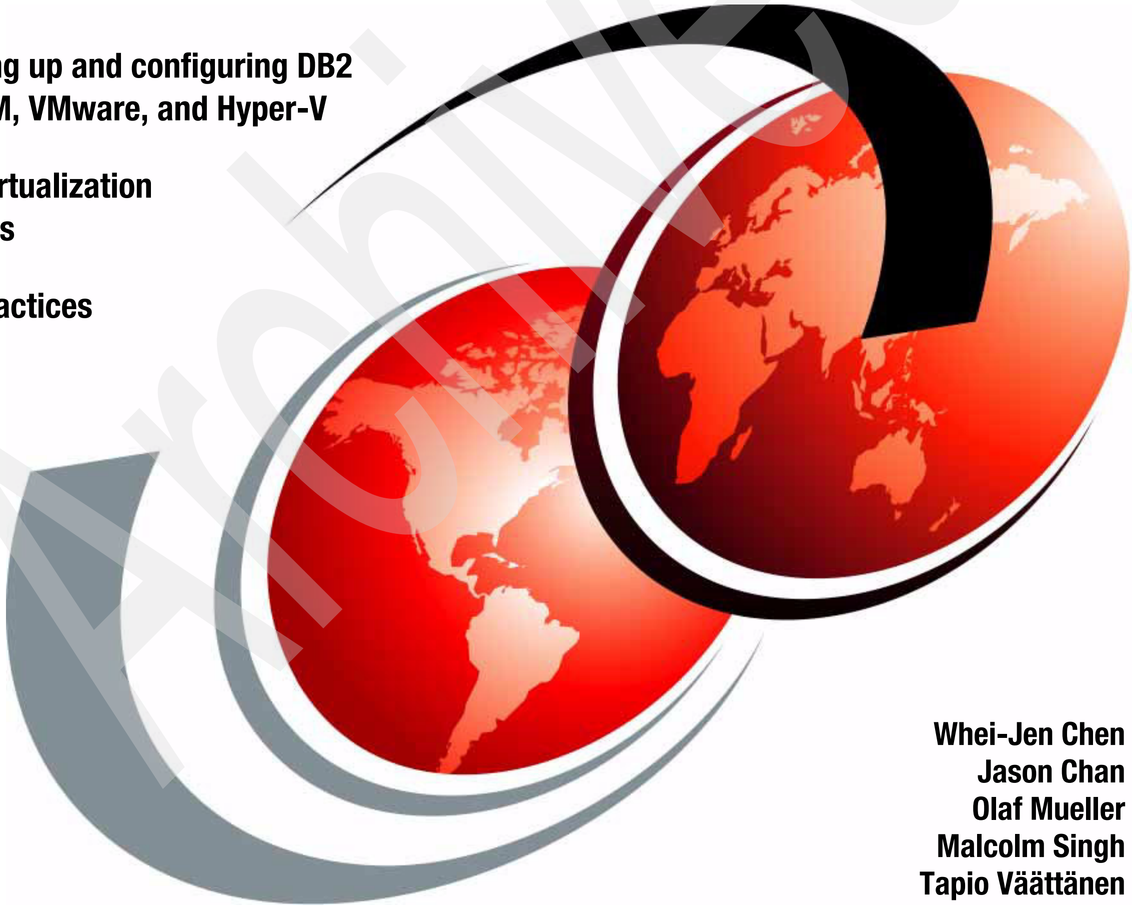
IBM

# DB2 Virtualization



- Learn setting up and configuring DB2 on PowerVM, VMware, and Hyper-V
- Leverage virtualization technologies
- See best practices



Whei-Jen Chen
Jason Chan
Olaf Mueller
Malcolm Singh
Tapio Väättänen

Redbooks

ibm.com/redbooks

International Technical Support Organization

# DB2 Virtualization

September 2009



SG24-7805-00

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



**First Edition (September 2009)**

This edition applies to DB2 for Linux, UNIX, and Windows Version 9.1 or later, PowerVM, POWER5, POWER6, VMware Virtual Infrastructure 3 or later, vSphere 4 or later, and Microsoft Windows Server 2008 SP2 with Hyper-V RTM (Update KB950050).

# Contents



# Notices

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

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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



# Trademarks

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

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

| | | |
|---|---|---|
| AIX® | Power Systems™ | Redbooks® |
| DB2® | POWER5™ | Redbooks (logo) ¬Æ |
| IBM® | POWER6® | System z® |
| Optim™ | PowerVM™ | z/VM® |
| POWER Hypervisor™ | POWER® | |

The following terms are trademarks of other companies:

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

AMD, AMD-V, the AMD Arrow logo, and combinations thereof, are trademarks of Advanced Micro Devices, Inc.

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

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

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

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

Hyper-V, Microsoft, MS, SQL Server, Windows Server, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

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

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

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

# Preface

Server virtualization technologies are becoming more popular to help efficiently utilize resources by consolidating servers. IBM®, the first company that developed and made available the virtual technology in 1966, offers advanced, powerful, reliable, and cost-saving virtualization technologies in various hardware and software products including DB2® for Linux, UNIX, and Windows. This IBM Redbooks® publication describes using IBM DB2 9 with server virtualization.

We start with a general overview of virtualization and describe specific server virtualization technologies to highlight how the server virtualization technologies have been implemented. With this introduction anyone new to virtualization will have a better understanding of server virtualization and the industry server virtualization technologies available in the market.

Following the virtualization concept, we describe in detail the setup, configuration, and managing of DB2 with three leading server virtualization technologies:

- IBM Power Systems™ with PowerVM™
- VMware
- Hyper-V

We discuss the virtual machine setup with DB2 in mind to help IT support understand the effective ways of setting up a virtual environment specific for DB2. We explain the architecture and components of these three server virtualization technologies to allow DBAs to understand how a database environment using DB2 can benefit from using the server virtualization technologies.

In addition, we discuss the DB2 features and functions that can take advantage of using server virtualization. These features are put into practice when describing how to set up DB2 with the three virtualization technologies discussed in this book. This book also includes a list of best practices from the various tests performed while using these virtualization technologies. These best practices can be used as a guideline or a reference when setting up DB2 using these virtualization technologies.



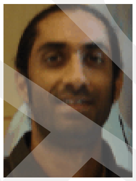
## The team who wrote this book

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

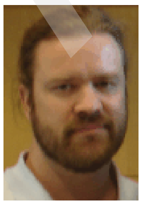
**Whei-Jen Chen** is a Project Leader at the International Technical Support Organization, San Jose Center. She has extensive experience in application development, database design and modeling, and DB2 system administration. Whei-Jen is an IBM Certified Solutions Expert in Database Administration and Application Development, as well as an IBM Certified IT Specialist.

**Jason Chan** is the Linux and Virtualization Lead of the Data Management Emerging Partnerships and Technologies team. He joined IBM as a full-time employee in 2003 and has specialized in various projects dealing with Linux and virtualization with DB2 and other data management products. He frequently engages various Linux, virtualization, and hardware partners to bring to the forefront new technologies for use with DB2, as well as participating in the team's mandate of enabling IBM Business Partners on DB2 through the many DB2 bootcamps around the world. Jason is based at the IBM Canada Lab in Toronto and holds a Bachelor of Applied Science degree in Computer Engineering from the University of Toronto.

**Olaf Mueller** is a DB2 and Optim™ Consultant in Germany. He has 20 years of experience in the database field. He holds a pre-degree in chemistry from the Johannes-Gutenberg Universitaet in Mainz/Germany. His areas of expertise include skills in DB2 LUW, Oracle, and Optim, as well as strong migration skills to DB2 LUW.



**Malcolm Singh** is a Software Development Analyst at the IBM Canada Lab - Toronto Site (formerly the IBM Toronto Lab). He works in the Information Management division within the IBM Software Group. Malcolm started his career at IBM as an intern working on DB2 for Linux, UNIX, and Windows (DB2 LUW). He continues to work with DB2 LUW and has an extensive knowledge of both DB2 LUW and Database Theory. His current focus is on the DB2 engine, which includes the DB2 Kernel, Recovery, and Optimizer components.



**Tapio Väättänen** is an Advisory IT Specialist with IBM Global Technology Services in Finland. He has more than 15 years of experience in the IT industry and extensive experience in the area of database and UNIX server administration. Tapio is an IBM Certified DB2 Administrator and VMware Certified Professional as well as *the* Open GROUP Certified IT Specialist. Tapio is currently working on the Finnish DB2 team, providing consultation services and supporting DB2 customers focusing on high availability, performance tuning, and disaster recovery solutions.

## Acknowledgement

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

Boris Bialek
Andre Albuquerque
Piotr Pruski
Anoop Sood
Melody Ng
Peter Kokosielis
**IBM Toronto Laboratory, Canada**

Pauli Rämö
Pekka Siekkinen
**IBM Finland**

Pentti Karikko
**Isoworks, Finland**

Toni Einola
**DataInfo Helsinki, Finland**

Emma Jacob
**International Technical Support Organization, San Jose Center**

# Become a published author

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

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

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

**ibm.com**/redbooks/residencies.html

## Comments welcome

Your comments are important to us!

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

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

  **ibm.com**/redbooks

- Send your comments in an e-mail to:

  redbooks@us.ibm.com

- Mail your comments to:

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

1

# Introduction

Server virtualization is widely implemented to increase server resource utilization by consolidating servers. In this chapter we provide a general overview of virtualization and then describe server virtualization in more detail. The explanation is at a level of understanding for anyone new to the topic of virtualization as well as for those who must reacquaint themselves with server virtualization terminology. Along with this, we briefly discuss the benefits of virtualization. Also, we include a brief history of server virtualization to explain how this technology was first developed.



# 1.1 Overview

Virtualization is becoming more popular due to its increasing ease to efficiently utilize resources. Even though IBM has been using virtualization since the 1960s, there is a rapid growth for virtualization on UNIX and x86 platforms. This growth in virtualization is first evident with server consolidation in data centers, but it also improves business flexibility to meet company needs on demand.

Server virtualization technologies are becoming more mainstream to help efficiently utilize resources by consolidating servers. A consolidated server can host more than one virtual machine by sharing hardware resources. The virtual machines themselves are provided to users as an isolated working environment. In fact, these working environments could easily be perceived as being hosted by a separate stand-alone server and not from a virtual environment created by a virtual machine.

## 1.1.1 What is virtualization

Virtualization is, from a computer science and engineering perspective, the abstraction of a physical computing environment using generated virtual resources to create a logical simulated environment.

There are many types of virtualization, but they all do one of two things:

- Create a smaller working environment.
- Create a larger working environment.

Multiple working environments created from a single physical computing environment result in a smaller but similar working environment, whereas a larger working environment is built upon many physical computing environments to create one working environment. So virtualization, in a general sense, either creates a smaller or larger working environment that is similar to the underlying hardware.

The most recognizable virtualization that everyone can relate to is the partitioning of a hard disk drive (HDD). In a personal computer (PC) environment, a large HDD is usually divided into smaller partitions. Each partition is then identified as a separate disk drive to the system user. But in reality each separate disk drive is from the same HDD with the same underlying characteristics. In this case, smaller logical working environments are created from one physical environment similar to the underlying hardware.

In the other case, a set of HDDs can be combined to create one larger storage space. This larger storage space is viewed as one homogeneous disk to the system user, which is commonly referred to as a *logical volume group* (LVG). The LVG comprises the HDDs with the same underlying characteristics, so a larger logical working environment is built from more than one physical environment that is similar to the underlying hardware.

There are many types of virtualization being used today. The most common types of virtualization are:

- Server virtualization

  Server virtualization creates multiple virtual servers within a single physical server. These virtual servers are independent working environments that use virtual resources, where the virtual resources are an abstraction of the underlying hardware from the physical server. As a result, the virtual resources share the same characteristics as underlying hardware. So the virtual server is exactly like the physical server, only smaller in capacity.

  The types of virtual resources that are used by the virtual server include CPU and memory, which can be shared or dedicated resources among the virtual servers hosted on a single physical server.

  For instance, two enterprise servers each have two 4-core CPUs. These two enterprise servers are both under utilized. If the capacity allows, you can have two virtual servers on one enterprise server sharing the two CPUs. You also can dedicate one CPU to each virtual server. This consolidation frees you one enterprise server for other applications and maximizes physical resource usage while maintaining capacity.

  This virtualization is the primary focus of this book and is discussed in more detail in the remaining chapters.

- Storage virtualization

  Storage virtualization used in enterprise environments is essentially the amalgamation of physical storage. Multiple physical storage devices are combined into a single logical resource. This single logical resource appears as a single storage device to the system user. The use of logical resources creates an abstraction by hiding the complexities of the physical storage devices. This abstraction improves the management and administration of the storage devices.

- Network virtualization

  Network virtualization usually involves the splitting of available bandwidth into separate smaller channels. The smaller channels allow the network to be shared among different devices, which include servers and storage arrays. However, even though the bandwidth is shared, the separate channels can be isolated from each other. This helps improve the network resource utilization and the management of the network infrastructure.

## 1.1.2 History

While server virtualization is becoming more popular, it is based on a technology developed in the late 1960s. This technology was developed and made available by IBM when it shipped the System/360 Model 67 mainframe in 1966. This was achievable by using the CP-67/CMS, which was the successor to the experimental prototype CP-40. The CP-67/CMS was the Virtual Machine Monitor (VMM) that virtualized all of the hardware interfaces on the mainframe.

However, at that time the CP/CMS was only available in source code form without any support. Full support for this ground-breaking virtualization technology commenced in 1972. This occurred after the CP/CMS was reimplemented for the System/370 mainframe as the VM/370. It was also at this time that the term *hypervisor* was coined for this new technology. (This was in relation to when the mainframe operating system was referred to as the supervisor.)

## 1.1.3 Benefits

Server virtualization provides numerous benefits by consolidating many physical server environments into fewer servers by sharing resources. This allows one physical server to function as multiple virtual servers. The consolidation of working environments helps simplify the overall infrastructure, lower the total cost of ownership (TCO), and address environmental issues. Along with this, server virtualization aids in improving responsiveness and business resiliency.

### Infrastructure simplification

The consolidation of servers takes the use of multiple servers by reducing them into fewer servers. This allows one server to host many once-dedicated servers that would be under utilized on separate physical servers. Server consolidation:

- Reduces server sprawl
- Increases physical server utilization
- Improves infrastructure manageability

### Total cost of ownership

The use of fewer servers to deliver and meet business needs reduces the overall total cost of ownership. This produces an increase in the return of investment (ROI) when using virtualization. This is achieved by:

- Increasing server utilization
- Decreasing management infrastructure costs
- Lowering the cost to deploy new environments

### Environmental issues

The current focus on using energy resources more efficiently can be aided with using virtualization. This is primarily achieved by reducing the number of physical servers. With fewer servers needed, the following environmental concerns are addressed:

- Reduce electrical energy consumption.
- Decrease cooling resources.
- Decrease physical space.

### Improved responsiveness

The use of virtualization allows resources to be shared among the virtual servers. Shared resources can be re-allocated as needed to maintain capacity needs for expected and unexpected workloads. Using shared resources can effectively:

- Dynamically respond to application workloads.
- React to changing business needs and cycles.
- Improve overall resource manageability.

### Business resiliency

Virtualization can aid in creating a resilient and highly available (HA) infrastructure. This type of infrastructure lessens the impact of planned and unplanned outages, which can include a full disaster recovery. So this virtualization can help:

- Increase the availability for application software.
- Insulate users from system failures.
- Manage high availability (HA) environments with less cost.

## 1.2 Terminology and definitions

Server virtualization is described as an abstraction of physical hardware resources to create virtual working environments. The virtual working environments are created by using virtual resources to make virtual servers. As a result of using virtual resources, multiple virtual servers can be hosted on one physical server, which is based on capacity requirements, while the virtual servers are managed and controlled by a virtual machine monitor (VMM). However, server virtualization is implemented using different techniques:

- Full virtualization
- Paravirtualization
- Hardware-assisted virtualization
- Operating system (OS)-based virtualization

The first three types of virtualization are considered to be types of machine-based virtualization, which is different from OS-based virtualization, which is based on where the virtual machine monitor is located. But all types of server virtualizations use virtual servers and virtual machine monitors.

## 1.2.1 Virtual server

The virtual server is also commonly referred to as the virtual machine (VM). The virtual machine is the working environment created from virtual resources. These virtual resources include CPU, RAM, hard drives, and other I/O interfaces. The encapsulation of the virtual resources creates an isolated but compatible environment similar to the underlying hardware. This allows the VM to run its own operating system, which is referred to as the guest OS. So the VM seems like a physical server to the system users.

## 1.2.2 Virtual machine monitor

The governing of the virtual machines on the physical server is handled by the VMM. The virtual machine monitor is also known as the *hypervisor*. The hypervisor controls the resources between the physical hardware and the virtual machine. With this control the hypervisor also manages each guest OS used by each virtual machine. This allows each guest OS to run concurrently in isolation from each other. Depending on the hypervisor implementation, the guest OS can be different among the virtual machines.

The hypervisor is an additional layer within the software stack, which is different when comparing a virtualized server to a non-virtualized server. In a non-virtualized server there is only the hardware, operating system, and software applications, as illustrated in Figure 1-1.

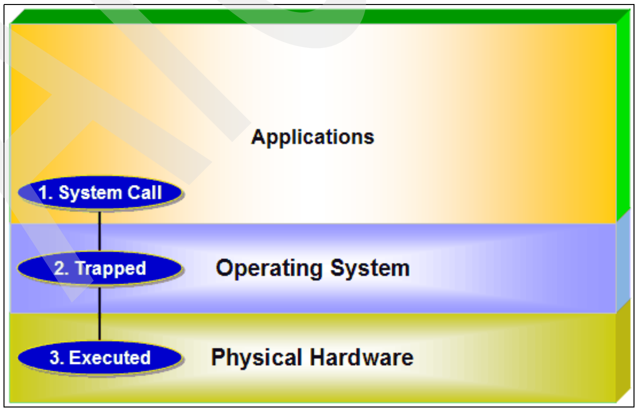



*Figure 1-1 Non-virtualized server*

The hypervisors used in server virtualization are classified as either type 1 or type 2:

- Type 1

  This type of hypervisor runs directly on top of the host hardware. This provides a higher level of virtualization and security since the hypervisor controls the hardware. Using this model, the guest OS is on the second layer above the hardware, as illustrated in Figure 1-2. This hypervisor is also referred to as *bare-metal* or *native*.

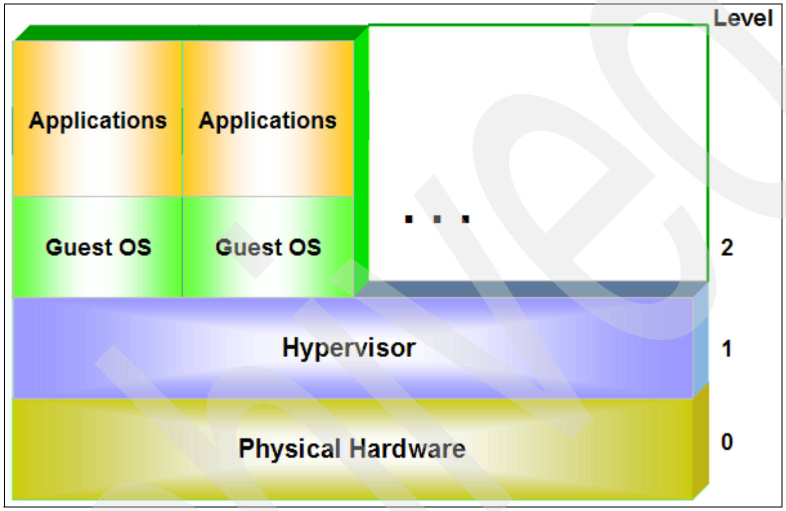



*Figure 1-2 Type 1 hypervisor*

- Type 2

  This type of hypervisor runs on top of an existing operating system. This provides wider support of hardware resources since the operating system manages the resources. Using this model, the guest OS is on the third layer above the hardware, as illustrated in Figure 1-3. This type of hypervisor is also referred to as *hosted*.

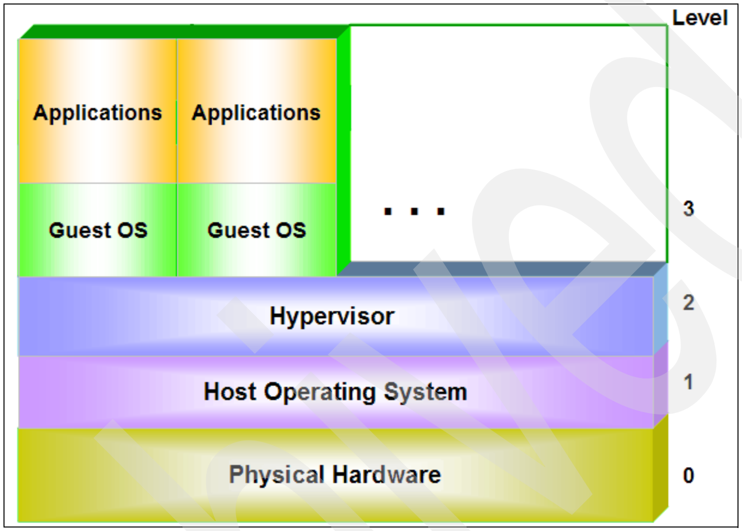



*Figure 1-3 Type 2 hypervisor*

### 1.2.3 Machine-based virtualization

There are three types of machine-based virtulization:

- Full virtualization
- Paravirtualization
- Hardware-assisted virtualization

With machine-based virtualization, the hypervisor is placed directly on top of the hardware. This allows the hypervisor to control the hardware while managing the virtual machines. Depending on the hypervisor, different operating systems or the same operating system at different levels can be used within each separate virtual machine. But the key difference between these three types of virtualizations is how privileged-mode or kernel-mode calls are handled and executed on the CPU, whereas user-mode calls always run directly against the CPU.

### Full virtualization

In a full virtualized environment the hypervisor must intercept privileged instructions from the guest OS. The privileged instruction then must be simulated by the hypervisor to fulfill the request on the hardware. This is illustrated in Figure 1-4. Using this implementation, the guest OS does not need to be modified. However, trapping instructions inside the hypervisor takes longer to execute than if running the same privileged instructions directly on the CPU. Therefore, full virtualization causes performance to greatly decrease in comparison to physical implementations.

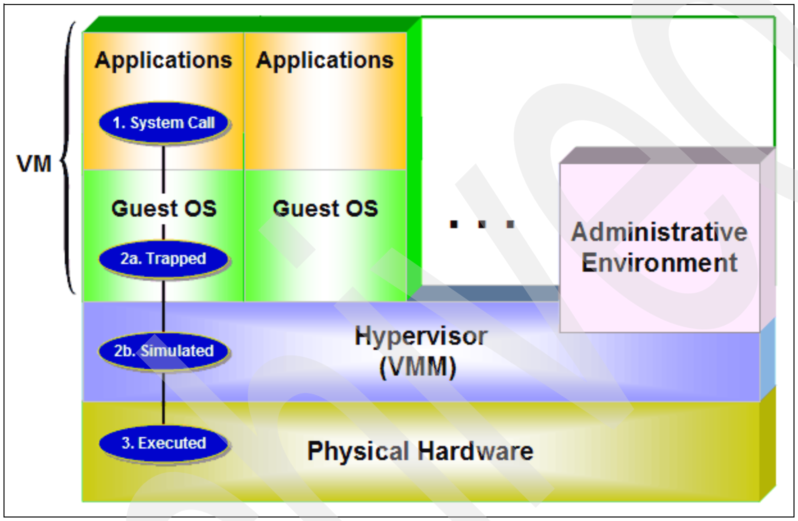



*Figure 1-4 Full virtualization*

## Paravirtualization

Unlike full virtualization, paravirtualization allows privileged instructions to be run directly against the CPU. This means that the hypervisor does not need to intercept the privileged instruction for simulation. This is illustrated in Figure 1-5. However, this can only be achieved if the guest OS is modified to cooperate with the hypervisor. The guest operating system must be ported with the hypervisor API, which might not be adapted by all operating systems.

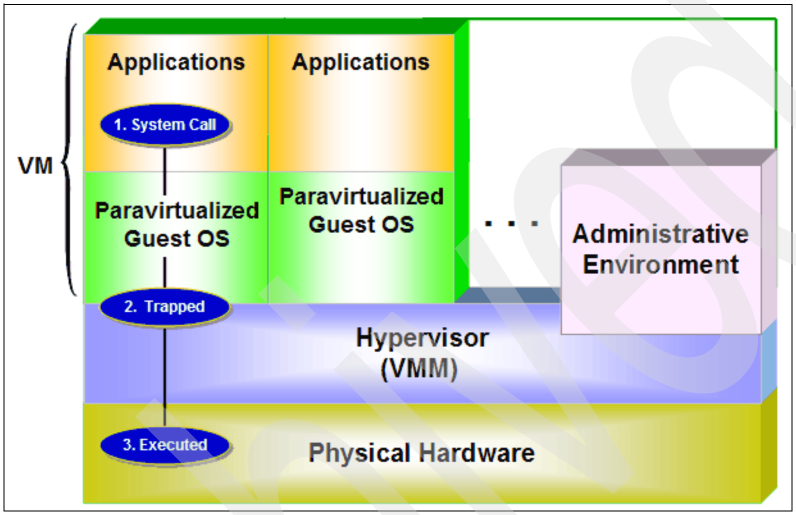



*Figure 1-5 Paravirtualization*

### Hardware-assisted virtualization

The benefits of full virtualization and paravirtualization are combined with hardware-assisted virtualization. This is where the guest OS can directly execute privileged instructions on the CPU without being modified. Figure 1-6 illustrates this. However, the CPU must be able to handle the privileged-mode or kernel-mode calls by using virtualization extensions. This allows the trapping instruction to be handled at the hardware layer rather than at the software layer.

