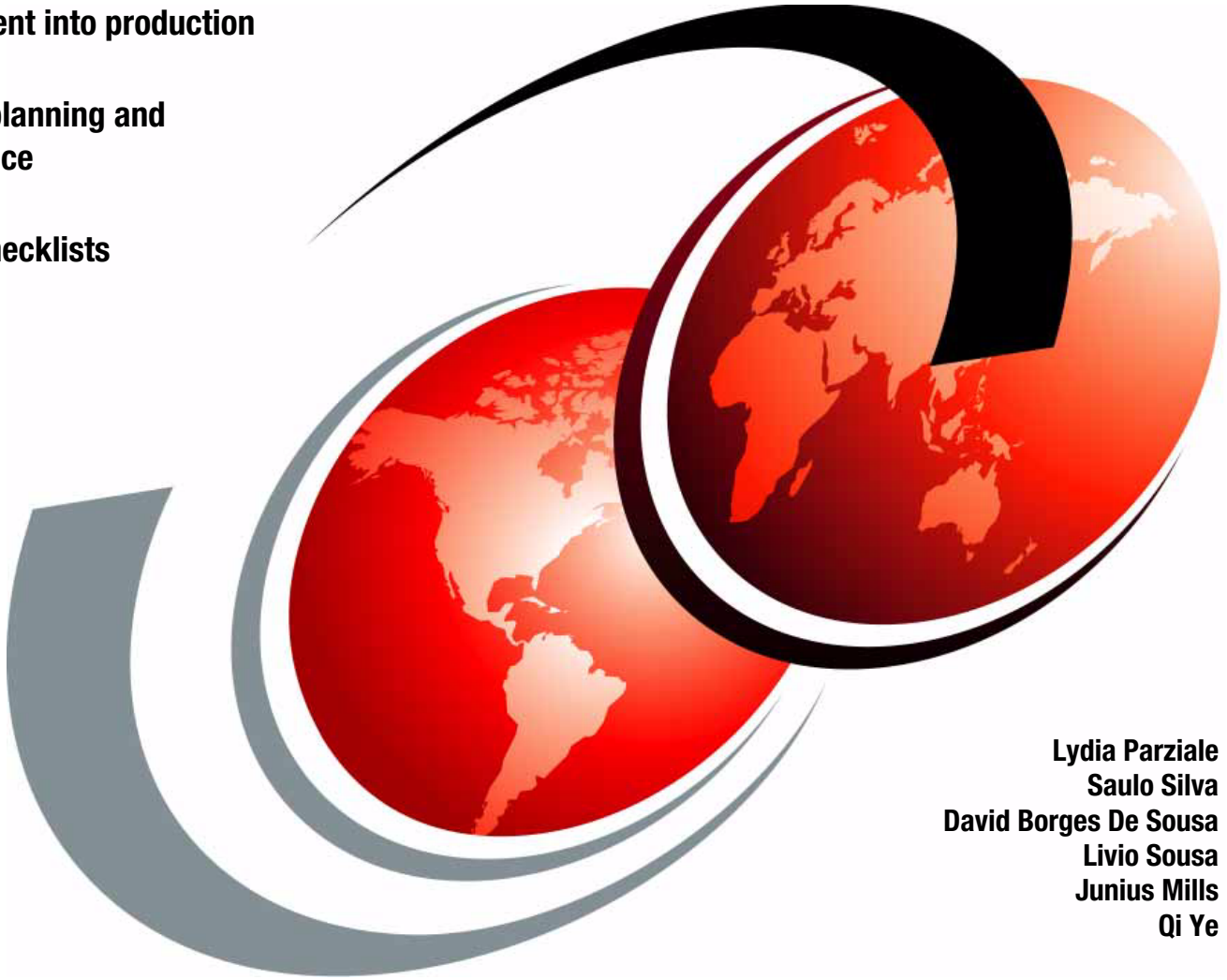
IBM

Set up Linux on IBM System z for Production

- Considerations to move from test and development into production
- Capacity planning and performance
- Sample checklists



Lydia Parziale
Saulo Silva
David Borges De Sousa
Livio Sousa
Junius Mills
Qi Ye

Redbooks

ibm.com/redbooks

International Technical Support Organization

Set up Linux on IBM System z for Production

November 2013

SG24-8137-00

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

First Edition (November 2013)

This edition applies to Version 6, Release 2 of IBM z/VM as well as Red Hat Enterprise Linux versions 6.1 and SUSE Linux Enterprise Server 11 SP1.

© Copyright International Business Machines Corporation 2013. All rights reserved.
Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

© Copyright IBM Corp. 2013. All rights reserved.

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.

© Copyright IBM Corp. 2013. All rights reserved.

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:

Cognos®
DB2®
DirMaint™
DS8000®
ECKD™
ESCON®
FICON®
FlashCopy®
GDPS®
HiperSockets™
HyperSwap®
IBM SmartCloud®
IBM®
MVS™
OMEGAMON®
PR/SM™
Processor Resource/Systems Manager™
RACF®
Redbooks®
Redpaper™
Redbooks (logo)  ®
Resource Link®
System z10®
System z®
Tivoli®
WebSphere®
z/Architecture®
z/OS®
z/VM®
z/VSE®
z10™
zEnterprise®

The following terms are trademarks of other companies:

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

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

This IBM® Redbooks® publication shows the power of IBM System z® virtualization and flexibility in sharing resources in a flexible production environment. In this book, we outline the planning and setup of Linux on System z to move from a development or test environment into production. As an example, we use one logical partition (LPAR) with shared CPUs with memory for a production environment and another LPAR that shares some CPUs, but also has a dedicated one for production. Running in IBM z/VM® mode allows for virtualization of servers and based on z/VM shares, can prioritize and control their resources.

The size of the LPAR or z/VM resources depends on the workload and the applications that run that workload. We examine a typical web server environment, Java applications, and describe it by using a database management system, such as IBM DB2®.

Network decisions are examined with regards to VSWITCH, shared Open Systems Adapter (OSA), IBM HiperSockets™ and the HiperPAV, or FCP/SCSI attachment used with a storage area network (SAN) Volume Controller along with performance and throughput expectations.

The intended audience for this IBM Redbooks publication is IT architects who are responsible for planning production environments and IT specialists who are responsible for implementation of production environments.

Authors

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

Lydia Parziale is a Project Leader for the ITSO team in Poughkeepsie, New York, with domestic and international experience in technology management including software development, project leadership, and strategic planning. Her areas of expertise include Linux on System z and database management technologies. Lydia is a Certified IT Specialist with an MBA in Technology Management and has been employed by IBM for over 25 years in various technology areas.

Saulo Silva has 15 years of experience in Linux Architecture and Solutions and has specialized in Linux/390 for the last eight years. He has worked with the Linux on System z team doing the performance and monitoring job at the IBM Big Green project. He is the Linux on System z and Cloud Solution Leader for the Brazil geography. Saulo holds a Computer Information Systems degree from Faculdade de Ciencias Gerencias - UNA. He has RHCE certifications and an LPI certification. He co-authored other IBM Redbooks publications on subjects such as SystemTap Programming and Practical Migration to Linux on System z.

David Borges De Sousa is an IT Specialist in Brazil. He has five years of experience in IT infrastructure support for government and private companies and has been working at IBM for three years supporting Linux on System z. He is a member of the Linux on System z support team, which supports more than 1800 servers for the IBM internal global accounts. He is a post graduate in Quality and IT Management and is LPIC-I Certified.

© Copyright IBM Corp. 2013. All rights reserved.

Livio Sousa is a Certified IT Specialist with over 12 years of experience with high-end platforms, which encompasses servers, storage, and networking equipment. Throughout his career, he has had the opportunity to work with several different software platforms, such as IBM z/OS®, UNIX, Linux, and Windows, as well as different hardware processor architectures such as the IBM z/Architecture® and the IA-32 architecture and its extensions. For the past 10 years, he has been working as a System z Client Technical Specialist, responsible for provisioning support for the System z clients in Brazil, as well as responsible for supporting and developing skills for the z/VM hypervisor in Latin America. Livio holds an MBA in IT Management.

Junius Mills (J.B.) is a Certified I/T Specialist for z/VM and Linux on System z and is with the STG Team Eastern integrated marketing team (IMT) region. He has 35 years of experience with IBM and 13 years of client experience on z/VM and Linux on System z, specializing in the areas of implementation, customization, capacity, and performance tuning. He is the Regional Designated Support (RDS). Another area of focus for Junius is Business Analytics for the Eastern IMT.

Qi Ye is a Certified IT Specialist working for the Advanced Technical Skills (ATS) team in the IBM Growth Market Unit as a System z Technical Lead. He has over nine years of experience in System z technical support. He mainly focuses on System z hardware architecture, z/OS, DB2 for z/OS, z/VM, and Linux on System z. He currently provides System z technical sales support to IBM growth markets.

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

Roy P. Costa
IBM International Technical Support Organization, Poughkeepsie Center

Bruce J. Hayden
IBM ATS Specialist: z/VM and Linux on System z, IBM US

Edi Lopes Alves, Fernando Costa, Eric Marins, Willian Rampazzo, and Luiz Rocha
IBM IT Specialists: z/VM and Linux on System z, IBM Brazil

Martin Kammerer
IBM Germany

Filipe Miranda
Red Hat Global Lead for IBM System z, Red Hat US

Rafael D Tinoco
IBM Mainframe Lab Services Technical Leader, IBM Brazil

Thiago Sobral
SUSE Sales Engineer

Gary Loll
Solutions Architect IBM z/VM Linux on System z, IBM US

Eric Farman
System z Virtualization, IBM US

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:

  http://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

Introduction

There are two reasons why you might be reading this book. Either it is because you are looking for more information that is related to virtualized environments and want to learn more about Linux running on the IBM System z servers in a true virtualized environment. Or, you already are a Linux on System z user and are now in discussion with your colleagues and managers about how to guarantee that all your applications continue to run when you migrate to a new production environment.

In this section, we provide a number of reasons why your business should use Linux on System z as well as provide an overview of virtualization using System z. We also address some of the advantages of using Linux on System z for your production cloud environment.

© Copyright IBM Corp. 2013. All rights reserved.

1.1 Hardware

There are a number of reasons to use Linux on System z. System z hardware was originally delivered in the early 1960s and has evolved over the decades. In 2000, Linux on System z became a commercial reality and development investments were made to improve the Linux usability on the System z hardware. Resources can be shared among multiple Linux images running on the same z/VM system. Resource improvements include improvements to the CPU clock speed, size of the memory allocated per logical partition (LPAR), and the use of Fibre Channel Protocol (FCP) with IBM FICON® devices. For example, where once the System z 990 had 64 GB of memory, the zEC12 now has 3 TB of memory.

Additionally, there has been much investment in the IBM z/VM virtualization operating system that benefits Linux on System z guests. An example of this would be the faster speed of the Integrated Facility for Linux (IFL) on the new IBM zEnterprise® EC12 (zEC12), which enables you to host more virtual servers per processor core than other server platforms. Combined with the up to 101 user-configurable IFLs, the total Linux processing capacity has enormously increased with the zEC12.

Consolidation of server hardware resources allows for the running of tens or hundreds of Linux instances on a single System z server, offering savings in space and personnel that are required to manage real hardware.

1.2 Reliability, availability, and serviceability

Reliability, availability, and serviceability (RAS) is a concept that describes computers that can work for a long time without a single stop. The IBM System z is designed with redundancy of components to provide RAS. Batteries, power connections, cooling systems, processors, as well as memory, have been developed with full redundancy as the objective so that any device failure does not become an unplanned outage.

In 2012, the IBM System zEC12 machine was released and designed with a number of processor cores that can reach up to 101 active specialized processors that are called *Integrated Facilities for Linux* (IFL). There are 120 active cores in the zEC12. The reason for this is that there are up to 16 cores that are used as system assist processors (SAPs). Two cores are always used as spare cores and one core is for reserved use. The two spare cores can assume all software requests without stopping the application if one of the active specialized cores fails. Also, physical RAM has an algorithm that prevents the applications from stopping due to a failure of one single RAM.

Some other features that are related to high availability characteristics are Enhanced Book Availability (EBA), concurrent memory replacement using Licensed Internal Code Configuration Control (LICCC), redundant I/O interconnection, and more. For information about System z processors, memory, and other devices, see the *IBM zEnterprise EC12 Technical Guide*, SG24-8049.

1.3 System z virtualization

The virtual machine environment is highly flexible and adaptable. New Linux guests can be added to a z/VM system quickly and easily without requiring dedicated resources.

The System z hardware has two levels of virtualization. The first level is managed by the IBM Processor Resource/Systems Manager™ (PR/SM™) microcode. The PR/SM provides the ability to split single system hardware into up to 60 LPARs that are able to share the processors and I/O channel devices including storage devices and network device access. In an LPAR, it is possible to install and run a z/VM hypervisor or a Linux operating system as well as IBM operating systems, such as z/OS, z/TPF, or IBM z/VSE®.

By using the System z LPAR configuration, it is possible to increase memory, processor, and channels dynamically. These characteristics prevent a running application from stopping because of a possible capacity problem. Because RAM memory is the only resource at the PR/SM level that is not shared, it is recommended in a production System z environment that all memory area addresses not be allocated to all LPARS while the logical partition is defined. It is better to reserve some memory to increase an LPAR dynamically, if necessary.

Because the PR/SM is able to control and define the weight of priority for each LPAR defined on the system, the system administrator should separate the development environment from the production environment, whether sharing resources or not, but always prioritizing the order of execution. The definition of the LPAR weight can be in a relative basis. Suppose that you have 10 IFL System z machines and you shared those 10 IFLs between two LPARs, defined as *LPAR A* and *LPAR B*. LPAR A executes the production servers and LPAR B executes the development or even quality assurance (QA) servers, as described in Figure 1-1.

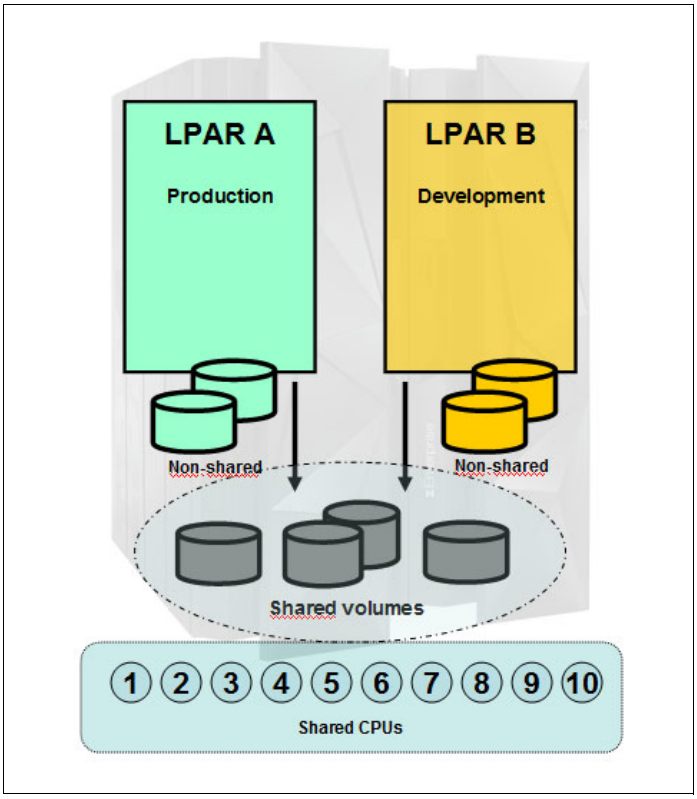


Figure 1-1 Production and development environments

In a shared processor environment, if you define the number of 80 to LPAR A and 20 to LPAR B, it means that LPAR A executes four times more instructions than LPAR B when all systems are running at 100%. This feature allows you to prepare your system and ensure that the development server will not interfere with the production servers. System z was designed to

run at full capacity, so if there were free resources in the system and the development LPAR was requesting more processing, it would be able to use any free IFL resources.

It is also possible to mix dedicated IFLs with shared IFLs in LPARs on the same machine for highly critical system configurations. In Figure 1-2, we show two production environments and one development LPAR environment. The production environment, LPAR A, has a dedicated IFL. LPAR B is also a production LPAR but it is sharing an IFL with development LPAR C.

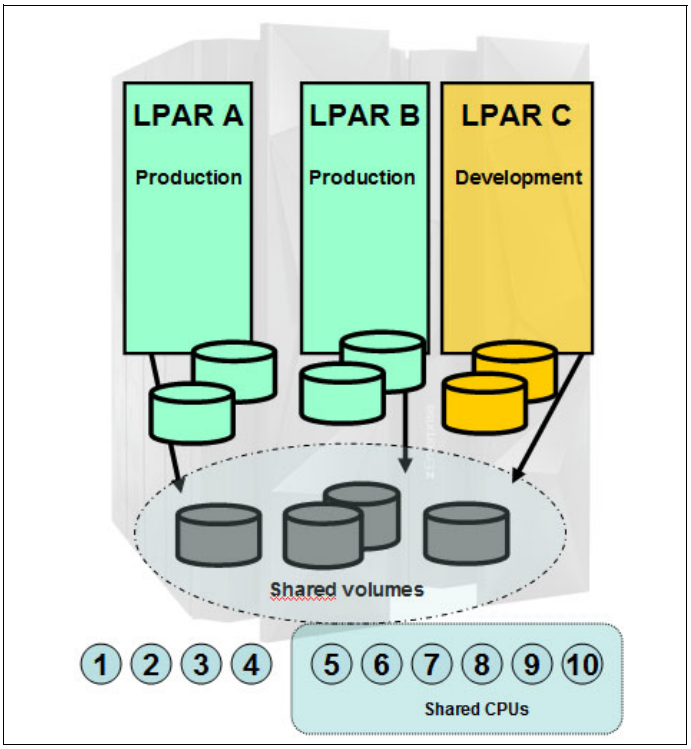


Figure 1-2 Two production LPARs and one development LPAR

This example can be used to allocate Linux on System z database guests in a high availability solution where the active nodes are running in LPAR A and the passive production servers in LPAR B. In this case, all highly critical with high utilization system guests in LPAR A have the benefit of the reduced overhead that exists when the PR/SM must balance the weight and guarantee that all LPARs have their own CPU running time.

Another situation that can benefit from this kind of solution is to have very large database servers running directly on the LPAR. This includes databases that need to be scaled in numbers not possible in other hardware architectures such as x86 or x64 servers. In this case, all hardware cycles are dedicated to the Linux guests and the database application, and also still have the benefits of increasing or decreasing memory and IFLs as needed by application peak load times.

Second-level virtualization is equally efficient and includes features such as sharing and overcommitment of memory allocation as well as the number of virtual processors allocated. The second level of virtualization is managed by the z/VM operating system. z/VM provides a fully virtualized System z environment to run Linux, z/OS, z/TPF, z/VSE, or another z/VM as a guest system.

One single z/VM LPAR can run hundreds to thousands of Linux guests. But in a production environment, it is necessary to understand the peak load times of each server and the utilization of the resources during those periods. You do not want all of your servers to have

the same peak time. This scenario creates a processor queue in z/VM and within the System z hardware. As an example, a 16-IFL LPAR called *LPAR A* has six Linux guest servers (LNX A-LNX F) with a total of 37 virtual processors (vCPU), as shown in the z/VM production LPAR A in Figure 1-3.

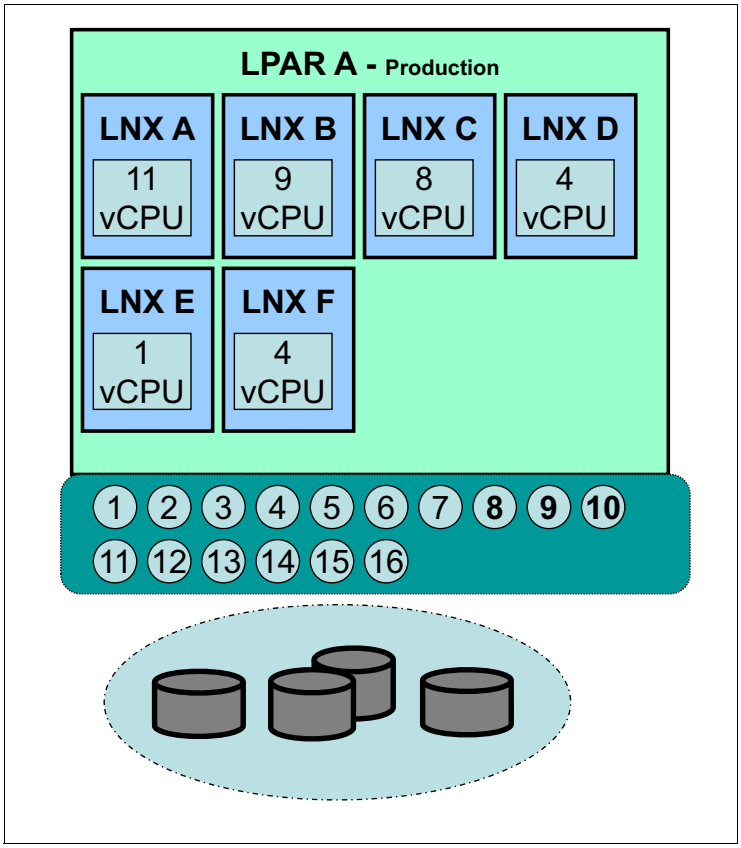


Figure 1-3 z/VM production LPAR A

This scenario is critical and some consideration is needed to take into account the number of virtual processors and real processors. None of the Linux guests have an equal or higher number of virtual processors than the real processor number in the logical partition. It is also important to understand that the peak load time of the two servers should not be the same for a long time. Always monitor the system and see if that situation is going on in your system.

In the processor utilization chart for LPAR A (Figure 1-4), we describe the utilization of z/VM production LPAR A and how it works on a busy day.

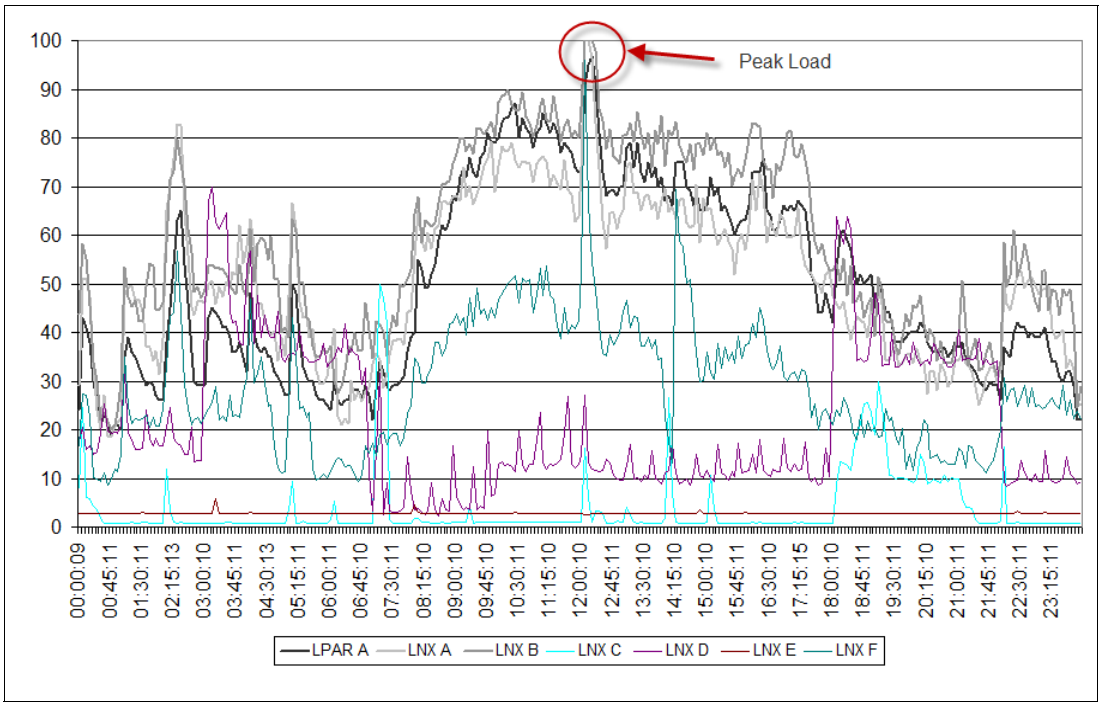


Figure 1-4 Processor utilization chart for LPAR A

This chart demonstrates how well System z works and how well it manages virtualization. For *peak time*, indicated by a red circle and labeled “Peak Load”, about 20 minutes elapsed and the LPAR utilizations ran at almost 100%. The real number was 91 - 97% of the 16 real IFL processors. The LNX A guest used 73% - 100% of its 11 virtual processors and LNX B used 83 - 99% of its nine virtual processors. Although the total number of virtual processors is 28, z/VM and PR/SM are able to scale all requests in 16 real IFLs, but only for a short time. If that interval is extended for a longer period, the Linux guest might have slow response time. Keep in mind, the system will not stop, but the response time is slower.

System peak load time is an important consideration that you should look at when you are planning your production environment. For more information about processor considerations and z/VM scheduling, see Chapter 10, “Performance considerations” on page 123.

Memory allocation is another important topic that is related to virtualization. It is important to understand that on System z, the memory requirement is analyzed in a different way. There are several factors such as z/VM memory allocation and z/VM system paging, as well as the ability to define a certain amount of z/VM memory as virtual disks and use that on the Linux guest as a swap disk. The high performance I/O system of System z hardware allows you to use more of the hardware and decrease the need to cache files and databases systems and should be considered during the development phase of your production environment. It is also possible to change the size of available Linux guest memory dynamically, which provides a dynamic scalability for the applications and systems. More about memory considerations can be found in Chapter 3, “Hardware planning considerations” on page 29.

If you do good planning, setup, and monitoring of your Linux guests and z/VM logical partitions, considering the peak load utilization aspects, I/O channel utilization, and memory definitions, it is possible to allocate twice the size of real memory of logical partitions as virtual memory if you consider the sum of all of the Linux guests’ memory in the LPAR.

1.4 Linux as a System z guest

Linux on System z is the same Linux that runs anywhere else. All the commands such as **ls**, **ps**, **top**, **cp**, and **mv** work the same as in an X86 Linux system. Also, all open source applications such as MySQL, PostgreSQL, Apache Web Server, Nagios Monitoring Service, and all others work the same. Every open source application available for X86 Linux server can be compiled on the System z platform using the same **configure** and **make** commands, and as on any platform, it is necessary to fulfill the dependencies in order to do the software compilation. And as on other Linux platforms, it is possible to take advantage of configuration options when you compile Linux applications from source. In Chapter 7, “Software planning considerations” on page 85 of this book, we provide more information about those options.

1.5 Linux on System z in a cloud environment

Placing a workload on a Linux virtualized environment such as Linux on System z is the first step in running a cloud environment. It provides a shared infrastructure with the z/VM virtualized platform. There are proprietary cloud applications such as IBM SmartCloud® Provisioning, which provides a low-cost, highly scalable option that puts virtual systems into a pool of virtualized hardware running a supported hypervisor such as z/VM, an open source, IBM supported Extreme Cloud Administration Toolkit, or the CSL-WAVE management tool. Learn more about the CSL-WAVE management tool at the following site:

http://www-03.ibm.com/systems/z/news/announcement/csl_announce.html

You can also use z/VM resources to provision servers by using REXX and bash scripts and Linux guest templates, as described in the *The Virtualization Cookbook for z/VM 6.3, RHEL 6.4 and SLES 11 SP3*, SG24-8147.

2

Architectural considerations

This chapter starts by describing some of the architectural issues that should be considered when designing Linux on IBM System z for production. We start by describing reliability, availability, and serviceability (RAS) objectives, all areas of continuous IBM focus. We also describe the importance of using Linux as a guest on z/VM to meet these objectives, versus running Linux on System z natively. Additionally, we examine the advantages of using the z/VM single system image (SSI) over a non-SSI environment.

We next introduce three different scenarios that are most commonly used in the field as examples for production environments as we examine the following planning considerations:

- Two logical partitions (LPARs) on an IBM zEnterprise EC12 (zEC12) with z/VM that can either be SSI or non-SSI
- A four-member SSI cluster that has two LPARs on an IBM z196 with two different recommended service upgrade (RSU) levels and two LPARs on a zEC12
- Small Computer System Interface (SCSI) installations of both z/VM and Linux

We also briefly describe security considerations, backup considerations, and performance considerations and direct you to other IBM Redbooks publications that describe these topics in detail.

© Copyright IBM Corp. 2013. All rights reserved.

2.1 z/VM

IBM System z has a rich heritage of innovation in the area of virtualization and has refined the infrastructure with a coordinated investment in hardware, firmware, hypervisors, and operating systems to enable exceptional qualities of service in the support of virtualization. z/VM is optimized for consolidating workloads on the System z servers, helping clients build an even more cost-effective dynamic infrastructure with exceptional levels of business resilience, speed-to-market, and the flexibility to expand and contract system resources to match business needs.

The z/VM hypervisor is designed to help clients extend the business value of mainframe technology across the enterprise by integrating applications and data while providing exceptional levels of availability, security, and operational ease. z/VM virtualization technology is designed to allow the capability for clients to run hundreds to thousands of Linux servers on a single mainframe or as a large-scale Linux-only enterprise server solution.

Many IBM System z clients have found that the combination of Linux on System z and z/VM not only offers server consolidation savings, but also enables the flexible deployment of business-critical enterprise solutions on System z servers that are configured with Integrated Facility for Linux (IFL) specialty engines.

z/VM virtualization technology is designed to provide the capability for clients to run hundreds or thousands of Linux servers as z/VM guests on a single mainframe that hosts other System z Operating Systems for non-Linux workloads, such as z/OS, z/VSE, and z/TPF, on the same System z server or as a large-scale Linux-only enterprise-server solution. See Figure 2-1 on page 11.

The advanced virtualization technologies available with the z/VM hypervisor combined with the highly attractive economics on the highly secure and reliable System z servers help clients extend the business value of mainframe technology across the enterprise by integrating applications and data while providing exceptional levels of availability, security, and operational ease.

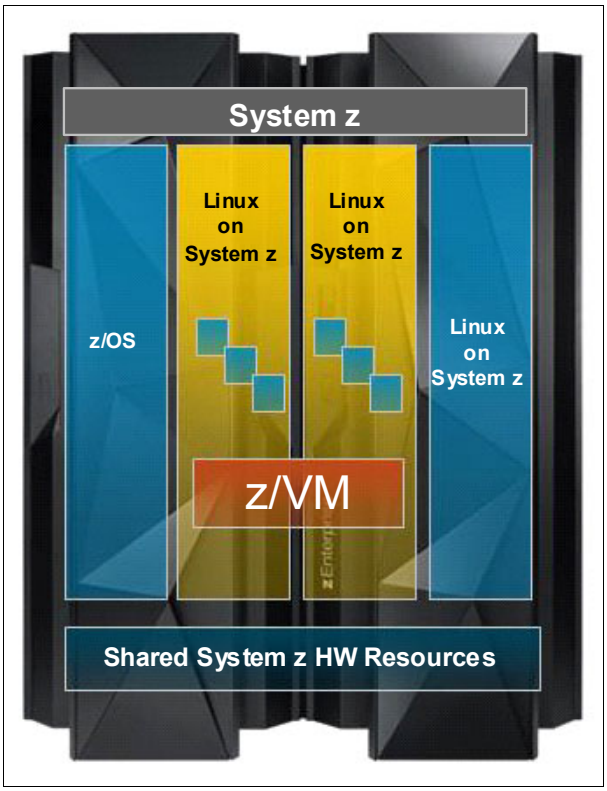


Figure 2-1 System z hosting multiple operating systems

2.1.1 z/VM single system image

The z/VM 6.2 announcement introduced new features and enhancements. This release provides a multi-system virtualization clustering technology, which allows up to four z/VM instances to be clustered in a single system image (SSI). This configuration is illustrated in Figure 2-2 on page 12. This feature allows *Live Guest Relocation* (LGR) of Linux guests from one member of the cluster to another.

The following new features and enhancements were introduced:

- Simplified z/VM Systems Management
  - Relief from the challenges that are associated with virtual machine sprawl on competitive systems
  - Simplifies system management of a multi-z/VM environment
  - Concurrent installation of multiple-system cluster
  - Single maintenance stream
- Share system resources with high resource utilization
  - A more manageable infrastructure for cloud computing by providing a set of shared resources that can be managed as a single resource pool
  - Enhanced workload balancing with the added ability to move work to available system resources

- Non-disruptive maintenance and upgrades
  - Live Guest Relocation to move Linux virtual servers without disruption to the business, helping to avoid planned outages for z/VM and hardware maintenance
  - Facilitates horizontal growth of z/VM workloads

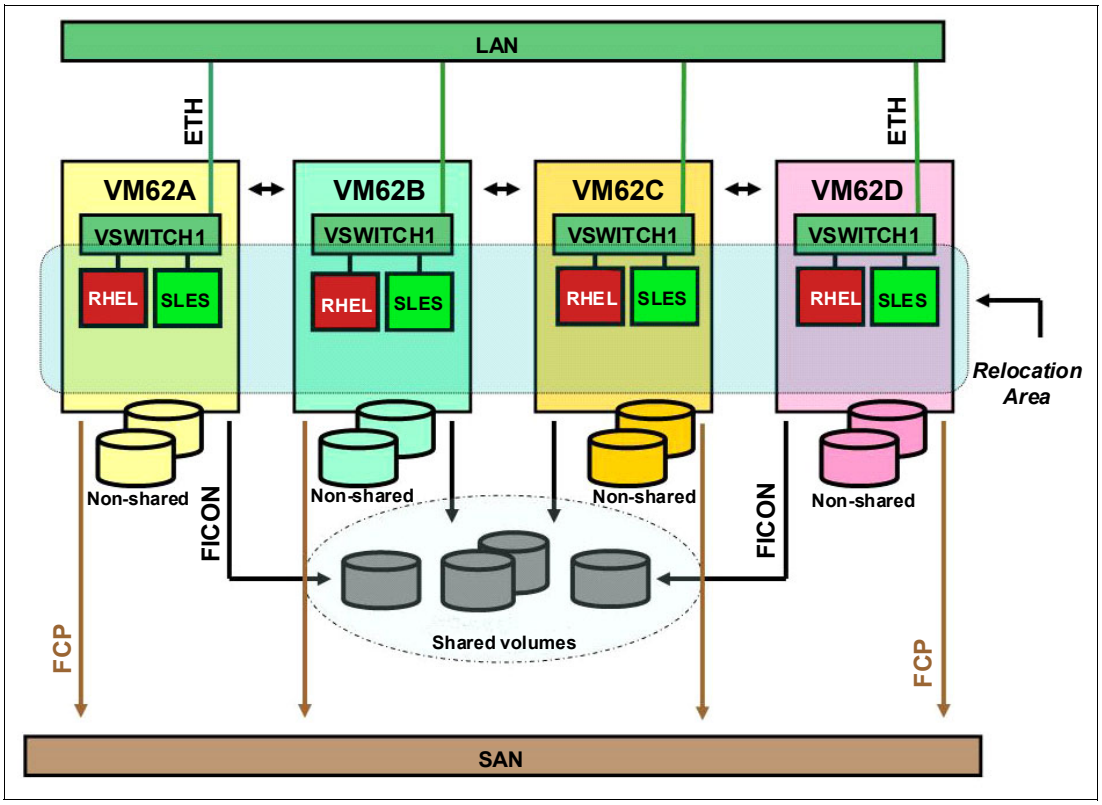


Figure 2-2 z/VM single system image

Simplified z/VM Systems Management

When z/VM multiple system virtualization with SSI is used to create clusters, the members can be serviced, managed, and administered as though they are one integrated system. The coordination of members joining and leaving the cluster, the maintenance of a common view of cluster member and resource states, and the negotiated access to shared cluster resources are all done seamlessly.

z/VM multi-system virtualization helps clients avoid virtual machine sprawl challenges. These challenges include the creation of uncontrolled numbers of virtual machines that IT managers do not know about, whether they are up or down, and whether they are secure.

Shared system resources with high resource utilization

Sharing all system resources with high levels of resource utilization is extended with z/VM V6.2. Within SSI, resources that are used by the z/VM hypervisors and the virtual machines are shared. The shared resources are managed as a single resource pool and provide a more manageable infrastructure. Resources include the user directories, minidisks, spool files, network device Media Access Control (MAC) addresses, and security definitions, if implemented.

Sharing resources among members improves the integrity and performance of the system. Through resource sharing, virtual servers have access to the same devices and networks, regardless of which z/VM member they are logged on to within the SSI cluster. Sharing

resources allows service to be rolled out to each member of the cluster on individual schedules, avoiding an outage for the entire cluster.

High levels of resource sharing within z/VM include the sharing of Linux program executables and file systems.

Non-disruptive maintenance and upgrades

LGR is a process where a running Linux virtual guest can be relocated from one z/VM member system in an SSI cluster to another member. Guests can be moved to other members that are on the same or separate System z servers without disruption to the business. Virtual Linux guests can even be moved across the System z family between IBM System z10® EC, IBM System z10 BC, IBM z196, IBM z114, and IBM zEC12. This flexible manual workload balancing allows work to be moved non-disruptively to available system resources. The business benefit is the reduction of the impact of planned z/VM outages when performing updates to the z/VM software, hardware maintenance, or upgrades. This reduced impact delivers application continuity is an important factor contributing to an optimized system.

Note: z/VM SSI is an optional, priced feature. In a non-SSI environment, the preceding z/VM clustering features are not available. If a non-SSI environment is selected, you are not able to take advantage of the SSI features.

2.2 Overview of architectures used in this book

Based on our client experiences, we developed typical scenarios that address the most common client requirements. These scenarios are based on the following common characteristics:

- Hardware availability
- Scalability
- Shared resources
- Disk management
- Network management
- z/VM maintenance

In this section, we describe the three scenarios that will help you to select the best path for your business needs:

- Two z/VM LPARs (SSI) on a single System z central processor complex (CPC)
- Four z/VM LPARs (SSI) on two to four System z CPCs
- Small Computer System Interface (SCSI)-only (non-SSI) solutions

2.2.1 Scenario 1: Two z/VM LPARs (SSI) on a single System z CPC

Many System z clients around the world have a System z CPC that supports their core business applications because of the RAS of the mainframe platform. z/VM is a hypervisor that holds all Linux images in a logical partition (LPAR). To leverage System z hardware and firmware availability (such as Processor Resource/Systems Manager (PR/SM)), implement a z/VM SSI cluster that uses PR/SM LPARs.

This server consolidation scenario implements two z/VM LPARs that use SSI under a single System z CPC that addresses a number of Linux on System z implementations. This configuration is shown in Figure 2-3.

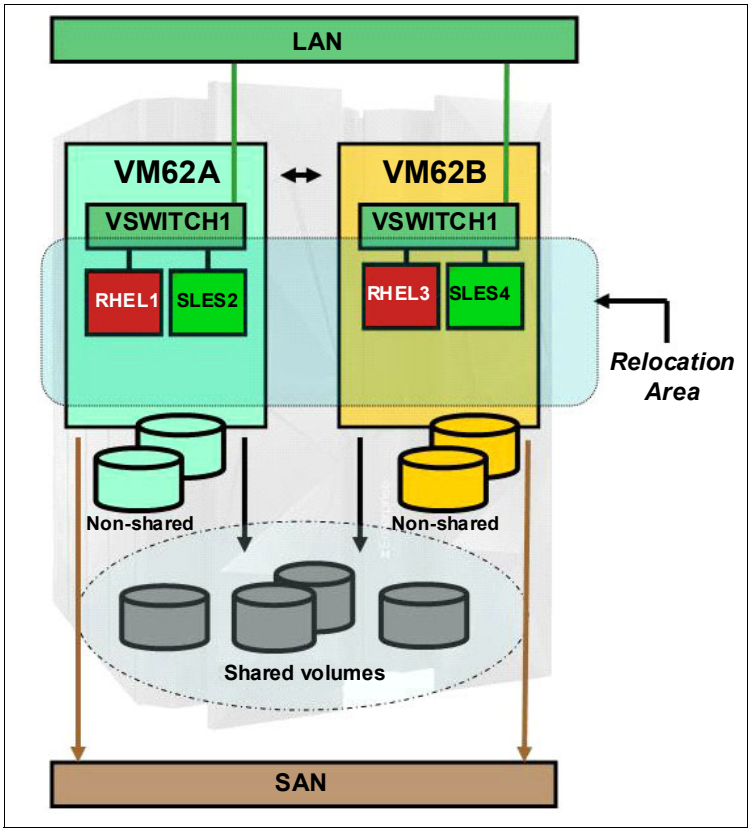


Figure 2-3 Two z/VM LPARs (SSI) on a single System z server

In this scenario, we created a cluster of two z/VM members. Each member of this cluster shares System z resources such as processors, cache, channels, disk, network, and cryptographic cards.

It also provides the flexibility to move Linux virtual servers (guests) from one to another z/VM member without application interruption and all other z/VM SSI benefits that were described in section 2.1.1, “z/VM single system image” on page 11.

Hardware availability

In this scenario, both z/VM LPARs are hosted by a single System z CPC. The System z architecture has redundancy in many areas that are built into the hardware to keep the environment up and running if there happens to be a hardware problem. Support for hot-pluggable hardware part replacements, such as the IFL processors, memory, and I/O cards is also available. For more information about System z architecture availability, see the IBM Redbooks publication, *IBM zEnterprise EC12 Technical Guide*, SG24-8049.

Scalability

The System z server has tremendous scalability. It has up to 101 central processors, up to 3 terabytes (TBs) main storage, and up to 320 I/O channels[1]. Support for on-demand processors and storage is also available. A single System z zEnterprise server has enough scalability to address general client requirements.

[1] Machine Type/Model *2827-HA1*

Shared resources

In this scenario, both z/VM LPARs share System z CPC resources, except for main storage. Linux guest servers in two z/VM LPARs can share those resources, thus increasing the server utilization.

When a workload is moved from one z/VM member to another, the overall IFL and channel utilization remains the same. The result of this architecture is better efficiency.

Figure 2-4 shows an example of this efficiency. In this example, z/VM LPAR A and z/VM LPAR B share the resources of a single System z CPC, which is 80% occupied. Consider a need to shut down z/VM LPAR A to apply maintenance. All Linux guests hosted by this LPAR are moved to z/VM LPAR B. The overall environment utilization remains at 80% during this maintenance. After maintenance, when z/VM LPAR A is back online, Linux guests can be moved back to their original LPAR.

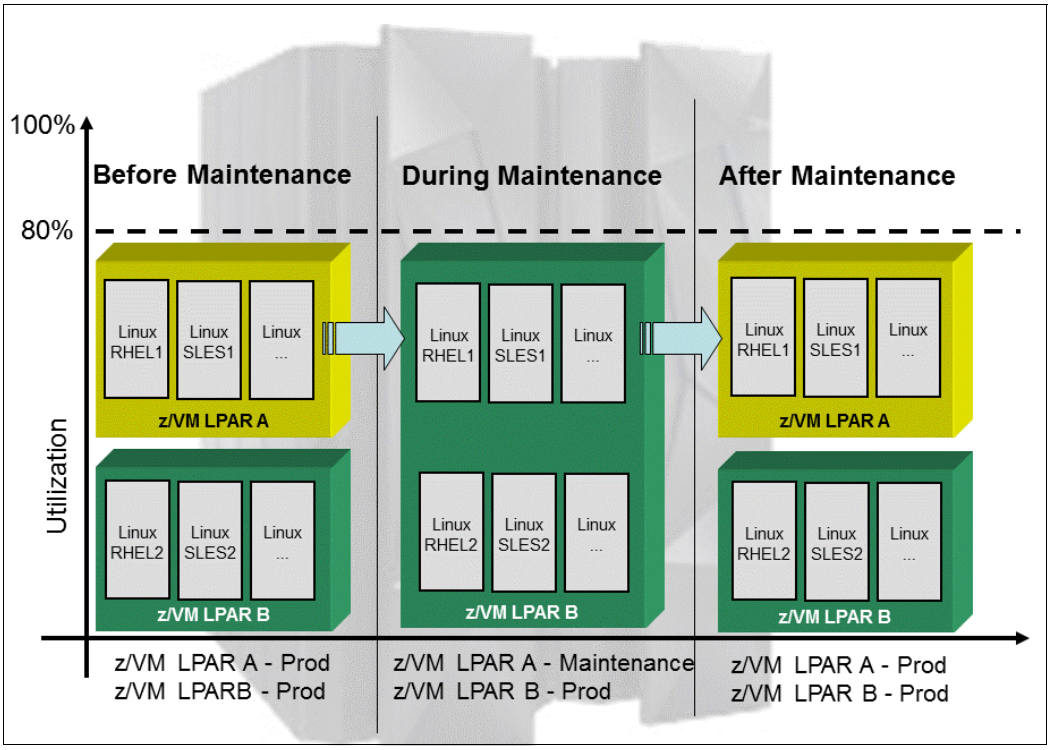


Figure 2-4 Two z/VM LPARs (SSI) sharing resources on a single System z server

z/VM SSI supports up to four members in the same cluster. Those additional members can be used for QA or development purposes and can share the System z CPC resources. Each LPAR must have enough main storage to support the z/VM host and guest Linux systems.

Disk management

When installing an SSI environment, it must be installed on shared IBM extended count key data (ECKD™) devices for consistency. Each z/VM member has shared and dedicated devices, as shown in Figure 2-5.

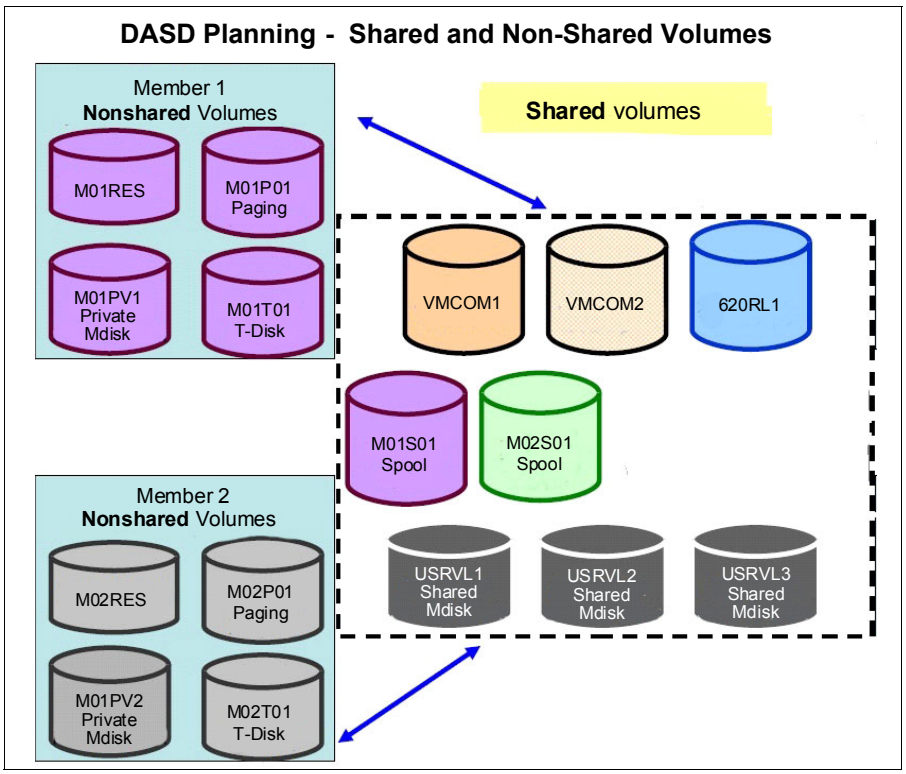


Figure 2-5 DASD Planning: Shared and Non-Shared Volumes for a two member z/VM SSI cluster

z/VM SSI supports ECKD disk management between z/VM members. By default, z/VM does not allow concurrent ECKD read/write (R/W) access to the same direct access storage device (DASD) area. This also holds true for the SSI environment.

Linux guests can use ECKD or SCSI devices, or both. ECKD devices are available in this scenario, as shown in Figure 2-6 on page 17. For better management of devices, install Linux guests on ECKD devices. For better performance, use a storage area network (SAN) and SCSI disks for those Linux applications that demand high input/output (I/O) rates or have a large storage requirement, or both.

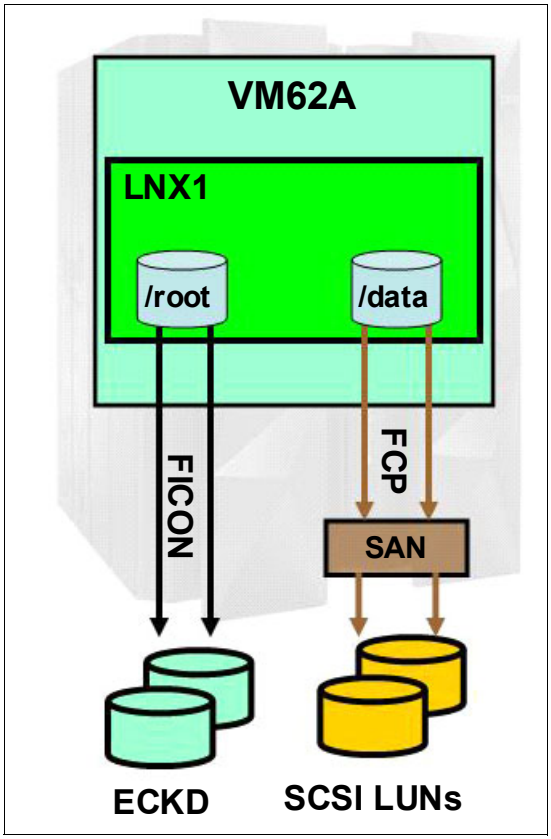


Figure 2-6 Combination of ECKD and Fibre Channel Protocol (FCP) in the same environment

Network management

Typically, System z virtualization provides four widely used networking options:

- IBM HiperSockets
- Guest LANs
- Virtual switches (VSWITCHs)
- Direct attached Open Systems Adapters (OSAs)

These options give Linux on System z guests the ability to communicate over the network. In any scenario, there is always the option to use direct attached OSA, where a Linux guest uses all OSA features as its network interface card (NIC). IBM HiperSockets is a good option to connect LPARs on the same server. z/VM provides Guest LANs and VSWITCHs, where Linux guests use virtual devices as their own physical network adapters. For complex environments that require outside LAN communication, use virtual switches. A virtual switch allows grouping of several OSA-Express devices to create one logical link for providing fault-tolerance and high-speed connections between the physical OSA devices and the Linux guests, as shown in Figure 5-3 on page 73.

