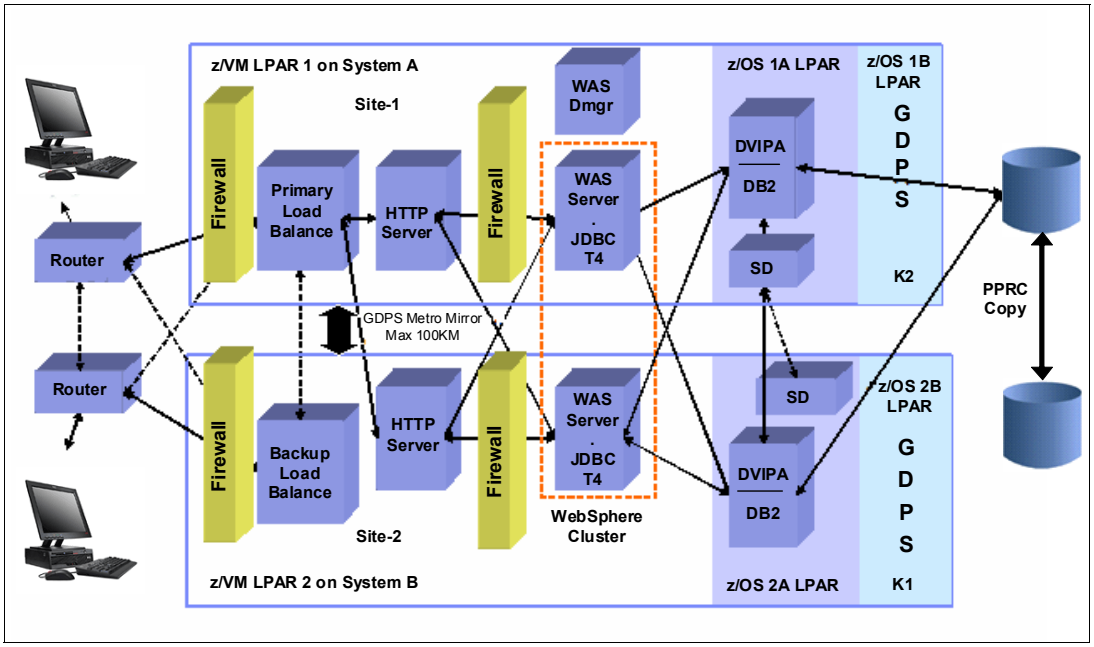


Figure 6-22 GDPS solution for near-continuous availability

Distances greater than 100 km (62 miles) require an asynchronous copy where the application resumes after the write is performed to the primary copy. The write to the remote mirror takes place later, so it is not synchronized with the primary copy. More detailed description of this topic is beyond the scope of this book.

6.8.5 Linux-HA Project

The Linux-HA Project provides HA solutions for Linux through an open development community. The majority of Linux-HA software is licensed under the Free Software Foundation GNU Public License (GPL) and the Free Software Foundation GNU Lesser General Public License (LGPL).

You can read more about Linux-HA project in their official website:

`http://www.linux-ha.org`

Note: For more details about Linux-HA and examples of its use in a z/VM Linux on System z environment, refer to *Achieving High Availability on Linux for System z with Linux-HA Release 2*, SG24-7711.

6.8.6 High Availability add-ons provided by SUSE and Red Hat

There are two solutions that are worth your attention when you are looking for a High Availability third-party solution. Depending on the distribution that you have chosen for your environment (SUSE or Red Hat), you can find add-ons that will facilitate the HA implementation.

For SUSE Linux Enterprise High Availability Extension, see the following site:

http://www.suse.com/products/highavailability

For Red Hat Enterprise Linux High Availability Add-On, see the following site:

http://www.redhat.com/products/enterprise-linux-add-ons/high-availability

6.8.7 Understanding the availability requirements of your applications

This section describes how service level agreements (SLAs) and the cost of providing availability can help you achieve a better understanding of the availability requirements of your applications.

6.8.8 Service level agreements

To determine the availability requirements of applications that you want to migrate to Linux on System z, you must take into account the needs of the business units that rely on these applications. Ideally, SLAs are in place that state requirements, such as availability needs, response time, maximum system utilization, DR requirements. This should be the basis for the design of the target system on Linux.

If SLAs do not exist, before starting to design a solution, discuss with the business units what levels of service you can offer and what level of investment they are willing to make. The key to the success for an SLA is that it is both achievable and measurable with defined penalties for failure to deliver. You also need to ensure that there are regular reviews because things will change.

According to IT Service Management principles, a service level agreement would typically define or cover the following topics:

- The services to be delivered
- Performance, tracking, and reporting mechanisms
- Problem and change management procedures
- Dispute resolution procedures
- The recipient's duties and responsibilities
- Security
- Legislative compliance
- Intellectual property and confidential information issues
- Agreement termination

Some of these components might 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
  - 6:00 am to 6:00 pm, weekdays
  - 9:00 am to 5:00 pm, weekdays, and so on
  - Definition of how availability is measured and who will do the measurement. For example, system availability, application availability, database availability, network availability
- Minimum system response time

  Defined number and definition of where and how is it measured

6.8.9 The cost of availability

As shown from the examples in this chapter, there is a great degree of difference in cost and complexity of the various availability options discussed. Providing CA and a DR plan is not an insignificant expense but with the degree of reliance on IT systems by most businesses today, it is a cost that cannot be ignored.

If you have a web-facing revenue-generating application, you can calculate the cost of downtime by simply monitoring the average revenue generated over a period of time. This provides an idea of the amount of revenue that may be lost during an outage and how much you should spend to make the application more resilient. Other businesses will have different ways of calculating the cost of downtime.

Keep in mind that for any HA configuration to be successful in a real DR situation, there needs to be a fully documented DR plan in place that is fully tested at least once every year.

7

Deployment of workloads

Enough talk, and enough planning; it is time to deploy workloads to Linux on System z. There are many things to analyze and consider leading up to the deployment of workloads to the mainframe. When the proper planning is completed, the migration should move smoothly.

As mentioned in 6.3, “Application analysis” on page 79, there are many workloads that represent a “perfect fit” on System z. Not all can be demonstrated in this book. The migration of some very practical applications, such as IBM DB2, are illustrated as a hands-on exercise in Chapter 8, “Hands-on migration” on page 155. Mission critical applications, ERP, CRM, business intelligence, and more, are clearly what you want running on System z, but only generic examples can be included in a guide such as this; your specific details for your migration do not necessarily distill into a demonstration. Following the guides, the checklists, and the information contained previously in this book, and using this chapter of examples, will lead you to success.

Standard infrastructure applications are also very well suited on the IBM mainframe, and these are just as critical. In this chapter, the deployment of some standard services is demonstrated. Such an illustration of deploying standard services should likewise represent a pattern that can be followed.

In this chapter, we provide examples of deploying workloads using High Availability clustering as well deploying the application MediaWiki and My SQL. We provide an example of deploying OpenLDAP, a central log server, and a file and print service.

7.1 Deploying High Availability clustering

Both Red Hat and SUSE deliver add-on products that provide High Availability failover clustering for their respective Linux operating systems. These products can be deployed to increase the availability and reliability of Linux workloads on System z. In the event of a catastrophic event that takes one cluster member down, the applications running can be automatically brought online on one of the partner members of the cluster, with no perceived downtime. SLES High Availability Extensions (HAE) even offers geodistribution clustering, enabling the restart of a database or application from a remote site.

7.2 Deploying MediaWiki and MySQL

With the open source Linux operating system comes a wide variety of open source applications. A very popular application for Linux is MediaWiki, the general-purpose wiki that originated with Wikipedia. It is written in PHP and uses MySQL as its backend database. This configuration is commonly known as a LAMP server, meaning that the application employs Linux, Apache, MySQL, and PHP. This Web 2.0 stack is an ideal workload for Linux on System z.

The Linux environment on x86 is largely the same as it is on System z (with a few notable exceptions). Configuration files on the x86 will be in the same place on your Linux guest on System z, unless you deliberately choose to keep them in a different place. Hence, the MySQL configuration files, for example, will typically only need to be copied from the x86 server to the Linux on System z server, placed in the same location in the file system, /etc/my.cnf.

Best practices dictate that migrating to System z be performed first to a test environment. Then, after successfully testing the deployment in the test environment, migration to the production environment is appropriate.

In this example, the MySQL database is contained on its own disk partition on an external iSCSI disk. Likewise, the DocumentRoot of the Apache webserver is also stored on its own external iSCSI disk, different than the LUN where the MySQL database is stored.

In today's data centers, it is common to store application data on external disks in this way. This practice makes it much easier to move (or migrate) services from one host to another. It commoditizes the operating system and the various services. This is a best practice in the industry: maintaining the configurations of the operating system and services using a configuration management tool, and deploying the operating system and services configurations out to virtual machines either by using a cloud tool or an external application such as Puppet, Chef, or caffeine-hx. The infrastructure can easily be adapted and scaled to meet demand, while keeping the data available universally from a main storage system.

In our example, we stored our application data on external disks. While it is possible to back up the data, transfer the data from one host to another, and reimport the data into the new running service, such a method will always be one way to accomplish migration, although it is a slow method. The method demonstrated here is a much faster way to accomplish migration tasks.

7.2.1 Analysis and planning

Following the guidelines and recommendations outlined in Chapter 5, "Migration planning" on page 49, and Chapter 6, "Migration analysis" on page 57, appropriate planning and analysis

should be performed before these migration activities. The checklists are very helpful in identifying how virtual resources should be dedicated and organized.

For this example scenario, the z/VM guest has already been set up and a minimal Linux operating system installed. The Linux guest is called LNSUDB2 and it is running SLES11 SP3, with one virtual CPU, and 1 GB of virtual memory. It is presumed that an adequate package management (RPM) repository for installation source is already set up and available for the installation of the application software that will be used.

7.2.2 Installing the LAMP stack

The installation of the application software can be done using YaST for SLES11. To better illustrate the universality of LAMP on both SLES and RHEL, the command-line interface (CLI) will be used for these instructions, with sample commands for both SLES and RHEL.

Install LAMP on SLES

First, ensure that SLES has a pattern (a collective group of related packages) for a LAMP server. Run **`zypper info -t pattern lamp_server`** to see the packages that are associated with a LAMP server.

Example 7-1 shows the helpful information that is displayed about LAMP by running the command:

```
zypper info -t pattern lamp_server
```

Example 7-1 LAMP pattern output from zypper

```
lnsudb2:~ # zypper info -t pattern lamp_server
Loading repository data...
Reading installed packages...


Information for pattern lamp_server:

Repository: SLES_DVD
Name: lamp_server
Version: 11-38.44.33
Arch: s390x
Vendor: SUSE LINUX Products GmbH, Nuernberg, Germany
Installed: No
Summary: Web and LAMP Server
Description:
Software to set up a Web server that is able to serve static, dynamic, and inter
active content (like a Web shop). This includes Apache HTTP Server, the database
management system MySQL, and scripting languages such as PHP, Python, Ruby on R
ails, or Perl.
Contents:

S | Name                  | Type    | Dependency
--+-----------------------+---------+-----------
  | mysql                 | package |
  | apache2-mod_php5      | package |
  | apache2-mod_python    | package |
  | apache2-prefork       | package |
  | libapr-util1          | package |
```

```
  | libapr1               | package |
  | apache2               | package |
  | apache2-doc           | package |
  | apache2-example-pages | package |
lnsudb2:~ #
```

Note: The lamp_server pattern includes the Apache and MySQL components, but is missing the PHP component. That is simply because the “P” in “LAMP” could be PHP, or Perl, or even Python. In fact, though it does not start with a “P,” Ruby is often used as the server-side dynamic web page engine.

Install the packages for Apache and MySQL by running the command:

```
zypper install -t pattern lamp_server
```

Zypper reports which packages are expected to be installed, then prompts for confirmation to continue. Pressing “y” and “Enter” will begin the installation of the packages.

Install the remaining PHP packages by running the command:

```
zypper install apache2-mod_php53 php53-mysql
```

Install LAMP on Red Hat Enterprise Linux

Although RHEL has a similar mechanism for representing collections of packages as groups as SLES does as patterns, RHEL does not have a group for LAMP packages. So installing the LAMP packages involves specifying four different groups:

```
yum groupinstall “Web Server” “MySQL Database server” “PHP Support”
```

Yum reports which packages are expected to be installed, then prompts for confirmation to continue. Pressing “y” and “Enter” will begin the installation of the packages.

In addition to the packages selected by the indicated groups, another package must be installed on the RHEL server:

```
yum install php-mysql
```

7.2.3 Starting and testing LAMP components

Before migrating the MediaWiki software, there is wisdom in choosing to configure and test Apache, PHP, and MySQL on the target system to ensure that they’re working. This reduces the number of variables to debug if something goes wrong.

The Apache and MySQL configurations in this example scenario are simple, whereas your configuration may be more complex. Migrating the Apache and MySQL configurations may be a more complex process. This example presumes that MediaWiki is the only application configured for Apache and that no other data exists in the MySQL database than what is used by MediaWiki.

Although version information about the packages may have been noticed while installing the packages, it is helpful to confirm that the version of Apache is what is expected. A common method of displaying the version is by running **`apachectl -v`**. This is the same command for SLES as it is for RHEL.

Example 7-2 shows the version of apache2 as displayed by issuing the command in SLES:

apachectl -v

Example 7-2 Output of **apachectl -v**

```
lnsudb2:~ # apache2ctl -v
Server version: Apache/2.2.12 (Linux/SUSE)
Server built: March 27 2013 18:57:40
```

Historically, it was common to have the installed services started automatically when the package was installed. Today, it is more common that the distribution takes a more active role in ensuring that potentially renegade software does not start automatically. Hence, it is necessary to start Apache manually, and to set it to start automatically each time the system is booted.

Apache services on SLES

Set the apache2 service to automatically start each time the server is booted, then manually start the service using the following commands:

chkconfig apache2 on

service apache2 start

Example 7-3 shows the expected output from the commands that start the apache2 web service.

Example 7-3 Starting apache2 web service

```
lnsudb2:~ # chkconfig apache2 on
lnsudb2:~ # service apache2 start
Starting httpd2 (prefork)                   done
```

Apache services on Red Hat Enterprise Linux

The commands on RHEL are identical to SLES, but the name of the service is httpd rather than apache2, as shown in the following commands:

chkconfig httpd on

service httpd start

Verifying web server is running

With the web service started, a web browser should be used to verify that the web server is actually working as expected, and as shown in Figure 7-1 on page 130. Start a web browser and point it to the IP address of the Linux server, in our case, the URL was:

http://9.12.7.90

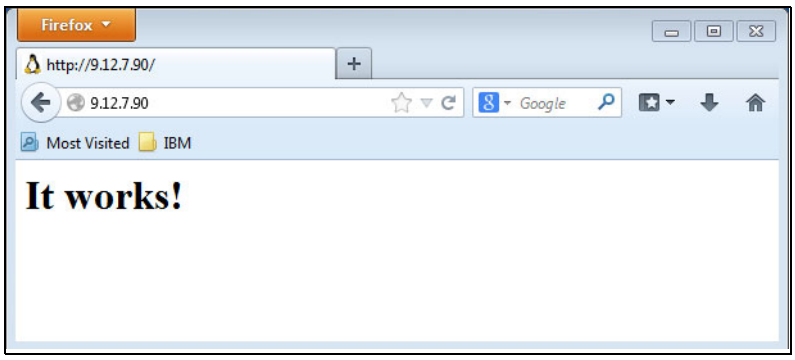


Figure 7-1 Successful test of Apache installation

Verifying PHP is working

Before a test can be conceived and executed for PHP, the location of the DocumentRoot directory of the Apache server must be determined.