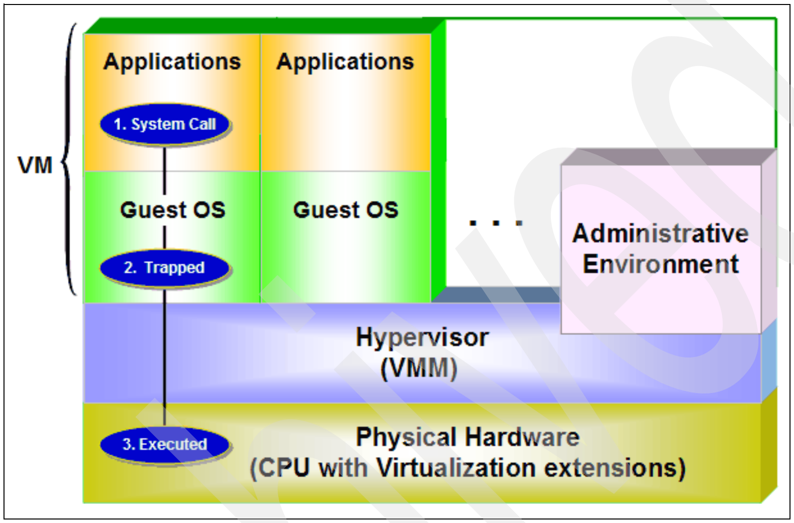



*Figure 1-6 Hardware-assisted virtualiztion*

## 1.2.4 Operating-system-based virtualization

This type of server virtualization is commonly referred to as OS-level virtualization. Operating-system-level virtualization uses a different technique from machine-based virtualiztion to isolate the virtual machines, which are also referred to as virtual instances. Instead of using a separate hypervisor on top of the hardware, the hypervisor is built into the operating system. This requires the operating system kernel to be modified. Therefore, there is no separate hypervisor level since the hypervisor is at the operating system level, as illustrated in Figure 1-7 on page 12. The main advantage is that since there is no separate hypervisor level native performance is maintained. However, each instance is tied to the main host operating system, so different levels of the operating system cannot be used, nor can different OS be used.

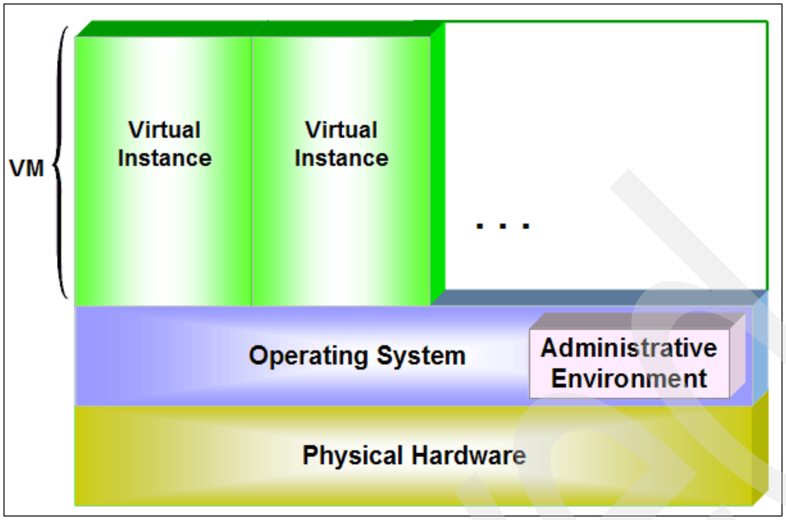



*Figure 1-7 OS-level virtualizaion*

# 2

# Virtualization technologies

This chapter provides an high-level introduction to various virtualization technologies and products currently available in the market. We focus on those supported by DB2 for Linux, UNIX, and Windows Version 9 (DB2 9) and later.

First we describe the DB2 9 built-in support for virtualization, followed by the supported virtualization technologies. The order of the solutions discussed is random. There is no rank implication. In the subsequent chapters we provide detail discussions about a number of virtualization solutions.

## 2.1 DB2 support for virtualization

In this section we first introduce the virtualization environments supported by DB2 9. Then we describe the DB2 9 built-in features and functions that enable DB2 9 to run very well in different virtualization environments. Finally, we discuss the DB2 9 licensing model as it relates to virtualization.

### 2.1.1 Support matrix

DB2 9 provides support for many virtualization environments. For the most up-to-date list of the supported environments, refer to:

http://www.ibm.com/developerworks/wikis/display/im/DB2%20Virtualization%20Support

That Web site also contains information about the restrictions of the supported environments. Table 2-1 to Table 2-4 on page 15 show the virtual environments supported by DB2 9 and what versions of DB2 9 support these architectures.

*Table 2-1 Full virtualization environments for x86 and x64 architectures*

| Hypervisor | Architecture | Minimum guest OS (Windows) | Minimum guest OS (Linux) | Minimum DB2 level |
|---|---|---|---|---|
| VMware ESX 3.0.1 and later<br>VMware vSphere 4 | x86/x64 System listed on ESX HCL | Any Microsoft Windows level supported by both DB2 and ESX | Any Linux distribution supported by both DB2 and ESX | DB2 9.1<br>DB2 9.5<br>DB2 9.7 |
| Red Hat Enterprise Linux (RHEL) 5.2 and later<br>Xen HVM | x64 System with INTEL-VT or AMD-V | Not supported | RHEL 5.2 64 bit | DB2 9 FP4<br>DB2 9.5 FP1<br>DB2 9.7 |
| SUSE Linux Enterprise Server (SLES) 10 SP2 and later<br>Xen HVM | x64 System with INTEL-VT or AMD-V | Not supported | SLES 10 SP2 64 bit | DB2 9 FP4<br>DB2 9.5 FP1<br>DB2 9.7 |
| Microsoft Windows 2008 SP2<br>Hyper-V | x64 System with INTEL-VT or AMD-V | Windows 2008 Server SP2 | Not supported | DB2 9.5 FP4<br>DB2 9.7 |

*Table 2-2 OS virtualization environments for x86 and x64 architectures*

| Technology | Architecture | Operating system | Minimum DB2 level |
|---|---|---|---|
| Solaris Zones | x64 | Solaris 10 | DB2 9 FP4<br>DB2 9.5 FP1<br>DB2 9.7 |
| Parallels Virtuozzo Containers | x64 | Windows 2003<br>SLES 10 | DB2 9.1<br>DB2 9.5 |

*Table 2-3 Full virtualization environments for non-x86 platforms*

| Hypervisor | Architecture | Minimum guest OS | Minimum DB2 level |
|---|---|---|---|
| PowerVM | IBM Power Systems | AIX¬Æ 5.3 TL05<br>AIX 6.1<br>RHEL 5<br>SLES 10 SP1 | DB2 9.1<br>DB2 9.5<br>DB2 9.7 |
| z/VM¬Æ 5.2<br>z/VM 5.3 | IBM System z¬Æ | SLES 9 SP3<br>SLES 10 SP1<br>RHEL 4 U4<br>RHEL 5 (64-bit only) | DB2 9.1<br>DB2 9.5<br>DB2 9.7 |
| HP-UX Virtual Partitions (vPars) | HP Integrity Servers | HP-UX 11i v3 | DB2 9 FP5<br>DB2 9.5 FP2<br>DB2 9.7 |

*Table 2-4 OS virtualization environments for non-x86 platforms*

| Technology | Operating system | Minimum DB2 level |
|---|---|---|
| AIX System Workload Partitions (WPARS) | AIX 6.1 | DB2 9 FP4<br>DB2 9.5<br>DB2 9.7 |
| Solaris Zones | Solaris 10 | DB2 9.1<br>DB2 9.5<br>DB2 9.7 |

## 2.1.2 Features and functions

DB2 9 contains several features and functions that are not built in particular for virtualized environments but that are very beneficial in those environments. In this section we focus on the built-in autonomic features and the various data compression features enabled.

The DB2 built-in autonomic features significantly reduce the amount of time that a DBA must spend on keeping up a database. This is very important, as the number of databases keeps increasing in this information explosion era. The autonomic features help increase DBA's productivity and reduce a company's total cost.

I/O throughput is a major concern with all databases, especially in virtualized environments where several virtual machines share the same physical I/O interface. The DB2 compression features help to increase the I/O throughput significantly. This capability makes them a perfect feature for virtualized environments.

### Autonomic features

In today's business environment, database vendors face many challenges and business requirements. Some of these are:

- Databases are getting larger and more complicated.
- Return on investment.
- DBA skills: For example, are all DBAs as highly qualified as necessary and do they have time to enhance or obtain their skills?
- Efficient use of manpower: Highly skilled DBAs should spend less time in routine maintenance tasks.
- Maintenance windows are becoming smaller and less frequent.

All of these topics are related to one another. They all can be alleviated with DB2 9 autonomic features.

When looking at built-in autonomic features in DB2 9, there are four major areas to discuss:

- Self-configuration
- Self-healing
- Self-managing
- Self-optimization

These areas are depicted in Figure 2-1.

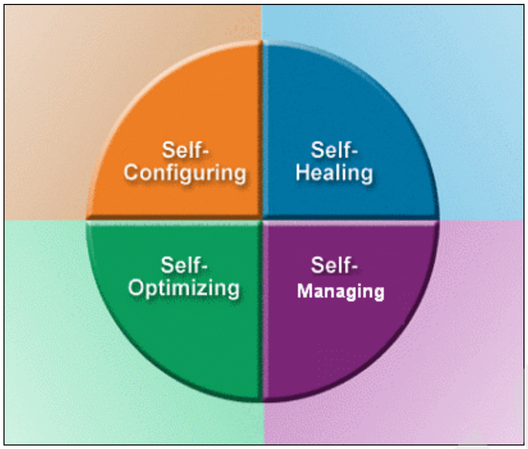



*Figure 2-1 Autonomic computing areas*

### *Self-configuration*

Self-configuration of DB2 starts at the creation of a database. If you create a DB2 database the so-called Configuration Advisor (CA) is started automatically in the background to collect the environment characteristics of your system. Based on the collected information, several instance and database parameters, including buffer pools, are adjusted to make your database run well in your environment from the beginning. You also can provide additional information to the CA such as number of concurrent applications, number of statements per transaction, workload type, and so on.

Combined with a mathematical model of each configuration parameter, based on expert heuristics, the CA calculates optimal values for several parameters and buffer pools. Figure 2-2 shows this process.

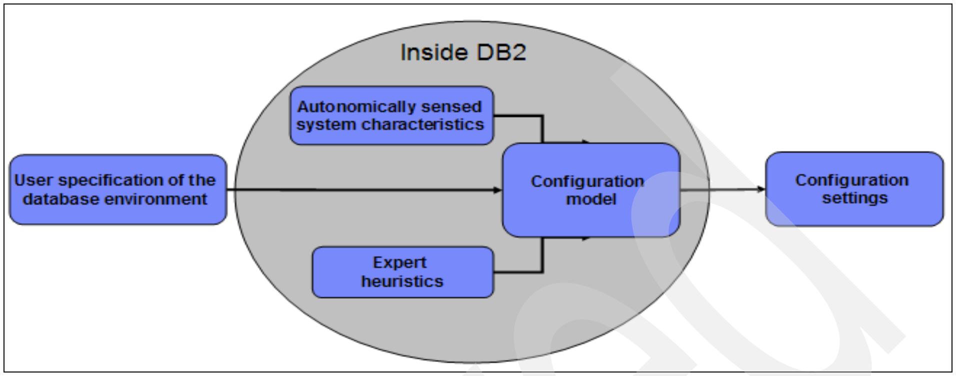



*Figure 2-2 Self-configuration*

For more detailed information about CA refer to:

http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.dbobj.doc/doc/c0052481.html

### Self-healing

The built-in DB2 9 Health Monitor checks the healthiness of all active database objects as frequently as deemed necessary. It is able to generate alerts based on whether a health indicator exceeds a threshold or is in a non-normal state. In case of an alert, it sends notifications to the DB2 administration notification log and e-mails or pages the contacts on the notification list. The DB2 9 Health Monitor also advises about the severity of an alert. You can define corrective actions (scripts or tasks) for health alerts to be executed automatically. You can use both the DB2 command line and the Health Center GUI to administrate the Health Monitor. The Health Center allows you to define the corrective actions, health indicator settings, and many other features.

Figure 2-3 illustrates the health checking algorithm.

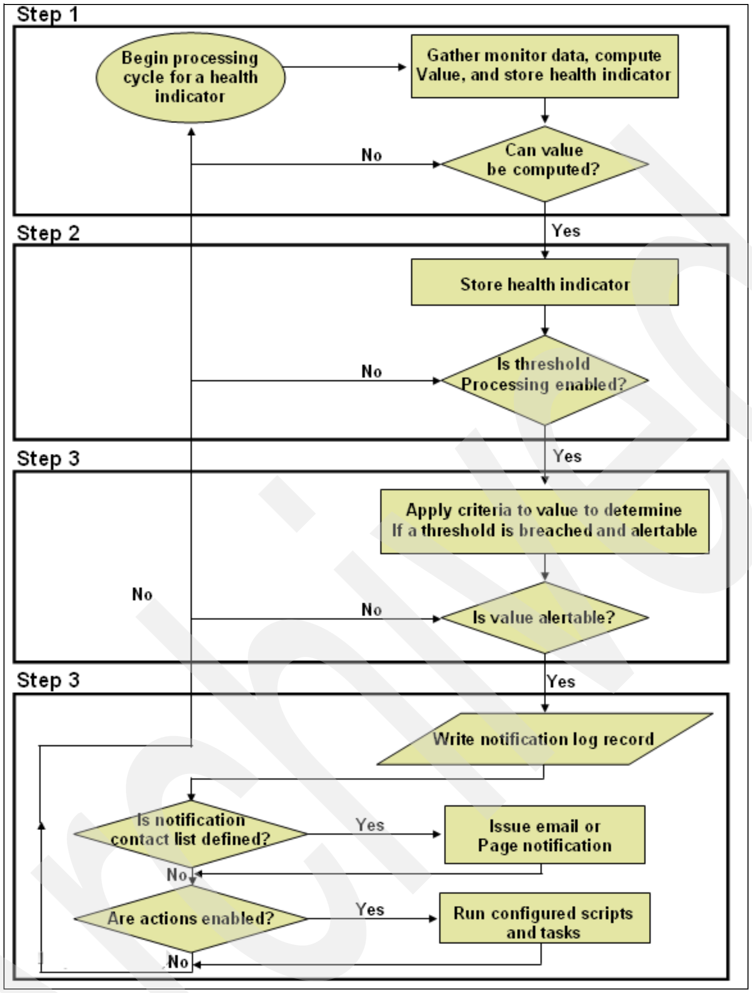



*Figure 2-3 Self-healing algorithm*

For more detailed information see:

http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.mon.doc/doc/c0011709.html

Note that the DB2 9 Health Monitor and Health Center are deprecated in DB2 9.7. New tools are available with the IBM Optim solutions. This tool suite replaces the old tools that came with DB2. You can find more detailed information at:

http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.idm.tools.doc/doc/c0055013.html

### *Self-managing*

The self-managing capabilities include automatic object maintenance and automatic storage management:

- Automatic object maintenance

  The automatic object maintenance self-managing features are enabled and disabled through the database configuration parameters. There is a hierarchy between these parameters, as shown in Example 2-1.

  *Example 2-1 Automatic maintenance parameters*

```
AUTO_MAINT (ON
  AUTO_DB_BACKUP (OFF)
    AUTO_TBL_MAINT (ON)
      AUTO_RUNSTATS (ON)
        AUTO_STMT_STATS (OFF)
      AUTO_STATS_PROF (OFF)
        AUTO_PROF_UPD (OFF)
      AUTO_REORG (OFF)
```

  AUTO_MAINT is the master on/off switch. Individual child parameters can be set to ON/OFF and the settings are persisted in the database configuration file. These automatic maintenance features are integrated with the Health Monitor. On the following Web site you can find much more detailed information about the automatic maintenance in DB2 9:

  http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.dbobj.doc/doc/c0021757.html

- Automatic storage management

  With automatic storage management, DB2 will allocate storage on demand as the table consumption grows. This feature intends to be a single point of storage management for table spaces. DBAs are no longer required to define the containers for table spaces but just specify a group of storage devices for DB2, for example, file systems. DB2 creates the necessary containers automatically across the storage paths. The growth of the existing containers and the additional new ones is completely managed by DB2. To learn more about automatic storage management visit the following Web site:

  http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.dbobj.doc/doc/c0052484.html

### *Self-optimization*

Self Tuning Memory Manager (STMM), introduced with DB2 9.1, is a revolutionary feature that manages DB2 memory allocation and usage. STMM is able to adjust certain memory heaps of DB2 according to the workload of a database. All those memory heaps are part of the database shared memory set. Figure 2-4 depicts the different memory sets of DB2.

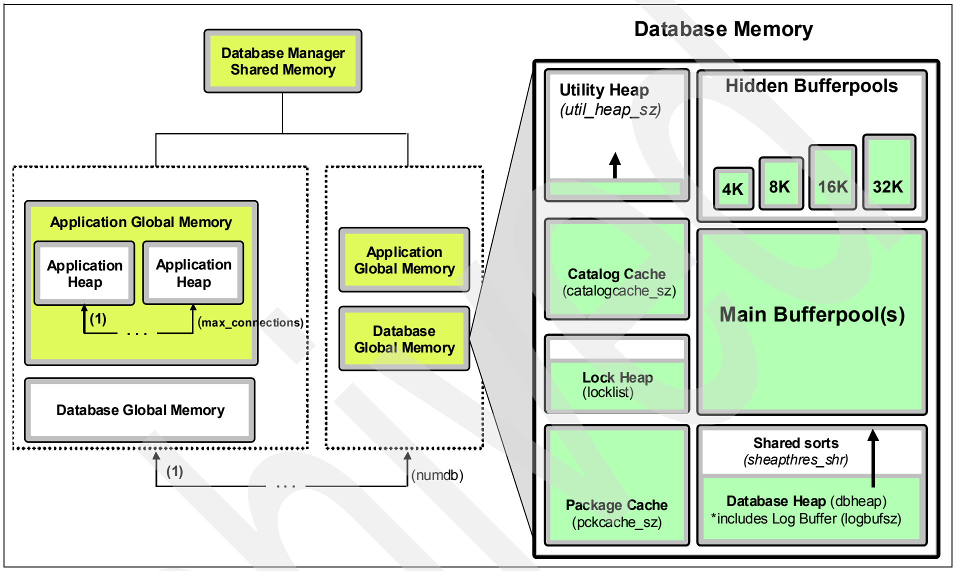



*Figure 2-4 DB2 memory heaps*

STMM constantly monitors the system to make use of or return any free memory to the OS. It works iteratively to determine an optimal memory configuration for all heaps. The iterative approach prevents instability of the system. Control algorithms help determine interval length and prevent oscillations. In each interval, each heap can grow only by 50% or decrease by 20%.

Figure 2-5 shows the STMM work flow.

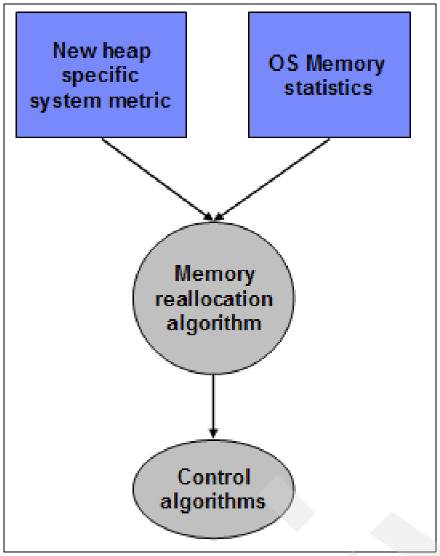



*Figure 2-5 STMM work flow*

Figure 2-6 on page 23 describes the algorithm used for STMM during each check interval. The general process of this algorithm is:

1. The tuner process wakes from sleep.
2. Determine whether memory configuration is sub-optimal. Some heaps are in need of memory, while others own more than required.
3. If the DATABASE_MEMORY database configuration (dbm cfg) parameter has been set to automatic, DB2 checks whether OS has free memory and uses the free memory from OS to satisfy the needy heaps.
4. If the set value of DATABASE_MEMORY has all been used and there is no available memory in OS, DB2 will allocate memory from the heaps with excessive memory to those with a shortage.
5. Continue the process until no more memory can be moved.
6. Determine the tuning frequency based on workload.

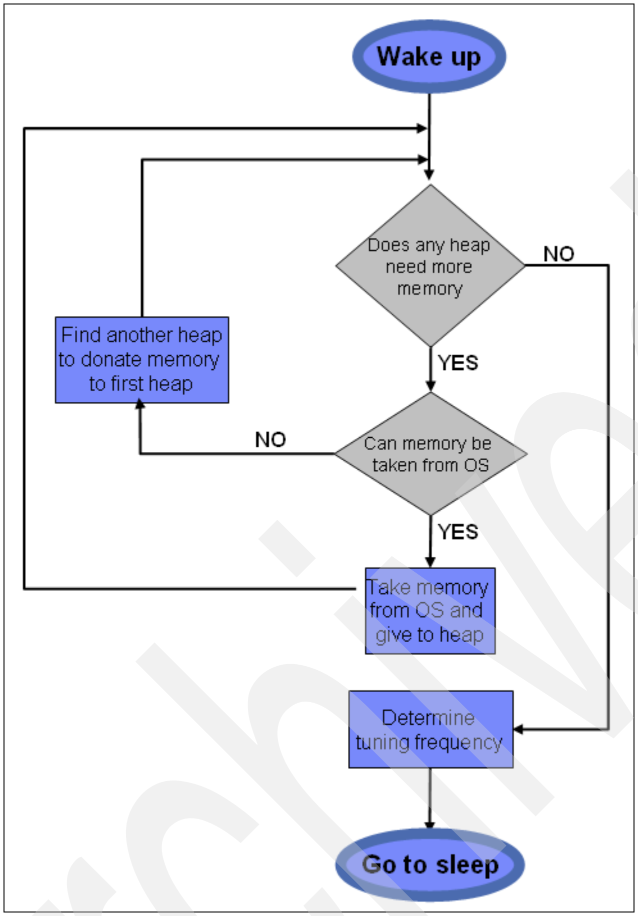



*Figure 2-6 STMM algorithm*

More information about STMM can be found at:

http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.perf.doc/doc/c0021627.html

## Compression

A discussion about compression is mainly about saving storage. Storage is usually the most expensive component of a database solution. Compressing data can save floor space and personnel cost for managing storage, as well as power and cooling.

A second aspect is the performance. Compression helps to improve the I/O efficiency of your database. Because of compression, the database requires fewer I/O operations to retrieve the same amount of data. This is very important because accessing data from disk is the slowest database operation.

DB2 9 provides various compression options. The DB2 Storage Optimization Feature includes all of the compression features except the NULL and default compression and the Extensible Markup Language/large object (XML/LOB) inlining. Enabling the DB2 Storage Optimization Feature requires a separate license that is available for the DB2 9 Enterprise Server Edition only.

In this section we introduce all the compression features. For more information see:

http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.dbobj.doc/doc/c0055401.html

### NULL and default value compression

These two compression techniques were introduced prior to DB2 9. They are the first compression features established in DB2. If NULL value compression is enabled for a table, NULL and zero-length data assigned to the variable-length data types will not be stored on disk. Default value compression helps you to further save disk space by not storing inserted/updated values that equal the system default values.

### LOB inlining and XML inlining

XML and large objects (LOBs) are either stored outside the base table in a separate storage object or, if adequately sized, stored in the formatted rows of the base table. The adequate size depends on the page size of the table. For example, for a 32 K page size the maximum size for inlining is 32,669 bytes. The descriptors stored in the base table rows are used to keep track of the associated XML/LOB data in the storage object. Figure 2-7 depicts the strategy of inlining.

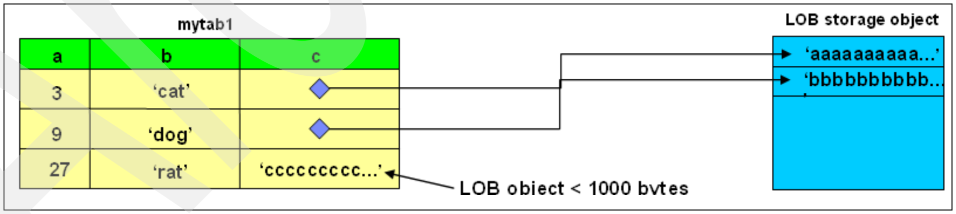



*Figure 2-7 XML and LOB inlining*

If a table possesses XML or LOB data that can be stored inline, there are considerable benefits with respect to performance and storage usage. Inlined XML or LOBs can be buffered, reducing I/O costs. Storage allocated to the storage object is reduced by inlining the XML/LOB data in the base table though the base table storage increases. Inlining small XML/LOBs can result in a noticeable decrease in the net total storage since the decrease in the storage size is greater than the increase in the base table storage size. XML/LOBs inlined within the base table data can be compressed when the row compression is enabled.

### *Row compression*

DB2 9.1 introduces the row compression. It uses a dictionary-based symbol table and a Lempel-Ziv-based algorithm for compressing and decompressing data records. The compressed data is replaced by 12-bit symbols. The dictionary is about 100 KB in size and is stored within the data pages of the compressed table. The reoccurring strings and trailing or leading blanks are compressed for the text data. Figure 2-8 shows the effects of the row compression.

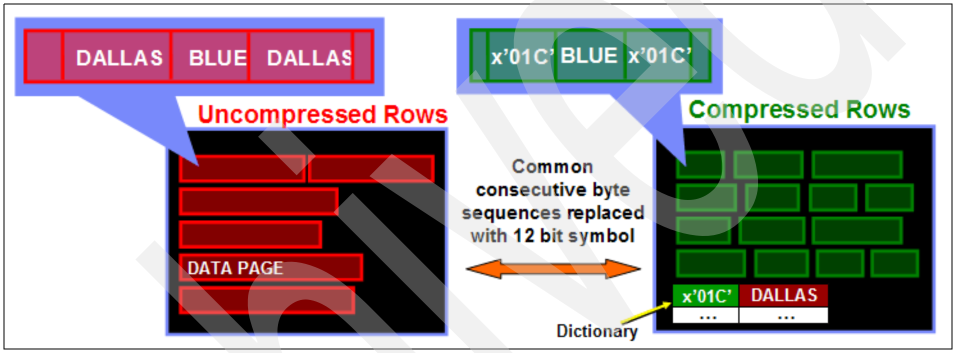



*Figure 2-8 Row compression*

The compressed data remains on disk in the file containers and in the log files. It also remains compressed in memory in the buffer pools and in the log buffer. Thus, we achieve significant I/O bandwidth and memory (buffer pool) savings. DB2 decompresses rows before evaluation. Figure 2-9 depicts all areas where compressed data resides.