In general, decisions regarding the best methods for networking are based on reliability, performance, availability, security, and management. In this scenario, we cover the most commonly preferred method, that is, *virtual switches*.

Figure 2-7 on page 18 shows two OSA adapters that are being shared by two z/VM LPARs. This basic network architecture provides failover capabilities with VSWITCH implementation. Each virtual switch uses one OSA adapter for all network traffic and the other for failover.

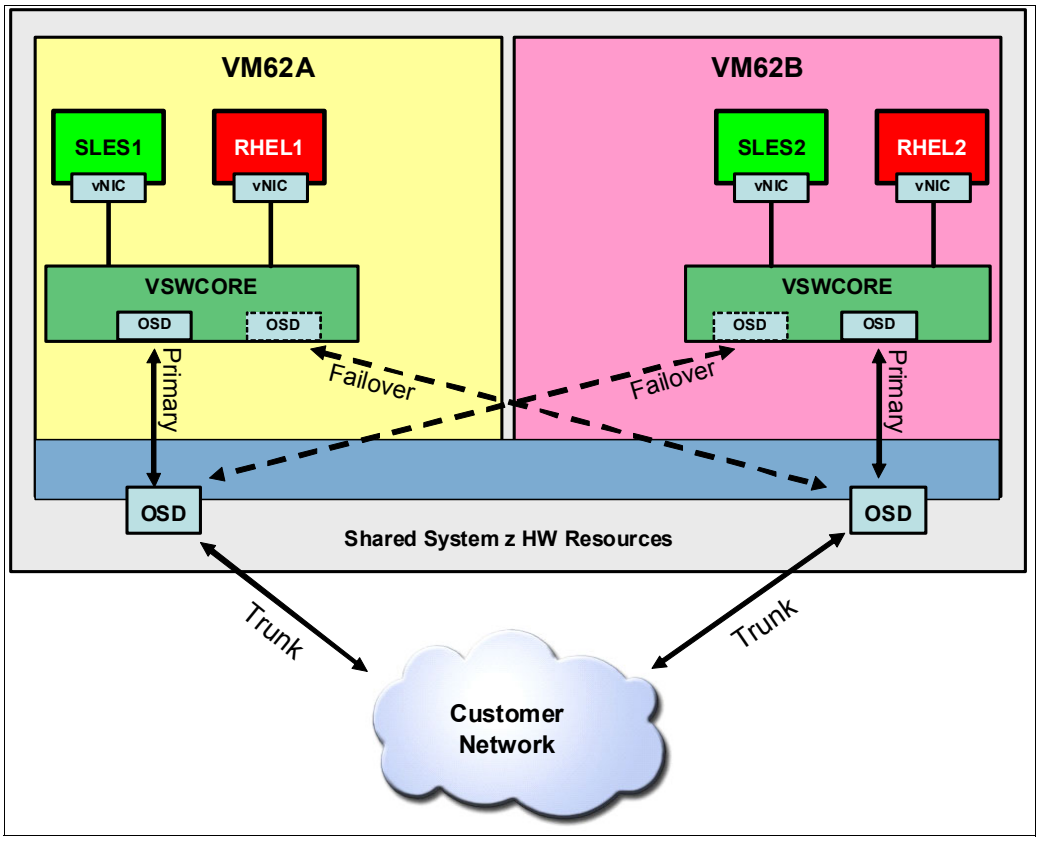


Figure 2-7 z/VM VSWITCH network

The network configuration changes according to the architecture. The use of trunk ports allows the exploitation of Level 2 features. This capability provides the ability to choose PortType Access or PortType Trunk for each Linux virtual network interface card (vNIC).

More details about this topic can be found in Chapter 5, “Network planning considerations” on page 67.

z/VM maintenance

Usually a new z/VM recommended service upgrade (RSU) is available every 3 - 6 months. Keep your z/VM environment up-to-date by applying the latest available service at least every 6 months.

Planned software or hardware maintenance can be applied to a single member without affecting the other members of the cluster or disrupting running Linux guest systems. The Linux guests can be relocated to another member in the cluster before maintenance is applied to the original member that hosted the Linux guest.

System management is simplified in an SSI cluster. Service can be applied to z/VM from any member within the cluster, and while it needs to be put into production on each member, it needs to be applied only *once* from any member. This provides the ability to plan when service upgrades are put into production.

Summary

The two z/VM LPARs (SSI) on a single System z CPC scenario allows implementation of a scalable Linux on System z virtualization solution that uses System z high availability technology, providing a very flexible solution for business needs.

Using VSWITCH as part of the network infrastructure provides the most efficient network management approach in this scenario.

2.2.2 Scenario 2: Four z/VM LPARs (SSI) on two to four System z CPCs

Some clients select multiple servers for different purposes. An example of this configuration can be seen in Figure 2-8. Some of the reasons for this architecture include:

- Government regulations, certifications, and compliancies
- Company IT infrastructure policies and requirements
- High availability
- High scalability
- Service and upgrade flexibility

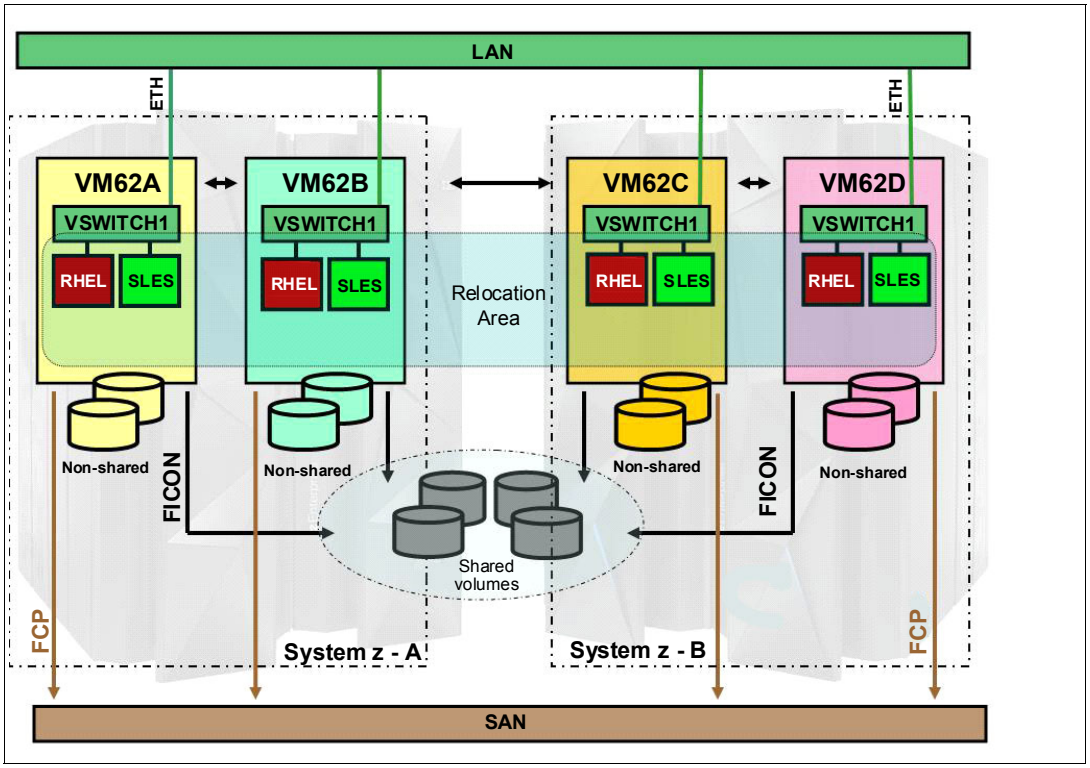


Figure 2-8 Four z/VM LPARs (SSI) on two to four System z servers

Hardware availability

In this scenario, we have two System z CPCs hosting two to four z/VM members participating in an SSI environment. This architecture addresses those cases where more than one server is a requirement for government regulations, certifications, compliancies, or company IT infrastructure policies.

Scalability

This scenario has much more scalability than the scenario described in section 2.2.1, “Scenario 1: Two z/VM LPARs (SSI) on a single System z CPC” on page 13 because of the addition of another CPC. Each one provides the scalability of up to 101 central processors, up to 3 TBs main storage, and up to 320 I/O channels.

Shared resources

In this scenario, both servers are sharing resources. Two z/VM members share resources from a single CPC. z/VM SSI provides the capability to move all Linux guests from one System z member to another, whether or not it is on the same or different CPCs. Both System z CPCs have to be configured to support all cluster workload, which means that both have to have available capacity.

Figure 2-9 shows an example of z/VM LPAR A and LPAR B sharing resources from one System z CPC that is 40% occupied, and z/VM LPAR C and LPAR D sharing resources from another one that is also 40% occupied. If there is a need to shut down one LPAR, all Linux servers hosted by this member have to be moved to another SSI member. What this means is that both System z CPCs need to have enough resources to support the workload of the entire environment.

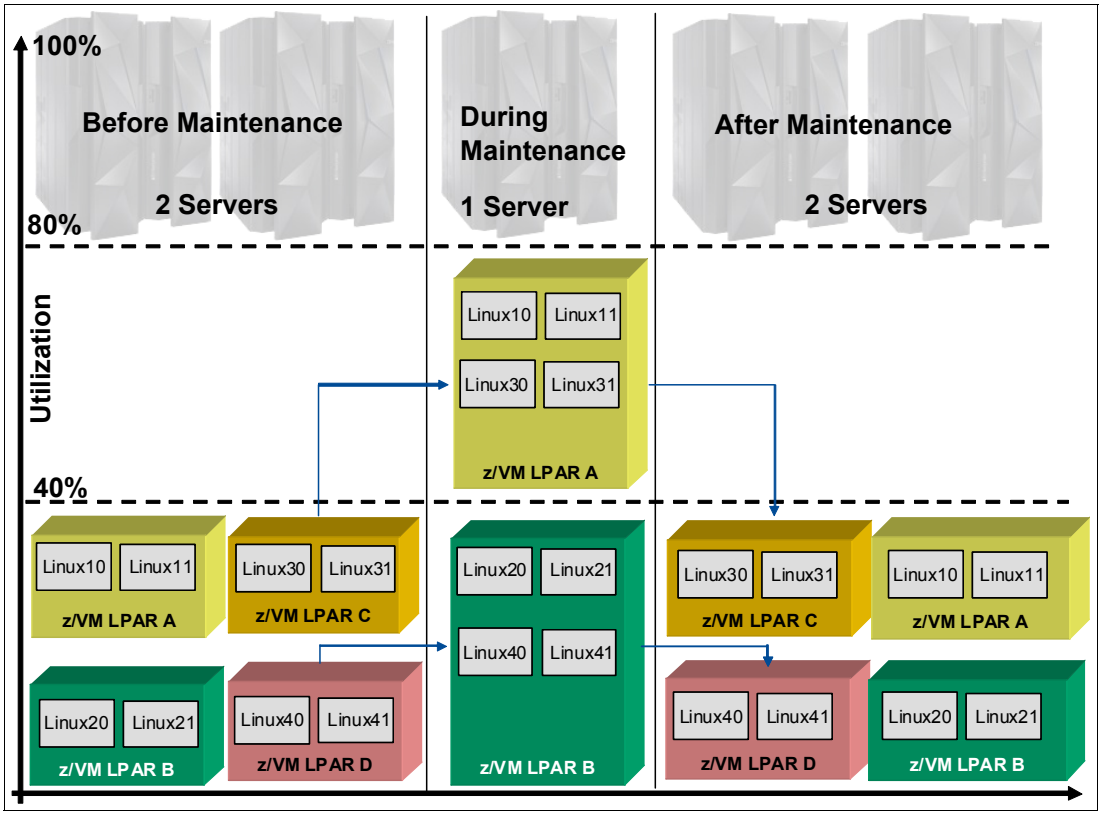


Figure 2-9 Four z/VM LPARs (SSI) sharing resources on two to four System z cPCs

Disk management

Because this scenario is built in an SSI cluster, ECKD devices are also required. Each z/VM member has shared and dedicated devices, as shown in Figure 2-10. Notice that the z/VM SSI members on two different CPCs have access to the same storage devices.

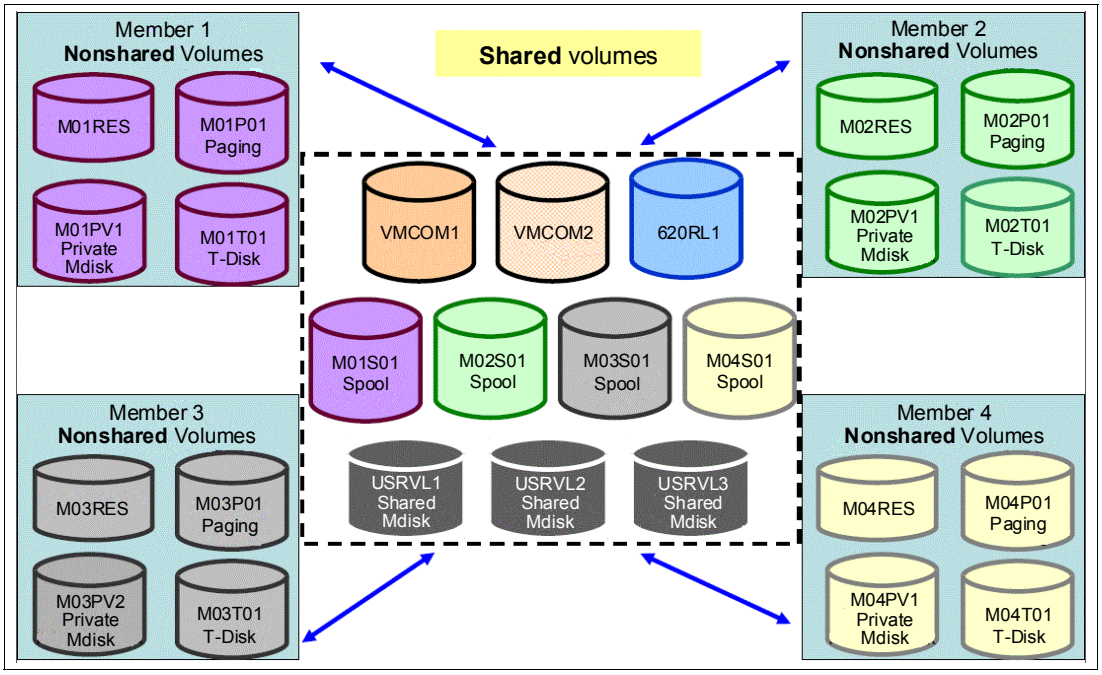


Figure 2-10 DASD Planning: Shared and Non-Shared Volumes for z/VM SSI members

Linux guests can use ECKD or SCSI devices, or both. If a decision is made to use SCSI disks, both System z CPCs must have access to the same SAN fabric.

Network management

In this scenario, we applied the same concepts that are described in the previous scenario. Both System z CPCs must have access to the same LAN segments. Figure 2-11 shows the network connection between the two CPCs.

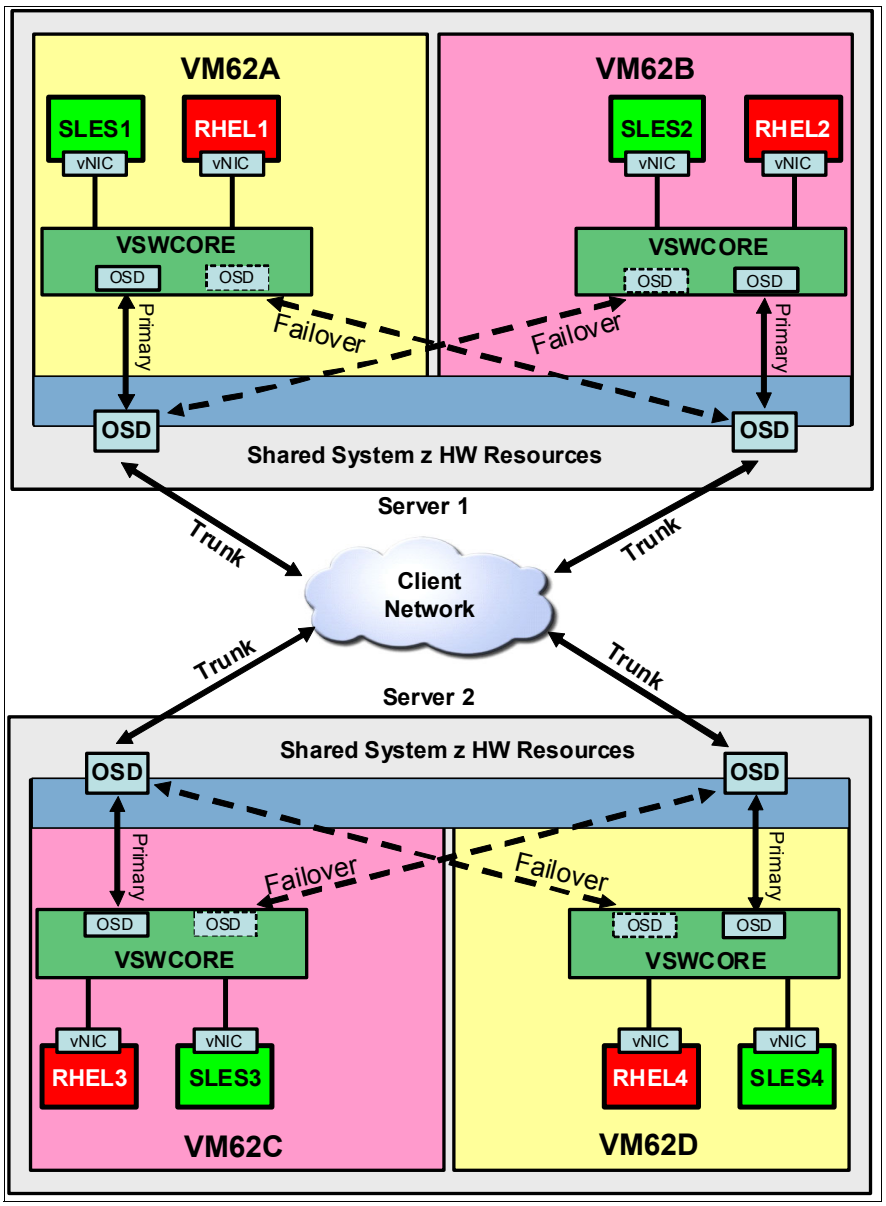


Figure 2-11 Network in a two-CPC environment

In some cases, HiperSockets are adopted to implement a high-speed TCP/IP connection between LPARs on a System z CPC. This implementation eliminates the need for any physical cabling or external networking connection between guests that run in different LPARs on the same System z CPC.

A Linux guest that is connected via a HiperSocket cannot be moved from one CPC to another because this HiperSocket is not available in the targeted CPC. To address this issue, z/VM v6.2 implements a new feature called *virtual switch HiperSockets bridge*. This z/VM virtual switch is enhanced to transparently bridge a guest virtual machine network connection on a

HiperSockets LAN segment. This bridge allows a single HiperSockets guest virtual machine network connection to also directly communicate with the following components:

- Other guest virtual machines on the virtual switch
- External network hosts through the virtual switch OSA UPLINK port

z/VM maintenance

Maintenance in this type of architecture is the same as described in section 2.2.1, “Scenario 1: Two z/VM LPARs (SSI) on a single System z CPC” on page 13.

Summary

In this scenario, we selected four z/VM LPARs (SSI) on two to four System z CPCs. It offers the clients the same benefits as in section 2.2.1, “Scenario 1: Two z/VM LPARs (SSI) on a single System z CPC” on page 13 as well as additional scalability via leveraging multiple System z CPCs, which increases availability.

Moreover, in addition to the VSWITCH network infrastructure, we described the new z/VM network feature, virtual switch HiperSockets bridge.

2.2.3 Scenario 3: SCSI-only (non-SSI) solutions

From a worldwide perspective, some Linux installations prefer to use their existing SAN network. In these cases, some choose z/VM as a hypervisor for their Linux on System z guests. Other clients prefer to install Linux on native LPARs, or even combine the two options for different purposes. We describe the two approaches in this section. Figure 2-12 illustrates the SCSI-only solutions.

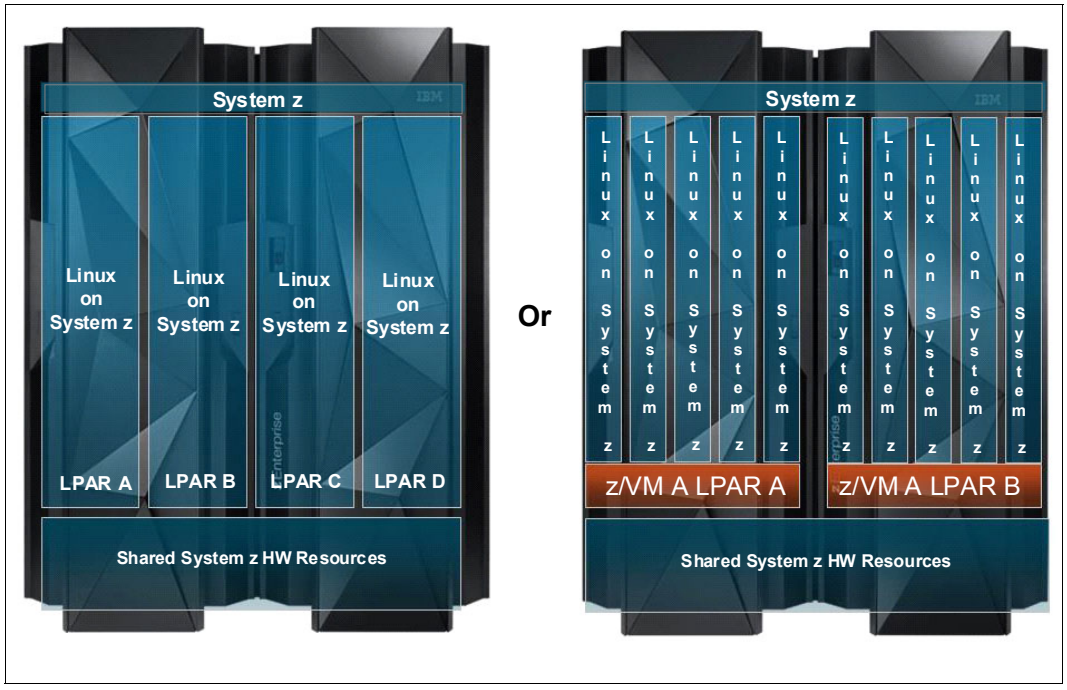


Figure 2-12 SCSI-only solutions

Note: For a SCSI-only solution, the z/VM SSI feature is not available. ECKD devices are mandatory for SSI installations.

Hardware availability and scalability

From the hardware availability and scalability perspective, it is completely the same as described in section 2.2.1, “Scenario 1: Two z/VM LPARs (SSI) on a single System z CPC” on page 13, and 2.2.2, “Scenario 2: Four z/VM LPARs (SSI) on two to four System z CPCs” on page 19.

Shared resources

When using z/VM as a hypervisor, Linux guests share resources within the System z CPC. However, Linux guests cannot be moved from one z/VM LPAR to another without interruption in a non-SSI environment.

If there is a Linux on native LPAR solution, resource sharing is limited.

Disk management

From the z/VM perspective, each SCSI logical unit number (LUN) has to be defined as an emulated device (EDEV) to be managed by z/VM. Linux guests are able to use FCP fabric as usual.

Figure 2-13 demonstrates how z/VM and Linux volumes appear in a typical z/VM SCSI-only installation. We show the SAN and its associated LUNs. For more technical information about this SCSI-only installation, see Chapter 5 in the *z/VM V6R2 Installation Guide,* GC24-6246-00.

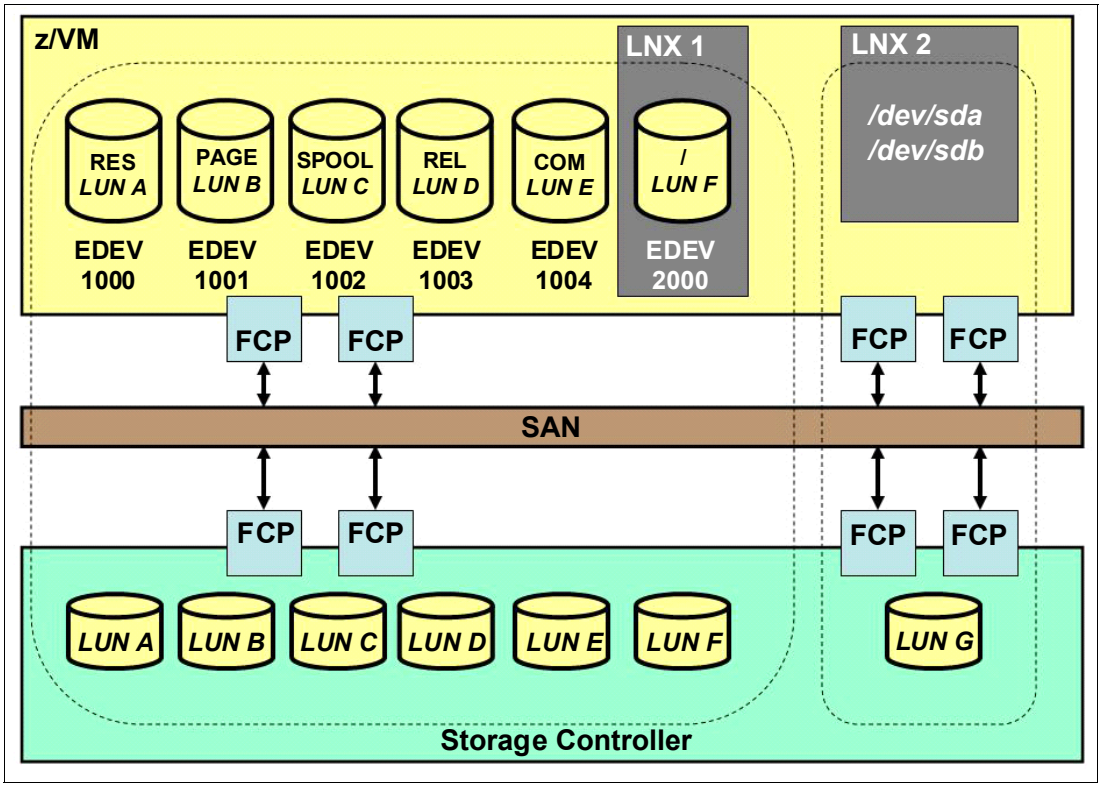


Figure 2-13 z/VM SCSI-only architecture

From the Linux perspective, SCSI disk management is the same as in previous scenarios. Figure 2-14 shows how a typical native Linux SCSI-only installation appears. We show the SAN and its associated LUNs. For more technical information about this SCSI-only installation, see section 9.6 in the Red Hat Enterprise Linux 6 Installation Guide, which is available at the Red Hat website:

https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Installation_Guide/Storage_Devices-x86.html

Also, see the SUSE Linux Enterprise Server 11 Deployment Guide, which is available at the SUSE website:

https://www.suse.com/documentation/sles11/singlehtml/book_sle_deployment/book_sle_deployment.html

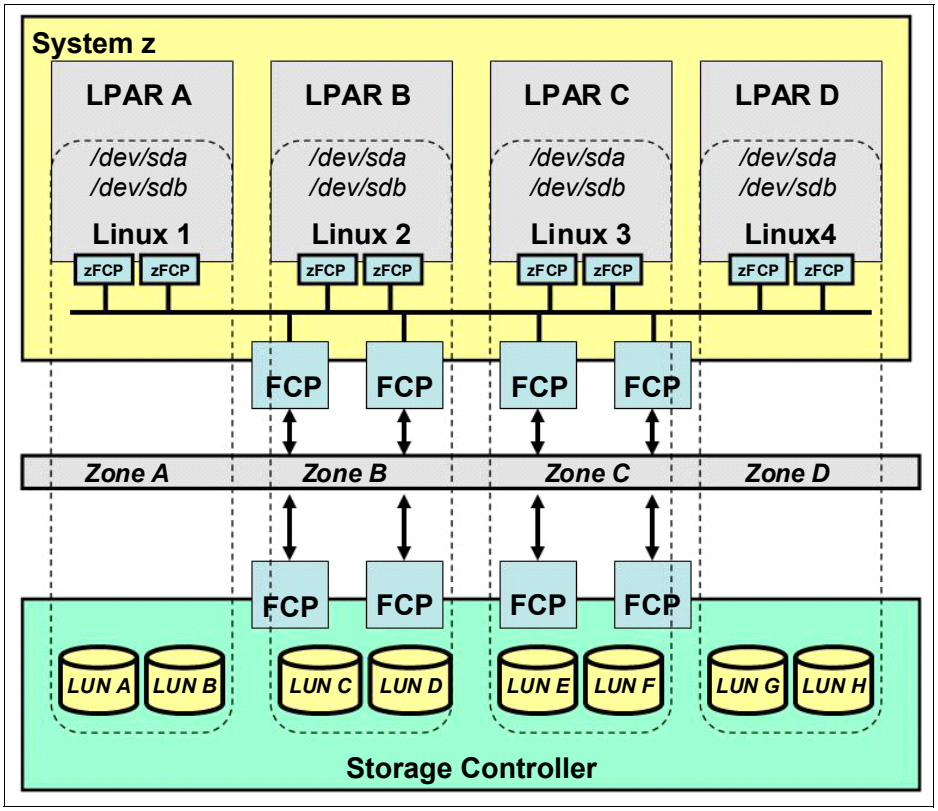


Figure 2-14 Native Linux architecture using SCSI

Network management

From a z/VM perspective, use the VSWITCH as explained in the two previous scenarios.

From a native Linux perspective, VSWITCH is not available and Linux uses the OSA cards as its network interface card (NIC). For more technical information about how this configuration can be set up, see the publication, *Device Drivers, Features, and Commands on Red Hat Enterprise Linux 6.4*, SC34-2597-04:

http://public.dhe.ibm.com/software/dw/linux390/docu/lfu4dd04.pdf

Or, refer to the publication, *Device Drivers, Features, and Commands on SUSE Linux Enterprise Server 11 SP2*, SC34-2595-02:

http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss?CTY=US&FNC=SRX&PBL=SC34-2595-02

Figure 2-15 shows an example of network connectivity for a native Linux on System z installation.

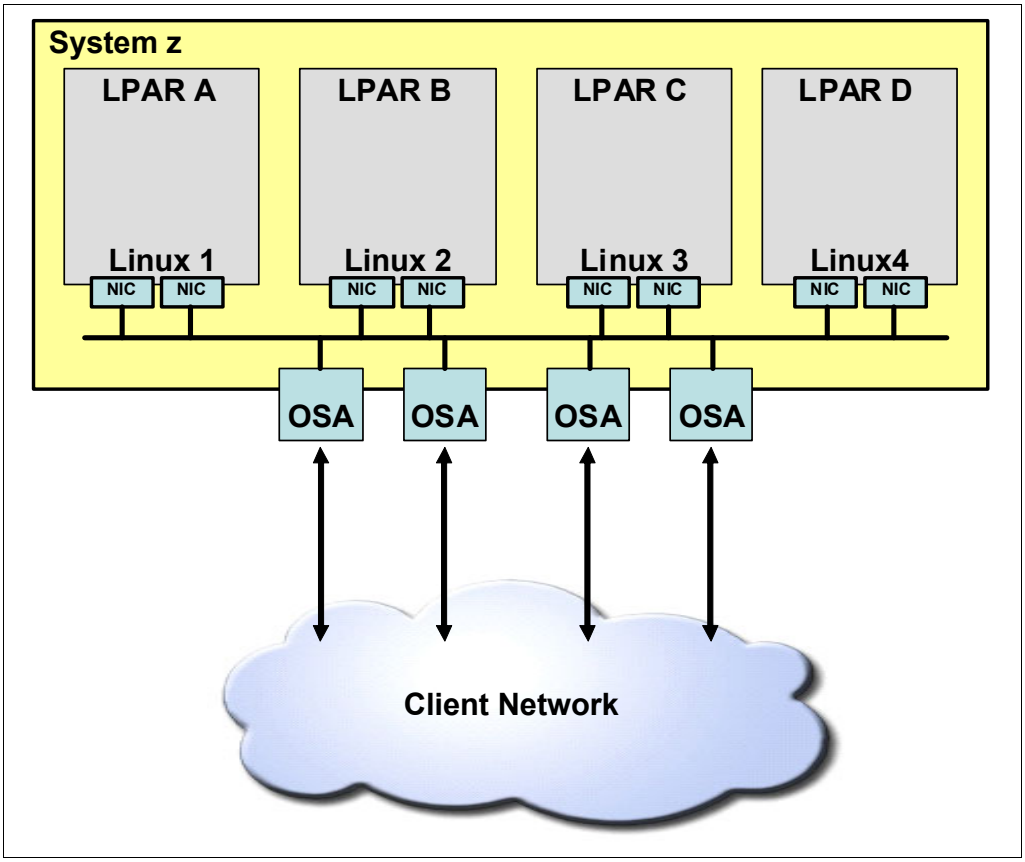


Figure 2-15 Native Linux on System z network connectivity

z/VM maintenance

The SSI feature is not available for SCSI-only installations with z/VM. What this means is that the method of applying maintenance is disruptive. You have to schedule planned outages, because without SSI, it is not possible to perform Linux live guest relocation from one LPAR to another.

For the Linux SCSI-only installations on a native LPAR, z/VM maintenance does not apply.

Summary

The Linux on System z platform supports SCSI-only environments. This solution is an option for those cases where ECKD devices are not available. If the SCSI z/VM method is selected, the lack of the SSI feature results in planned outages to apply maintenance. If a native Linux on System z solution is chosen, it has limited virtualization and flexibility.

2.2.4 Summary

We compared and assessed the described scenarios and created Table 2-1 where, based on our experiences, we rated each scenario as *very good*, *good*, *fair*, or *poor*, based on various criteria.

Table 2-1 Scenario assessments

	Hardware availability	Scalability	Shared resources	Disk management	Network management	z/VM maintenance
Scenario 1	Good	Good	Very good	Good	Good	Good
Scenario 2	Very good	Very good	Good	Good	Good	Very good
Scenario 3: z/VM	Good	Good	Very good	Fair	Good	Poor
Scenario 3: Native	Good	Good	Poor	Fair	Fair	N/A

Scenario 1: Two z/VM LPARs (SSI) on a single System z CPC

Widely used in most client environments.

Scenario 2: Four z/VM LPARs (SSI) on two to four System z CPCs

For large environments that require more than one CPC.

Scenario 3: SCSI-only (non-SSI) solutions

Used a scenario for cases where ECKD devices are not available.

3

Hardware planning considerations

This chapter describes the hardware considerations that must be taken in to account when moving from a Test/Dev environment to a production environment. We describe the memory management from a Linux and z/VM perspective as well as the characteristics. We illustrate the adding of memory to a Linux guest and the steps necessary to expand memory dynamically. Fibre Channel connection (FICON) and Fibre Channel Protocol (FCP) channel considerations are described and where we would select the guest that would best use the channel considerations.

The following topics are covered:

- Linux memory
- z/VM memory
- FICON and FCP channel considerations

© Copyright IBM Corp. 2013. All rights reserved.

3.1 Memory planning for Linux on System z guests

Memory planning is always a tricky topic because it usually results in the answer: *It depends*. Yes, it depends on which type of application is used, how large it is, and mainly, its expected usage.

Linux tends to consume all available memory and use it as cached memory, which means that it does not matter if you have 1, 2, or 4 GB. The kernel takes it all and manages it accordingly with the software or application requirements. "*The goal is to reduce the number of disk accesses because even the fastest disk devices are slower than memory access*." (*Linux on IBM System z: Performance Measurement and Tuning*, SG24-6926, chapter 5, section 5.4.2).

A good rule of thumb is to allocate memory on a "just enough" basis for each Linux server. This is because either within z/VM or the LPAR, there will always be the dynamic extension feature, which is known as *hotplug memory*, which basically extends memory size with systems and applications online.

Note: Keep in mind that after a reboot, the amount of memory is back to default. If you want to set the new amount as default, update the guest's user directory file on z/VM or the logical partition (LPAR).

If you intend to use this feature, set it in the user directory on z/VM and reboot the guest. Add this feature when you build the guest or if the server has been built already and you are moving it from development to production.

To enable it in the user directory, use the following steps:

- Set stand-by memory in z/VM:
  a. Get the guest user directory from DIRMAINT using the following command. Replace **itsolnx3** with the name of your Linux guest:

     ```
     dirm for itsolnx3 get
     ```

  b. Receive the file sent from DIRMAINT and open it for editing, as shown in Figure 3-1.

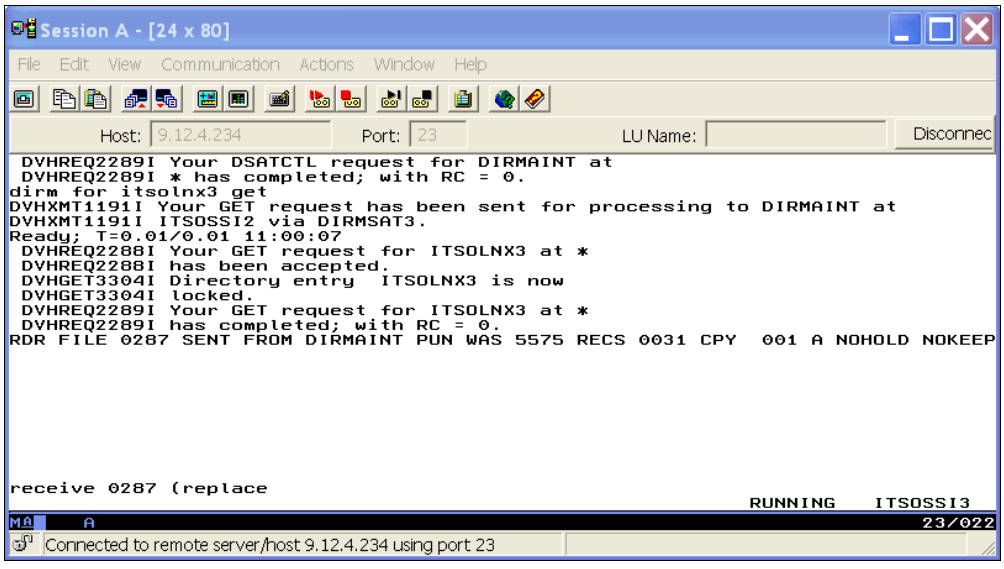


Figure 3-1 Receive the file

c. Edit the file using the command that is shown in Figure 3-2.

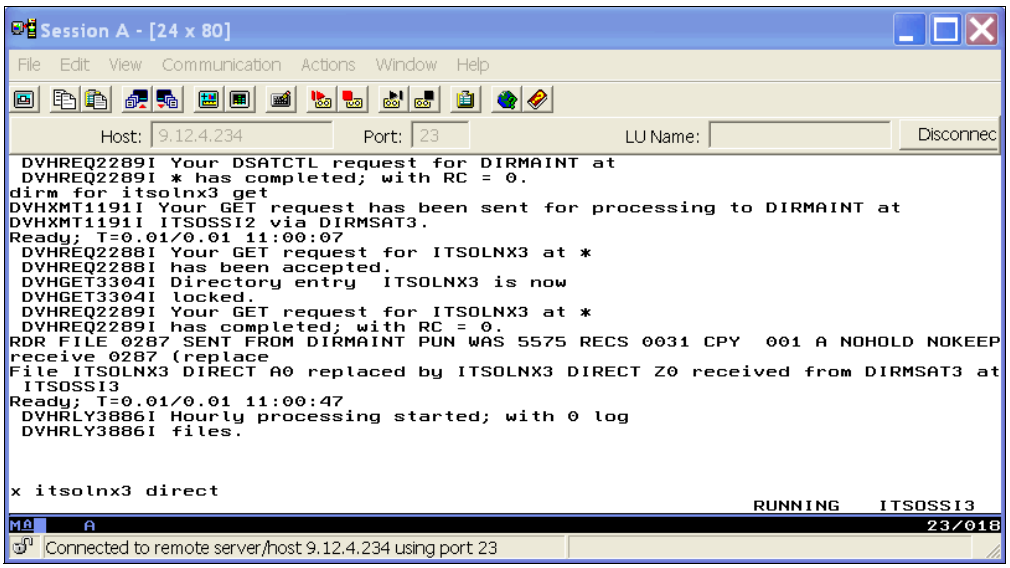


Figure 3-2 Edit the file

a. Add the **define storage** command to the file, as shown in Figure 3-3.

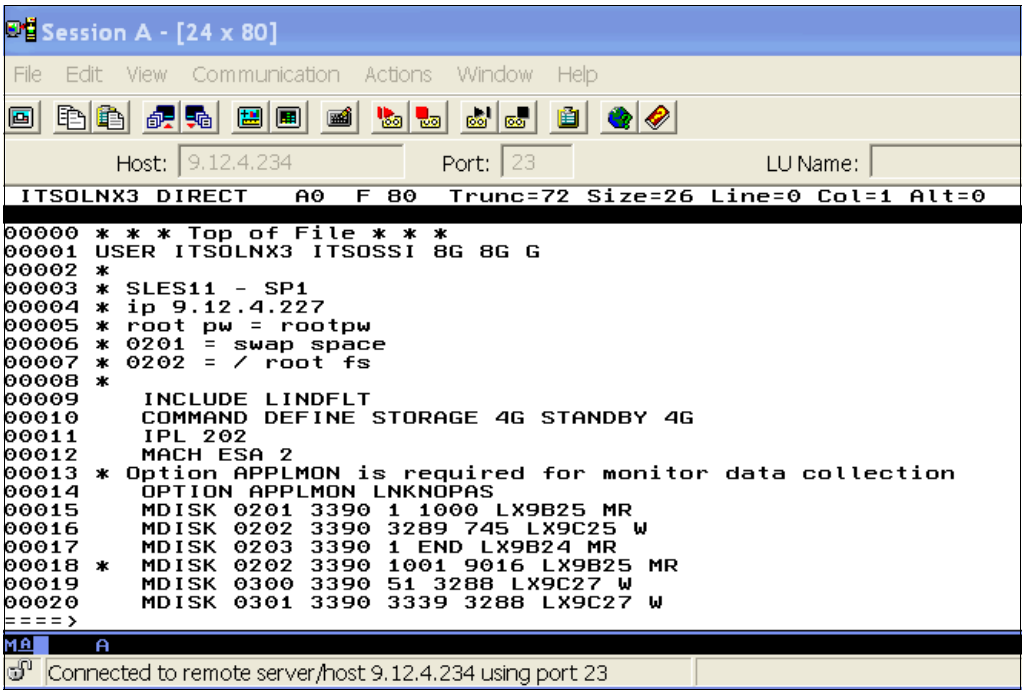


Figure 3-3 The define storage statement

Note: If you just update the first line with the wanted range, for instance, `USER VM_ID 4G 8G G`, it will not work. And if you try to increase it dynamically, the guest freezes and you need to reboot. Set the total memory available for the guest (Active + Stand By).

b. Replace the guest directory on DIRMAINT with the new file by using the following command. Replace **itsolnx3** with the name of your Linux guest: dirm for itsolnx3 replace.

With the guest memory set on the z/VM side, use the following steps to dynamically change the memory size:

- Dynamic memory changes from the Linux guest

  a. Check available active and stand-by memory as shown in Example 3-1.

Example 3-1 lsmem displaying total of memory online and offline

```
# lsmem
Address Range                           Size(MB)  State   Removable  Device
==========================================================================
0x0000000000000000-0x000000000fffffff       256  online   no         0-63
0x0000000010000000-0x000000002fffffff       512  online   yes        64-191
0x0000000030000000-0x000000007fffffff      1280  online   no         192-511
0x0000000080000000-0x00000000ffffffff      2048  offline  -          512-1023
Memory device size  : 4 MB
Memory block size   : 256 MB
Total online memory : 2048 MB
Total offline memory: 2048 MB
```

  b. Dynamically increase memory as shown in Example 3-2.

Example 3-2 lsmem displaying the new total of offline memory after dynamically increasing

```
itsolnx1:~ # chmem -e 1G
itsolnx1:~ # lsmem
Address Range                           Size(MB)  State   Removable  Device
==========================================================================
0x0000000000000000-0x000000000fffffff       256  online   no         0-63
0x0000000010000000-0x000000001fffffff       256  online   yes        64-127
0x0000000020000000-0x000000007fffffff      1536  online   no         128-511
0x0000000080000000-0x00000000bfffffff      1024  online   yes        512-767
0x00000000c0000000-0x00000000ffffffff      1024  offline  -          768-1023
Memory device size  : 4 MB
Memory block size   : 256 MB
Total online memory : 3072 MB
Total offline memory: 1024 MB
```

  c. If necessary, dynamically decrease memory as shown in Example 3-3.

Example 3-3 lsmem displaying the new total of offline memory after dynamically decreasing

```
itsolnx1:~ # chmem -d 1G
itsolnx1:~ # lsmem
Address Range                           Size(MB)  State   Removable  Device
==========================================================================
0x0000000000000000-0x000000000fffffff       256  online   no         0-63
0x0000000010000000-0x000000001fffffff       256  online   yes        64-127
0x0000000020000000-0x000000007fffffff      1536  online   no         128-511
0x0000000080000000-0x00000000ffffffff      2048  offline  -          512-1023
Memory device size  : 4 MB
Memory block size   : 256 MB
Total online memory : 2048 MB
Total offline memory: 2048 MB
```

Note: The **lsmem** and **chmem** commands are available to RHEL 6.0 and higher, and SLES 11 SP2 or higher.

3.1.1 Swap

You can use more than one swap device at a time. By default, swap devices are set online (swapon) with different priorities and are then used one after the other.

On stand-alone servers, we frequently hear that you should use double the amount of memory for swap purposes on Linux guests. However, when using Linux on System z, in order to mitigate outage risks that are caused by hanging or kernel panic, our recommendation is to use a 512 MB VDisk set as the highest priority swap disk because it is faster than regular disks. If needed, add a medium-sized mini disk (1 - 2 GB). Memory is necessary for the application demands; therefore, monitor memory usage carefully in case of unexpected demands.

For more information about swap management for Linux on System z, see the following documents:

- z/VM and Linux Guest Performance Best Practices:

  http://www.vm.ibm.com/education/lvc/lvclbest.pdf

- *The Virtualization Cookbook for z/VM 6.3, RHEL 6.4 and SLES 11 SP3*, SG24-8147:

  http://www.redbooks.ibm.com/abstracts/sg248147.html?Open

If you are moving a guest from development to production, clean memory pages and swap space ahead of time for a clearer monitoring picture. This way, you are able to know if the guest is sized correctly and if it is not, you can take the applicable actions to fix it and deliver a healthy system to production.

If the available free memory is larger than the swap space, you can clean swap space without killing any processes or rebooting the operational system. You can also increase memory (Example 3-2 on page 32) or create a temporary swap file, as shown in Example 3-7 on page 35, to move pages.

- Cleaning memory pages (Example 3-4)

  In order to move a Linux on System z guest from development to production, clean the memory buffer and cache to obtain free space. The **sync** command flushes the buffer and moves all unused pages to disk.

Example 3-4 How to clean memory pages

```
# free
total       used       free     shared    buffers     cached
Mem:       4105924    3308760     797164<-        0     205056    2515524
-/+ buffers/cache:     588180    3517744
Swap:      1039808     263460     776348
# sync
# echo 3 > /proc/sys/vm/drop_caches
# free
total       used       free     shared    buffers     cached
Mem:       4105924    2094232    2011692<-       0       4940    1540192
-/+ buffers/cache:     549100    3556824
Swap:      1039808     263460     776348
```

Note: This feature has been implemented since Kernel 2.6.16.

- Cleaning the swap pages (Example 3-5)

  Use the **swapon** and **swapoff** commands from your Linux on System z guest to enable and disable devices and files for paging and swapping.

  Cleaning the swap pages is a simple procedure; however, the devices should be taken off line one at a time, not simultaneously, starting from the lowest to highest priority devices. If immediately after clean-up of a device, you notice that it starts to page again, review the guest sizing and consider performing a memory upgrade.

  First, as root from your Linux on System z guest, issue the **free** command to see what is being used. Next, run the actual commands that clear the swap, as shown in Example 3-5. The **swapoff /dev/dasda1** command disables swapping on the specified device (/dev/dasda1) and files. The **swapon -s** command displays a swap usage summary by device. Notice in this example, we run the **swapoff** command against the device named */dev/dasda1* and then when we display the swap usage summary by device, it no longer shows in the list of swap devices.

Example 3-5 How to clean swap space

```
# free
total       used       free     shared    buffers     cached
Mem:       4105924    3308760     797164<-        0     205056    2515524
-/+ buffers/cache:     588180    3517744
Swap:      1039808     263460     776348
# swapoff /dev/dasda1
# swapon -s
/dev/dasdb1                             partition       259952  88940   40
/dev/dasdc1                             partition       259952  88832   30
/dev/dasdd1                             partition       259952  33468   20
# free
total       used       free     shared    buffers     cached
Mem:       4105924    2166348    1939576          0      15400    1573940
-/+ buffers/cache:     577008    3528916
Swap:       779856     211240     568616
# swapoff /dev/dasdb1
# free
total       used       free     shared    buffers     cached
Mem:       4105924    2199580    1906344          0      16652    1581092
-/+ buffers/cache:     601836    3504088
Swap:       519904     122300     397604
```

  After cleaning all disks, execute the **swapon -a** command to add the swap space back in. Run the **swapon -a** command in order to make all devices marked as “swap” swap devices in `/etc/fstab` available as shown in Example 3-6.

Example 3-6 swapon results after swap clean up

```
# swapon  -a
Filename                                Type            Size    Used    Priority
/dev/dasda1                             partition       259952  0       50
/dev/dasdb1                             partition       259952  0       40
/dev/dasdc1                             partition       259952  0       30
/dev/dasdd1                             partition       259952  0       20
```

- If the free memory is lower than the swap size

If the free memory is not enough to allocate the swap space and you do not have available memory for dynamically increasing it, create a temporary swap file, which allows you to clean up pages without taking the server down.