In SLES, the default location is /srv/www/htdocs. However, a non-default location might have been configured. The document root directory can be determined by running the commands shown in Example 7-4.

Example 7-4 Finding the DocumentRoot on SLES

```
root@lnsudb2:~ # grep 'DocumentRoot "' /etc/apache2/default-server.conf
DocumentRoot "/srv/www/htdocs"
```

Under RHEL, the default location is /var/www/html, but the definitive value is revealed by following the example in Example 7-5.

Example 7-5 Finding the DocumentRoot on RHEL

```
[root@zs4p01-r1 ~] grep 'DocumentRoot "' /etc/httpd/conf/httpd.conf
DocumentRoot "/var/www/html"
```

After confirming the Document Root of the Apache server, a one-line PHP script is created that will print the standard PHP installation information. Using vi or some other appropriate text editor, create a script file called `phpinfo.php`, as shown in Example 7-6, and place the script file in the appropriate DocumentRoot directory.

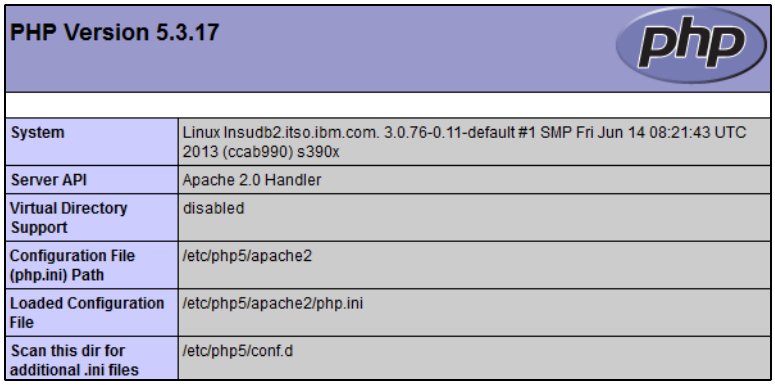
Example 7-6 Simple PHP script that displays functional characteristics

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

With the PHP script file in the DocumentRoot directory, the PHP script can be run using a web browser. Connect to your web server, using the following URL as an example:

http://9.12.7.90/phpinfo.php

Figure 7-2 on page 131 shows the expected PHP information that is generated in the browser by the PHP script running on SLES 11 SP3.



PHP Version 5.3.17

System	Linux lnsudb2.itso.ibm.com. 3.0.76-0.11-default #1 SMP Fri Jun 14 08:21:43 UTC 2013 (ccab990) s390x
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/conf.d

Figure 7-2 PHP configuration information generated by phpinfo.php

Start MySQL services on SLES

Set the MySQL database service to automatically start each time the server is booted, then manually start the service:

```
chkconfig mysql on

service mysql start
```

Start MySQL services on Red Hat Enterprise Linux

As before with Apache, the MySQL database service is named slightly differently in RHEL than it is in SLES. To set MySQL to start each time the server is booted, and to manually start the service, issue the following commands:

```
chkconfig mysqld on

service mysqld start
```

Note: SLES 11 SP3 and RHEL 6.4 both use standard SysVInit commands. SLES 12 and RHEL 7 are both slated to replace SysVInit with systemd for the management of services.

Verifying MySQL is working

MySQL must be configured and working properly before MediaWiki can even be installed. There are two configuration steps to complete for MySQL:

1. Copy a sample configuration file to MySQL's production configuration, `/etc/my.cnf`. Then, apply the appropriate ownership and access. (Reading through the configuration file to understand its contents is a wise thing to do.) Example 7-7 shows sample commands.

   Later, when migrating from the x86 server, you will likely copy the my.cnf file from the x86 server to Linux on System z. For now, the example my.cnf configuration file is sufficient in order to test the functionality of the system before migrating.

   Example 7-7 Configure MySQL configuration file

   ```
   cp /usr/share/mysql/my-medium.cnf /etc/my.cnf
   chown root:root /etc/my.cnf
   chmod 640 /etc/my.cnf
   ```

Note: The default permissions of /etc/my.cnf are 644, allowing anyone to read the MySQL configuration settings. Best practices in security suggest that system services should not provide any unnecessary information to unprivileged users. Setting the permissions to 640 prevents unprivileged users from discovering information about the configuration of the MySQL server.

2. Set a temporary password for the database administrative user. Remember this password because it is required during a few additional steps of the process before migrating the MediaWiki application from the x86 server. (This may or may not be the same password of the MySQL database that will later be migrated.) Use the command shown in Example 7-8 to set the password.

Example 7-8 Set administrative password for the MySQL service

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

With the admin password set for the root user, all future interactions with the MySQL database will require providing a password. General administrative functions will require the root password, whereas commands involving MediaWiki will use a different password.

Note: Quotation marks in Linux can be a bit tricky. When setting the root password, keep in mind that the quotation marks are not strictly necessary. If the password will contain special characters like a space, then the quotation marks are necessary. Do not use quotations marks unless you are certain that they are necessary. Copying a string from somewhere and pasting the string as the password can give unexpected results, and may make reproducing the password later an inconvenient mystery.

Test the MySQL capabilities by running this sample command:

```
mysql -u root -p  -e "show tables" mysql
```

The preceding command will prompt you for the root password that you set in the previous steps with the **mysqladmin** command. The sample output displayed in Example 7-9 shows the list of tables contained in the mysql database, suggesting that you have properly set the password.

Example 7-9 Output from the “show tables” mysql command after providing password

```
lnsudb2:~ # mysql -u root -p  -e "show tables" mysql
Enter password:
+---------------------------+
| Tables_in_mysql           |
+---------------------------+
| columns_priv              |
| db                        |
| func                      |
| help_category             |
| help_keyword              |
| help_relation             |
| help_topic                |
| host                      |
| proc                      |
| procs_priv                |
| tables_priv               |
| time_zone                 |
```

```
| time_zone_leap_second     |
| time_zone_name            |
| time_zone_transition      |
| time_zone_transition_type |
| user                      |
+---------------------------+
```

With the MySQL administrative password properly set, it is now possible to proceed to installing the MediaWiki software. If perchance the MySQL administrative password has been set up incorrectly, an error message similar to Example 7-10 is displayed.

Example 7-10 Bad password supplied to MySQL

```
lnsudb2:~ # mysql -u root -p  -e "show tables" mysql
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password:
ES)
```

To correct this problem, run the **mysqladmin** command again as shown in Example 7-8 on page 132, taking extra care to set the password to a value that you will remember. If the original password cannot be remembered or is otherwise lost, it will be necessary to reinstall MySQL.

Note: Another migration option for MySQL is to run **mysqldump** on the x86 server, transfer the resulting MySQL dump files to the Linux on System z server, and restore the database with **mysqlimport**. This is generally a recommended and widely used practice, but your environment may dictate a different practice. Your proper pre-migration analysis will help you to understand which approach is best for your circumstances.

With the preliminary Apache, MySQL, and PHP configurations functioning properly on the new Linux on System z server, the iSCSI disks can now be migrated from the x86 server.

7.2.4 Migrating iSCSI disks containing MySQL and MediaWiki

One of the particularly useful aspects of using external storage is that disks can effectively be unplugged from one server and plugged into another, fast-tracking the migration process. This is not an appropriate approach for all cases, but it is very exciting when circumstances exist that allow this approach. Such is the case for this MediaWiki migration example.

Recall from the original explanation of the scenario that the MySQL database and the Apache DocumentRoot are each self-contained on their own iSCSI disk partitions. The file systems on those partitions will be unmounted from the zs4p01-s1 host (an x86 system), then mounted on the lnsudb2 host running on System z.

Note: When dealing with remote disk storage, it is important to mount and manage the remote LUNs by referring to them by-path rather than any other method. udev will not ensure that the same name or ID will be used persistently when the host is rebooted. The only persistent identification is by-path.

Prepare Linux on System z for iSCSI

Before making any changes on the x86 host that is currently using the iSCSI LUNs, the Linux guest running on System z should have the minimum iSCSI software already setup:

1. Connect to the Linux on System z guest called *lnsudb2* using Secure Shell (SSH).

2. Ensure that the iSCSI initiator software is installed on Linux for System z guest lnsudb2:
   - For SUSE, use **zypper install open-iscsi**
   - For RHEL, use **yum install iscsi-initiator-utils**
3. Stop Apache and MySQL services running on lnsudb2. This guest has been running Apache and MySQL for the earlier tests. Stopping these services now is important for the migration.
   - For SUSE, use **service apache2 stop**

     service mysql stop
   - For RHEL, use **service httpd stop**

     service mysqld stop
4. Move the content of the /srv/www directory and the /var/lib/mysql directory out of the way so that the file systems of the iSCSI remote LUNs can be mounted in their places. But keep their content available as a backup. (Remember that the default DocumentRoot for RHEL is /var/lib/html/. Use the proper directory for your circumstances.)

   mv /srv/www /srv/www.orig

   mv /var/lib/mysql /var/lib/mysql.orig

   mkdir -p /srv/www /var/lib/mysql

Prepare x86 system for migration

1. Connect to the zs4p01-s1 (x86) host using SSH.
2. Display the mount table of zs4p01-s1, taking note of which file system contains the MySQL partition and the Apache www partition:

   mount

   In this example, the /var/lib/mysql directory is mounted via /dev/sdd1 and the /srv/www directory is mounted via /dev/sdc1. Example 7-11 shows a similar method of displaying the mounted file systems by using **df -h**.

Example 7-11 Mounted disk partitions on zs4p01-s1

```
zs4p01-s1:/var/lib # df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda2       491G   29G  462G   6% /
udev            2.0G  116K  2.0G   1% /dev
tmpfs           2.0G   72K  2.0G   1% /dev/shm
/dev/loop0      3.2G  3.2G     0 100% /srv/ISO
/dev/sdc1      1018M  199M  820M  20% /srv/www
/dev/sdd1       4.0G   33M  3.9G   1% /var/lib/mysql
```

3. Take note of which remote LUNs are currently being mounted on host zs4p01-s1:

   ls -l /dev/disk/by-path/

   Look for symlinks in the directory that has the following format:

   <IPADDRESS>:3260-iscsi-iqn.<iSCSI_Target_Identifier>:<iSCSI_unique_LUN>

   Note which remote LUN will be moved, and what symlink it is linked to. In this example, the remote by-path partition for the Apache www disk is identified as the following file system object:

   ip-9.12.7.97:3260-iscsi-iqn.2014-04.ibm.itso:LUN2-lun-2-part1

   where:

   - The IP address of the remote iSCSI target = 9.12.7.97

- The iSCSI Target Identifier = iscsi-iqn.2014-04.ibm.itso
- The unique iSCSI LUN for the Apache www partition = LUN2

Likewise, for the MySQL data directory, the IP address of the remote iSCSI target and the iSCSI Target Identifier are the same as those of the Apache www disk, with the only difference being that the unique iSCSI LUN for the MySQL data partition is equal to LUN3.

Example 7-12 shows the by-path allocations for this example, with LUN2 being referenced as sdb1 and LUN3 being referenced as sdc1.

Example 7-12 Output showing the by-path assignments of remote iSCSI disks

```
zs4p01-s1:~ l /dv/disk/by-path/
total 0
lrwxrwxrwx 1 root root 9 May 8 09:14 ip-9.12.7.97:3260-iscsi-iqn.2014-04.ibm.itso:LUN2-lun-2 ->
../../sdc
lrwxrwxrwx 1 root root 10 May 8 09:14
ip-9.12.7.97:3260-iscsi-iqn.2014-04.ibm.itso:LUN2-lun-2-part1 ->../../sdc1
rwxrwxrwx 1 root root 9 May 8 09:14 ip-9.12.7.97:3260-iscsi-iqn.2014-04.ibm.itso:LUN3-lun-3 ->
../../sdd
lrwxrwxrwx 1 root root 10 May 8 09:14
ip-9.12.7.97:3260-iscsi-iqn.2014-04.ibm.itso:LUN3-lun-3-part1 ->../../sdd1
```

Take note that LUN2 is a symbolic link to /dev/sdc1 and that LUN3 is a symbolic link to /dev/sdd1. These are the partitions from the LUNs that were identified in the previous step, and hence are the correct file systems that should be unmounted.

Note: Pay very close attention to these details. Selecting the wrong disk may result in unmounting the wrong file system, which can have disastrous consequences.

4. Stop Apache and MySQL running on zs4p01-s1 (x86) host:
   - For SUSE, use **`service apache2 stop`**

     `service mysql stop`
   - For RHEL, use **`service httpd stop`**

     `service mysqld stop`
5. Unmount the disks that contain the file systems that will be moved to lnsudb2:

   `umount /srv/www`

   `umount /var/lib/mysql`
6. For completeness, log out of the two LUNs of the iSCSI target connected from zs4p01-s1:

   `iscsiadm --mode=node --portal=9.2.7.97 --targetname=iqn.2014-04.ibm.itso:LUN2 --logout`

   `iscsiadm --mode=node --portal=9.2.7.97 --targetname=iqn.2014-04.ibm.itso:LUN3 --logout`
7. Remove the records referring to the /srv/www and /var/lib/mysql from the zs4p01-s1 host's /etc/fstab. (You may choose to retain the information before removing it; it is likely that you will use the exact same information on the new host, lnsudb2.)

Move iSCSI disks to Linux on System z

1. Use an SSH to connect again to the console of guest LNSUDB2. (No more work will be done using the console on the x86 host zs4p01-s1.)
2. Discover the remote disk services that are running on the remote iSCSI target:

   iscsiadm --mode=discovery --type=sendtargets --portal=9.12.7.97

   Example 7-13 shows three of the LUNs that are available from the iSCSI target. Our example uses only two of the LUNs.

Example 7-13 The LUNs discovered from the iSCSI target

```
lnsudb2:~ # iscsiadm --mode discovery --type sendtargets --portal 9.12.7.97
9.12.7.97:3260,1 iqn.2014-04.ibm.itso:LUN2
9.12.7.97:3260,1 iqn.2014-04.ibm.itso:LUN1
9.12.7.97:3260,1 iqn.2014-04.ibm.itso:LUN3
```

   Note that the output seen here is quite similar to the information seen on zx4p01-s1, representing the iSCSI data that you gathered in the preceding steps, such as the IP address of the iSCSI target, the iSCSI Target Identifier, and the LUNs that the iSCSI Target is exporting. In this example, the correct iSCSI devices that will be mounted on lnsudb2 are LUN2 (which contains the Apache www file system) and LUN3 (containing the MySQL database files). Although LUN1 is also listed, LUN1 is not used in this exercise:

   9.12.7.97:3260,1 iqn.2014-04.ibm.itso:LUN2

   9.12.7.97:3260,1 iqn.2014-04.ibm.itso:LUN3

3. Log in to the remote iSCSI disk, specifying the wanted LUN (LUN2) for the Apache www disk:

   iscsiadm --mode=node --portal=9.12.7.97 --targetname=iqn.2014-04.ibm.itso:LUN2 --login

   The example output shown in Example 7-14 represents a successful login.