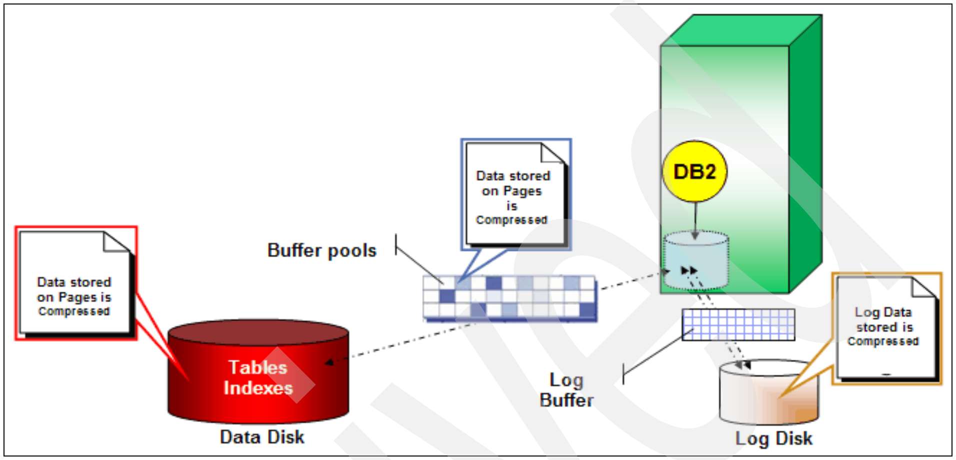



*Figure 2-9 Row compression everywhere*

Before enabling the row compression on a table, check whether your environment is CPU bound. In CPU-bound environments performance becomes worse because of adding the compression and decompression overhead to the CPU. In I/O-bound environments the I/O savings outperform the overhead of compression/decompression.

### *Index compression*

Index compression, introduced in DB2 9.7, uses different algorithms from the row compression. With the index record identifier (RID) list compression, the database manager can compress an index with a large number of duplicate keys by storing an abbreviated format of the RID for the duplicate keys. Figure 2-10 illustrates a simple example.

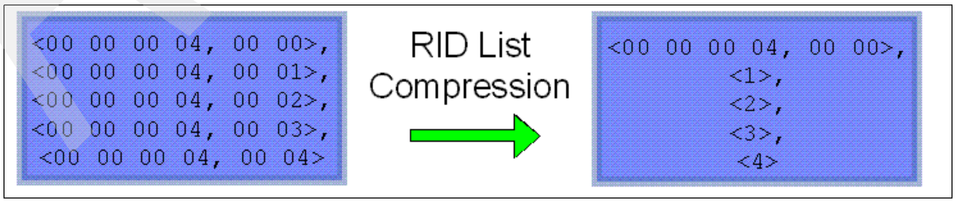



*Figure 2-10 Index RID list compression*

For an index with a high degree of commonality in the prefixes of the index keys, the database manager can apply compression based on the similarities in prefixes of index keys. Figure 2-11 shows an example of this algorithm.

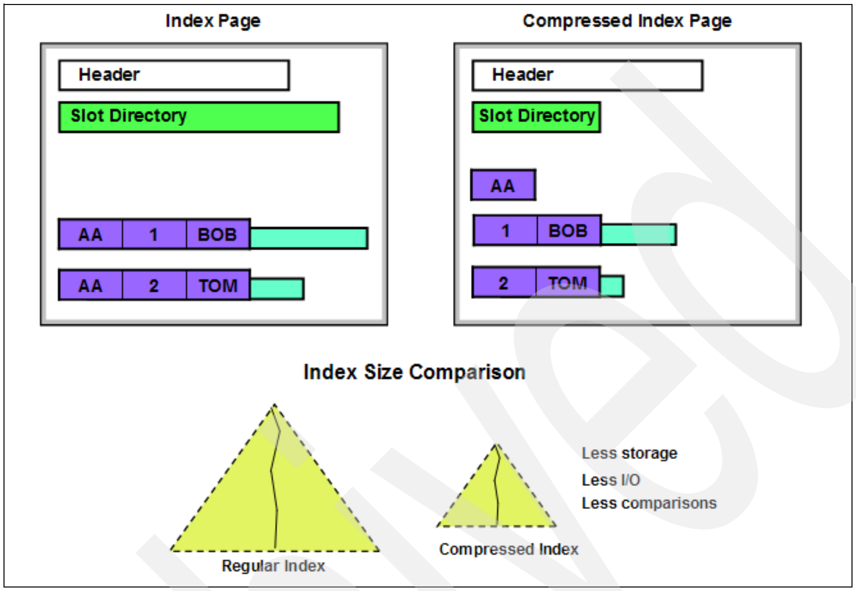



*Figure 2-11 Index prefix compression*

DB2 automatically chooses the most appropriate algorithm to compress indexes. There is no option available to force DB2 to use a specific algorithm.

Index compression is enabled by default for all indexes on a table that is enabled for row compression. It is disabled for all indexes when row compression is disabled for that table. You can overwrite this behavior by using the COMPRESS YES/NO option with the CREATE/ALTER INDEX statements.

Index compression is not available for multi-dimensional clustering (MDC) indexes and XML indexes.

### *Temporary table compression*

The temporary table compression is provided in DB2 9.7. It is applicable to user temporary tables and system temporary tables. User temporary tables are either declared global temporary tables (DGTTs) or created global temporary tables (CGTTs). System temporary tables are created by the DB2 engine mainly during sort or join operations. The compression of temporary tables aims to:

- Reduce the amount of temporary disk space required.
- Have no performance penalty as a result of the extra processing required for row compression.
- Enhance the query performance.

If the DB2 Storage Optimization Feature is licensed, CGTTs and DGTTs are compressed automatically by default.

### *XML compression*

This new DB2 9.7 feature has a similar compression approach to the row compression. XML compression uses a dictionary to replace data that qualifies for compression. In Figure 2-12 we show the XML compression mechanism.

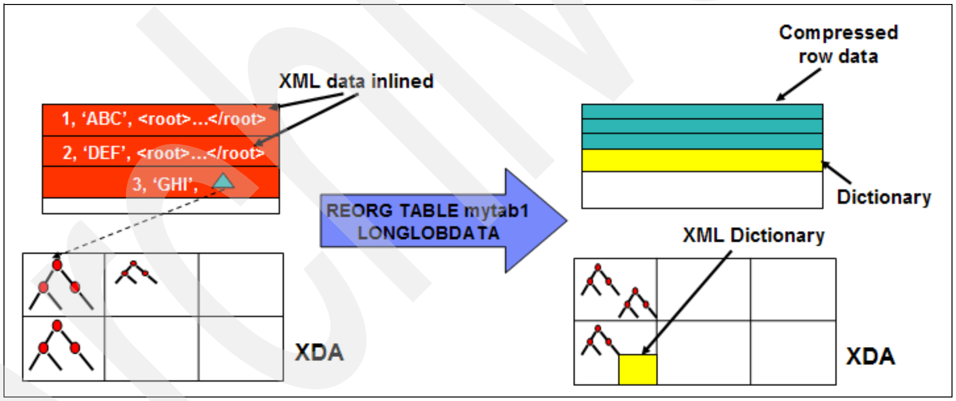



*Figure 2-12 XML compression*

## 2.1.3 Licensing

A major advantage of virtualization is that several virtual machines can run on the same processor or use just some of the processors in a multi-core environment. With the former IBM software licensing plan, you would be charged for the full physical processor capacity of the machine used even if the DB2 VM is just using a part of a multi-core machine. To provide IBM customers more value for a lower price, IBM introduced a new licensing model in April 2005 called sub-capacity licensing. This model was introduced first for the DB2 Enterprise

Edition only. Starting February 2009, this model was valid for all other charged editions as well. Sub-capacity licensing is especially beneficial for the virtualized environments because you only pay the license fees for the resources used by the VM running DB2. The virtualization environments eligible for sub-capacity licensing are listed on the following Web site:

http://www-01.ibm.com/software/lotus/passportadvantage/Counting_Software_licenses_using_specific_virtualization_technologies.html

### Sub-capacity licensing

For this new licensing model IBM introduces a new unit of measure called processor value unit (PVU). To help you understand the PVU licensing model, we first explain some terms used in this context:

- Core: a functional unit within a computing device that interprets and executes software instructions.
- Chip: electronic circuitry containing, but not limited to, at least one core on a silicon wafer.
- Socket: the mount that secures a chip to a motherboard.
- Processor: IBM defines a processor as the core. For example, a dual-core chip has two processor cores on it.

These terms are illustrated in Figure 2-13.

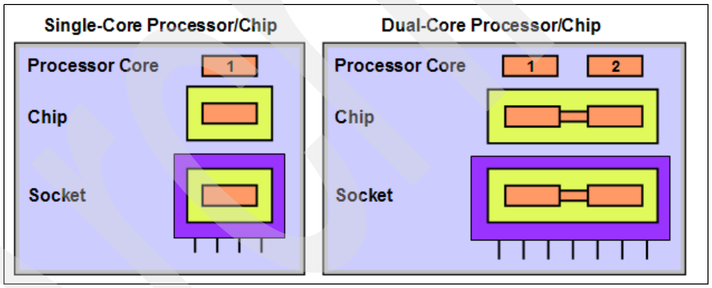



*Figure 2-13 Processor definition*

A PVU is a unit of measure used to differentiate licensing of middleware on distributed processor technologies (defined by processor vendor, brand, type, and model number). IBM defines a certain PVU value per core for each of the supported processor types. Figure 2-14 lists the PVU definitions for different processor types.

| Processor Technologies | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Processor Vendor | Processor Brand | Processor Type: One-Core (1) | Multi-Core: Dual-Core (2) | Multi-Core: Quad-Core (4) | Multi-Core: Hexa-Core (6) | Multi-Core: Octi-Core (8) | IFL Engine | Processor Model Number [1] | PVUs per Core |
| IBM | POWER6 550,560,570,575,595 [2] | | ■ | | | | | All Existing | 120 |
| | POWER6 520,JS12,JS22,JS23,JS43 [2] | | ■ | | | | | " " | 80 |
| | POWER5, POWER4 | | ■ | | | | | " " | 100 |
| | POWER5 QCM | | | ■ | | | | " " | 50 |
| | System z10 [3] | | | | | | ■ | " " | 120 |
| | System z9, z990, S/390 [3,4] | | | | | | ■ | " " | 100 |
| | PowerPC 970 | | ■ | | | | | " " | 50 |
| | PowerXCell™, Cell/B.E.™ 8i [5] | ■ | | | | | | " " | 30 |
| HP (Intel®) | Itanium® | | ■ | | | | | All Existing | 100 |
| | PA-RISC | | ■ | | | | | " " | 100 |
| Sun / Fujitsu | SPARC64 VI, VII | | ■ | ■ | | | | All Existing | 100 |
| | UltraSPARC IV | | ■ | | | | | " " | 100 |
| | UltraSPARC T2 | | | ■ | ■ | ■ | | " " | 50 |
| | UltraSPARC T1 | | | ■ | ■ | ■ | | " " | 30 |
| Intel® | Xeon® | | ■ | ■ | | | | 3500 to 3599, [6] 5500 to 5599 [6] | 70 |
| | | | ■ | ■ | ■ | | | 3000 to 3499, 5000 to 5499, 7000 to 7499 | 50 |
| AMD | Opteron | | ■ | ■ | ■ | | | All Existing | 50 |
| Any | Any single-core (i.e. Xeon Single-Core) | ■ | | | | | | All Existing | 100 |

*Figure 2-14 PVU definitions*



The most current PVU definitions can be found at:

http://www-01.ibm.com/software/lotus/passportadvantage/pvu_licensing_for_customers.html

Figure 2-15 shows that DB2 is running in a partition that uses three processor cores in a six-processor-core activated architecture. With the full-capacity licensing model, the license PVUs for six processor cores will be charged. With the cost-saving sub-capacity model, you only must pay the license for three cores.

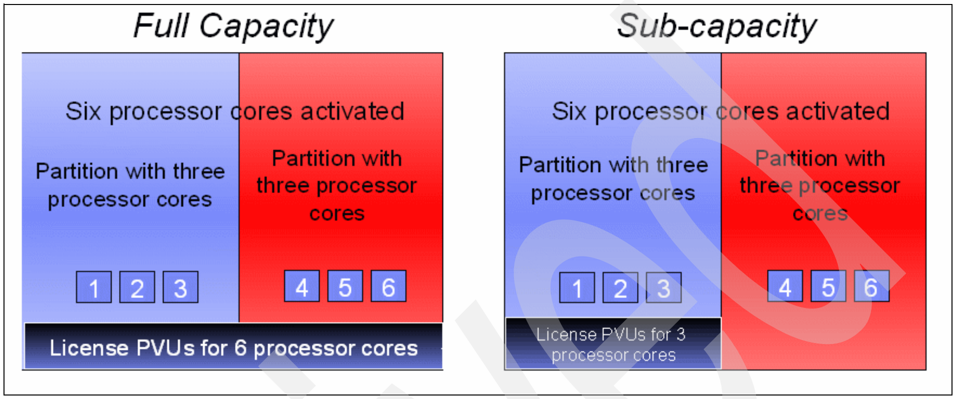



*Figure 2-15 Sub-capacity licensing*

Sub-capacity licensing using PVUs provides:

- A licensing structure that avoids fractional licensing or processor factors for multi-core chips
- Flexibility and granularity, enabling customers to run a product on as few or as many processor cores as they require
- The capability to deliver software price performance improvements as new processor families are introduced
- A sustainable licensing foundation for the future
- Transferability of licenses across distributed systems

All DB2 editions are eligible for PVU licensing. For more detailed information refer to the following Web page:

http://www.ibm.com/software/lotus/passportadvantage/subcaplicensing.html?OpenDocument

## 2.2 PowerVM on Power Systems

PowerVM is the new brand for Power Systems virtualization. It is a combination of hardware, firmware, and software that provides virtualization for CPU, network, and disk. It implements a hardware-assisted virtualization technique.

PowerVM on Power Systems offers industry-leading virtualization capabilities for AIX and Linux. With the Standard Edition of PowerVM (PowerVM-SE), micro-partitioning allows businesses to increase the utilization of their servers, with server definitions down to one-tenth of a processor and the ability to allow server size to flex with demand. In addition, with PowerVM-SE, there is the Virtual I/O Server, which allows the sharing of expensive disk and network resources, while minimizing any management and maintenance costs.

With the introduction of the PowerVM Enterprise Edition, all of these features are joined by the ability to migrate running partitions and their applications from server to server. Combining these PowerVM features, we can help today's businesses further transform their computing departments into the agile, responsive, and energy-efficient organization demanded by today's enterprises.

We discuss more details about the PowerVM capabilities in Chapter 3, "Power Systems and PowerVM" on page 47. Detailed information also can be found at:

http://www.redbooks.ibm.com/abstracts/sg247940.html

Table 2-3 on page 15 provides a list of supported operating systems.

### POWER Hypervisor

The POWER¬Æ Hypervisor‚Ñ¢ is the foundation of IBM PowerVM. Combined with features designed in the IBM POWER processors, the POWER Hypervisor delivers functions that enable capabilities including dedicated-processor partitions, micro-partitioning, virtual processors, IEEE VLAN compatible virtual switch, virtual Ethernet adapters, virtual SCSI adapters, and virtual consoles.

The POWER Hypervisor is a firmware layer sitting between the hosted operating systems and the server hardware, as shown in Figure 2-16. The POWER Hypervisor is always installed and activated, regardless of system configuration. It is controlled and managed by Hardware Management Console (HMC), the focal management point of Power Systems. The POWER Hypervisor has no specific or dedicated processor resources assigned to it.

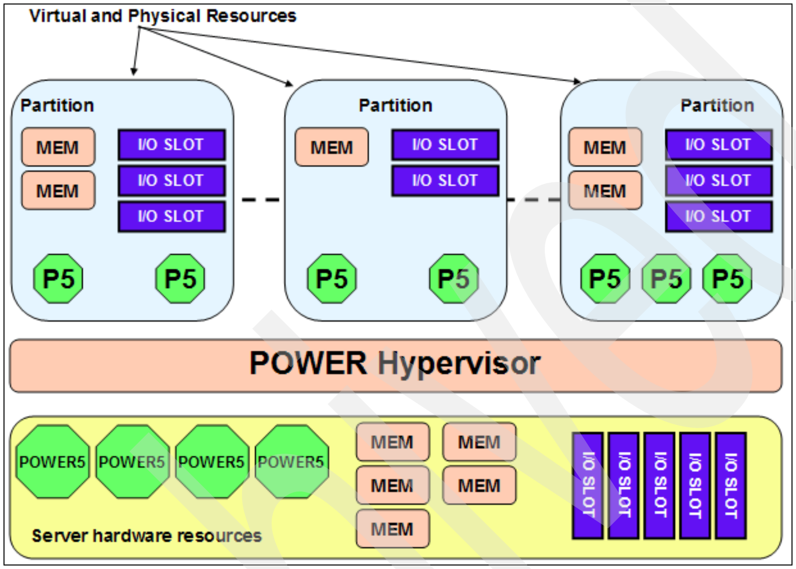



*Figure 2-16 POWER Hypervisor*

In partitioned environments where business-critical applications are consolidated onto the same hardware, exceptional availability and serviceability are needed. This ensures a smooth recovery from unplanned service interruptions. The POWER Hypervisor ensures that issues affecting one partition do not propagate into other logical partitions on the server.

The POWER Hypervisor does not own any physical I/O devices, nor does it provide virtual interfaces to them. All physical I/O devices in the system are owned by logical partitions or the Virtual I/O Server. To support virtual I/O, the POWER Hypervisor provides:

- Control and configuration structures for virtual adapters
- Controlled and secure transport to physical I/O adapters
- Interrupt virtualization and management

## HMC

The primary hardware management solution that IBM has developed relies on an appliance server called HMC, packaged as an external tower or rack-mounted server.

The HMC is a centralized point of hardware control. A single HMC can handle multiple POWER5,Ñ¢ and POWER6¬Æ systems, and two HMCs may manage the same set of servers in a dual-active configuration, providing resilience.

Hardware management is done using the HMC interface (Web-browser-based starting with HMC Version 7), which communicates with the servers using a standard Ethernet connection to the service processor of each POWER5 or POWER6 system. Interacting with the service processor, the HMC is able to:

- Create, manage, and modify logical partitions.
- Modify the hardware configuration of the managed system.
- Manage the service calls.

The HMC also provides functions to simplify the management of the Virtual I/O Server environment. It is also possible to execute Virtual I/O Server commands from the HMC.

## Virtual I/O Server

The Virtual I/O Server is part of the IBM Power Systems PowerVM Standard Edition or Enterprise Edition hardware feature. The Virtual I/O Server allows the sharing of physical resources between partitions to allow more efficient utilization. The Virtual I/O Server can use both virtualized storage and network adapters, making use of the virtual small computer system interface (SCSI) and virtual Ethernet facilities.

For storage virtualization, these storage devices can be used:

- Direct-attached entire disks from the Virtual I/O Server
- SAN disks attached to the Virtual I/O Server
- Logical volumes defined on either of the previous disks
- File-backed storage, with the files residing on either of the first two disks
- Optical storage devices

For virtual Ethernet we can define Shared Ethernet Adapters on the Virtual I/O Server, bridging network traffic from the virtual Ethernet networks out to physical Ethernet networks.

The Virtual I/O Server technology facilitates the consolidation of LAN and disk I/O resources and minimizes the number of physical adapters that are required, while meeting the non-functional requirements of the server. Figure 2-17 shows a very basic overview of a Virtual I/O Server configuration.

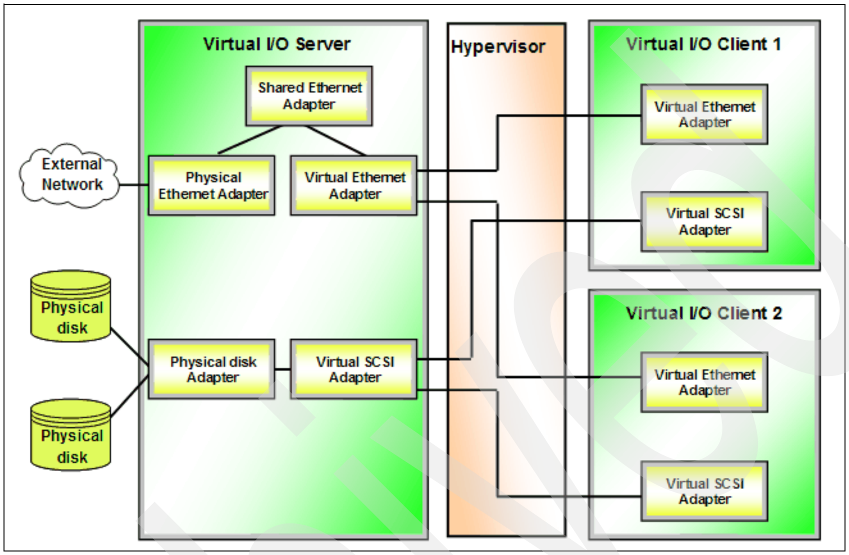



*Figure 2-17 Virtual I/O Server*

## Live Partition Mobility

Live Partition Mobility, licensed through PowerVM Enterprise Edition, is a feature that relies on a number of different components, including:

- POWER Hypervisor
- Virtual I/O Server
- Hardware Management Console

Live Partition Mobility allows you to move running AIX or Linux partitions from one physical POWER6 server to another without disruption. The movement of the partition includes everything that partition is running, that is, all hosted applications. Some possible applications and advantages are:

- Moving partitions from servers to allow planned maintenance of the server without disruption to the service and users
- Moving heavily used partitions to larger machines without interruption to the service or disruption to users

- Moving partitions to appropriate servers depending on workload demands and adjusting the utilization of the server-estate to maintain an optimal level of service to users at the optimal cost
- Consolidation of under utilized partitions out-of-hours to enable unused servers to be shut down, saving power and cooling expenses.

## 2.3 VMware vSphere

VMware vSphere (that is, Virtual Infrastructure 4) is a virutalization solution that delivers IT infrastructure as a service, masking all the complexity of the infrastructure and exposing an easily accessible service to applications. It consists of products and features including:

- A set of infrastructure vServices to aggregate and allocate on-premise servers, storage, and network for maximum infrastructure efficiency
- A set of cloud vServices to federate the on-premise infrastructure with third-party cloud infrastructure
- A set of application vServices to guarantee the correct levels of availability, security, and scalability to all applications independent of hardware and location
- A set of management vServices that allow you to manage proactively the virtual datacenter operating system and the applications running on it

Unlike a traditional operating system, which is optimized for a single server, the virtual data center operating system serves as the operating system for the entire data center.

In this section we introduce some of the vSphere features. In Chapter 4, “VMware vSphere” on page 75, we discuss more details about the VMware products. For more information directly from VMware, visit:

http://www.vmware.com/

### ESX

VMware ESX is a full virtualization environment. VMware provides isolated guest environments called virtual machines. The guest OS is not aware that it is being virtualized and requires no modification. No separate master or parent virtual machine is required to start the guest virtual machines. Figure 2-18 shows how this architecture is built.

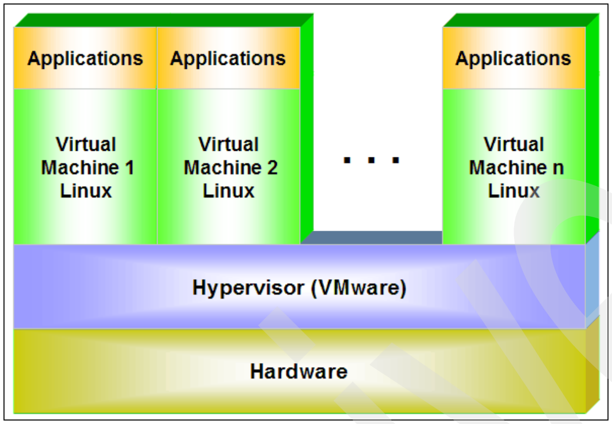



*Figure 2-18 VMware ESX*

VMware can virtualize any x86 operating system. For supported operating systems, check Table 2-1 on page 14.

## 2.3.1 vCenter

VMware vCenter Server is the universal hub for virtualization management that focuses on managing pooled infrastructures instead of individual components. vCenter is designed to aggregate physical hardware (networking, storage, memory, and CPU) and manage it as a collection of resources that can be allocated dynamically on business needs.

VMware's vCenter management platform provides a proven approach to managing the virtualized datacenter, allowing you to streamline IT management and reduce operating costs.

## 2.3.2 VMotion

VMware VMotion technology enables you to move an entire running virtual machine instantaneously from one server to another. Virtual Machine File System (VMFS), VMware's cluster file system, is used to control access to a virtual machine's storage. The active memory and the current execution state of a virtual machine are transmitted from one physical server to another using a high-speed network connection. The access to the virtual machine's disk storage is switched to the new physical host. The virtual machine retains its network identity and its connections because the network is also virtualized by ESX.

VMware VMotion allows you to perform live migrations with zero downtime.

### 2.3.3 Distributed Resource Scheduler

Distributed Resource Scheduler (DRS) continuously monitors utilization across resource pools. It provides the capability to create rules and policies to prioritize how resources are allocated to virtual machines. This enables you to balance your computing capacity among the different resource pools and virtual machines. DRS capabilities ensure that each virtual machine has access to appropriate resources at any point in time. If additional resources are made available to DRS, it will take advantage of them by redistributing virtual machines without system disruption.

## 2.4 Hyper-V

Hyper-V comes with Microsoft Windows Server 2008. It is a hardware-assisted solution that provides isolated operating system environments called partitions. One so-called parent partition running Windows 2008 is required. This parent partition is also known as a root partition. From within this parent partition other partitions, named child partitions, can be started. They host the guest operating systems. Figure 2-19 shows this architecture.

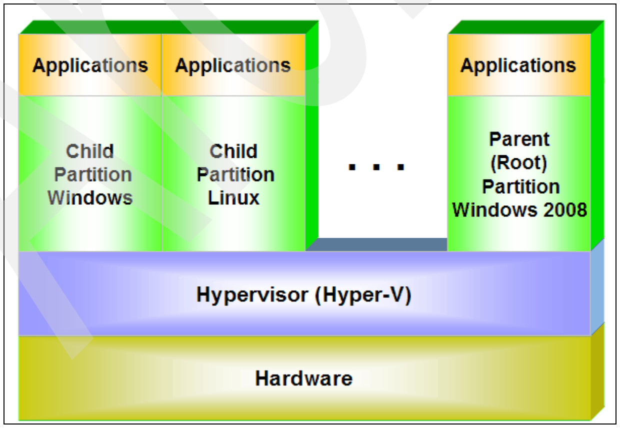



*Figure 2-19 Hyper-V*

For supported operating systems check Table 2-1 on page 14. You can retrieve more detailed information from the following Web site:

http://www.microsoft.com/windowsserver2008/en/us/hyperv-main.aspx

## 2.5 Linux Kernel-based Virtual Machine (KVM)

KVM is the new Linux Kernel-based Virtual Machine. It is a hardware-assisted virtualization solution. As the name indicates, KVM is integrated into the Linux kernel. Therefore, the Linux kernel itself becomes the hypervisor. This approach takes advantage of all the improvements made to the Linux kernel, as they become beneficial for the hypervisor as well.