To do this, first ensure that you have enough free space to copy an entire swap device. In Example 3-7, because our server uses 256 MB VDisk swap devices, we created a temporary swap file that needed at least 256 MB.

Example 3-7 How to create a swap file

```
# dd of=/opt/swapfile if=/dev/zero bs=1k count=262144
262144+0 records in
262144+0 records out
268435456 bytes (268 MB) copied, 1.9229 seconds, 140 MB/s
```

After creating the file, format that as a swap device before activating it as a swap area (Example 3-8).

Example 3-8 Formatting the swap file and activating it

```
# mkswap /opt/swapfile
Setting up swapspace version 1, size = 268431 kB
# swapon /opt/swapfile
# swapon -s
Filename                                Type            Size    Used    Priority
/dev/dasda1                             partition       259952  272     50
/dev/dasdb1                             partition       259952  0       40
/dev/dasdc1                             partition       259952  0       30
/dev/dasdd1                             partition       259952  0       20
/opt/swapfile                           file            262136  0       -1 <-
```

After the temporary swap device is created, clean up the swap page.

Note: In this specific example, clean one device at a time and use the **`swapon`** command before cleaning the next disk.

3.2 Memory planning for z/VM

The use of z/VM as a hypervisor for Linux on System z guests provides the ability to overcommit its z/VM main storage. What this means is that the sum of all defined memory of all Linux guests could be higher than the z/VM LPAR. z/VM uses its page subsystem to hold those frames that do not fit in its main storage. There is no rule about how much storage can be virtualized. For production environments, the virtual to real (V/R) ratio should be no more than 1.5:1. If you have a very efficient paging subsystem or if this environment will be used for quality assurance (QA) or development purposes, you can have a higher V/R ratio, 3:1 for example. z/VM dynamically allows the increase of real storage by bringing designated amounts of standby storage online.

3.2.1 z/VM storage

Consider the following factors when planning for z/VM storage.

Size of the Linux virtual servers

If the z/VM LPAR hosts 20 Linux virtual servers that require 4 GB of storage each, z/VM has to address 80 GB of storage. If we apply the 1.5:1 V/R factor, this LPAR should be sized as 54 GB. As explained in section 3.2, “Memory planning for z/VM” on page 35, z/VM explores its paging subsystem to support this storage overcommitment, in this case 26 GB. See section 3.2.2, “Paging subsystem definitions” on page 37 for more details about how the page area should be defined.

Note: In an SSI environment, clients should also consider the ability of moving Linux servers among z/VM members, meaning other z/VM SSI members must have enough storage to host the relocated Linux servers.

LPAR image profile definition

In order to increase the number of Linux guests or increase the amount of storage that those guests are using, define an amount of reserved storage in the LPAR image profile configuration. Figure 3-4 shows a sample configuration of storage definitions in the Hardware Management Console (HMC) menu that is found under the menu option, “Customize image profiles”. This example shows a system with 30 GB of main storage and 4 GB of reserved. With this configuration, you are able to dynamically increase the size of the z/VM LPAR’s main storage up to 34 GB.

Up to z/VM v6.2, you are able to use a high speed paging area called *expanded storage*. Define 2 GB to 4 GB of expanded storage in the z/VM LPAR image profile configuration for performance issues. z/VM supports up to 128 GB of expanded storage.

z/VM v6.2 supports up to 256 GB of main storage. z/VM v6.3 increases the real memory limit on individual virtual machines from 256 GB to 1 TB. It also proportionately increases total virtual memory based on tolerable over-commitment levels and workload dependencies.

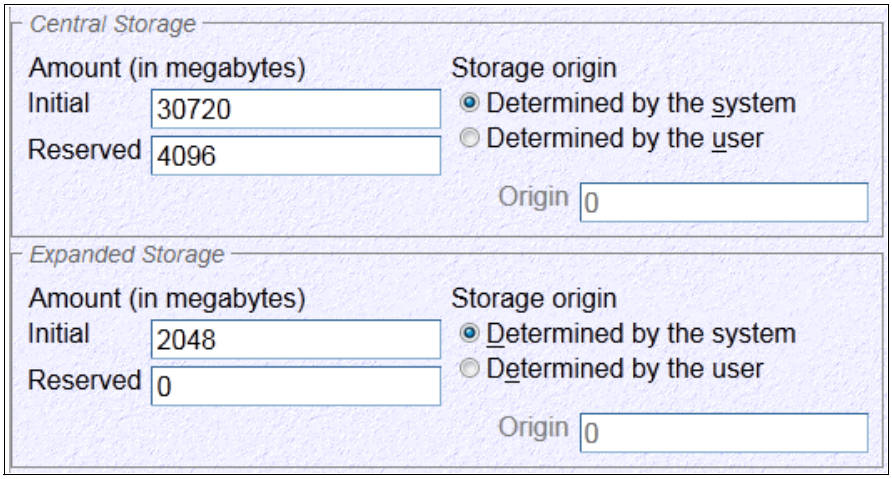


Figure 3-4 Main storage reserved definition in HMC image profile configuration

Note: z/VM does not support dynamic storage removal.

Commands

Example 3-9 shows the z/VM control program (CP) **QUERY STORAGE** command. This command displays the storage environment configuration and the z/VM CP command, **SET STORAGE**, allows the z/VM administrator to dynamically increase the amount of real storage. In our example, we dynamically increased the z/VM real storage by 4 G to bring it up to 34 GB.

Example 3-9 Increasing z/VM storage dynamically

```
QUERY STORAGE
STORAGE = 30G CONFIGURED = 30G INC = 256M STANDBY = 4G RESERVED = 0

SET STORAGE +4G
STORAGE = 34G CONFIGURED = 34G INC = 256M STANDBY = 0  RESERVED = 0
```

Terms that are used in Example 3-9 are defined as follows:

STORAGE Specifies the amount of storage that is currently allocated to z/VM.

CONFIGURED Specifies the amount of real storage that is allocated for the z/VM LPAR.

INC Specifies the size of the real storage increment.

STANDBY Specifies the amount of real storage in standby state that is available to be brought online with the **SET STORAGE** command. Standby storage is a calculated value based on the amount of installed storage that is not currently claimed by active logical partitions and the reserved storage specified for the logical partition in which z/VM is running.

RESERVED Specifies the amount of real storage in reserved state that can become available (for example, if another logical partition is deactivated).

Note: RESERVED storage that is defined in the LPAR image profile is shown in z/VM as STANDBY.

3.2.2 Paging subsystem definitions

To determine how direct access storage device (DASD) should be allocated to this paging subsystem, we took 26 GB and multiplied by 2 to satisfy the 50% rule of thumb for paging subsystem health. In this case, the z/VM page subsystem should have at least 52 GB, which means that approximately twenty-two 3390-3 DASDs or eight 3390-9 DASDs are needed for the paging subsystem. Table 3-1 on page 38 shows the details.

We did not consider 3390-27 for paging because z/VM does not support parallel access volume (PAV, Dynamic PAV, or HyperPAV) for CP_OWNED devices. What this means is that multiple extended count key data (ECKD) devices are needed to avoid queues in the paging subsystem that would result in performances issues.

If using FCP chpids and SCSI controllers, consider using them for paging. Small Computer System Interface (SCSI) logical unit numbers (LUNs) can be defined as z/VM EDEVICEs and be used for paging. This is an option to consider when you have a very high page rate. The use of SCSI LUNs for paging provides I/O concurrency without needing multiple volumes. However, it uses more CPU cycles than the use of ECKD DASDs. z/VM also supports the dynamic addition of paging devices.

Table 3-1 Paging subsystem calculation

20 Linux servers storage sum	=	20 x 4 GB	=	80 GB
z/VM LPAR (1.5:1 V/R factor)	=	80 GB x 1.5	=	54 GB
z/VM paging subsystem	=	80 GB - 54 GB	=	26 GB
z/VM paging subsystem health (50%)	=	26 GB x 2	=	52 GB
3390-3 DASDs option (2.3 each)	=	52 GB/2.3 GB	=	22 DASDs
3390-9 DASDs option (6.8 each)	=	52 GB/6.8 GB	=	8 DASDs

More performance information about z/VM page subsystems can be found in 10.2.9, “Paging subsystem” on page 139.

Note: Do not mix ECKD and EDEV paging volumes on the same system.

VIR2REAL tool

The VIR2REAL EXEC calculates and displays the ratio of the virtual storage (memory) to the real storage of a z/VM environment. A system with a virtual to real storage ratio that is too large might experience excessive paging rates or other performance problems because the real memory is overcommitted. A system with a small virtual to real storage ratio (less than or near to 1:1) that has sufficient CPU capacity available is able to handle a larger workload. This tool can be download from the following website:

`http://www.vm.ibm.com/download/packages`

The output of this tool shows the current ratio for the virtual servers that are active on the system at the time it runs. For tracking the ratio over time, obtain and use a performance monitor product. Example 3-10 shows part of the VIR2REAL EXEC execution. In this example, the z/VM V/R ratio is 3.1:1 and seventy-eight 3390-3 DASDs are available for the paging subsystem.

Example 3-10 VIR2REAL execution

```
Total Virtual storage (all logged on userids):    206416 MB (201.6 GB)
Total LPAR Real storage:                           81920 MB ( 80.0 GB)
Expanded storage usable for paging:                 2048 MB (  2.0 GB)

Total Virtual disk (VDISK) space defined:          43256 MB ( 42.2 GB)

Virtual + VDISK to Real storage ratio:             3.1 : 1

Percent of paging space needed for these virtual storage totals:
Virtual+VDISK for all logged on guests:            136%
Paging:  78 volumes active, usable space is:       183072 MB (178.8 GB)
Total Paging space in use, 25% utilization:         46188 MB ( 45.1 GB)
```

Note: By default, the output is written to the console, but it can also be saved in a file by specifying the FILE argument.

3.3 Channel planning

In this section, we describe I/O connectivity for Linux on System z solutions. System z supports various types of channels. IBM plans not to offer IBM ESCON® channels as an orderable feature on System z servers that follow the z196 (machine type 2817) and z114 (machine type 2818). In addition, ESCON channels cannot be carried forward on an upgrade to subsequent versions of servers. The following two major channel types are described in this section:

- FICON
- FCP

Note: Both z/VM and Linux on System z support FICON and FCP.

FICON

The Fibre Connection (FICON) channel on System z has been designed to replace Enterprise Systems Connection (ESCON) channels. Today, the FICON channel supports a link data rate of 2, 4, or 8 Gbps autonegotiated. The ESCON channel link data rate is constant, at 17 megabytes per second (MBps).

FICON features that are supported by the FICON protocol include access to ECKD peripheral devices as well as FICON channel-to-channel (CTC). Additionally supported is z High Performance FICON (zHPF). zHPF is an extension to the FICON architecture designed to improve the execution of small block I/O requests. zHPF is a performance and reliability, availability, serviceability (RAS) enhancement of the IBM z/Architecture and the FICON channel architecture that is implemented in zEnterprise EC12, zEnterprise 196, zEnteprise 114, and System z10 servers. Exploitation of zHPF by the FICON channel, the operating system, and the control unit is designed to help reduce the FICON channel overhead. This is achieved with protocol simplification and a reduced number of information units (IUs) processed, resulting in more efficient use of the channel.

Figure 3-5 on page 40 shows an example of a 4 KB read FICON channel program. This demonstrates that zHPF is particularly beneficial for small block transfers. zHPF channel programs can be used by online transaction processing (OLTP) I/O workloads that transfer small (4 KB) blocks of fixed-sized data, such as from a database management system.

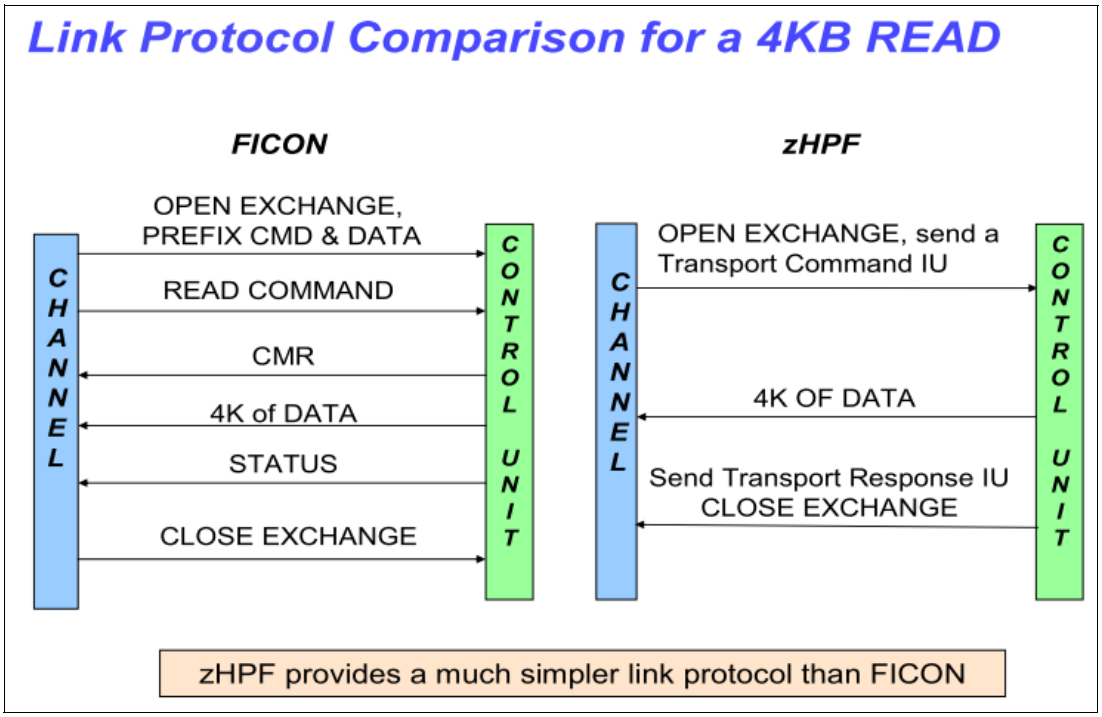


Figure 3-5 Link protocol comparison

zHPF Notes:

- z/VM 6.2 can provide Linux guest support for zHPF with APAR VM65041. If using z/VM PTK, it requires APAR VM65044.
- IBM is working with its Linux distribution partners to include support in future Linux on System z distribution releases. Following are the currently supported Linux on System z distributions:
  - Single-track mode
    - SLES 11 SP1
    - RHEL 6.0
  - Multi-track mode

    RHEL 6.1
- The z High Performance FICON feature is required when using the IBM DS8000® family. This is a priced licensed feature (one-time charge) and has a monthly maintenance charge.

FCP

Fibre Channel Protocol (FCP) supports access to Small Computer System Interface (SCSI) peripheral devices. The operating systems supported today are z/VM, z/VSE, and Linux on System z.

System z FCP support enables z/VM and Linux running on System z to access industry-standard SCSI devices. For disk applications, these FCP storage devices use fixed block (512-byte) sectors rather than ECKD format.

FCP architecture defines three separate topologies to support connectivity between endpoints:

- Point-to-point
- Arbitrated loops
- Switched fabric

Note: System z FICON Express adapters support both FICON and FCP. So there is no change from the hardware configuration perspective when clients select FICON or FCP.

For more information about FICON and FCP channels, see the following IBM Redbooks publications:

- *IBM System z Connectivity Handbook*, SG24-5444
- *Fibre Channel Protocol for Linux and z/VM on IBM System z*, SG24-7266

3.3.1 Fabric transport and addressing

In this section, we describe some of the FICON and FCP terminology that is used in a Fibre Channel storage area network (SAN).

The FICON environment is shown in Figure 3-6. Each unit (UA) is represented on the host side as a subchannel (SCH) and managed by a device number (DEVNO), each unit is associated to a control unit (CU), which can be reached via a path group (process group ID (PGID)). Each path group consists of multiple paths (channel-path identifiers (CHPIDs)). System z connects via Fibre Channel connections to the FICON Director. The relationship between each UA, CU, and CHPID is defined in the input/output configuration data set (IOCDS).

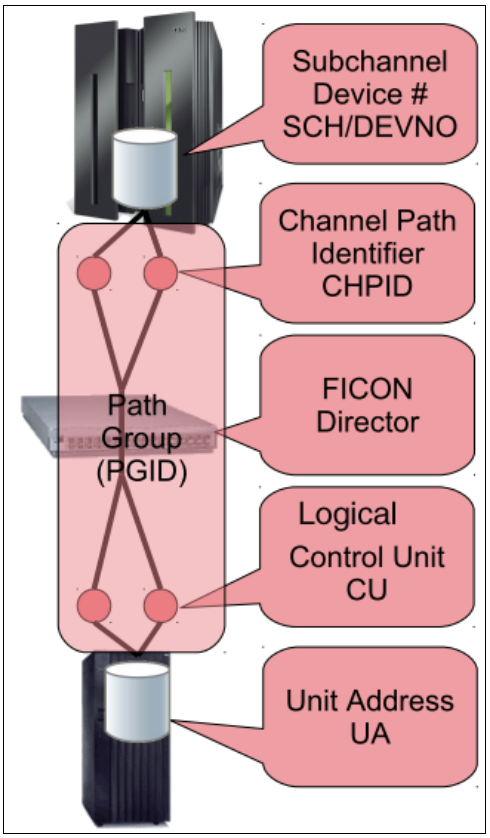


Figure 3-6 FICON terminology

Figure 3-7 shows FCP fabric connectivity. Units (LUNs) are managed by the operating system. A virtual host bus adapter (HBA) is shown on the host side as a subchannel (SCH) and device number (DEVNO). Each FICON adapter (CHPID) hosts multiple HBAs with their own N-Port IDs. Addressing is done through worldwide port names (WWPNs). No path grouping is performed by the hardware or firmware. System z is connected via Fibre Channel connections to a FICON switch. The association is managed by the System z IOCDS for virtual HBAs but not for LUNs.

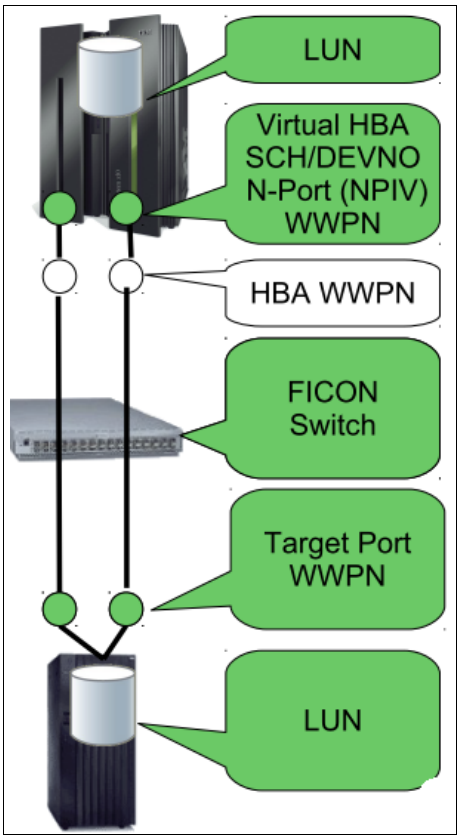


Figure 3-7 FCP terminology

On the fabric layer, there is not much of a difference. A FICON Express Card is used for both attachments. Each port can be configured as FCP or FICON, mutually exclusive. Both FICON and FCP use a Fibre Channel network for transporting packets from the host to the storage controller. Addressing in the Fibre Channel network is based on WWPNs. A WWPN is analogous to MAC addresses in Ethernet networks.

Both FICON and FCP support point-to-point and switched fabric. FICON supports up to one fabric hop (two switches on the path). FCP supports an arbitrary number of hops. If FICON is adopted, point-to-point or switch fabric can be used depending on the size of the production environment. For an FCP-based solution, use a SAN switch between System z servers and storage servers from channel resource sharing and security perspectives. We describe this configuration further in “Sharing via FCP” on page 44.

Note: Arbitrated loop topology is not supported, even if there are only two nodes in the loop.

3.3.2 Channel configuration and management consideration

Table 3-2 compares the differences in configuration and security between FICON and FCP.

Table 3-2 FICON versus FCP in channel configuration and security

FICON	FCP
Fabric is configured by System z firmware and FICON Director	Fabric is configured by SAN administrator
System z I/O configuration manages connectivity and device access	► SAN zoning manages connections ► LUN masking manages device access
Subchannel represents a target device	► Subchannel represents a source adapter (HBA) ► The operating system manages devices that are attached to a port
System z IOCDS and z/VM I/O configuration controls the access to devices based on LPAR and guest IDs using CU and unit numbers	► SAN zoning and LUN masking control access to devices based on initiator and target WWPNs and LUNs

FCP access control

In this section, we describe the consideration of FCP access control in a System z environment. In a switched fabric network, SAN switch provides an ability to control access to nodes and devices, which is called *LUN masking and zoning*. LUN masking and zoning can be used to prevent servers from accessing storage that they are not permitted to access.

But when multiple Linux images share an FCP channel in System z, the use of zoning and LUN masking cannot effectively create appropriate access controls among the Linux images. Use of N-Port ID Virtualization (NPIV) along with zoning and LUN masking can ensure data integrity among Linux images sharing an FCP channel. This process is described with more details in “Sharing via FCP” on page 44.

Multiple path management

Multiple paths provide backup in the event of a hardware failure and improve I/O performance through load balancing. FICON and FCP use multiple paths differently.

Figure 3-8 shows the difference between FICON and FCP in multiple paths.

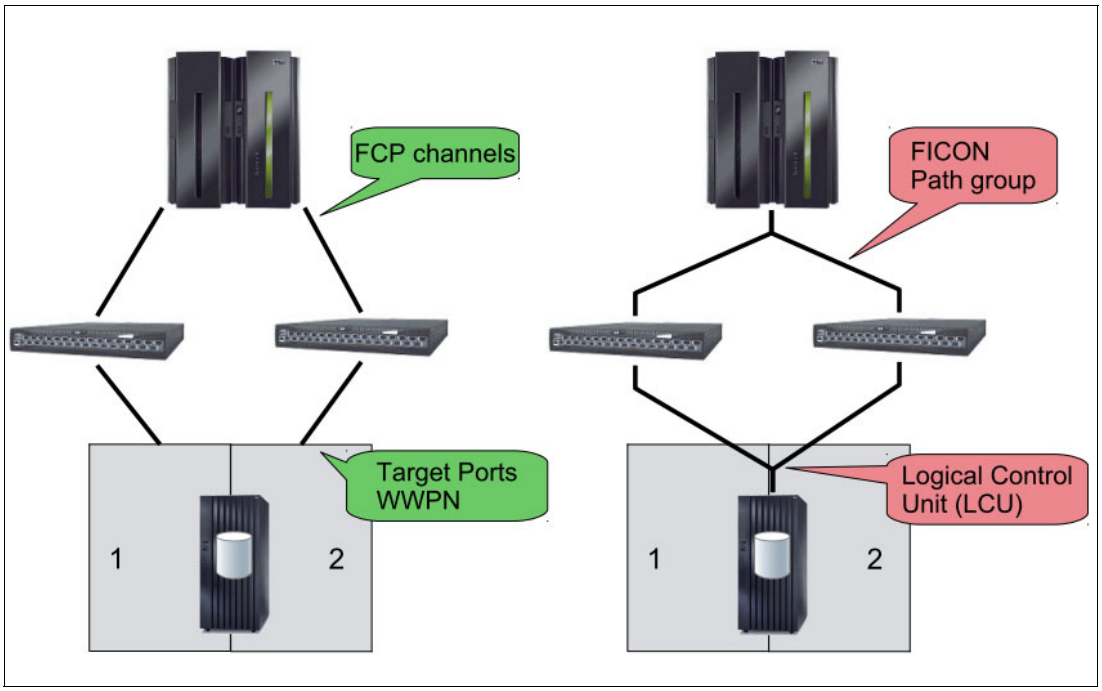


Figure 3-8 Difference between FICON and FCP in multiple path management

FICON paths are grouped into path groups in the System z I/O configuration. System z firmware manages everything in multipathing including such things as failover and failback, retries on alternate path on error, and so on.

In an FCP channel, each path is presented to the operating system individually. So unlike FICON, the operating system manages everything in a multipathing environment. The cost to manage multipathing is usually additional system overhead.

For more multipathing configuration details, refer to the IBM Redbooks publication *Fibre Channel Protocol for Linux and z/VM on IBM System z*, SG24-7266.

3.3.3 Channel sharing

Channels on System z servers are always shared by different LPARs and Linux guests in z/VM. FICON and FCP use different ways to manage shared channel resources.

Sharing via FICON

In FICON based solutions, the channel and disk sharing management is simple. The System z hardware and firmware handles all FICON channels management, including multipathing management. The System z IOCDS and z/VM I/O configuration controls access to devices based on LPAR and guest IDs. This includes definitions whether a device is shared or not. The same is true in a zHPF environment.

Sharing via FCP

FCP channels can be shared by multiple LPARs. Each port on the FICON Express adapter is assigned a permanent 64-bit WWPN by the manufacturer. This is used during fabric login (FLOGI). Previously, all FCP subchannels shared a common WWPN burned into the associated FICON Express adapter. Within the SAN fabric, therefore, the actual I/O initiator (a

specific subchannel) could not be determined because the initiator was always the WWPN of the FCP adapter. Access control could be managed only at the adapter level. LUN sharing conflict would occur and result in errors.

N_Port ID Virtualization (NPIV) allows a single FCP port to register multiple WWPNs with a fabric name server. Each registered WWPN is assigned a unique N_Port ID. With NPIV, a single FCP port can appear as multiple WWPNs in the FCP fabric. I/O transactions are separately identified, managed, transmitted, and processed just as though each OS image had its own unique physical N_Port. Figure 3-9 illustrates that NPIV provides unique WWPNs to servers sharing an FCP port.

Notes:

- NPIV requires support in the entry switch used to attach the channel to the SAN fabric.
- The NPIV must be enabled on SAN fabric before it is enabled on the System z server.
- And the NPIV and permanent (default) WWPNs must be defined in the fabric (switch) zoning and LUN masking on the storage server.

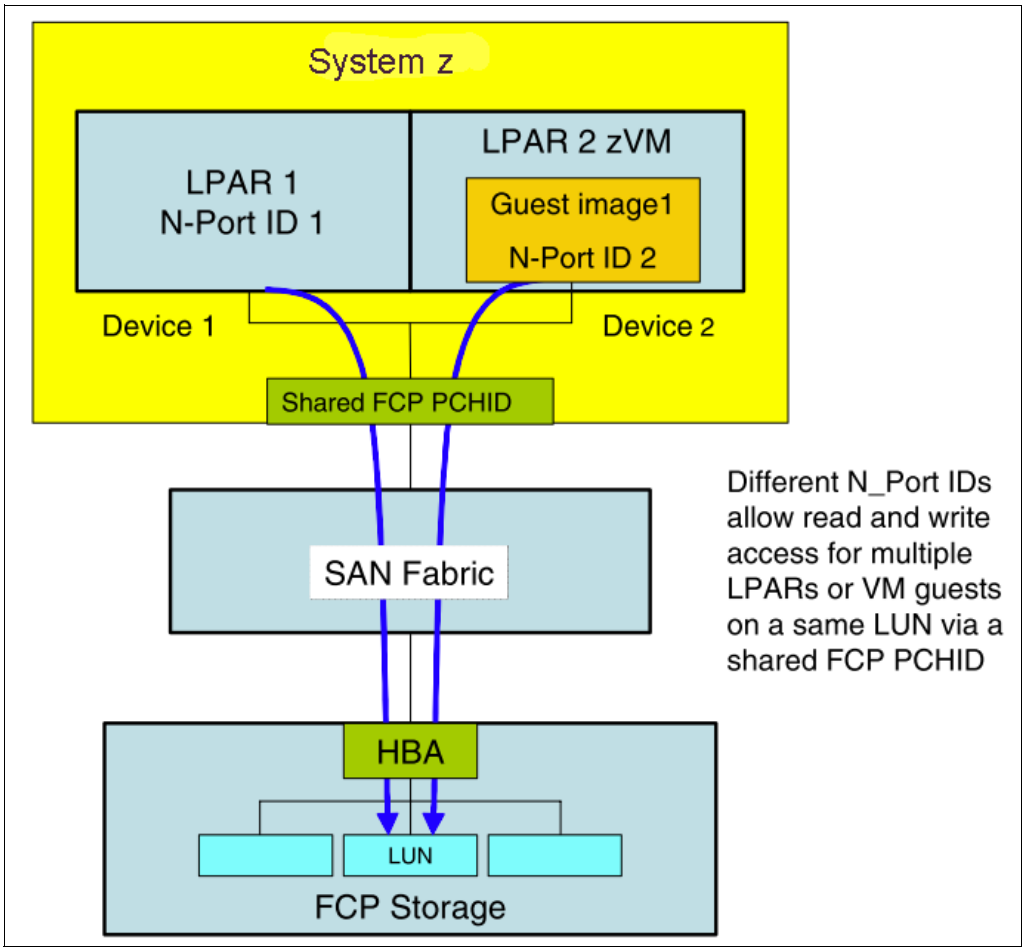


Figure 3-9 Sharing FCP port with NPIV

z/VM uses the adapter capability to define multiple virtual FCP channels, each with its own unique Fibre Channel port name and Fibre Channel identifier (FC_ID). By assigning distinct virtual port names to different guests, the guests can use the virtual FCP channels as though they were using dedicated physical FCP channels. Access controls based on the virtual port names can be applied in the SAN fabric using standard mechanisms like zoning in the

switches and LUN masking in the storage controllers, thereby providing access control at the FCP subchannel level.

z/VM support of NPIV-enabled hardware is automatic and transparent. No special initialization or command is required. The **QUERY** command displays the hardware-assigned WWPN of NPIV-enabled FCP subchannels.

For more information about this topic, see *z/VM: CP Planning and Administration*, SC24-6178-03, which is available at the following website:

http://publib.boulder.ibm.com/infocenter/zvm/v6r2/index.jsp?topic=%2Fcom.ibm.zvm.v620.hcpa5%2FhcsgOc11221.htm

Do not deploy any more than 32 virtual nodes per port-pair link because of fabric and switch port limitations.

Use the command that is shown in Example 3-11 to verify if the NPIV has been enabled. If both port names are the same, the FCP subchannel does *not* use NPIV. If they differ, the FCP subchannel uses NPIV. In this example, they use NPIV.

Example 3-11 The command to verify if the NPIV is enabled

```
# lszfcp -a | grep port_name
    permanent_port_name = "0xc05076ffe5005611"
    port_name           = "0xc05076ffe5005350"
```

For more information about the NPIV feature and configuration on the System z server, refer to Chapter 10*:* The N_Port Virtualization feature in the IBM Redbooks publication, *Fibre Channel Protocol for Linux and z/VM on IBM System z*, SG24-7266. Zoning and LUN masking configuration are specific to the switch and storage server that are used in the fabric, so consult with the server switch and storage manuals corresponding to your installation.

A WWPN prediction tool is now available from IBM Resource Link® to assist you with pre-planning of your SAN environment before the installation of your System z server. This stand-alone tool is designed to allow you to set up your SAN in advance so that you can be up and running much faster after the server is installed. The tool assigns WWPNs to each virtual FCP channel/port using the same WWPN assignment algorithms that a system uses when assigning WWPNs for channels using NPIV. Contact your IBM representatives for the tool and usage information.

3.3.4 Performance assessment

Systems using Linux on System z can access disks either using FICON, zHPF, native FCP, or FCP via z/VM. From the I/O performance perspective, the results are various when using a different channel infrastructure base. In this section, we discuss the performance differences between channel types.

FICON

Compared to FICON, use zHPF whenever possible. zHPF uses a new I/O channel program format referred to as *transport-mode I/O*. As a result of lower overhead, transport-mode I/Os complete faster than traditional FICON command-mode I/Os do, resulting in higher I/O rates and less CPU overhead.

Figure 3-10 shows the FICON performance on System z. Although the results shown in the figure are based on z/OS, Linux on System z has similar results. Using zHPF and Linux on System z provides greater I/O performance improvements.

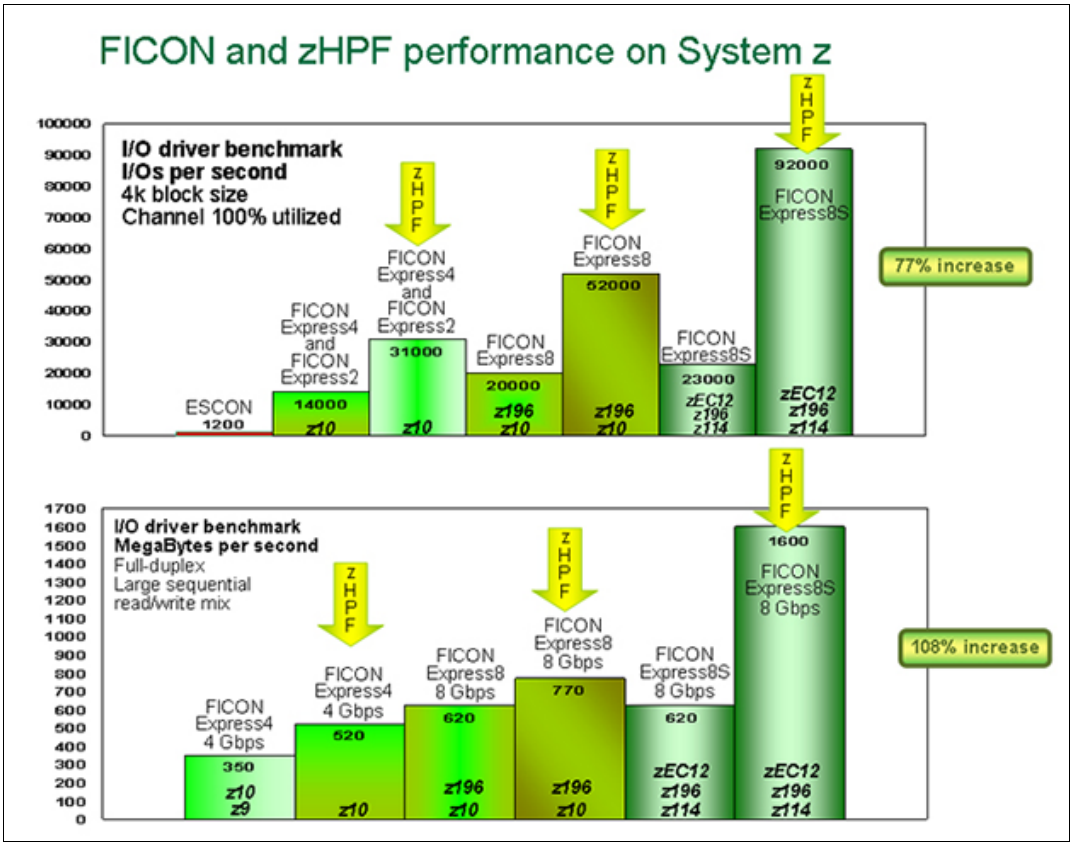


Figure 3-10 FICON performance on System z

During large data transfers with zHPF, the FICON channel processor utilization is much lower than traditional FICON. That means zHPF can get better I/O response time than FICON given the same conditions.

For z/VM itself, when zHPF is being used, compared to FICON, z/VM CMS applications with zHPF achieve a 35% increase in I/O rate (IOs/volume/second), an 18% decrease in service time per I/O, and a 45 - 75% decrease in %CP-CPU per I/O. This is because transport-mode I/O is less complex than command-mode I/O.

For more zHPF performance details for z/VM, see the following website:

`http://www.vm.ibm.com/perf/reports/zvm/html/620jb.html`

FCP

A FICON Express8S feature, when defined as CHPID type FCP, conforms to the FCP standard to support attachment of SCSI devices to complement the classical storage attachment supported by FICON and zHPF channels.

Figure 3-11 shows the FCP performance on System z.

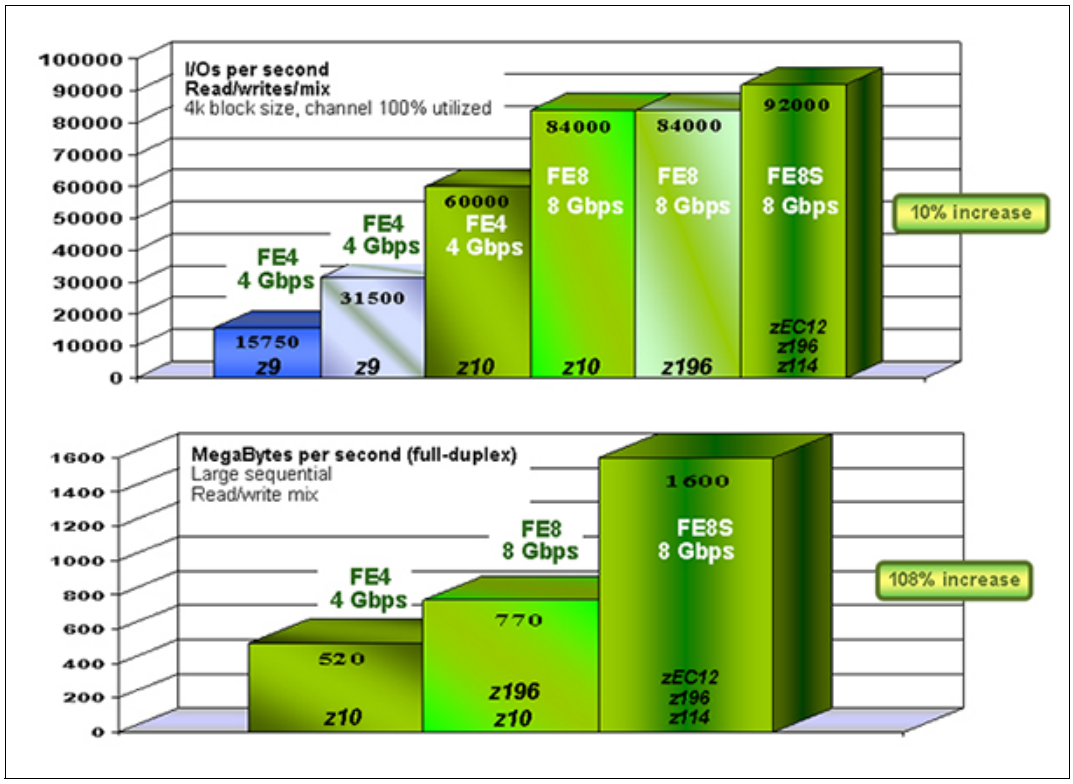


Figure 3-11 FCP performance on System z

In IBM lab measurements, using FICON Express8S in a z196 with the FCP protocol for small data transfer I/O operations, FICON Express8S operating at 8 Gbps achieved a maximum of 92,000 input/output operations per second (IOPS), compared to the maximum of 84,000 IOPS achieved with FICON Express8 operating at 8 Gbps. This represents approximately a 10% increase and applies to reads, writes, and a read/write mix. Results on zEC12 are comparable.

In IBM lab measurements, using FICON Express8S in a z196 with the FCP protocol and an internal driver supporting the hardware data router, executing a mix of large sequential read and write data transfer I/O operations, FICON Express8S operating at 8 Gbps achieved a maximum throughput of 1600 MBps (reads + writes) compared to the maximum of 770 MBps (reads + writes) achieved with FICON Express8 operating at 8 Gbps. This represents approximately a 108% increase. Results on zEC12 are comparable.

Decide between native Linux or a z/VM guest

As described in the scenario introduction, Linux for System z can be running on z/VM or on LPAR directly. We prefer to have Linux run on an LPAR directly as a native LPAR Linux or native Linux. There is little difference, either in throughput or response time, when running FCP natively or as a guest, as shown in Figure 3-12 on page 49, with a 50-50% read/write workload. Similar results were obtained for 100% reads and 100% writes.

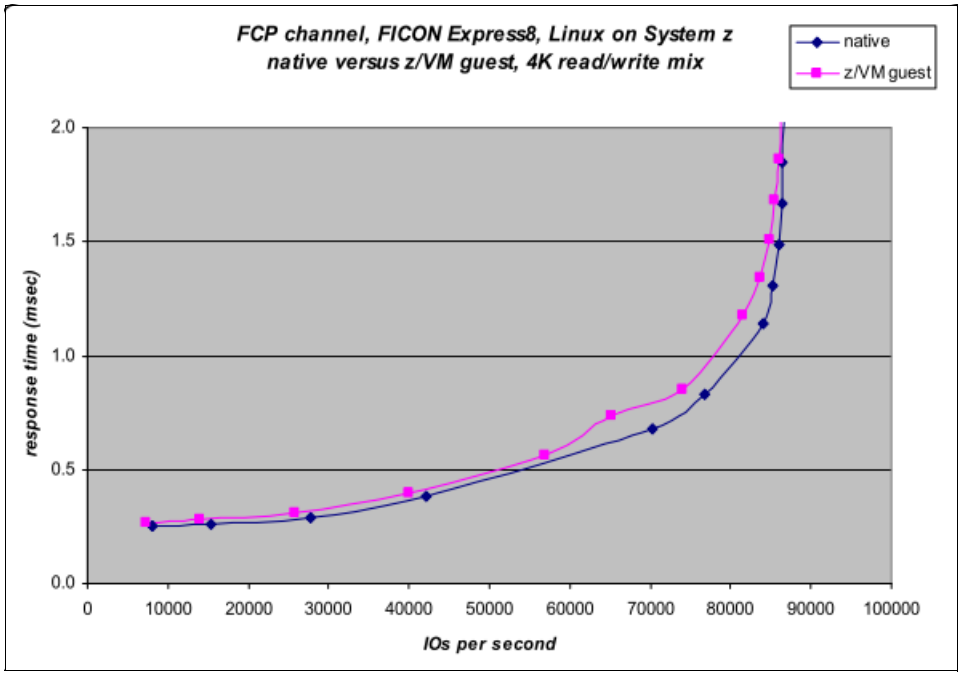


Figure 3-12 Native Linux versus z/VM guest: FCP response time

From the bulk data transfer (bandwidth performance) perspective, native Linux and z/VM guest are also very close. See Figure 3-13.

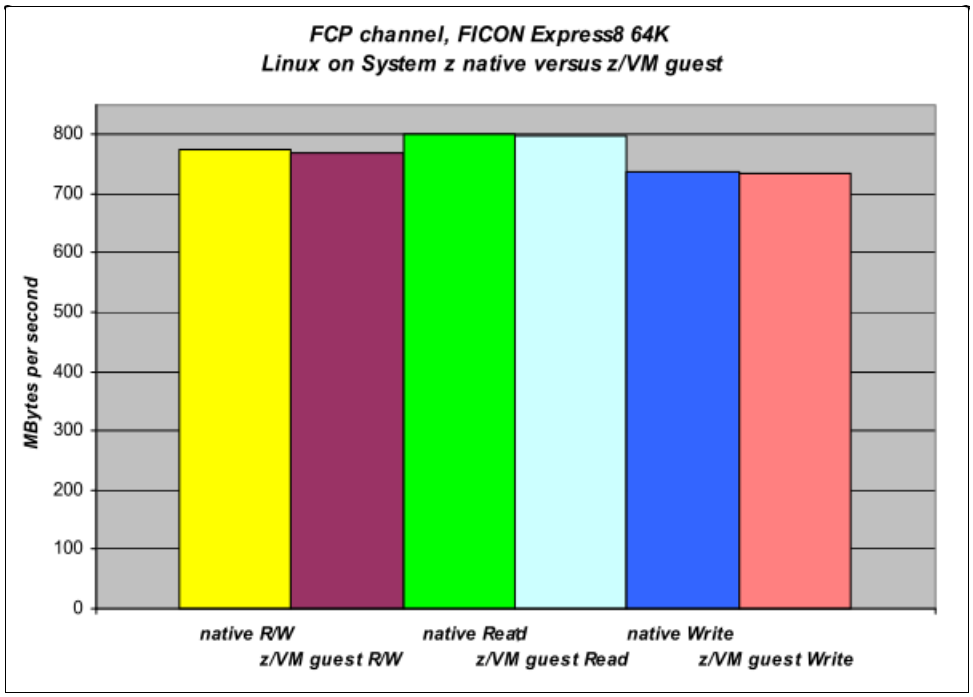


Figure 3-13 Native Linux versus z/VM guest: FCP bandwidth

Linux systems running as z/VM guests benefit from an SSI environment from management, maintenance, and consolidation perspectives. For some very large installations, native Linux is a good choice when the benefits of z/VM are not required in the production environment.

Native Linux accesses SCSI disks directly. Linux as a z/VM guest also can access SCSI disks directly. We call both configurations native FCP or direct-attached SCSI. Native FCP allows significantly higher I/O rates and data rates than traditional FICON. z/VM provides another FCP channel disk management method called *emulated FBA*. Emulated FBA is slower than native FCP, but except for large writes, still better than traditional FICON. We describe this comparison further in “Emulated FBA disk versus direct-attached SCSI” on page 61.

zHPF versus FCP

At the time of writing this book, native FCP and zHPF are comparable with regards to IOPS rate and data rates.

Note: When the zHPF for Linux on System z solution is chosen, avoid mixing FICON and zHPF because it decreases the maximum I/O rate dramatically.

3.3.5 Considerations of choosing FICON or FCP

As described in section 2.2, “Overview of architectures used in this book” on page 13, Linux on System z solutions can use FICON or FCP, or both in an installation. The choice of FICON or FCP usually is not performance-only oriented in a production environment. Usually, clients need to consider the backup and recovery, management and provisioning, disaster recovery, and data administration. Table 3-3 outlines the benefits and costs of using FICON versus FCP.

Table 3-3 FICON versus FCP

	Pros	**Cons**
FICON	► Easy channel and disk management ► System z hardware and firmware take care of multiple paths management ► Can integrate with z/OS IBM GDPS® and IBM HyperSwap® architecture (Disaster Recovery solution) ► ECKD-based backup solution with z/OS ► Provides better I/O monitoring capability ► With the zHPF feature, the I/O performance is comparable to FCP regarding I/O and data rates	► Fair performance for I/O intensive workload with traditional FICON ► Distributed platform administrators are not familiar with FICON technology and configuration ► Single disk capacity is limited to the size of 3390 device model; for example, 224 GB (Model A) ► zHPF is an additional priced feature on the storage server (vendor-specific)
FCP	► Good I/O performance compared to traditional FICON ► Distributed platform administrators are more familiar with FCP and SCSI technology ► Single disk capacity literally unlimited for Linux. The maximum z/VM EDEV (FBA) device size is limited to 1 TB ► FBA disks provide better performance for z/VM paging	► It requires more channel and device management efforts, either on z/VM or Linux ► Operating systems (z/VM or Linux on System z) take care of multiple paths management. It takes additional system overhead ► Cannot integrate with z/OS GDPS and HyperSwap architecture ► Needs more efforts to monitor I/O activity

There are other aspects to consider when using FICON or FCP:

- When using an FCP channel, use a SAN switch. This is a benefit from a channel and device management perspective. Also, enable NPIV in the production environment.
- In the z/VM SSI environment, FICON is mandatory because z/VM must be installed on ECKD devices.
- Whichever channel protocol is chosen, FICON Express 8s dramatically improves performance.
- When choosing zHPF, remember to not mix zHPF and FICON.

4

Storage planning considerations

This chapter provides a comprehensive description of the use of available storage options and methods of operation. We describe the use of IBM HyperParallel Access Volume (HyperPAV), its implementation, and performance considerations. We also describe the selection of extended count key data (ECKD) versus Small Computer System Interface (SCSI) disk for use in a production environment and the performance considerations that it offers. Furthermore, we describe the paging considerations and best practices of determining how many paging devices are required.

© Copyright IBM Corp. 2013. All rights reserved.

4.1 HyperPAV overview

IBM HyperParallel Access Volume (HyperPAV) support complements the existing basic PAV support in z/VM V5.2, for applicable supporting disk storage systems. The HyperPAV function potentially reduces the number of alias-device addresses needed for parallel I/O operations since HyperPAVs are dynamically bound to a base device for each I/O operation instead of being bound statically like basic PAVs. Figure 4-1 illustrates a conceptual overview of a HyperPAV layout.

z/VM provides support of HyperPAV volumes as linkable minidisks for guest operating systems, such as z/OS, that use the HyperPAV architecture. This support is also designed to transparently provide the potential benefits of HyperPAV volumes for minidisks that are owned or shared by guests that do not specifically use HyperPAV volumes, such as Linux and CMS.

z/VM provides support for the HyperPAV feature of IBM direct access storage device (DASD) subsystems. IBM DASD HyperPAV volumes must be defined to z/VM as a 3390 Model 3, 9, 27, 54. OEM storage providers also have included support for HyperPav on their respective models.

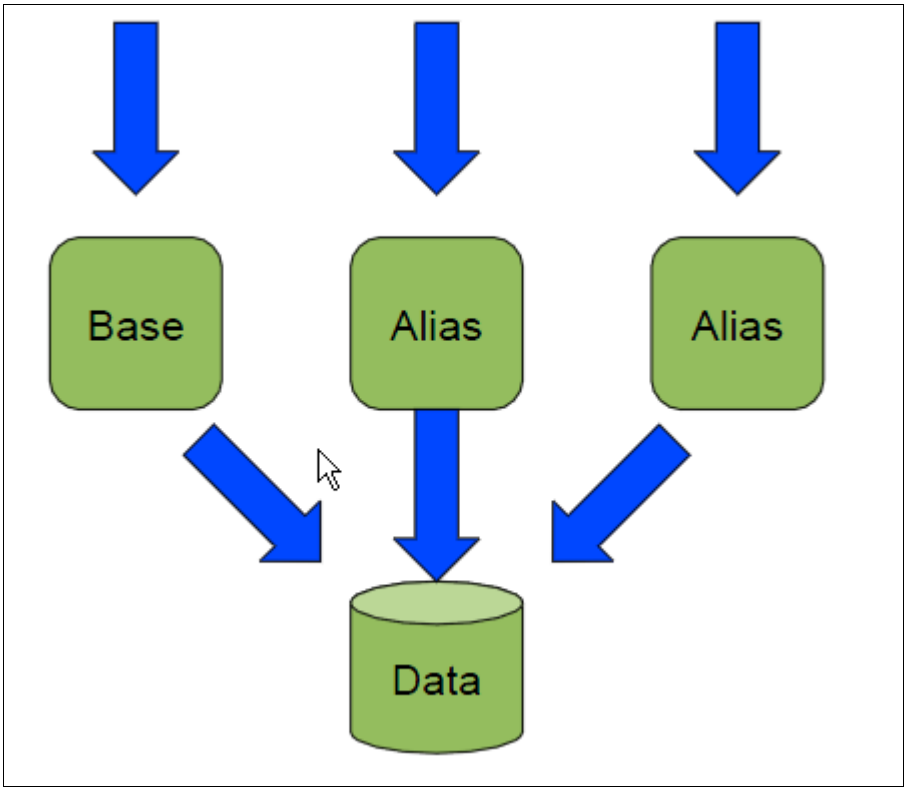


Figure 4-1 Conceptual HyperPAV overview

4.1.1 Benefits of using HyperPAV

To help with the selection of HyperPAV in a production environment, one must ask the question “What are the benefits of selecting HyperPAV?”