Example 7-14 Successful login to iSCSI target's LUN2

```
lnsudb2:~ # iscsiadm --mode=node --portal=9.12.7.97 --targetname=iqn.2014-04.ibm
.itso:LUN2 --login
Logging in to [iface: default, target: iqn.2014-04.ibm.itso:LUN2, portal: 9.12.7
.97,3260] (multiple)
Login to [iface: default, target: iqn.2014-04.ibm.itso:LUN2, portal: 9.12.7.97,3
260] successful.
```

   In your environment, it is highly likely that the iSCSI target requires more sophisticated authentication for login. In this simplistic example, the iSCSI target requires no credentials of any kind.

4. Repeat the login, this time specifying LUN3, which contains the MySQL data partition:

   iscsiadm --mode=node --portal=9.12.7.97 --targetname=iqn.2014-04.ibm.itso:LUN3 --login

Note: The **--login** subcommand command on SLES will cause the iSCSI initiator to connect *only* to the specified LUN, and consequently only the specified LUN will be viewable. However, with RHEL, the **--login** subcommand to the iSCSI target will allow you to see and manipulate *all* the LUNs that are available on the iSCSI target that are authorized by the login. This behavior on RHEL has one helpful consequence, but also a few challenges. The helpful consequence is that only one login step is needed. One particular challenge is that once the login is accomplished, all the iSCSI LUNs are given /dev/sdX assignments, whether they're all wanted.

5. See that the remote LUNs are now connected to host lnsudb1:

   ls -l /dev/disk/by-path/

   Example 7-15 shows the partitions sda1 being mapped to iSCSI LUN2 and sdb1 being mapped to LUN3.

Example 7-15 iSCSI LUNs mapped to disk devices after login

```
lnsudb2:- # l /dev/disk/by-path/
total 0
drwxr-xr-x 2 root root 200 May 6 09:16 ./
drwxr-xr-x 5 root root 100 Apr 18 11:41 ../
lrwxrwxrwx 1 root root 11 Apr 18 11:41 ccw-0.0.0201 ->../../dasda
lrwxrwxrwx 1 root root 12 Apr 18 11:41 ccw-0.0.0201-partl -> ../../dasda1
lrwxrwxrwx 1 root root 11 Apr 18 11:41 ccw-0.0.0202 -> ../../dasdb
lrwxrwxrwx 1 root root 12 Apr 1$ 11:41 ccw-0.0.0202-part1 -> ../../dasdb1
lrwxrwxrwx 1 root root 9 May 6 09:12 ip-9.12.797:3260-iscsi-iqn.2014-04.1bm.
itso:LUN2-lun-2-> ../../sda
lrwxrwxrwx 1 root root 10 May 6 09:12 ip-9.12.7.97:3260-iscsi-iqn.2014-04.1bm.
itso:LUN2-lun-2-> part1 ../../sda1
lrwxrwxrwx 1 root root 9 May 6 09:16 ip-9.12.7.97:3260-iscsi-iqn.2014-04.1bm.
itso:LUN3-lun-3 -> ../../sdb
lrwxrwxrwx 1 root root 10 9 May 6 09:16 ip-9.12.7.97:3260-iscsi-iqn.2014-04.1bm.
itso:LUN3-lun-3-part1 -> ../../sdb1
```

6. Add entries to /etc/fstab for the remote iSCSI disks to be routinely mounted in their appropriate places. Again, be sure to use the by-path designation. Example 7-16 shows a snippet from /etc/fstab, with the two new file system entries.

Example 7-16 New /etc/fstab containing the new MySQL and www disks on lnsudb2

```
# external iSCSI disk LUN2 for www filesystem
/dev/disk/by-path/ip-9.12.7.97:3260-iscsi-iqn.2014-04.ibm.itso:LUN2-lun-2-part1
/srv/www            xfs        nofail                1 2
# external iSCSI disk LUN3 for MySQL filesystem
/dev/disk/by-path/ip-9.12.7.97:3260-iscsi-iqn.2014-04.ibm.itso:LUN3-lun-3-part1
/var/lib/mysql      xfs        nofail                1 2
```

7. Mount the remote iSCSI disks in their proper places using the following commands:

   mount /var/lib/mysql

   mount /srv/www

   The file systems from the remote iSCSI disks are now mounted and available on the new LNSUDB2 host.

Complete migration of services

The remaining tasks are critical and the most specialized. It is necessary to copy the Apache and MySQL configuration files from the x86 host to the Linux on System z guest, then adapt them appropriately. Notice that the configuration of MediaWiki requires nothing special, since it was all moved when the www partition was moved. In some cases, the configurations may be simple enough to warrant a basic copy of the files. Other circumstances may be more complex, which will require rewriting the configuration files. But with the data having been migrated so effortlessly via iSCSI, it is easy to tolerate a small amount of editing of the configuration files.

For this example, a simple copy of the configuration files is all that is necessary. To do this:

1. Start an SSH console on LNSUDB2 (Linux on System z).
2. Synchronize the Apache configuration files to LNSUDB2 from zs4p01-s1. Use the method that makes the most sense for your environment, following this example:

   ```
   rsync -qa zs4p01-s1:/etc/apache2/* /etc/apache2/
   rsync -qa zs4p01-s1:/etc/my.cnf /etc/
   ```

3. Start the Apache and MySQL services on LNSUDB2:
   - For SUSE, use **`service apache2 start`**

     `service mysql start`
   - For RHEL, use **`service httpd start`**

     `service mysqld start`
4. Ensure that Apache and MySQL will start each time that the server is booted:
   - For SUSE, use **`chkconfig apache2 on`**

     `chkconfig mysql on`
   - For RHEL, use **`chkconfig httpd on`**

     `chkconfig mysqld on`

Having successfully migrated Apache, MySQL, and their respective data, including the MediaWiki data, the MediaWiki application should now be functional. Opening the MediaWiki URL using a browser, the web page will look like the picture in Figure 7-3, representing a successful installation of MediaWiki.

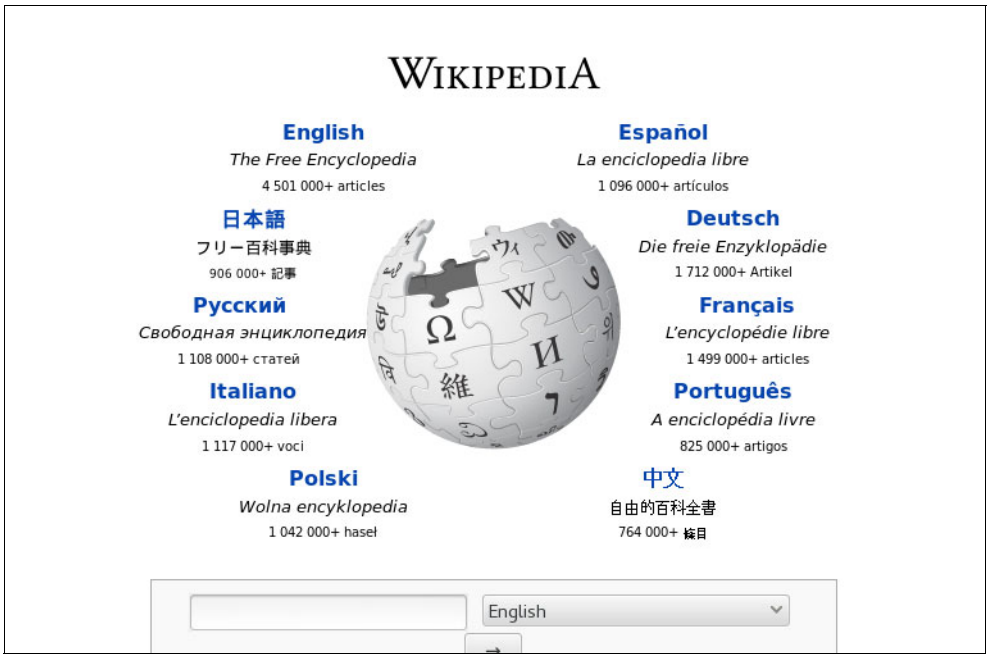


Figure 7-3 A successful migration of MediaWiki

The page shown in this figure originally comes from www.wikipedia.org. A small portion of a backup file for the Wikipedia site was used for this example. The backup file was retrieved from http://dumps.wikimedia.org and restored in our test environment in our labs for demonstration purposes only. Content and copyrights of the page shown are property of

Wikimedia Foundation, Inc. and are used in accordance with the Terms of Use supplied by www.wikipedia.org.

7.3 Deploying OpenLDAP

Enterprises of all sizes need to manage the users of their computing resources. And with the user management comes the various characteristics of the user, such as user ID, authentication, file system rights, printer rights, and more, all needing to be managed. One of the most common products used for managing this data is the Lightweight Directory Access Protocol, commonly known as *LDAP*.

LDAP is widely used throughout the industry for directory services, as an open standard running over an IP network. Although there are several commercial LDAP products available, OpenLDAP is the implementation that is most commonly used in Linux. OpenLDAP is a fully featured suite of tools and applications. It is readily available as a workload on System z from both RHEL and SUSE. LDAP is a perfect workload for Linux on System z, due to the centrality of System z among many other systems and services, its fast I/O, and its low CPU and memory overhead. And of course OpenLDAP is open source. Migrating OpenLDAP to Linux on System z is straight forward.

In section 7.2, “Deploying MediaWiki and MySQL” on page 126, we installed a LAMP server with MediaWiki, and iSCSI external storage was used to facilitate the migration. In this example, the LDAP database on an x86 server will be exported, the database will be transferred to a Linux guest running on System z, and the data will be imported into the LDAP service.

7.3.1 Analysis and planning

As with the MediaWiki example described in section 7.2, “Deploying MediaWiki and MySQL” on page 126, it is important that you follow Chapter 5, “Migration planning” on page 49, and Chapter 6, “Migration analysis” on page 57, knowing that appropriate planning and analysis should be performed before any migration activity. The checklists have been created to help identify the many considerations that should be made which will help prevent problems migrating.

Again for this example scenario, the z/VM guest has already been set up and a minimal Linux operating system has been installed. The Linux guest is called LNSUDB2 and is running SLES11 SP3, with one virtual CPU and 1 GB of virtual memory. An OpenLDAP server typically does not require a large amount of CPU or RAM running on Linux on System z. It is presumed that an adequate RPM repository installation source is already set up and available for the installation of the application software that will be used.

The x86 server is called zs4p01-r1 and is running RHEL 6.4. For this example, this is the current OpenLDAP server providing directory services for the hypothetical organization. This server has a very rudimentary (small) LDAP directory already configured.

Although there is much to consider when setting up an enterprise directory service, a very simple OpenLDAP scenario will be covered here. More extensive documentation can be found at the following site:

`http://www.openldap.org`

This example is a stand-alone server with a local, non-replicated directory service. Nevertheless, migrating an existing OpenLDAP installation on x86 to Linux on System z should be very straight forward.

7.3.2 Installing LDAP software

The OpenLDAP server is technically a very simple application, consisting of a single package. Consequently installing the software is relatively easy. The software must first be installed on the Linux on System z guest before other migration steps should be attempted. If you are going to install OpenLDAP on SLES, run the following command to install the package on SLES:

```
zypper install openldap2
```

To install OpenLDAP on RHEL, run the following command:

```
yum install openldap-servers
```

7.3.3 Configuring the OpenLDAP service

The principal player in the OpenLDAP server suite of applications is the Standalone LDAP Daemon, known as **slapd**. This example configures the **slapd** service to operate as a stand-alone, local, non-replicated directory. The package, in RPM format, contains functional sample configuration files, which will serve as the basis of the example service that is configured here.

The initial configuration of OpenLDAP on SLES running on System z will be accomplished using YaST, while a parallel example on Red Hat will be done by manually modifying configuration files and running commands.

Before migrating the LDAP database to Linux on System z, it is necessary to establish a basic configuration of OpenLDAP. Using different terminology, the OpenLDAP configuration must be started, also known as *bootstrapped*.

Note: OpenLDAP maintains its configuration using one of two different configuration methods. The “old” method involves maintaining the primary configuration in /etc/openldap/slapd.conf. This method is very simple, but does not have as many features. The “new” way (called the cn=config format) uses several configuration files below /etc/openldap/slapd.d/. The default behavior with OpenLDAP 2.4 is to use the cn=config method.

Configuring OpenLDAP on SLES using YaST

All of the activities to create a basic configuration of OpenLDAP are facilitated by the LDAP server YaST module. By following a few simple screens in YaST, the LDAP services can be configured and running in short order. To do this, perform the following steps:

1. From a command prompt, start YaST, calling specifically the ldap-server module:

   ```
   yast2 ldap-server
   ```

   Figure 7-4 on page 141 shows the first panel.

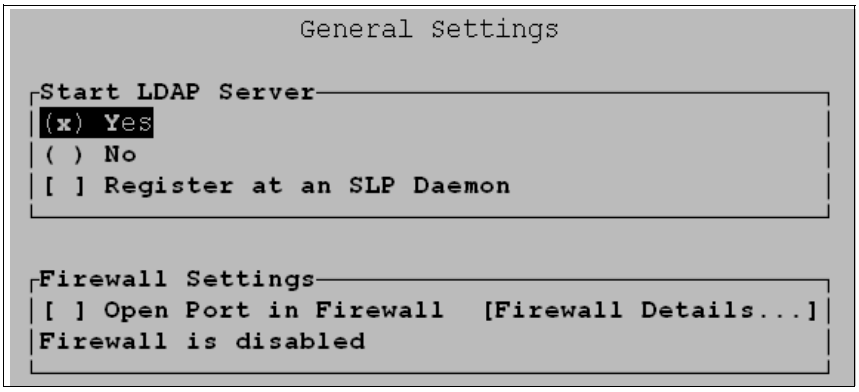


Figure 7-4 yast2 ldap-server module

Select “Yes” to start the LDAP server automatically, and be certain to open a port in the firewall. In our example, since the firewall is disabled, we did not have the option to select “Open Port in Firewall”.

Press F-10 to go to the next panel.