The KVM virtualization solution requires processors that support virtualization for different operating systems. KVM itself is responsible for virtualizing the memory and QEMU. A processor emulator is required to virtualize the I/O. QEMU virtualizes disks, graphic adapters, network devices, and so on. A copy of it runs in each guest system. Any I/O requests that a guest operating system makes are routed to be emulated by the QEMU process.

As the hypervisor is a part of a regular Linux kernel, you can run any Linux application on the hypervisor. The guest operating systems run on the top of the hypervisor, each one in a separate process. Figure 2-20 provides an overview of the KVM architecture.

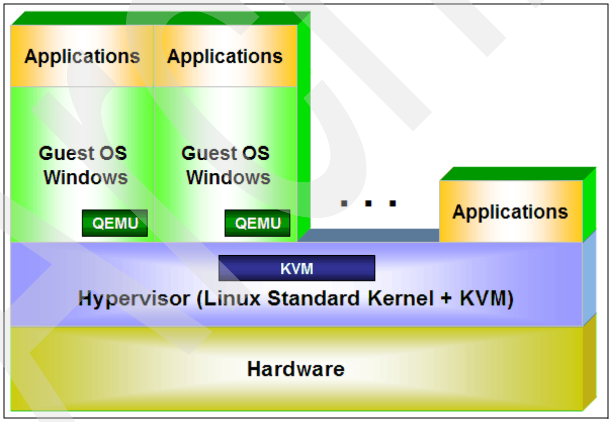



*Figure 2-20 KVM architecture*

For more information see:

http://www.linux-kvm.org/page/Main_Page

## 2.6 z/VM

The z/VM hypervisor offers a base for customers who want to exploit IBM virtualization technology on the IBM System z10 servers. It provides a full virtualization environment on IBM System z servers, allowing several guest operating systems on these servers. Besides the various System z platform operating systems, z/VM also allows Linux operating systems to run on the IBM z servers. Figure 2-21 shows these features.

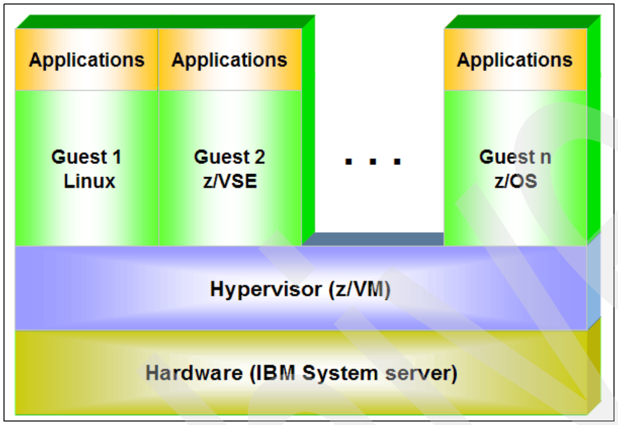



*Figure 2-21 z/VM features*

For supported operating systems check Table 2-3 on page 15. For more detailed information see:

http://www.vm.ibm.com/

## 2.7 Xen

Xen is an open-source paravirtualization product that runs on various processors such as IA-32 (x86, x86-64), IA-64, and PowerPC 970. During startup Xen boots a first guest operating system called domain 0 (dom0). This domain receives special management privileges to maintain the other guest operating systems running in domain U (domU). Figure 2-22 depicts this architecture.

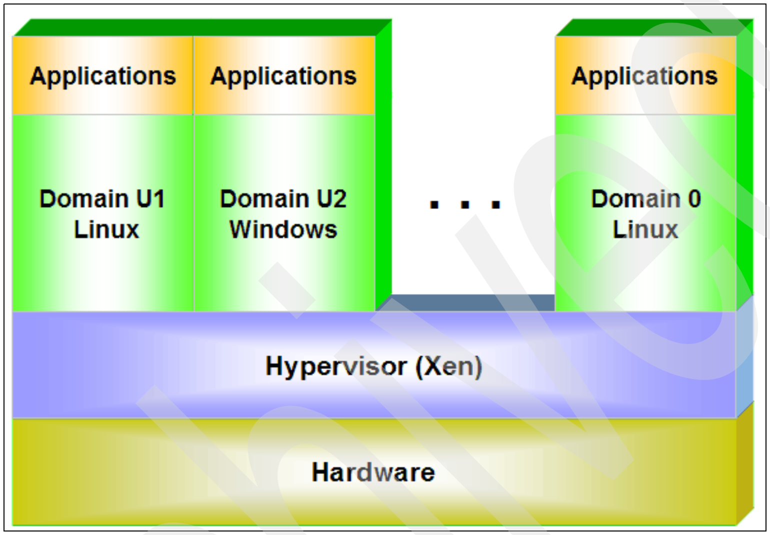



*Figure 2-22 Xen architecture*

For supported operating systems check Table 2-1 on page 14. You can find more detailed information at:

http://www.xen.org/

## 2.8 Parallels Virtuozzo Containers

Parallels Virtuozzo Containers is an operating system-level virtualization product designed for large-scale homogenous server environments and data centers. This solution is compatible with x86, x86-64, and IA-64 platforms. It creates isolated partitions or containers on a single physical server and operating system instance to utilize hardware and software. Figure 2-23 depicts the architecture of Parallels Virtuozzo Containers.

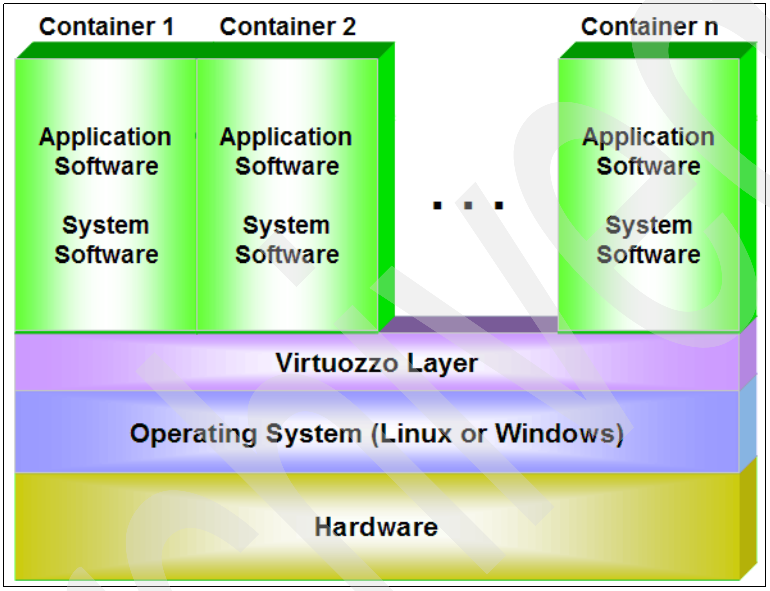



*Figure 2-23 Parallels Virtuozzo Containers architecture*

For supported operating systems check Table 2-2 on page 15. You can find more detailed information at:

http://www.parallels.com/products/virtuozzo/

## 2.9 Solaris Zones (containers)

Solaris Zones (non-global zones) are complete execution environments for applications within a single Solaris instance called the global zone. A zone allows application components to be isolated from each other by mapping system resources to non-global zone interfaces. The zone definition establishes boundaries for resource consumption, as well as providing isolation from other zones on the same system. Figure 2-24 depicts the virtualization approach of Solaris Zones.

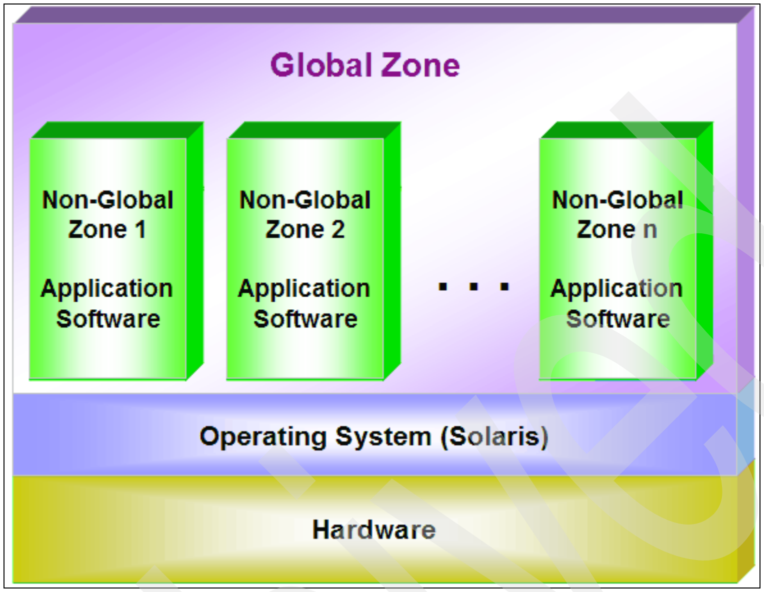



*Figure 2-24 Solaris Zones*

For supported operating systems check Table 2-4 on page 15. To read more about this topic see:

http://www.sun.com/software/solaris/containers_learning_center.jsp

## 2.10 HP Integrity VM

HP Integrity Virtual Machines (Integrity VM) is a soft-partitioning and virtualization technology that enables you to create multiple software-controlled Itanium-based virtual machines within a single HP Integrity server or nPartition. The Integrity server or nPartition acts as a VM Host for the virtual machines. (Virtual machines are also called guests.) The VM Host is a platform manager. It manages hardware resources such as memory, CPU allocation, and I/O devices, and shares them among multiple virtual machines. The VM Host runs a version of the HP-UX operating system and can be managed using standard HP-UX management tools.

The virtual machines share a single set of physical hardware resources, yet each virtual machine is a complete environment in itself and runs its own instance of an operating system (called a guest OS). As with a real machine, the virtual machine contains:

- At least one processor core, also referred to as a virtual CPU or vCPU
- Memory
- Disks
- Networking cards
- A keyboard
- A console
- Other components of a computer

All these elements are virtual, meaning that they are at least partially emulated in software rather than fully implemented in hardware. However, to the guest OS they appear as though they are real physical components.

No guest OS can access memory allocated to another guest OS. One virtual machine is not affected by software events on another virtual machine, such as faults or planned software downtimes. Integrity VM optimizes the utilization of hardware resources, quickly allocating resources such as processor cores, memory, or I/O bandwidth to the virtual machines as needed. Any software that runs on supported versions of HP-UX can run in an Integrity VM virtual machine. No recompiling, recertification, or changes are required for applications to run in a guest OS. Applications run in the guest OS as they do on any operating system.

Figure 2-25 depicts the virtualization architecture of the HP Integrity VM.

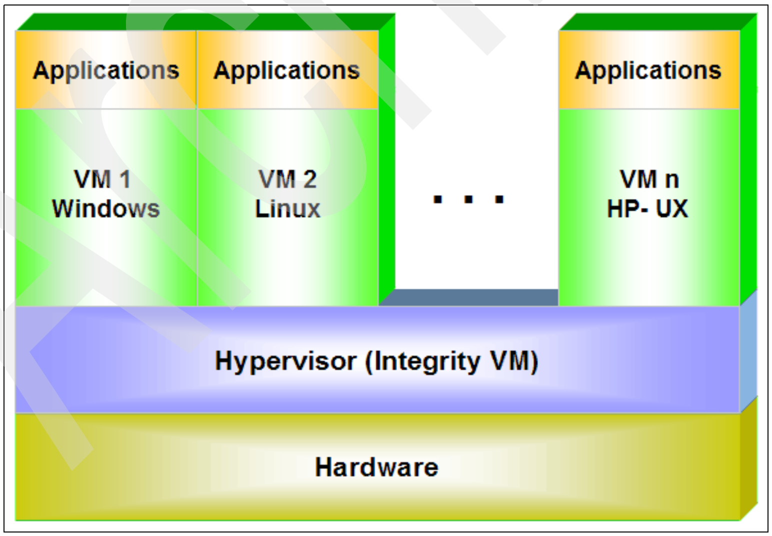



*Figure 2-25 HP Integrity VM*

For supported operating systems check Table 2-3 on page 15. More information can be found at:

http://docs.hp.com/en/

3

# Power Systems and PowerVM

IBM Power Systems offers well-developed and full-grown virtualization capabilities, which have been used together with DB2 for a long time. In fact, since the operating systems running on Power Systems have been virtualized for quite a while, most of the database administrators working with DB2 running on Power Systems are so used to the virtualization that they do not pay any attention to it.

Virtualization on Power Systems is implemented through the PowerVM virtualization technology. PowerVM is the combination of hardware and software technologies that together provide the virtualization capabilities. It provides a set of rich features that include maintenance tools to set up, maintain, and monitor virtualized Power Systems environments.

In this chapter we discuss the following topics:

- PowerVM virtualization terminology, capabilities, features, and architecture
- How to set up Power Systems and PowerVM optimized for different DB2 workload scenarios
- How to set up and configure DB2 to get the most out of the virtualized environment on Power Systems and PowerVM

## 3.1 Architecture

PowerVM architecture consists of the following main components and features:

- POWER Hypervisor (PHYP)
- Hardware Management Console (HMC)
- Integrated Virtualization Manager (IVM)
- Logical partition (LPAR)
- Dynamic logical partitioning
- Virtual I/O Server (VIOS)
- Live Partition Mobility (LPM)
- Workload partition (WPAR)

In the following subsections we explain these components and the features related to each of them.

### 3.1.1 POWER Hypervisor

A hypervisor acts as a layer between the physical server and all of the virtual servers running on top of it. It is required in order to run one or more virtualized machines on a hardware set. The POWER Hypervisor is integrated in all current Power Systems servers as part of the system firmware. Since the introduction of POWER5, the POWER Hypervisor is active on all Power Systems servers. It is not possible to run a server on POWER5 or POWER6 without it being virtualized.

### 3.1.2 Hardware Management Console

The Hardware Management Console is the primary hardware management solution for Power Systems servers. With one HMC you can manage multiple POWER5 and POWER6 servers. The HMC is a dedicated server running on separate hardware. Since Version 7, the HMC user interface has been Web based.

The HMC is the focal management point for your Power Systems server environment and hardware. It communicates to the servers using a standard Ethernet connection to the service processor of each Power Systems server. You can utilize the HMC to initially create your logical partitions, as well as change parameters and assigned resources afterwards.

### 3.1.3 Integrated Virtualization Manager

The Integrated Virtualization Manager is, just like the HMC, used to initially create and maintain logical partitions. However, the IVM performs a subset of the

HMC for only a single Power Systems server. The IVM is integrated within the Virtual I/O Server product.

### 3.1.4 Logical partition

A logical partition can be seen as a virtual machine itself. The entire Power Systems server can be divided into one or more LPARs, where each can run its own operating system (OS). These operating systems are isolated from each other just like normal physical servers. You are able to interact with the LPARs just as you are able to interact with any physical server.

When dividing your Power Systems server into LPARs you can choose from two different partitioning methods, depending on how you decide to share your processor capacity between the LPARs. These two partitioning methods are:

- Shared processor partition
- Dedicated processor partition

Here we discuss both of the partition types and their capabilities and features in more detail.

#### Shared processor partition

Shared processor partitioning has been available since POWER5 was introduced. This method implements micro-partitioning, where a fraction of the CPU attached to the system can be assigned to the LPAR. You are able to slice your CPU into units of one-hundredth (1/100th) of the CPU capacity, but the minimum amount of physical CPU given to a specific LPAR is one-tenth (1/10th). For example, by using shared processor partitioning, you are able to partition your 8-way Power Systems server to as many as 80 partitions, although it might not be the most efficient partitioning plan to have such a large amount of small partitions.

When using micro-partitions you can define how much CPU capacity your LPAR is entitled to use. You can think of your Power Systems CPUs as a pool from which CPU capacity will be assigned to the shared processor partitions. Remember that you are able to assign the CPU resources from this pool using fractions of a CPU and not just entire CPUs. To define the partitions, you must initially set the minimum processing units, the desired processing units, and the maximum processing units:

- Minimum processing units

  This is the minimum CPU resource units required to activate a LPAR. If the available CPU resource units do not meet the number defined in this parameter, the partition will not be activated. When dynamically changing the number of virtual processors assigned to the system, this parameter sets the lower limit.

- Desired processing units

  With this you define the value that you want to give to a partition. As long as there are enough CPU resource units to meet the minimum processing unit value, the partition will be activated even if there are not enough CPU resource units available to satisfy the number defined in this parameter. When there are enough free processing units to fulfill what was defined in desired processing units, this will be guaranteed for a logical partition. You are able to change this value while the system is online. This value is shown as the entitled capacity at the OS level.

- Maximum processing units

  This value defines the maximum processing units available for a logical partition when you want to dynamically increase the amount of the processing units for this partition. This parameter does not define how much processing capacity an uncapped partition can use. Uncapped partitions can temporarily exceed the number of processing units defined in this parameter.

There are also two modes for shared processor partitions:

- Capped

  With this method your CPU resources for the logical partition are limited to the maximum processing units value defined for the logical partition. The partition cannot go beyond the limit defined in the maximum processing units at any time without changing its entitled capacity, either dynamically or by changing the partition profile.

- Uncapped

  With this method, when there are free resources available in the shared processor pool, your CPU resources for a logical partition can go beyond the value defined in the desired processing units up to the number of virtual processors.

  If there is contention with the CPU resources from the shared processor pool between logical partitions defined as uncapped, the capacity will be shared among the partitions based on their initially defined uncapped weight value. This value can range from 0 to 255, where 0 is the lowest priority and 255 is the highest priority. The default value for the uncapped weight is 128.

#### *Shared processor pool*

Since POWER6, shared processor partitions can share more than one shared processor capacity pool. For the examples in this book, we use only one shared processor pool. There is always the default shared processor pool called

Shared-Processor Pool$_0$. Besides the default shared processor pool, you can define as many as 63 additional multiple shared processor pools.

### *Virtual processors*

A virtual processor represents a single physical processor to the guest operating systems running on the logical partition. One physical processor can be divided into up to 10 virtual processors. When you define a profile for a logical partition, you must set three parameters related to the virtual processor:

- Minimum virtual processors

  This parameter defines the minimum number of virtual processors that must be available for this partition when it is activated. When dynamically changing the number of virtual processors assigned to the partition, this parameter sets the lower limit.

- Desired virtual processors

  This is the desired virtual processor number for a LPAR. When a LPAR is activated the hypervisor will try to assign the number of virtual processors to this value. You can set this parameter to any value between the minimum number of virtual processors and the maximum number of virtual processors and change it while the system is online.

- Maximum virtual processors

  This parameter sets the upper boundary of the number of virtual processors assignable for a logical partition.

## Dedicated processor partition

For a dedicated processor partition you are able to assign one or more dedicated processors to an LPAR. For instance, if your server has eight processors, you can partition it to four 2-way LPARs, each having two dedicated physical processors. Since POWER6, you are also able to share the unused CPU cycles of the dedicated processor partitions with the shared processor partitions. In this way you can ensure maximum usage and capacity for your Power Systems servers and LPARs.

For the shared processor partitions, you must define minimum, desired, and maximum values for the processor and memory capacity. For the dedicated processor partitions, there is always a guaranteed amount of dedicated processor capacity that is mapped to actual physical processors. This is why you do not define virtual processors for the dedicated processor partition. The number of desired processors is shown as the entitled capacity in the OS level.

### Partitioning example

You can combine the two partitioning methods, shared processor partitions and dedicated processor partitions, on one Power Systems server. For instance, you can have a system with eight physical processors partitioned into three dedicated processor partitions and three shared processor partitions. For the three dedicated processor partitions, two LPARs have a dedicated physical processor each and one LPAR has two dedicated physical processors. The other four physical processors are assigned to the shared processor partitions.

Figure 3-1 illustrates this example of partitioning a Power Systems server by using dedicated and shared processor partitions.

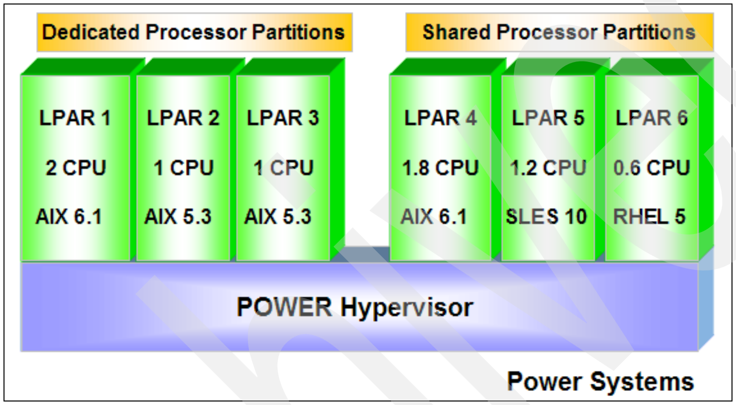



*Figure 3-1 LPAR with dedicated and shared processor partitions*

In this example the shared processor partitions are micro-partitioned so that LPAR 4 has 1.8 times one physical CPU capacity, LPAR 5 has 1.2 times one physical CPU capacity, and the remaining LPAR 6 has 0.6 times one physical CPU capacity. This leaves us free capacity of 0.4 CPUs to be used for uncapped mode partitions in the shared processor pool. If we have configured all our LPARs in the shared processor pool to operate in uncapped mode, this would mean that the remaining extra capacity will be shared on an as-needed basis between the LPARs based on their uncapped weight value, which is the same for all by default.

To illustrate the concept of virtual processor, in this example we can configure LPAR 4 to have four virtual processors, LPAR 5 to have two virtual processors, and LPAR 6 to have two virtual processors, even though the physical CPU assigned to these LPARs is just a fraction of the total four shared processors. The number of virtual processors is not necessarily related to the actual CPU capacity that the LPAR is assigned to use. Later in this chapter we go through the best practices for how to define the virtual processors for the LPARs on the shared processor pool.

### 3.1.5 Dynamic logical partitioning

With the dynamic logical partitioning (DLPAR) feature you can change assigned resources for the logical partitions without rebooting the operating system running on a partition. This means that you are able to change the amount of memory or processing units for your LPAR without a service break. This feature is very useful. For instance, you can add additional CPU resources to a logical partition that is temporarily running out of CPU resource and remove the resource once it is no longer needed.

When you define values for the maximum processing units or for the maximum memory, keep in mind that these values define the upper limit for how far you can dynamically add resources to the logical partition without rebooting your partition. However, too large of a value for the maximum memory parameter causes the hypervisor to consume an unnecessary amount of memory.

The concept of setting the limits for the processing units on a LPAR can be applied to the memory. There are minimum, desired, and maximum values for memory when initially setting up a LPAR or when changing its profile. These parameters define the upper and lower memory limits for a specific LPAR. You can change the values by changing the profile for LPAR through the HMC or the Integrated Virtualization Manager (IVM).

### 3.1.6 Virtual I/O Server

With the VIOS you can assign physical resources such as storage devices and network devices across logical partitions. The main benefit of VIOS is that you can virtualize the physical I/O devices, allowing you to share one physical device with one or more virtual servers. You are not bound to get dedicated individual physical peripherals for the number of LPARs that you are running on your Power Systems server. This saves hardware costs and makes the maintenance easier even for a large number of servers.

VIOS is a stripped-down version of the AIX OS. Its only function is to take care of I/O virtualization for other LPARs relying on the server. Let us take a closer look at the features and the benefits of virtualizing storage and network:

- Virtual storage

  With VIOS you are able to virtualize

  - Direct-attached storage (DAS): You are able to virtualize physical storage devices as a whole. With certain restrictions, these storage devices can be moved between two VIOSs or between different LPARs.
  - Storage area network (SAN) disks: You are able to virtualize SAN disks from your centralized storage network for individual LPARs. However,

logical volumes cannot be virtualized through multiple VIO servers, making them unsuitable for configurations requiring redundancy.

   – Logical volumes: You are able to virtualize logical volumes for LPARs. The LPARs will see virtualized logical volume as physical disks. This is beneficial since it reduces the need for physical peripherals.

- Virtual network

  One of the main benefits for the virtual network is that it not only reduces the requirements for network peripherals, but also makes it easier to increase availability for the network devices used by LPARs.

  There are several methods to provide continuous network availability such as dead gateway detection (DGD), virtual IP addresses (VIPAs), and IP address takeover (IPAT). By using failover techniques and one or more virtual I/O servers, you are able to provide continuous network access to the logical client partitions. Shared Ethernet Adapter (SEA) failover features, combined with other available network failover techniques, not only provide the continuous network access, but also lessen the amount of network equipment required on the server system.

To provide continuous access to I/O resources you must have the traditional failover techniques in your architecture. These include multi-path I/O (MPIO) access to disks as well as redundant disk devices. When using VIOS this might not be enough since you also must be prepared for VIOS downtime. For instance, a planned outage for performing a VIOS software update must be allowed without causing service downtime. To achieve this goal, you can have more than one VIOS on your system. For smaller systems one virtual I/O server might be sufficient. However, for larger system setup, it is good to have two or more virtual I/O servers on one system. When Virtual I/OS servers and clients are properly configured, the clients are able to utilize peripherals from another VIOS while one is unavailable during a system upgrade.

### 3.1.7 Live Partition Mobility

The Live Partition Mobility feature was introduced on POWER6 hardware and is not available on earlier hardware releases. With LPM you are able to move your logical partitions from one hardware to another without powering down the LPAR. With this feature you can perform hardware upgrades without a service break. This greatly increases your system availability on mission-critical systems, where continuous service is needed 24x7.

Using this feature is preferred in migrating partitions. There are two types of migration:

- Inactive migration
- Active migration

LPM can be used with DB2 as well. To find more information about DB2 and LPM, refer to the IBM white paper *DB2 and Power Systems PowerVM Live Partition Mobility,* available at:

https://www-304.ibm.com/jct09002c/partnerworld/wps/servlet/ContentHandler/whitepaper/aix/v6r1_db2_powervm/live_partition?pageId=pw.technical.resources

## 3.1.8 Workload partition

WPAR is a software-based virtualization solution introduced with AIX 6.1. WPAR is supported on POWER4, POWER5, and POWER6. There are two types of WPAR:

- System WPAR
- Application WPAR

At the time that this book was written, DB2 is supported only on System WPAR.

### System WPAR

You can have multiple System WPARs running on one LPAR. Essentially, a WPAR uses OS-level virtualization, and a LPAR is based on full virtualization, so a WPAR is a virtual instance within a virtual machine. System WPARs are isolated from each other and can have their own IP addresses and file systems. Although they can share the global environment and `/usr` and `/opt` file systems, the sharing is in read-only mode. You can run one or more full operating WPAR environments with its own applications on one LPAR. With System WPAR you can, for example, run two or more instances of DB2 totally isolated from each other on different IP addresses. One instance can be used for testing while the other instance can be used for development.