What HyperPAV provides is this: HyperPAV volumes are linkable minidisks for guest operating systems, such as z/OS, that use the HyperPAV architecture. Support is also designed to

transparently provide the potential benefits of HyperPAV volumes for minidisks that are owned or shared by guests that do not specifically use HyperPAV volumes, such as Linux and CMS.

Traditional PAV support operates by statically assigning one or more PAV alias subchannels to a specific PAV base device. The storage administrator is able to manually reassign PAV aliases from one base device to another by using the DASD subsystem's configuration menus. And certain software can dynamically reassign PAV aliases. When there are many PAV bases and aliases, it is possible to begin to exhaust the supply of subchannels that are available. The potential for exhausting the supply of available subchannels and easier systems operations had led to the creation of HyperPAV.

4.1.2 Configuring HyperPAV

HyperPAV devices are defined within a storage controller when the proper Licensed Internal Code (LIC) is installed and enabled. The logical subsystem (LSS) is configured as a PAV environment, and when the HyperPAV feature is enabled by z/VM, the static PAV aliases are converted to HyperPAV aliases and they are joined to form the pool for the LSS. z/VM can be configured to operate each LSS in Non-PAV, PAV, or HyperPAV mode by using the CU DASD statement in its configuration file or the **SET CU** command.

The PAV base subchannels are defined in the IOCP as Unit= 3990, 2105, or 2107 on the CTLUNIT statement and the UNIT=3390B on the IODEVICE statement. Each base or alias subchannel can be assigned any available z/VM real device number. Use the DASD subsystem console to initially define which subchannels are base subchannels, which are alias subchannels and, which are associated with each base volume. We used the **CP QUERY PAV** command to view the current allocation of base and alias subchannels.

Certain virtual HyperPAV operations require the persistent use of the same real base or alias subchannel. To facilitate this, each virtual HyperPAV base and alias has an assigned real device subchannel that can be displayed with the **QUERY VIRTUAL vdev DETAILS** and **QUERY VIRTUAL PAV** commands. The assignment is automatic and cannot be changed. One example of this would be the execution of the **READ CONFIGURATION DATA** command. The scheduling of I/O to an assigned device is automatically handled by z/VM during its analysis of the virtual channel program. Because each virtual HyperPAV base or alias must have a uniquely assigned real HyperPAV base or alias subchannel, you cannot have more virtual HyperPAV aliases than real HyperPAV aliases for an LSS.

A dedicated HyperPAV base volume or alias can be assigned only to one guest. I/O operations that are initiated through a HyperPAV alias can be directed only to base volumes that are ATTACHED or LINKED to the issuing virtual machine.

ATTACH/DETACH command

Unlike traditional PAV DASD, HyperPAV base and alias devices can be attached and detached to or from a guest or the system in any order. There is no base before alias (or vice versa) restrictions. HyperPAV aliases can be attached to the system and are used for VM I/O if they contain temporary disk (TDSK) or minidisk (PERM) allocations.

Other CP volume allocations receive no benefit from system attached HyperPAV Aliases.

Unlike traditional PAV DASD, HyperPAV base and alias devices can be attached and detached to and from a guest or the system in any order. There is no base before alias restrictions.

HyperPAV aliases can be attached to the system and are used for VM I/O if they contain temporary disk (TDISK) or minidisk (PERM) allocations. We selected minidisk PERM allocation.

Other CP volumes allocations receive no benefit from system attached HyperPAV aliases.

Minidisk cache settings do not apply to HyperPAV aliases. Cache settings are only applicable to HyperPAV base devices.

The **Set MDCACHE** command cannot be used with HyperPAV aliases or it results in an error.

To define a HyperPAV alias, use the following command:

```
>>--DEFine--HYPERPAValias--vdev--. FOR-'-----.--BASE--basevdev--------><
```

The **DEFINE HYPERPAVALIAS** command is used to create new virtual HyperPAV alias minidisks. A newly defined alias is automatically assigned to a unique underlying real HyperPAV alias. The command fails if no more unique, real aliases are available in the real hardware pool to be associated with the virtual alias (per virtual guest machine).

There can be only 254 aliases per pool, and a limit of 16,000 pools per image. Also, the command is restricted to full pack minidisks.

Example 4-1 Query PAV command

Query Virtual PAV Command

Dedicated

QUERY VIRTUAL PAV ALL

HYPERPAV BASE 0200 ON E100 YAC001 POOL 1
HYPERPAV ALIAS 0201 ON E101 POOL 1

Minidisks
QUERY VIRTUAL PAV ALL
HYPERPAV BASE 0200 ON E100 YAC001 ASSIGNED E100 POOL 1
HYPERPAV ALIAS 0201 ASSIGNED E101 POOL 1

4.1.3 User directory entry

In this section, we explain how to add entries to the user directory for the use and exploitation of HyperPAV and aliases.

Important: You would never directly edit the user directory if your installation uses DIRMAINT.

CP cannot read the source directory, so the **DIRECTXA** utility is used to create a CP readable version of the user directory (USER DIRECT). This compiles the file and places a copy of it in the area that has been allocated as DRCT on one of the CP-owned volumes. Use the following format of the command to execute the **DIRECTXA** utility:

```
DIRECTXA fn ft
```

Where: “fn” is the file name of your directory (`USER` is the DEFAULT fn), and “ft” is the file type of your directory (`DIRECT` is the DEFAULT ft).

1. To access the user directory, log on to z/VM with a user ID that has system administrator privileges. In this case, we used MAINT620.

2. From the command line, enter: **DIRECTXA USER DIRECT**.
3. Enter the **filelist user direct a** command and then use XEDIT to add, delete, or change directory entries as required.
4. The following directory statements would result in creating a virtual HyperPAV alias minidisk:

   COMMAND DEFINE HYPERPAVALIAS vdev **FOR BASE** basedev

   Where: “vdev” is the virtual device number of the alias HyperPAV you are defining and “**FOR BASE** basevdev” is the device number of an existing virtual base HyperPAV.
5. After additions are made to the directory, save the file, and exit.
6. Before bringing the directory online, check the directory for errors by using the following command:

   DIRECTXA USER (EDIT

   Where: USER” is the file name of the entire z/VM directory. If you made a syntax error, you receive an error message.

   When you verify that the directory file is correct, replace the old directory with the updated directory by entering the following command: **directxa filename**.

   After the directory is updated, directory changes for any virtual machines currently logged on to the system do not take effect until the user logs off the system and then logs back on.

4.1.4 Making HyperPAV available to Linux

After updating the user directory with the appropriate entries and making the updated directory available to z/VM, the next step is to make the HyperPAV available to the Linux guests. To use HyperPAV with z/VM DASD volumes, it must be defined as a full pack, cylinder 0-end with either:

- DEDICATE vDEV rDEV statements for the BASE and any aliases *or* the MDisk vDEV 3390 DEVNO rDEV mr statement.
- DEDICATE 9902 **9902** (as it would be entered in the user directory) *and* DEDICATE 99BF **99BF** (defines one alias in the user directory).

This uses the entire DASD volume including cylinder 0, which in most cases, has not been fully used by Linux. When we define a volume for a Linux guest, we define a full pack minidisk starting at cylinder 1 and going to the end to preserve the z/VM volume label and drive characteristics.

Perform the following steps to confirm your HyperPAV environment:

1. Set your base device online by using the **chccwdev -e 0.0.9902** command from your Linux on System z guest. An alias device might not be represented in the Linux virtual file system called *sysfs* until the base device is set online. If your Linux system runs as a z/VM guest, each device has a sysfs attribute, *use_diag*, that by default is set to 0. Do not change this attribute to 1 for any of the aliases.
2. Set the alias to online by typing **chccwdev -e 99BF** on the Linux command line.
3. An optional step is to confirm the mapping of the base and alias devices by listing the DASDs with the **lsdasd -u** command.

In a HyperPAV environment, alias devices are not dedicated to a particular base device but can be used for any base device in the same logical subsystem on the storage system. Instead of a device identifier, alias devices have xx as the fourth section of their user ID (uid).

An alias is eligible for a base device if the first three sections of its uid match the first three sections of the uid of the base device, as shown in Example 4-2.

Example 4-2 HyperPAV example

```
lsdasd -u
Bus-ID Name UID
============================================
0.0.9902 dasdb IBM.75000000092461.2a00.1a
0.0.9903 dasdf IBM.75000000092461.2a00.1a
0.0.99BF alias IBM.75000000092461.2a00.xx
...
```

In this example, `0.0.99BF` is an alias that is eligible for base devices `0.0.9902`, and 0.0.9903.

You would format the base device with the command issued from your Linux guest:

```
dasdfmt -b 4096 -p -l YC000 -f /dev/dasdb
```

You are now ready to work with the base devices just as you would without PAV. The DASD device driver automatically uses the aliases as the need arises.

Noted restrictions:

- CMS does not support virtual aliases (whether traditional PAV or HyperPAV). Defining these virtual devices under CMS can cause damage similar to that caused by issuing multi-write (MW) links.
- A virtual alias (whether traditional PAV or HyperPAV) cannot be restarted.
- PAV aliases (whether traditional PAV or HyperPAV) cannot be used as VM installation volumes (for example, do not use for the SYSRES volume).
- VM paging and spooling operations do not take advantage of PAVs or HyperPAVs traditionally. It is recommended that PAGE and SPOOL areas be placed on DASD devices that are dedicated to this purpose. For PAGE devices, use 3390 Mod 3 as the preferred method of efficiently handling paging to DASD. SPOOL devices can use 3390 Mod 3 or Mod 9. Space constraints on the SPOOL device can be minimized by the use of 3390 Mod 9.
- Virtual HyperPAV devices can be defined only as full-pack minidisks.
- Diagnoses x18, x20, xA4, x250, and the *BLOCKIO system service do not support HyperPAV alias devices because there is no means for specifying the associated base. An attempt to do so results in an error.

4.2 ECKD versus SCSI

In this section, we describe the choice of using extended count key data (ECKD) or SCSI, or both attached devices.

The choices that are made regarding ECKD devices or SCSI are based on performance and cost considerations. Generally, when selecting ECKD, consider when minimal administration effort and low CPU requirements are a high priority. ECKD devices with HyperPAV are good candidates. When maximum performance should be reached, FCP devices are good candidates, but it requires additional CPU capacity to drive the workload.

We describe the characteristics of ECKD and SCSI devices. For more information about which type of devices to use with a database management system, see the following website:

http://public.dhe.ibm.com/software/dw/linux390/perf/Performance_considerations_for_databases_on_Linux_on_System_z.pdf

4.2.1 ECKD over FICON

Following are some of the I/O processing characteristics when using ECKD over FICON:

- In an ECKD environment, the mapping of the host subchannel to DASD is 1:1. The multiple paths are handled in the channel subsystem.
- Serialization of I/Os per subchannel.
- I/O requests are queued in the Linux guest.
- Disk blocks sizes are 4 KB.
- High availability by FICON path groups is realized.
- Load balancing by FICON path groups and PAVs.

Figure 4-2 illustrates the general layout for ECDK over FICON channel.

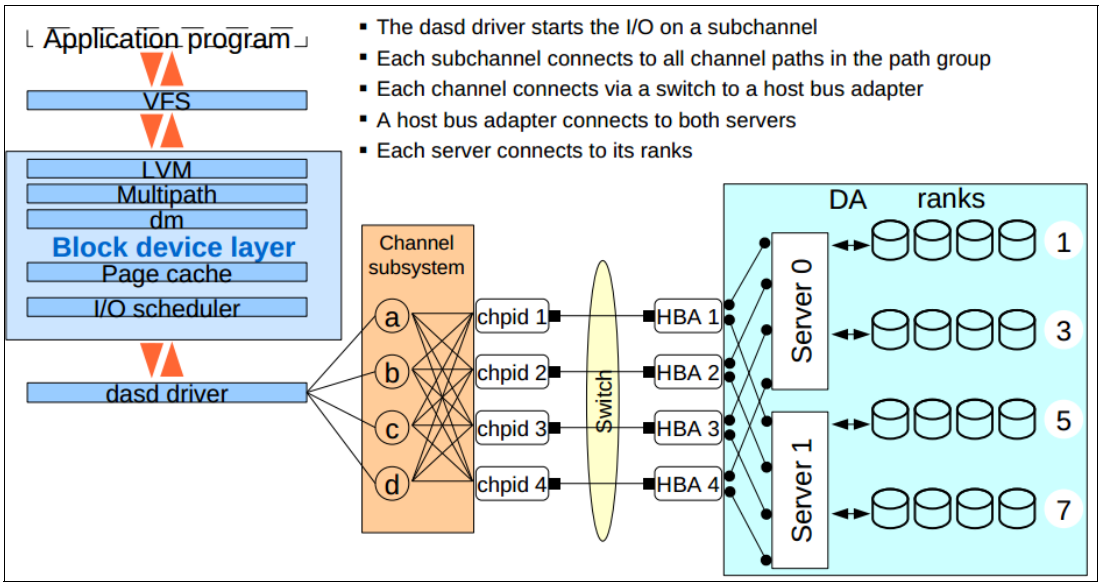


Figure 4-2 General layout for ECKD over FICON

For more information about how to use ECDK devices in z/VM and Linux, see *z/VM V6R2 Getting Started with Linux on System z,* SC24-6194-02.

ECKD with HyperPAV

Using ECKD devices with HyperPAV can improve performance. Following are some characteristics of ECKD with HyperPAV:

- Virtual file system (VFS) sees one device
- The DASD driver sees the real device and all alias devices
- Each alias device uses its own subchannel
- Load balancing with HyperPAV and static PAV is done in the DASD driver. The aliases need only to be added to Linux. Load balancing works better than on the device mapper layer
- Less additional processor cycles are needed than with Linux multipath

A potential point of performance degradation is the fact that only one disk is used in the storage server. This implies the use of only one rank, one device adapter, and one server.

4.2.2 SCSI over FCP

The Linux zFCP device driver adds support for Fibre Channel Protocol (FCP)-attached SCSI devices to Linux on System z. FCP is an open, standards-based alternative and supplement to existing FICON connections. Following are important I/O characteristics of this type of configuration:

- FCP is faster than FICON
  - Several I/Os can be issued against a LUN immediately (asynchronous I/O)
  - No ECKD emulation overhead
- I/O queues occur in the FICON Express card or in the storage server
- No disk size restrictions
- Disk blocks are 512 bytes
- High availability is provided by Linux multipathing, type failover, or multibus, which are managed by either the z/VM or Linux operating systems
- Load balancing is provided via Linux multipathing, type multibus
- Dynamic configuration

  Add new storage subsystems without IOCDS changes

Figure 4-3 on page 61 illustrates how the SCSI/FCP layout looks.

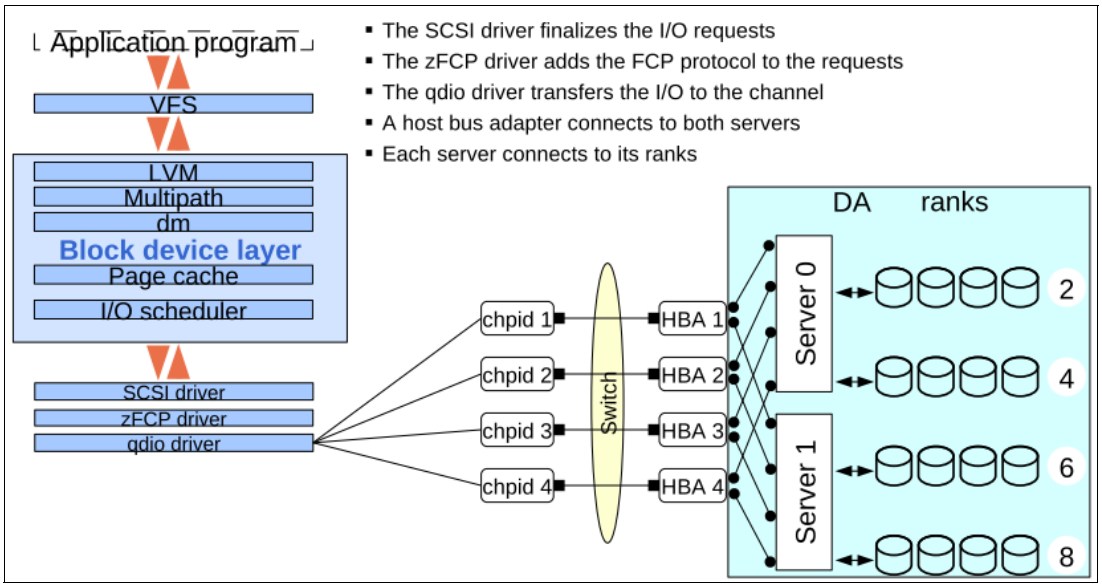


Figure 4-3 FCP/SCSI layout

To define and configure SCSI disks in z/VM and Linux on System z, see the following sources:

- *Fibre Channel Protocol for Linux and z/VM on IBM System z*, SG24-7266
- RHEL6: Part III. IBM System z Architecture - Installation and Booting

  https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Installation_Guide
- SLES 11 Deployment guide

  https://www.suse.com/documentation/sles11/book_sle_deployment/data/book_sle_deployment.html

Emulated FBA disk versus direct-attached SCSI

z/VM supports SCSI FCP disks for both system and guest use:

- Direct-attached SCSI: SCSI disks can be used directly by a Linux guest operating system when an FCP subchannel is dedicated to a guest. A Linux guest machine must contain its own SCSI device driver. The guest uses queued direct I/O (QDIO) operations to communicate with the device. For more information about SCSI FCP support, see the following site:

  http://publib.boulder.ibm.com/infocenter/zvm/v6r2/topic/com.ibm.zvm.v620.hcpf2/hcsf9c11255.htm?path=2_1_4_4_48#wq318
- Emulated fixed-block architecture (FBA): SCSI disks can also be used as emulated 9336 model 20 fixed-block-architecture (FBA) disks. It can be used by Linux guest machines and z/VM system volumes, such as page and spool volumes. For more information about emulated FBA disks on SCSI disks, see the following site:

  http://publib.boulder.ibm.com/infocenter/zvm/v6r2/topic/com.ibm.zvm.v620.hcpa5/hcsg0c11217.htm#wq1347

Figure 4-4 shows the ways that z/VM can be used to manage SCSI disks. Operating systems manage multipathing in the FCP environment. So when it is an emulated FBA disk architecture, z/VM manages the multipathing. When Linux uses the SCSI disks directly, the Linux operating system manages the multipathing.

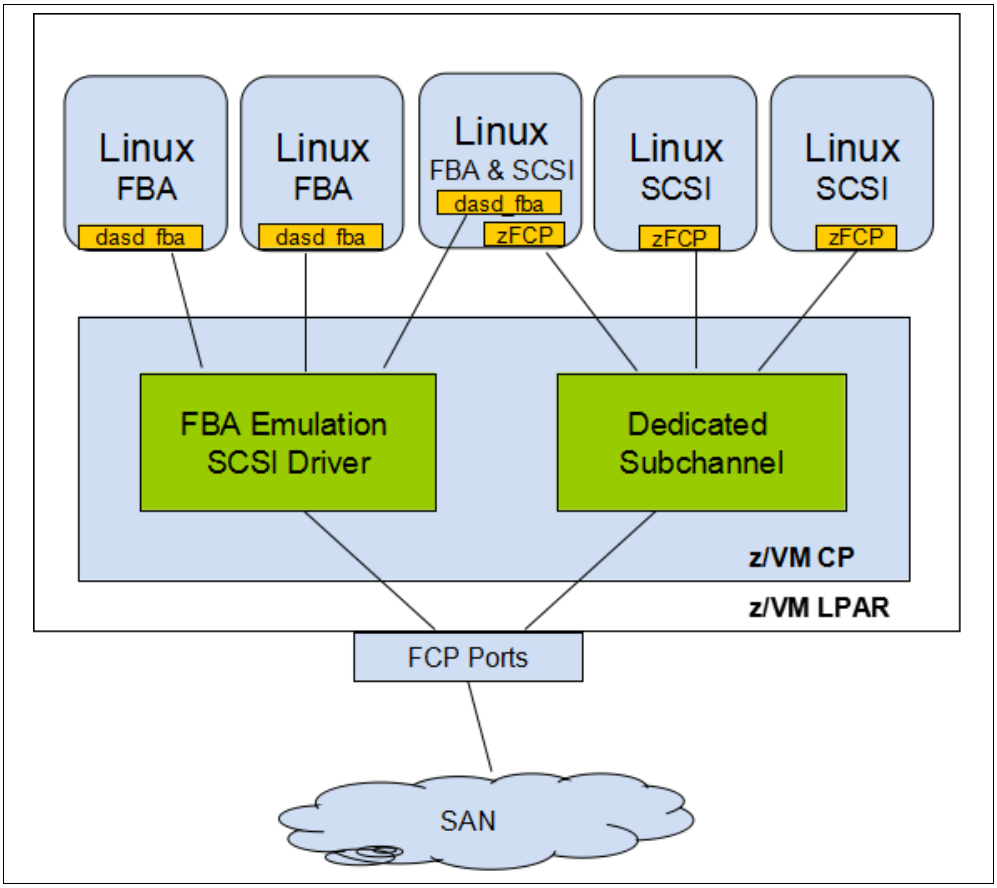


Figure 4-4 z/VM SCSI management approaches

Although both methods are using FCP architecture, be aware that the performance of the two configurations is different. In general, using FBA emulation requires a high number of I/O work units running concurrently to achieve any significant throughput. Figure 4-5 on page 63 shows a comparison of the standard Linux FCP driver with FBA emulation with 50-50 read/write mix workload, both running 48 concurrent work units. For 100% read or write, the results are similar.

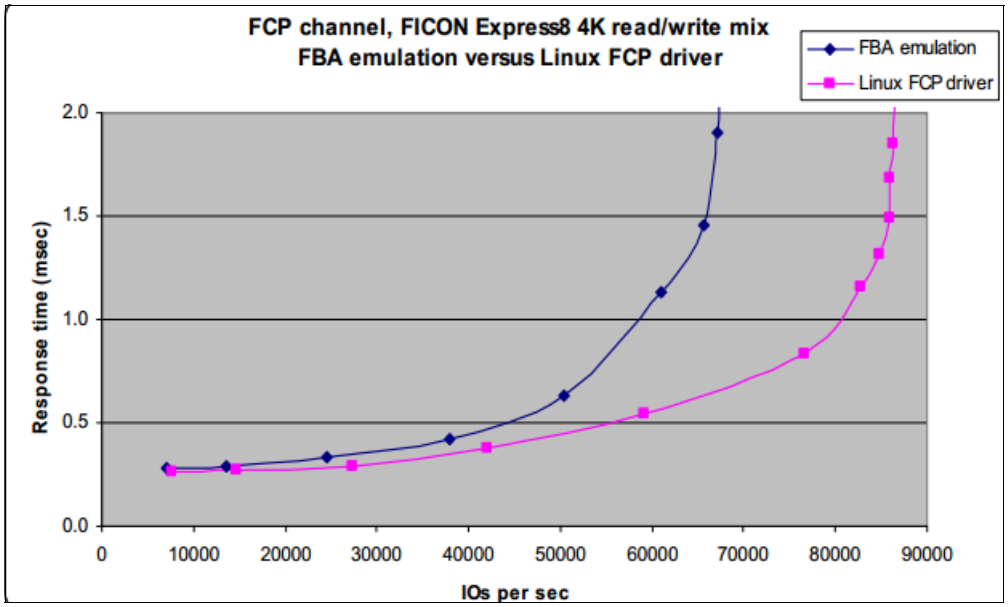


Figure 4-5 Emulated FBA versus direct-attached SCSI: Response time

As shown in Figure 4-6, for a 50-50 read/write mix, FICON Express8 with FBA emulation achieved 635 MBps versus 776 MBps with the Linux FCP driver. For 100% reads, FICON Express8 with FBA emulation achieved 800 MBps, as did the Linux FCP driver. For 100% writes, FICON Express8 with FBA emulation achieved 500 MBps versus 746 MBps with the Linux FCP driver.

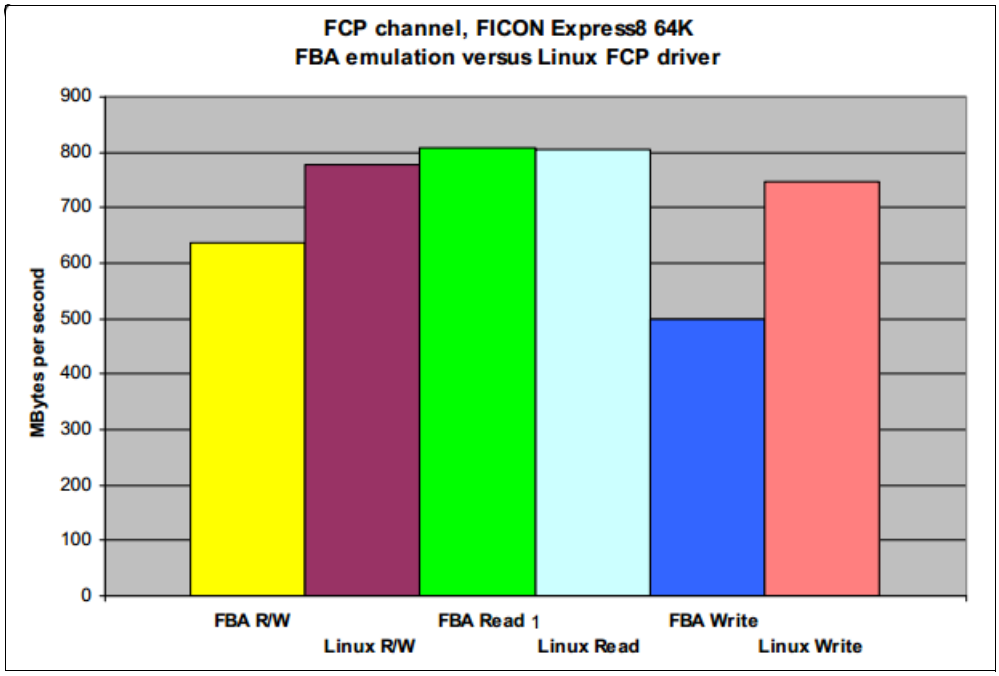


Figure 4-6 Emulated FBA versus direct-attached SCSI: Throughput

Figure 4-7 shows that host CP usage using FBA was higher than with the Linux FCP driver, when fewer 4 KB transfers were being handled. This figure demonstrates that the more work units running concurrently, the less performance difference there was between the two approaches.

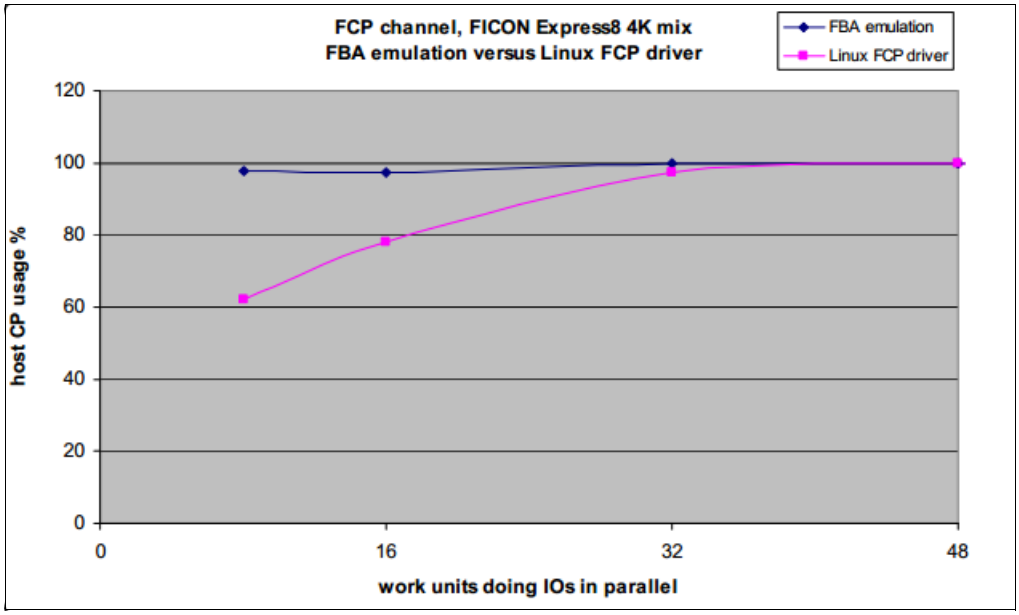


Figure 4-7 Emulated FBA versus direct-attached SCSI: CPU usage

In general, when very high performance is required, direct-attached SCSI is a good choice for Linux guest machines. However, FBA still is a good choice for situations such as the following:

- In SCSI-only Linux on System z installations, FBA emulation is required for z/VM system volumes, paging volumes, and minidisks of server virtual machines
- When minidisk cache usage provides hit ratios that are sufficient to avoid enough I/O to mitigate the processor costs of FBA emulation
- When minidisks are used for administrative purposes, such as cloning and provisioning
- For some disaster recovery environments, some people find it easier to manage the configuration (for example, WWPNs) at the z/VM host level

Based on the preceding results, select the proper SCSI type that is based on your business applications environment and monitor the results first in a test environment. Remember, to keep the production environment's performance optimal, it requires an understanding of the specific production workload.

4.2.3 Summary

This section summarizes some of the general suggestions regarding ECKD and SCSI disk.

FICON/ECKD:

- Leverage storage pool striped disks (no disk placement)
- Enable HyperPAV (SLES11, RHEL6)
- Large volume (future distributions)
- Enable High Performance FICON (zHPF)

FCP/SCSI:

  - Enable Linux LV with striping (disk placement)
  - Configure multipathing with failover

Whichever disk type is selected, following are some general suggestions regarding disk planning for a production environment:

- Use as many paths as possible (CHPID → host adapter)
- Spread the host adapters that are used across all host adapter bays
  - For ECKD, switching of the paths is done automatically
  - FCP needs a fixed relation between disk and path
- Use Linux multipathing for load balancing
- Select disks from as many ranks as possible
- Switch the rank for each new disk in a logical volume
- Switch the ranks that are used between servers and device adapters
- Avoid reusing the same resource (path, server, device adapter, and disk) as long as possible

The goal is to get a balanced load on all paths and physical disks. In addition, striped Linux logical volumes and storage pool striping might help to increase the overall throughput.

5

Network planning considerations

In this chapter, we describe the network planning considerations necessary for the transition from a test or development environment to a production environment. The discussion points that we cover in this chapter help you decide which method best fits your networking needs. We describe HiperSockets, when they should be deployed, and setup considerations. We describe the virtual switch (VSWITCH): the *when* and *why* a VSWITCH should be deployed in a production environment and the benefits of its use. We describe the z/VM guest local area network (LAN) and its support. The Open Systems Adapter (OSA), its configuration of the OSA, and benefits of the OSA and availability characteristics are also described. We also illustrate the OSA failover capability and the use of link aggregation in support of bandwidth demands and recovery of a failed link. We show a configuration of a native Linux running in a logical partition (LPAR) and using multiple, directly attached OSA cards. We describe the OSA Express4S and the 10 GB adapter.

The following topics are covered:

- HiperSockets description
- VSWITCH use
- OSA configuration
- OSA configuration using Link Aggregation
- Guest LAN support

As described in “Network management” on page 17, we presented some widely used networking options to connect operating systems to client networks and guest-to-guest communications.

© Copyright IBM Corp. 2013. All rights reserved.

5.1 Network overview

In this section, we describe the types of networking facilities available on System z and z/VM

- OSA
- OSA with Link Aggregation
- HiperSockets
- Guest LAN
- Virtual switch (VSWITCH)

5.1.1 Open Systems Adapters

The Open Systems Adapter (OSA) is a hardware network controller that you can install in a mainframe I/O cage.

The option to attach an OSA adapter to the Linux server is available in any scenario. This is the only option available for running Linux native in its own LPAR. In this case, the OSA adapter can be shared between Linux LPARs.

Figure 5-1 shows an architecture with four Linux servers in a single System z server. There are four OSA ports available and the two LPARs share two OSA ports. In this case, Linux is responsible for the administration of the NICs. Linux administrators have to set up the bonding module driver to handle multipath and failover capabilities.

For more information, see the presentation “Networking with Linux on System z”, which can be found at the following website:

http://www.vm.ibm.com/education/lvc/LVC1109C.pdf

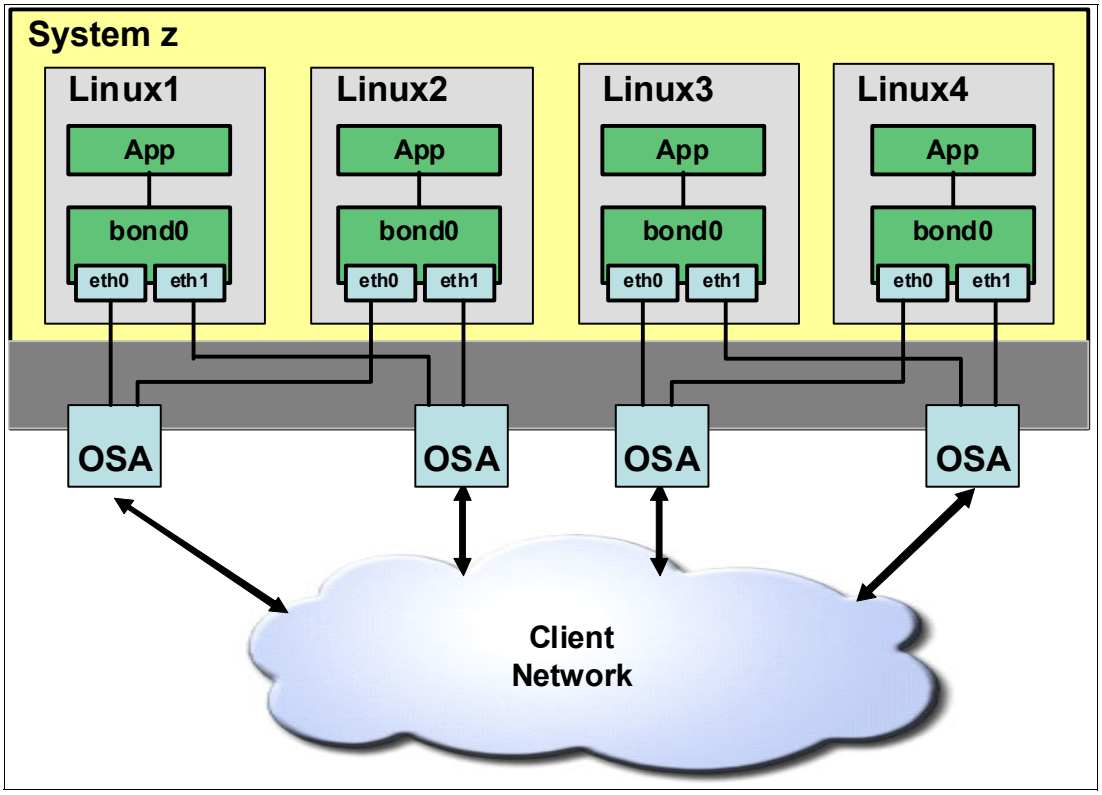


Figure 5-1 Linux using an OSA adapter

5.1.2 OSA with Link Aggregation

Link Aggregation is a computer networking term that describes the various methods of bundling multiple network connections in parallel to increase bandwidth throughput and provide redundancy. The combination of multiple physical Ethernet ports to form a virtual logical link is called a *link aggregation group* (LAG). Cisco called it *EtherChannel*; Juniper called it *Aggregated Ethernet*.

There are two methods in a LAG configuration:

- Using a negotiation protocol
- Static

The common protocols used are Port Aggregation Protocol (PAgP) and Link Aggregation Control Protocol (LACP).

PAgP is Cisco's proprietary networking protocol for channel negotiation between two Cisco switches or between a Cisco switch and a server that supports PAgP (some server/NIC manufacturers license this feature from Cisco). LACP, also known as *IEEE 802.3ad*, is an open standard that is supported by most networking vendors. Because LACP covers all the functionality of PAgP, the only reason to use PAgP would be for compatibility with earlier version purposes. According to Cisco:

"*LACP is the recommended solution for configuration of port channel interfaces to the Nexus 7000, as NX-OS does not currently support PAgP.*"[1]

LACP is a vendor-neutral standard that provides a method to control the bundling of several physical ports together to form a single logical channel. LACP is a negotiation protocol that allows a network device to establish a multilink channel by sending Link Aggregation Control Protocol Data Unit (LACPDU) packets to the peer device. Both devices must be running LACP and are directly connected.

The following requirements must be met before a multilink channel can be formed:

- Same speed/duplex on each port.
- Access VLAN (if not trunked).
- Same trunking type (VLAN and native VLAN (if trunked)) is allowed.
- Each port must have the same Server Time Protocol (STP) cost per VLAN within the multilink channel.
- No SPAN or monitoring ports.

A typical Cisco switch supports three modes of EtherChannel:

- LACP: Channel-group x mode *active*
- PAgP: Channel-group x mode *desirable*
- Static: Channel-group x mode *on*

Other vendors, such as Juniper Networks, support LACP and static mode.

Unless the network is managed by highly qualified professionals and governed by strict change management control processes, avoid the use of static configured LAG. The use of LACP and PAgP helps in detecting and mitigating configuration errors and the impact of using them is justified. Static mode should be used only as a last resort when there is no other alternative.

[1] Source: http://www.cisco.com/en/US/docs/solutions/Enterprise/Data_Center/nx_7000_dc.html

5.1.3 IBM HiperSockets

HiperSockets technology is supplied as part of the System z hardware with near zero latency at memory speed.

Note: At least one LPAR with a direct-hosted operating system (OS) or an LPAR with a hypervisor (z/VM) hosting an OS must be implemented on a System z server for any applications to execute.

HiperSockets is a technology that provides high-speed Transmission Control Protocol/Internet Protocol (TCP/IP) connectivity between servers within a System z. This technology eliminates the requirement for any physical cabling or external networking connections among these virtual servers. It works similar to an internal LAN. HiperSockets is very useful if you have a large data flow among virtual servers.

HiperSockets uses *internal queued input/output* (iQDIO) at memory speeds to pass traffic among these virtual servers and is a Licensed Internal Code (LIC) function that emulates the Logical Link Control (LLC) layer of an OSA-Express QDIO interface.

HiperSockets can be used by z/VM, Linux as a guest of z/VM, or Linux running in its own LPAR. The following is a list of HiperSockets benefits:

- Cost savings

  You can use HiperSockets to communicate among consolidated servers in a single processor. Therefore, you can eliminate all the hardware boxes running these separate servers. With HiperSockets, there are zero external components or cables to pay for, to replace, to maintain, or to wear out. The more consolidation of servers, the greater your savings potential for costs that are associated with external servers and their associated networking components.

- Simplicity

  HiperSockets is part of z/Architecture technology, including QDIO and advanced adapter interrupt handling. The data transfer itself is handled much like a cross address space memory move, using the memory bus. HiperSockets is application transparent and appears as a typical TCP/IP device. Its configuration is simple, making implementation easy. It is supported by existing, management, and diagnostic tools.

- Availability

  With HiperSockets, there are no network hubs, routers, adapters, or wires to break or maintain. The reduced number of network external components greatly improves availability.

- High performance

  Consolidated servers that have to access corporate data residing on the System z can do so at memory speeds with latency close to zero, by bypassing all the network overhead and delays. Also, you can customize HiperSockets to accommodate varying traffic sizes. With HiperSockets, you can define a maximum frame size according to the traffic characteristics for each HiperSockets. In contrast, Ethernet LANs have a maximum frame size that is predefined by their architecture.

- Security

  Because there is no server-to-server traffic outside the System z, HiperSockets has no external components, and therefore it provides a very secure connection.

  However, there is the option to define a HiperSockets Bridge. z/VM virtual switch was enhanced to transparently bridge a Linux virtual server network connection on a

HiperSockets LAN segment. This bridge allows a single HiperSockets guest virtual machine network connection to directly communicate with external network hosts through the virtual switch OSA UPLINK port. What this means is that this feature provides the ability to communicate with hosts residing external to the central processor complex (CPC). This feature is available for those clients who need to use HiperSockets in a z/VM SSI environment between different System z servers.

All security features, such as firewall filtering, are available for HiperSockets interfaces in the same way as they are with other Internet Protocol network interfaces.

Additional information

The following publications provide more information:

- The IBM Redbooks publication, *HiperSockets Implementation Guide*, SG24-6816, provides more details about HiperSockets.
- The IBM Redpaper™ publication*, Advanced Networking Concepts Applied Using Linux on System z: Overview of Virtualization and Networking*, TIPS0982, provides an overview of IBM System z virtualization and networking that uses z/VM Guest LANs and VSWITCHs.
- *z/VM Connectivity,* SC24-6174 provides details about z/VM network capabilities.

5.1.4 z/VM Guest LANs

Guest LANs are virtual networks that are used to connect Linux guests in the same z/VM LPAR. They facilitate communication between guests without any additional hardware. Guest LANs can be defined as QDIO or HiperSockets types.

Although the Guest LAN method is still available and used in some scenarios, do not use it for complex environments because this technology requires a z/VM service machine (TCP/IP), or a Linux guest that is acting as a router, to forward packages to the outside network. For performance purposes and complex networks, a separate hardware device (such as a Cisco or Juniper product) should act as the router and provide this communication.

For complex network environments where there is intense network traffic activity and external connectivity is required, virtual switches are the best choice.

5.1.5 z/VM virtual switches

The virtual switch (VSWITCH) method allows Linux on System z guests to connect over the network. This method is both efficient and secure. In addition, the virtual switches support VLANs and Link Aggregation (IEEE 802.1Q and 802.3ad).

Figure 5-2 on page 72 shows an example where two OSA adapters are being shared by two z/VM LPARs. This basic network architecture requires at least two OSA adapters for failover capabilities. Each virtual switch uses one OSA adapter for all network traffic and the other for failover. The configuration of the second VSWITCH should be reversed. In this case, both network channels are in use in a reliable architecture.

The configuration of virtual switches changes according to the network architecture. Use Level 2 features, plugging OSA adapters into trunk ports. This capability provides the ability to choose PortType Access or PortType Trunk for each Linux virtual network interface card (vNIC).

Figure 5-2 shows an example of a z/VM VSWITCH basic network.

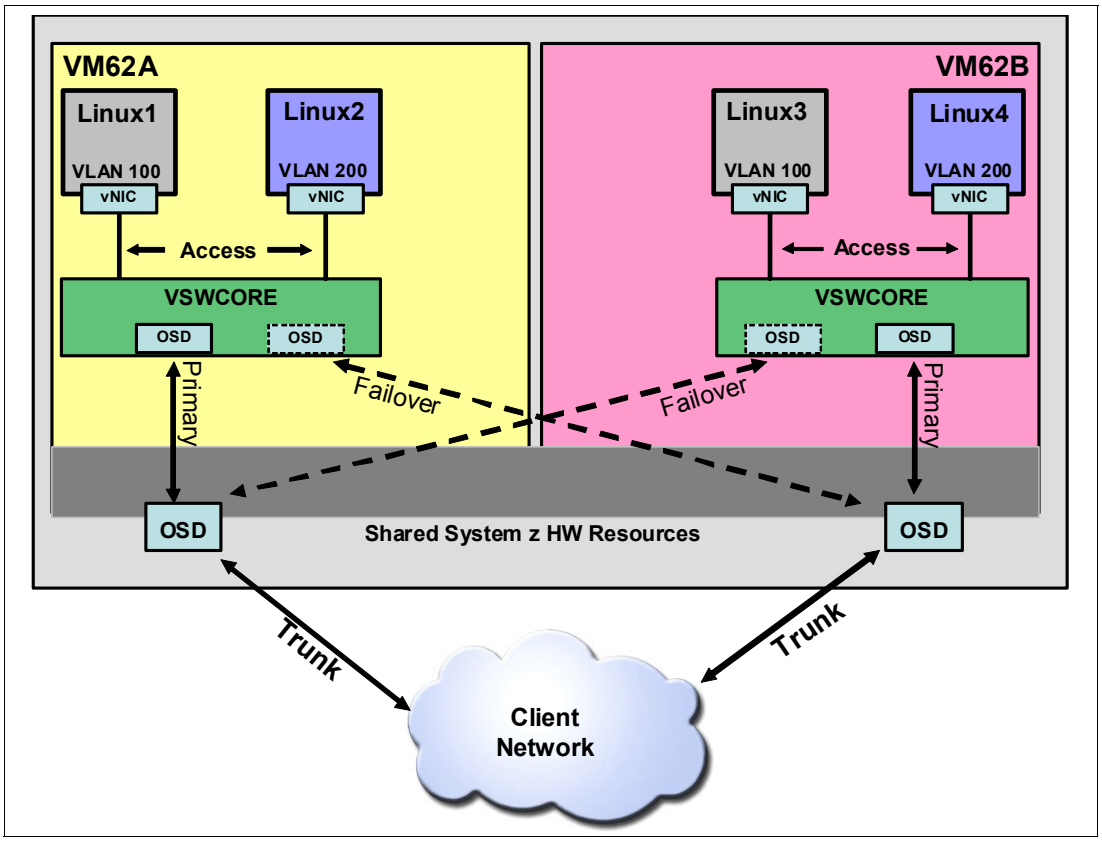


Figure 5-2 z/VM VSWITCH basic network

For those Linux virtual servers that need to be connected in more than one VLAN, there are two options:

- A PortType Trunk can be defined for this specific server. The z/VM administrator can define which VLANs this server has access. The Linux administrator has to set up Linux network configuration to get access to all needed VLANs. An example of this option can be seen in Figure 5-3 on page 73, where Linux 4 has a single vNIC PortType Trunk port to connect to VLAN 200 and 250. In this case, Linux is responsible to tag all its packages.
- The second option is the definition of two or more vNICs PortType Access for a single Linux virtual server. It is possible to specify for each vNIC which VLAN it is tagged by the VSWITCH and sent to the client network. In this case, the VLAN configuration is transparent to the Linux server. An example of this option can be seen in Figure 5-3 on page 73, where Linux 2 has two vNICs PortType Access to connect to VLAN 200 and 250. In this case, VSWITCH is responsible to tag all Linux 2 packages.

z/VM VSWITCH also supports IEEE 802.3ad Link Aggregation Control Protocol (LACP). Aggregating links with a partner switch box allows multiple OSA-Express adapters to be deployed in transporting data between the switches. This feature can benefit clients that plan to have multiple Linux virtual servers on System z. This configuration increases bandwidth and near seamless recovery of a failed link within the aggregated group.

Figure 5-3 shows an example of a VSWITCH configuration with four LACP ports.

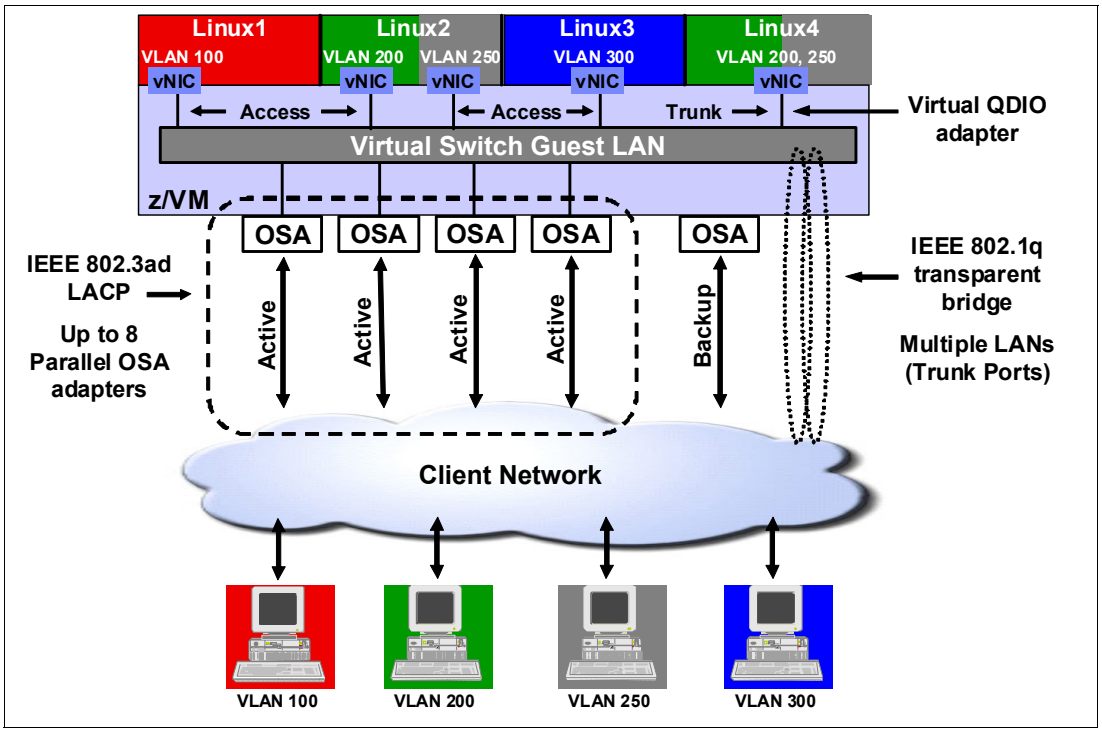


Figure 5-3 z/VM VSWITCH using VLANs and LACP

For higher throughput, System z servers have 10 Gigabit (10 GbE) OSA-Express4S as an optional feature. z/VM VSWITCH LACP implementation provides the ability to concatenate up to eight ports.