2. Select “Stand-alone server” as the server type, as shown in Figure 7-5.

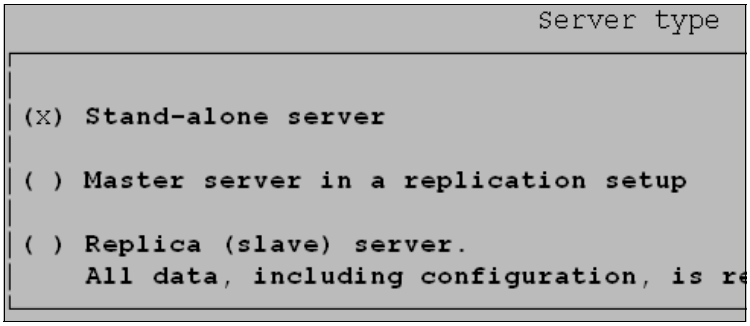


Figure 7-5 Stand-alone server type of the LDAP server

Press F-10 to go the next panel.

3. Select the proper security settings for OpenLDAP, as shown in Figure 7-6.

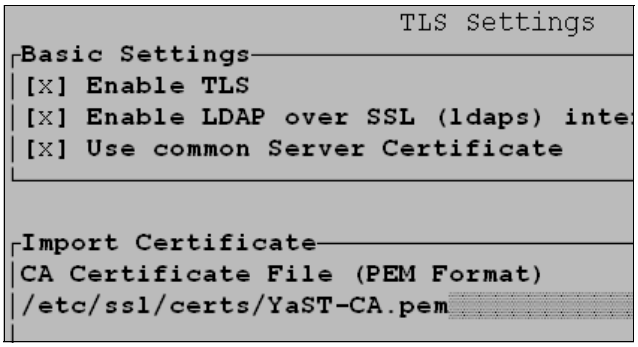


Figure 7-6 TLS and SSL certificate settings for OpenLDAP

Using a proper SSL certificate is highly recommended, but not necessary for this demonstration. Here, we use the self-signed system certificates generated when SLES was installed. More importantly, note that using SSL for the LDAP protocol (LDAPS) is

essential. Without LDAPS, passwords and other sensitive data will be exchanged with the LDAP server in plaintext. No one needs or even should want that.

Press F-10 to go to the next panel.

4. Figure 7-7 illustrates the “Basic Database Settings” panel, which includes fields for setting an administrative password for LDAP.

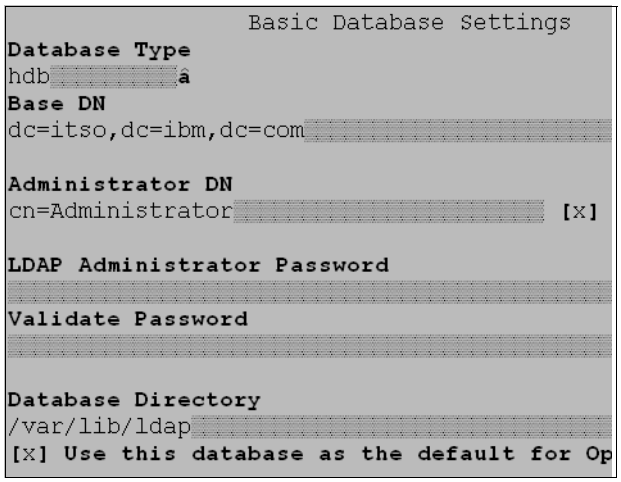


Figure 7-7 Basic database settings for OpenLDAP configuration

In a production environment, proper distinguished name (DN) data should be entered, but for this demonstration it is adequate to use the sample values supplied by YaST. What is most important to note here is the need to provide an administrator password. Best practices dictate that this should not be the same as the system’s root password, and all other best practices for creating an administrative password should likewise be employed. For this demonstration, the password “ldapadmin” will be used.

Press F-10 to go to the next panel.

5. The configuration summary is displayed, as shown in Figure 7-8.

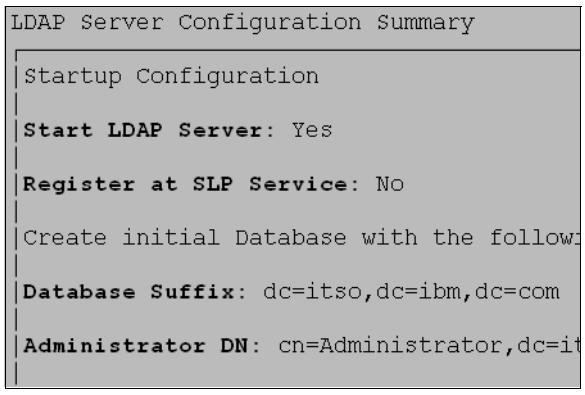


Figure 7-8 OpenLDAP configuration summary

With all the configuration information sufficiently gathered, the YaST configuration steps can be completed, by pressing F-10 to finish. The configuration files are written, and the slapd daemon is started. The running daemon process can be seen in Example 7-17 on page 143. Note that the “-F /etc/openldap/slapd.d” argument indicates that the service is configured using the cn=config feature format.

Example 7-17 slapd daemon shown running using the cn=config method

```
lnsudb2:- # ps -ef | grep slapd
ldap      17224    1    0 16:26 ?      00:00:00 /usr/lib/openldap/slapd -h ldap:
/// ldaps:/// ldapi:/// -F /etc/openldap/slapd.d -u ldap -g ldap -o slp=off
root    17251    7470    0  16:27 pts/0    00:00:00 grep slapd
```

Configuring OpenLDAP manually on Red Hat Enterprise Linux

The configuration on the Red Hat Enterprise Linux (RHEL) server is also a relatively easy task because all that is needed is a basic, bootstrappable configuration. This basic configuration by itself is not useful for running a proper directory, but it will allow the migration of the openLDAP directory from another server. OpenLDAP 2.4 on RHEL6 also uses the cn=config feature configuration format by default:

1. Ensure that the **slapd** daemon is running.

   service slapd start

2. From a command prompt on the RHEL server, edit a basic OpenLDAP configuration file, perhaps using vi as in the following example:

   vi /tmp/config.itso.ibm.com.ldif

   Put the content shown in Example 7-18 into the file:

   Example 7-18 /tmp/config.itso.ibm.com.ldif file to bootstrap the OpenLDAP database

   ```
   dn: olcDatabase=hdb,cn=config
   objectClass: olcDatabaseConfig
   objectClass: olcHdbConfig
   olcDatabase: {1}hdb
   olcSuffix: dc=itso,dc=ibm,dc=com
   olcDbDirectory: /var/lib/ldap
   olcRootDN: cn=Administrator,dc=itso,dc=ibm,dc=com
   olcRootPW: ldapadmin
   olcAccess: to attrs=userPassword by dn="cn=Administrator,dc=itso,dc=ibm,dc=com"
   write by anonymous auth by self write by * none
   olcAccess: to attrs=shadowLastChange by self write by * read
   olcAccess: to dn.base="" by * read
   olcAccess: to * by dn="cn=Administrator,dc=itso,dc=ibm,dc=com" write by * read
   ```

   Save the file, and exit the editor.

3. Bootstrap the database and import the configuration from the file created in Example 7-18 using the following command:

   ldapadd -Y EXTERNAL -H ldapi:/// -f /tmp/config.itso.ibm.com.ldif

Now the basic configuration of OpenLDAP will allow a migration of the database.

7.3.4 Export OpenLDAP data from x86 server

The LDAP directory tree running on the x86 server now needs to be exported, so that the data can be transferred to the Linux guest on System z. To do this, perform the following steps:

1. Connect to the x86 host, zs4p01-r1, using an SSH. We are doing this on an RHEL server.
2. Stop the slapd daemon so that the data can be exported from OpenLDAP:

   service slapd stop

3. Export the data from the OpenLDAP database. The tool used to accomplish this is called *slapcat*, and this is a common method of extracting whole data sets from OpenLDAP. The output is written in LDAP Data Interchange Format (LDIF), which is a standard plain text data interchange format for representing LDAP:

   ```
   slapcat -b 'dc=itso,dc=ibm,dc=com' -l /tmp/migrate.ldif
   ```

   The “-l” argument tells slapcat to export the database (in the LDIF format) to the file /tmp/migrate.ldif. The “-b” argument identifies the specific domain of data to export (known as the suffix in the OpenLDAP vernacular).

4. (Optional) Restart the slapd daemon on zs4p01-r1. Since the daemon is being migrated to another server, it may not be necessary to restart it.

   ```
   service slapd start
   ```

5. Transfer the database file to the Linux guest, LNSUDB2, running on System z. Use the transfer mechanism that is most suitable. This example uses a utility software and network protocol called **rsync**:

   ```
   rsync /tmp/migrate.ldif 9.12.7.90:/tmp/
   ```

   Note that the server with the IP address 9.12.7.90 is LNSUDB2 and is the Linux guest on System z. Provide appropriate credentials when prompted. When the transfer is complete, the process of exporting the data from this x86 server to the Linux guest running on System z has been successfully completed.

7.3.5 Import OpenLDAP data to Linux on System z

In the previous section, the OpenLDAP database export file was transferred to lnsudb2, the Linux guest running on System z. All that is required now is to import the data and start the OpenLDAP daemon:

1. Reconnect to the System z guest, lnsudb2, using SSH.
2. Ensure that slapd is not running. Importing data for migration requires that the service is not running:

   ```
   service slapd stop
   ```

3. Import the data that was copied. This process employs a tool called *slapadd*. This is a common method of importing whole data sets into OpenLDAP:

   ```
   slapadd -F /etc/openldap/slapd.d \
   -b 'dc=itso,dc=ibm,dc=com' -l  /tmp/migrate.ldif
   ```

   Because the basic configuration was established in section 7.3.3, “Configuring the OpenLDAP service” on page 140, the itso.ibm.com domain already exists in the new OpenLDAP database, making it very easy to import the data. The “-b” argument identifies the domain, and the “-l” argument indicates the LDIF file from which the database information will be imported.

   A successful import shows “100%” success, as illustrated in Example 7-19 on page 145. Any value other than 100% means that something went wrong and the import of the data was not successful.

Example 7-19 Import of OpenLDAP data is 100% successful

```
lnsudb2:- # slapdd -F /etc/openldap/slapd.d -b ‘dc=itso,dc=ibm,dc=c
om’ -l /tmp/migrate.ldif
hdb_monitor_db_open: monitoring disabled; configure monitor database to enable
_################## 100.00% eta none elapsed none fast!
Closing DB...
```

4. Once the database has been successfully imported, OpenLDAP can be started again, ready to receive queries:

   service slapd start

7.3.6 Verify OpenLDAP is working

The slapd process is running, and sample data is presumed to exist in the directory, but that does not necessarily mean that OpenLDAP is usable by any clients. It is important to test that the LDAP server responds to client requests. In this example, the user “fred” is queried:

ldapsearch -xLLL -H ldapi:/// -b "dc=itso,dc=ibm,dc=com" uid=fred sn givenName cn

Example 7-20 shows the results of the ldapsearch query.

Example 7-20 Output from ldapsearch, showing user fred exists in the directory

```
lnsudb2:- # ldapsearch -xLLL -H ldapi:/// -b "dc=itso,dc=ibm,dc=com" uid=fred sn
 givenName cn
dn: uid=fred,ou=employees,dc=itso,dc=ibm,dc=com
sn: frandsen
cn: fred
```

But in the preceding example, the OpenLDAP client and the server are both running on the same system, LNSUDB2. That is not necessarily a convincing demonstration. A better verification is whether an external client can query the OpenLDAP server over the network. Example 7-21 shows that a different client, zs4p01-s1, queries the LDAP directory running on lnsudb2 (9.12.7.90).

Example 7-21 Output from ldapsearch, querying the LDAP directory over the network

```
zs4p01-s1:~ # ldapsearch -xLLL -H ldap://9.12.7.90 \
>                        -b "dc=itso,dc=ibm,dc=com" \
>                        uid=fred sn givenName cn
dn: uid=fred,ou=employees,dc=itso,dc=ibm,dc=com
sn: frandsen
cn: fred
```

This second verification in Example 7-21 indicates a successful migration of an OpenLDAP service from Linux on x86 to Linux on System z. Not only that, but the service has been quite easily migrated from a system running RHEL to one running SLES. OpenLDAP, Linux, and System z are all very happy regardless of the distribution, and the migration of OpenLDAP is unhampered regardless of the distribution.

7.4 Deploying central log server

As you saw in section 6.6.10, “Logging and recording events” on page 103, forwarding local log records to a remote secure system is a good practice to keep your log records safe. When someone does attempt to attack one of the servers, they will probably try to clean up their tracks. By using remote centralized log servers, you can keep a safe copy even if they remove the local copies or stop the service. Also, you will be able to centralize all logs from your environment and use a real-time search and analytics tool to create business insights or a monitoring tool.

To create a centralized log server we will use the default log daemon from SUSE, **syslog-ng** (version 2.09). It can be easily installed on RHEL via Extra Packages for Enterprise Linux (EPEL6).

7.4.1 Analysis and planning

Use the logical volume manager (LVM) to create a logical volume for log files because log files tend to grow very fast, and with different hosts writing logs to a centralized log server at the same time, log files can fill your disk even faster. So, leave some space available in a volume group that can be used in case of emergency.

7.4.2 Initial configuration

The default path for the **syslog-ng** configuration file is /etc/syslog-ng/syslog-ng.conf. It is made up of key words that define the message route and global options. You can see all the available global options by issuing the command:

`man syslog-ng.conf`.

Global options

You can specify several global options in the options statement of **syslog-ng**. You can define how the host name of the client will appear in the log files, enable or disable DNS cache, the ownership of the files and some other features that you will use depending on the size or specific requirements of your environment.

Source options

The source keyword is used to add source drivers or define your own sources. For example, the syntax to enable syslog-ng and define a udp or tcp connection to listen for remote logs is the following:

```
source s_net { tcp((ip(127.0.0.1) port(40000) max-connections 5000)); udp (); };
```

This entry in the configuration files defines a source called **`s_net`** that will listen on localhost to port 40000 using TCP and listen port 514 (default for syslog-ng) using UDP on all interfaces. Also, it is limiting the maximum number of connections of the TCP port. There are more parameters that you can define. You can always check it in syslog-ng.conf manual.

Filter options

Filters let you set different destinations depending on certain key words. The syntax for the filter statement is:

```
filter <identifier> { expression; };
```

where <identifier> is the name that you give your filter and <expression> contains the function, and boolean operators (and,or,not). Example 7-22 demonstrates this.

Example 7-22 Filters examples

```
filter f_iptables   { facility(kern) and match("IN=") and match("OUT="); };
filter f_messages   { not facility(news, mail) and not filter(f_iptables); };
filter f_warn       { level(warn, err, crit) and not filter(f_iptables); };
```

Destination options

Destination options define the files to which log messages are sent. The `syslog-ng.conf` default file covers the destination for local files, but what about the incoming messages? By default, the remote log messages are written to the default local files depending on what kind of message it received, so if it was an authentication message the log would be written to the file `/var/log/auth` with the appended information defined in the global options, host name, date, and time, and so on. Example 7-23 uses files as a destination, but you can use databases or even another remote server, depending on how you configure your server.