### Application WPAR

Application WPAR makes it possible to isolate applications or groups of applications from each other. Applications running on different application WPARs cannot see or communicate with each other on the process level, but they do share the file systems of the global environment.

### Live Application Mobility

Live Application Mobility allows you to move your workload partitions from one system to another. You are able to migrate your applications from one LPAR to another without causing any service outage. The LPAR that you are moving your application to can be located on the same physical server or on a separate physical server.

For more information about WPAR and its features, refer to IBM Redbooks publication *Introduction to Workload Partition Management in IBM AIX Version 6.1*, SG24-7431, at:

http://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/sg247431.html

### 3.1.9 Overall architectural picture

In the example illustrated in Figure 3-1 on page 52, we only cover how the logic of virtualization on Power Systems works, but we do not include how the actual physical hardware is seen by the logical partitions. To bring together the main components of virtualization under Power Systems that we have discussed so far, Figure 3-2 illustrates how these components are virtualized.

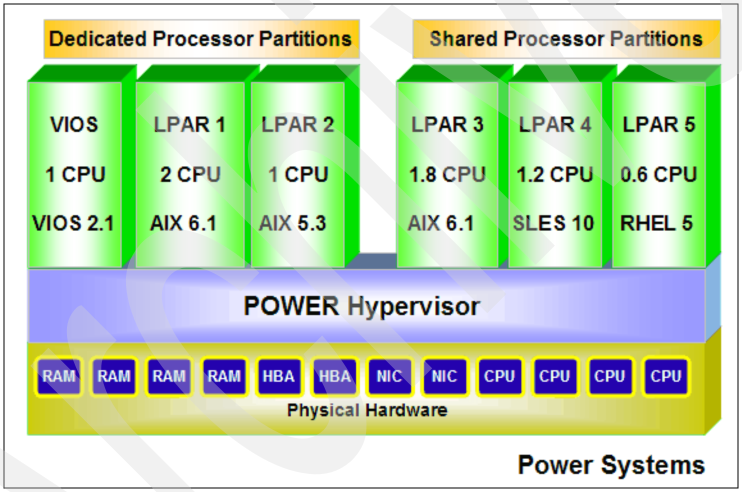



*Figure 3-2 Overall architectural picture*

The difference between Figure 3-2 and Figure 3-1 on page 52 is that we used one of the dedicated-processor LPARs as the VIOS and left only five partitions for application servers. VIOS virtualizes the underlying physical I/O peripherals for sharing among other LPARs. POWER Hypervisor is between other system resources and does the sharing and allocation of CPU and random access memory (RAM) just as it did on the previous example, although the physical peripherals were not included in Figure 3-1 on page 52. Besides RAM and CPU, we also have host bus adapters (HBAs) and network interface cards (NICs) attached to our system. Both NICs and RAM are virtualized by VIOS, and LPARs are able to utilize them just as they have their own dedicated hardware resources. However, these actual physical devices are shared among the LPARs.

## 3.2 Power Systems and PowerVM setup

When setting up a Power Systems environment, there are many options that you must consider and plan properly ahead of time in order to achieve the optimal performance goal. Especially for a database server, the database system is the primary application in the system and database configuration considerations should also be incorporated in the Power Systems planing setup. However, the Power Systems setup tasks are usually performed by the system support team without DBA's involvement, or the DBA may not have the expertise in this area. In this section we discuss the Power Systems setup, concentrating on providing useful tips for setting up a virtualized environment for DB2 running on the Power Systems server. We discuss the following items:

- Setting up and configuring logical partitions
  - Choosing a partition type
  - Configuring physical and virtual processors
  - Configuring the memory
- Considerations for disk storage and choosing from different options
- Considerations for the network and choosing from different options
- Setting up the VIOS

### Power Systems environment planning

Dynamic logical partitioning provides an easier option to fix bottlenecks on CPU or memory resources. This is achievable since the hardware can be shared with the LPAR running the database engine. On the other hand, in online transaction processing (OLTP) and especially data warehouse (DW) environments, the bottleneck is most often the I/O on the disk storage. This makes planning the disk setup on both database and system levels the most important area.

For CPU and memory capacity you should carefully investigate your workload and setup requirements. Usually, requirements for CPU and memory are relatively easy to define through the load tests. Some references can also be derived from the proof-of-concept environments as well as from the development and test systems. On partitioned environments, the memory and CPU capacity is easily changed as long as there is enough capacity on the Power Systems server to be utilized.

In partitioned environments there are a few additional settings that must be planned carefully when compared with the traditional systems running on bare metal. The additional settings to be considered are:

- Number of minimum, desired, and maximum processors
- Number of minimum, desired, and maximum virtual processors
- The amount of minimum, desired, and maximum memory
- For disk storage, choosing the storage type between different types of local storage and storage options provided by VIOS
- Deciding on either a dedicated processor partition or a shared processor partition running in capped or uncapped mode
- Choosing the network setup from a locally attached network and different options that VIOS provided

### *Choosing the LPAR type*

You must choose between a dedicated and a shared processor partition. Dedicated processor partitions can have one or more processors exclusively reserved for them. PHYP schedules the same physical cores for the dedicated processor partitions, which makes it possible for the dedicated processor partitions to benefit from a warm CPU cache. Although this could be beneficial for the database servers, the actual benefits that the database servers received usually are less than what is expected. Although you can benefit from dynamic logical partitioning and change the amount of both physical processors and memory assigned to the dedicated partition, you are not able to benefit from the shared processor pool at peak times. This is why we recommend shared processor partitions for DB2 partitions unless there is a special need for an isolated dedicated partition.

You should set the LPAR for a database server as a shared processor partition in uncapped mode to be able to utilize the shared processor pool. The uncapped shared processor pools are very effective for the workloads that vary on at different times and when workloads are unpredictable. Ensure that you tune the uncapped weight properly to prioritize the shared processor pool usage among the partitions. You can also consider putting all the DB2 server LPARs into a separate processor pool to optimize your DB2 license usage.

### *Memory settings*

When setting up the LPAR you should set the amount of desired memory the same way as you would set it for a non-virtualized server. This amount depends on the database requirements, for instance, the size of buffer pools that are required for your workload. Since the DB2 server is running on a logical partition, it is able to utilize dynamic logical partitioning. For setting the maximum memory size, you should consider setting a high number, including a prediction of future room for growth.

When setting the maximum memory size, you must consider the hypervisor overhead and your memory requirements. Here are some guidelines:

- As a minimum, round up the amount of memory to the nearest power of two. This does not cause any overhead.
- For future growth, multiply the value obtained above by two. This allows at least doubling the amount of memory without production break.

For example, if you have a partition that requires 5 GB of memory, rule 1 would give you 8 GB, and rule 2 would double that to 16 GB.

#### *Processor settings*

The same as with the memory settings, you should set the desired processor capacity based on the actual need. If your workload is going to change, set the amount of actual physical processor capacity high enough to allow changes in the future. Usually, as time progresses, the need for both memory and CPU usage will grow as the amount of data and the number of rows in the tables increase. Dynamic logical partitioning helps to provide continuous service without breaks even when the capacity demands have changed.

Since there is virtually no overhead in having a large number of maximum processing units, consider setting the maximum processing units for all partitions to the number of processors in your server.

#### *Number of virtual processors*

Since the DB2 server will be running on the same physical hardware as other servers, it is not enough to just specify the need for processor capacity for the database logical partition. You must tune the logical partitions with the consideration of the overall effectiveness on the system. By setting the number of virtual processors for your DB2 server, you can influence the total system performance as well as the performance of one specific logical partition.

By setting up the number of virtual processors, you set up how small the fractions are that are used to divide the physical processor capacity for the logical partitions. By setting this value, you set the ratio between physical and virtual processors. The number of virtual processors is the number of processors that the OS running on the logical partition will see that it has as though they are physically attached to it. The OS scheduler does not distinguish between virtual or physical processors. It schedules the workloads as though the virtual processors were physical ones, whereas PHYP can schedule physical CPU cycles the way that it sees the best. There is no straight connection between how these physical CPU cycles are divided among the LPARs and the virtual processors. However, in general, from the entire system point of view, the more virtual processors are defined, the better CPU resource utilization will be. For each individual LPAR, if the number of virtual processors defined for the LPAR is equal to or is the round-up value of the defined desired physical processor, the LPAR is guaranteed the maximum available CPU resources.

You can use following formula to derive the number of virtual processors:

```
RVP = {round up (CEC + [UW/TUW] * FPC) , round up (CEC + FPC)}
```

Where:

**RVP** Represents the rounded-up range of virtual processors.

**EC** Is the current entitled capacity.

**UW** Is the uncapped weight for the shared processor partition. The default value is 128, and the range is 0–255.

**UW** Is the total uncapped weight of all uncapped shared processor partitions.

**PC** Is the free-pool capacity (free unassigned processor capacity).

This formula gives you a range from which to choose the value for the desired virtual processors. When using this formula you should also consider the following:

- The number of the rounded-up value for virtual processors should not go beyond two virtual processors per one entitled physical processor.
- To maximize total system processor utilization, use the higher number of virtual processors derived with the formula.
- To optimize the performance of the corresponding logical partition, use the lower number of virtual processors derived from the formula.

### *Network type*

You can choose from three different types of network:

- Shared Ethernet Adapter (SEA)
- Virtual Ethernet
- Locally attached network adapter

SEA requires VIOS, and the two later options are provided directly through PHYP. The advantage of SEA is that the logical partitions can share the same physical network adapters that will decrease the required physical adapters on the system. Virtual Ethernet is a special network adapter type that does not require physical adapters at all. Instead, it is provided through PHYP as a fast way of communication between logical partitions. You can, for instance, set up

WebSphere Application Server (WAS) to connect to the DB2 server through a virtual Ethernet.

Considering the performance, there are no major differences between the various types of network options. You should choose the appropriate network type based on your hardware and overall system setup.

### *Disk I/O type*

Choosing the correct disk I/O type is the most critical decision that you must make. It effects the overall database performance significantly and is also one of the hardest to change later once the setup is complete. Though there are multiple types of options to select from, since disk I/O is critical, it limits the choices.

For the OS you should choose the disk type as you would choose any other type of logical partition. In this book we concentrate on choosing a storage option for DB2 use. Pay attention while choosing a storage option for the data and transaction logs. The same best practises for storage hold true on the virtualized environments as they do on the physical servers with a few exceptions.

First of all, you must choose between the locally attached storage and the virtual storage. Although the virtualized storage is easier to manage, it comes with a cost. There is always some overhead when storage is virtualized. On some systems and workloads this overhead may be acceptable. However, many systems and workloads require optimal disk I/O, for instance, database server.

In general, DAS is not fast enough for the database servers. SAN disk is a better choice for the database servers. SAN disks have large disk caches and can also be easily maintained. Also, they can be configured to use the available RAID levels. The SAN systems can be used through VIOS or as a local attached storage.

For optimal performance, we recommend the locally attached storage. When I/O through VIOS can be used without effecting I/O performance, this option can be chosen. Choose the VIOS-managed storage only if the I/O performance is not affected by the virtualization layer. For the reset of database storage planning, follow the best practices discussed in *Best Practises: Database Storage* at:

http://www.ibm.com/developerworks/data/bestpractices/databasestorage/

Best practises that we would like to highlight are:

- Think about real spindles, not just storage space. Most storage specialists are happy to give DBA the amount of storage space required, but not the number of spindles needed.
- Have dedicated LUNs and file systems. File systems should be created on their own LUNs, not disk partitions.
- Stripe at most in two places. For instance, if you are using multiple storage paths for your tablespace containers, you can stripe your storage on the SAN system or at the OS level, but not on both.
- Separate DB2 transaction logs and data. You must have logs and data on their own disks, not just on their own disk partitions.
- Use file systems instead of raw devices for LUNs—one file system per LUN. Raw devices might offer a slight performance boost in special cases, but the price, in addition to complicating database maintenance, is too high.
- Use the NO FILE SYSTEM CACHING clause. This option prevents double buffering and is highly recommended.
- Use DB2 automatic storage to stripe everything everywhere. Automatic storage offers optimum performance and ease of use and maintenance. In case you need more I/O, it is as simple as adding a storage path.
- Do not hand-tune the NUM_IOCLEANERS, NUM_IOSERVERS, and PREFETCHSIZE configuration parameters. With dynamic logical partitioning, setting these database parameters to AUTOMATIC is even more important than on the non-virtualized environments.

### *Virtual I/O Server*

You can have a Power Systems environment without VIOS, but this would be a special case. Most systems have at least one VIOS. To provide high availability, there must be at least two VIOSs. To achieve the most from your system, define your VIOS as an uncapped shared partition and its tune uncapped weight value properly.

For more information about setting up VIOS refer to the following resources:

- *PowerVM Virtualization on IBM System p: Introduction and Configuration Fourth Edition*, SG24-7940, available at:

  http://www.redbooks.ibm.com/abstracts/sg247940.html

- *IBM PowerVM Virtualization Managing and Monitoring*, SG24-7590, available at:

  http://www.redbooks.ibm.com/abstracts/sg247590.html

- VIOS in IBM System Information Center at:

  http://publib.boulder.ibm.com/infocenter/systems/index.jsp?topic=/com.ibm.aix.security/doc/security/capp_vios.htm&tocNode=toc:front/front.cmb/0/0/11/0/0/2/11/13/

## 3.3 DB2 setup and configuration

Now that the LPAR is set up, it is time to discuss how to set up the DB2 environment. Most DBAs may not realize that they are running on a virtualized environment when DB2 is with AIX or pLinux on a Power Systems server and thus neglect tuning and configuring DB2 to take the benefits that virtualization has offer on a Power Systems LPAR.

In this section we discuss the following topics:

- Installing and setting up DB2
  - Setting up storage
  - Setting DB2 configuration parameters
- Configuration example
- Best practices

### 3.3.1 Installing and setting up DB2

In this section we discuss DB2 installation and initial configuration. We especially concentrate on how to configure DB2 detecting the changes in a dynamic Power Systems environment.

#### DB2 installation

Install DB2 on Power Systems logical partition as you would install it on a non-virtualized environment. On AIX, DB2 is installed under the `/opt` file system, which is usually on its own logical volume under the *rootvg* volume group. On pLinux there are no differences either. DB2 is installed under the `/opt` file system on its own subdirectory with no changes to the default for the installation. Virtualization does not effect the installation type or the options.

### Setting up DB2

DB2 will benefit from the autonomic features, Self Tuning Memory Manager (STMM) and automatic storage, that have evolved especially in the DB2 Version 9 family. For instance, when dynamically changing processor or memory for a logical partition, DB2 is able to detect the changes dynamically online without the need to restart the instance. DB2 will make the necessary adjustments for the database heap and the buffer pools as well as for instance I/O servers if this will improve DB2 performance and increase transaction throughput.

#### *Setting up storage*

You should set up your transaction logs and data on separate file systems and, ideally, on external storage systems on their own dedicated LUNs. In our examples we set up transaction logs under the `/db2logs` directory and data under `/db2data` on different subdirectories. These file systems are located in a SAN system on their own dedicated LUNs. We defined our database paths as shown in Example 3-1 using the database creation clause.

*Example 3-1 Database creation clause*

```
CREATE DATABASE virtdb AUTOMATIC STORAGE YES ON /db2data/data1, /db2data/data2,
/db2data/data3, /db2data/data4 DBPATH ON /db2data/data1
```

As you see from the above example, we use automatic storage. This significantly reduces time spent on maintenance when there is a need to add storage paths to our database. Automatic storage provides optimal performance as well, so it is an ideal choice for storage type.

When you need more storage space or I/O you will need to provide additional LUNs from your SAN system and then make them available to your OS. After this you will be able to add the new storage path for your database environment. This increase in storage paths for your database table spaces will not only increase the amount of storage, but I/O capacity as well.

Example 3-2 shows the command for adding two additional storage paths to our database. Ideally, each storage path should be equal in size and lay on its own file system under dedicated LUNs.

*Example 3-2 Adding storage path*

```
ALTER DB virtdb ADD STORAGE PATH /db2data/data5, /db2data/data6
```

When the I/O capacity has increased, you may need more CPU capacity in order to handle the increased I/O. For an uncapped shared processor logical partition this would not be a problem as long as there is enough CPU capacity in the shared processor pool. If the increase in the needed CPU capacity will be permanent, you should consider increasing the entitled processor capacity for the database logical partition. With dynamic logical partitioning this can be achieved without any effect to service. You just must change the value of the desired processor units for the LPAR.

As discussed earlier, for a Power Systems server environment you must choose between the virtualized storage and the locally attached storage. Although the virtualized storage provides easier maintenance, it has a slight overhead. On most systems this overhead is not noticeable, so we are able to take advantage of the benefits for easier maintenance when using virtual storage. VIOS makes it easier to add more storage to your system without a service break. With more than one virtual server, you are always able to add physical adapters and SAN systems to your environment without service breaks on your production environment. This ability to provide additional storage and I/O resources without service downtime, combined with the ease of use and maintenance of DB2 automatic storage, makes Power Systems environments ideal for dynamic computing environments with dynamic resource needs.

### *Setting DB2 configuration parameters*

Take advantage of the DB2 9 autonomic computing features on your virtualized environment to be able to optimize the system performance with dynamic changes. By utilizing DB2 autonomics, your system is capable of adopting changes in the computing environment automatically. For instance, over time the number of users accessing the system might increase. On an OLTP system, this is usually most noticeable through the increase in reads and writes in the database usage. When the amount of data and the number of user accesses grow on the system, you might need larger buffer pools and more disk space and disk I/O. We have discussed how to handle the demand changing for storage and I/O. In this section we concentrate on providing information about how to benefit from DB2 autonomic features combined with Power Systems dynamic capabilities.

You should check that the configuration parameters on both the instance and the database level are set properly so that DB2 can automatically reallocate the resources such as heaps when there are changes in system resources. Since the processing capabilities of a uncapped shared processor partition will change dynamically depending on the need for the computing resources, you should pay special attention to those CPU-bound configuration parameters.

Buffer pools and other memory-related DB2 parameters are automatically re-sized by DB2 when you have STMM enabled and when you have set the buffer pool sizes as AUTOMATIC. This makes it possible to dynamically add system memory to the database LPAR. DB2 will notice this change in memory resources and tune the database memory, the buffer pools, and other memory-related parameters correspondingly.

Follow general guidelines and best practices about on how to tune your DB2 configuration parameters.

## 3.3.2 Configuration example

To show you how the DB2 autonomic computing features work in a virtualized environment, we build a test environment. The tests that we run on our test environment are not meant to be an example of how to build or set up a real-life production environment. The purpose of setting up the environment is to show how the autonomic DB2 features react to the dynamic changes in a virtualized environment.

### Setting up Power Systems and LPARs

We set up a Power Systems environment on an 8-way POWER5 server with 16 GB memory, as illustrated in Figure 3-3.

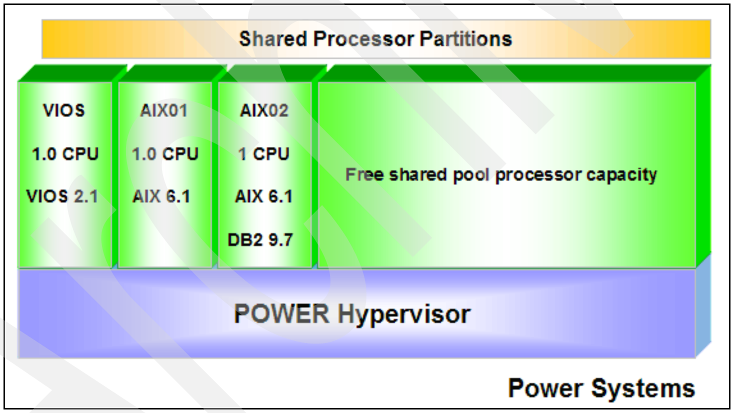



*Figure 3-3 Test environment setup*

We create a very basic environment with only three LPARs. One of the LPARs is used as the VIOS. All of our LPARs are shared processor partitions in uncapped mode. Table 3-1 shows the memory and CPU settings.

*Table 3-1 LPAR setup*

| System name | Minimum CPU | Desired CPU | Maximum CPU | Minimum memory | Desired memory | Maximum memory |
|---|---|---|---|---|---|---|
| VIOS | 0.5 | 0.4 | 1.0 | 512 MB | 1.0 GB | 2.0 GB |
| AIX01 | 0.5 | 1.0 | 2.0 | 1.0 GB | 1.0 GB | 2.0 GB |
| AIX02 | 0.5 | 1.0 | 6.0 | 1.0 GB | 1.0 GB | 8.0 GB |

## Storage setup

We run all of our tests with AIX02 running DB2 9.7 Enterprise Server Edition. We configure the database to use automatic storage on two file systems located on the external storage device locally attached to our system. Example 3-3 shows the file system setup—two file systems capable of storing 20 GB data together and one 10 GB file system for the database transaction logs.

*Example 3-3 Storage setup for database data and logs*

```
$ df -g | egrep "Files|data|logs"
Filesystem    GB blocks      Free %Used    Iused %Iused Mounted on
/dev/datalv1      10.00     10.00    1%        4     1% /data/data1
/dev/datalv2      10.00     10.00    1%        4     1% /data/data2
/dev/logslv       10.00     10.00    1%        4     1% /data/logs
```

To create the database on the newly created file systems we use the command shown in Example 3-4.

*Example 3-4 Creating the database*

```
CREATE DATABASE testdb AUTOMATIC STORAGE YES ON /data/data1, /data/data2 DBPATH
ON /data/data1 USING CODESET ISO8859-1 TERRITORY US COLLATE USING IDENTITY;
```

Next we create the tables under USERSPACE1 and populate them with test data. All the tables are created with compression enabled by using the COMPRESS YES option. This not only saves our storage costs, but also keeps the requirements for I/O and memory smaller due to the row compression. Note that, in general, it is not a good idea to create all the tables with the COMPRESS YES option, unless you know that all the tables will benefit from the compression. We are sure that all our test tables can benefit from the compression. Therefore, we use the COMPRESS YES option in all of them.

## DB2 configuration

When creating a database on DB 9.7, the autonomic features, STMM and automatic storage feature, are enabled by default unless explicitly defined otherwise. It is a good practice to verify these settings, especially for the migrated databases. Table 3-2 lists how our database, TESTDB, parameters are set.

*Table 3-2 Data parameters for the TESTDB database*

| Parameter no. | Parameter name | Value |
|---|---|---|
| 1. | SELF_TUNING_MEM | ON |
| 2. | DATABASE_MEMORY | AUTOMATIC |
| 3. | LOCKLIST | AUTOMATIC |
| 4. | MAXLOCKS | AUTOMATIC |
| 5. | PCKCACHESZ | AUTOMATIC |
| 6. | SHEAPTHRES_SHR | AUTOMATIC |
| 7. | SORTHEAP | AUTOMATIC |
| 8. | DBHEAP | AUTOMATIC |
| 9. | STMTHEAP | AUTOMATIC |
| 10. | APPLHEAPSZ | AUTOMATIC |
| 11. | APPL_MEMORY | AUTOMATIC |
| 12. | STAT_HEAP_SZ | AUTOMATIC |
| 13. | NUM_IOCLEANERS | AUTOMATIC |
| 14. | NUM_IOSERVERS | AUTOMATIC |
| 15. | DFT_PREFETCH_SZ | AUTOMATIC |
| 16. | MAXAPPLS | AUTOMATIC |
| 17. | AVG_APPLS | AUTOMATIC |

Parameters 1–12 are memory related and have a dependency on how much the physical memory on the LPAR is available for the database. Parameters 13 to 15 will be effected by changes in the processing capacity. The remaining parameters, 16 and 17, will be calculated based on both memory and processing capacity.

To simplify the test environment, we have only one user table space for all tables. This table space has one buffer pool, IBMDEFAULTBP. We verify the buffer pool setting to ensure that DB2 STMM can dynamically adjust the buffer pool when the memory requirements or available memory changes. Example 3-5 shows that the buffer pool size is set to -2, which stands for AUTOMATIC. On DB2 9.7, this is the default setting.

*Example 3-5 Checking buffer pool size*

```
$ db2 "SELECT SUBSTR(BPNAME,1,12) AS BUFFERPOOL, NPAGES AS SIZE FROM
SYSCAT.BUFFERPOOLS"

BUFFERPOOL   SIZE
------------ -----------
IBMDEFAULTBP          -2

  1 record(s) selected.
```

## Running the test

Now our test database setup is complete. We run three tests to check how dynamic changes in memory capacity affect DB2 performance. We begin with only 1 GB memory. This amount of memory is not much nowadays, but it will be interesting to see how STMM behaves with this little memory. We run our tests that simulate an OLTP workload for four and half hours. This is the first point to check how the system is performing. We check the following values:

- Computed size for the following database configuration parameters:
  - DATABASE_MEMORY
  - LOCKLIST
  - SORTHEAP
  - DBHEAP
- Statistics and values for the buffer pool, IBMDEFAULTBP:
  - Computed size for number of pages
  - Index and data hit ratio
- Average OS CPU utilization including:
  - User CPU
  - System CPU
  - I/O wait

We continue on increasing the amount of memory from 1 GB to 2 GB and keep the OLTP load test running for the next four hours, which is our next check point. The last step is to double the memory dynamically once again and record the final numbers.

After we started the test, we did not change anything on DB2. STMM took care of all the tuning. Figure 3-4 shows the database performance changes during the test. The transactions per second (TPS) increased significantly during the test. The average TPS was 80 with 1 GB memory. By increasing the memory to 2 GB we received a 91% increase in average TPS. When doubling the memory once again, we received another 8% increase on average TPS. The later number could be better though since the increase in memory was noticeable.