Note: z/VM 6.2 LACP implementation requires exclusive and same type OSA ports.

Limitations on either connectivity or data throughput are related to the OSA-Express cards. For more details about the OSA-Express devices, see the *OSA-Express Implementation Guide*, SG24-5948.

External networking with an OSA is accomplished by two main methods:

- Through the wired network external to and from the System z computer.
- Using QDIO direct between LPARS in a single mainframe.

6

Linux planning considerations

This chapter describes Linux guest planning and provides a few tips that can guide you in moving your guests into production. Topics such as security, high availability, compliance, performance, and best practices, are addressed. This chapter is based on our own experiences and technical references.

© Copyright IBM Corp. 2013. All rights reserved.

6.1 File system management

Expressions such as high availability and reliability are always present on service level agreements. But, what do you actually do if your disk is running out of space? How can you avoid an outage and keep your system and applications online?

Using the Logical Volume Manager (LVM) in a Linux guest running on System z platform is one way to answer this question. By using the LVM, you gain the reliability, high availability, and flexibility to address file system space issues. You have this flexibility because you are able to add disks, create, extend, and reduce file systems, all with the operating system and applications online.

LVM allows you to merge all disks, called physical volumes (PVs), and attach the disks in a volume group (VG), which is partitioned into logical volumes.

For more information about how LVM works and how to create file systems by using LVM, see *The Virtualization Cookbook for z/VM 6.3, RHEL 6.4 and SLES 11 SP3*, SG24-8147.

6.1.1 File system hierarchy with LVM

Create an LVM in your Linux on System z guest to take advantage of the ability to resize partitions dynamically as well as combining multiple logical hard disk drives into logical physical devices. In this section, we describe the file system hierarchy when using the LVM.

Do not include the root file system in the LVM structure because, if for any reason the LVM fails, the operating system will not boot. A 350 MB disk is enough to store the root file system. Table 6-1 shows a typical file system hierarchy by using an LVM.

Table 6-1 File system hierarchy using LVM

Mount point	Volume group	Logical volume	Size
/usr/	system	usr-lv	2.5 G
/tmp/	system	tmp-lv	400 MB
/var/	system	var-lv	1.5 GB
/srv/	system	srv-lv	16 MB
/opt	system	opt-lv	512 M
/home	system	home-lv	512 M

Note: The naming convention that is shown here is what was used for this IBM Redbooks publication. You can create your own naming convention.

This structure is useful from the system administration point of view because you can easily identify the file systems and check their usage. It also helps to set up monitoring tools to check each file system separately with their own thresholds, as shown in Example 6-1 on page 77.

Example 6-1 Example of file system visualization using LVM

```
# df -hP
Filesystem            Size  Used Avail Use% Mounted on
/dev/dasda1           516M  261M  229M  54% /
devtmpfs              2.0G  212K  2.0G   1% /dev
tmpfs                 2.0G     0  2.0G   0% /dev/shm
/dev/mapper/system-opt--lv  4.3G  3.2G  918M  78% /opt
/dev/mapper/system-usr--lv  4.8G  3.4G  1.2G  75% /usr
/dev/mapper/system-var--lv  1.5G  229M  1.2G  16% /var
/dev/mapper/system-tmp--lv  388M  106M  263M  29% /tmp
/dev/mapper/system-srv--lv   78M  4.2M   70M   6% /srv
/dev/mapper/system-home--lv  291M   17M  260M   6% /home
/dev/mapper/appvg-db2inst1  2.0G  1.8G  164M  92% /home/db2inst1
/dev/mapper/appvg-db2fenc1   20M  4.1M   15M  23% /home/db2fenc1
/dev/mapper/appvg-dasusr1   31M  8.3M   22M  28% /home/dasusr1
/dev/mapper/appvg-db2logs   31M  4.1M   26M  14% /db2logs
/dev/mapper/yum-repo  5.5G  3.2G  2.0G  63% /srv/www/htdocs/yum/repo
/dev/mapper/yast-repo  5.5G  2.9G  2.4G  56% /srv/www/htdocs/yast/repo
```

Leave some space available in the volume group (as shown in Example 6-2) that can be used in case of emergency because even if it is not enough to solve the issue, it gives you time to add the required amount of disk space to extend the file system properly.

Example 6-2 Free space in the LVM

```
# vgs
  VG     #PV #LV #SN Attr   VSize  VFree
  appvg    2   4   0 wz--n-  4.51G 2.43G
  system   4   6   0 wz--n- 13.64G 2.21G
```

6.1.2 Disk naming convention

From a system administration perspective, simplifying tasks that help to identify a missing disk, or know which label should be used to add a new disk is a welcomed addition.

When using Linux on System z, you are able to label the disks at your convenience, so take advantage of this feature and create a disk naming standard to help the system administrators easily identify the disks, as shown in Table 6-2.

Table 6-2 Example of disk name convention

Disk range	Destiny
100-1 FF	Swap disks
200-2 FF	Non-lVM disks
300-3 FF	System volume group
400-4 FF	Customer or application volume group

When running the **lsdasd** command, as shown in Example 6-3, you can then easily identify the disk by its *Bus-ID* and you know its purpose.

Example 6-3 Use Table 6-2 on page 77 to identify the disks that are listed

```
# lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
==============================================================================
0.0.0202   active      dasda     94:0    ECKD  4096   523MB     134100
0.0.0201   active      dasdb     94:4    ECKD  4096   703MB     180000
0.0.0203   active      dasdc     94:8    ECKD  4096   7042MB    1802880
0.0.0300   active      dasdd     94:12   ECKD  4096   2311MB    591840
0.0.0301   active      dasde     94:16   ECKD  4096   2311MB    591840
0.0.0302   active      dasdf     94:20   ECKD  4096   2311MB    591840
0.0.0400   active      dasdg     94:24   ECKD  4096   2311MB    591840
0.0.0401   active      dasdh     94:28   ECKD  4096   2311MB    591840
0.0.0402   active      dasdi     94:32   ECKD  4096   5625MB    1440180
0.0.0403   active      dasdj     94:36   ECKD  4096   5625MB    1440180
```

6.2 Network management

Linux on System z has many advantages that are related to network configuration. In Chapter 5, “Network planning considerations” on page 67, there is description of the type of network layouts that you can set up with Linux on System z and z/VM systems. In this section, we describe some configuration practices and setup.

6.2.1 Maximum transmission unit

Maximum transmission unit (MTU) is the maximum size (in bytes) of one packet of data that can be transferred in a network. All hops (a portion of a signal's journey from source to receiver) that are a part of the communication must be able to receive and transmit at the same MTU. However, if one of the hops in a network path is using a lower MTU size from the originating host, the package is retransmitted in a smaller size.

The default MTU size for Ethernet devices is 1514 bytes, and in a distributed environment, it is necessary to configure all Linux servers and switches/routers to communicate at the same MTU size to take advantage of it. In a Linux on System z environment, it is a relatively low effort to set up all internal communications using 8992 bytes of MTU size.

By adjusting the MTU sizes, you can improve the performance of your application and database servers. For example, typically, the larger the packet size, the better the performance because fewer packets are needed to communicate between the driver and the database. Fewer packets means fewer network round trips to and from the database. Fewer packets also require less disassembly and reassembly, and ultimately, use less CPU.

It is possible to set up multiple application servers in a clustered environment where all of the servers are using Heartbeat and the database servers are using internal virtual switches. In Figure 6-1 on page 79, all these servers are using an MTU of 8992 bytes. All application network traffic goes through the VSWITCH named *VSWITCH2*. The load balance servers (LB01 and LB02) are configured as active/passive high-availability services and are attached directly to an OSA. All outside traffic from an application is received by a designated address. The package is received and sent to one of the application servers (APP01, APP02, or

APP03) based on a balance algorithm. The database servers (DB01 and DB02) are configured as an active/passive cluster and receive and send the requests to the application servers using VSWITCH2. VSWITCH1 is used for backup, monitoring, and administration services.

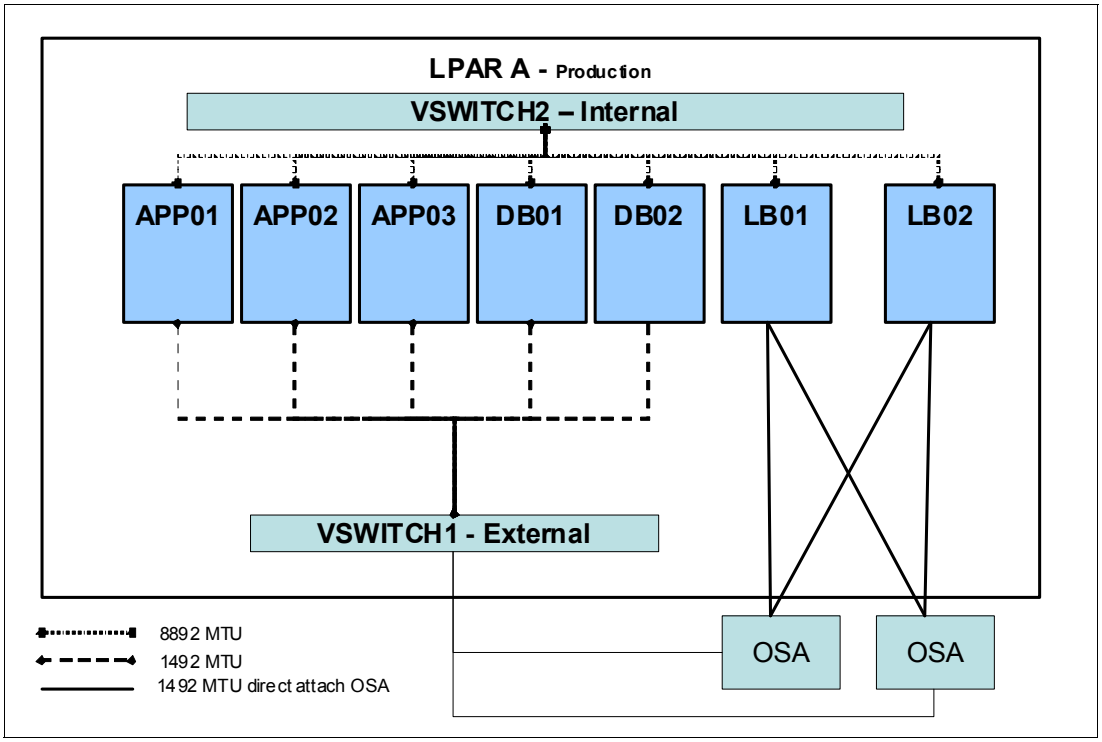


Figure 6-1 Network MTU config schema

Tip: For more information about high availability, see *Achieving High Availability on Linux for System z with Linux-HA Release 2*, SG24-7711.

6.2.2 Buffer count configuration

The buffer count parameter is a network parameter for QDIO devices for Linux on System z. This parameter allows Linux servers to receive more network packets in order to increase throughput.

The default buffer count value for Linux on System z is `16`. This parameter must be defined for each network device and allocates 1 MB of memory. A buffer count of 128 leads to 8 MB of memory consumption.

You can check the actual buffer count by using the **`lsqeth -p`** command.

The configuration of the buffer count must be done with the virtual interface in an offline state. Therefore, the best time to set it is when the interface starts. For SUSE Enterprise Linux 11 servers, you must edit the hardware channel `udev` file and include the line, **`buffer_count=128`**, as shown in Example 6-4 on page 80. Save the file and exit it when editing is complete.

Example 6-4 SLES 11 buffer_count setup

```
# vi /etc/udev/rules.d/51-qeth-0.0.0620.rules
ACTION=="add", SUBSYSTEM=="ccwgroup", KERNEL=="0.0.f200",
ATTR{buffer_count}="128"
```

For a Red Hat Enterprise Linux server, edit the network **ifcfg** file and add the line, **buffer_count=128**, as shown in Example 6-5. Again, save and exit when you have completed editing the file.

Example 6-5 RHEL buffer_count setup

```
#vi /etc/sysconfig/network-scripts/ifcfg-eth0
OPTIONS="buffer_count=128"
```

6.3 Compliance considerations

Most companies have controls or internal rules that provide standardization of procedures when it involves production environments. These controls are applied to all machines in order to ensure that all security measures, service levels agreements, and internal procedures are being followed when a new guest is built or moved to a production environment. In this section, we describe the various checklists that should be created to ensure that compliance considerations are met.

6.3.1 Production checklist

A simple and efficient way to control if the guest is compliant with the internal standards is to create a checklist that is used throughout the move to production process. Each item that is listed in the checklist must be reviewed by the system administrator and, if possible, evidence[1] should be created and attached to the checklist. After the checklist is completed, it should be sent for management approval, which guarantees that the guest is set up in production in compliance with all internal rules.

The following items should be included in the checklist:

- Privilege revalidation

  The system administrator must check and certify that only authorized IDs or groups have privileged access (root access for example). See Chapter 4 from *Security for Linux on System z*, SG24-7728 for further information about authentication and access control.

- Set root password to expire periodically

  Ensure that the default root password is changed and set to expire in a regular period of time; for example, 30, 60, 90 days.

- Configure and check if backup is working

  Create a backup policy and ensure that the backup service is running and working as expected before putting the Linux guest into production.

- Properly register or apply the guest to management tools (monitoring, vulnerability scanning, inventory, and so on).

  Monitoring tools

[1] As evidence, the system administrator can attach a log file or a screen print.

There are many solutions available, such as Nagios, Zabbix, Zenoss, IBM Tivoli® Monitoring, and Omegamon that are used for monitoring. Ensure that your new Linux on System z production guests are registered with the tool used by your enterprise.

– Vulnerability scanning

A vulnerability scan provides a report with information regarding open ports left by services or applications or even OS patch levels, and accordingly, it provides the system administrator with the actions that should be taken in order to mitigate the risks of a possible exposure. Tools such as IBM Security Health Scan, Nmap, and Nessus, are easily accessible on the web.

Another tool that can be used before moving the guest to a production environment that assists in searching for possible security breaches is a rootkit scanner (such as, Rkhunter, Chkrootkit), also available on the market either for a fee or open source. Rootkit is basically software that runs stealthily trying to obtain privileged access, mainly root access.

– Inventory tools

Inventory tools are applications that help the system administrator identify all assets from a hardware and software perspective. Every server should have all relevant information such as host name, IP address, amount of memory, CPU and disk, OS level, patch level, patching schedule, and environment status (production, development, test) registered to the inventory tool.

- Check if the server is updated with the latest patches and service packs applied

Patches and service packs are fundamental to environment safety, but often, you see servers being moved to production, or already in production with many levels of patches or service packs overdue.

Maintain a regular patching schedule for servers on production, and apply all patches on development or test servers to avoid major complications and possible outages on the production environment.

You can find sample checklists in Appendix B, “Migration checklists” on page 155.

6.3.2 Local repository

In order to keep track of all compliance and security measures for a healthy production environment, a good practice is the use of a local repository for software and system update management. The idea is to use only one guest with Internet access to keep local copies of the SUSE or Red Hat repositories and have the repository file systems that are shared with another host, which is used by all guests in the same logical partition.

The following process works either for a SUSE or Red Hat installation:

1. Configure a guest as the local repository. Our system used `lnxpth1`

Because this server is used only to synchronize local the file system with the Linux distribution repositories, you just need to set the following:

a. Local file system

Create a separate file system to store all patches, preferably by using LVM because it can be expanded as needed.

b. Copy files from disk images

Mount the image disks of your preferred Linux distribution in a temporary folder and copy all packages from disk to your local file system.

Example 6-6 shows an example of mounting ISO files and copying to the local file system.

Example 6-6 Mounting ISO files and copying to the local file system

```
# mount -o loop /mnt/repo/RHEL5.8-Server-20120202.0-s390x-DVD.iso
/mnt/rhel5.8
# cp -a /mnt/rhel5.8/. /exports/yumrepo
```

c. Synchronize the repositories

Synchronize your repository with the SUSE, or Red Hat official repository.

For SUSE, you need to have the Subscription Management Tool (SMT). This tool is an add-on that is available to all clients with an active SUSE Linux Enterprise Server subscription as well as a separate product called *SUSE Manager*.

Red Hat has the Red Hat Network Satellite, which is a product that can be purchased along with the Red Hat Enterprise Linux server subscription.

More information about these products can be found at the following sites:

- https://www.suse.com/solutions/tools/smt.html
- http://www.redhat.com/products/enterprise-linux/rhn-satellite

d. Activate the NFS server

Activate the NFS server deamon (`nfsserver`). Set it to start automatically on boot and share the file system with **`lnxpth2`**.

Example 6-7 How to start and activate nfsserver upon startup

```
# rcnfsserver start && chkconfig nfsserver on
# echo “/exports/yumrepo 9.12.4.227(ro,sync)” >> /etc/exports
# exportfs -av
exporting 9.12.4.227:/exports/yumrepo
```

2. Configure **`lnxpth2`**

This server is the official local repository, which means that the following services need to be set:

a. NFS client

Update **`fstab`** and mount the exported file system according to the parameters that are set on the NFS server. See Example 6-8.

Example 6-8 Mounting NFS file system

```
# mkdir -p /srv/www/htdocs/yum/repo
# echo "itsolnx3:/exports/yumrepo /srv/www/htdocs/yum/repo nfs ro,intr 0 0"
>> /etc/fstab
# mount /srv/www/htdocs/yum/repo
itsolnx3:~ # df -h /srv/www/htdocs/yum/repo
Filesystem            Size  Used Avail Use% Mounted on
itsolnx3:/exports/yumrepo
                      5.5G  3.2G  2.0G  63% /srv/www/htdocs/yum/repo
```

b. HTTP server

Install the web server and set it to start automatically on boot. See Example 6-9 on page 83.

Example 6-9 Apache web server settings

```
# yum install apache2
# rcapache2 start && chkconfig apache2 on
```

Note: For this IBM Redbooks publication, we chose Apache in our example but you can use any web server.

c. Configure repositories

To configure the repositories, you need to install the tool, *createrepo*. This tool can usually be found among software development kit (SDK) packages. This tool reads your file system file structure and creates an XML-based rpm metadata to be used by your software management tool, as shown in Example 6-10.

Example 6-10 Repository configuration

```
# createrepo /srv/www/htdocs/yum/repo/
2883/2883 - policycoreutils-gui-1.33.12-14.8.el5.s390x.rpm
Saving Primary metadata
Saving file lists metadata
Saving other metadata
```

3. Configuring guests

After you complete all settings for the repository server, the last step is to configure the guests to use the local repository instead of the online repository.

a. SLES11 guests:

To set up the local repository on SUSE guests, use YaST, or simply run the command that is shown in Example 6-11.

Example 6-11 Setup local repository for SUSE guests

```
# zypper ar http://9.12.4.227/yast/repo/suse/ ITSO_Yast_Repo
Adding repository 'ITSO_Yast_Repo' [done]
Repository 'ITSO_Yast_Repo' successfully added
Enabled: Yes
Autorefresh: No
GPG check: Yes
URI: http://9.12.4.227/yast/repo/suse/
```

b. RHEL guests:

To set up the local repository for RHEL guests, update the `/etc/yum.conf` file with the base-local settings, as shown in Example 6-12.

Example 6-12 Set up local repository for RHEL guests

```
# echo "[base-local]
> name=ITSO Yum Local
> failovermethod=priority
> baseurl=http://9.12.4.227/yum/repo/
> enabled=1
> gpgcheck=0 " >> /etc/yum.conf
```

6.3.3 Authentication

As an additional security measure, you can deny root access on **OpenSSH** and instead use encrypted personal keys to access the servers. For more information about how to implement the RSA key pair method, see section 8.7 of the book, *Security for Linux on System z*, SG24-7728.

7

Software planning considerations

In this chapter, we describe some software planning considerations when moving your z/VM and Linux on System z guests into a production environment. We provide information about management tools, database management systems, and Java application servers.

© Copyright IBM Corp. 2013. All rights reserved.

7.1 Management tools

In this section, we describe z/VM management tools that are commonly used. They are very useful and efficient tools that can help to improve the availability of a production environment:

- z/VM Directory Maintenance Facility (IBM DirMaint™)
- Provisioning Management - xCAT

Note: z/VM Directory Maintenance Facility is a z/VM preinstalled licensed feature.

7.1.1 z/VM Directory Maintenance Facility

When clients manage the z/VM user directory in a traditional way, they manage the z/VM user profiles via updating the `USER DIRECT` file and making the changes go into effect via issuing the **`DIRECTXA`** utility command. All user directory information is kept in one or multiple files. It is not easy to manage various individual files manually. Administrators must maintain the consistency and backup plan of their `USER DIRECT` files. So the managing of the z/VM user directory can be a manual procedure and can increase the possibility of human error.

z/VM does not control data consistency on a direct access storage device (DASD) volume. When using USER DIRECT to manage minidisks, administrators must check the available disk space manually via the **`DISKMAP`** command before assigning free minidisk spaces to a guest user directory. That means the existing data could be overwritten or destroyed by defining a minidisk with a range that overlaps, even if by one cylinder.

In an SSI environment, managing user directory files in a traditional way becomes more difficult because user directory files are shared by z/VM members in an SSI cluster. It takes more effort to maintain user directory consistency and protect user directory files from intentional or unintentional changes.

Using the IBM z/VM Directory Maintenance Facility can help alleviate this task. The IBM Directory Maintenance facility (DirMaint) is a CMS application that helps manage an installation's z/VM directory. Directory management is simplified by the **`DirMaint`** command interface and automated facilities. DirMaint directory statement-like commands are used to initiate directory transactions. DirMaint error checking ensures that only valid changes are made to the directory, and that only authorized personnel are able to make the requested changes. Any transaction requiring the allocation or deallocation of minidisk extents can be handled automatically. All user-initiated transactions can be password controlled and can be recorded for auditing purposes. DirMaint service can be ordered in the same way as z/VM service.

To use DirMaint in z/VM v6.2, administrators must enable the DirMaint service in the z/VM MAINT620.

Perform these steps to enable DirMaint:

1. Log on to MAINT620.
2. Check that MAINT620 has write access to MAINT's minidisk 51D by using the **`query accessed`** command. If 51D is not accessed as R/W, type the following command and press Enter:

   ```
   link maint 51d 51d mr
   ```

3. Access the minidisk:

   ```
   access 51d d
   ```

4. Enable DirMaint by using the following command:

```
service dirm enable
```

For more detailed configuration instructions, refer to Chapter 4 *Configuring the Directory Maintenance Facility,* in *z/VM: Getting Started with Linux on System z,* SC24-6194-02.

To configure z/VM DirMaint, you need an installation and maintenance virtual machine. In our z/VM 6.2 environment, the machine name is `6VMDIR20`. This virtual machine owns:

- All DASD space containing IBM supplied DirMaint product code
- Customer tailored files
- Customized exit routines

DirMaint consists of three service machines running as CMS guests to provide user directory management:

DIRMAINT The primary server, the DIRMAINT server, handles all aspects of source directory manipulation and controls the actions of all other servers. There is only one DIRMAINT server per SSI cluster. Only one DIRMAINT server can manipulate a single source.

DATAMOVE A DATAMOVE server is responsible for manipulating minidisks on behalf of the DIRMAINT server. These tasks can include formatting, copying, and cleaning. There can be multiple DATAMOVE servers being used by the one DIRMAINT server. In an SSI cluster, there must be one DATAMOVE server for each member of the SSI where available DASD to that member is formatted, copied, or cleaned.

DIRMSAT The DIRMSAT server is responsible for manipulating the object directory on systems other than the system the DIRMAINT server is on, or on the same system if maintaining duplicate copies of the object directory. There can be multiple DIRMSAT servers all being used by the one DIRMAINT server. If using DirMaint to synchronize the object directory in an SSI cluster, there must be one DIRMSAT server running on each system that is not running the DIRMAINT server.

It is not necessary to define all of these virtual machines, only the 6VMDIR20 and the DIRMAINT user IDs are required, but if you do not have these machines, you will not have the functions that they provide.

The DIRMAINT server handles all aspects of source directory manipulation and controls the actions of all other servers. DIRMAINT is the primary server and there is only one DIRMAINT server. DirMaint does not automate the process of starting or stopping servers when the DIRMAINT server stops or restarts.

When the z/VM user directory is managed by DirMaint, the user directory files and changes are maintained in special minidisks by DirMaint automatically. There is no way for general users to change the directory configuration directly by using the traditional manual management method. Only authorized user IDs can get a copy of one or multiple user profiles from the object directory disk via DirMaint commands and receive them into local minidisks for further processing, such as update and replace user definitions.

Also, DirMaint can back up the user directory definitions automatically if clients configure the backup resources and schedule this in DirMaint. By default, a backup is taken once per day, after midnight.

How DirMaint works

Figure 7-1 shows the DirMaint configuration in a non-SSI environment.

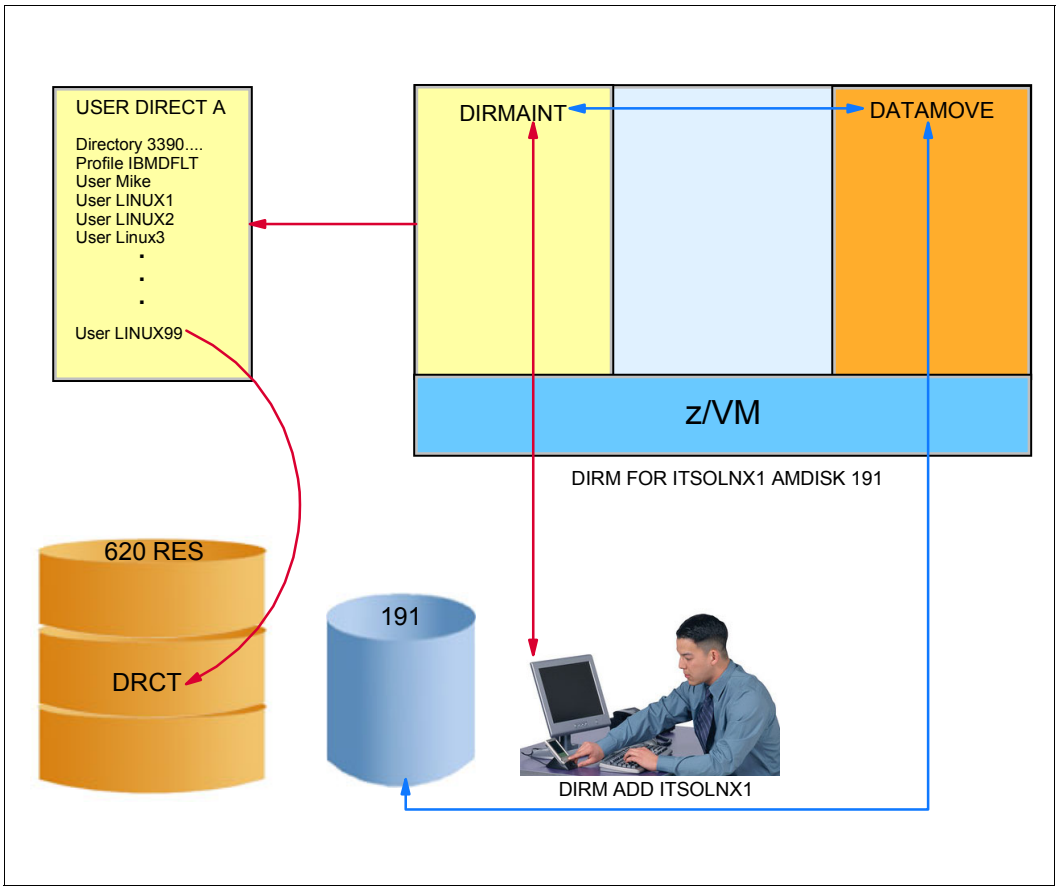


Figure 7-1 Non-SSI DirMaint environment

In this environment, when a client adds a new user, `ITSOLNX1`, DIRMAINT server handles all aspects of source directory file manipulation, including checking the directory file syntax. In the meanwhile, if the DATAMOVE server is available, DIRMAINT requests DATAMOVE to format the minidisks that are defined in the source user directory profile. When everything has been processed, the source user directory file is compiled onto a DRCT minidisk (where cylinders or pages are allocated as space for directory files) on a CP-owned volume by the DIRMAINT server. Whenever a user direct profile needs to be changed, the current active user directory profile can be obtained from the z/VM system and received via the commands that are shown in Example 7-1. This means that you can always be sure that you are updating the latest version of the user directory profile by getting the existing active ones every time. With this approach, you can maintain the consistency of the user directory profiles. When the DIRMAINT server is down, no DirMaint commands can be processed.

Example 7-1 Get user directory profile and receive it to local disk

```
DIRM FOR ITSOLNX1 GET
DVHXMT1191I Your GET request has been sent for processing to DIRMAINT at
DVHXMT1191I ITSOSSI2 via DIRMSAT.
Ready; T=0.01/0.01 09:33:29
 DVHREQ2288I Your GET request for ITSOLNX1 at *
 DVHREQ2288I has been accepted.
 DVHGET3304I Directory entry  ITSOLNX1 is now
 DVHGET3304I locked.
```

```
 DVHREQ2289I Your GET request for ITSOLNX1 at *
 DVHREQ2289I has completed; with RC = 0.
RDR FILE 1485 SENT FROM DIRMAINT PUN WAS 6289 RECS 0032 CPY  001 A NOHOLD NOKEEP
RECEIVE 1485 (REP
File ITSOLNX1 DIRECT A0 replaced by ITSOLNX1 DIRECT Z0 received from DIRMSAT at
ITSOSSI1
Ready; T=0.01/0.01 09:36:26
```

In an SSI cluster environment, it is suggested to have at least one DATAMOVE server per SSI member. Also, it is suggested that there be at least one DIRMSAT server on each SSI member that is not running the DIRMAINT server. Figure 7-2 shows the DirMaint configuration in an SSI environment.

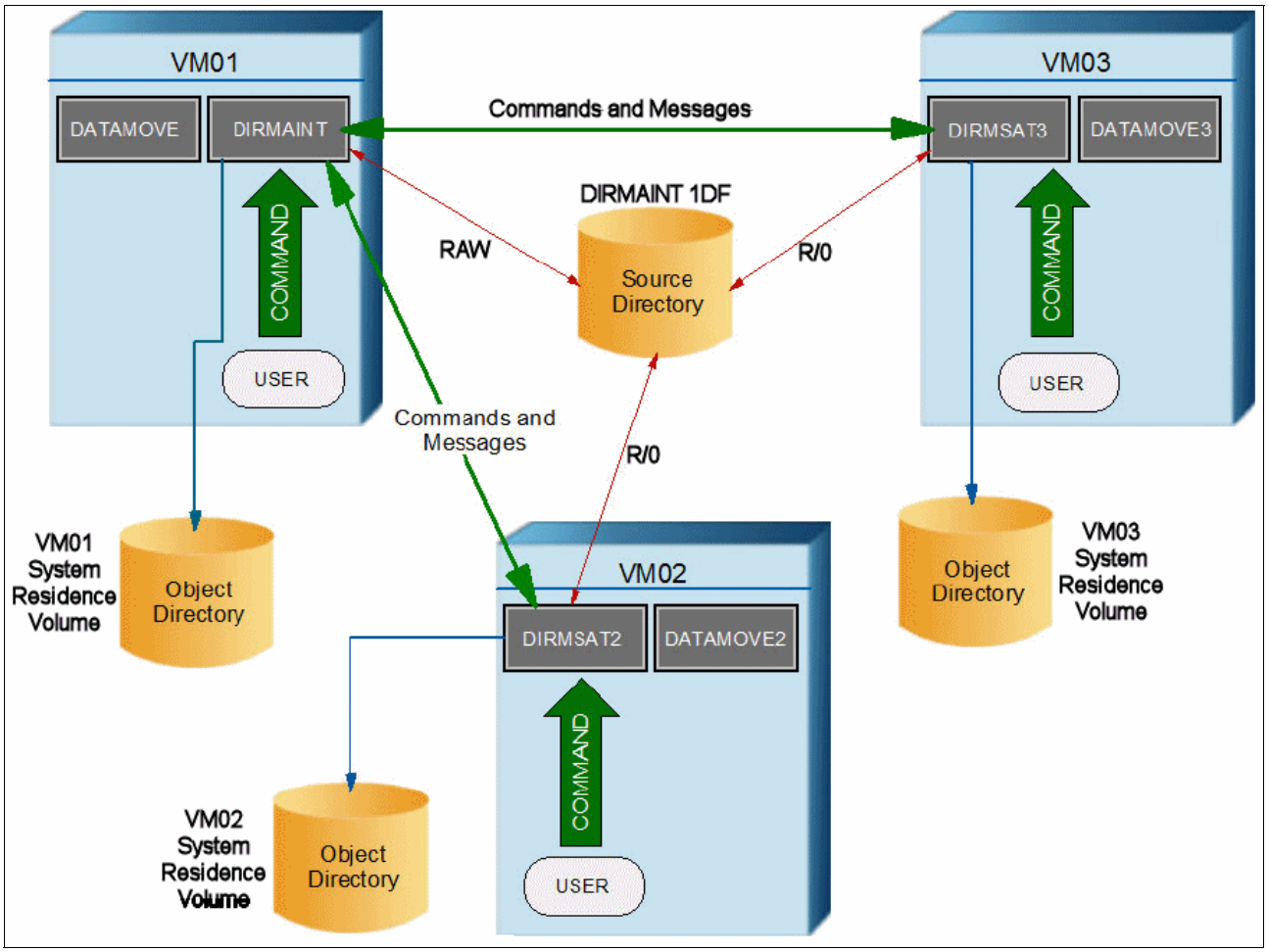


Figure 7-2 SSI DirMaint environment

In this environment, the source user directory file is shared by all members of the cluster, but each member has its own object directory. There is one DIRMAINT server running on an SSI member that controls the source directory files for the cluster. The DIRMAINT server can run on any system in the cluster and is a single-configuration virtual machine. The DIRMAINT server has R/W control of the source directory regardless of the member it is running on. Satellite DirMaint servers, running on the other members of the cluster, provide an interface to users on their local system to the DIRMAINT server. DIRMAINT and satellite servers control

the object directory for their local member systems. In the SSI environment, DIRMAINT and the satellite servers communicate through the shared spool.

When the DIRMAINT server in an SSI environment is down, it is the same as in a non-SSI environment, no DirMaint commands can be processed in the whole SSI environment. If a DIRMSAT server running on a cluster member goes down while the member is still joined, users on that system cannot issue DirMaint commands and changes made to the directory by DIRMAINT will not be reflected in the object directory of DIRMSAT. When the DIRMSAT server is restarted, it processes updates that are made while it was out of service.

In an SSI environment, DirMaint is updated to handle new IDENTITY and SUBCONFIG entries in a multi-configuration user directory. For more information about new user directory entries, refer to *z/VM V6R2 CP Planning and Administration,* SC24-6178-03*,* which can be found at the following site:

http://publib.boulder.ibm.com/infocenter/zvm/v6r2/index.jsp?topic=%2Fcom.ibm.zvm.v620.hcpa5%2FhcsgOc1110.htm

DASD management

DATAMOVE servers can handle disk pool management automatically. Administrators can define different disk pools in `EXTENT CONTROL` files, as shown in Example 7-2. In this file, we can define regions, groups, SSI_Volumes, and so on.

Example 7-2 EXTENT CONTROL file

```
:REGIONS.
*RegionId  VolSer    RegStart      RegEnd  Dev-Type  Comments
SSIUR1     SSIUR1    002           end     3390-27
LX9B25     LX9B25    001           10016   3390-09
LX9B26     LX9B26    001           10016   3390-09
SSIAC2     SSIAC2    001           10016   3390-09
SSIAR2     SSIAR2    001           10016   3390-09
SSIBC2     SSIBC2    001           10016   3390-09
SSIBR2     SSIBR2    001           10016   3390-09
:END.
:GROUPS.
VM620A     SSIAC2 SSIAR2
VM620B     SSIBC2 SSIBR2
:END.
00079 :SSI_VOLUMES.
00080 * Added during Installation, Do not remove.
00081 *VolumeFamily     Member   VolSer
00082 IBM_RES           ITSOSSI1 SSI1I2
00083 IBM_RES           ITSOSSI2 SSI2I2
00084 IBM_RES           ITSOSSI3 SSI3I2
00085 IBM_RES           ITSOSSI4 SSI4I2
00086 :END.
:EXCLUDE.
* USERID ADDRESS
MAINT* 012*
MAINT* 013*
PMAINT 013*
SYSDUMP2 012*
SYSDMP* 012*
DATAMOV* 05F*
:END.
```

```
:AUTOBLOCK.
:DEFAULTS.
  3390-09     10017
  3390-27     32760
:END.
```

When adding or removing disks, you do not need to specify volumes in the traditional way. As shown in Example 7-3, DirMaint selects the proper extension for minidisks within the disk pool automatically. In this example, we specify the device type as “XXXX” because automatic allocation is used. “AUTOG 10” refers to automatically assigning 10 cylinders at a suitable place on volume group VM620B. This volume can be either extended count key data (ECDK) or fixed-block architecture (FBA) devices.

Example 7-3 Add minidisk to ITSOLNX1

```
DIRM FOR ITSOLNX1 AMDISK 499 XXXX AUTOG 10 VM620B M
```

Administrators do not need to worry about overlap on disk volumes. The only concern is if the disk group size can meet business requirements. The administrator can add or remove DASD to or from storage groups as necessary.

Disk pool management can improve the disk utilization and work efficiency, especially for installations that have many minidisks to be managed. Disk management in DATAMOVE is also required by external guest management solutions, for example, zCloud or xCAT.

Configuration considerations

Following are some considerations for DirMaint installation or configuration in an SSI environment:

- An SSI installation creates the service machines, the CONFIGxx DATADVH, and EXTENT CONTROL configuration file statements necessary to run DirMaint in the cluster.
- The **SERVICE DIRM ENABLE** command (to enable the product in z/VM) only needs to be run on only one member.
- **PUT2PROD** needs to be run on every member.
- Configuration files are shared. They can be created from any member of the cluster.
- Change DIRMAINT’s default password from AUTOONLY to some other password before installation. You can change it back after you successfully test DirMaint.

Summary

In this section, we summarize the benefits of using DirMaint, as well as the DirMaint internal minidisk matrix.

The benefits from DirMaint are as follows:

- DirMaint operates as a CMS application in a z/VM operating system for support of the system directory. It is preinstalled in the z/VM system. There is no additional installation effort required.
- DirMaint minimizes the possibility of human error through an automated process of managing the directory.
- DirMaint ensures the integrity of the directory.
- DirMaint ensures the integrity of MDisk by preventing new minidisk space from being inadvertently allocated over existing extents.

- When using DirMaint, it is easier to integrate with other management tools smoothly, for example, the IBM Tivoli Provisioning Manager.
- A menu/panel is displayed for the complex DirMaint commands.
- DirMaint service processes are simplified by using z/VM.
- Online HELP is available for all DirMaint commands and messages.
- The DirMaint service machines run disconnected and unnoticeable.

There are many configuration files in DirMaint to control the behaviors of DirMaint. The files reside on different minidisks in DirMaint. These files are listed in Table 7-1.

Table 7-1 DirMaint minidisk matrix

Minidisk	Owner	Function	Configuration files
491	6VMDIR20	Production server code	DVHNAMES DATADVH DIRMAINT DATADVH DIRMSAT DATADVH DATAMOVE DATADVH DVHPROFM DATADVH DVHPROFA DIRMAINT DVHPROFA DIRMSAT PROFILE EXEC
492	6VMDIR20	Test server code	DVHNAMES DATADVH DIRMAINT DATADVH DIRMSAT DATADVH DATAMOVE DATADVH DVHPROFM DATADVH DVHPROFA DIRMAINT DVHPROFA DIRMSAT PROFILE EXEC
11F	6VMDIR20	Production DirMaint interface code	CONFIGxx DATADVH DIRMMAIL DATADVH 140CMDS DATADVH 150CMDS DATADVH
41F	6VMDIR20	Alternate DirMaint interface code	CONFIGxx DATADVH DIRMMAIL DATADVH 140CMDS DATADVH 150CMDS DATADVH
1DF	DIRMAINT	Primary directory files	EXTENT CONTROL DVHLINK EXCLUDE* PWMON CONTROL* RPWLIST DATA* AUTHFOR CONTROL USER INPUT AUTHDASD DATADVH
1DB	DIRMAINT	Primary location of USER BACKUP file (optional)	The same as 1DF
551	PMAINT	Common DIRECTXA, DIRMAP, and DISKMAP utilities	N/A

Note: DIRMAINT can have multiple CONFIGxx DATADVH files. We suggest clients to write down the existing CONFIGxx DATADVH file names for consistency purposes.

If the administrator wants to change the configuration files on the 491 or 11F minidisks, they must be modified on minidisk 492 or 41F. Then, use the **DIRM FILE** command to replace the corresponding files on the 491 or 11F minidisks and issue the **DIRM RLDDATA** command to place the change into production. Another way to do this without requesting the DIRMAINT server detach the 492 and 41F disks, is to modify the configuration by using the following steps:

1. Use the **DIRM SEND** command to send the current copy of the file from the DIRMAINT server to your reader and receive it to your local minidisk for editing. Example 7-4 shows how to get and receive a configuration file.

   Example 7-4 Get the file from DIRMAINT and receive from reader

```
dirm send DIRMAINT DATADVH
DVHXMT1191I Your SEND request has been sent for processing to DIRMAINT
DVHXMT1191I at ITSOSSI1 via DIRMSAT2.
Ready; T=0.01/0.01 11:33:48
 DVHREQ2288I Your SEND request for MAINT620 at * has
 DVHREQ2288I been accepted.
 DVHREQ2289I Your SEND request for MAINT620 at * has
 DVHREQ2289I completed; with RC = 0.
RDR FILE 0302 SENT FROM DIRMAINT PUN WAS 4826 RECS 0010 CPY  001 A NOHOLD
NOKEEP
receive 302
File DIRMAINT DATADVH A1 created from DIRMAINT DATADVH Z1 received from
DIRMSAT2
 at ITSOSSI2
Ready; T=0.01/0.01 11:36:01
```

2. After editing the file, use the **DIRM FILE** command to send the updated file back to the DIRMAINT service machine. Example 7-5 shows how to replace the existing DIRMAINT configuration file.

   Example 7-5 Send back and update the existing DIRMAINT configuration file

```
dirm file DIRMAINT DATADVH
PUN FILE 0306 SENT TO   DIRMSAT2 RDR AS  4830 RECS 0014 CPY  001 0 NOHOLD
NOKEEP
DVHXMT1191I Your FILE request has been sent for processing to DIRMAINT
DVHXMT1191I at ITSOSSI1 via DIRMSAT2.
Ready; T=0.01/0.01 11:39:01
 DVHREQ2288I Your FILE request for MAINT620 at * has
 DVHREQ2288I been accepted.
 DVHRCV3821I File DIRMAINT DATADVH A1 has been
 DVHRCV3821I received; RC = 0.
 DVHRCV3821I File DIRMAINT DATADVH C2 has been
 DVHRCV3821I received; RC = 0.
 DVHRCV3821I File DIRMAINT DATADVH E1 has been
 DVHRCV3821I received; RC = 0.
 DVHREQ2289I Your FILE request for MAINT620 at * has
 DVHREQ2289I completed; with RC = 0.
```

Replace the previous copy on the appropriate production disk. Then, issue the **DIRM RLDDATA** command to place the changed files into production. If the `EXTENT CONTROL` file is changed, issue the **DIRM RLDEXTN** command to put the change into effect.

For more information about z/VM Directory Maintenance Facility configuration and commands, refer to the following IBM documentation:

- *z/VM V6R2 Directory Maintenance Facility Tailoring and Administration Guide*, SC24-6190-02
- *z/VM V6R2 Directory Maintenance Facility Commands Reference*, SC24-6188-02
- *z/VM V6R2 Getting Started with Linux on System z*, SC24-6194-01

7.2 Database management systems

Database applications servers such as IBM DB2, Oracle, MySQL, and PostgreSQL, all can run on Linux on System z.

Database servers write data to disk devices and read information from disk devices. Database administrators set up the memory in database configuration files to cache the data in order to reduce the number of disk reads. By reducing the amount of time that is used to read a disk, you can maximize the number of writes. However, there is a limit to scalability when there are numerous users and a grid system such as Oracle RAC or DB2 PureScale are used because the system must read, write, and also synchronize all data between the nodes. In certain cases, that can cause high escalation of CPU utilization (sometimes it can rise to more than 40% of CPU time). But that kind of configuration is necessary in distributed systems to allow the scalability of the system, and also to cover the lack of hardware availability.

In System z, there are specialty engines that are called *SAP* (see Chapter 3, “Hardware planning considerations” on page 29) and a subsystem channel that handles all I/O requests. There is also more processor time when using Integrated Facility for Linux (IFL) to process the database application requests. This advantage plus hardware high availability permits production database servers to use the active-passive cluster solution instead of the active-active cluster solution. It reduces the utilization of the processor resources.

Taking advantage of the I/O subsystem of System z and using either a HyperPAV solution (when using ECKD devices) or multipath solution (when using SCSI disks) reduces the IFL utilization and should be considered for a database production server.

Before moving a database server to production, it is important to correctly size the cache memory that will be used by the database. There is no single set of values for configuring memory that is valid in all environments and for all database management systems. A good book to refer to for setting up your IBM DB2 database management system is *DB2 10 for Linux on System z Using z/VM v6.2, Single System Image Clusters and Live Guest Relocation*, SG24-8036.

Use the z/VM virtual disks as a swap disk for Linux guests. The general swap recommendation for Linux on System z is to keep swapping to a minimum where possible. To keep swapping as fast as possible, this is why you might want to use the z/VM virtual disks for the first small swapping device.

For more information about virtual disk and swap management and allocation, see section 3.1.1, “Swap” on page 33.

7.2.1 Linux memory setup considerations for database servers

In this section, we describe some basic configurations in Linux that are related to kernel options parameters that you should set. Every database management system has its own parameter values, but here we describe those that are specific to DB2 databases.

The default kernel parameter values might cause issues such as memory starvation when running a DB2 database management system. To avoid such issues, semaphores and shared memory values should be updated (kernel.sem, kernel.shmall, kernel.shmmax, and kernel.shmmni). If DB2 is installed as root, these parameters might be configured automatically. For more information about shared memory allocation, see *Practical Migration to Linux on System z*, SG24-7727.

Interprocess communication kernel parameters

The Linux interprocess communication (IPC) kernel parameters allow for multiple processes to communicate with one another.

To improve the performance and memory usage of DB2, you need to adjust several of these kernel parameters. By adjusting these kernel parameters, the database manager prevents unnecessary resource errors. Table 7-2 shows the recommended kernel parameters that should be modified.

Table 7-2 Recommended kernel parameters

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

Modifying the kernel parameters

You must have root privileges to modify kernel parameters.

To update kernel parameters on Red Hat Enterprise Linux and SUSE Linux[1], perform the following steps:

1. Run the **ipcs -l** command to list the current kernel parameter settings.
2. Analyze the command output to determine if you must change kernel settings by comparing the current values with the *enforced minimum settings* that are shown in Example 7-6. This example shows the output of the **ipcs** command with comments that we added after the // to indicate the parameter names.

Example 7-6 ipcs -l command output

```
ipcs -l

   ------ Shared Memory Limits --------
   max number of segments = 4096               // SHMMNI
   max seg size (kbytes) = 32768               // SHMMAX
   max total shared memory (kbytes) = 8388608  // SHMALL
   min seg size (bytes) = 1

   ------ Semaphore Limits --------
   max number of arrays = 1024                 // SEMMNI
   max semaphores per array = 250              // SEMMSL
   max semaphores system wide = 256000         // SEMMNS
   max ops per semop call = 32                 // SEMOPM
   semaphore max value = 32767

   ------ Messages: Limits --------
   max queues system wide = 1024               // MSGMNI
   max size of message (bytes) = 65536         // MSGMAX
   default max size of queue (bytes) = 65536   // MSGMNB
```

3. Modify the necessary kernel parameters by editing the /etc/sysctl.conf file. If this file does not exist, create it. Example 7-7 shows an example of what should be placed into the file.

Note: The /etc/sysctl.conf file is provided via a package (RPM). On SUSE Linux Enterprise Server 11, you find it in the PROCPS RPM file; and in Red Hat Enterprise Linux, the INITSCRIPTS RPM file.

Example 7-7 Kernel parameters within /etc/sysctl.conf

```
#Example for a computer with 16GB of RAM:
kernel.shmmni=4096
kernel.shmmax=17179869184
kernel.shmall=8388608
#kernel.sem=<SEMMSL> <SEMMNS> <SEMOPM> <SEMMNI>
kernel.sem=250 256000 32 4096
kernel.msgmni=16384
kernel.msgmax=65536
kernel.msgmnb=65536
```

[1] See website for up-to-date information, under “Modifying kernel parameters (Linux)”: http://publib.boulder.ibm.com/infocenter/db2luw/v10r1

4. Run the **sysctl -p** command to load the sysctl settings from the /etc/sysctl.conf default file.

Example 7-8 sysctl execution for reloading settings from /etc/sysctl.conf

```
sysctl -p
```

Optional: To make the changes persist after every reboot:

- (SUSE Linux) Make boot.sysctl active. (Execute the **chkconfig sysctl on** command, as root).
- (Red Hat Enterprise Linux) The **rc.sysinit** initialization script reads the /etc/sysctl.conf file automatically.

For the latest information about supported Linux distributions, see the following website:

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

Additional kernel parameter settings

Additional kernel parameter settings are listed in Table 7-3.

Table 7-3 Configuring other Linux kernel parameters

Kernel parameter setting	Configuring the kernel parameters for DB2 data server
vm.swappiness=0	This parameter defines how prone the kernel is to swapping application memory out of physical random access memory (RAM). The default setting, vm.swappiness=0, configures the kernel to give preference to keeping application memory in RAM instead of assigning more memory for file caching. This setting avoids unnecessary paging and excessive use of swap space. This setting is especially important for data servers configured to use the self-tuning memory manager (STMM).
vm.overcommit_memory=0	This parameter influences how much virtual memory the kernel permits allocating. The default setting, vm.overcommit_memory=0, sets the kernel to disallow individual processes from making excessively large allocations, however the total allocated virtual memory is unlimited. Having unlimited virtual memory is important for DB2 data servers, which retain additional unused virtual memory allocations for dynamic memory management. Unreferenced allocated memory is not backed by RAM or paging space on Linux systems. Avoid setting vm.overcommit_memory=2 because this setting limits the total amount of virtual memory that can be allocated, which can result in unexpected errors.