Example 7-23 Destination example

```
#
# All messages except iptables and the facilities news and mail:
#
destination messages { file("/var/log/messages"); };
log { source(src); filter(f_messages); destination(messages); };
#
# Firewall (iptables) messages in one file:
#
destination firewall { file("/var/log/firewall"); };
log { source(src); filter(f_iptables); destination(firewall); };
#
# Warnings (except iptables) in one file:
#
destination warn { file("/var/log/warn" fsync(yes)); };
log { source(src); filter(f_warn); destination(warn); };
```

As you can see in Example 7-23, the *log* statement requires a *source*, *filter*, and *destination*.

7.4.3 Server configuration

We will not create a complex server configuration because it is out of the scope of this book. The default configuration file of syslog-ng comes with definitions that will split your files depending on the facility that will be logged. Figure 7-9 on page 148 shows a centralized syslog-ng server receiving log copies from syslog-ng clients.

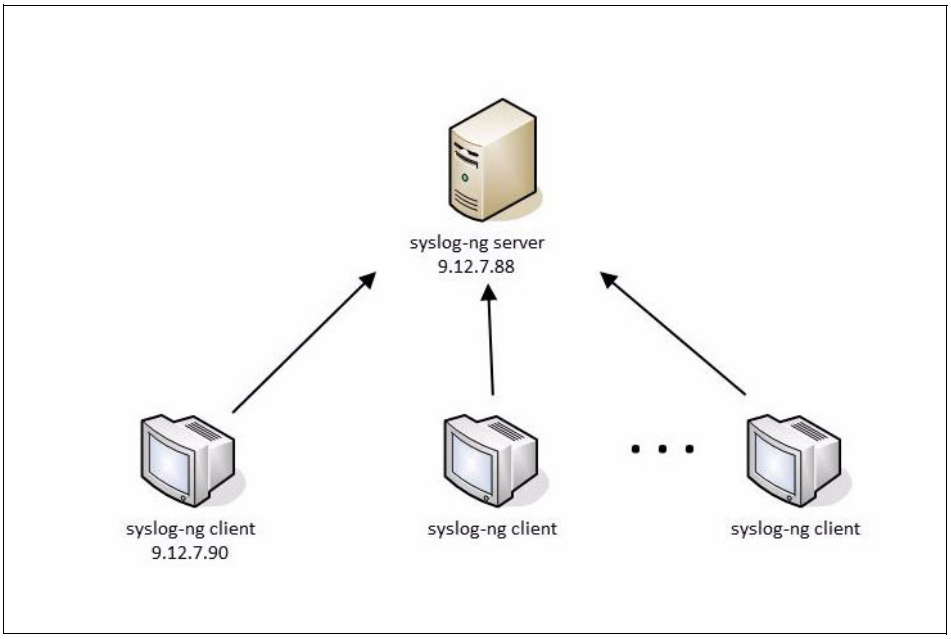


Figure 7-9 A centralized syslog-ng server and some syslog-ng clients

The global options for our test environment are shown in Example 7-24.

Example 7-24 Global options for syslog-ng server

```
options {
sync(0); # The number of lines buffered before written to file
perm(0640); # Permission value for created files.
keep_hostname(yes); # Keeps the hostname from the origin.
};
```

To enable the listener, we defined the source (src) to use the *udp* statement, as shown in Example 7-25.

Example 7-25 Source example for syslog-ng server

```
source src {
        #
        # include internal syslog-ng messages
        # note: the internal() soure is required!
        #
        internal();

        #
        # the default log socket for local logging:
        #
        unix-dgram("/dev/log");

        #
        # uncomment to process log messages from network:
        #
        udp(ip("0.0.0.0") port(514));
};
```

We restarted the syslog service to load the syslog-ng configuration file, as shown in Example 7-26.

Example 7-26 Restarting syslog-ng server to update the new configuration

```
syslog-server-1:~ # service syslog restart
Shutting down syslog services done
Starting syslog services done
```

To test the listener, you can use **lsof**, as in Example 7-27.

Example 7-27

```
syslog-server-1:~ # lsof -i UDP:514
COMMAND     PID USER   FD   TYPE  DEVICE SIZE/OFF NODE NAME
syslog-ng 60700 root    4u  IPv4 3279255      0t0  UDP *:shell
```

You can also use TCP if you would like, but to start a simple syslog-ng server, you only need the udp statement, as shown in Example 7-25 on page 148.

In the next section the syslog-ng client is set up.

7.4.4 Client configuration

To configure the syslog-ng client, a destination needs to be created to point to the log server, and a new log statement needs to be added to point the client source to the new destination. Append the lines shown in Example 7-28 to the syslog-ng.conf of the client, using your client's IP address.

Example 7-28 Append these lines to your client configuration to create a destination to the log server

```
destination logserver { udp("9.12.7.88" port(514)); };
log { source(src); destination(logserver); };
```

In our example, all logs from the client will be also sent to "9.12.7.88" (lnsuwas1) via UDP using port 514. This is a simple configuration, but you can set up filters, new destinations, and sources depending on the requirement of your environment.

Restart your syslog-ng client as shown in Example 7-29.

Example 7-29 Restarting the syslog-ng client to update the configuration

```
syslog-client-1:~ # /etc/init.d/syslog restart
Shutting down syslog services done
Starting syslog services done
```

7.4.5 Testing syslog-ng

From the log server (syslog-server-1), you can use the command **tail -f** to monitor the messages written to the `/var/log/message` file. See Example 7-30 on page 150.

Example 7-30 Monitoring the syslog-ng server

```
syslog-server-1:~ # tail -f /var/log/messages
May  2 11:45:01 syslog-server-1 syslog-ng[32617]: Configuration reload request
received, reloading configuration;
May  2 11:45:01 syslog-server-1 syslog-ng[32617]: New configuration initialized;
```

To test the client, you can use the **logger** command:

```
Logger "Testing syslog-ng"
```

Example 7-31 shows the result from the log server (syslog-server-1).

Example 7-31 Getting logs from the syslog-ng clients

```
syslog-server-1:~ # tail -f /var/log/messages
May  2 11:45:01 syslog-server-1 syslog-ng[32617]: Configuration reload request
received, reloading configuration;
May  2 11:45:01 syslog-server-1 syslog-ng[32617]: New configuration initialized;
May  2 11:50:39 syslog-client-1 root: Testing syslog-ng
```

For alternate setups, see the official syslog documentation website:

http://www.balabit.com/support/documentation

7.4.6 Migrating using syslog-ng

Syslog-ng can be used as a tool for your migration. You can set up a centralized log server to keep a copy of the log files for all the servers that you are migrating. Therefore, if a problem happens on the server and you lose access, you can easily fetch information or error messages.

If you are migrating an existing syslog-ng server/client, you need to check if syslog-ng is installed on the target server and ensure that the former configuration file is compatible to the version available on Linux on System z. To migrate the old data, you can use an LVM snapshot to transfer the logical volume to the new server. Other commands such as **tar** and **rsync** can be used to transfer the old log files. You can see a practical example of *LVM snapshot*, **tar,** and **rsync** in the IBM Redbooks publication, *Set up Linux on IBM System z for Production*, SG24-8137.

7.5 Deploying Samba

Samba is an open software suite that runs the Server Message Block (SMB) protocol over the Internet Protocol network and provides seamless file and print services to users. Although there are several similar commercial products available, Samba is the implementation that is most commonly used in Linux environments to share files and printers. It is available for Linux on System z from both Red Hat and SUSE and allows interoperability between UNIX/Linux servers and Windows/Linux based clients. Samba runs easily on Linux on System z because System z has fast I/O that provides high performance access to applications and files.

Before deploying Samba, ensure that appropriate analysis and planning has been performed before any migration activity. The checklists provided in this book have been created to help identify the many considerations that should be made which will help prevent problems during migration.

In our sample scenario, the z/VM guest has already been set up and a minimal Linux operating system has been installed. The Linux guest is named LNSUDB2, and has SLES11 SP3 installed with one virtual CPU and 1 GB of virtual memory. Like LDAP, a Samba server typically does not require a large amount of CPU or RAM to run on Linux on System z. It is presumed that an adequate RPM repository installation source is already set up and available for the installation of the application software that will be used.

More extensive documentation about Samba can be found at the following site:

http://www.samba.org

This example will be a stand-alone server with a local, non-replicated directory service. Nevertheless, migrating an existing Samba installation on x86 to Linux on System z should be straight forward.

7.5.1 Installing Samba software

Installing the software is relatively easy.

- To install Samba on SUSE Linux Enterprise Server:

  Run **zypper install samba** to install the Samba and its dependencies packages on SLES.

- To install Samba on Red Hat Enterprise Linux:

  Run **yum install samba** to install the Samba and its dependencies packages on RHEL.

7.5.2 Configuring SAMBA

In this section, we describe how to configure Samba first on SLES and then on RHEL.

Configuring file server on SAMBA on SLES using YaST

All of the activities to create a working configuration are facilitated by the Samba server YaST module. By following a few simple panels in YaST, the SAMBA services can be configured and running in short order.

To configure a Samba server, start YaST and select **Network Services** → **Samba Server** and complete the fields that are shown in Example 7-32 with your network information.

Example 7-32 Initial configuration of Samba on YaST

```
Workgroup name or Domain Name: Select your existing name from the Workgroup or
Domain
Samba Server Type (PDC, BDC or Stand Alone): Specify whether your server should
act
Start service : To start after the server reboot, choose during the boot
Firewall Settings : If you are running the firewall servers inside this server,
mark the option Open Port in Firewall
Samba root Password: choose a password for your Samba service
```

After the initial installation step is completed, confirm by selecting **OK**. You will be able to change the settings later in the Samba configuration dialog on YaST, as shown in Figure 7-10 on page 152.

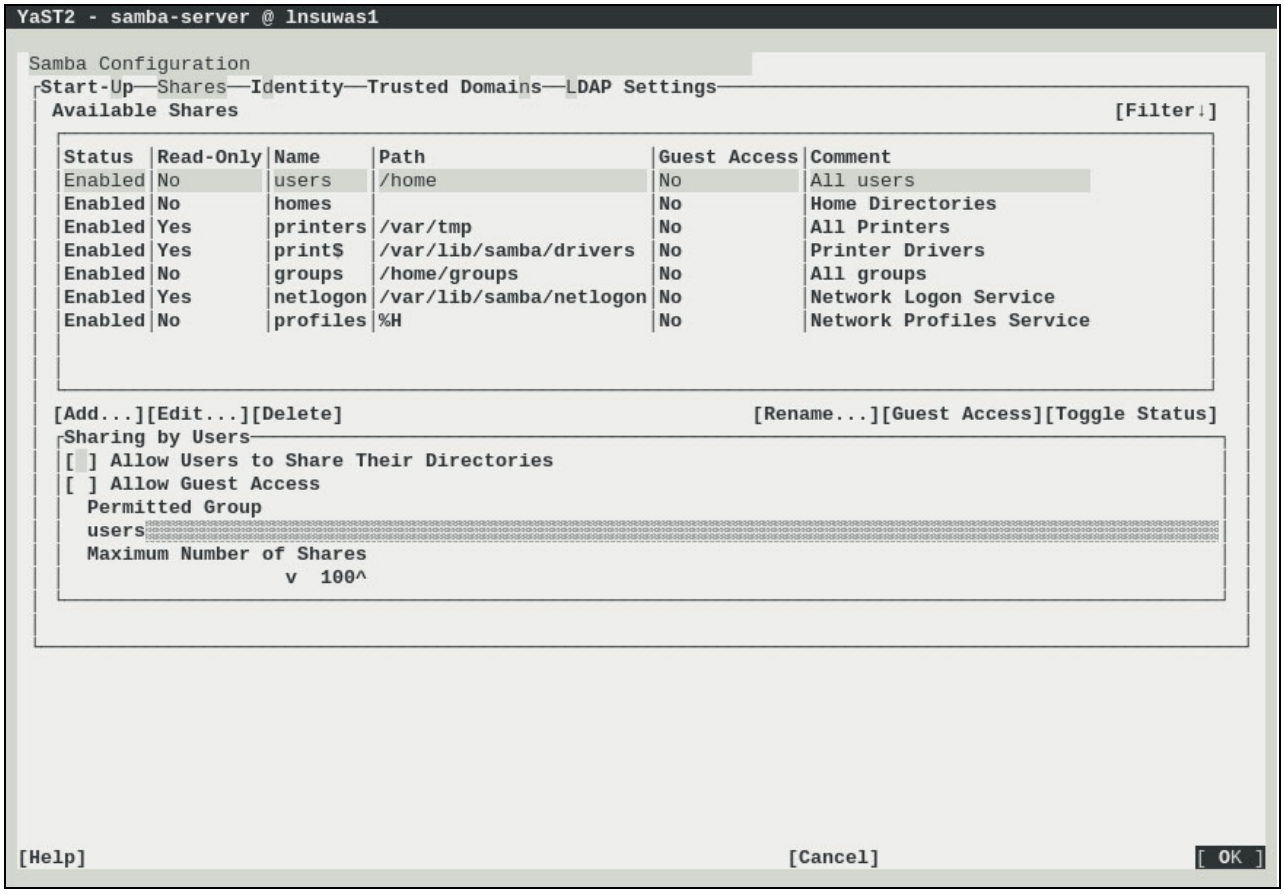


Figure 7-10 Samba Server Configuration tab

Samba shares

In order to share resources such as folders or printers on a Samba server, you must first identify these “shares”. You can configure your shares on YaST in the Samba Server. In the Samba configuration tab, shown in Figure 7-10, select the **Shares** tab and then select **Add**. Provide the information shown in Example 7-33 and shown on the YaST2 panel in Figure 7-11 on page 153.

Example 7-33 Creating new share on Samba using Yast

```
Share Name : Fill out the share name
Share Description : brief description of the share
Share Type : select if you are sharing a folder or a printer
Share Path : browser the folder name. Make sure that the folder is set up with the
correct permission on the Linux filesystem
Select If you need Read only access and Inherit the config to the subdirectories
```

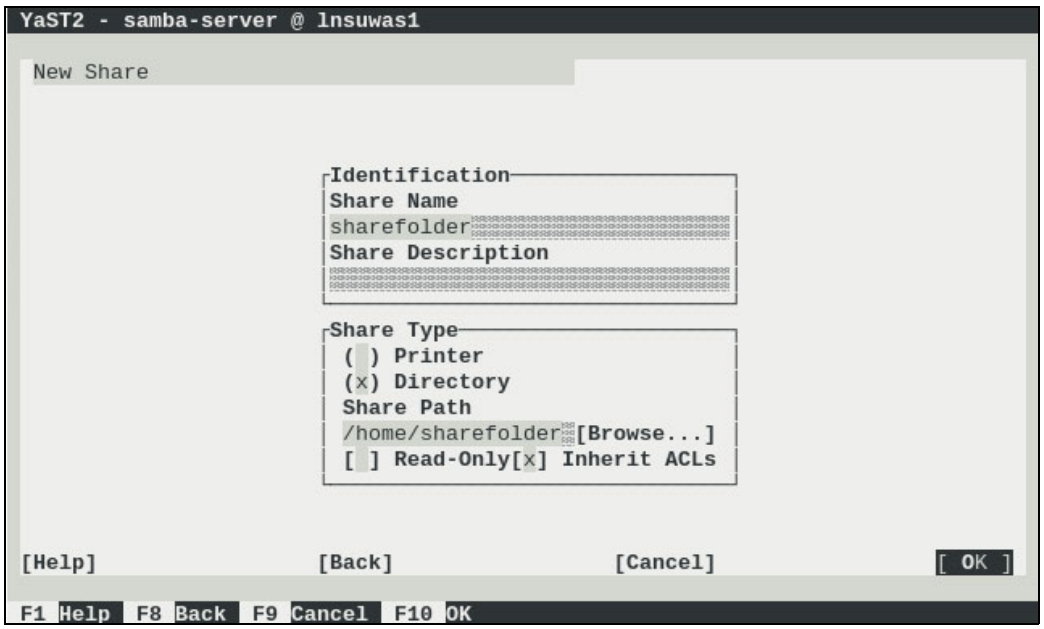


Figure 7-11 Sample of creating new share on Samba using Yast

After completing this task, you should be able to map to the server from your client machine.