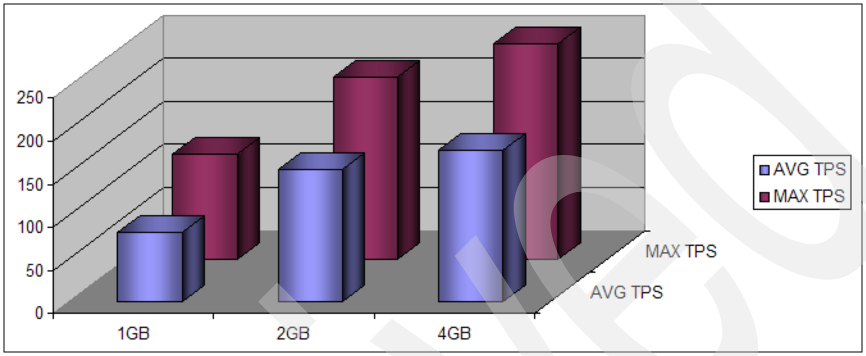



*Figure 3-4 Increase in transaction rate per second*

Let us see how we do with other numbers. Figure 3-5 shows the changes in buffer pool size, hitratio for IBMDEFAULTBP, and other memory-related DB2 parameters. At the start of the test the hitratio for buffer pool IBMDEFAULTBP was 81.28%. When changing the memory from 2 GB to
4 GB we did not get a better hitratio. It actually went down a bit from 90.59% to 90.47%. There are also no significant changes in DB2 parameters that were set to be taken care of by STMM.

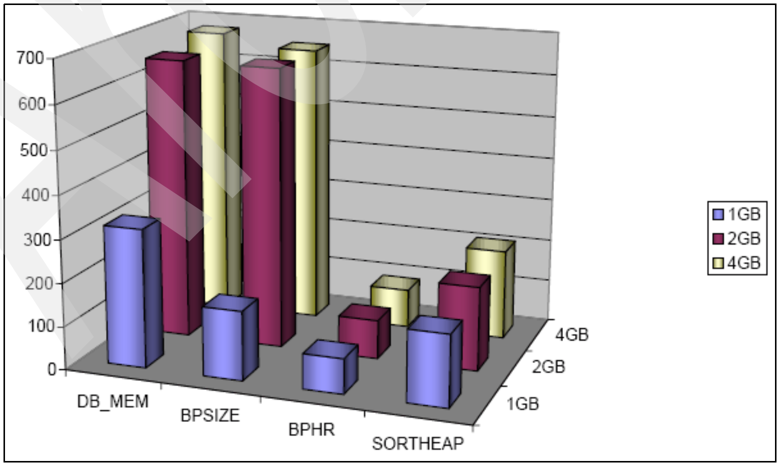



*Figure 3-5 DB2 memory-related parameters*

Now we check the system performance in total. When the memory is increased from 2 GB to 4 GB, we obtain an additional good 8% TPS increase with the fine-tuning automatically performed by DB2. The load test used to test the system performance reads data extensively. Since there was no increase in buffer pool hitratio, it implies that it has reached the optimum level. On the other hand, 90% buffer pool hitratio implies that we must read from disk 10% of all the data that we are accessing.

Figure 3-6 reveals where is our bottle neck. Although we have been able to utilize more CPU while increasing the memory, we are still waiting on I/O a lot. This I/O wait did not significantly go any lower when memory was increased. This was expected, since the load test that we use reads data extensively. To get better performance, we could add storage paths on the dedicated LUNs to our database. This would increase the I/O, but only if the amount of data would increase enough over time to start spreading into the newly created storage path containers. Just adding storage paths to the system that already has slow I/O does not necessarily make the system any faster. Adding storage paths will help on the system where I/O is becoming the bottleneck due to the increase in data. In our case, the easiest to resolve the I/O wait issue would be to recreate the database with more storage paths on dedicated LUNs with enough I/O capacity.

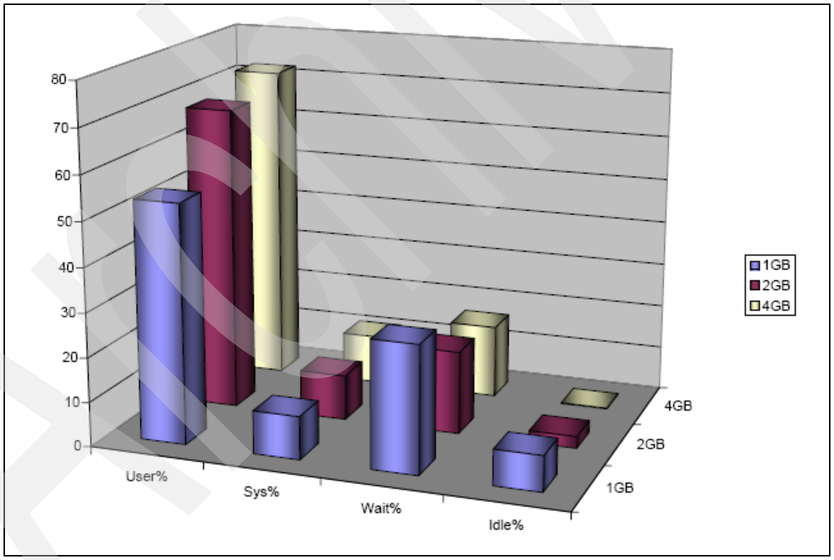



*Figure 3-6 Operating system utilization*

Looking at our physical CPU usage, we notice that we are using at maximum of 60% of our entitled capacity. Since we have a total of five unused processors in our shared processor pool, we should reach a much higher CPU usage rate than the entitled capacity, but not until the current I/O bottle neck is removed.

When building our test system we used two file systems for data. These file systems are not located on a dedicated LUN of their own. Instead, the file systems share the same LUN, which partially explains the I/O bound result. If this system was supposed to go into production we would need to get a more efficient I/O solution.

The test results were extremely encouraging considering how much of a performance increase we were able to achieve by letting DB2 take care of the tuning. Even when DB2 just slightly fine-tuned the memory usage, we were able to achieve 8% performance increase without an manual effort. This achievement resulted from relatively small changes in memory parameters by DB2. An 8% performance increase would be a very challenging task even for a very experienced DBA, especially under the condition that the overall DB2 memory usage is practically not increased at all.

By utilizing Power Systems dynamic logical partitioning capabilities with DB2 autonomics, we were able to achieve significant performance increase in just a few hours by letting DB2 take care of tuning. The overall performance increase was 119%, which is not bad at all.

## 3.4 Conclusions

By setting up a test environment, running tests, and following the DB2 best practises, we have come to the following conclusions:

- You should place the DB2 server in an uncapped shared processor partition. This gives you the ability to fully utilize system resources including usage peaks and unpredictable workloads.
- For maximum performance, use fast SAN disks attached locally to the LPAR. Use dedicated LUNs for the file systems. Utilize the DB2 automatic storage feature to stripe over the file systems. As observed in our tests, I/O can become the bottleneck very easily. Remember that whatever choice you make for your I/O and storage during the planning phase, it is the hardest to change later.
- Use DB2 autonomics to fully utilize dynamic changes in the Power Systems environment. To be able to detect the changes in a dynamic operating environment and act as required, use DB2 STMM and automatic storage features. This will guarantee that DB2 will tune itself to reflect the changes that it detects on both workloads and the operating environment.
- For the network interface choose among SEA, Virtual Ethernet, and locally attached network adapter. There is no benefit, performance-wise, to attach network adapters locally. You can select any virtualized network solution or use locally attached network adapters.

- Utilize VIOS for easier maintenance and efficiency. Although you might need to use locally attached storage, you are still able to benefit from virtualized I/O. To be able to fully maximize the system utilization, install your VIOS as an uncapped shared processor partition and tune uncapped weight properly.

# 4

# VMware vSphere

VMware is the market leader in virtualization technologies for x86 platforms. VMware vSphere is a cloud operating system. DB2 is capable of utilizing and taking full benefit of the virtualization features that VMware virtualization solutions have to offer.

In this chapter we discuss VMware virtualization technologies and features. We discuss how to set up and build a VMware environment including VMware ESX/ESXi Servers, vCenter Servers, vSphere Client, and virtual machines running DB2. We provide useful information about how to set up your DB2 to get most out of a VMware virtualized environment. We focus on the following topics:

- VMware architecture
- Setting up the VMware virtualized environment (vSphere)
- VMware virtual machine installation
- Configuring DB2 for VMware

# 4.1 Architecture

In this section we describe the high-level architecture of VMware vSphere and some of its components. We focus on the most essential components for setting up a virtualized environment for DB2.

## 4.1.1 vSphere

VMware vSphere is a suite of virtualization applications. vSphere provides virtualization technologies for running multiple operating systems on a single physical machine simultaneously. It can reclaim idle resources and balance workloads across multiple physical machines. vSphere is able to move a running virtual machine to another virtual machine without a service break. This feature provides constant system availability even when a hardware failure is encountered. It is also helpful for scheduled maintenance.

VMware vSphere includes the following major components:

- ESX
- VMware file system (VMFS)
- vCenter
- Distributed Resource Scheduler (DRS)
- VMware High Availability (HA)
- VMware Fault Tolerance
- VMotion

These major components enable vSphere to provide both infrastructure services and application services. Figure 4-1 illustrates an overall VMware architecture and how vSphere fits into it. The infrastructure services of VMware vSphere offer the manageability for the underlying hardware. These services virtualize the computing, storage, and network devices. The application services of VMware vSphere include services for high availability, security, and scalability in the virtualized environments. The entire vSphere environment is controlled by the VMware vCenter suite. This is a management suite that can easily administer even the very large virtualized environments.

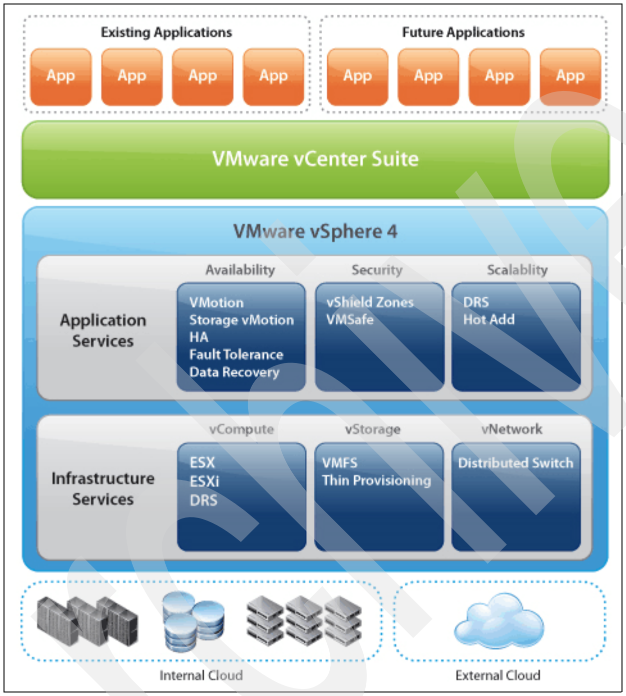



*Figure 4-1 vSphere architecture (Copyright © 2009 VMware, Inc. All rights reserved.)*

## 4.1.2 ESX

ESX uses a bare-metal hypervisor that allows multiple virtual machines to run simultaneously on the same physical server. The ESX hypervisor called VMkernel is a proprietary, kernel-optimized high-performance component for running virtual machines. The ESX service console is used only to provide a management interface to the VMkernel. The service console is a cut-down, secure version of Linux. The service console is scheduled to use hardware resources in a manner similar to normal virtual machines.

ESX 4 and ESXi are both bare-metal hypervisors that partition physical servers into multiple virtual machines. The difference is that ESXi does not use a Linux service console as a management interface. ESXi is managed by VMware vCenter, using an embedded hardware agent. Figure 4-2 depicts this architecture.

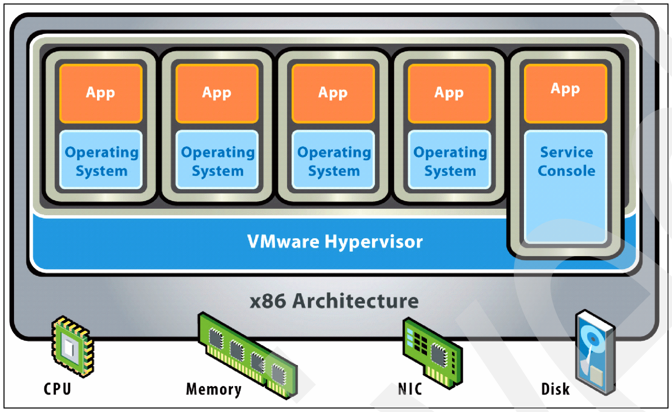


*Figure 4-2 ESX architecture (Copyright © 2009 VMware, Inc. All rights reserved.)*

All hardware components are virtualized and are under control of ESX. This allows VMware to provide advanced resource management features for ESX. The important features regarding DB2 are:

- Resource management for virtual machines allows establishment of minimum, maximum, and proportional resource shares for CPU, memory, disk, and network bandwidth. Modifications are allowed while virtual machines are running.
- Intelligent CPU virtualization provides a mechanism to execute virtual machine processes with intelligent process scheduling and load balancing across all available CPUs on the physical host.
- Network traffic shaping ensures that critical virtual machines receive priority access to network bandwidth. Network traffic from virtual machines can be prioritized.
- Storage I/O traffic prioritization ensures that critical virtual machines receive priority access to storage devices by prioritizing I/O traffic.

## System requirements

ESX is compatible with a variety of systems, I/O devices, operating systems, and storage arrays. For the information about the supported systems for different versions of ESX, refer to:

http://www.vmware.com/resources/compatibility/search.php

## Supported operating systems

VMware supports many different Microsoft Windows versions and Linux implementations. DB2 also supports different versions for both types of operating systems. To build a DB2 sever on a VMware virtualized environment, an operating system version supported by both DB2 and ESX is required. To find the operating systems supported by both DB2 and ESX, refer to the following Web sites:

- DB2 and virtualization: supported environments

  http://www.ibm.com/developerworks/wikis/display/im/DB2%20Virtualization%20Support

- VMware: supported operating systems

  http://www.vmware.com/pdf/GuestOS_guide.pdf

- DB2 9.7 for Linux: supported environments

  http://www.ibm.com/developerworks/wikis/display/im/DB2+9.7+for+Linux+-+Supported+Environments

- Installation requirements for DB2 database products

  http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.qb.server.doc/doc/r0025127.html

## 4.1.3 VMFS

VMware vStorage is a cluster file system that allows multiple instances of ESX to read and write to the same storage, concurrently. By storing the entire virtual machine state in a central location, VMFS enables you to simplify virtual machine provisioning and administration. Through these capabilities VMFS provides easy support for live migration of running virtual machines or automatic restart of a failed virtual machine on a separate physical server. Figure 4-3 depicts the VMFS usage.

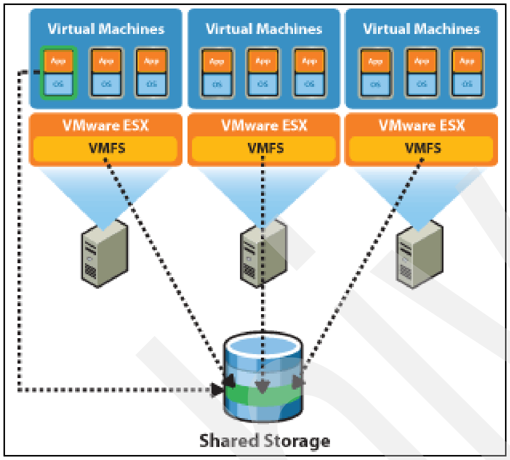



*Figure 4-3 VMFS (Copyright © 2009 VMware, Inc. All rights reserved.)*

Virtual machines have virtual disks that are seen by the guest operating system (OS) as physical disks. These virtual disks are actually files under VMFS file systems. VMFS can be created under any storage device, which can be either internal or external disk as well as Network Attached Storage (NAS) or iSCSI disk. These disk devices, which are used to store virtual machine data and virtual disks under VMFS file systems, are called datastores. Typically, SAN solutions offer the best performance and features to be used as datastores.

## 4.1.4 vCenter

VMware vCenter Server is a powerful application built to manage VMware's virtual infrastructure, vSphere. Through the vCenter you can centrally manage hundreds of ESX hosts, and each ESX host can have 10 - 50 virtual machines. vCenter delivers the highest levels of efficiency, reliability, and security required to manage a virtualized IT environment of any size.

vCenter is designed to aggregate physical hardware (networking, storage, memory, and CPU) and manage it as a collection of resources that can be allocated dynamically on business needs. Figure 4-4 provides an overview of the vCenter components.

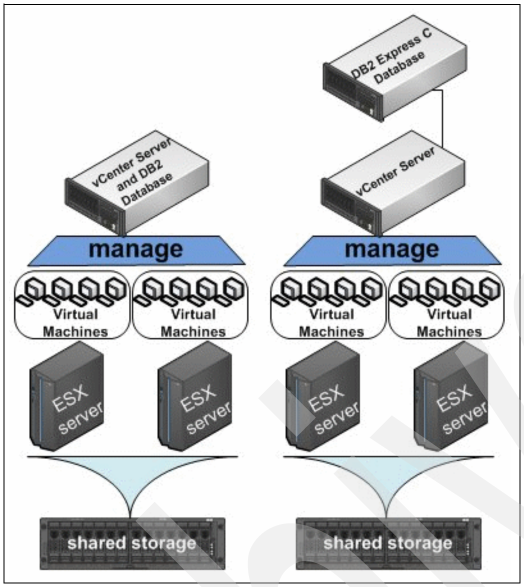



*Figure 4-4 vCenter overview*

The vCenter management information is stored in a dedicated vCenter database to store and organize all its information about the vSphere management environment. The database can be installed on the same machine where the vCenter is installed or on a dedicated database machine. The data in the vCenter database can be divided into three broad categories:

- Inventory

  This includes host and virtual machine configurations, resources, and virtual machine inventory, user permissions, and roles. The inventory information in vCenter stores extra host information.

- Alarms and events

  This includes notifications that are triggered based on virtual machine state changes. The alarm and event data is store indefinitely in the vCenter database.

- Performance statistics

  This includes the performance and resource utilization statistics for datacenter elements, such as hosts, clusters, resource pools, and virtual machines. vCenter provides configurable settings to control how performance statistics are collected across the VMware Infrastructure environment. Performance statistics comprise up to 90% of the vCenter database size and therefore are the primary factor in performance and scalability of the vCenter database.

### 4.1.5 VMware High Availability (HA)

VMware HA provides high availability for the hosts running on the virtual machines. If the host server on the VMware cluster goes down due to a hardware failure or any other reason, all of the virtual machines running on it will go down as well. With VMware HA, if one host fails, the virtual machines will be booted automatically on a different host within the same VMware cluster. The virtual machine will loose only the in-memory state of the virtual machine and the service break will be minimal. VMware HA does not require any additional cluster software to be configured on the virtual machines, which makes it easy to deploy.

### 4.1.6 VMware Fault Tolerance

VMware Fault Tolerance provides continuous availability for your virtual servers and applications by utilizing VMware vLockstep technology. With the VMware Fault Tolerance you have a shadow copy of your running virtual machine, which is able to take place in case of hardware failure. VMware Fault Tolerance is easy to deploy for the existing VMware HA clusters and it does not require the traditional HA capabilities from your software running on virtual machine.

### 4.1.7 Distributed Resource Scheduler

The Distributed Resource Scheduler provides the feature to control resource allocation across hosts and resource pools. Therefore, it collects resource usage information for all the hosts and virtual machines in a cluster and generates recommendations for virtual machine placement. Depending on the configured DRS automation level, those recommendations can be applied manually or automatically. The recommendations can be applied during start of a virtual machine or during runtime.

#### Initial placement

DRS makes recommendations for the placement of a virtual machine or starts it automatically on the recommended host.

#### Load balancing

At runtime, DRS either uses VMotion to automatically migrate virtual machines or provides recommendations for virtual machine migrations.

### 4.1.8 VMotion

VMotion enables you to migrate a virtual machine and its disk files from one machine to another while the virtual machine is running. You can choose to place the virtual machine and all its disks in a single location, or select separate locations for the virtual machine configuration file and each virtual disk.

With VMotion you are able to upgrade VMware Infrastructure without any downtime of the virtual machines. You can then use Storage VMotion to migrate virtual machines back to the original datastore without service interruption. VMotion is also very useful during storage maintenance windows. You can use it to redistribute your storage loads as well to achieve better balance capacity or performance.

## 4.2 VMware setup

In this section discuss how to set up a VMware virtual environment that can also be referred as vSphere. We discuss general concepts and setup steps. We focus on how to set up vSphere so that DB2 servers are able to get most out of it. We discuss the following items:

- Planning and setting up a VMware virtualized environment
- Installing and setting up VMware ESX/ESXi servers
- Installing and setting up vCenter Servers
- Configuring a vSphere through vSphere Client
- Considerations on disk storage and choosing from different options

#### VMware virtual infrastructure planning

Before you start installing and setting up your VMware environment, you should carefully plan your environment. You must think properly about how much and what kind of hardware you need. Another decision to make is what kind software is required for managing and maintaining your VMware virtualized environment.

#### Stand-alone solution

If you have a small environment or you are just exploring VMware capabilities, you might get along with just one stand-alone VMware ESX/ESXi Server. Normally, you would want to have more than one VMware ESX/ESXi Server to be able to provide high availability for your virtualized servers. After all, since you are supposed to run more than one virtualized server on top of your VMware ESX/ESXi Server, you do not want all of them to go down in case of hardware failure.

To be able to provide continuous service you need at least two VMware ESX/ESXi servers. This way you can ensure that in case of failure you are able to move your virtual servers to another VMware ESX/ESXi Server. Moving the virtual machines from one server to another can be done either automatically or manually. Depending on your requirements, this can be done in many different ways.

For instance, DB2 can provide continuous service by utilizing High Availability Disaster Recovery (HADR) technology. This solution does not require external storage and can be implemented through two stand-alone VMware ESX/ESXi Servers both running one the same physical DB2 server. There are similar clustering solutions with other server software. Some of these solutions require external shared storage, but for HADR external shared storage is not required.

### VMware vSphere

To be able to utilize centralized maintenance technologies that VMware has to offer and to fully benefit from VMware HA solutions, you probably want to deploy VMware vCenter. With vCenter your VMware virtualized environment will eventually extend to what VMware calls vSphere. With stand-alone VMware ESX/ESXi Servers your virtual environment will be on some level very much like traditional non-virtualized environments. Although you can benefit from easy server deployments and better hardware utilization by virtualization, you are still lacking many of the superior virtualization features. Once you have moved to a full vSphere environment, you can benefit from the following VMware virtualization features:

- VMware HA: With VMware HA you are able to provide DB2 high availability without deploying any native DB2 HA solutions. Note that VMware HA does not automatically provide continuous service for your applications.
- VMware Fault Tolerance: With VMware Fault Tolerance you are able to provide continuous service without deploying traditional HA solutions.
- VMware DRS: With DRS you are able make sure that your workloads will get the most out of your hardware.
- VMware VMotion: With VMotion you are able to do, for instance, hardware upgrades without service breaks by migrating your virtual server from one physical server to another during the upgrade.
- VMware Storage VMotion: SVMotion enables you to move your workloads from a storage system to another without out service breaks.

Although you are able to achieve most of the benefits that these features have to offer by using DB2 HA technologies, you still might want to consider taking advantage of VMware vSphere. These VMware vSphere features, which are enabled in full vSphere solution, offer even more flexibility if combining with DB2 technologies.

You do not have to upgrade to a full vSphere environment at once. You can start with a stand-alone VMware ESX/ESXi Server and gradually extend your virtualized infrastructure to a full VMware vSphere environment as needed. VMware vSphere is a full package of virtualization features and technologies. VMware vSphere consists of the following software and hardware components:

- One or more VMware ESX/ESXi Servers
- One or more vCenter servers
- Database server
- The underlying hardware that consists of:
  - The physical servers and peripherals connected to them.
  - Storage systems connected to the host servers. These can consist of either external or internal storage solutions, or both.

The number of VMware ESX/ESXi Servers required is dependant on the hardware that you choose to use and how many virtual servers you are going to deploy on your environment. To be able to get the most out of the VMware vSphere, you need at least one vCenter Server. You do not necessarily need additional software for the vCenter Server servers since you can run these on the VMware virtual machines as well.

We highly recommend having external SAN solutions. SAN technologies offer the ability to provide a flexible storage solution with very fast I/O and database servers are usually dependent on fast storage. Also, many of the vSphere features such as VMotion, Storage VMotion, VMware HA, and DRS are dependent on external storage.

### VMware ESX/ESXi installation and setup

There are three options that you can choose from for your VMware ESX/ESXi server:

- VMware ESXi Embedded Edition: This option is available if you use the hardware that VMware ESXi embedded in the hardware firmware.
- VMware ESXi installable Edition: If you choose this option, you can use any supported hardware and perform the VMware ESXi installation.
- VMware ESX Edition: This option has a slightly larger memory foot print, but it does include VMware Service Console, which you might want to have for the configuration and maintenance tasks.

Refer to the VMware documentation for hardware requirements as well as for how to install and set up your VMware ESX/ESXi servers.

## vCenter Server installation and setup

vCenter Server acts as a focal point for a vSphere environment. You are able to configure and maintain the entire vSphere environment through vCenter. To store and access vSphere configuration data vCenter requires a supported database. vCenter Server can be installed both on physical and on virtual hardware. The database server can be installed on the same virtual or physical machine as vCenter, or you can dedicate additional hardware for it.

Before starting the vCenter Server installation you must check the hardware requirements. Up-to-date hardware requirements, including installation and setup guides, are available at:

http://www.vmware.com/support/pubs/vs_pubs.html

You also must decide what database solution you are going to use with your environment. vCenter Server is bundled with Microsoft SQL Server 2005 Express database, which allows you to set up to five hosts and 50 virtual machines. vCenter server supports other database engines including DB2 for larger deployments. As for vCenter Servers, you do not necessarily need additional hardware for the database server. You can run it on a virtual server.

## vSphere Client

VMware vSphere Client can be installed on a workstation, mobile computer, or server depending on your needs. Check the up-to-date data for hardware and software requirements for vSphere Client from VMware documentation. By using one vCenter Client you are able to connect to one or more vCenter Servers. You can also have more than one vSphere Clients installed to configure and manage your vSphere environment.

## Managing vSphere

Once you have your environment set up you are ready to start configuring your VMware vSphere. Through vSphere Client you are able to:

- Create data centers. You can define a data center, for example, based on the geographic location of your servers.
- Add or remove ESX/ESXi hosts. You are able to add or remove hosts to and from your vSphere. Hosts must be added under a certain data center.
- Create and manage clusters. You are able to add and modify host clusters under the data centers. The host clusters must be able to utilize vSphere features such VMotion, VMware HA, and DRS.
- Create and manage resource pools. You can manage your workloads with resource pools. Use vSphere Client to manage the resource pools.
- Create and manage virtual machines. The vSphere Client is the focal point for creating and managing the virtual hosts. You are able to change resource

settings and get information about virtual machine performance through the vSphere Client.