For more information about kernel parameter recommendations, see the following website:

http://www.ibm.com/developerworks/linux/linux390/perf

7.2.2 Linux storage setup considerations for database server

All database servers are designed to execute reads and writes to the disk. On a production Oracle database server, set up the database proprietary file system as the Oracle ASM. In the case of MySQL or PostgreSQL servers that use the Linux native file system, set it up as

an ext3 file system and set up the correct journaling mode. In most cases, the ordered or writeback option is enough. Also turn off the file system access time option (atime). There is no need to request access time of the database files. To set up journaling as writeback and turn off the atime option in an ext3 file system, define the /etc/fstab entry for the file system as shown in Example 7-9.

Example 7-9 fstab entry

```
/dev/vg00/lv01/dataext3defaults,journal=writeback,notime00
```

To maximize the utilization of the disk channel subsystem, use the LVM logical volume with striped devices that maximizes the utilization of the channel. However, do not set up more stripe segments than the number of channels that you have on your LPAR. If you have four distinct LUNs allocated to your database server, set up at most four striped segments to your logical volume. When using a SCSI disk, set up the multipathing tool with the failover option. But if you decide to use an active-active pool in the multipathing configuration, you might reduce the value of the **rr_min_io** parameter to 100 in the /etc/multipath.conf file, as shown in Example 7-10. This parameter controls the number of I/O requests issued to a path before it changes to the next path in the pool.

Example 7-10 Defaults of /etc/multipath.conf file

```
   defaults {
   user_friendly_names yes
   path_group_policy multibus
   rr_min_io 100
}
```

If possible, choose several smaller disks instead of one unique and large disk. If you need 400 GB for the database server, instead of using a single LUN of 400 GB, set up eight 50-GB LUNs in a unique volume group and combine the number of stripes at the logical volume group with the number of I/O channels available to access the disks. Figure 7-3 on page 99 shows an example of one option for an optimal setup for I/O access.

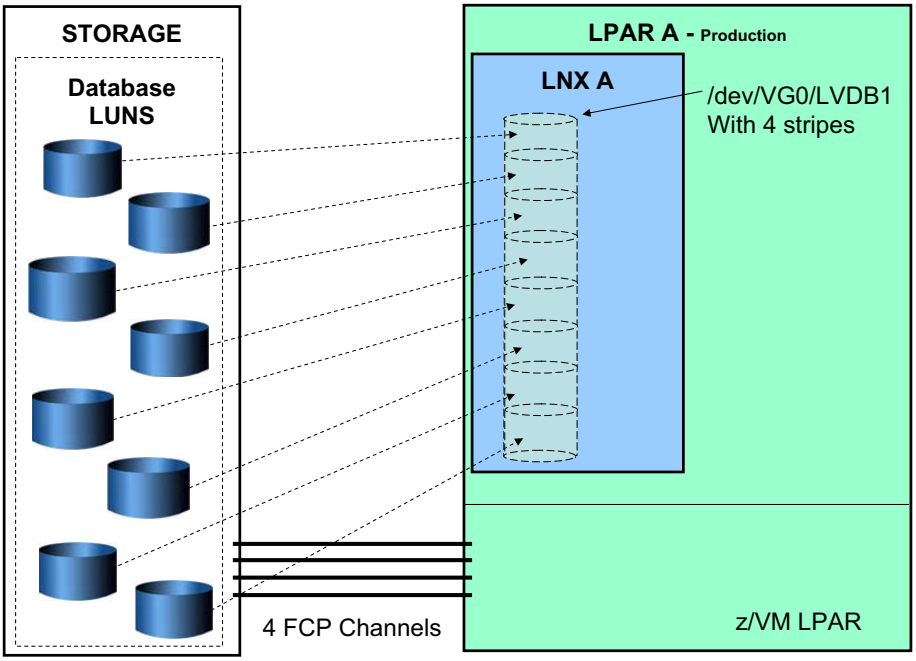


Figure 7-3 FCP SCSI setup for databases

The environment that is shown in Figure 7-3 describes a z/VM partition with four FCP channels configured. Each FCP channel address is virtualized by using the NPIV solution. Based on the NPIV solution, each Linux guest has a unique WWPN ID for each channel configured. In the Linux server, there are four virtual FCP channels set up. Based on that, the logical volume is configured with a four-striped definition and the multipathing Linux feature is defined as a multibus configuration.

It is also important to set up the log file system in a different area from the data file system, not only on a different logical volume, but also ensure that different logical volumes are not using the same LUN. To verify if those file systems are configured to use the same device in a different logical volume, use the command that is shown in Example 7-11.

Example 7-11 Logical volume devices

```
# lvs -o +devices
```

In most database management systems, there are some specific configurations that can be made for direct I/O. The direct I/O feature is an option that bypasses the Linux page cache system. Using this, the number of processor cycles are fewer because it avoids copy operations and it also saves memory when the page cache from Linux is not used.

Another feature that database application server configurations depend on is asynchronous I/O. In synchronous I/O, when a read request is issued by the software, it must finish before the next application request is processed. That normal I/O creates a kernel processor queue. In asynchronous I/O mode, after the kernel I/O request, the application continues processing other requests. When it receives an answer from the I/O request, the application processes it and closes the cycle. Use of asynchronous I/O increases the number of processor requests and demands more processor resources.

You can also take advantage of the new storage features on the market. For example, the new IBM DS8000 storage has a feature that allows striping of the storage pool volumes across multiple Redundant Array of Independent Disks (RAID) arrays. The use of that resource increase read and write performance and reduce storage administration time.

7.2.3 Linux network consideration for database servers

When you set up a Linux on System z database server, take advantage of the virtual network connections. The use of VSWITCHs and HiperSockets helps maximize the throughput between application servers and database servers. If possible, set up all system-related communications using application servers and database servers configured using internal VSWITCHs or HiperSockets devices, as shown in Figure 7-4.

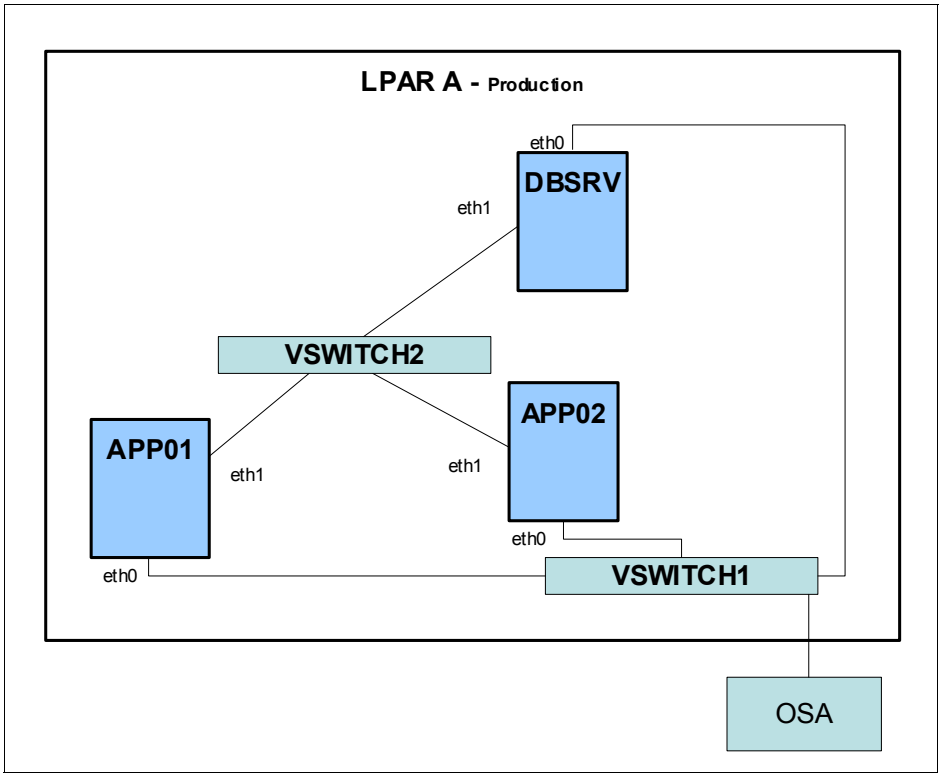


Figure 7-4 Database server network setup

In this configuration, it is possible to take advantage of the large maximum transmission unit (MTU) size configuration. The default configuration is based on an MTU size of 1492 bytes. If you increase the value to 8992 bytes, it increases the performance of transaction data and reduces utilization of the processor.

In the configuration that is shown in Figure 7-4, the **eth0** interfaces of the Linux guests are used for the application servers APP01 and APP02, which are used to receive external requests and system administration tasks. For database server DBSRV, the eth0 is for system administration tasks only. The **eth0** interface was set up using the standard MTU size of 1492 and was connected to VSWITCH1, which is attached to an OSA channel and allows outside access to the application. The **eth1** interface at each server is connected to VSWITCH2, which is an internal-use-only switch and all virtual interfaces are configured to use an MTU size with 8992 bytes.

The reason that we set up two different virtual switches was to take advantage of bigger MTU sizes. All hops on the path must be configured with the same size to avoid fragmentation of the data packages.

7.2.4 IBM DB2 Enterprise Database Server considerations

There are performance gains for the database server using Linux on System z. In this section, we provide a checklist of topics that should be reviewed before the server goes into production. There are also some specific DB2 Enterprise Server configurations that database administrators must set up to ensure maximum performance:

- Set up the disks for data and log file systems.
- Set up multiple FCP channels for disk access and configure it as multipathing.
- Set up kernel memory parameters calculating the total guest virtual memory. Use 90% of the total memory as a starting place.
- Set the kernel parameter for minimal free memory to 5% of the total virtual memory on the server.
- Set the DB2 Enterprise Server to use asynchronous I/O.
- Set up the DB2 Enterprise Server to use direct I/O.
- Set up the DB2 Enterprise Server to use the concurrent I/O feature. The configuration of this option is set during table space creation or also can be set up after by altering the table space information.
- Manually set the maximum DB2 Enterprise Server instance memory. If there is more than one instance on the server, do not allow the total size of memory per instance to be more than 90% of the total server memory size.
- Set up the internal connection between the application servers and DB2 servers with an 8 KB MTU size.
- Set up `/etc/security/limits.conf` to limit the number of open files that are used by the db2inst1 ID, as shown at Example 7-12.

Example 7-12 /etc/limits.conf file changes for db2 user ID

```
db2inst1     soft nofile 65536
db2inst1     hard nofile 65536
```

Example 7-13 shows the `/etc/sysctl.conf` file of a DB2 Enterprise Linux guest with 16 GB of memory.

Example 7-13 Sixteen-gigabytes server /etc/sysctl.conf file

```
# Memory Settings

kernel.shmmax=13285206016
kernel.shmall=3243458
kernel.sem=250 256000 32 1024
kernel.msgmni=1024
kernel.msgmax=65536
kernel.msgmnb=65536
vm.swappiness=0
vm.page-cluster=1
```

```
vm.min_free_kbytes=720768

#Network Settings

net.core.rmem_default=16777216
net.core.wmem_default=16777216
net.core.rmem_max=16777216
net.core.wmem_max=16777216
net.ipv4.tcp_rmem=4096 87380 16777216
net.ipv4.tcp_wmem=4096 65536 16777216
```

Example 7-14 shows a sample script that copies the database management system's recommended configuration into the `/etc/sysctl.conf` file.

Example 7-14 Bash script to db2 database memory configuration

```
#!/bin/bash
# Configuring kernel parameters on /etc/sysctl.conf

shmmax="$(echo "(($(cat /proc/meminfo | grep ^MemTotal | awk '{print
$2}')*90)/100)*1024" | bc)"
shmall="$(echo "$shmmax/4096" | bc)"
min_free_kbytes="$(echo "(($(cat /proc/meminfo | grep ^MemTotal | awk '{print
$2}')*5)/100)" | bc)"

cat <<EOF>>/etc/sysctl.conf
"# Memory Settings "
kernel.shmmax=${shmmax}
kernel.shmall=${shmall}
kernel.sem="250 256000 32 1024"
kernel.msgmni="1024"
kernel.msgmax="65536"
kernel.msgmnb="65536"
vm.swappiness=0
vm.page-cluster=1
vm.min_free_kbytes=${min_free_kbytes}
"# Network Settings "
net.core.rmem_default="16777216"
net.core.wmem_default="16777216"
net.core.rmem_max="16777216"
net.core.wmem_max="16777216"
net.ipv4.tcp_rmem="4096 87380 174760"
net.ipv4.tcp_wmem="4096 16384 131072"
EOF
```

7.2.5 Oracle Database Server considerations

For more information about sizing, set up, management, and migration of Oracle Database Server on Linux on System z, refer to the following IBM Redbooks publications:

- *Experiences with Oracle 11gR2 on Linux on System z*, SG24-8104
- *Using Oracle Solutions on Linux for System z*, SG24-7573

7.2.6 Summary

Database management and tuning for production environments is a complex subject and has close relations to the applications that use the databases. In this chapter, we described some general considerations for database management systems.

There are some other database performance tuning and tips available on the following IBM websites:

- Database tuning and tips of Linux on System z on IBM developerWorks:

  http://www.ibm.com/developerworks/linux/linux390/perf/tuning_database.html

- IBM white paper: *Performance considerations for databases on Linux on System z:*

  http://public.dhe.ibm.com/software/dw/linux390/perf/Performance_considerations_for_databases_on_Linux_on_System_z.pdf

7.3 Java application considerations

Java based applications are run on Linux servers as front-end applications or as the application servers in a service-oriented architecture where users do not have access to the core application. On Linux on System z, the IBM Java JDK is the primary version for Java workloads. There are several ways to get the IBM JDK for Linux on System z:

- It is included with many products, such as IBM WebSphere® Application Server
- It is included with the distributions delivered by SUSE and Red Hat
- It can be downloaded from IBM developerWorks:
  http://www.ibm.com/developerworks/java/jdk/linux/download.html

Java workloads can use much processor and memory; therefore, the use of the newest IBM Java SDK that is available is recommended. Improvements from one version to another can be more than 50%. Improvements in the Java virtual machine (JVM) and just-in-time (JIT) compilers are continuous as are improvements in garbage collection (GC) technologies.

Because of new and faster processors in z196 or zEC12 Enterprise machines, there are significant improvements in execution of applications on Java application servers compared to business class machines.

To promote a Linux on System z guest to production, the first step is to determine the application's memory requirements and Java heap size. The first rule is that Java heap size must be lower than the total virtual server memory. A correct Java heap avoids excessive paging in Linux and z/VM.

It is possible to identify the correct size of the Java heap during development and application testing. During a monitored process, if the Java heap size is too small, the system constantly reclaims the garbage collection function and a Linux kernel Out Of Memory (OOM) exception occurs. If the Java heap is too large, the system starts swapping. Use the monitor garbage collection parameter (**-verbose:gc**) to identify the most desirable size for your application. This parameter should be set during the test or quality assurance phases only. When you migrate the server to the production environment, this parameter must be removed from the Java interpreter.

The z/VM hypervisor and Linux OS guest are enough to manage memory, so you should set the heap size to a fixed value. Use the Java parameters for fine-tuning the application and set the initial heap size (**-Xms**) and the maximum heap size (**-Xmx**) to the same values.

For more information about Java heap size, see the following site:

http://publib.boulder.ibm.com/infocenter/javasdk/tools/topic/com.ibm.java.doc.igaa/_1vg00014557b090-11cd67f58fb-7fe1_1008.html?resultof=%22%68%65%61%70%22%20%22%73%69%7a%65%22%20

Some Java applications demand a large memory space to perform, and the use of the Linux **hugepages** parameter benefits those applications. Support for large pages can be included into the Linux kernel by choosing one of the following:

- Add hugepages=<npages> to the kernel parameter line, where “npages” is the number of 1 M pages that you want to allocate. So for example, hugepages=1024 allocates 1 GB worth of memory as large pages (the zEC12 supports 2 GB large pages).
- Dynamically: do a “echo <npages> > /proc/sys/vm/nr_hugepages” to receive npages worth of large pages. This only works if enough free memory is available that can be remapped as large pages. For 1 GB of large pages “echo 1024 > /proc/sys/vm/nr_hugepages” tries to allocate them.

To check if the allocation worked, do a “cat /proc/sys/vm/nr_hugepages”. Usage information can usually be found in `/proc/meminfo`.

You can enable large page support by starting Java with the **--Xlp** option.

The utilization of the processor is also a stress point of Java applications. Part of the processor time from a Java application is used to manage the resources that each JVM allocates. Because z/VM provides a very low overhead to manage Linux guests, give your Linux guests at least two virtual processors. In a Linux on System z production environment, application throughput is better when the system is set up based on horizontal growth instead of vertical growth. It is better to develop a production environment with five virtual Linux guests with each one running two JVMs instead of one Linux guest with 10 JVMs running on it.

See *WebSphere Application Server Horizontal Versus Vertical JVM Stacking Report* from January 2012 by Dr. Juergen Doelle and David Sadler for more information about Linux horizontal scalability. This paper can be found at the following website:

http://www.ibm.com/support/techdocs/atsmastr.nsf/5cb5ed706d254a8186256c71006d2e0a/5deed702bbdb14da852579b800543d25/%24FILE/ZSW03213-USEN-00.pdf

Figure 7-5 on page 105 shows a proposed production environment using Java services. This solution setup uses horizontal scalability of the system’s two HTTP servers with the SSL security protocol and was configured with an active-passive high availability service to load balance the workload among the Java servers.

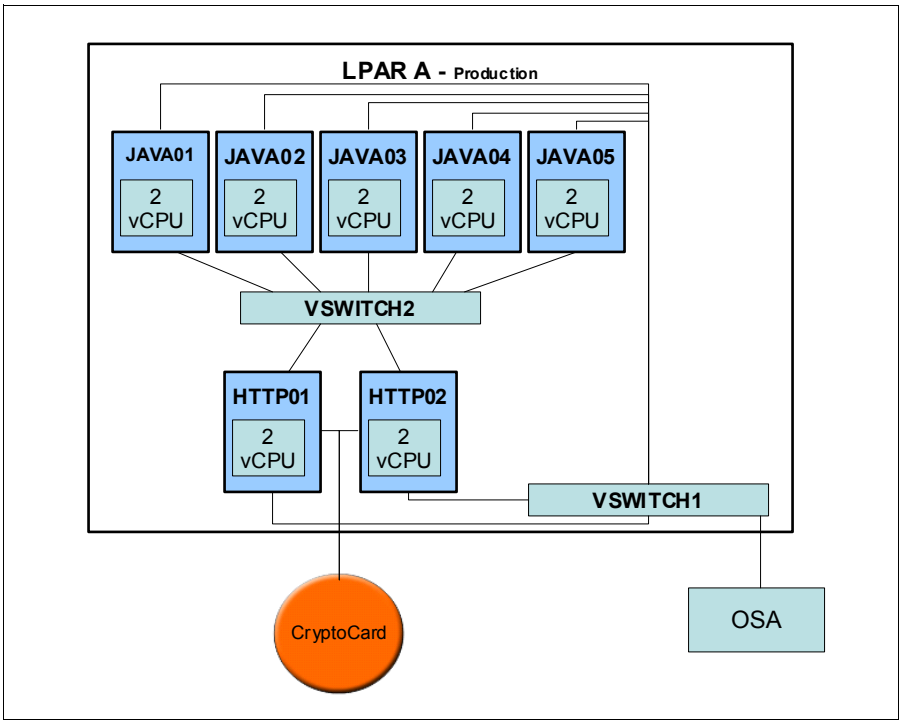


Figure 7-5 Java production system

To minimize the utilization of the IFLs for each SSL handshake, set up the IBM System z Cryptocard to handle it. For more information about how to configure the System z Cryptocard in a Linux environment, see *Security for Linux on System z*, SG24-7728. For information about how to set up the IBM WebSphere Application Server and the IBM HTTP Server, see *Configuring WebSphere V7.0* and *IBM HTTP Server V7.0 to use Cryptographic Hardware for SSL Acceleration on LInux on System z*, M. A. Tebolt, which can be found at the following website:

`http://www.ibm.com/systems/resources/was7_ish7_hwcrypto.pdf`

7.4 Web and application servers

Simply put, a *web server* handles HTTP requests and serves pages for viewing in a web browser, whereas an *application server* serves business logic to application programs through any number of protocols.

There are many options for web servers, such as the IBM WebSphere Application Server, IBM HTTP Server, Apache, and Tomcat.

In this section, we provide information about tuning the IBM WebSphere Application Server and the Apache web server for the production environment.

7.4.1 IBM WebSphere Application Server

Although an application can have the largest affect on performance, tuning and tweaks can be made to the IBM WebSphere Application Server itself.

The WebSphere Application Server has had a common code base across all platforms since version 6, thus, migrating Java applications across platforms is much simpler and WebSphere skills are applicable across multiple platforms.

The WebSphere Application Server provides tunable settings for its major components so that you can adjust the runtime environment to match the characteristics of your application. This is of particular importance when moving from a development environment to a production environment.

IBM WebSphere Application Server performance and tuning guides for the various versions of WebSphere Application Server can be found at the following website:

http://www-01.ibm.com/software/webservers/appserv/was/performance.html

Various performance tuning scripts have been integrated into the IBM WebSphere Application Server service stream that uses the application server's property file configuration features.

The *JVM heap* is the memory that is used by an application running in an application server. Simply put, it is the storage for Java objects. Tune the JVM heap size so that it is as small as possible, but still large enough so that the JVM is not doing excessive garbage collection. Adjust the maximum value so that the JVM heap remains about 30% free.

The following reference document introduces you to the information that you need to create an infrastructure that allows WebSphere applications to run efficiently on Linux for IBM System z:

http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101803

A no-charge HTTP server (the IBM HTTP Server) is included with IBM WebSphere Application Server and is a full-featured web server that is based on the Apache HTTP Server and provides a rich set of Apache features in addition to IBM enhancements.

7.4.2 Apache Web Server

For information about how to install and configure the plug-in for the Apache Web Server on Linux on System z, see the following publication:

The Virtualization Cookbook for z/VM 6.3, RHEL 6.4 and SLES 11 SP3, SG24-8147.

A few adjustments need to be done in order to set up your Apache Web Server to withstand high workloads:

- Multi-processing modules (MPM)

Also known as *pluggable concurrency models*, choosing the best MPM to fit your application is important because it affects the speed and scalability of the Hypertext Transfer Protocol Daemon (that is, web server) or HTTPd. The most common MPMs are *worker* and *prefork*:

Worker:

- Multiple child processes with many threads each
- Each thread handles one connection at a time

Prefork:

- Multiple child processes with one thread each
- Each process handles one connection at a time

According to the Apache website (http://httpd.apache.org/docs/2.2/mpm.html), "*Sites that need a great deal of scalability can choose to use a threaded MPM like worker or event, while sites requiring stability or compatibility with older software can use a prefork.*"

- ServerLimit and MaxClients

The *ServerLimit* is the upper limit on the configurable number of processes and it must be set carefully because if it is oversized, the number of allocated, unused shared memory is more than necessary.

By default, Apache is set to support 150 clients simultaneously, either for worker or prefork installations. This value must be tuned accordingly with your usage expectations.

MaxClients is the maximum number of connections that are processed simultaneously. Any connection attempts over the MaxClients limit are normally queued, up to a number based on the ListenBacklog directive. When a child process is freed at the end of a different request, the connection is then serviced.

Both parameters are directly connected and if you are planning to increase the MaxClients, you must also raise the ServerLimits. According to the Apache website, "*If both ServerLimit and MaxClients are set to values higher than the system can handle, Apache may not start or the system may become unstable.*" [2]

For more information about performance tuning of the Apache web server, see the following website:

http://httpd.apache.org/docs/2.2/misc/perf-tuning.html

[2] http://httpd.apache.org/docs/2.2/mod/mpm_common.html#serverlimit

8

Security considerations

No IT server platform is 100% secure and useful at the same time. If your server is installed in a secure vault, three floors underground in a double-locked room, not connected to any network and switched off, one would say that it was reasonably secure, but it would be a stretch to call it useful.

With z/VM as a hypervisor, Linux on System z gains some of its most valuable security and integrity features. Linux running in an LPAR on System z is attractive for certain highly demanding workloads but for flexible server consolidation exercises, particularly those involving multiple security zones, running Linux as a guest of z/VM provides the best way to gain maximum benefit from the System z investment.

Hypervisors on other platforms do not give the level of virtual machine management provided by z/VM. For this reason, discussing the capabilities of z/VM as a hypervisor, the ways that using z/VM benefits Linux, and the ways to set up z/VM to provide the best environment for operating Linux virtual machines are all important.

One of the most important aspects of moving to a production environment is ensuring that your system is secure. In this section, we provide an overview of security considerations. A more extensive description about security for Linux on System z can be found in the book, *Security for Linux on System z*, SG24-7728. It is strongly recommended that you read that book as well.

© Copyright IBM Corp. 2013. All rights reserved.

8.1 Manage directory in z/VM

The role of directory management in ensuring a secure environment must not be overlooked. Efficient and consistent maintenance of the user directory is important for keeping the system secure and maintainable.

In z/VM, directory management refers to the process that is used to manage the definitions of users and their resources in the z/VM user directory.

Additional information: For more information about the user directory, refer to the z/VM manual, *z/VM: CP Planning and Administration,* SC24-6178-03.

The directory can be managed by using XEDIT and other supporting utilities. This is a manual that can be used when your system has fewer than about 50 virtual machines. As your system becomes more complex, manual directory administration can become cumbersome.

In this section, we describe several features of the IBM Directory Maintenance (DirMaint) facility, which is a priced, optional feature of z/VM. More details about usage of DirMaint are in *Security for Linux on System z*, SG24-7728.

8.2 Secure console access to z/VM virtual machines

Every virtual machine running under z/VM has a console device. For a Linux on System z guest, this device is the virtual equivalent of the panel and keyboard that is attached to a distributed server.

Some virtual machines use this console device more than others. The console on Linux virtual machines is not designed for day-to-day use and is only intended for interactive use during installation or system recovery.

8.3 Secure network access to z/VM

Transport Layer Security (TLS), and its predecessor Secure Sockets Layer (SSL), are cryptographic protocols that provide end-to-end encrypted communications over unsecure networks, at the transport layer.

Establish secure connections to your z/VM system. The use of digital certificates and trust hierarchies help to convey trust across unsecured networks.

Additionally, protect the communication between your host and clients (IBM Personal Communications, or x3270, for instance). By default, Telnet 3270 session data flows unencrypted over the network. A machine that is located between the client and the host can then intercept and dump all communications to get the user IDs and passwords. This is a “man in the middle” attack.

z/VM provides an SSL-capable virtual machine, called *SSLSERV*. Starting with release V5R4.0, this virtual machine is a pure CMS service machine that provides encrypted communications to clients connecting to the z/VM partition. SSL server code is preinstalled as part of a standard z/VM installation, and simply needs to be configured. For the SSL server to operate properly, it is required to have access to a certificate authority (such as Thawte or Symantec Powered by VeriSign) to be able to provide certificates.

The z/VM FTP server is one of the first TCP/IP servers that the user configures on a system. Exchanging data between a workstation and the z/VM systems, for instance to upload the Linux boot files to the 191 MDisk of a guest, is extremely useful.

By default, the z/VM FTP server does not allow any secured connections. Its configuration can be updated to allow or require secure connections. The FTP protocol uses at least two ports. By default, port 20 is used for control connections, and port 21 is used for transferring data. If secured data connections are configured, control connections must be secured as well. z/VM FTPSERVE can be configured to accept SSL-secured connections by way of an update to the `SRVRFTP CONFIG E` file (details about how to do this can be found in *Security for Linux on System z*, SG24-7728).

8.4 Secure your z/VM resources

Securing the access to the z/VM partition is a first step to securing z/VM resources. Most of the work in securing a z/VM partition is to secure the access to the resources managed by z/VM. By securing these resources, you are securing your information and determining who can and who cannot access data from inside Linux and from z/VM.

The z/VM hypervisor has a set of built-in functions that allow a systems administrator to define groups of users according to their needs, to secure access to the resources used by the virtual machines under the control of the CP, and to perform user authentication and authorization. These built-in functions include:

CP privilege classes Default z/VM installations come with a set of commands divided into eight groups, or privilege classes. Depending on their functions, users are part of one group or another.

LAN access restriction Unless specified otherwise, z/VM guest LANs are created in UNRESTRICTED mode, meaning that anybody can connect to the virtual LAN. Specifying the highly recommended RESTRICTED option on the DEFINE LAN command enables LAN access restriction and will help to protect z/VM guest LANs.

Minidisk access Each virtual machine defined in the z/VM user directory is given access to a set of disks or minidisks. These disks (or minidisks) are defined with a given access mode: they can be set up for exclusive use by a virtual machine, or they can be shareable between users. In the latter case, the system administrator can set a password on the disk, which is required each time that a user accesses a disk. The password is set in the z/VM user directory entry for the user.

Encrypted file systems The use of data encryption is one way to protect information. Encryption is the transformation of plain text data into encrypted data by using a key. Without the key, data cannot be converted back to plain text.

The System z platform leverages the use of cryptographic hardware and cryptographic functions that are built into the central processor (CP). By using Central Processor Assist for Cryptographic Function (CPACF), it is possible to offload cryptographic cipher calculations from the central processor, thus considerably reducing the processor cycles of such operations compared to the cost of having them done through software emulation.

You could also consider having all your disks encrypted at a lower level in your storage array subsystem.

8.5 Secure Linux on System z email servers

Mail servers are one of the most common components in a service provider architecture. Often times, companies might use a Linux on System z guest to act as a host for their corporate email server. And although System z enjoys the highest Common Criteria evaluation, email servers are also one of the most common targets for Internet-based attacks.

Many companies rely on both server-side and client-side antivirus and anti-spam applications to protect themselves. Vendors such as TrendMicro, Network Associates, F-PROT, and others provide solutions for Linux on System z to address safety of the enterprise and its data. For more information and a list of currently available anti-virus applications and pointers to their websites, see the following link:

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

8.6 Secure users

Supporting multiple distributed user IDs can be an effort. Even more difficult is providing security for them. To manage and secure user IDs, do the following functions:

- Centralize the repository: Plan to use a centralized repository to handle and maintain use information for multiple Linux on System z guests. This provides simplicity in maintenance and consistency of security policies that are applied to user management.
- Secure network connections: Secure network connections to the user information repository by using an encrypted channel.

8.7 Use an external security manager

An external security manager (ESM) is an external software product that is designed to manage user identities and control access to system resources.

z/VM provides isolation and protection of virtual machines from each other, and between virtual machines and the system, overall. These functions are provided by the z/VM Control Program (CP) and supported by features of the z/Architecture and System z hardware. Although the core capability of security and integrity is provided by CP without an ESM, the management of this capability is quite basic.

For more information, refer to the *z/VM Security and Integrity* document, found at the following website:

http://www-07.ibm.com/systems/includes/content/z/security/pdf/gm130145.pdf

Often, an ESM is not used when designing a development or test environment; the z/VM internal security is often considered “enough”. By the time applications are ready to be deployed into production, it becomes clear that an ESM is required to support the overall business environment that Linux on System z must become a part of.

Following are some of the important reasons to implement an ESM:

Auditability

Additionally, organizations that must comply with government and industry regulations on the control and management of customer and client data will usually require a level of security protection beyond what can be provided using z/VM internal security mechanisms.

Also, to satisfy the requirements of a security audit, it is oftentimes necessary to demonstrate that data owned and managed by a system cannot be accessed by a system belonging to a different security profile. This fact can be difficult to demonstrate without an ESM because minidisks can be attached to any virtual machine with only the minidisk password to protect it.

In addition, providing information detailing which systems successfully accessed certain data is often necessary. z/VM internal security has no simple method of providing such information. (Although deriving access information from z/VM MONITOR data might be possible, this would not be trivial to implement.)

Passwords in the user directory

Without an ESM, passwords for users and minidisks are managed in the z/VM user directory. In the machine-readable binary directory space, the passwords are kept in an obscured format but the source file for the directory holds the passwords in clear text. z/VM administrators are the only users that can see this source file, so the exposure is limited, but an administrator can easily determine any password on the system.

Because the user directory source is only accessible by a z/VM administrator, users are unable to change their own password directly. Some kind of administration interface can be used to allow users to change a password, but this is not part of the base function.

Consistency across systems

When managing a number of z/VM systems that might be sharing disks, security of these systems must be managed consistently to ensure that the data managed in one environment is not leaked through to another system. For example, multiple z/VM systems might have user definitions that define identical direct access storage device (DASD) ranges so that a virtual machine can be started on any available system. If the minidisk definitions are not synchronized between the systems, minidisks can possibly be accessed from an alternate system by using weaker credentials.

By default, the installation documentation for your ESM might enable all functions and features regardless of whether you require those capabilities. For example, the installation process for IBM RACF® enables both the VMMDISK class for minidisk protection, and the VMRDR class for unit record devices. If your installation does not require unit-record device protection, you might want to disable the VMRDR class when the installation process has completed.

The RACF installation process generates a series of RACF commands that are based on the contents of the z/VM user directory, including user and minidisk passwords. This set of commands can be modified before execution, allowing you to enable only the classes and security settings that are needed to implement your required security profile.

Caution: When modifying this command list, be careful to ensure that the system is not made too open, or that essential permission commands are not removed, which can cause features and functions to become inoperative. We recommend editing the file based on the removal of an entire unrequired class, rather than specific permissions within the class.

Refer to Chapter 8, “Best practices” in the IBM Redbooks publication, *Security for Linux on System z*, SG24-7728 for more information about how best to enable the parts of the ESM function that you require.

9

Backup and restore considerations

When migrating from a development system to a production system, it is crucial to have a backup and restore strategy in place before actually needing one. Backing up and restoring data are essential components of data storage management. Backing up your data regularly helps protect your system against the loss of data in the event of a major disaster, or when data is accidentally deleted or becomes corrupted. In this section, we provide an overview of some things to consider when backing up your z/VM system and Linux on System z guests.

Backups can be classified by the way they are created:

- Offline backups are disruptive and require planning
- Online backups do not interrupt operations

There are several different scenarios that you must consider for backup and recovery procedures of both your z/VM system and your Linux on System z guests:

- Image-level backup of z/VM
- File-level backup of z/VM data
- Image-level backup of Linux on System z guests
- File-level backup of Linux on System z guests
- Disaster recovery of both the z/VM system and Linux on System z guests

For more information about topics in this section, including relevant scripts, see the IBM Redbooks publication, *z/VM and Linux Operations for z/OS System Programmers*, SG24-7603.

9.1 Image-level backup of z/VM

The IBM Backup and Restore Manager for z/VM is just one product that facilitates the job of backup and restore functions more efficiently by optimizing operations for each data type. At the very basic level, however, a backup of the z/VM operating system involves making copies of data so that these copies can be used to either restore a state following a disaster (a disaster recovery) or to restore a few files after they have been accidentally deleted or corrupted.

© Copyright IBM Corp. 2013. All rights reserved.

9.1.1 z/VM offline backups

Due to the disruptive nature of this type of backup, proper planning is required because the production system must be stopped. There are two ways of backing up the z/VM operating system: copying to disk and backing up to tape.

Copying a z/VM system to disk

Backing up a z/VM system to disk requires two different z/VM partitions, the one to back up and the one on which the commands are issued. This also requires that the second partition has access to the disks of the z/VM partition to back up.

The process involves the following steps:

1. Shutting down the z/VM system, system A, for backup
2. On the other system, system B, varying system A disks online
3. From system B, copying system A disks by using DDR - z/VM DASD Dump/Restore program, or IBM FlashCopy® (if enabled)
4. Varying system A disks offline
5. ReIPLing system A

This is a disruptive process. All operations have to be stopped before the system is backed up.

Note: Both z/VM and Linux can be saved to disk by using this method.

Backing up a z/VM system to tape

As discussed in the previous chapter, this backup also requires two z/VM partitions, the second accessing the system disk of the first partition. The second partition also needs access to a tape drive.

The process involves the following steps:

1. Attach the tape drive to z/VM system B
2. Rewind the tape
3. Vary online system A disks
4. Copy system A disks onto the tape, one after the other
5. When done, detach the tape drive, and vary offline system A disks

Note: Both Linux and z/VM disks can be saved to tape by using this method.

Backup of z/VM and Linux disks that are controlled from z/OS

It is possible to integrate z/VM and Linux into pre-existing z/OS backup and restore procedures. Linux disks need to be prepared for use with the **dasdfmt** command. This command allows you to format a disk using a specific layout: Linux disk layout, or compatible disk layout.

Compatible disk layout is the default layout for dasdfmt. It means a special handling of the first two tracks of the volume by writing a volume table of contents (VTOC) on the disk. This enables other System z Operating Systems to access this device (for example, for backup purposes).

This VTOC allows a Linux disk to be accessed from z/OS. This disk is seen as a data set that is called, for instance, LINUX.V0X0200.PART0001.NATIVE, and this data set can be saved by using standard z/OS DFSMSdss DUMP commands.

If you format VM DASD from VM by using the CPFMTXA EXEC, it will *always* install a Format 5 label, and a few other and absolutely critical things that VM requires. You must take care to read the ICKDSF doc when initializing (INITing) a VM DASD from IBM MVS™. Otherwise, you can easily write a VTOC that makes it appear that the DASD has a lot of free space available. Do that just *one* single time on a VM page DASD, and get it mounted (by accident, of course) on an MVS system as a public volume, and your VM system will crash in seconds.

Not mentioned when describing backing up z/VM from a z/OS system, but critical if the z/VM system is up and running, just like with z/OS, there are open files and databases that can span multiple volumes. Backing up a running z/VM system from any other system can easily result in “inconsistent file systems”. That is especially true of backing up Linux guests from anywhere but an agent on that Linux guest. Linux heavily caches files in memory, so many open files might not be fully committed to disk at the time of the backup.

Either shut down your Linux guests and their z/VM system before backing it up from any other system (z/OS or even z/VM), or prepare for sporadic system, or application failures as the inconsistent file systems are encountered.

When backed up by jobs on z/OS (and its forerunners) and performing restores, run a CMS file system checker on every CMS minidisk to prevent problems.

9.2 z/VM online backups

Online backups can also be called *hot backups*. They do not require the system to be shut down before performing the backup.

9.2.1 Using SPXTAPE to back up spool files

It is possible to back up spool and system data files—that are printer, reader, and punch files, as well as saved segments, native language support (NLS) files, image libraries—to tape by using the **SPXTAPE** CP command.

The **SPXTAPE** command allows the operator to selectively back up to tape and restore to disk spool and system data files. All files can be dumped, or a filter applied to save only the matching files.

9.2.2 Copying z/VM CPOWNED minidisks

It is possible to perform a hot backup of a running z/VM system by copying z/VM system disks. This is a nondisruptive backup because the system continues to run while the disks are copied.

Each CPOWNED disk has a corresponding fullpack MDisk definition in the `MAINT` user directory.

9.3 File-level backup of z/VM data

If you manually back up your z/VM file-level data, ensure that you have all the necessary information:

- Directory information
- Configuration files
- Log files

You could create tools such as REXX EXECs and automation scripts. One thing that is certain, however, is that you must consider the backup of file-level data in z/VM.

9.4 Linux file-level backup tools

Linux distributions include many tools that can be used to back up and restore data. Some are very basic tools, such as **tar** or **cpio** for instance; some are more complex, such as the IBM Tivoli Storage Manager.

This section introduces some of the tools that are provided by the Linux distributions to perform backups, but is by no means exhaustive.

9.4.1 Tar archiving utility

The **tar** command is a standard tool that is used to create archives from a set of files or directories. This archive can then be compressed and saved onto another set of disks or a tape. Example 9-1 demonstrates the use of this command.

Example 9-1 Archiving files using the tar command

```
itsolnx1:~ # tar cvf varlog_backup.tar /var/log
tar: Removing leading `/' from member names
/var/log/
/var/log/YaST2/
/var/log/YaST2/y2logRPM
/var/log/YaST2/y2log
/var/log/YaST2/y2logmkinitrd
/var/log/YaST2/y2log_bootloader
/var/log/YaST2/volume_info
/var/log/YaST2/disk_dasda
/var/log/YaST2/disk_dasdb
/var/log/YaST2/disk_dasdc
[...]
/var/log/scpm
/var/log/slpd.log
/var/log/boot.omsg
/var/log/dump/
lnxguill:~ # ls -al
total 5280
drwx------  5 root root    4096 May 29 14:36 .
drwxr-xr-x 22 root root    4096 May 29 13:54 ..
-rw-------  1 root root    2098 May 29 13:59 .bash_history
-rw-r--r--  1 root root    1332 Nov 23  2005 .exrc
drwx------  2 root root    4096 May 20 15:07 .gnupg
-rw-------  1 root root    1024 May 20 15:13 .rnd
```

```
-rw-------  1 root root    4072 May 28 10:12 .viminfo
drwxr-xr-x  2 root root    4096 May 20 15:14 .wapi
-rw-r--r--  1 root root   31732 May 20 15:14 autoinst.xml
drwxr-xr-x  2 root root    4096 Jun 16  2006 bin
-rw-r--r--  1 root root 5324800 May 29 14:36 varlog_backup.tar
```

9.4.2 Disk dump: Using the dd command

The **dd** standard command can be used to perform a backup of a directory, a file system, a whole partition, or a disk to a disk or to a file, as shown in Example 9-2. The same command can be used to restore the dumped data.

Example 9-2 Backup using the dd command

```
itsolnx1:~ # dd if=/dev/dasdc1 of=/dev/dasdd1
3654528+0 records in
3654528+0 records out
1871118336 bytes (1.9 GB) copied, 181.17 seconds, 10.3 MB/s
```

9.4.3 rsync command

The **rsync** command copies files either to or from a remote host, or locally on the current host (it does not support copying files between two remote hosts).

The first time that it is called, **rsync** does a full backup of a directory. All subsequent calls to **rsync** will only back up the modified files, hence reducing the time need for the backup.

Example 9-3 Backup with rsync command

```
itsolnx1:~ # rsync -av /var/log/ /mnt
building file list ... done
./
boot.log
boot.msg
boot.omsg
faillog
lastlog
mail
mail.err
mail.info
mail.warn
messages
ntp
scpm
slpd.log
warn
wtmp
zmd-backend.log
zmd-messages.log
YaST2/
YaST2/disk_dasda
YaST2/disk_dasda-1
YaST2/disk_dasdb
YaST2/disk_dasdc
```

```
YaST2/disk_dasdc-1
YaST2/macro_inst_cont.ycp
YaST2/macro_inst_initial.ycp
YaST2/volume_info
YaST2/volume_info-1
YaST2/y2log
YaST2/y2log-1
YaST2/y2log.SuSEconfig
YaST2/y2logRPM
YaST2/y2log_bootloader
YaST2/y2logmkinitrd
YaST2/y2start.log
apparmor/
apparmor/reports-archived/
apparmor/reports-exported/
apparmor/reports/
audit/
audit/audit.log
cups/
dump/
krb5/
news/
news/news.crit
news/news.err
news/news.notice
smpppd/

sent 5282837 bytes  received 906 bytes  3522495.33 bytes/sec
total size is 5279388  speedup is 1.00
lnxguill:~ #

lnxguill:~ # rsync -av /var/log/ /mnt
building file list ... done
lastlog
messages
wtmp

sent 112424 bytes  received 86 bytes  225020.00 bytes/sec
total size is 5279901  speedup is 46.93
```

9.4.4 LVM2 snapshot

Snapshots are a feature of Linux Logical Volume Manager (LVM) 2 that allow an administrator to create a new block device, which presents an exact copy of a logical volume, frozen at some point in time.

Typically, this would be used when some batch processing, a backup for instance, needs to be performed on the logical volume, but you do not want to halt a live system that is changing the data. When the backup of the snapshot device is finished, the system administrator can just remove the device. This facility does require that the snapshot be made at a time when the data on the logical volume is in a consistent state.

In Example 9-4, a snapshot of the logical volume /dev/mapper/vgsystem-varloglv has been created. This logical volume holds the /var/log directory, and the snapshot /dev/mapper/vgsystem-varlogsnapshot presents the contents of this directory frozen at one point in time, in a consistent state, that allows back up of the data.

Example 9-4 Creating an LVM snapshot

```
lnxguill:~ # mount
/dev/dasda1 on / type ext3 (rw,acl,user_xattr)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
debugfs on /sys/kernel/debug type debugfs (rw)
udev on /dev type tmpfs (rw)
devpts on /dev/pts type devpts (rw,mode=0620,gid=5)
/dev/dasdc1 on /usr type ext3 (rw,acl,user_xattr)
/dev/mapper/vgsystem-varloglv on /var/log type ext3 (rw,acl,user_xattr)
securityfs on /sys/kernel/security type securityfs (rw)
lnxguill:~ # lvcreate -s -L 3500MB -n varlogsnapshot /dev/vgsystem/varloglv
  Logical volume "varlogsnapshot" created
lnxguill:~ # mount /dev/mapper/vgsystem-varlogsnapshot /mnt/backup/
lnxguill:~ # cd /mnt/backup/
lnxguill:/mnt/backup # ls -al
total 48
drwxr-xr-x  4 root root  4096 May 29 15:09 .
drwxr-xr-x  3 root root  4096 May 29 15:17 ..
-rw-r--r--  1 root root 11398 May 29 15:09 boot.msg
drwxr-xr-x 10 root root  4096 May 29 15:06 log
drwx------  2 root root 16384 May 29 15:05 lost+found
-rw-r-----  1 root root  1411 May 29 15:11 messages
-rw-r-----  1 root root   347 May 29 15:09 warn
```

9.5 Other kinds of backups

Database vendors also provide their own database backup tools. For instance, IBM DB2 has its own backup and restore commands and utilities, no matter what platform it is being run on. Oracle provides RMAN, Recovery Manager, to handle backup and restore of their databases.

10

Performance considerations

Conducting regular health checks on a system provides utilization and performance information that you can use for capacity planning. For that reason, we provide information about monitoring tools that can be useful to you in ensuring that your system is running at peak performance.

For more information about performance monitoring and tuning, see the IBM Redbooks publication, *Linux on IBM System z: Performance Measurement and Tuning*, SG24-6926.

A single failure of one Linux on System z guest can cause serious performance problems on the entire logical partition (LPAR) because you are running in a virtualized environment. You must consider how a new Linux guest server will affect other servers that you already have on the production logical partition. Keep in mind the following three points:

- Is the number of physical Integrated Facilities for Linux (IFLs) enough to run one more server without affecting the other servers?
- Is the number of Open Systems Adapter (OSA) channels sufficient enough to handle thousands of new connections to the system?
- Is the z/VM paging system already set up for the new server or servers?

Capacity planning and performance knowledge are not only important to solve issues, but also to avoid future problems, especially with overnight workloads. In this chapter, we describe various monitoring tools that are available for both z/VM and Linux on System z.

It is important to keep the entire production environment healthy, including z/VM and Linux on System z. Leverage different approaches to monitor performance and conduct performance tuning to make the System z environment meet business requirements. There are several ways to monitor z/VM and Linux on System z. We examine the following methods of monitoring in this section:

- Use of the z/VM **`INDICATE`** and **`QUERY`** commands
- The IBM Performance Toolkit for VM
- IBM Tivoli OMEGAMON® XE on z/VM and Linux on System z
- Monitoring Linux performance inside the Linux system

© Copyright IBM Corp. 2013. All rights reserved.

10.1 Using z/VM commands

Some very useful and current information from the z/VM system can be obtained when using z/VM control program (CP) commands, such as **INDICATE** and **QUERY**, including CPU, memory, and paging. But sometimes the information provided by the commands is not enough and it is not easy to get an historical report, trend report, and so on. It can also be cumbersome because often times there are only text results available. For more information about these commands, see *z/VM V6R2.0 CP Commands and Utilities Reference,* SC24-6175-03.

10.1.1 CP INIDICATE command

The **INDICATE** command can be used by system resource operators, system programmers, system analysts, and general users to display on the console the use of and contention for system resources. It is often used to get a quick, immediate look at system or user resource consumption. If a performance problem is suspected, this is often the quickest way to identify the characteristics of the problem. There are several variations of the **INDICATE** command that are useful in gathering performance data.

Use the **INDICATE** commands with the following operands to get the following information (see the *z/VM V6R2.0 CP Commands and Utilities Reference,* SC24-6175-03 for more precise information about operands):

- LOAD: The percentage of usage and CPU type for each online processor in your system
- XSTORE, paging, MDC, queue lengths, storage load
- STORAGE value not very meaningful
- USER EXP: More useful than plain USER
- QUEUES EXP: Great for scheduler problems and quick state sampling
- The **INDICATE** command is used for eligible list assessments
- PAGING: Lists users in page wait
- I/O: Lists users in I/O wait
- ACTIVE: Displays number of active users over given interval

10.1.2 CP QUERY command

Use the **QUERY** command with the following operands to get the related information:

- USERS: Number and type of users on system
- SRM: Scheduler/dispatcher settings (LDUBUF, and so on)
- SHARE: Type and intensity of system share
- FRAMES: Real storage allocation
- PATHS: Physical paths to device and status
- ALLOC MAP: Direct access storage device (DASD) allocation
- ALLOC PAGE: How full your paging space is
- XSTORE: Assignment of expanded storage
- MONITOR: Current monitor settings
- MDC: MDC usage
- VDISK: Virtual disk in storage usage
- SXSPAGES: System Execution Space

10.2 IBM Performance Toolkit for VM

The IBM Performance Toolkit for VM provides real-time console (3270 or web) operations and performance monitoring of z/VM and its guests. It also has facilities to archive performance data for historical data processing, which can be handy for capacity-planning exercises.

Performance Toolkit for VM provides enhanced capabilities for a z/VM systems programmer, system operator, or performance analyst to monitor and report performance data. Performance Toolkit for VM is preinstalled in z/VM and runs as a guest user.

PERFKIT is a CMS module and operation that is based on the IUCV *MSG service, a standard facility of z/VM. It also intercepts I/O, SAN Volume Controller, and external interrupts.