Note: If you intend to use basic Linux authentication, that is, using the *passwd* file, you must change the Samba user password using the command **`smbpasswd -a <userid>`**.

LDAP settings for Samba

Many companies use LDAP to provide a single signon, where a password for one user is shared between services. The activities to create a working configuration are facilitated by the OpenLDAP server YaST module. Although the LDAP configuration on SAMBA is an important feature, the configuration is out of the scope of this book.

For more information on how to set up LDAP on Samba, visit the Samba official website:

`http://www.samba.org`

Configuration files

You can manually set up configuration files for Samba. The main configuration file on SLES is stored in `/etc/samba/smb.conf`. You will find two sections:

- `[ global ]` to general settings
- `[ share ]` to specify specific settings about sharing files and printers

For more information about Samba configuration on SLES, see the SUSE Linux Enterprise Server 11 Administration Guide, Samba section:

`bit.ly/Zexlvc`

Note: RHEL 6 uses the same structure as SLES 11 for the Samba main configuration files.

Starting and stopping the Samba service

The *smb* service controls the server daemon and can be stopped and started by the commands as shown in Example 7-34.

Example 7-34 Stopping and starting the Samba service

```
To stop and start the service both SLES and RHEL:
Stop the service: /etc/init.d/smb stop
Start the service: /etc/init.d/smb start
```

8

Hands-on migration

Now that migration planning and analysis have been completed, first test your migration plan by performing a hands-on migration into a test environment. Here, you will want to load test the environment and ensure the performance meets the business expectation and needs. After that, you will migrate from test to production.

There are many ways of migrating your data from your source x86 server to your new Linux on System z server, and many ways of configuring your new Linux on System z environment.

In this chapter, we describe a hands-on migration scenario performed here in our lab where we migrated a WebSphere Application Server, DB2, and a simple load testing Java application called *Trade 6* from our x86 SLES 11 SP3 environment to our Linux on System z environment.

8.1 Setting up the system

In this section, we describe the tasks used to create our virtual guests on System z. We first completed our planning checklists to determine what resources we were using in the x86 environment. Next, we determined what we were going to use in the System z environment, keeping performance in mind. We then used IBM Wave for z/VM to quickly and efficiently create the Linux on System z guests in our LPAR. Finally, we performed the actual migration tasks.

8.1.1 Software products and tools checklist

In Table 8-1, we listed the software products.

Table 8-1 Software products and tools checklist for the x86 environment

SOFTWARE PRODUCTS AND TOOLS CHECKLIST - x86 environment				
Name	**Version**	**Vendor/Source website**	**License type**	**Linux on System z**
DB2	10.5.0.4	IBM www.ibm.com		
WebSphere Application Server				

8.1.2 Hardware checklist

In this section, we listed the hardware resources that we anticipate we will need, either physical or virtual. In the checklist used in this project, the source environment's hardware resources were examined and we cross referenced that with what we would use on Linux for System z.

HARDWARE PLANNING CHECKLIST			
SERVERNAME :			
RESOURCE	**SOURCE**	**DESTINATION**	**OBSERVATION**
Number of CPU	4	2	Real to Virtual
System memory (in GB)	8	8	
OS SWAP Memory (in GB)	4	4	
Network connection[a]			
Connection Description	Gigabit Ethernet	Gigabit Ethernet	
Connection Type	Gigabit Ethernet	Vswitch/GbE	
Logical Volumes : Volume Group OS : 20GB Volume Group DB : 150GB Volume Group WAS: 80GB Volume Group MGM: 20GB			

HARDWARE PLANNING CHECKLIST			
SERVERNAME :			
IP Address/Netmask	9.12.7.88/28	9.12.7.88/28	
Vlan number : Vswitch	2	2 : Vswitch1	
Disk Resource[b]			
OS Filesystem	/ : 30 : Ext3	/ : 2 :Ext4	Root
Mount Point : Size(in GB) : Type		/opt : 3 :Ext4 LV OS	Logical Volume
Mount Point : Size(in GB) : Type		/var : 5 :Ext4 LV OS	
Mount Point : Size(in GB) : Type		/var : 5 :Ext4 LV OS	
Mount Point : Size(in GB) : Type		/tmp : 1 :BRTFS LV OS	
DATA Filesystem			
Mount Point : Size(in GB) : Type	/DB : 100 : Ext3	/DB:100:Ext4 LV DB	Logical Volume
Mount Point : Size(in GB) : Type	/WAS : 50 : Ext3	/WAS:50:Ext4 LV WAS	
CUSTOM Filesystem			
Mount Point : Size(in GB) : Type		/MGM:10:Ext4 LV MGM	Logical Volume
Logical Volumes : Volume Group OS : 20GB Volume Group DB : 150GB Volume Group WAS: 80GB Volume Group MGM: 20GB			

a. For IBM System z, the available network connections are:
- QETH
- HiperSockets
- Direct OSA-Express connection

b. We used the Logical Volume Manager (LVM) for the Linux environment since it provides flexibility and reduces downtime of the environment with online resizing of the logical volumes

8.2 Migrating DB2 and its data

In this section, we discuss the migration of DB2 data from the source system (x86 named zs4p01-s1) to our target system (System z named LNSUDB1). To perform these steps, you should use the DB2 administrator user ID.

1. Copy the CREATE DATABASE command used to create the database on the source system to a file called CREATEDB_TRADE6DB.SQL and transfer it to the target system. This can be done via FTP or any FTP application such as Filezilla. Edit the file locations to values that are applicable to the target system. In our environment, we use an iSCSI disk for our data called /db2_data. We want to use the same iSCSI disk on the new system. See section 8.4, “Migrating Fibre Channel devices” on page 159 for more information about switching the disk. Example 8-1 shows our Create Database command. Do not run this command just yet.

Example 8-1 Command to create database

```
CREATE DATABASE TRADE6DB ON /db2_data
```

2. The **DB2LOOK** tool is used to extract the required data definition language (DDL) statements needed to reproduce the database objects of one database into another database. The tool can also generate the required SQL statements needed to replicate the statistics from the one database to the other, as well as the statements needed to replicate the database configuration, database manager configuration, and registry variables. This is important because the new database might not contain the exact same set of data as the original database but you might still want the same access plans chosen for the two systems. The **DB2LOOK** tool should only be used on databases running on DB2 servers of Version 9.5 and higher levels.

   Generate the DDL using the **DB2LOOK** command:

   ```
   db2look -d trade6db -e -x -l  -o trade6db.sql
   ```

   where:

   -d: name of database

   -e: extract the database objects

   -x: generates authorization DDL statements such as GRANT statements.

   -l: generates DDL statements for user-defined database objects

   -o: the name of the output file

   For more information about the **DB2LOOK** tool, see the following site:

   https://ibm.biz/BdRJw6

   Send this file via FTP to your target location.

3. Since the **DB2 Backup** command cannot be used to move data between operating systems, you need to use the **DB2MOVE EXPORT** command, as shown in Example 8-2. Ensure that the output of **DB2MOVE** is directed to an empty directory, as both an *.ixf file and an *.msg file will be created for every table in the database.

Example 8-2 DB2MOVE export command

```
db2move trade6db export
```

4. While installing DB2 on LNSUDB1, we ran out of space. We simply opened IBM Wave for z/VM, found our guest, right-mouse clicked it and selected **More Actions** → **Manage Storage**. We created a new partition in a matter of seconds.
5. On the target system, using the DB2 administrator user ID, execute the file with the **CREATE DATABASE** command. Example 8-3 demonstrates how we ran our command using the file we created in Step 1.

Example 8-3 Executing the create database command

```
db2 -tvf CREATEDB_TRADE6DB.SQL -z create db.log
```

6. Create the database objects by using the DDL file created by the **DB2LOOK** tool:

   ```
   db2 -tvf trade6db.sql -z ddl.log
   ```

Note: This assumes that you are migrating from a Linux on x86 to Linux on System z environment. See *Practical Migration to Linux on System z*, SG24-7727 for specifics on migrating from MS Windows to Linux on System z.

7. After the DDL file has been successfully processed, import the data into the database by using the **DB2MOVE LOAD** tool as shown:

   ```
   db2 db2move trade6db load
   ```

Your database is now ready to be used.

8.3 Migrating the WebSphere Application Server

The process of migrating from one WebSphere Application Server environment to another is straightforward. The application was migrated using the source system's WebSphere Administrative Console **EXPORT** command. This created an enterprise archive (EAR) file on the client running the WebSphere Application Server console using the Firefox web browser. 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 6 (RHEL6) was undertaken using the target system's WebSphere Administrative Console **IMPORT** command. Both the export and import actions completed without incident.

8.4 Migrating Fibre Channel devices

Disk storage on servers often is made available by using a storage area network (SAN). A widely spread technology to distribute storage devices is Fibre Channel.

In the distributed world, the term Fibre Channel is often abbreviated as FC. FC in mainframe terminology stands for FICON, so a new abbreviation was introduced, FCP, which stands for Fibre Channel Protocol.

Typically, a SAN with Fibre Channel consists of two independent fabrics, which are point-to-point connectivity between processor and peripheral devices. All the adapters have their own unique worldwide number (WWN) which is put into a zone within the fabric. The servers typically have two interfaces that reside in different fabrics.

Modern Fibre Channel adapters can be virtualized by using N_Port ID Virtualization (NPIV). They provide a number of different virtual devices that all have their unique WWN and thus can be put into their specific zone. Distributed servers that have only one operating system commonly do not need this feature because the base WWN is sufficient to make all needed disk devices available.

When consolidating multiple servers to a single server, NPIV is a method to separate the disk devices for the different servers. These separations are important, and NPIV makes the separation possible even though devices are attached over the same physical connection. This holds true for the mainframe, where the FCP devices can be switched to NPIV mode. Unlike distributed systems, the mainframe has its own idea about which WWN is used for which device number. It is important to understand that the WWNs also differ from machine to machine, and when migrating from a distributed environment to a mainframe, an update of the zoning and of the storage system normally is needed. These changes are described in detail below.

8.4.1 Zoning for FCP

To connect the NPIV adapters to the respective logical unit number (LUN) on the storage device, the Fibre Channel switches must support zoning, and zoning must be set up accordingly. Each adapter must be added to a zone that also contains the adapter of the storage device. The storage then selects the correct disk by using the NPIV WWN.

In theory, just one zone with all adapters and storage adapters would be sufficient. For actual production deployments, create a separate zone for each of the NPIV devices. The reason is

that during logon and logoff of a single NPIV device, the whole zone is rediscovered. While this does not cause errors, it still can cause short hangs depending on the size of the zone. If a separate zone is created for each NPIV device, only the local zone is discovered, which has no affect on other zones.

8.4.2 FCP and multipath

The failover configuration for FCP is not handled by PR/SM or z/VM, but must be done from within Linux for System z. Therefore, two NPIV adapters must be attached to the guest system that is connected over the two different Fibre Channel fabrics.

The multipath setup itself is configured inside the Linux guest system with the configuration file `/etc/multipath.conf`. After the multipathd daemon is started, all available LUNs together with their paths can be checked with the command:

```
multipath -ll
```

Note regarding SLES: The behavior of SCSI devices changes between SLES11 and SLES12. In SLES11, all LUNs had to be configured entirely manually. With SLES12, automatic LUN scanning has been switched on, and therefore all LUNs will automatically be detected after the zfcp host has been configured.

The actual configuration of multipath depends on the storage device used. For DS8000 storage systems, the configuration shown in Example 8-4 can be used.

Example 8-4 multipath.conf sample configuration file

```
defaults {
  path_grouping_policy multibus
  failback 1
  rr_min_io 10
  path_checker tur
  checker_timeout 60
}

devices {
  device {
    vendor "IBM"
    product ""
    path_grouping_policy group_by_prio
    prio alua
  }
}
```

Multipath is needed when using FCP to prevent disk failures. Without multipath, any failure in a host bus adapter (HBA), fiber optic cable, or transceiver can cause unrecoverable data loss. Multipath is needed for normal operations, when important updates to the HBA microcode are required. Disk traffic can continue on one path while the other path is down for the update. Without multipath, all services involving the FCP disk would have to be halted while upgrading the microcode.

8.4.3 FCP migration setup tasks

The migration of Fibre Channel devices from a distributed system to the mainframe with NPIV involves several different tasks:

1. Dedicate NPIV adapters

   Assuming that NPIV adapters already have been configured within the CEC, a pair of NPIV adapters that is attached to the two different zones must be dedicated to the new guest system. For redundancy reasons, those adapters should come from two different physical cards.

   It is good practice to always dedicate the same virtual device (vdev) number. For example, if you configured two device ranges FA00-FA1D and FC00-FC1D for the two fabrics, dedicate the same generic vdevs as pairs for each guest:

   ```
   DEDICATE FA00 FA06

   DEDICATE FC00 FC06
   ```

   That way, all the virtual addresses are always FA00 and FC00 and the numbering is relatively obvious.

2. NPIV WWNs

   To retrieve the WWNs of the respective NPIV adapter, proceed as follows:

   a. Note the name of the LPAR that runs your z/VM system.

   b. Log on to the System z Support Element (SE) of the mainframe, either via the Hardware Management Console (HMC) with “Single Object Operations”, or directly.

   c. On the SE, find the NPIV adapter information at “System Management” → <system name> → CPC Configuration → FCP Configuration. Where <system name> is your system name.

   d. The easiest way to retrieve the information is to transfer the file with the WWNs for the z/VM LPAR to a remote FTP server via FTP.

   e. The resulting file is a comma-separated list, which looks like the following:

   ```
   LP1,00,01,14,00,f91c,c05076e0f3002d70,0n,Yes,05a0,c05076e0f3005a01
   ```

   f. To retrieve the WWNs for “fa06” and “fc06” on LP1, use the following command:

   ```
   # grep LP1 npiv-list.csv | grep -e fa06 -e fc06 | cut -d, -f 6,7
   fa06,c05076e0f3000798
   fc06,c05076e0f3001e18
   ```

3. Zoning update

   The WWNs that have been retrieved from the SE must be used to update the Fibre Channel zones. If all servers already have their own individual zone, just add the WWNs from step 2 to the respective fabric. If just one big zone exists at this time, split the zone up into smaller zones for each server during the process. The intended result is to have a pair of zones for the two fabrics for each of the migrated servers.