- Manage and configure the datastores. You are able to deploy and manage datastores. You are able to configure the storage paths for datastores. If you need to upload, download, or browse files and folders on your datastores, use the vSphere Client.
- Monitor host and virtual machine performance. You can use the vSphere Client to view and create performance graphs about the hardware resource usage including I/O, memory, and CPU on both the hosts and guests.

vSphere Client offers all the tools that you need to configure and manage your vSphere. vSphere Client is a good example of how virtualization can make managing and monitoring a large number of servers much easier than it would be on a traditional physical server environment.

You are able to monitor how the servers use the I/O, CPU, and memory resources. If there is a need for more I/O, CPU, or memory, you are able to re-allocate these resources among the servers. For example, if you observe that one server has a lot of idle memory, you can re-assign the memory for other servers that might be lacking it.

Setting up or creating a virtual machine on a properly designed vSphere simplifies the tasks remarkably if compared with similar tasks required on a physical environment. You are able to create a virtual machine in minutes. You can utilize VMware templates to make the operating system deployment much easier than it would be normally. If there is additional software required to be installed for all the servers due to company policy or other issues, with templates you get the ready-build operating systems containing all the required software pre-installed and configured.

### Considerations on storage

Storage plays a critical role for the performance of a database system. DB2 is no exception. You must design and build the storage in such a way that DB2 will have enough I/O resources to manage the I/O bound workloads. A good way is to utilize the SAN systems with properly configured arrays. Just as the best practices suggest, you should think about the number of spindles, not just how much storage space is required. The more spindles you have in your RAID array, the more I/O they can generate.

### *Failover configuration*

You should design your system to have the failover capabilities in case of hardware failures. Figure 4-5 shows one example of a SAN failover configuration.

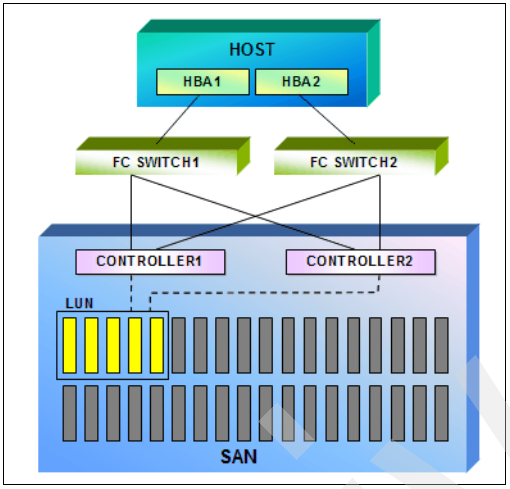



*Figure 4-5 A SAN failover configuration*

In general, a SAN solution with a failover configuration consists of the following components:

- The SAN box itself.
- The disks in the SAN.
- A minimum of two controllers in the SAN. All of them have at least two Fibre Channel ports. SAN controllers can operate in active/passive or active/active mode. With active/passive mode, all disk arrays are accessed through the active controller. Load balancing is not possible on active/passive mode. With active/active mode both controllers can be used simultaneously. All our examples assume the use of active/active mode.
- The host machine typically has at least two Fibre Channel host bus adapters (HBAs).
- Two Fibre Channel switches. In our example we have connected the Fibre Channel switches to both controllers. With this configuration, both HBAs on the host can access the LUN through both SAN controllers.

We use a lab configuration similar to this one for our discussion in this chapter.

When a storage administrator configures the SAN, he first creates a RAID array. In our lab, we have five disks. The plan is to configure the RAID array on these five disks—four for data and one for parity, a normal RAID-5 4+1 configuration.

Using the SAN configuration software, we can assign the logical unit number (LUN) on the RAID array and present the LUNs to the host machine so that the host machine is able to see the LUNs as physical disks. Our host machine has two Fibre Channel cards, which are connected to the SAN box controllers through the Fibre Channel switches. This makes a total of four paths to the LUNs, where two of them are controlled by the host machine through the Fibre Channel cards. Note that without correct multi-path Fibre Channel drivers, the host machine would see each LUN as four physical disks. By using supported hardware we ensure that the host machine will see correctly only one disk per LUN and can take advantage of multi-path drivers.

This simple failover configuration provides us a certain level of hardware failure protection. We can lose one Fibre Channel HBA on the host machine, one Fibre Channel switch, or one Fibre Channel controller on the SAN box without service interruption. The Fibre Channel multi-path drivers are able to find the working path in case of any of these failures happens.

### *Configuring disks for DB2 data*

You should always use dedicated LUNs for DB2 data and transaction logs to achieve optimum performance as suggested in DB2 best practices. This means that you should not use the LUNs or the file systems under the LUNs for any other purpose. Otherwise, you might end up generating competition on I/O resources.

You can choose from two different techniques on how you are going to use the dedicated LUNs:

- Raw device mapping (RDM)
- VMFS

VMware has a feature called raw device mapping that allows you to access the LUNs as they were locally attached to the virtual machine. Normally, you present a LUN to the host machine and use it as your datastores.

Each VMware datastore has a file system called VMFS. On the top of this file system you are able to create virtual disks, which you can use to store your virtual machine data. Virtual machines see these virtual disks as physical disks.

There are certain differences in how RDM and VMFS flat files will perform as your virtual disks. RDM does not have the overhead of an additional file system, but lacks the flexibility of VMFS datastores. Tests imply that there is no noticeable difference in performance when using RDM instead of VMFS datastores. VMFS can even be faster than RDM on certain workloads. Thus, we recommend always using VMFS instead of RDM virtual disks unless there are other reasons to use it.

Figure 4-6 shows one possible way of setting up datastores for DB2 storage paths.

**Datastores**

| Identification | Device | Capacity |
|---|---|---|
| eptesx1:storage1 | vmhba1:0:0:3 | 675.50 GB |
| ESXSTOR_1 | vmhba3:0:0:1 | 339.25 GB |
| ESXSTOR_2 | vmhba3:0:1:1 | 339.25 GB |
| ESXSTOR_3 | vmhba3:0:2:1 | 339.25 GB |
| ESXSTOR_4 | vmhba3:0:3:1 | 339.25 GB |

*Figure 4-6 Datastores for DB2 storage paths (Copyright © 2009 VMware, Inc. All rights reserved.)*

Here we have four datastores:

- ESXSTOR_1
- ESXSTOR_2
- ESXSTOR_3
- ESXSTOR_4

We could use these as our DB2 storage paths. We could now create one virtual disk for each of these storage paths, and use this virtual disk as a DB2 storage path. When the storage paths are dedicated for DB2 storage paths, we are able to achieve optimal I/O performance. In 4.3, "Virtual machine setup" on page 92, we show how we use VMware datastores to set up virtual disks and create storage paths on them.

### *Setting the path switching policy*

With the path switching policy, you define how you access the datastores with the failover capable equipment. The path switching policy is not only used to recover from the hardware failures, but also to provide load balancing when required. Once you have added your datastores you must choose which path switching policy you are going to use for your datastores. You should have multiple paths available to access the LUNs, as explained in "Failover configuration" on page 88. These paths can be used not only for failover, but also for load balancing. The patch switching policy that you choose not only effects on how your systems behave on failover, but also how much the load is balanced over the multiple paths.

There are three different options that you can choose from:

- Fixed: This policy uses a fixed active path to access the LUN. You are able to define which HBA is the preferred path for the target LUN.
- Most Recently Used (MRU): With this option you are able to access the LUN through the most recently used HBA.

- Round Robin (RR): With this option you get load balancing across the HBAs. The I/O requests are balanced over the HBAs.

Even though you choose fixed as your path switching policy, you can still have load balancing over your datastores. When setting fixed for the path switching policy you are able to set the preferred path. The preferred path will always be used to access the LUN when the LUN is accessible through the preferred path.

In our test environment we have two Fibre Channel HBAs. We can set up our datastores so that datastore1 and datastore3 are accessible through HBA1, and datastore2 and datastore4 are accessible through HBA2. With this setup, the load is balanced between two Fibre Channel HBAs.

If we choose MRU as the path switching policy, we will not have as much control over the load balancing since we cannot guarantee that the path will stay fixed. In case of failover, no matter what path switching policy you use, the LUN will be accessed through the working paths.

In case of failover, in all three path switching policies the path to the LUN will always be changed so that the LUN is usable. Even if you use the fixed policy and have defined a preferred HBA, the preferred HBA is not used if the LUN is not accessible through it. When the path through the preferred HBA is available again, the preferred HBA will be used if you use the fixed path switching policy. If you are with the MRU, the path will remain where it was set after the failover.

To obtain the best load balancing, we recommend that the RR path switching policy goes with the dedicated datastores. If you are not using the dedicated datastores, you might want to consider using balanced fixed paths.

You are able to change the path switching policy for your hosts through the vSphere Client using the following steps:

1. Choose the host that you want to modify.
2. Choose the **Configuration** tab.
3. From Hardware panel at the left, choose **Storage**.
4. Right-click the datastore that you want to modify and choose **Properties**.
5. Choose **Manage Paths** and define the path switching policy for your datastore.

## 4.3 Virtual machine setup

In this section we discuss how to set up and deploy virtual machines. We concentrate on setting up a virtual machine for DB2. We have emphasized the importance of I/O throughout this book, and this section is no exception. We concentrate on how to set up optimized storage for DB2. We focus on the following topics:

- Setting up the virtual machine
- Setting up CPU, memory, and storage
- Configuring the virtual machine resource settings

### 4.3.1 Creating virtual machine

To get started you must create a virtual machine. This can be done through vSphere Client:

1. Choose **Inventory: Hosts and clusters**.
2. Right-click either Datacenter, Cluster, resource pool, host cluster, or individual host machine and choose **New Virtual Machine**.

3. This launches the New Virtual Machine Wizard. The wizard guides you through the initial virtual machine setup steps. Figure 4-7 shows the summary page of the wizard.

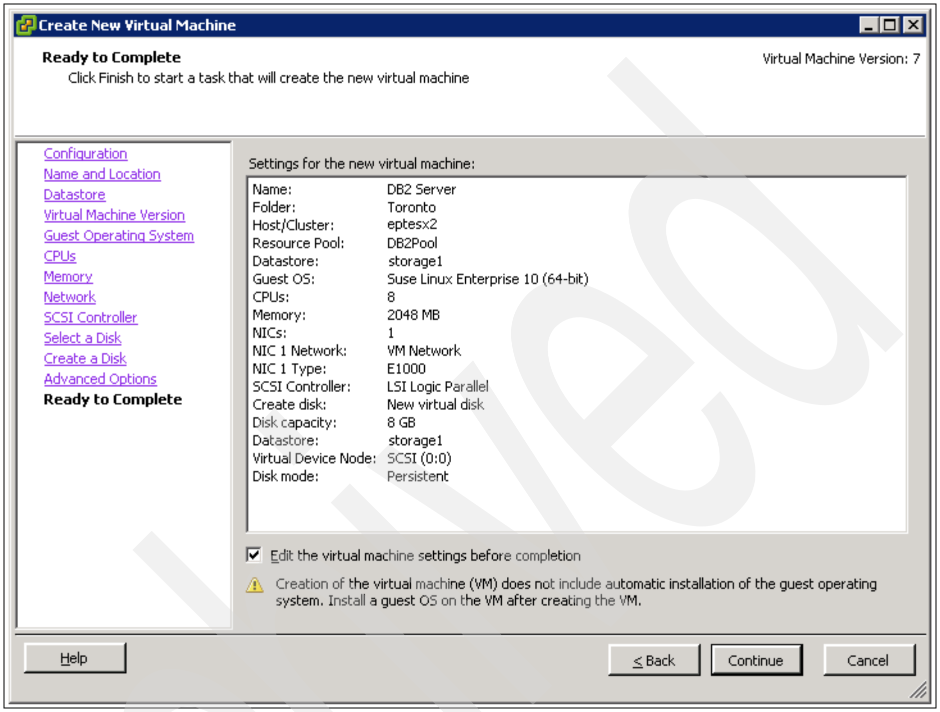



*Figure 4-7 Create New Virtual Machine wizard (Copyright © 2009 VMware, Inc. All rights reserved.)*

Among the few items that you must set up, the most important ones for DB2 performance are CPU, memory, and I/O.

## CPU

The processor load between virtual processors will be shared between logical processors on the host machine. If the host machine is hyper threading (HT) capable and the HT is enabled, each processor core equals to two logical processors, so the total number of the logical processors would be two times the number of processor cores. If the host machine does not have HT enabled, the number of logical processors equals the number of physical processor cores.

As shown in Figure 4-7, we set up our DB2 virtual machine to have eight virtual processors, which is the maximum that ESX/ESXi supports. By defining the maximum of eight virtual processors, we ensure that the processor load will be divided between the maximum number of logical cores. Note that if your host machine has eight or fewer logical processors, you should think carefully about how many virtual processors you will assign to the virtual machine. It would not be the best practise to assign as many virtual processors to the virtual machine as the system has logical processors.

### Memory

We start with 2 GB of memory. The amount of memory can be adjusted easily later just as the number of virtual processors can.

For the database server we want to have a dedicated memory. VMware is capable of creating and using swap file for virtual machines. This swap file is used when VMware does not have enough physical memory resources to be allocated for virtual machines. The size of the swap file is the difference of the values between the actual virtual machine *memory size* and *reservation*. To ensure that we have all the memory dedicated and not to use a swap file, we set the reservation value to be the same as the amount of virtual machine memory.

When editing the virtual machine settings with vSphere Client, we choose the **Resources** tab, select **Memory**, and set reservation to 2048 MB, as shown in Figure 4-8.

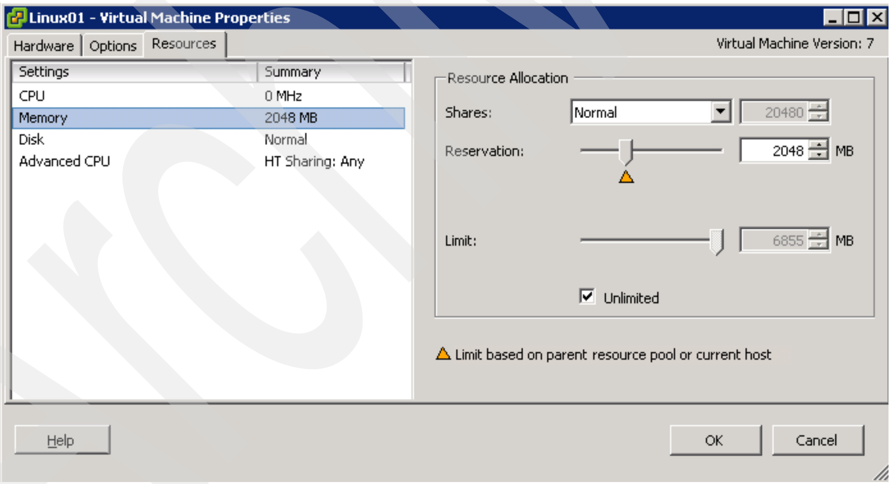



*Figure 4-8 Using dedicated memory (Copyright © 2009 VMware, Inc. All rights reserved.)*

Be aware that although the amount of memory and the number of virtual processors can be changed dynamically with a proper VMware license, you might not be able to make changes without rebooting if the guest OS does not support hot-plug memory or CPU. If this is the case, you must first shut down the virtual machine, then adjust the values and boot the virtual machine with the adjusted memory or CPU values.

## I/O and disk setup

We first place both the virtual machine configuration files and the operating system virtual disk under datastore storage1. Next we create the storage paths for DB2. Since our test environment only has four storage paths, we are not able to use dedicated datastores for DB2 data and transaction logs.

We checked the **Edit virtual machine settings before completion** option on the New Virtual Machine Wizard. When we click **Continue**, we are able to add more hardware. We choose **Add ∅ Hard Disk ∅ Next ∅ Create a new virtual disk** to reach the panel where we are able to specify a datastore on which we want to store our virtual disk, as shown in Figure 4-9.

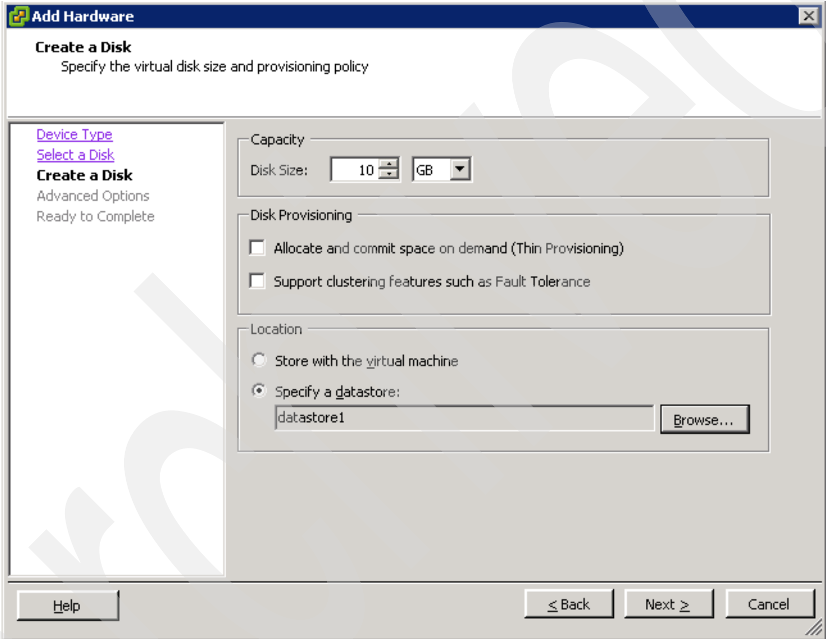



*Figure 4-9 Creating virtual disk for DB2 storage path (Copyright © 2009 VMware, Inc. All rights reserved.)*

Note that we do not check *Thin Provisioning,* which allocates the disk space in an on demand basis. We use *Thick Provisioning* to have the disk space specified allocated immediately.

We click **Next** and get the panel that lets us choose which virtual device node to use. The default is next available. Since the operating system disk is under SCSI(0:0), the next default is SCSI(0:1). We do not choose that. Instead, we choose **SCSI(1:0)** as our virtual device node for our first storage path because we want our data disks to be available through a different virtual SCSI adapter from our operating systems. See Figure 4-10.

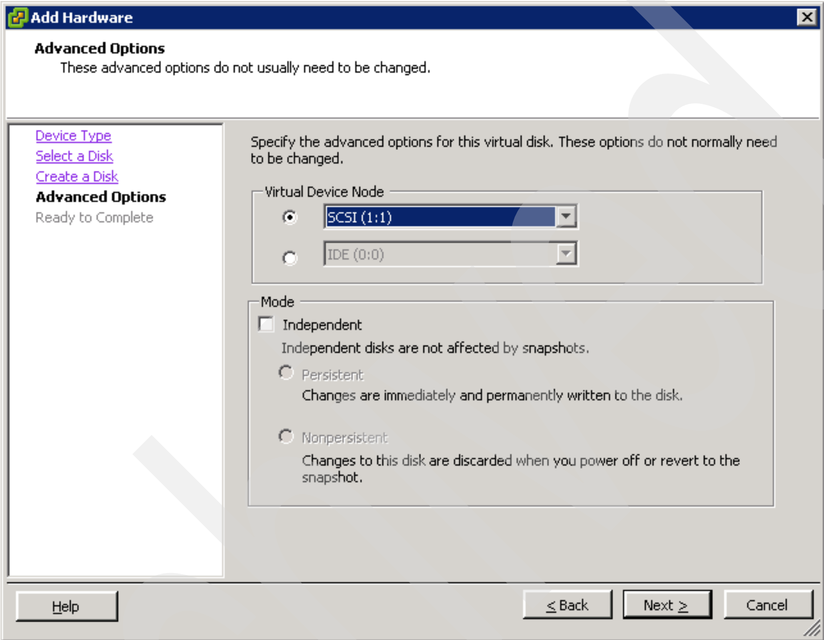



*Figure 4-10 Virtual device node (Copyright © 2009 VMware, Inc. All rights reserved.)*

After we click **Next** ∅ **Finish** our first disk is ready. Since we are going to have a total of four storage paths for our DB2 data, we continue by adding three virtual disks more after we have complemented creating the first one. The remaining three ones are similar to the first one, but are placed in different datastores: datastore2, datastore3, and datastore4 with the virtual device nodes SCSI(1:1), SCSI(1:2), and SCSI(1:3), respectively. We still need two additional virtual disks for DB2 transaction logs and test data. We assign virtual device nodes SCSI(0:1) and SCSI(0:2) for the last two virtual disks.

Depending on the guest OS support for hot-plug disks and the type of VMware license, you are able to add, remove, or even resize disks dynamically without powering off the virtual machine.

## 4.3.2 Setting up operating system environment

Now that we have set up the physical aspect of our virtual machine, we can continue installing the operating system (OS) on it. In our test environment we use SuSE Linux Enterprise 10 64-bit edition and Microsoft Windows 2008 Server as our test virtual machines. The OS installation is done the standard way.

Before setting up the OS environment, you should install VMware Tools. With the VMware Tools, you can benefit from a significantly better display and the network adapter drivers that are designed especially for VMware virtual machines. To be able to utilize all the capabilities of clones and templates VMware Tools are required as well. Refer to VMware documentation for how to perform the VMware tools installation for your guest OS.

In our example, the virtual disks for DB2 data were added when the virtual machine was created. We did not create any file system on these disks during the OS installation. It is possible to add virtual disks online to the virtual machine. Depending on the operating systems, the disks are usually handled as the hot-pluggable disks, and the OS is able to initialize them without the need for rebooting the OS. In our tests both SuSE Linux Enterprise 10 and Microsoft Windows 2008 Server were able to detect virtual disks that were added to the system online. Also, both of the operating systems are able to detect and utilize changes on the virtual disk sizes online without reboot.

### Setting up the file systems for DB2 on Linux

We create four virtual disks to be used for DB2 database storage paths. We add one additional file system for transaction logs. Our virtual disk for the OS is first seen as the attached SCSI disk on our system. It is seen as `/dev/sda` by the OS. For the OS some prefer to use Logical Volume Management (LVM). With LVM you are able to benefit from all the features and flexibility that it offers. You also can use LVM for the DB2 database storage paths. However, for the maximum performance we recommend not utilizing LVM, since it adds overhead to the system.

To set up the file systems for DB2 data, we must find the device names for the disks first. This can be done, for example, by checking with the **dmesg** command what disk has been attached to the system, as shown in Example 4-1.

*Example 4-1 Finding attached disks*

```
# dmesg | grep "Attached scsi disk "
sd 0:0:0:0: Attached scsi disk sda
sd 0:0:1:0: Attached scsi disk sdb
sd 0:0:2:0: Attached scsi disk sdc
sd 1:0:1:0: Attached scsi disk sdd
sd 1:0:2:0: Attached scsi disk sde
```

```
sd 1:0:3:0: Attached scsi disk sdf
sd 1:0:4:0: Attached scsi disk sdg
```

We set up our test environment so that we have file systems for DB2 data under the virtual device node SCSI(1:x), as shown in Figure 4-10 on page 96. We created two additional virtual disks for transaction logs and test data under the storage1 datastore. We set their virtual device node to SCSI(0:1) and SCSI(0:1). These two disks are seen as /dev/sdb and /dev/sdc by the OS. The devices /dev/sdd, /dev/sde, /dev/sdf, and /dev/sdg are the disks on which we place our DB2 data. Note that since we have set up our transaction logs under the same LUN where we are storing the OS and our first DB2 storage path, this is not an optimal solution considering performance. On a high-load production system you should place transaction logs and DB2 storage paths under dedicated LUNs.

Now that we have identified our disks, we create one primary partition for each of them. We used the **fdisk** command to create the partition. We then created an ext3 journal file system for each disk by using the **mkfs -t ext3 /dev/sd*x*1** command, where *x* refers to the corresponding disk device. Next we labeled all the file systems with the **tune2fs** command. For instance, /dev/sdd1 was labeled as data1 with command **tune2fs data1 /dev/sdd1**. Next we added the lines shown in Example 4-2 to /etc/fstab for DB2 data and transaction logs.

*Example 4-2 Example of /etc/fstab*

```
LABEL=data1          /db2data/data1    ext3    defaults     1 2
LABEL=data2          /db2data/data2    ext3    defaults     1 2
LABEL=data3          /db2data/data3    ext3    defaults     1 2
LABEL=data4          /db2data/data4    ext3    defaults     1 2
LABEL=db2logs        /db2logs          ext3    defaults     1 2
```

By labeling the partitions, you can ensure that when you add new virtual disks, the file systems will be mounted properly even if the device names for the virtual disks are changed. Now we are able to mount all our file systems for DB2 data and transaction logs. Before creating the database under them, we must change the file system owner and corresponding group to the DB2 instance owner. Example 4-3 shows the file system layout for DB2.

*Example 4-3 Linux file system layout for DB2*

```
Filesystem            Size  Used Avail Use% Mounted on
/dev/sdd1              10G   0G  10G  100% /db2data/data1
/dev/sde1              10G   0G  10G  100% /db2data/data2
/dev/sdf1              10G   0G  10G  100% /db2data/data3
/dev/sdg1              10G   0G  10G  100% /db2data/data4
/dev/sdb1              10G   0G  10G  100% /db2logs
```

When you add or change the size of the virtual disks on Linux, you can use the rescan-scsi-bus.sh utility to detect the changes dynamically.

### Setting up the file systems for DB2 on Windows

As on modern Linux distributions, you are able to add and change the size of the existing virtual disks dynamically on the Windows 2008 Server. You can add and initialize your virtual disks by using the Disk Management tool. We initialized the disks as dynamic disks and used NTFS. We labeled the disks to reflect the usage for DB2, as illustrated in Figure 4-11.

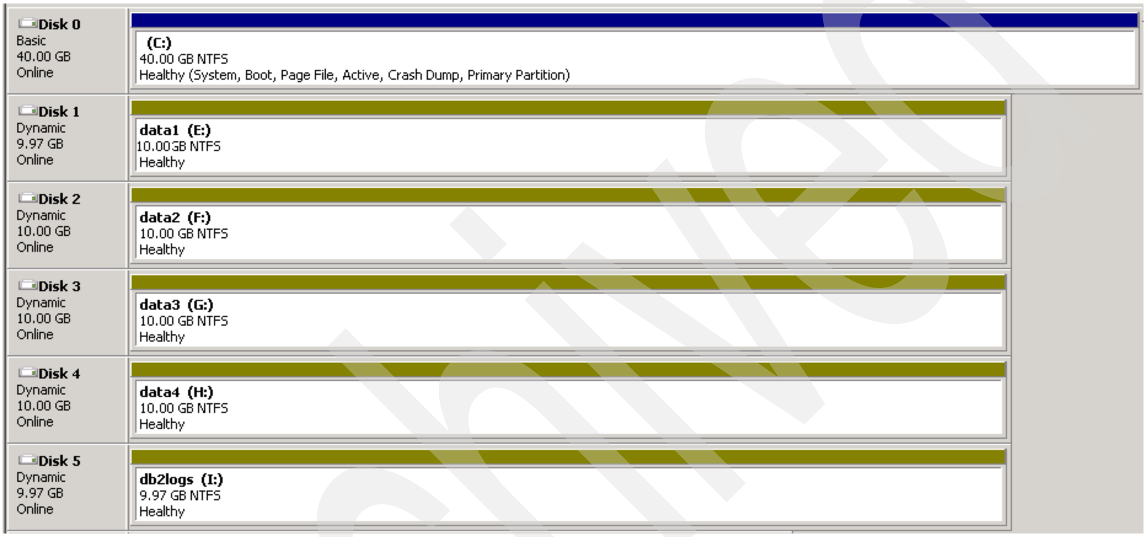



*Figure 4-11 Windows file system layout*

On Windows 2008 Server, to detect dynamic changes for virtual disks, use the Server Manager utility and right-click **Disk Management** under Storage and choose **Rescan Disks**.