The following benefits are provided when using Performance Toolkit for VM:

- Operation of the system operator console in full-screen mode.
- Management of multiple z/VM systems in a single point (local or remote).
- Post-processing of Performance Toolkit for VM history files and VM monitor data captured by the **MONWRITE** utility. It provides health checks, problem determination, and trend analyses information.
- Viewing of performance monitor data using either web browsers or PC-based 3270 emulator graphics.
- TCP/IP performance reporting.
- Sophisticated monitoring software can proactively check for performance issues and automatically act to prevent severe outages.

The IBM Performance Toolkit for VM can also get performance information from the Linux guest kernel. To monitor the Linux data from the kernel, you must first set the APPLMON option in the user directory.

For information about the steps that are required for setting up your Linux virtual servers to be monitored by Performance Toolkit for VM, see the following site:

http://publib.boulder.ibm.com/infocenter/zvm/v6r1/index.jsp?topic=/com.ibm.zvm.v610.hcpl0/fconxcfg.htm

Additionally, see *The Virtualization Cookbook for z/VM 6.3, RHEL 6.4 and SLES 11 SP3*, SG24-8147 for more details.

After this is set, you are able to see the monitor data from the Performance Toolkit for VM 3270 console or from a web browser. Figure 10-1 shows Linux monitoring information from a web browser.

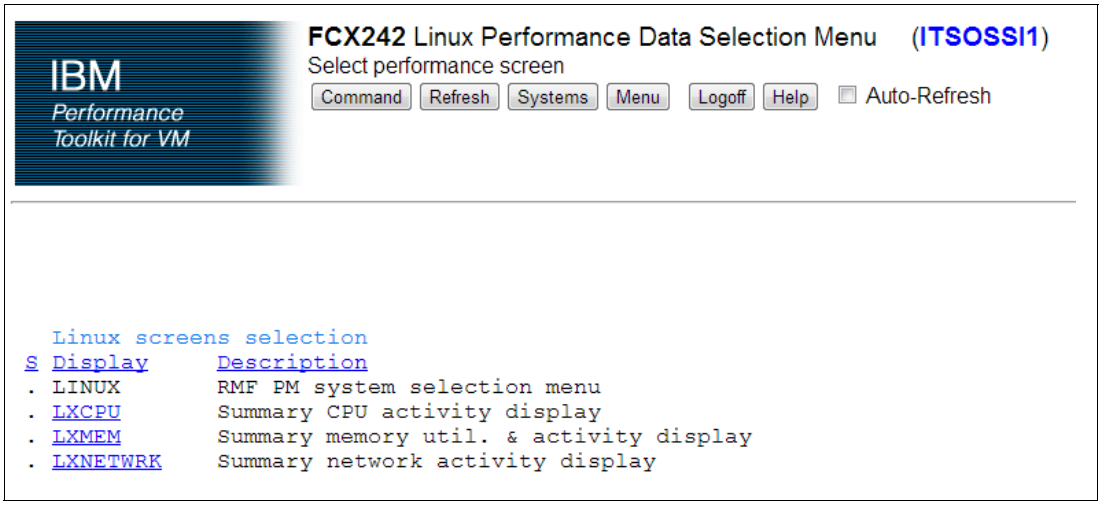


Figure 10-1 Monitoring Linux by using the Performance Toolkit for VM

By clicking the appropriate options, you can see CPU, memory, and network data from the Linux on System z kernel. Figure 10-2 shows memory monitoring data from our Linux on System z guest, ITSOLNX1, from the ITSOSSI SSI member.

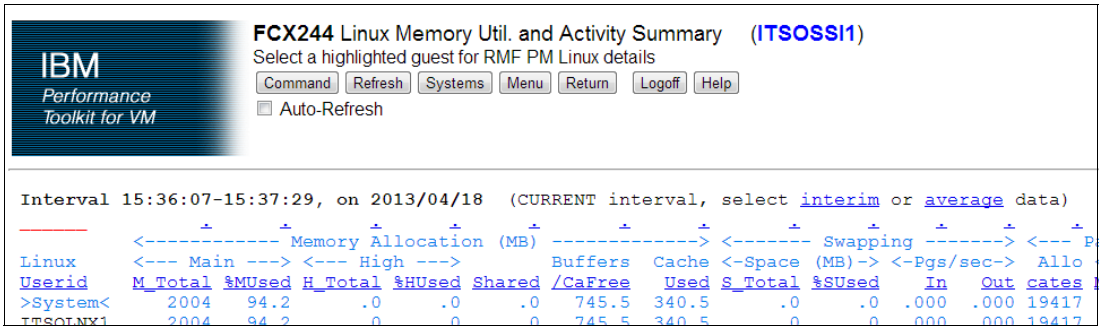


Figure 10-2 Linux memory monitoring data via Performance Toolkit

Note: The IBM Performance Toolkit for VM is an optional priced feature.

The Performance Toolkit for VM user ID is *PERFSVM* and it must be started to collect the z/VM and Linux guest information. After you log on to the system, it is possible to access the Full Screen Operator Console (FCONX) and analyze real-time performance data. In this section, we describe some of the information found in various IBM Performance Toolkit for VM reports that you will find useful for monitoring performance data.

10.2.1 Logical partition information

The IBM Performance Toolkit for VM provides data regarding utilization of an LPAR, and with that information, it is possible to view some LPAR performance-related problems. Access the LPAR (FCX126) and LPARLOG (FCX202) reports to view the LPAR data. To access the reports from the main menu, type the report name (LPAR, for example) on the command area of the monitor window, as shown in Example 10-1 on page 127.

Example 10-1 LPAR data window

```
FCX124          Performance Screen Selection  (FL620       )    Perf. Monitor

General System Data        I/O Data                  History Data (by Time)
1. CPU load and trans.     11. Channel load          31. Graphics selection
2. Storage utilization     12. Control units         32. History data files*
3. SSI data menu*          13. I/O device load*      33. Benchmark displays*
4. Priv. operations        14. CP owned disks*       34. Correlation coeff.
5. System counters         15. Cache extend. func.*  35. System summary*
6. CP IUCV services        16. Reserved              36. Auxiliary storage
7. SPOOL file display*     17. DASD seek distance*   37. CP communications*
8. LPAR data               18. I/O prior. queueing*  38. DASD load
9. Shared segments         19. I/O configuration     39. Minidisk cache*
A. Shared data spaces      1A. I/O config. changes   3A. Storage mgmt. data*
B. Virt. disks in stor.                              3B. Proc. load & config*
C. Transact. statistics    User Data                 3C. Logical part. load
D. Monitor data            21. User resource usage*  3D. Response time (all)*
E. Monitor settings        22. User paging load*     3E. RSK data menu*
F. System settings         23. User wait states*     3F. Scheduler queues
G. System configuration    24. User response time*   3G. Scheduler data
H. VM Resource Manager     25. Resources/transact.*  3H. SFS/BFS logs menu*
                           26. User communication*   3I. System log
Select performance screen with cursor and hit ENTER
Command ===> LPAR
F1=Help  F4=Top  F5=Bot  F7=Bkwd  F8=Fwd  F12=Return
```

The LPAR option provides a view of the utilization for each IFL processor on each logical partition. The information that is displayed on the LPAR load screen includes data about the percentage of busy time of the processor. One of the strengths of the FCX126 LPAR report is that it shows separate utilization values for every logical physical unit (PU) of every partition on the whole central processor complex (CPC). In fact, the interim version of the report, FCX126 INTERIM LPAR, gives you the specified breakout on a time-interval by time-interval basis. Because of this granularity, it is easy to use FCX126 LPAR to see a runaway, overburdened, underused, or stalled partition, or even to see such a logical PU, no matter its partition.

The **%Busy** column is the total activity that the Processor Resource/System Manager (PR/SM) charged to the logical PU. In other words, **%Busy** accounts for both the logical PUs' own work and the PR/SM overhead that the logical PU induced.

The column **%Susp**, also called *suspend time*, describes the amount of time that the real processor is busy and is not serving the monitored LPAR. **%Susp** is just 100% minus the number of z/VM systems that you are accounting for. That is it. No CP Monitor counter directly reports **%Susp.** PERFKIT just calculates **%Susp** by starting with 100% and subtracting out what the z/VM buckets account for. This same information can be seen on the LPARLOG monitor screen also.

Note: The information that is displayed on the LPAR load screen about the percentage of load is not available for partitions that use IFLs.

10.2.2 Processor utilization and waiting time

The definition of the number of IFLs to one LPAR does not follow any specific rule, such as having the same number of real processors as the virtual processors. But, is very important that, in a production LPAR, the number of virtual processors in a unique guest never be higher than the total number of real IFLs designated to the LPAR.

The total number of virtual processors, vCPUs, defined to all of your Linux guest systems can exceed the total number of real processors, IFLs or CPs, that are available to the host z/VM LPAR. In a production environment, it is not recommended that any single Linux guest have more virtual processors than real processors. Doing so can actually degrade performance of that guest.

In Figure 10-3, each example shows that the total number of virtual processors for the Linux guest is more that the six real CPUs available to the LPAR. However, Example A shows a Linux guest with seven virtual CPUs defined on a host LPAR that has only six. Example B shows a better distribution of virtual processors for the Linux guest. As with all production environments, system monitoring should be done to identify long spikes in CPU utilization and actions should be taken to avoid performance issues.

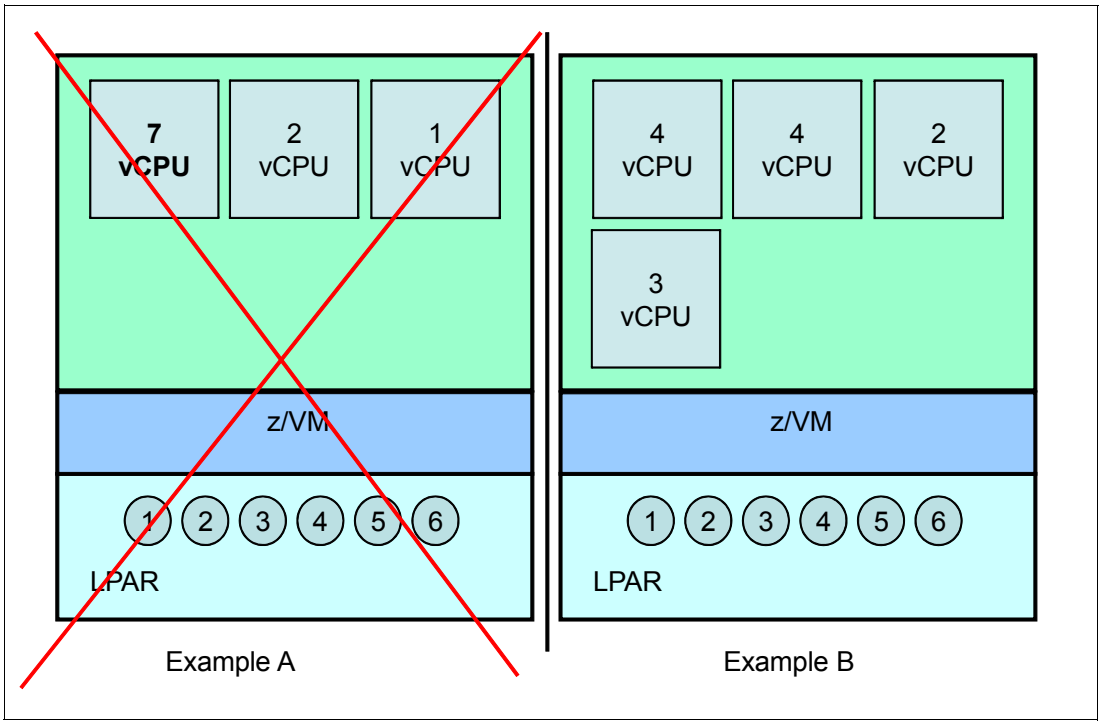


Figure 10-3 Number of real versus virtual processors

You can obtain information about processor time by accessing the CPU monitor panel in the IBM Performance Toolkit for VM, as shown in Example 10-2 on page 129. From the CPU panel, the information is more consolidated and not only processor utilization is on the panel, but also the login LPAR information appears on the panel. Important information includes the **%CP** time, which indicates that the supervisor is running on the processors, and **%EMU**, which is the amount of time that VM guests are running on the processors. The wait time, **%WT**, includes information about the idle time (page wait or I/O wait) so do not worry about this number. Total logical load, **%LOGLD**, is useful to determine CPU bottlenecks because the difference between **%CPU** and **%LOGLD** is the total elapsed time, which is not displayed.

The User Status information shows how your Linux guests are running and if the system is processing your request in the way that it should. If you see high values for **%in-Q users in PGWAIT** or **% in-Q users in IOWAIT**, access the USTAT and USTLOG reports to get a better understanding about these wait times and the system load.

Example 10-2 CPU report panel

```
FCX100      CPU 2817  SER A3BD5  Interval 12:04:07 - 12:05:07    Perf. Monitor

CPU Load                                                        Status or
PROC TYPE %CPU  %CP %EMU %WT %SYS %SP %SIC %LOGLD               ded. User
P00  CP     13    1   12  87    0   0   59     13               Master
P01  CP     16    0   15  84    0   0   33     16               Alternate
P02  CP     12    0   11  88    0   0   71     12               Alternate
P03  CP     12    0   11  88    0   0   45     12               Alternate

Total SSCH/RSCH    325/s     Page rate        .0/s     Priv. instruct.   175/s
Virtual I/O rate    74/s     XSTORE paging    .0/s     Diagnose instr.    24/s
Total rel. SHARE    3100     Tot. abs SHARE     0%

Queue Statistics:     Q0     Q1     Q2     Q3     User Status:
VMDBKs in queue        1      0      0      1     # of logged on users      21
VMDBKs loading         0      0      0      0     # of dialed users          0
Eligible VMDBKs               0      0      0     # of active users          8
El. VMDBKs loading            0      0      0     # of in-queue users        2
Tot. WS (pages)     2897      0      0 547364     % in-Q users in PGWAIT     0
Reserved                                         % in-Q users in IOWAIT    34
85% elapsed time    .372   .062   .496  2.976     % elig. (resource wait)    0

Transactions      Q-Disp    trivial   non-trv     User Extremes:
 Average users         .0         .0        .0     Max. CPU %   ITSOLNX3   48.7
 Trans. per sec.       .9         .3        .3     Reserved
 Av. time (sec)      .064       .003      .000     Max. IO/sec  ITSOLNX3   80.3
 UP trans. time                 .003      .000     Max. PGS/s   ........  .....
 MP trans. time                 .000      .000     Max. RESPG   ITSOLNX3 547375
 System ITR (trans. per sec. tot. CPU)     2.9     Max. MDCIO   ........  .....
 Emul. ITR (trans. per sec. emul. CPU)     3.1     Max. XSTORE  ........  .....
```

The CPU monitor panel can be your first step to discover performance problems because this also provides information regarding the most used guests on the LPAR in the User Extremes section.

The user “wait state” panel (USTAT) in Example 10-3 on page 130 shows, in detail, information about how the z/VM guests are using the processors and waiting for resources. The **%ACT** column describes the value of percentage of time that the system is active, using resources or waiting to use a resource. The **%RUN** column describes the amount of time that a user was actually using the real processor and the **%CPU** column describes the amount of time the user was waiting for the resource. If any of your Linux guests have a higher **%CPU** value, most likely, one of the Linux services is experiencing a slow response time. In this case, it is recommended to look at the application running on the Linux guest to better understand what is causing the wait state. In a production environment, you might want to dynamically add one or more virtual processors to help alleviate the problem.

Example 10-3 User “wait state” report panel

```
FCX114      CPU 2817  SER A3BD5  Interval 11:56:07 - 11:57:07    Perf. Monitor
.          ____     .    .    .    .    .    .    .    .   .   .   .    .    .
                                                          <-SVM and->
Userid     %ACT  %RUN %CPU %LDG %PGW %IOW %SIM %TIW %CFW %TI %EL %DM %IOA %PGA
>System<     6    22    0    0    0    0    0   61    0   0   0   0   17    0
ITSOLNX3    97    28    0    0    0    0    0   52    0   0   0   0   21    0
TCPIP       23     0    0    0    0    0    0  100    0   0   0   0    0    0
ACCSRV1      0   ...  ...  ...  ...  ...  ...  ...  ... ... ... ...  ...  ...
BKRBKUP      0   ...  ...  ...  ...  ...  ...  ...  ... ... ... ...  ...  ...
BKRCATLG     0   ...  ...  ...  ...  ...  ...  ...  ... ... ... ...  ...  ...
DIRMSAT3     0   ...  ...  ...  ...  ...  ...  ...  ... ... ... ...  ...  ...
DISKACNT     0   ...  ...  ...  ...  ...  ...  ...  ... ... ... ...  ...  ...
DTCVSW1      0   ...  ...  ...  ...  ...  ...  ...  ... ... ... ...  ...  ...
DTCVSW2      0   ...  ...  ...  ...  ...  ...  ...  ... ... ... ...  ...  ...
EREP         0   ...  ...  ...  ...  ...  ...  ...  ... ... ... ...  ...  ...
FTPSERVE     0   ...  ...  ...  ...  ...  ...  ...  ... ... ... ...  ...  ...
GCS          0   ...  ...  ...  ...  ...  ...  ...  ... ... ... ...  ...  ...
MAINT        0   ...  ...  ...  ...  ...  ...  ...  ... ... ... ...  ...  ...
OPERATOR     0   ...  ...  ...  ...  ...  ...  ...  ... ... ... ...  ...  ...
OPERSYMP     0   ...  ...  ...  ...  ...  ...  ...  ... ... ... ...  ...  ...
PERFSVM      0   ...  ...  ...  ...  ...  ...  ...  ... ... ... ...  ...  ...

; when you go right with the screen

FCX114      CPU 2817  SER A3BD5  Interval 12:00:07 - 12:01:07    Perf. Monitor
 .          ____ >.   .   .   .    .    .    .    .   .   .   .    .
                 >  <-SVM and->                    <--%Time spent in-->  Nr of
 Userid     %ACT >W %TI %EL %DM %IOA %PGA %LIM %OTH  Q0  Q1  Q2  Q3 EQ-3  Users
 >System<     6 >0   0   0   0   32    0    0    0  21   0   0  79    0     21
 ITSOLNX3   100  0   0   0   0   40    0    0    0   0   0   0 100    0
 TCPIP       27  0   0   0   0    0    0    0    0 100   0   0   0    0
 ACCSRV1      0 .. ... ... ...  ...  ...  ...  ... ... ... ... ...  ...
 BKRBKUP      0 .. ... ... ...  ...  ...  ...  ... ... ... ... ...  ...
 BKRCATLG     0 .. ... ... ...  ...  ...  ...  ... ... ... ... ...  ...
 DIRMSAT3     0 .. ... ... ...  ...  ...  ...  ... ... ... ... ...  ...
 DISKACNT     0 .. ... ... ...  ...  ...  ...  ... ... ... ... ...  ...
 DTCVSW1      0 .. ... ... ...  ...  ...  ...  ... ... ... ... ...  ...
 DTCVSW2      0 .. ... ... ...  ...  ...  ...  ... ... ... ... ...  ...
 EREP         0 .. ... ... ...  ...  ...  ...  ... ... ... ... ...  ...
 FTPSERVE     0 .. ... ... ...  ...  ...  ...  ... ... ... ... ...  ...
 GCS          0 .. ... ... ...  ...  ...  ...  ... ... ... ... ...  ...
 MAINT        0 .. ... ... ...  ...  ...  ...  ... ... ... ... ...  ...
 OPERATOR     0 .. ... ... ...  ...  ...  ...  ... ... ... ... ...  ...
 OPERSYMP     0 .. ... ... ...  ...  ...  ...  ... ... ... ... ...  ...
 PERFSVM      0 .. ... ... ...  ...  ...  ...  ... ... ... ... ...  ...
 Select a user for user details or IDLEUSER for a list of idle users
```

A common cause of wait states occurs when either the number of IFLs, memory, or I/O channels were defined too low during the capacity planning phase.

Use the **%IOA** column to help identify channel constraints. However, be aware that this constraint could be either disk, network, or memory I/O constraints. You must also use the **%CFW** column because it might be related to memory reorganization. This happens frequently when a guest has a large memory allocated (16 GB or more). A large guest

allocates a significant number of resident pages and z/VM performs demand scans to reorder pages. If you experience a situation such as this, the **REORDER** CP command can help prevent the reorganization of the pages.

Page reorder is a process that supports z/VM storage management. This process looks for storage frames that are not being referenced and updates an index of which virtual server frames have been referenced for a period. The page reorder process does not really reorder pages. It is more accurate to think of it as reordering a set of pointers to pages.

If z/VM runs out of main storage memory, any unreferenced frames are the first ones to be paged. A peculiarity about **REORDER** is that during this process the virtual machine is frozen. The cost of **REORDER** is proportional to the number of resident frames for the virtual machine and can take up to 1 second per each 8 GB Linux virtual server storage resident frame. This process is done periodically on a virtual machine basis and was widely used and had significant value at one time. That value diminished over time, particularly for Linux environments. Up to z/VM 6.2, it is possible to enable **REORDER** for a specific virtual server or for the entire environment. We suggest disabling it for the entire environment.

The **REORDER** CP command prevents the reorganization of the pages. Therefore, when you have a large Linux guest and you are seeing high values in the **%CFW** column of the USTAT report, you can use the command that is shown in Example 10-4 to stop the **REORDER** process.

Example 10-4 Reorder command

```
q reorder itsolnx3
Reorder is ON for user ITSOLNX3
Ready; T=0.01/0.01 15:11:11

set reorder off itsolnx3
Reorder is OFF for user ITSOLNX3
Ready; T=0.01/0.01 15:15:31
```

For more information about the reorder schedules, see the following website:

http://www.vm.ibm.com/perf/tips/reorder.html

The processor reports are only a start; not all performance problems are solved by adding more virtual processors or virtual memory.

10.2.3 Total/Virtual processor ratio

The Total/Virtual (T/V) processor ratio is a z/VM system indicator that shows the average ratio of total to virtual processor time for all processors. This indicator can be seen on the System Summary Log Screen (SYSSUMLG) of Performance Report FCX225. The closer to 1.0 T/V value, the better the z/VM efficiency. A ratio of over 1.30 should be further examined. Example 10-5 shows the FCX225 report with the T/V indicator.

Example 10-5 Total/Virtual processor ratio

```
FCX225      CPU 2827  SER CB8D7  Interval 00:00:51 - 11:45:51    Perf. Monitor

          <------- CPU --------> <--Users--> <---I/O---> <Stg> <-Paging--> <Spl
               <--Ratio-->                    SSCH  DASD Users <-Rate/s-->
Interval    Pct        Cap-  On-  Log-       +RSCH  Resp    in PGIN+ Read+ Page
End Time   Busy   T/V  ture line   ged Activ    /s  msec Elist PGOUT Write    /
>>Mean>>    3.5  1.14 .9564  4.0    22     8 331.5   1.4    .0    .0    .0    .
```

```
11:32:51    3.5  1.14 .9571  4.0    22     8 328.8   1.4    .0    .0    .0    .
11:33:51    3.5  1.14 .9563  4.0    22     9 334.6   1.4    .0    .0    .0    .
11:34:51    3.4  1.14 .9557  4.0    22     8 331.2   1.4    .0    .0    .0    .
11:35:51    3.4  1.14 .9563  4.0    22     8 329.7   1.4    .0    .0    .0    .
11:36:51    3.5  1.14 .9563  4.0    22    10 334.4   1.4    .0    .0    .0    .
11:37:51    3.4  1.14 .9562  4.0    22     8 323.3   1.4    .0    .0    .0    .
11:38:51    3.4  1.14 .9553  4.0    22     8 325.1   1.4    .0    .0    .0    .
11:39:51    3.4  1.14 .9565  4.0    22     8 332.2   1.4    .0    .0    .0    .
11:40:51    3.4  1.14 .9561  4.0    22     8 328.6   1.4    .0    .0    .0    .
11:41:51    3.5  1.14 .9570  4.0    22    10 328.6   1.4    .0    .0    .0    .
11:42:51    3.6  1.13 .9581  4.0    22     8 329.9   1.4    .0    .0    .0    .
11:43:51    3.5  1.14 .9562  4.0    22     8 331.8   1.4    .0    .0    .0    .
11:44:51    3.3  1.14 .9542  4.0    22     8 328.2   1.4    .0    .0    .0    .
11:45:51    3.4  1.14 .9560  4.0    22     8 328.3   1.4    .0    .0    .0    .
See also REDISP for a more extensive load summary
Command ===>
F1=Help  F4=Top  F5=Bot  F7=Bkwd  F8=Fwd  F10=Left  F11=Right  F12=Return
```

10.2.4 z/VM resource manager (SRM)

System resource managers (SRMs) are system-wide parameters that are used by the z/VM scheduler to set the priority of system resource access. They determine which virtual server receives a resource. z/VM can over commit system resources and sometimes it can cause what is known as *thrashing*.

Thrashing is a lot of unproductive paging activity and is caused by servers wanting more memory than exists, all at the same time. At the point where z/VM does not have enough memory to meet each server's needs, z/VM starts paging. There is a point where z/VM can spend more time doing paging than performing work. This is thrashing.

The scheduler process is shown in Figure 10-4 on page 133 and is a collection of algorithms that manage the scheduling of virtual machines for real processor time. It controls three lists: the dispatch list, the eligible list, and the dormant list. This figure illustrates the lists that are maintained by the scheduler and indicates the flow of virtual machines from one list to another.

The z/VM scheduler has a sophisticated mechanism to stop users from thrashing. It controls thrashing from three different perspectives: storage, paging, and processor.

The mechanism for controlling thrashing is to put users on an eligible list, meaning that these users want to consume resource, but there is not enough resource to sustain them, so they are not dispatched. When there is enough resource or certain deadlines pass, the user is dispatched. This can look like users are “hung”. However, they are *not* hung; they are waiting until there is sufficient resource to run efficiently.

Figure 10-4 on page 133 shows a sample virtual machine scheduling flow.

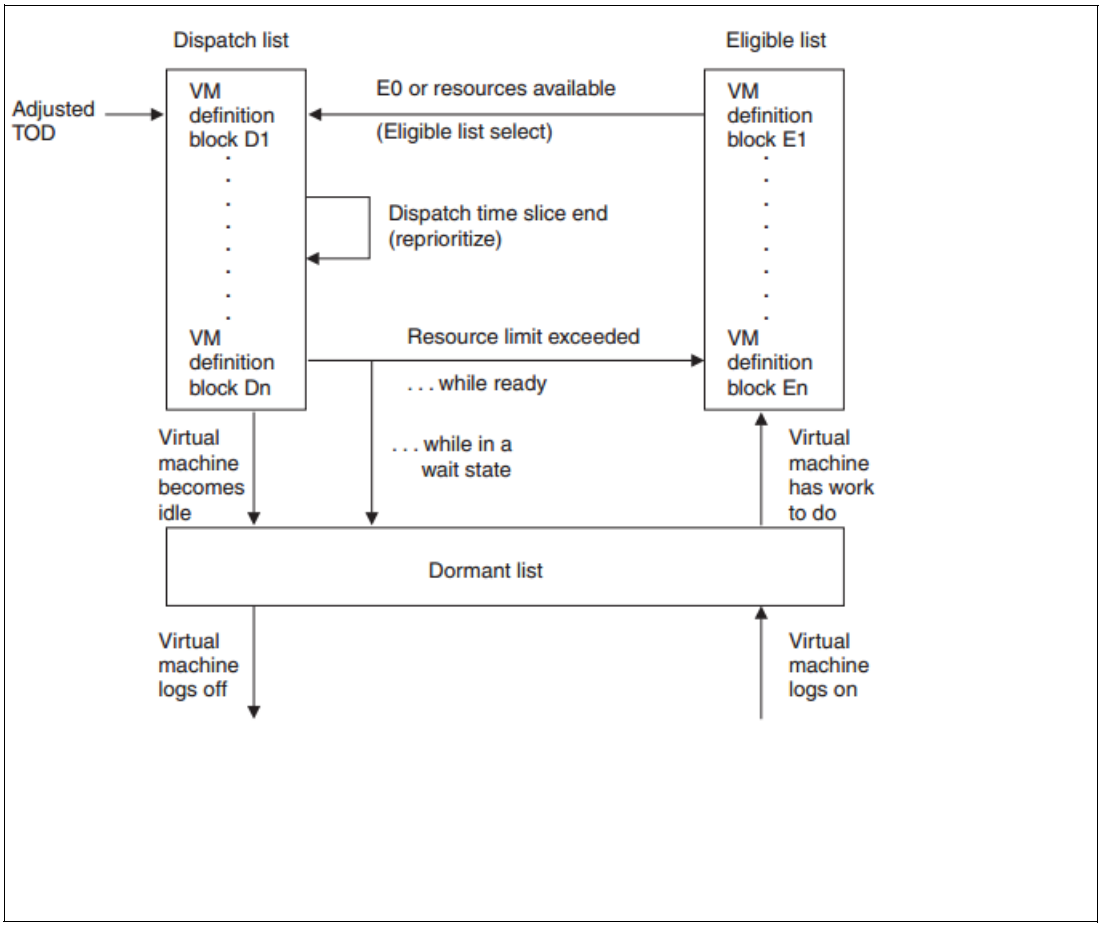


Figure 10-4 Virtual machine scheduling flow

Example 10-6 shows an example of report FCX154. This report shows the SRM configuration for the z/VM environment. Ensure that the SRM STOREBUF values are over 100% to avoid thrashing situations.

Example 10-6 z/VM SRM definitions

```
FCX154      CPU 2827  SER CB8D7        System   Settings         Perf. Monitor

Initial Scheduler Settings: 2013/04/11 at 15:02:51
DSPSLICE (minor) 5.000 msec.             IABIAS Intensity    90 Percent
Hotshot T-slice  1.999 msec.             IABIAS Duration      2 Minor T-slices
DSPBUF Q1        32767 Openings          STORBUF Q1 Q2 Q3   300 % Main storage
DSPBUF Q1 Q2     32767 Openings          STORBUF Q2 Q3      250 % Main storage
DSPBUF Q1 Q2 Q3  32767 Openings          STORBUF Q3         200 % Main storage
LDUBUF Q1 Q2 Q3    100 % Paging exp.     Max. working set  9999 % Main storage
LDUBUF Q2 Q3        75 % Paging exp.     Loading user         5 Pgrd / T-slice
LDUBUF Q3           60 % Paging exp.     Loading capacity     5 Paging expos.
LIMITHARD algorithm    Consumption

Command ===>
F1=Help  F4=Top  F5=Bot  F7=Bkwd  F8=Fwd  F12=Return
```

In development environments, some contention generated by the overhead of the numbers of virtual processor and memory that is related to the number of real IFL and main storage memory is expected. If it is not causing a problem in the development of an application, it is acceptable. If it is causing development servers to hang, the system administrator still can increase the *SHARE* value and gives more priority to a specific server; but in a production environment, that should not happen. This is why it is so important to monitor and understand z/VM schedules and the overall production system health. Ensure before going into production that there is enough capacity to move a server to production.

We describe more about SHARE value in the next section.

10.2.5 SHARE values

A primary factor in the calculation of both the eligible and dispatch priorities of a virtual machine is the share factor that is allocated by the system administrator. The *SHARE* value is a percentage of the system available resources that a Linux guest is allowed to use. This can be set as *absolute* or *relative* with the default being a relative percentage of 100 for each guest.

SHARE RELATIVE specifies that this guest is to receive a target minimum relative share value. The amount of scheduled system resources available to relative share users is the total of resources. The portion that this user receives is **nnnnn** divided by the sum of the **nnnnn**'s of all relative share users. For example, if one guest's relative share is 100 and another guest's relative share is 200, the second user gets twice as much access to system resources as the first. If no share is specified in a guest's z/VM user directory entry, the share defaults to a relative share of 100. The `nnnnn` operand can range 1 - 10000. Be aware that the share value is divided by the number of virtual processors available at the virtual machine, which means that production virtual machines that have more processors need to have high share values.

10.2.6 QUICK DISPATCH option

When the quick dispatch (`QUICKDSP`) option is assigned to a virtual server, the server is added to the dispatch list immediately, whenever it has work to do, without waiting in the eligible list. Because the quick dispatch virtual servers are always in the set of virtual machines being dispatched, they are considered for dispatching more frequently than other virtual machines, which might spend time waiting for system resources. Only exceptional Linux virtual servers should have `QUICKDSP ON`, otherwise the z/VM dispatcher is overloaded with dispatch requests.

10.2.7 z/VM memory subsystem

To maintain system health, it is necessary to monitor the paging subsystems as well as Linux memory utilization. Because we are already working with two levels of memory management, the z/VM hypervisor and the Linux operating system, avoid a third memory management system such as those from applications such as database automatic memory allocation and a variable Java heap size.

When using the IBM Performance Toolkit for VM, it is possible to identify memory constraints in the z/VM logical partition. The first indicator that you are running out of memory is the allocation of the paging system. The full overview of the memory allocation can be accessed using the storage panel from Performance Toolkit.

Some of the following information can be found in this memory allocation report:

- Total real storage that is allocated and available
- Total expanded memory that is available
- Total minidisk cache and utilization
- Size of the VDisk area and VDisk allocation

Example 10-7 shows the STORAGE report.

Example 10-7 Storage report

```
FCX103      CPU 2827  SER DB8D7  Interval 10:53:10 - 10:54:10    Perf. Monitor

Main storage utilization:                XSTORE utilization:
Total real storage         30'720MB      Total available              2'048MB
Total available            30'720MB      Att. to virt. machines           0kB
Offline storage frames           0       Size of CP partition         2'048MB
SYSGEN storage size        30'720MB      CP XSTORE utilization            0%
Shared storage             50'868kB      Low threshold for migr.      1'200kB
FREE stor. subpools         3'836kB      XSTORE allocation rate           0/s
Subpool stor. utilization       91%      Average age of XSTORE blks      ...s
Total DPA size             30'416MB      Average age at migration        ...s
Locked pages                  3247
Trace table                 1'300kB      MDCACHE utilization:
Pageable                   30'402MB      Min. size in XSTORE              0kB
Storage utilization              0%      Max. size in XSTORE          2'048MB
Tasks waiting for a frame        0       Ideal size in XSTORE             0kB
Tasks waiting for a page         0/s     Act. size in XSTORE          7'276kB
Standby real stor. size      4'096MB     Bias for XSTORE               1.00
Reservd real stor. size          0kB     Min. size in main stor.          0kB
                                         Max. size in main stor.     30'720MB
Paging / spooling activity:              Ideal size in main stor.     8'192MB

Page moves <2GB for trans.       0/s     Act. size in main stor.    114'592kB
Fast path page-in rate           0/s     Bias for main stor.           1.00
Long path page-in rate           0/s     MDCACHE limit / user         1'024MB
Long path page-out rate          0/s     Users with MDCACHE inserts       1
Page read rate                   0/s     MDISK cache read rate            0/s
Page write rate                  0/s     MDISK cache write rate       ...../s
Page read blocking factor       ...      MDISK cache read hit rate        0/s
Page write blocking factor      ...      MDISK cache read hit ratio      80%
Migrate-out blocking factor     ...
Paging SSCH rate                 0/s     VDISKs:
SPOOL read rate                  0/s     System limit (blocks)         3606k
SPOOL write rate                 0/s     User limit (blocks)          144000
                                         Main store page frames           0
Reorder Settings:                        Expanded stor. pages             0
Reorder for System               On      Pages on DASD                    0

Memory Constraint Relief:
Pageable memory <2G              On

Paging SSCH rate                 0/s     VDISKs:
SPOOL read rate                  0/s     System limit (blocks)         3606k
SPOOL write rate                 0/s     User limit (blocks)          144000
                                         Main store page frames           0
Reorder Settings:                        Expanded stor. pages             0
```

```
Reorder for System              On        Pages on DASD                   0

Memory Constraint Relief:
Pageable memory <2G             On
Pageable memory >2G             On
Demand scans <2G                On
Demand scans >2G                On
Allocate pageable freeze        Off
Multiplier before Min/Max      274
Turnover Rate Mult <2G         274
Turnover Rate Mult >2G        3822
Requests waiting anywhere        0
Requests waiting <2G             0
```

Main storage memory information is more detailed in the Storage Utilization Log (STORLOG) report where it is possible to identify the number of available frames in the dynamic paging area (DPA). The DPA holds the virtual storage of virtual machines and the storage used per CP for spool buffers, trace tables, and virtual disks. There is a direct relationship between the expanded storage area and the dynamic storage area. For however large the expanded storage area is, the dynamic storage area is lower. But if you have fast enough DASD devices and sufficient control unit cache, it is possible to achieve optimum performance. In a production z/VM system, it is not recommended to have many paging requests. Therefore, ensuring that you have enough DPA is the best practice that you can follow. If you are satisfying paging area needs, allocate fast disk devices such as Small Computer System Interface (SCSI) devices.

To simulate the DPA, we allocated, in one single Linux guest, four different RAM disks of 2.1 GB each and wrote zeros to a specific file to allocate memory in Linux, as shown in Example 10-8.

Example 10-8 Linux guest memory allocation

```
[root@itsolnx4 ~]# for i in 1 2 3 4 ; do mount -t tmpfs none /var/tmp/ram$i -o
size=3200M  ; done

[root@itsolnx4 ~]# for i in 2 3 4 ; do dd if=/dev/zero of=/var/tmp/ram$i/zero.txt
bs=3200000000 count=1 ; sleep 60 ; done
0+1 records in
0+1 records out
2147479552 bytes (2.1 GB) copied, 1.59541 seconds, 1.3 GB/s
0+1 records in
0+1 records out
2147479552 bytes (2.1 GB) copied, 1.78486 seconds, 1.2 GB/s
0+1 records in
0+1 records out
2147479552 bytes (2.1 GB) copied, 1.7951 seconds, 1.2 GB/s
```

The result of these commands is shown in Example 10-9 on page 137. In this example, the STORLOG report (FCX253) shows information about used and available memory frames in the dynamic paging area. In a z/VM system, the higher utilization of the DPA (Stor Util %) results in a higher number of page activities. You want to avoid that in production environments.

Example 10-9 Storage utilization log panel

```
FCX253      CPU 2827  SER DB8D7  Interval 00:00:10 - 17:02:10    Perf. Monitor

          <----------------------------- Storage Utilization (Page Frames) ----
          <------- DPA ------->  Stor                 <---------- Locked ----
 Interval <--Pageable--> Nonpgb  Util    Save   Track  <-LOCK REAL-> <-SXS Alia
 End Time   <2GB    >2GB   <2GB     %   Areas   Cache    <2GB   >2GB  Total   L
 >>Mean>> 520940 7261924   3347     3      72   75056       0      0     33
 16:49:10 520726 7261356   3561     8      72  175212       0      0     40
 16:50:10 520726 7261356   3561     9      72  175200       0      0     40
 16:51:10 520726 7261356   3561    10      72  175200       0      0     40
 16:52:10 520726 7261356   3561    10      72  175224       0      0     40
 16:53:10 520722 7261357   3565    11      72  179300       0      0     40
 16:54:10 520720 7261356   3567    11      72  179040       0      0     40
 16:55:10 520641 7261356   3646    12      72  202961       0      0     40
 16:56:10 520647 7261356   3640    16      72  202948       0      0     40
 16:57:10 520648 7261284   3639    22      72  202972       0      0     40
 16:58:10 520648 7261274   3639    29      72  203046       0      0     40
 16:59:10 520648 7261274   3639    29      72  203046       0      0     40
 17:00:10 520648 7261274   3639    29      72  203070       0      0     40
 17:01:10 520648 7261271   3639    29      72  203058       0      0     40
 17:02:10 520648 7261270   3639    29      72  203058       0      0     40
```

It is also necessary to understand and monitor the z/VM paging system (3.2, “Memory planning for z/VM” on page 35). By understanding paging utilization and server peak times, you are able to allocate more virtual memory to Linux guests than the real memory that you have in z/VM.

Performance Toolkit also has a report to describe the utilization of memory per Linux guest. The User Page Data (UPAGE) panel describes information including details about how many pages are in real z/VM central memory storage and how much memory is in the paging memory area. It also describes the number of pages in expanded storage memory and also DASD paging devices. In Example 10-10, the information about memory utilization per Linux guest and by all other z/VM guests is displayed. Each page is 4 KB in size.

Example 10-10 User Page Data panel

```
FCX113      CPU 2827  SER DB8D7  Interval 17:46:10 - 17:47:10    Perf. Monitor
 ______        .      .     .      .      .    .    .    .       .       .      .
           Data  <--------- Paging Activity/s ---------->  <----------------- N
         Spaces  <Page Rate>   Page  <--Page Migration-->                 <-Resi
 Userid   Owned  Reads Write Steals  >2GB> X>MS MS>X X>DS    WSS Resrvd  R<2GB
 >System<    .0     .0    .0     .0     .0   .0   .0   .0 214389      0  14688
 ACCSRV1      0     .0    .0     .0     .0   .0   .0   .0    182      0     42
 BKRBKUP      0     .0    .0     .0     .0   .0   .0   .0    833      0      1
 BKRCATLG     0     .0    .0     .0     .0   .0   .0   .0    855      0      1
 DIRMSAT2     0     .0    .0     .0     .0   .0   .0   .0    607      0    548
 DISKACNT     0     .0    .0     .0     .0   .0   .0   .0   1248      0      8
 DTCVSW1      0     .0    .0     .0     .0   .0   .0   .0   2674      0     10
 DTCVSW2      0     .0    .0     .0     .0   .0   .0   .0   2678      0      8
 EREP         0     .0    .0     .0     .0   .0   .0   .0   1228      0      0
 FTPSERVE     0     .0    .0     .0     .0   .0   .0   .0   1432      0      0
 GCS          0     .0    .0     .0     .0   .0   .0   .0     51      0      4
 ITSOLNX1     0     .0    .0     .0     .0   .0   .0   .0  40811      0   3063
```

```
 ITSOLNX2      0     .0    .0     .0     .0   .0   .0   .0 191251      0  13410
 ITSOLNX3      0     .0    .0     .0     .0   .0   .0   .0 313982      0  22045
 ITSOLNX4      0     .0    .0     .0     .0   .0   .0   .0  4558k      0 310350
 MAINT         0     .0    .0     .0     .0   .0   .0   .0   1608      0    593

;moving screen to right (F11)

FCX113      CPU 2827  SER DB8D7  Interval 17:47:10 - 17:48:10    Perf. Monitor
 ______        . >     .      .      .      .      .      .      .      .
           Data >------------ Number of Pages ----------------->
         Spaces >        <-Resident-> <--Locked-->                    Stor  Nr of
 Userid   Owned >Resrvd  R<2GB  R>2GB L<2GB  L>2GB  XSTOR   DASD    Size  Users
 >System<    .0 >     0  14688 199733     5     19      0      0  1346M     24
 ACCSRV1      0       0     42    140     0      0      0      0    32M
 BKRBKUP      0       0      1    832     0      0      0      0   128M
 BKRCATLG     0       0      1    854     0      0      0      0   128M
 DIRMSAT2     0       0    548     61     0      0      0      0   128M
 DISKACNT     0       0      8   1285     0      0      0      0    32M
 DTCVSW1      0       0     10   2665     0      1      0      0    32M
 DTCVSW2      0       0      8   2720     8     42      0      0    32M
 EREP         0       0      0   1272     0      0      0      0    32M
 FTPSERVE     0       0      0   1433     0      1      0      0    32M
 GCS          0       0      4     48     0      1      0      0    16M
 ITSOLNX1     0       0   3063  37788     6     13      0      1  2048M
 ITSOLNX2     0       0  13410 177860     7     12      0      1  4096M
 ITSOLNX3     0       0  22045 291948     1     10      0      1  4096M
 ITSOLNX4     0       0 310353  4248k     0     68      0      1    20G
 MAINT        0       0    593   1015     0      0      0      0   256M
```

The rule of thumb for memory overcommitment is 2:1, which means that twice the main storage memory should be allocated to Linux guests. To achieve that number, and higher values, it is necessary to monitor the following usage measurements:

- Processor utilization
- Memory allocation and utilization
- Paging allocation and utilization

10.2.8 Minidisk cache guidelines

Minidisk cache is a configuration resource in z/VM that can provide better response for disk I/O requests. The amount of data that exists is much larger than the amount of data that is frequently used. This tends to be true for systems as well as individual virtual machines. The concept of caching builds off this behavior by keeping the frequently referenced data where it can be efficiently accessed. For minidisk cache, CP uses real or expanded storage or both as a cache for data from virtual I/O. By default, both real and expanded storage are used. Accessing electronic storage is much more efficient than accessing DASD. [1]

By using the minidisk cache, the number of paging I/O requests increases, which is not necessarily bad. If your system reduces the number of I/O requests per second for DASD to 200 seconds and increases the paging I/O requests per second to 30 seconds, overall reduction of I/O requests is 170 I/O requests per second. The minidisk cache configuration in this example reduced the number of processor cycles to I/O that is good.

[1] http://www.vm.ibm.com/perf/tips/prgmdcar.html

Example 10-7 on page 135 shows the STORAGE report panel where information about the number of minidisk cache reads per second, page reads, and page writes per second is available.

Initially, z/VM controls the configuration of the minidisk cache memory area in automatic setup. You might consider setting the limits of the minidisk cache utilization by using the following commands:

- **SET MDC STOR 0M 512M**
- **SET MDC XSTOR 0M 0M**

As minidisk cache benefits read requests, disable the minidisk option for devices that are designated for writes on, for example, the Linux guest minidisk that handles the /var mount point or Linux swap disks.

10.2.9 Paging subsystem

The paging/spooling activity area provides information about reads and writes. If the page reads are running in values larger than 14, look at the space that is allocated by z/VM and verify that there is adequate space to allocate block sets.

Table A-1 on page 154 in Appendix A, “Performance Toolkit reports” on page 153 shows a reference list of Performance Toolkit reports and commands. Example 10-11 shows the FCX109 report that displays, among other information, the load and performance of z/VM paging devices. In this report it is possible to identify if there are queues in the z/VM paging subsystem.

Queue Lngth tells you whether paging operations are queuing at the volume. Queue formation at paging volumes is a bad thing. If you see this value as nonzero, you either need to add volumes or to do DASD tuning. Non-zero queue lengths are the cause of elevated **MLOAD Resp Time.**

PAGE slot utilization tells you how full the paging system is altogether. You want this number to be 50% or less for versions of z/VM that are earlier than z/VM 6.3. If it is too large, add paging volumes or reduce the workload's memory requirement. For reliability, keep monitoring **PAGE slot utilization**.

Example 10-11 FCX109, CP Owned Device panel: DEVICE CPOWNED

```
FCX109      CPU 2827  SER CB8D7  Interval 20:16:51 - 20:17:51    Perf. Monitor

Page / SPOOL Allocation Summary
PAGE slots available      9014568          SPOOL slots available      1802880
PAGE slot utilization           0%         SPOOL slot utilization          80%
T-Disk space avail. (MB)        0          DUMP slots available             0
T-Disk space utilization      ...%         DUMP slot utilization           ..%

____ .            .                     >    .     .     .     .     .     .
< Device Descr. ->                      >  I/O        Serv MLOAD Block %Used
            Volume Area   Area      Used  >Inter Queue  Time  Resp  Page   for
Addr Devtyp Serial Type   Extent       %  >feres Lngth /Page  Time  Size Alloc
9A20 3390-9 SSI1P3 PAGE     1803060    0       0     0   0.0   0.0   ...   ...
9B21 3390-9 SSI1P4 PAGE     1802880    0       0     0   0.0   0.0   ...   ...
9B22 3390-9 SSI1P5 PAGE     1802880    0       0     0   0.0   0.0   ...   ...
9B23 3390-9 SSI1P6 PAGE     1802880    0       0     0   0.0   0.0   ...   ...
9E2A 3390-9 SSI1P2 PAGE     1802880    0       0     0   0.0   0.0   ...   ...
```

Monitor the LPAR memory to determine the correct LPAR size and the Linux guest memory size. To do this, use the PAGELOG report (FCX143), shown in Example 10-12. This report includes information about the total memory in the logical partition, information about the pages that are moving from central memory to expanded memory, central memory to DASD, and turn around.

The Paging Log (PAGELOG) report and the historical information provided by that should be monitored regularly, or at least before and after moving a Linux guest to a production environment. It provides systematic studies of paging behavior. Interval by interval, PAGELOG comments on page ins (PGINs), page outs (PGOUTs), migrations, reads, and writes. It also alerts you to single-page reads and writes.

The PAGELOG report contains two pages, side-by-side. To browse the pages, use the F10 and F11 keys on the keyboard.