4. Storage system update

   Inside the storage system, the host connections are configured according to the WWN of the Fibre Channel adapter. Without NPIV, this is just the base number. While with NPIV, the respective NPIV WWN is used. Normally, it is possible to add several host connections to a specific storage group. This allows several Fibre Channel adapters to access the same LUNs. To prepare a migration, add the new WWN to the configured host connections. After the restart of the service on the mainframe has finished, remove the old WWN from the host connection.

9

Post migration consideration

This chapter describes general post migration consideration concepts for getting acceptance, measuring performance, and tuning. Topics covered in this chapter include an acceptance list, performance measurement understanding, and performance tuning key items.

Every migration poses a big challenge for IT organizations because each stakeholder has different expectations and requirements from the project. Most of the topics after migration will center around “Performance” and “Functionality”. IT organizations face some difficult questions:

- What exactly has been done?
- Is there anything missing?
- Is everything working?
- Is the performance as expected?
- Is the process completed?
- Did we get approvals?

In order to answer these questions, some important steps are needed before and after the migration implementation phase. In this chapter, we focus on three important topics once the migration is completed

- Acceptance
- Performance measurement
- Performance tuning

9.1 Gaining acceptance

Migration projects are generally recognized as major changes to the IT environment. Each change requires significant test and acceptance by various stakeholders. Decisions must be made by these stakeholders whether the migration was a success.

Acceptance requires an understanding of the big picture, before and after migration:

- Before implementation phase start:
  - Decide and document test scope
  - Decide and document test case (including test scenario)
  - Create post migration checklist for all components
  - Collect performance data on system
  - Get acceptance from stakeholder for test
- After implementation done:
  - Use post-migration checklist and check whether implementation done or not
  - Test system using by documented Test Case (Complete and document all test scenario)
  - Measure performance and compare with previous performance data
  - If necessary, perform performance tuning

Based on project scope and context, items used for acceptance testing can change but the following list is the most common acceptance tests performed before gaining stakeholder acceptance:

- Application testing

  In some cases Usability testing may be required
- Functional testing
- Performance testing
- Security testing
- User acceptance testing

9.2 Performance measurement

In this section, we describe performance measurement and its impact on the success of your migration. The most important point to consider is that you need to measure the performance of the application when it is running in production on the source environment and then compare that with the performance of the application on the target environment.

We also describe monitoring commands and tools that can assist you in identifying and resolving performance inhibitors.

9.2.1 What is performance

“Performance” in computer systems is very much a relative term. Usually computer performance is described in measurable terms, such as transactions per second, response time, time to process a booking or insurance sale. However, when a migration project is

undertaken, it is important to understand the performance metrics used on the source environment so that you can understand the relative performance on the target system.

The initial performance of a new system may often not be as expected especially when changing hardware platforms. Therefore, tuning must be undertaken to improve the performance of the target system. Without having proper metrics, it is impossible to validate the performance of the new platform relative to the former platform. For this reason, the migration project team first needs to agree on what performance metrics from the source platform will be used in the migration project plan to measure the performance of the target platform.

9.2.2 Choosing what to measure

To determine the success of a migration, simply having the application on the target platform provide the same answers as the source platform does not prove success. The natural expectation of a migration onto Linux on System z is that the application will not only be more resilient and available because of System z, but that it will also provide equal or better performance than the source platform. To ensure that the performance improvements are easy to show, it is important to choose the right metrics. But what are these metrics, and how should they be measured?

Response time

Response time is the measure of the time it takes for something to happen in a computer system. Generally we choose to measure the response time of a unit of work called a *transaction*. This could entail something as simple as checking an account balance, to something as complex as the time taken to issue a new insurance policy or open a new bank account.

The point to remember with computer systems is that the response time of a single transaction is the sum of a number of response times. Figure 9-1 shows the various components that make up user response time.

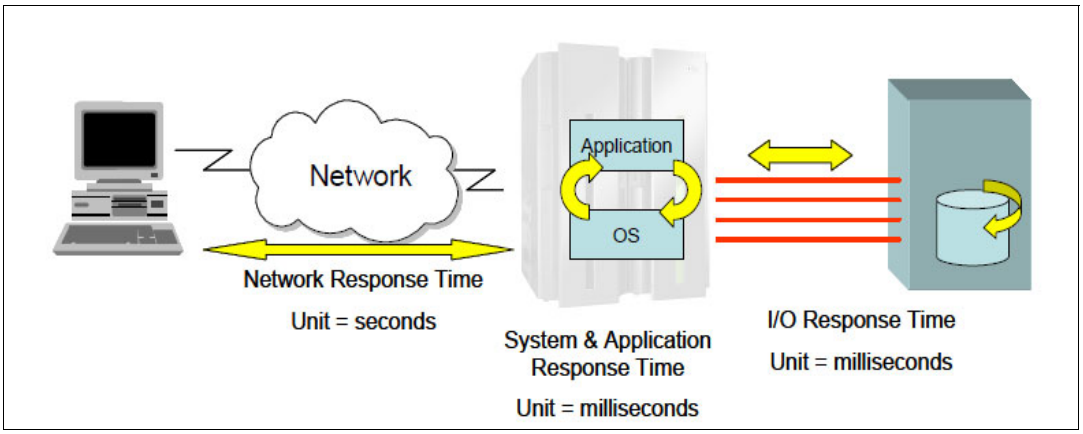


Figure 9-1 Components that make up the response time of transactions

The figure shows that there are two points where response time could be measured: system response time and user response time. When you are trying to understand the relative performance improvement from a new system, the only point to measure response time is from when a system receives the request and when it provides a response of some sort to the request.

In the case illustrated by Figure 9-1 on page 165, the system response time will include application time and the I/O response time to access the data. If you choose to measure the response time of user experiences at their terminal or over the web, you will be adding in the network response time, which can vary greatly for the same transaction because it can be influenced by network load.

To compare the source and target systems directly, the recommended approach is to measure system response time on the source system, and assuming the application has not changed greatly, measure the system response time on the target platform.

Transaction throughput

The transaction throughput performance metric may provide a more meaningful measure of system performance because it measures the number of transactions processed over a period of time. This is typically one second, but could be any time period that you prefer.

In both cases, you should have baseline performance metrics for the source system to properly compare both the new and old systems.

9.3 Performance tuning

Tuning any system should follow some principles because every hardware and software platform has unique features and characteristics that must be considered when you tune your environment. The art of tuning performance in a system to require success performance analyses, multi-step tuning process and change management strict combination.

Regardless of which tools you choose, the best methodology for analyzing the performance of a system is to start from the outside and work way down to the small tuning details in the system. Start gathering data about overall health of the system hardware and processes. The following list is a sampling of the types of questions you should answer about both your source and target systems:

- How busy is the processor during the peak periods of each day?
- What happens to I/O response times during those peaks?
- Do they remain fairly consistent, or do they elongate?
- Does the system get memory constrained every day, causing page waits?
- Can current system resources provide user response times that meet service level agreements?

It is important to know what tuning tools are available and what type of information they provide. Equally important is knowing when to use those tools and what to look for. How will you know what is normal for your environment and what is problematic unless you check the system activity and resource utilization regularly? Conducting regular health checks on a system also provides utilization and performance information that you can use for capacity planning.

Tuning is not a one-size-fits-all approach, as a system tuned for one type of workload performs poorly with another type of workload. This means that you must understand the workload that you want to run and be prepared to review your tuning efforts when the workload changes. A simple workload is a server that shows one or more peaks during the day, while a complicated workload is an application that is CPU-intensive during part of the day and I/O-intensive during another part. The most cost-efficient approach to running these workloads is to adjust the capacity of the server during the day. This is exactly what z/VM

carries out. Portions of the virtual machine are brought in to run in main memory while inactive virtual machines are moved to paging to create space.

Multi-step tuning process requires the skills of a systems performance detective. A systems performance analyst identifies IT problems using a detection process similar to that of solving a crime. In IT systems performance, the crime is a performance bottleneck or sudden degrading response time. The performance analyst asks questions, searches for clues, researches sources and documents, reaches a hypothesis, tests the hypothesis by tuning or other means, and eventually solves the mystery, which results in improved system performance. Bottleneck analysis and problem determination are facilitated by sophisticated tools such as IBM Tivoli OMEGAMON on z/VM and Linux. OMEGAMON detects performance problems and alerts you before degraded response time becomes evident. OMEGAMON detects a potential performance or availability problem and sends a warning alert for z/VM and OMEGAMON console.

Change management that is not strictly related to performance tuning is probably the single most important factor for successful performance tuning. The following considerations highlight this point:

- Implement a proper change management process before tuning any system.
- Never start tweaking settings on a production system.
- Never change more than one variable at a time during the tuning process.
- Retest parameters that supposedly improve performance; sometimes statistics come into play.
- Document successful parameters and share them with the community no matter how trivial you think they are. System performance can benefit greatly from any results obtained in various production environments.

A

Additional use case scenarios

The complexity of a migration from Linux on the x86 may change by platform architecture and context of the migration. The Linux operating system is more straightforward and well-known and makes the possibility for migration much easier for technical people. However, when you consider an application, database management system, or middleware migration, you need to consider degrees of complexity, cost, and risk.

In this appendix, we describe additional use case scenarios where a telecommunications company, a healthcare company, and an energy and utilities company all want to migrate from x86 to Linux on System z. We discuss the challenges inherent to each industry and describe their respective migration scenarios.

Telecom industry consolidation and cloud

In this scenario, the fictional telecom provider, Fictional Telco Company T1, selects the IBM System z platform for their Linux operating system consolidation and virtualization. Telco Company T1 wants to build a cloud platform but they also want to reduce their cost of operation and overall data center footprint. The company's strategy is to improve provisioning time for its business support system (BSS) and operational support system (OSS) to satisfy server requests of its users. In this example, the following technology can be employed:

Consolidated hardware infrastructure:

- IBM System z zEC12 or zBC12
- IBM z/VM 6.3
- Red Hat Enterprise Linux or SUSE Linux Enterprise Servers on the System z platform
- IBM protecTIER Gateway TS7680: Deduplication and Virtual Tape Library

Cloud:

- IBM SmartCloud®:
  - Automation with cloud: IBM Tivoli System Automation
  - Automated provisioning: Tivoli Provisioning Manager
  - Service Lifecycle Management: IBM SmartCloud Control Desk

Build monitoring and system management:

- IBM Tivoli OMEGAMON on z/VM and Linux: Information about your Linux instances running as z/VM guests and the Linux workloads reveal how they are performing and impacting z/VM and each other:
  - Compare Linux operations side by side with detailed performance metrics.
  - Data collection from the Performance Toolkit for VM (PTK is a prerequisite) complements data collection by the IBM Tivoli Monitoring for Linux for System z agent.
  - With new Dynamic Workspace Linking, you can easily navigate between Tivoli Enterprise Portal workspaces.
  - View and monitor workloads for virtual machines, groups, response times and LPAR reporting, and view reports about z/VM and Linux usage of resources such as CPU utilization, storage, mini-disks, and TCP/IP.
  - High-level views help executives understand how systems performance influences business and the bottom line.
  - With granular views, IT staffs can more easily track complex problems that span multiple systems and platforms and share related information.
- IBM Wave for z/VM v1.1: IBM Wave is a new virtualization management product for z/VM and Linux virtual servers that uses visualization to dramatically automate and simplify administrative and management tasks:
  - Automate, simplify management, and monitor virtual servers and resources, all from a single dashboard.
  - Perform complex virtualization tasks in a fraction of the time compared to manual execution.
  - Provision virtual resources (servers, network, storage) to accelerate the transformation to cloud infrastructure.

– Use advanced z/VM management capabilities such as Live Guest Relocation with a few clicks.
– Delegate responsibility and provide more self-service capabilities to the appropriate teams.

Figure A-1 shows the solution architecture overview for a cloud solution using Linux on System z.

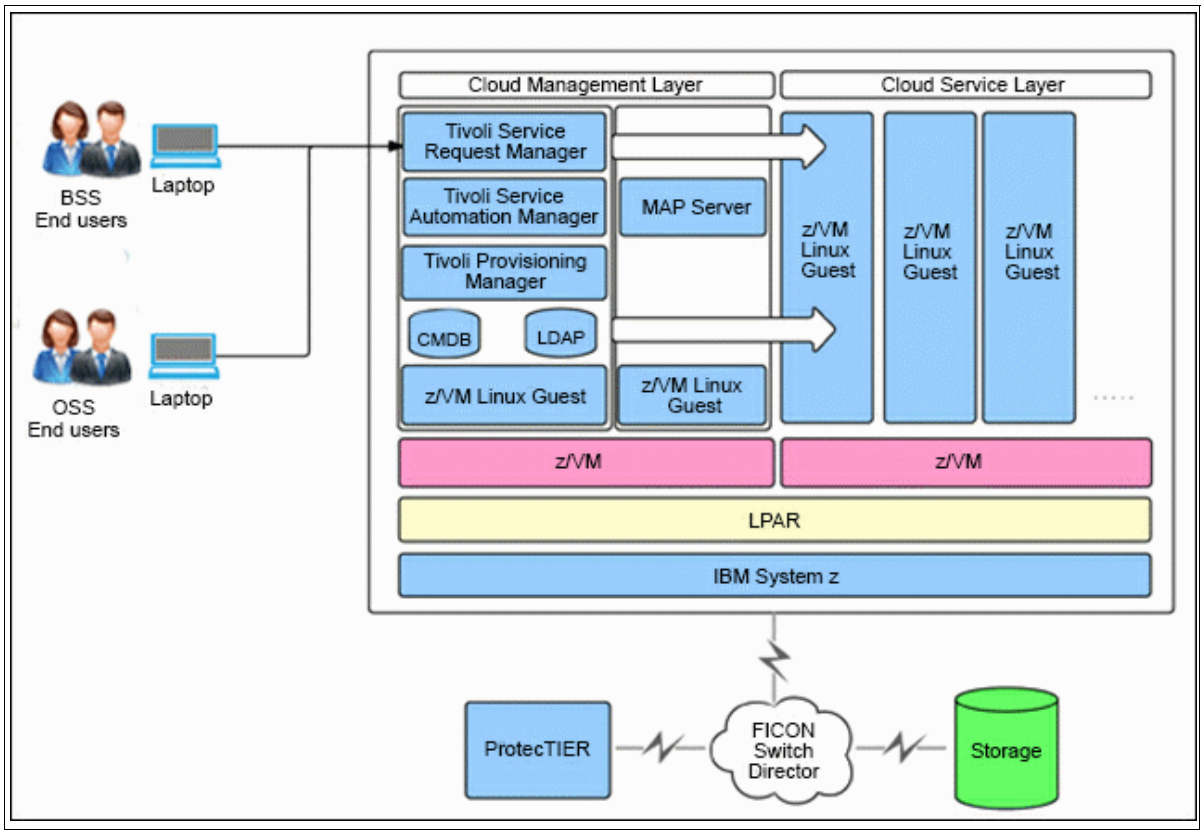


Figure A-1 Cloud solution using Linux on System z

Healthcare industry: Mobile and Internet solution

In this scenario, the fictional healthcare company, Fictional Hospital H1, also chooses Linux on System z as its mobile application platform. Hospital H1 wants to build a secure platform, increase responsiveness, and value perception, and reduce multi-platform development costs.