Example 10-12 Paging log panel

```
FCX143      CPU 2827  SER DB8D7  Interval 00:00:10 - 11:21:10    Perf. Monitor

          <---------- Expanded Storage ----------> <-Real Stor-> <----------
                      Fast-              Est.  Page     DPA  Est.
 Interval  Paging  PGIN  Path PGOUT Total  Life  Migr Pagable  Page Reads Write
 End Time  Blocks    /s     %    /s    /s   sec    /s  Frames  Life    /s    /s
 >>Mean>>  523313    .0    .0    .0    .0  ....    .0 7781662  ....    .0    .0
 11:08:10  523313    .0    .0    .0    .0  ....    .0 7781460  ....    .0    .0
 11:09:10  523313    .0    .0    .0    .0  ....    .0 7781460  ....    .0    .0
 11:10:10  523313    .0    .0    .0    .0  ....    .0 7781460  ....    .0    .0
 11:11:10  523313    .0    .0    .0    .0  ....    .0 7781460  ....    .0    .0
 11:12:10  523313    .0    .0    .0    .0  ....    .0 7781460  ....    .0    .0
 11:13:10  523313    .0    .0    .0    .0  ....    .0 7781459  ....    .0    .0
 11:14:10  523313    .0    .0    .0    .0  ....    .0 7781460  ....    .0    .0
 11:15:10  523313    .0    .0    .0    .0  ....    .0 7781460  ....    .0    .0
 11:16:10  523313    .0    .0    .0    .0  ....    .0 7781460  ....    .0    .0
 11:17:10  523313    .0    .0    .0    .0  ....    .0 7781425  ....    .0    .0
 11:18:10  523313    .0    .0    .0    .0  ....    .0 7781308  467M    .0    .0
 11:19:10  523313    .0    .0    .0    .0  ....    .0 7781309  ....    .0    .0
 11:20:10  523313    .0    .0    .0    .0  ....    .0 7781309  ....    .0    .0
 11:21:10  523313    .0    .0    .0    .0  ....    .0 7781309  ....    .0    .0

FCX143      CPU 2827  SER DB8D7  Interval 00:00:10 - 11:30:10    Perf. Monitor

         ><-Real Stor-> <----------- Paging to DASD ------------> <Page Table>
         >    DPA  Est.                          <-Single Reads--> <Management>
 Interval >Pagable  Page Reads Write Total  Shrd Guest Systm Total Reads Writes
 End Time > Frames  Life    /s    /s    /s    /s    /s    /s    /s    /s     /s
 >>Mean>> >7781657  ....    .0    .0    .0  24.6    .0    .0    .0    .0     .0
 11:17:10  7781425  ....    .0    .0    .0  24.6    .0    .0    .0    .0     .0
 11:18:10  7781308  467M    .0    .0    .0  24.6    .0    .0    .0    .0     .0
 11:19:10  7781309  ....    .0    .0    .0  24.6    .0    .0    .0    .0     .0
 11:20:10  7781309  ....    .0    .0    .0  24.6    .0    .0    .0    .0     .0
 11:21:10  7781309  ....    .0    .0    .0  24.6    .0    .0    .0    .0     .0
 11:22:10  7781309  ....    .0    .0    .0  24.6    .0    .0    .0    .0     .0
 11:23:10  7781308  ....    .0    .0    .0  24.6    .0    .0    .0    .0     .0
 11:24:10  7781309  ....    .0    .0    .0  24.6    .0    .0    .0    .0     .0
 11:25:10  7781309  ....    .0    .0    .0  24.6    .0    .0    .0    .0     .0
 11:26:10  7781309  ....    .0    .0    .0  24.6    .0    .0    .0    .0     .0
```

```
11:27:10  7781309  ....    .0    .0    .0  24.6    .0    .0    .0    .0    .0
11:28:10  7781309  ....    .0    .0    .0  24.6    .0    .0    .0    .0    .0
11:29:10  7781308  ....    .0    .0    .0  24.6    .0    .0    .0    .0    .0
11:30:10  7781309  ....    .0    .0    .0  24.6    .0    .0    .0    .0    .0
```

10.2.10 Final memory considerations

Always understand your Linux guest server memory utilization and peak times that influence the health of your production environment. Following are some guidelines that you might want to follow in your production environment:

- Define the expanded storage memory with at least 25% of the total memory of the LPAR. You can go as high as 4 GB.
- Maintain DASD paging less than or equal to 50%.
- When planning DASD, use many smaller volumes instead of one or two large volumes.
- Use separate I/O channels for DASD, if possible. Allocate more than one path per page of DASD. The more page disks and paths to the disks that you have, the more work can be done concurrently.
- Paging space must allocate all cylinders from one disk and all page disks must have the same size. It works if you do it differently, but performance suffers.
- Reserve a few slots in the CP-owned list. If you need more pages in the future, you can add them without stopping the system.
- Paging to FCP SCSI (EDEVICES) might offer higher paging bandwidth, but with higher processor requirements.
- As a start during planning, figure the starting memory size for the logical LPAR memory to be the sum of all planned Linux guest memory definitions, plus the page space. The allocation of the memory must be around 75% of main storage memory and 25% of expanded storage memory. The page space following the 25% of the total memory of the LPAR can be up to 4 GB.

Important: These are all recommendations that are based on the authors' own experiences. Always monitor your memory and paging space to ensure an environment with high performance.

For more information, see "Understanding and Tuning z/VM Paging" at the following website:

http://www.vm.ibm.com/perf/tips/prgpage.html

10.3 IBM Tivoli OMEGAMON XE on z/VM and Linux

IBM Tivoli OMEGAMON XE on z/VM and Linux provides a wide range of information about the z/VM operating system, its resources, and workloads. Information on Linux instances running as z/VM guests and the Linux workloads reveal how these instances and workloads on Linux are performing and impacting z/VM and each other.

IBM Tivoli OMEGAMON XE on z/VM and Linux uses the data collection from the Performance Toolkit for VM (the Performance Toolkit is a prerequisite) and complements it with data collection by the IBM Tivoli Monitoring agent for Linux on System z. The Performance Toolkit is the foundation for gathering z/VM metrics and provides the base for z/VM data input to OMEGAMON XE on z/VM and Linux. The agent has been the basis for

monitoring Linux on all platforms. OMEGAMON XE on z/VM and Linux takes advantage of the Tivoli Enterprise Portal and allows for all of the Tivoli Enterprise Portal alerting, action, and integration capabilities to be used.

If the IBM Tivoli Monitoring or other OMEGAMON XE products are already implemented in an environment, IBM Tivoli OMEGAMON XE on z/VM and Linux can be integrated into the existing monitoring architecture smoothly.

Moreover, Business Analytics reports can be obtained by leveraging the Tivoli Data Warehouse in IBM Tivoli Monitoring architecture together with a business intelligence (BI) product, for example, IBM Cognos®. The required monitoring data can be customized into the Tivoli Data Warehouse from OMEGAMON XE on z/VM and Linux.

In general, the following benefits are realized by using Tivoli OMEGAMON XE:

- Allow monitoring of z/VM and Linux on System z without logging in to either.
- Ability to see system interdependencies and monitor bottlenecks and other problems to tune for performance and availability.
- Investigate performance across all of your mainframe and distributed systems–without the need for advanced scripting or coding skills.
- Extends management reach by alerting you at the first sign of trouble, so you can monitor your systems more proactively.
- Give your technical, management, and business teams' views of the data they need based on their job responsibilities.

For more details about IBM Tivoli Monitoring and OMEGAMON XE on z/VM and Linux, see the following website:

`http://www-01.ibm.com/software/tivoli/products/omegamon-xe-zvm-linux`

Figure 10-5 shows the z/VM monitoring architectural overview. z/VM Performance Toolkit is a prerequisite of Tivoli OMEGAMON XE on z/VM and Linux.

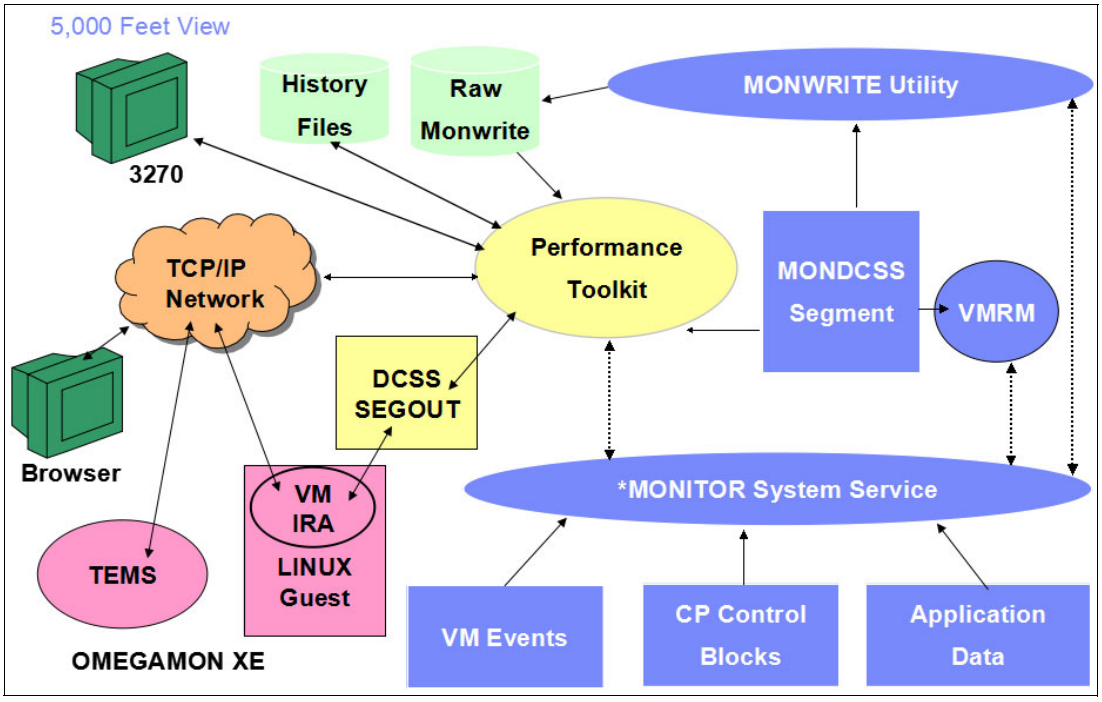


Figure 10-5 z/VM monitoring architectural overview

Monitoring data from inside Linux on System z can also be obtained via Tivoli OMEGAMON XE on z/VM and Linux. OMEGAMON can monitor the following information from Linux on System z:

- Workload statistics
- Disk utilization
- File usage
- Process resource usage
- Overall health of the Linux guest

You can use OMEGAMON XE to perform some management and automation tasks, for example:

- Monitor and set alerts on the system with predefined situations or custom situations
- Use policies to perform actions, schedule work, and automate manual tasks

Consider using the following two options for Linux on System z monitoring:

- For mission critical Linux on System z guests, install the monitoring agent on those Linux guests to get more detailed information. An installed IBM Tivoli agent has a very small overhead of approximately 0.01 - 0.03% of a processor for each guest.
- For Linux on System z guests with less critical workload, we suggest using Simple Network Management Protocol (SNMP) to monitor those guests. This is also called an *agentless method*.

10.4 Open source tools and Linux on System z commands

The **nmon** tool is very common on the distributed platform. It is an open source tool. It supports various platforms, including Linux on System z. But **nmon** is not a tool that can monitor all Linux on System z guests from a single point. For more information about **nmon**, see the following website:

http://nmon.sourceforge.net/pmwiki.php

Another useful tool is Nagios. It provides more functions than **nmon**. It also is an open source tool. For more information about Nagios, see the following website:

http://www.nagios.org

There are several useful commands to get real-time performance data from Linux on System z:

- **vmstat**: The **vmstat** command displays current statistics for processes, usage of real and virtual memory, paging, block I/O, and CPU.
- **top**: The **top** command displays process statistics and a list of the top CPU-using Linux processes.
- sysstat package: The **syssta** package is a widely used Linux standard tool.
- **iostat**: The **iostat** command provides CPU and input/output statistics for the devices and partitions.
- **netstat**: **netstat** is a powerful tool for checking your network configuration and activity.
- ziomon tool: The ziomon tool collects information and details about FCP configuration, I/O workload, and utilization of the FCP resources. Ziomon contains a set of commands, such as **ziomon**, **ziorep_config**, **ziorep_utilization**, **ziorep_traffic**.

The ziomon tool is available to the following Linux on System z distributions:

- s390-tools package starting with SLES10 SP3 and SLES11 SP1
- s390utils package starting with RHEL 5.4 and provided as s390utils-ziomon beginning with RHEL 6

For more information about Linux on System z monitoring commands, see the following website:

http://www.ibm.com/developerworks/linux/linux390/perf/tuning_resources.html

11

Accounting

One of the advantages of consolidating an enterprise's distributed systems onto IBM System z using Linux and z/VM is the ability to establish a precise accounting system. Information technology costs are always a key factor when considering server consolidation, so it is necessary to establish a charge-back method.

z/VM has an accounting system service to collect accounting information for guest machines. If the accounting is enabled, CP collects resource usage information about guests and stores that data in memory.

The System z Processor Resource/System Manager (PR/SM) hypervisor makes it possible to divide a physical System z computer into disjointed computing zones called *logical partitions* (LPARs). These partitions are equipped with logical physical units (PUs), which PR/SM dispatches on physical PUs to run the workloads of the partitions.

Recognizing this partitioning scheme, and realizing that PR/SM itself also consumes some CPU for its own ends, we can break down the consumption of System z physical CPU time into three very specific buckets:

- Cycles that are consumed by partitions' logical PUs, running their own instructions.
- Cycles that are consumed by the PR/SM hypervisor, running its own instructions, but running them in direct support of the deliberate action of some specific logical PU, and consequently accounting the consumed cycles to said logical PU as overhead that the logical PU caused or induced.
- Cycles that are consumed by the PR/SM hypervisor, running its own instructions, but doing work not directly caused by, and therefore not chargeable to, any given logical PU.

The PR/SM hypervisor keeps counters that measure these three kinds of CPU time. It accounts for the first two kinds of time on a per-logical-PU basis. It accounts for the third on a per-physical-PU basis.

© Copyright IBM Corp. 2013. All rights reserved.

This hypervisor can provide full information about how much time each Linux guest executes instructions on the Integrated Facility for Linux (IFL). The information that is provided by z/VM is not the same information that you have when you are using Linux tools such as **vmstat** or **Systat**. The most that Linux tools can provide is the information about how much of the percentage of the allocated virtual processor is used during a certain amount of time and that could be between 1 second or 1 hour, depending on the configuration. The z/VM accounting system can provide information about how long a specific guest uses the IFL processor with software and hardware instructions in milliseconds.

11.1 Do it simple; do it right

To charge-back a service, it is not only the processor utilization time from one server that counts. A lot more information must be included in a charge-back service. The first step is to determine the type of information that must be collected in order to provide an efficient report. Table 11-1 describes the recommended resources to count and the unit that can be used.

Table 11-1 Accounting information

Resource	Description	What to use	Unit reports
Processor	Charge calculation that is based on the utilization time of the processor per Linux guest.	z/VM accounting information data.	Total of utilization minutes per period of time.
Disk	Charge calculated based on the size of disk storage that is allocated per Linux guest.	Linux line commands such as **df** with a simple script to filter and store information.	Total space (in GB) attached per period.
Support tools and other resources	Charge that is calculated on a fixed value per Linux guest that shares IT resources (monitoring tools and software, IT personal tools, help-desk).	Fixed value that is based on the sum of the resources per period costs. For example, you can split the support and subscription cost for the monitoring software by the total number of virtual servers.	Fixed value per running server.

It is important to split and document all services or applications that are running on each Linux on System z guest, whether it is an engineering department, or the financial department. Do not set up middleware or services for two different cost centers on the same Linux guest. See Figure 11-1 on page 148 for an example of setting up multiple environments for the different cost centers.

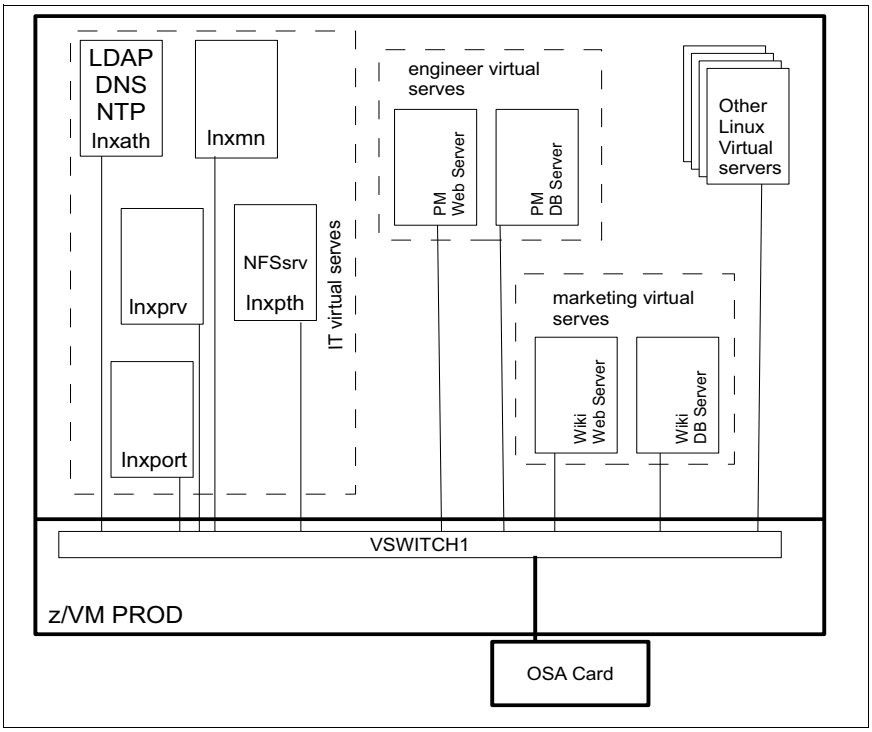


Figure 11-1 Multiple accounting environment LPAR

Consider an engineering cost center and a marketing cost center, as an example. Based on Figure 11-1, you would have four servers with the following configurations:

- Engineering web server
- One virtual processor
- 512 MB virtual RAM
- 256 virtual disk swap space
- 4 GB extended count key data (ECKD) or fixed-block architecture (FBA) disk for Linux operating system (OS)
- 10 GB additional storage area network (SAN) disks for file store
- Engineering database server
  - One virtual processor
  - 1 GB virtual RAM
  - 512 virtual disk swap space
  - 4 GB ECKD or FBA disk for Linux OS
  - 10 GB additional SAN disks for data store
- Marketing web server
  - One virtual processor
  - 2 GB virtual RAM
  - 512 MB virtual disk swap space
  - 512 MB direct access storage device (DASD) or FBA disk for Linux SWAP partition
  - 4 GB ECKD or FBA disk for Linux OS
  - 50 GB additional SAN disks for file store

- Marketing database server
  - One virtual processor
  - 1 GB virtual RAM
  - 512 virtual disk swap space
  - 4 GB ECKD or FBA disk for Linux OS
  - 10 GB additional SAN disks for data store

In this kind of environment, you would use more disk devices to store additional virtual servers. Therefore, instead of setting up an environment with a huge memory size that is not recommended for any virtualized environment, you would be able to set up several smaller servers and maintain control of their utilization.

The advantages are not only the accounting system but an easier deployment for the following types of functions:

- Servers capacity planning
- Application troubleshooting
- Servers performance troubleshooting
- Better control of resource utilization

And at the end of the month, it is then possible to create a report explaining to the engineering and marketing departments, in detail, which information technology (IT) resources that they are using.

You can use Example 11-1 as an example of the math calculation that might be used to calculate charge-back as a proposal for an internal accounting project. The report to each cost center must explain all information and details about the gathered data and how your IT team charges for their IT services and resources. If possible, include a graphic chart with the daily utilization report with a special mark indicating the peak load periods. This helps the non-IT person to understand the report better.

Example 11-1 Math calculation for charge-back

```
fv + ( dsk1 * sz1 ) + ( dsk2 * sz2 ) + (cpu * time) where ;

fv = fixe value per linux server based on a pre-defined value in a fictitious
currency;
dsk1 = value in a fictitious currency for one gigabytes allocated in a ECKD
device;
dsk2 = value in a fictitious currency for one gigabytes allocated in a SCSI
device;
sz1 = size of the additional allocated of ECKD device for linux guest in gigabytes
;
sz2 = size of the additional allocated of ECKD device for linux guest in gigabytes
;
cpu = value in a fictitious currency for one hour of IFL utilization ;
time = total of time of the IFL processor utilization ;
```

11.2 Configuring the z/VM accounting services

For more information about how to configure the z/VM accounting services, see the IBM Redpaper publication written by Erich Amrehn, Ronald Annuss, and Arwed Tschoeke: *Accounting and Monitoring for z/VM Linux guest machines*, REDP-3818. Although published in 2004, it still has relevant and useful information.

The first step in the z/VM accounting configuration is to verify if the DISKACNT service machine is started. The user ID for the virtual machine is `DISKACNT`. The user ID for the accounting virtual machine is defined as part of the SYSTEM_USERIDS statement in the system configuration file so that it is automatically logged on by the central processor (CP). It is also necessary to include the service machine called *ACCSRV* in the profile exec from AUTOLOG1 machine and also the **CP** command that guarantees that all machine data is collected.

ACCSRV is a service machine that is used to process the accounting records. ACCSRV has read-only access to the DISKACNT A-disk. ACCSRV is automatically started through AUTOLOG1 and operates in disconnected mode. The service machine periodically executes a REXX script via WAKEUP. This in turn executes the **CP ACNT ALL** command, which then processes account records to extract actual Linux guest resource usage values.

Example 11-2 Commands to be included at the PROFILE EXEC A file from the AUTOLOG1 user ID

```
PIPE CP XAUTOTLOG ACCSRV
PIPE CP RECORDING ACCOUNT ON
```

If you are using a single system image (SSI) configuration and running in multiple LPARs, it is necessary to set up the ACCSRV as an IDENTITY that is not so different from the original information that is described in *Accounting and Monitoring for z/VM Linux guest machines*, REDP-3818. Example 11-3 shows the directory entry for the ACCSRV user ID.

Example 11-3 Directory entry for the ACCSRV user ID

```
IDENTITY ACCSRV NOLOG 32M 32M AG                                    04161804
   INCLUDE IBMDFLT                                                   04161804
   BUILD ON ITSOSSI1 USING SUBCONFIG ACCACT-1                        04161804
   BUILD ON ITSOSSI2 USING SUBCONFIG ACCACT-2                        04161804
   BUILD ON ITSOSSI3 USING SUBCONFIG ACCACT-3                        04161804
   BUILD ON ITSOSSI4 USING SUBCONFIG ACCACT-4                        04161804
   IPL CMS                                                           04161804
   MACHINE XA                                                        04161804
   XAUTOLOG AUTOLOG1 OP1 MAINT                                       04161804
   LINK DISKACNT 0191 0192 RR                                        04161804
   LINK MAINT 0493 0493 RR                                           04161804
*DVHOPT LNKO LOG1 RCM1 SMSO NPW1 LNGAMENG PWC20130412 CRCeĩ          04190002
```

In Example 11-4, we show the user directory entry for user, ACCACT-1. All subconfiguration entries that are related to the ACCSRV user ID are linked here. Each MDisk entry has a start cylinder and volume.

Example 11-4 ACCACT-1 ID entry

```
SUBCONFIG ACCACT-1                                                    04121526
   MDISK 0191 3390 13867 0050 SSIUR1 MR                               04121526
*DVHOPT LNKO LOG1 RCM1 SMSO NPW1 LNGAMENG PWC20130412 CRCóo           04190002
```

In z/VM SSI systems, we use the IDENTITY type user ID because the ACCSRV user must run on every logical partition. To understand more about service machines in the SSI environment, see *An Introduction to z/VM Single System Image (SSI) and Live Guest Relocation (LGR)*, SG24-8006, in *Appendix B. New and Update Commands.*

When collecting and displaying data, have one central Linux server that downloads all daily reports for each ACCSRV service guest running on every logical partition and consolidate all information onto a single Linux report server, as shown in Figure 11-2.

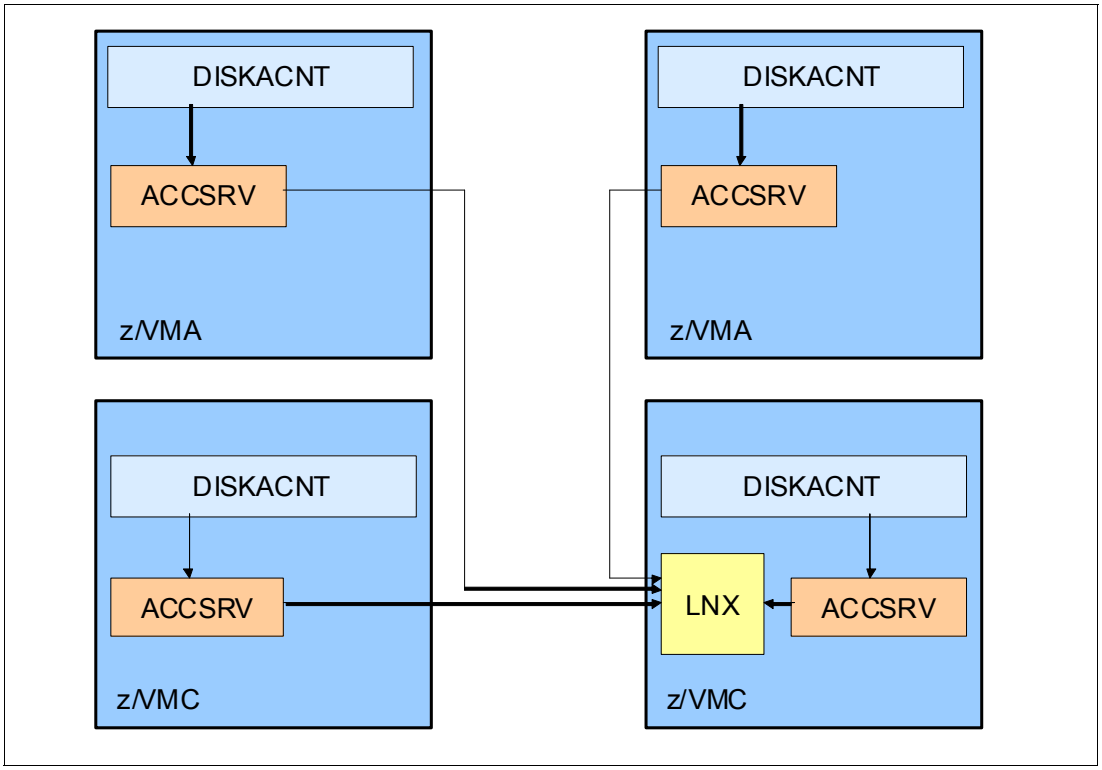


Figure 11-2 Multiple logical partition accounting

For more information about this topic, see *z/VM: CP Planning and Administration*, SC24-6178-03, at the following site:

http://publib.boulder.ibm.com/infocenter/zvm/v6r2/index.jsp?topic=%2Fcom.ibm.zvm.v620.hcpa5%2FhcsgOc11221.htm

A

Performance Toolkit reports

This appendix provides some Performance Toolkit reference commands and reports.

© Copyright IBM Corp. 2013. All rights reserved.

A.1 Performance Toolkit reference commands and reports

Table A-1 Performance Toolkit reference reports

REPORT NAME	REPORT CODE	COMMAND
PROCESSOR		
CPU Load and Transactions	FCX100	CPU
LPAR Load	FCX126	LPAR
Processor Log	FCX144	PROCLOG
LPAR Load Log	FCX202	LPARLOG
User Wait States	FCX114	USTAT
System Summary	FCX225	SYMSUMLG
SRM		
System Scheduler Settings	FCX154	SYSSET
USER		
User Configuration	FCX226	UCONF
STORAGE		
Auxiliary Storage Log	FCX146	AUXLOG
CP Owned Device	FCX109	DEVICE CPOWNED
User Page Data	FCX113	UPAGE
Shared Data Spaces	FCX134	DSPACESH
User Page Data	FCX113	UPAGE
SXS Available Page Queues Mgnt	FCX261	SXSAVAIL
Mini Disk Storage	FCX178	MDCSTOR
Storage Utilization	FCX103	STORAGE
Available List Log	FCX254	AVAILLOG
I/O		
General I/O Device	FCX108	DEVICE
DASD Performance Log	FCX131	DEVCONF
FICON Channel Load	FCX215	FCHANNEL
General I/O Device Data Log	FCX168	DEVLOG
I/O Processor Log	FCX232	IOPROCLG

B

Migration checklists

This appendix provides sample checklists for migrating from a development environment to a production environment. For more checklists and examples, see the IBM Redbooks publication, *Practical Migration to Linux on System z*, SG24-7727.

© Copyright IBM Corp. 2013. All rights reserved.

B.1 z/VM checklist

In this section, we provide a checklist of some of the things you should consider when migrating your z/VM into production. For more inclusive information and blank z/VM resources worksheets, see *The Virtualization Cookbook for z/VM 6.3, RHEL 6.4*, a*nd SLES 11 SP3*, SG24-8147.

B.1.1 Architecture

- ☐ Decide if you will use an SSI or non-SSI environment. Determine your need for live guest relocation (LGR). Do you require nondisruptive z/VM maintenance?
- ☐ If you require an SSI environment, determine the number of System z servers you will have in your production environment.
- ☐ Determine the number of SSI members you will require.
- ☐ If you do not require an SSI solution, determine if this will be a SCSI solution.
- ☐ Determine whether you will use ECKD, SCSI, or both attached devices. See section 4.2, “ECKD versus SCSI” on page 59.
- ☐ Be aware that a dedicated HyperPAV base volume or alias can be assigned only to one guest. I/O operations that are initiated through a HyperPAV alias can be directed only to base volumes that are ATTACHED or LINKED to the issuing virtual machine. There are many considerations, for example, before you can use PAV on your Linux guest. The PAV feature must be enabled on your storage system. You will need privilege class B authorization. For information about configuring base and alias volumes for PAV or HyperPAV, see your storage system documentation.
- ☐ Determine your accounting needs; who are your cost centers? Will there be a charge-back? See Chapter 11, “Accounting” on page 145 for more information.

B.1.2 Hardware and storage

- ☐ Examine your LPAR needs. Note the number of available Integrated Facilities for Linux (IFLs), weight, caps, total real memory, and memory allocation between cstore and xstore (especially if you will be hosting a web application server, such as IBM WebSphere Application Server). A good reference presentation for this topic can be found at the following website:

  `https://ibm.biz/BdxAhZ`
- ☐ Ensure that your virtual to real (V/R) ratio is not more than 1.5:1. See section 3.2.1, “z/VM storage” on page 35 for more details.
- ☐ If you are using an SSI environment, ensure that all SSI members have enough storage to hold relocated members.
- ☐ Define enough reserved storage in the LPAR image profile configuration in preparation of increasing the numbers of Linux virtual servers or the amount of storage that those servers are using.
- ☐ Calculate paging subsystem needs as described in section 3.2.2, “Paging subsystem definitions” on page 37.
- ☐ Calculate virtual storage (memory) needs as described in section“VIR2REAL tool” on page 38.
- ☐ Determine the type of Fibre Channel host adapters and cables you will need and use. See section 3.3.5, “Considerations of choosing FICON or FCP” on page 50 for selection criteria.

- FCP is a transport protocol that predominantly transports SCSI commands over Fibre Channel networks.
- If the FICON environment, each unit (UA) is represented on the host side as a subchannel (SCH) and managed by a device number (DEVNO), each unit is associated to a control unit (CU). Define these in the IOCDS.
- See section 3.3.4, “Performance assessment” on page 46 for considerations regarding performance.

☐ Consider your channel configuration and management. See section 3.3.2, “Channel configuration and management consideration” on page 43 for more information.

☐ Consider how you will manage shared channel resources, as outlined in section 3.3.3, “Channel sharing” on page 44.

☐ Consider using zHPF. See section 3.3.4, “Performance assessment” on page 46 for more information.

B.1.3 Security

In this section, we provide a checklist to secure z/VM. For further information, refer to *Security on z/VM*, SG24-7471 for details about z/VM security.

- Secure logical access to z/VM:
  - Using an external security manager (ESM), such as RACF, stores system user passwords in a backend database and provides accountability as well as manageability.
  - Use appropriate z/VM privilege classes for your Linux on System z guests. Start with the general user privilege class and allow access only to CP control functions over each guest’s own virtual machines and resources.
  - Use a secured channel such as SSL for all Telnet communications between a 3270 terminal and z/VM.
- Secure physical access to z/VM:
  - Secure minidisks: When an ESM is not in use, you must define passwords on the system’s directory for the disks that you want to share. Although this sounds like a fairly secure approach, a good practice is to use an ESM to make your configuration more resilient and less error-prone.
  - Reduce intrusion points: Virtually connect devices among guests within the same system. One possibility is to have a disk or a set of disks (an LVM group for example) shared among multiple servers without the need for network services of any kind. Your data will not flow across the network and consequently cannot be sniffed, reinforcing the overall security of your environment and reducing the number of intrusion points.
  - Protect data with encrypted file systems: The System z platform leverages advantages by using cryptographic hardware and cryptographic functions that are built in to the central processor (CP). By using Central Processor Assist for Cryptographic Function (CPACF), it is possible to offload cryptographic cipher calculations from the central processor, thus considerably reducing the processor cycles of such operations compared to the cost of having them done through software emulation.

B.2 Linux on System z

In this section, we provide a checklist of all the considerations you should take into account when migrating your Linux on System z guests into production. For more inclusive information and blank Linux on System z resources worksheets, see *The Virtualization Cookbook for z/VM 6.3, RHEL 6.4, and SLES 11 SP3*, SG24-8147.

B.2.1 Architecture

- ☐ Determine if you will run Linux on System z native (on the LPAR directly) or as a z/VM guest.
- ☐ Determine which Linux on System z distribution is appropriate for you. For a list of IBM tested and supported platforms versus distribution, see the following website:

  `http://www-03.ibm.com/systems/z/os/linux/resources/testedplatforms.html`

B.2.2 Hardware, memory, and storage

This template lists various hardware resources that need to be considered during a migration project. In the checklist template, the source environment's hardware resources are examined and we need to acquire similar or more advanced technology that is available for Linux on System z. Example B-1 on page 159 shows a sample hardware planning checklist.

Example B-1 Hardware planning checklist

Source		Destination
DEVICE	**Value**	**DEVICE**
Number Of CPUs		***Number Of Virtual CPUs***
Server Memory		***Server Memory***
Real Memory		Virtual Memory
SWAP Memory		**SWAP Memory**
		V-DISK
		M-DISK
		ECKD model 3 DASD
Network Connections		***Network Connections :*** [1]
Connection Description		**Connection Description**
Connection Type		Connection Type
IP Address		IP Address
		Device Connection Name/Address
Connection Description		**Connection Description**
Connection Type		Connection Type
IP Address		IP Address
		Device Connection Name/Address
Connection Description		**Connection Description**
Connection Type		Connection Type
IP Address		IP Address
		Device Connection Name/Address
OS File System		***OS File System***
		/ (root file system)
		mount point
		size
		/usr
		mount point
		size
		/var
		mount point
		size
		/tmp
		mount point
		size

1. Device connection types: QETH, HiperSockets, Direct OSA-Express2 connection

B.2.3 Security

In this section, we provide a brief overview of a checklist that can be used for Linux on System z security. Refer to *Security for Linux on System z*, SG24-7728 for details about Linux on System z security.

- Secure the logical access to the Linux servers:
  - Access control: Use of one or both of the following methods to control access:
    - Mandatory access control (MAC)
    - Discretionary access control (DAC)

Although DAC enables owners of objects to grant access to other users, MAC has the policy as the center of all decisions. With the advent of SELinux and AppArmor, MAC for Linux has become more common.

– Authentication - Access control defines who can access what and how this access can be made, but authentication involves determining whether someone really is who he or she claims to be. The users attempting to access a system or a resource must first give enough proof of their identity.

  Linux offers an extremely flexible interface to plug and unplug authentication mechanisms to meet the various security requirements your organization might have. The Pluggable Authentication Modules (PAM) can be used as an extremely powerful instrument to reinforce the compliance to your security policies by only authenticating users who meet specific characteristics.

B.3 Infrastructure checklist

In this section, we provide a high-level checklist for securing your infrastructure.

- Physical infrastructure

  Infrastructure security begins with the physical environment. Buildings, rooms, infrastructure services, servers, access points, and operational equipment all play a part in a secure environment. Ensure that the integrity of physical locations where the systems are installed and from which they can be accessed has been secured.

- Minimize installations

  Ensuring the bare-minimum gets installed not only simplifies administration but also avoids the potential risks. Keeping installations at minimal dependency needs should be a constant, that is, if services are eventually disabled, it is a good practice to also remove those services along with their dependencies, if they are not required by other packages.

- Protect the Hardware Management Console (HMC)

  Because this is a powerful entry point into controlling the System z environment, never share user IDs or passwords for the HMC.

- Protect the configuration

  z/VM supports the hardware dynamic I/O configuration facility. This facility allows you to dynamically add, delete, or modify the I/O configuration of the processor without requiring a power-on reset of the processor or IPL of z/VM. You would use these commands on the HMC. The authority to issue such commands should be restricted to a single LPAR on each server and access to the facility strongly protected.

- Multizone configurations

  In a multitier, or multizone, configuration of individual server images, or groups of server images, are isolated from one another using firewall technologies. Firewalls implement a set of authorization rules to restrict user access or avoid certain data flows between zones. Ensure that you have a good understanding of where your firewalls will be hosted. For more information about where to place firewalls, the types of firewalls available for Linux on System z, and some Linux firewall tools, see the IBM Redbooks publication, *Security for Linux on System z*, SG24-7728.

- Protect your direct access storage devices (DASD), also known as disks:
  - Linux and z/VM use a different security model than z/OS, ensure that the DASD used for Linux data is not part of the same shared-disk LPAR configuration as your z/OS disks. If DASD that is formatted as Compatible Disk Layout (CDL) is brought online to z/OS, it can still be recognized.
  - Use an ESM such as RACF to protect by data set name.
- Protect your Fibre Channel Protocol (FCP) disks

  The security configuration of Fibre Channel SANs is usually done differently than mainframe disk environments. Techniques such as zoning and LUN masking are essential for controlling access to SAN volumes. Again, we direct the reader to the IBM Redbooks publication, *Security for Linux on System z*, SG24-7728 for more information about how to secure this environment.
- Protect z/VM minidisks

  Minidisks can be accessed by only the users that they are intended for. If minidisk access is too lax, information can easily be leaked between virtual machines. Using an ESM provides greater granularity to access authorities and more extensive options for controlling the auditing of access.

B.4 Network checklist

In this section, we provide a checklist for ensuring that your network provides optimum availability for your production environment. For more detailed information about networks, see *Advanced Networking Concepts Applied Using Linux on IBM System z*, SG24-7995.

B.4.1 Architecture

We provide a checklist of network resources that you should consider for a production environment. This list does not include the very basics of networking resources, such as a TCP/IP address for z/VM, each of your Linux guests, and all of the associated TCP/IP address information such as a DNS host name or domain. For more information about the basic networking resources, refer to *The Virtualization Cookbook for z/VM 6.3, RHEL 6.4 and SLES 11 SP3*, SG24-8147.

☐ Determine your needs for VLAN trunking

VLAN trunking allows multiple VLANs to coexist on a single network connection. VLAN tagging is the most common and preferred method of doing this. The two most common methods of doing this are Cisco's proprietary inter-switch link (ISL) and the IEEE 802.1Q specification.

☐ Determine your needs for link aggregation

Link aggregation is a computer networking term that describes the various methods of bundling multiple network connections in parallel to increase bandwidth throughput and provide redundancy.

☐ Consider your need for VSWITCHs

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.

- ☐ Determine your need for VNICs

  A virtual network interface controller (VNIC) is a pseudo-network interface that is created within a system or a virtual server. The benefits of using a VNIC include providing a secure network environment that allows you to share and access resources on your network without the requirement of configuring and maintaining any physical components.

- ☐ Set the correct Ethernet autonegotiation

  Ethernet autonegotiation allows devices to automatically exchange, over a link, information about speed, duplex, and flow control abilities. As a rule, switchports for servers that are using FastEthernet should have autonegotiation disabled and switchports for servers that are using Gigabit Ethernet (both 1 and 10 Gbps) should have autonegotiation turned on.

- ☐ Determine the correct MTU

  Maximum transmission unit (MTU) refers to the size of the largest packet that a network protocol can transmit without fragmentation. A larger MTU brings greater efficiency because each packet carries more user data as compared to a packet with a smaller MTU while the IP headers remain fixed. The resulting higher efficiency improves throughput for bulk transfer protocols such as the file transfer protocol (FTP) and Internet Small Computer System Interface (iSCSI).

- ☐ Consider your need for load balancing

  Network traffic can be forwarded through multiple paths to achieve load balancing, load sharing, and redundancy. One drawback of load sharing is that when a failure happens on one of the links, the remaining links might not have enough capacity to support the wanted throughput unless the design has provided for it. Thus, the network designer must ensure that sufficient bandwidth is available to meet user requirements during a worst case scenario.

B.4.2 Networking

In this section, we outline a sample checklist to determine your networking needs for production. For more information, see Chapter 5, “Network planning considerations” on page 67.

- ☐ Determine your internal communication needs:
  - ☐ HiperSockets for LPAR-LPAR communications
  - ☐ z/VM guest LANs
  - ☐ z/VM VSWITCHs
- ☐ Determine your external communication needs:
  - ☐ OSA
  - ☐ z/VM VSWITCHs

B.4.3 Security

z/VM offers several possible network configurations, such as IBM HiperSockets adapters, and guest LAN and virtual switches. Using Linux on System z, guests with correct connectivity can dramatically reduce physical intrusion points (the configuration of these resources is not detailed within this document; for specific configuration details and system requirements, see z/VM V5R4.0 Connectivity, SC24-6080-07). Information about how to use z/VM native functions to protect guest LANs are in “Protecting z/VM guest LANs” in the IBM Redbooks publication, *Security for Linux on System z*, SG24-7728.

- Secure the virtual switches

  The z/VM default configuration automatically prevents users from coupling to a VSWITCH if not explicitly allowed. To create a layer-2 virtual switch, use the **DEFINE VSWITCH** command, then use the **SET VSWITCH** command to grant specific guests and users privileges to couple to the VSWITCH. If you are using an ESM such as RACF, see your documentation about how to grant this access.

- Secure VLANs

  The use of VLANs increases traffic throughput and reduces overhead by allowing the network to be organized by traffic patterns and not based on the physical locations of the servers. Because z/VM virtual networking capabilities are compatible with the standard specifications for virtual LAN tagging, the Linux virtual servers coupled to a VSWITCH can belong to VLANs that extend beyond z/VM's virtual network. Using a VLAN-aware VSWITCH is a good way to reduce the exposure for other access and authentication controls.

- Use VSWITCH port isolation

  A virtual switch and an OSA-Express card port can be shared by multiple TCP/IP stacks. This means that two servers that are hosted on the same hardware box can reach each other and the packets would be directly routed to the sharing TCP/IP stack without transversing the external network.

  A feature available on z/VM completely isolates local traffic by preventing one virtual guest from being able to reach others without going outside of the OSA-Express port. This isolation feature can help to completely isolate servers that are on exposed network zones and still maintain a uniform network addressing schema.

  To turn on the isolation mode on a virtual switch, run the following command:

  ```
  SET VSWITCH VSWITCH3 ISOLATION ON
  ```

- Consider use of VSWITCH PROMISCUOUS mode

  Promiscuous mode allows Linux users to take advantage of native network capture and troubleshooting tools when necessary. A recommended practice is not to allow any guests to couple to a virtual device in promiscuous mode on a permanent basis, unless the device is a monitoring station where other security mechanisms are in place to provide security against unauthorized access.

- Switch off backchannel communication

B.5 Product checklist

In the software checklist template (see Table B-1 on page 164), we list all the products and tools that are used in the source operating environment and then chart out whether the same or similar products and tools are available on the target Linux on System z operating environment.

Table B-1 Product and tools checklist

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

B.5.1 Application Implementation checklist

The Application Implementation checklist takes you one step further into the product checklist, where each product or tool is further drilled down to their features level. There are scenarios where the same product would not offer the same features on all the platforms. These details would be noted in this checklist (see Table B-2 on page 165).

Table B-2 Application Implementation checklist

Application Checklist		
Java Application Name : Database Connectivity :		
	Sun- Solaris	**Linux on System z**
JVM Version		
Compilation Options		
JIT / Optimization parameters		
Native dependencies		
Third party jar's dependencies		
Custom Application Name : Language Used : Database Connectivity :		
	SUN-Solaris	**Linux on System z**
Application Arch Model	32-bit	64 bit, 31 bit
Compiler Used		gcc
OS Version		SLES 11, RHEL 6
Compiler Version		
	Compiler Performance Options	
Compilation		
Linking		
Shared Library		
System Libraries Used		
For Debug Build		
	Compiler Build Options	
Compilation		
Linking		
Shared Object Creation		

As in Table B-2, each product or tool listed in the product checklist must be analyzed. All the parameters, dependencies, and optimization options have to be taken into account in the source operating environment and then the planning team needs to be assessed whether the same kind of features or build options are available in the target operating environment.

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

- Obtain similar features by linking other product or tools in the target operating environment.
- Make note of the other parameters that are available in the same tool of the target operating environment, which can be combined to give the same characteristics as in the source environment.
- If the products or product options are fully incompatible or not available, replacing the part of the application stack would also be a good approach, which would minimize the effort involved in migration. But care has to be taken so that all the features and parameters that are offered by the product in the source environment are also available in the assessing product for the target environment.
- Most of the times optimization features or performance options for a product would be only available for that specific platform. In this case, the optimization and performance options need to change to offer the same characteristics in the target environment.

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

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

C

Sample procedure

This appendix provides a sample procedure to move from a development environment to a production environment.

© Copyright IBM Corp. 2013. All rights reserved.

C.1 Pre-production steps

- ☐ Open a change request or a service request to document the environment state change
- ☐ Check if the server is updated

  For RHEL:

  ```
  # yum check-update
  ```

  For SUSE:

  ```
  # zypper lu
  ```

- ☐ Verify the umask

  ```
  # touch /etc/profile.local;echo "umask 077" >> /etc/profile.local;chmod 644
  /etc/profile.local
  ```

- ☐ Set “root” ID to expire periodically (30, 60, 90 days)

  ```
  # chage -m 1 -M 90 root
  ```

- ☐ Verify if “support” ID password has been locked and all authorized keys are in place

  ```
  # passwd -l support;chage -m 1 -M 90 support
  ```

  ```
  # cat /home/support/.ssh/autorized_keys
  ```

- ☐ Check if regular ids have been set up to expire periodically (30, 60, 90 days)

  ```
  # for i in `ls /home/`; do chage -l $i;done
  ```

- ☐ Privilege revalidation

  ```
  # cat /etc/sudoers
  ```

  If there are any privileged IDs or groups, send them an email confirming that the privilege is still required

- ☐ Check if all file system usages are below 65%

  ```
  # df -hP
  ```

  If any is higher than 65%, increase the file system before putting the system into production

- ☐ Verify if there is available space on volume groups (VGs)

  ```
  #vgs
  ```

  If not, extend the volume groups (VGs)

- ☐ Check if the server is registered on any monitoring tools that you are using (Nagios, ITM)
- ☐ Check if the server is registered on any asset management tool being used
- ☐ Ensure that backup is properly set (IBM Tivoli Storage Manager, Enterprise Content Manager)

  ```
  # dsmc q sched
  ```

- ☐ Enable and run **seccheck** if using SLES (**seccheck** is a security checker and a host security analyzer with three different levels of scans. When **seccheck** is installed, it automatically adds a crontab, /etc/cron.d/seccheck, to run daily, weekly and monthly security checks.)

  # zypper/yum install seccheck

  # cat /etc/sysconfig/seccheck|grep START_SECCHK

  If START_SECCHK="no"

  # sed -i 's/START_SECCHK="no"/START_SECCHK="yes"/g' /etc/sysconfig/seccheck

  Set support mail ID on **seccheck** user

- ☐ Scan system for vulnerabilities (ports and rootkit)

  Run:

  # rkhunter -c

  # nmap -v -sT 9.12.4.0/24

- ☐ Check aliases

  #cat /etc/aliases|egrep “(^root|^postmaster)”

  Support team email account is set as root

- ☐ Check if there is standby memory set for the guest

  # lsmem

  If there is no stand-by memory:

  On VM

  dirm for linux_guest get

  Add the define storage statement accordingly with the system requirements

  As an example, use the following command:

  COMMAND DEFINE STORAGE 4G STANDBY 4G

  dirm for linux_gest replace

- ☐ Verify if server is registered on DNS

C.2 Post-production steps

- ☐ Complete and submit the checklist
- ☐ Notify all teams that are related to the server
- ☐ Update asset information
- ☐ After the checklist is approved, close the change request or service request

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
- *Using z/VM v 6.2 Single System Image (SSI) and Live Guest Relocation (LGR),* SG24-8039
- *Advanced Networking Concepts Applied Using Linux on System z: Overview of Virtualization and Networking,* TIPS-0982
- *HiperSockets Implementation Guide,* SG24-6816
- *OSA-Express Implementation Guide,* SG24-5948

You can search for, view, download, or order these documents and other Redbooks, Redpapers, Web Docs, drafts, and additional materials, at the following website:

ibm.com/redbooks

Other publications

These publications are also relevant as further information sources:

- *Linux on System z - Device Drivers, Features, and Commands,* SC33-8411
- *z/VM Connectivity, version 6 release 2,* SC24-6174
- *z/VM Performance, version 6, release 2,* SC24-6208

Online resources

These websites are also relevant as further information sources:

- IBM VM Download Packages

  http://www.vm.ibm.com/download/packages
- IBM Advanced Technical Skills - Description and download of VIR2REAL EXEC

  http://www.vm.ibm.com/download/packages/descript.cgi?VIR2REAL
- z/VM V6R2.0 Information Center

  http://publib.boulder.ibm.com/infocenter/zvm/v6r2/index.jsp
- Dynamic Memory Upgrade

  http://www.vm.ibm.com/perf/reports/zvm/html/540dmu.html

© Copyright IBM Corp. 2013. All rights reserved.

- Understanding and Tuning z/VM Paging

  http://www.vm.ibm.com/perf/tips/prgpage.html
- z/VM System Limits

  http://www.vm.ibm.com/devpages/bitner/presentations/vmlimits.pdf
- Networking with Linux on System z

  http://www.vm.ibm.com/education/lvc/LVC1109C.pdf

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

IBM Redbooks

Set up Linux on IBM System z for Production

(0.2”spine)
0.17”<->0.473”
90<->249 pages

Set up Linux on IBM System z for Production

Redbooks®

Considerations to move from test and development into production

Capacity planning and performance

Sample checklists

This IBM Redbooks publication shows the power of IBM System z virtualization and flexibility in sharing resources in a flexible production environment. In this book, we outline the planning and setup of Linux on System z to move from a development or test environment into production. As an example, we use one logical partition (LPAR) with shared CPUs with memory for a production environment and another LPAR that shares some CPUs, but also has a dedicated one for production. Running in IBM z/VM mode allows for virtualization of servers and based on z/VM shares, can prioritize and control their resources.

The size of the LPAR or z/VM resources depends on the workload and the applications that run that workload. We examine a typical web server environment, Java applications, and describe it by using a database management system, such as IBM DB2.

Network decisions are examined with regards to VSWITCH, shared Open Systems Adapter (OSA), IBM HiperSockets and the HiperPAV, or FCP/SCSI attachment used with a storage area network (SAN) Volume Controller along with performance and throughput expectations.

The intended audience for this IBM Redbooks publication is IT architects who are responsible for planning production environments and IT specialists who are responsible for implementation of production environments.

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

SG24-8137-00 ISBN 0738438871