Build a secure platform

- IBM Worklight® provides an extensible authentication model as part of its function. To comply with the Federal Information Processing Standards (FIPS), Hospital H1 uses Worklight with WebSphere Application Server for added protection. The hospital configures WebSphere Application Server to protect the application and adapters for the back-end servers and data.
- Using Worklight, Hospital H1 can grant access to data on a role, time, and location basis. Doctors can access patient records on mobile devices. However, it requires extra authentication approval if they are at home or on call to review the latest observations of

patients. In addition, although doctors have access to the information of their patients, medical suppliers have access to check inventory and update stock.

Increase responsiveness and perceived value perception

- Hospital H1 is looking for a communication solution to find employees anywhere in the hospital. Using Worklight, the hospital can build an application that allows instant and secure communication. Doctors and nurses can quickly find colleagues without stopping what they are doing.
- Doctors at Hospital H1 must input prescriptions when their mobile devices are not connected to the network. JSONStore, the document-oriented storage system in Worklight, uses an encrypted container and ensures that the documents in the application are always available to doctors even when the devices running the application are offline.
- With the application, patients can pre-register for appointments and input their allergies and health history by using mobile devices. Worklight uses Secure Sockets Layer with server identity verification and enables communication over HTTPS to protect the information.

Reduce multi-platform development costs

- Worklight provides a standards-based platform and allows Hospital H1 to use third-party libraries and frameworks.
- Using Worklight, Hospital H1 can also create mobile applications quickly by using any combination of HTML5, native, and hybrid development methods.

Figure A-2 on page 173 shows the secured access from a mobile device to a back-end transactional core system on the Linux on System z platform by using the global security policies and end-to-end secure transactions.

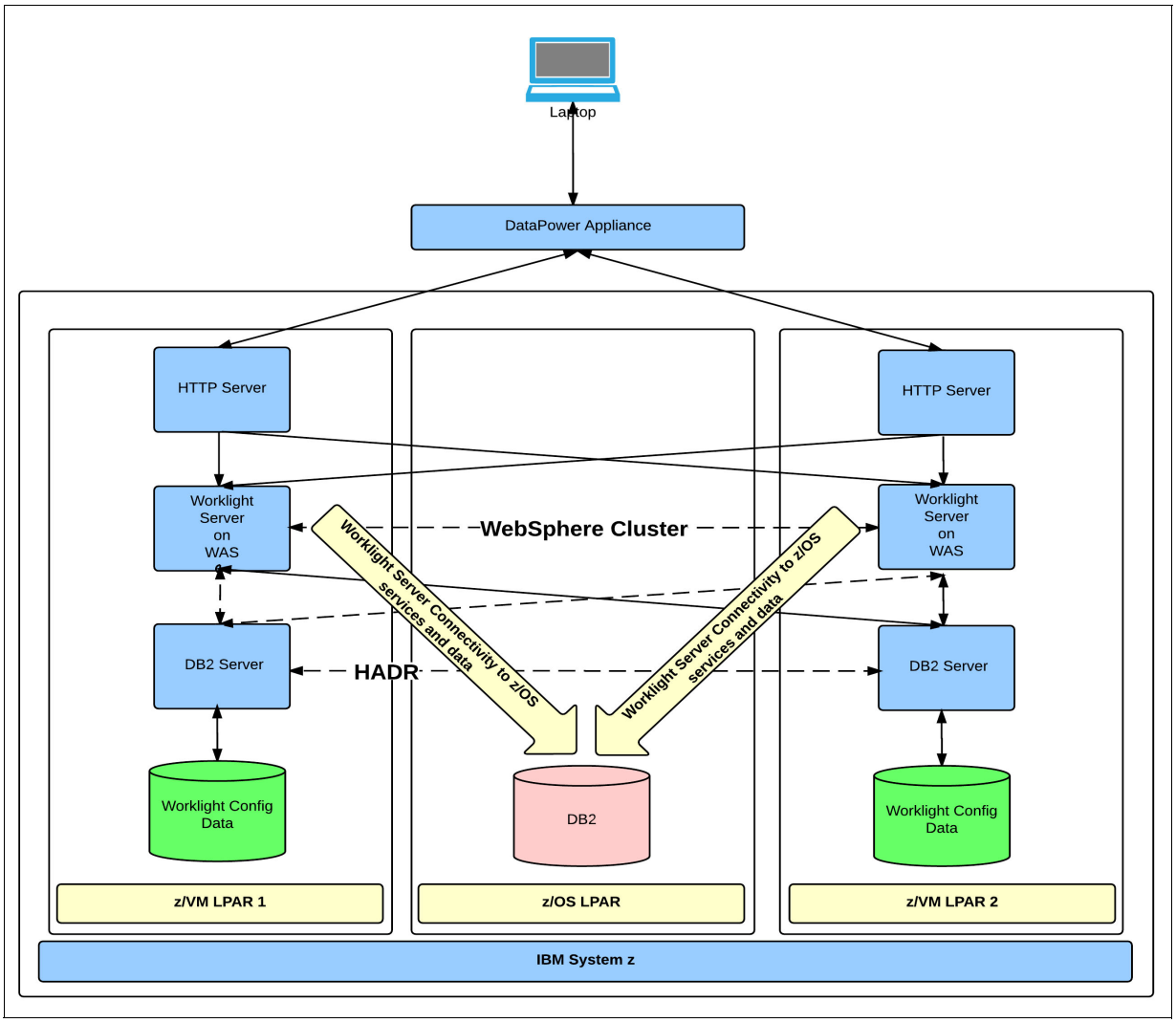


Figure A-2 Access from a mobile device to a back-end transactional core system

Energy and utilities industry: SAP Cloud and Automation solution on System z

In this scenario, the fictional energy and utilities company, Fictional Energy E1, also chooses System z platform as its SAP application running on Linux and database on IBM z/OS. Energy E1 wants to reduce the time spent to copy and refresh complete SAP systems from days to hours with a cloud solution and SAP system automation, which can automate, standardize, and increase the speed of day-to-day operations for SAP systems, reducing the risk of mistakes caused by human error. The company wants to reduce time spent on complex, repetitive tasks, freeing up skilled staff for higher value work and deliver higher operational efficiency, helping to slash costs and accelerate the time-to-value ratio for new workloads.

Build a virtual platform:

- IBM zEC12 or zBC12
- IBM z/VM 6.3

- Red Hat Enterprise Linux or SUSE Linux Enterprise Server on System z platform
- IBM DB2 for z/OS
- IBM Database Provisioning System (DPS)
  - Web application JCL Engine
  - Database Management
  - Integrated with DB2 Cloning Tool
- IBM DB2 Cloning Tool for z/OS: The DB2 Cloning Tool automates the cloning process to provide usable DB2 clones within minutes, boosting efficiency and freeing up DBA time:
  - Quickly clones DB2 subsystems, DB2 table spaces, or index spaces to create up-to-date test environments.
  - Automates the cloning process to provide usable DB2 clones within minutes.
  - Clones a DB2 subsystem by renaming and cataloging the data sets, fixing the volume internals, and updating the DB2 internal control information.
  - Fast copy technology quickly copies DB2 data sets within a subsystem or to a different subsystem.
  - Automates the cloning process using any volume level technology, such as IBM FlashCopy, to clone DB2 subsystems and any data set copy technology, such as FlashCopy, to clone table and index spaces and automatically translates the object IDs to simplify and automate the refresh of data.
- SAP NetWeaver Landscape Virtualization Management (LVM): By streamlining and automating critical business processes, SAP NetWeaver Landscape Virtualization Management software enables your IT department to focus on responding to new initiatives, controlling IT costs, and differentiating your business:
  - Manage your SAP landscape in physical and virtualized environments
  - Central management point for your SAP landscape, start/stop, and mass operations
  - Automate standard, day-to-day administrative and lifecycle management tasks
  - Save time, effort, and money by automating copy, clone, and refresh
- Build IBM Entry Cloud Solution for SAP with automated lifecycle management operations:

  The IBM Entry Cloud Configuration solution automates complex tasks typically performed by administrators of databases, operating systems, storage systems, and SAP Basis. When combined with SAP NetWeaver LVM, the configuration can reduce the time that it takes to copy and refresh complete SAP systems from days to hours. The high degree of automation also improves the quality and efficiency of SAP operations.

  To build an IBM Entry Cloud solution for SAP, they will select automated lifecycle management operations such as:
  - SAP System Clone: Provision a fresh SAP system based on a new system copy
  - SAP System Copy: Create a customized SAP system based on an existing system
  - SAP System Refresh: Copy DB content from PRD to Non-PRD including post processing
  - Create an additional dialog instance: Adding additional application server instances, for example, for monthly closing

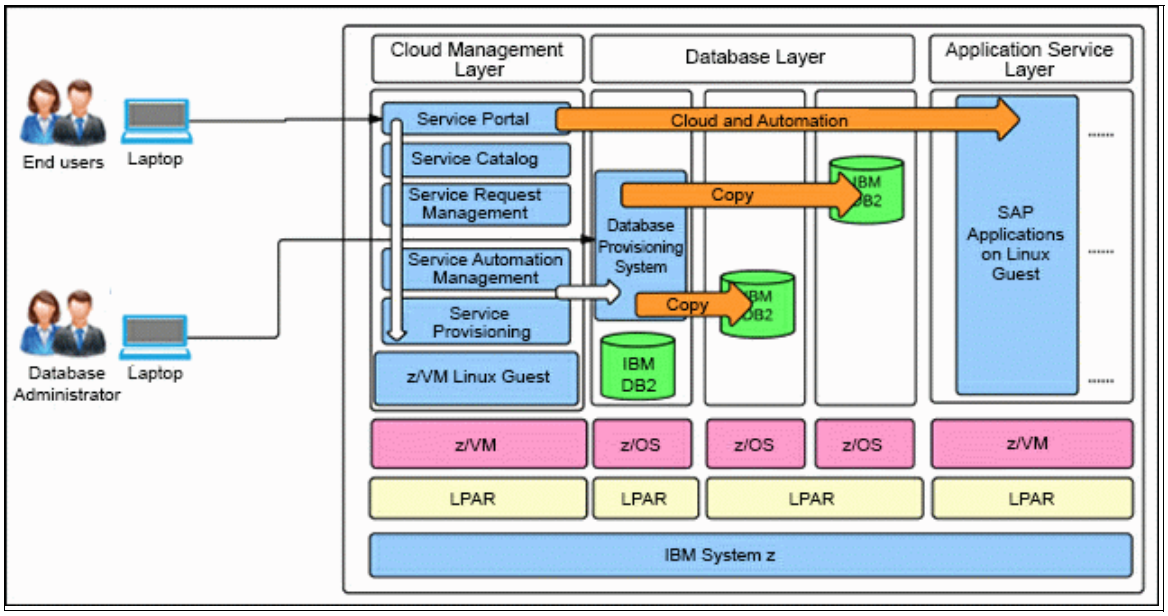


Figure A-3 Automate SAP System Copy with IBM Entry Cloud Solution for SAP

The Linux on System z, IBM Entry Cloud Configuration solution is the ideal productivity tool for any IT organization running SAP Business Suite on zEnterprise with IBM DB2 for z/OS. It is well-suited for computer services organizations hosting SAP systems for their clients, and for any IT organization seeking to run its SAP operations with zEnterprise in an on-premises, self-managed, cloud computing environment. Figure A-3 shows the added value of this solution and how it reduces operation and administration time when compared to traditional operations.

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

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.

- *IBM zEnterprise EC12 Technical Guide*, SG24-8049
- *Introduction to the New Mainframe: z/VM Basics*, SG24-7316
- *An Introduction to z/VM Single System Image (SSI) and Live Guest Relocation (LGR)*, SG24-8006
- *The Virtualization Cookbook for IBM z/VM 6.3, RHEL 6.4, and SLES 11 SP3*, SG24-8147
- *Using z/VM v 6.2 Single System Image (SSI) and Live Guest Relocation (LGR)*, SG24-8039
- *Advanced Networking Concepts Applied Using Linux on IBM System z*, SG24-7995
- *Set up Linux on IBM System z for Production*, SG24-8137
- *Implementing the IBM System Storage SAN Volume Controller V7.2*, SG24-7933
- *Introduction to Storage Area Networks and System Networking*, SG24-5470
- *IBM System Storage SAN Volume Controller Best Practices and Performance Guidelines*, SG24-7521
- *Implementing FlashSystem 840 with SAN Volume Controller*, TIPS1137
- *IBM Wave for z/VM: Installation, implementation and exploitation*, SG24-8192
- *Linux on IBM System z: Performance Measurement and Tuning*, SG24-6926
- *DB2 10 for Linux on System z Using z/VM v6.2, Single System Image Clusters and Live Guest Relocation*, SG24-8036
- *Experiences with Oracle 11gR2 on Linux on System z*, SG24-8104
- *Experiences with Oracle Solutions on Linux for IBM System z*, SG24-7634
- *Linux on IBM eServer zSeries and S/390: Application Development*, SG24-6807
- *Security on z/VM*, SG24-7471
- *Security for Linux on System z*, SG24-7728
- *IBM System z Connectivity Handbook*, SG24-5444

You can search for, view, download, or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Online resources

These websites are also relevant as further information sources:

- Linux on System z offers a variety of solutions

  http://www.ibm.com/systems/z/os/linux/solutions
- GNU assembler manual

  http://www.gnu.org/software/binutils
- IBM High Availability Center of Competency

  http://www-03.ibm.com/systems/services/labservices/solutions/hacoc.html

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

T

U

V

W

Z

IBM Redbooks

Practical Migration from x86 to Linux on IBM System z

(0.2”spine)
0.17”<->0.473”
90<->249 pages

Practical Migration from x86 to Linux on IBM System z

A guide to migrating popular applications and services from Linux on x86 to Linux on System z

Practical guidance on planning, analysis, and TCO

Comprehensive hands-on migration case study

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

- 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 guide and example for IT organizations to migrate from their x86 environment to Linux on System z. It begins by examining the benefits of migrating workloads to Linux on System z. Here, we describe the workload centric method of information technology and then discuss the benefits of migrating workloads to Linux on System z.

Next, we describe total cost of ownership analyses and we guide you in understanding how to analyze your environment before beginning a migration project. We also assist you in determining the expected consolidation ratio for a given workload type.

We also describe virtualization concepts along with describing the benefits of migrating from the x86 environment to guests residing on an IBM z/VM single system image with live guest relocation.

This IBM Redbooks publication walks you through a migration approach, includes planning worksheets, as well as a chapter to assist you in analyzing your own systems. We also discuss post migration considerations such as acceptance testing of functionality and performance measurements.

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

SG24-8217-00 ISBN 0738439894