## 4.3.3 Resource settings

On vSphere you are able to prioritize CPU, memory, and I/O usage between your virtual machines. You can define three different values for how the resources are allocated:

- Shares: By default all the virtual machines are set to have an equal amount of shares concerning CPU, memory, and I/O resources. This means that the virtual machines are provided with an equal amount of resources when there is contention among them.
- Reservation: You can set the minimum amount of processor resources in MHz and the minimum amount of memory in megabytes that a virtual machine will be granted when it is powered on. By default the reservation is set to 0.
- Limit: You can define the upper bound of processor and memory resources for a virtual machine. For the processor you define the maximum number of CPU resources in MHz and for memory the maximum amount of RAM in megabytes that a virtual machine is able to get on any condition. By default the limit is set to unlimited.

You are able to change the resource setting for CPU and memory dynamically when the virtual machines are online. If you want to change the shares for disk I/O, you must first power off the virtual machine.

To change the default resource settings with vSphere Client, right-click the corresponding virtual machine and select **Edit Settings**, then select the **Resources** tab. This opens the resources panel, as shown Figure 4-12.

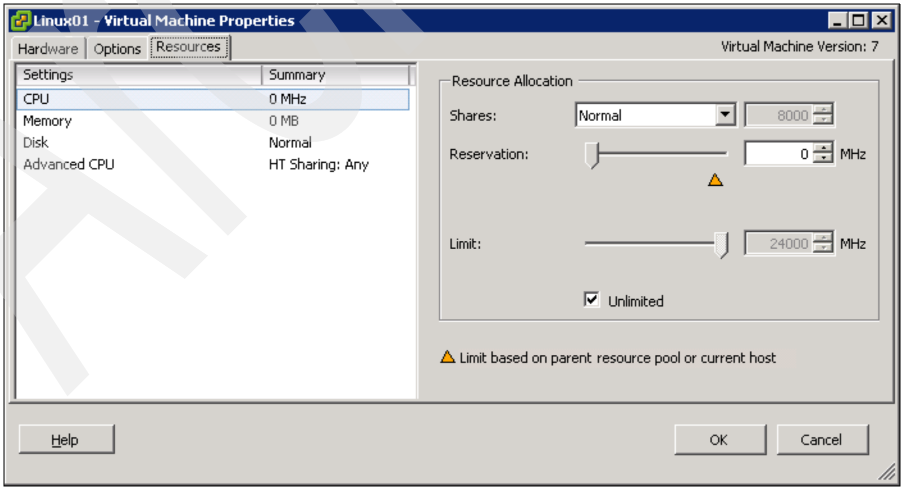



*Figure 4-12 Resource settings (Copyright © 2009 VMware, Inc. All rights reserved.)*

As you can see from Figure 4-12 on page 100, besides CPU, memory, and disk, there is also a selection for Advanced CPU. With this selection, you are able to set the *hyperthreaded core sharing* and the *scheduling affinity.* By default the hyperthreaded core sharing is set to Any. This means that the virtual processors are able to share the cores with any virtual processor on other virtual machines or on the same virtual machine.

With the scheduling affinity, you are able to allocate specific logical processors to be scheduled exclusively to this guest. Refer to the VMware documentation for how to change these settings from their defaults.

### Resource pools

You can create resource pools to have more control over how resources are shared between guests. For instance, you could create individual resource pools for the test servers and production servers. You might want to provide more resources for the production servers to be able to handle production workloads, which usually has higher priority.

Since vSphere 4 resource pools can be defined hierarchically. The child resource pool inherits the resources from its parent pool.

Use resource settings and resource pools to prioritize and tune the resource allocation for your database servers accordingly. For more information about resource management refer to *vSphere Resource Management Guide*, available at following site:

http://www.vmware.com/pdf/vsphere4/r40/vsp_40_resource_mgmt.pdf

## 4.3.4 Monitoring the resource usage

VMware vSphere offers well-developed and easy-to-use monitoring tools. You are able to monitor the resource usage for processor, memory, and I/O with vSphere Client. You can monitor the resources on both the host and the guest level. This information about resources and how they are used provides you with valuable data about how your virtual machines behave in a virtualized environment. You can use this information to tune your database server and manage your resource settings for a particular virtual machine, as well as the entire virtualized environment.

For instance, by monitoring the resource usage for your database server, you might be able to find the bottlenecks on the server very easily. By changing the resource settings for the virtual server, you might be able to fix the problem online just by adding the lacking resource.

### 4.3.5 Cloning and templates

With VMware vSphere you can clone the existing virtual machines. Alternatively, you can use an existing virtual machine to create a template, which you can use later to deploy new virtual machines with pre-installed software and standardized hardware components. When you create a clone or convert a template to a virtual machine, you are able to change, for instance, network settings before the cloned or converted virtual machine is booted. Use of the VMware cloning feature and templates simplifies system administrator work and saves a significant amount of time.

Utilizing VMware cloning capabilities and templates with DB2 you are able to, for instance, build a test environment from a production environment. Or you can build a DB2 server template with DB2 pre-installed and configured.

If you are cloning your production database, you probably want to change the name of the database and possibly even rename the instance. With DB2, both renaming the database and instance are easily done with the db2relocatedb utility. You must create a basic configuration file where you define the database and instance names.

Example 4-4 shows a simple configuration file to be used with db2relocatedb. With this configuration file you change only the database name and keep both the database path and the instance name as they were.

*Example 4-4 db2reloactedb configuration file*

```
DB_NAME=PRODDB,TESTDB
DB_PATH=/db2data/data1,/db2data/data1
INSTANCE=db2inst1,db2inst1
```

The db2relocatedb utility has the following syntax:

```
db2relocatedb -f <config_file>
```

When you clone or convert a template to a virtual machine, you must edit `db2nodes.cfg` according the changes in the host name and IP address.

## 4.4 DB2 setup and configuration

In this section we describe the installation and the setup of a DB2 environment on a VMware virtual machine. We focus on the best practices to achieve optimal performance for a DB2 database.

## 4.4.1 Installation

The DB2 installation in a VMware virtual machine depends on the chosen operating system for the virtual machine, just like installing DB2 on a physical server. There is nothing specific to have in mind for VMware. You can just follow the instructions from the DB2 installation guides at the Information Center:

http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.qb.server.doc/doc/c0024080.html

## 4.4.2 Setup

In a virtualized environment, you can particularly benefit from the DB2 autonomic features and the compression technologies introduced in 2.1.2, "Features and functions" on page 15. The database engine is capable of detecting changes in the number of processors and the memory size dynamically online without the need to restart the instance. DB2 will apply the necessary adjustments automatically for the database heaps and other parameters to optimize performance.

### Storage

When setting up the storage systems for DB2 there are several topics to keep in mind. The most important ones are:

- Place the transaction log files and the data log files on separate file systems or on separate disks. In Example 4-5 we show how to set up different paths for the datastore during the database creation.

  *Example 4-5 Database creation with storage paths*

```
CREATE DATABASE virtdb AUTOMATIC STORAGE YES ON /db2data/data1,
/db2data/data2, /db2data/data3, /db2data/data4 DBPATH ON /db2logs
```

- Note that in Example 4-5 we use automatic storage as well. This feature provides a great relief for DBAs by automating the management of the table space containers. DB2 creates the table space containers automatically on the given data paths. This feature also helps with performance, as DB2 places the table space containers in a way that allows the most concurrent access. If you are running out of storage you just must add additional data paths and DB2 will take care of the optimal distribution of the data. We show the simple command for this action in Example 4-6.

  *Example 4-6 Adding storage path*

```
ALTER DB virtdb ADD STORAGE PATH /db2data/data5, /db2data/data6
```

- You should set the EXTENTSIZE parameter of the CREATE TABLESPACE command to at least the number of pages that are included in an entire RAID stripe. A multiple of this RAID stripe size is fine as well. Figure 4-13 shows an example.

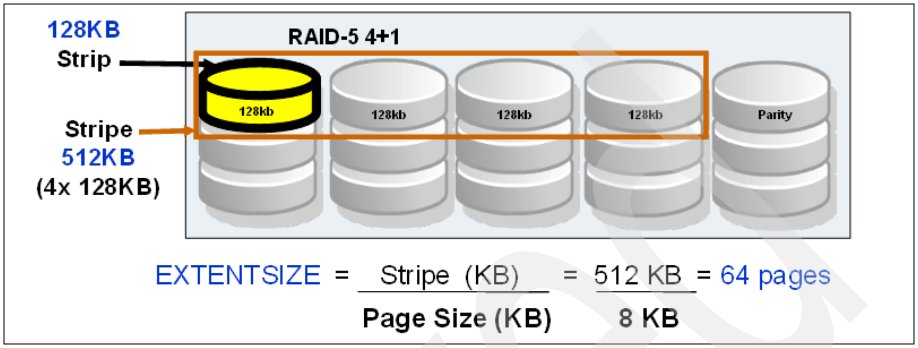



*Figure 4-13 Extentsize: RAID stripe size*

- Use the NO FILE SYSTEM CACHING parameter of the CREATE TABLESPACE command for all data and index table spaces. Because this data is already cached in the buffer pools, there is no need for additional caching at the file system level.
- Use the FILE SYSTEM CACHING parameter of the CREATE TABLESPACE command for LOB table spaces. This data is not cached in the buffer pools, so enabling caching on the file system level increases performance.

## Memory

For optimal memory usage the Self Tuning Memory Manager (STMM) should be switched on (SELF_TUNING_MEM=ON, default in DB2 9.7). STMM tunes several memory heaps according to the workload on your database when they are set to AUTOMATIC. Table 4-1 lists these memory heap parameters.

*Table 4-1 Memory heap parameters for STMM*

| Parameter name | Value |
|---|---|
| DATABASE_MEMORY | AUTOMATIC |
| LOCKLIST | AUTOMATIC |
| MAXLOCKS | AUTOMATIC |
| PCKCACHESZ | AUTOMATIC |
| SHEAPTHRES_SHR | AUTOMATIC |
| SORTHEAP | AUTOMATIC |

| Parameter name | Value |
|---|---|
| DBHEAP | AUTOMATIC |
| STMTHEAP | AUTOMATIC |
| APPLHEAPSZ | AUTOMATIC |
| APPL_MEMORY | AUTOMATIC |
| STAT_HEAP_SZ | AUTOMATIC |

In Table 4-2 we present the parameters that are responsible for an optimal I/O performance between the buffer pools and the storage.

*Table 4-2 Thread-based parameters for STMM*

| Parameter name | Value |
|---|---|
| NUM_IOCLEANERS | AUTOMATIC |
| NUM_IOSERVERS | AUTOMATIC |
| DFT_PREFETCH_SZ | AUTOMATIC |

Table 4-3 provides the application-specific parameters influenced by STMM.

*Table 4-3 Application specific parameters for STMM*

| Parameter name | Value |
|---|---|
| MAXAPPLS | AUTOMATIC |
| AVG_APPLS | AUTOMATIC |

For a detailed description of each of these parameters go to:

http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.config.doc/doc/r0005181.html

## Automatic maintenance

In this section we discuss self-healing features of DB2 as well as automatic database object management.

### *Self-healing*

To set up the self-healing features of DB2 you can either use the DB2 Health Monitor (deprecated in DB2 9.7) or the new feature called Data Studio Administration Console. You can enable several health indicators and define thresholds for them. If these thresholds are violated, DB2 raises an alert. You can define actions (scripts) that DB2 will execute in case of a violation. These self-healing features of DB2 can save a lot of time for the responsible DBAs.

### *Automatic object maintenance*

The automatic object maintenance feature enables DB2 to take care of the regular maintenance tasks BACKUP, RUNSTATS, and REORG for DBAs. DB2 senses the workload occurring on the database and on your tables. Based on internal indicators, the DB2 engine decides whether it is necessary to run one of the defined processes.

By defining policies for these processes you can, for example, tell DB2 what tables should be checked for RUNSTATS needs. These policies are provided to DB2 in XML format. You can find several XML example files in the samples directory of your instance.

You also can define maintenance windows on the periods when the database server has lower workloads to run the maintenance tasks. For BACKUP, RUNSTATS, and REORG processes, you can define whether these jobs must be executed within the maintenance window defined or whether they are allowed to run outside the maintenance window as well.

Be aware that most virtualization product vendors also provide backup utilities for their virtual machines. These utilities are not sufficient if you are running databases in the virtual machines. They cannot guarantee the consistency of your database because they just create snapshots of the database files.

## Compression

Though we have several types of compression built in DB2, be careful when you use them. If you enable compression features while the virtual machine is already CPU-bound (> 90%), then the system will probably become slower due to the compression/decompression overhead on the CPU. Because the created compression dictionary occupies storage as well, even if the system is not CPU-bound, you still must consider whether the compression (for example, row compression) makes sense for a certain table. Small tables do not qualify for compression then.

DB2 provides the **inspect** command for estimating the storage savings for enabling row compression on a table. You can run it from the DB2 command line. We introduce this command in Example 4-7. This command creates a binary output file in the `db2dump` directory of your instance. The binary file can be translated into a text file using the db2inspf utility. The usage of this utility is also shown in Example 4-7.

*Example 4-7 inspect and db2inspf command*

```
db2 "INSPECT ROWCOMPESTIMATE TABLE NAME RESULTS KEEP virtdb.bin"
db2inspf virtdb.bin virtdb.txt
```

Example 4-8 shows the output of the db2inspf utility. You can see from this example that the storage savings are 46% of the original storage amount.

*Example 4-8 db2inspf output*

```
DATABASE: SAMPLE
VERSION : SQL09050
2009-08-11-16.03.00.361000


Action: ROWCOMPESTIMATE TABLE
Schema name: OPTIM
Table name: EMPLOYEE
Tablespace ID: 2  Object ID: 6
Result file name: virtdb.bin

    Table phase start (ID Signed: 6, Unsigned: 6; Tablespace ID: 2) :
OPTIM.EMPLOYEE

      Data phase start. Object: 6  Tablespace: 2
      Row compression estimate results:
      Percentage of pages saved from compression: 46
      Percentage of bytes saved from compression: 46
      Compression dictionary size: 13312 bytes.
      Expansion dictionary size: 10224 bytes.
      Data phase end.
    Table phase end.
Processing has completed. 2009-08-11-16.03.00.407000
```

The **inspect** command works for row compression only. To get an impression of the amount of savings for index compression or XML compression you must run administrative table functions delivered with DB2.

# 4.5 Conclusions

By setting up a test environment, running thorough tests, and following the DB2 best practices on a VMware virtualized environment, we came to the following conclusions:

- On DB2 storage, use SAN disks and dedicated LUNs for maximum performance. Use virtual disks placed on dedicated VMFS file systems on the dedicated LUNs on the SAN system. RDM does not offer better performance and lacks the flexibility of virtual disks on VMFS. Utilize DB2 automatic storage to stripe over the file systems.
- Set the path switching policy accordingly. Setting the path switching policy to reflect disk and SAN setup has a great effect on the I/O performance. When you are using dedicated LUNs with active/active capable SAN equipment, set the path switching policy to round robin.
- Properly tune the DB2 resource settings. This gives you the ability to prioritize and handle the DB2 server workloads properly, especially when there is competition on resources.
- Use vSphere monitoring tools to set and tune resource settings. Utilize the rich set of monitoring capabilities that vSphere has to offer to tune your virtualized environment. You also can use this information to tune your DB2 configuration settings.
- Use the DB2 autonomics features to fully utilize dynamic changes in a virtual environment. To be able to detect the changes in the operating environment and act as required, use STMM and all other autonomic DB2 features.
- Utilize vSphere templates and cloning capabilities. vSphere offers efficient tools to create templates and clone virtual machines. DB2 can be cloned easily and deployed by converting the template to a virtual machine.
- Install VMware Tools: To be able to obtain all the benefits of a VMware virtualized environment, you should always install the latest VMware Tools for your guest OS.

5

# Hyper-V

Microsoft Windows Server 2008 Hyper-V technology is a key feature that is found on Microsoft Windows Server 2008. It brings forth many benefits of virtualization, such as server consolidation for Windows Server 2008 users without using and purchasing other third-party products. This chapter describes information related to DB2 virtualization with Hyper-V including:

- High-level overview of Hyper-V architecture and requirements
- High-level overview of Hyper-V and virtual machine setup and configuration
- Configuration of DB2 in a Hyper-V virtual machine environment

## 5.1 Overview

Microsoft Hyper-V hypervisor-based virtualization technology is included in many 64-bit editions of Windows Server 2008. A beta version of Hyper-V shipped with the generally available version of Windows Server 2008 R1 and the finalized version became available through a Windows update (KB950050) on June 26, 2008. Furthermore, a stand-alone version of Hyper-V is available without other Windows server components through the free Microsoft Hyper-V Server 2008 product.

### 5.1.1 Architecture

A hypervisor is a thin layer of software between the underlying hardware and each virtual machine partition that contains an operating system (OS). Hyper-V is a type 1 hypervisor, as it runs directly on the host hardware. Figure 5-1 illustrates the Hyper-V architecture.

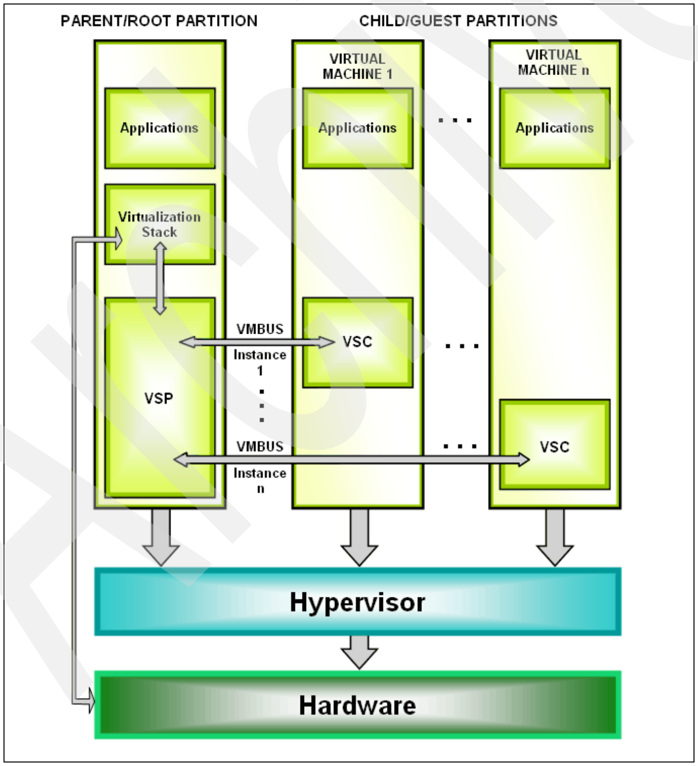



*Figure 5-1 Hyper-V architecture*

Each partition in the system is isolated from one another, providing an area where the virtual machine can execute without interference from other partitions. Regardless of the way that Hyper-V is installed in the system (see 5.2, “Installation and setup of Hyper-V” on page 112, for different ways to get Hyper-V), a parent partition must be present in order to hold the primary operating system installation, as well as the virtualization stack of Hyper-V. This virtualization stack in turn has direct access to the hardware, providing the link between the host hardware and the child partitions on the system. The child partitions holding other virtual machines are created through the parent partition by the use of a hypercall application programming interface (API).

Since child partitions do not have direct access to the hardware, such as the CPU, memory, and other devices, it views these resources through the VMBus as virtual devices. The VMBus allows for communication to occur between the parent and multiple child partitions. It is used by the virtualization service provider (VSP) running on the parent partition and virtualization service client (VSC) running on the child partition to exchange data, requests, and responses.

Hyper-V also has a specific feature in order to speed up I/O processes in child partitions, allowing the use of the VMBus directly by virtual devices. Often described as enlightened, these child partitions require the use of hypervisor-aware virtual device drivers that are already part of the OS or are installed as part of Integration services. See “Disk configuration recommendations for virtual machines” on page 121 for more information.

## 5.1.2 Requirements

Microsoft Hyper-V has specific hardware requirements that must be met:

- A machine containing an x86_64-based (commonly referred to as x64) processor.
- The availability of instructions on the CPU for hardware-assisted virtualization, commonly found in later Intel and AMD produced CPUs with the marketed labels of Intel VT and AMD-V, respectively.
- The ability for the CPU to support the No eXecute (NX) bit, marketed by Intel as the eXecute Disable (XD) bit and by AMD as Enhanced Virus Protection. In addition, hardware Data Execution Prevention (DEP) must be enabled.

For the second and third items in the above list, they must be enabled in the BIOS of the machine. Figure 5-2 shows an image of an IBM System x3650 BIOS, with the appropriate required settings highlighted.

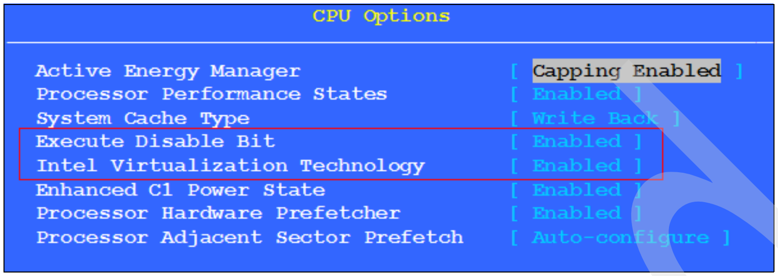



*Figure 5-2 Required CPU options of IBM System x3650 BIOS for Hyper-V*

It is important to note that if the BIOS changes shown in Figure 5-2 are necessary, you must invoke a hard reboot to the system (that is, turn off and then turn back on the system) in order to save the settings correctly.

# 5.2 Installation and setup of Hyper-V

To attain Hyper-V for your Windows-based machine, you must install one of the following:

- Microsoft Windows Server 2008 Standard Edition x64
- Microsoft Windows Server 2008 Enterprise Edition x64
- Microsoft Windows Server 2008 Datacenter Edition x64
- Microsoft Hyper-V Server 2008

Hyper-V is one of many roles that is available within Windows Server 2008. For more information about the Hyper-V role as well as the standalone Hyper-V Server 2008, visit the Microsoft Windows Server 2008 Virtualization with Hyper-V Web site at:

http://www.microsoft.com/windowsserver2008/en/us/hyperv-main.aspx

Also see the Hyper-V Server 2008 Web site at:

http://www.microsoft.com/hyper-v-server/

## 5.2.1 Adding the Hyper-V role in Windows Server 2008

The standalone Hyper-V Server 2008 is similar to a server core installation of Windows Server 2008 Standard Edition with the Hyper-V and with other roles disabled. The Hyper-V role is one of many found in Windows Server 2008 and can be used in conjunction with other roles. To install the Hyper-V role on a fully installed Windows Server 2008 OS:

1. Click **Start** and then **Server Manager** at the top to open the Server Manager window.
2. On the left pane of the Server Manager window, click **Roles**. Then on the right pane click the **Add Roles** option to open the Add Roles Wizard.
3. Click **Next** until you see the Select Server Roles menu. Select **Hyper-V** under Roles, then click **Next** (Figure 5-3).

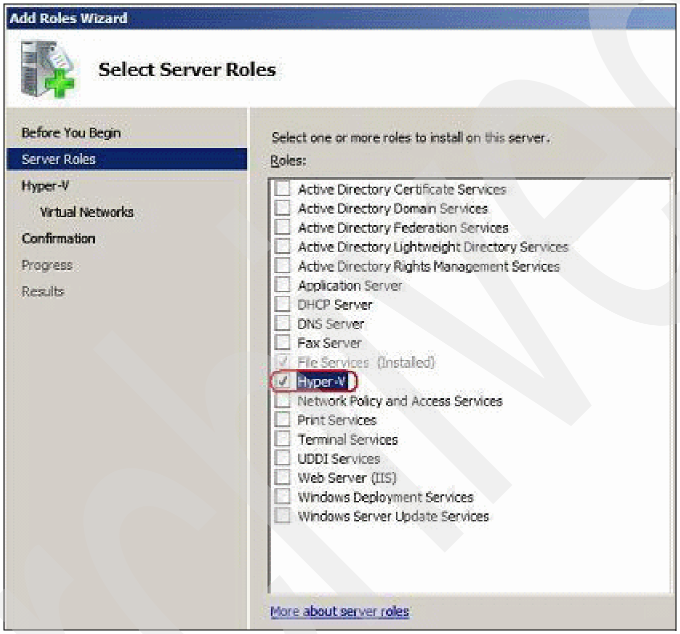



*Figure 5-3 Adding the Hyper-V role to Windows Server 2008 (Images are reprinted with permission from Microsoft Corporation.)*

4. You can optionally view more information about Hyper-V at this point. Click **Next** to move to the next step.
5. The Create Virtual Networks menu appears. Creating a virtual network allows your virtual machines running in Hyper-V to communicate through a network with other computers. You can also change your virtual network settings at a later time by using the Virtual Network Manager. Click **Next** and then **Install** to start the install of the Hyper-V role.
6. Once the installation is finished, click **Close**. Then you are prompted to restart the computer. The installation continues only after you have restarted the computer.

7. After the restart of the computer, the installation continues without further input. Once complete, you can review the installation results and then click **Close** to finish the install.

We recommend, as a normal best practice, that the Windows Server 2008 installation on the host machine only have the Hyper-V role installed on it, and the machine be dedicated to only this role. The inclusion of other roles on the physical server may affect performance of the virtualization server, slowing down all the virtual machines and their installed applications. Additionally, having only the Hyper-V role installed minimizes the need for service patches for the host server OS and, thus, downtime for the virtual machines.

For more information about the installation of the Hyper-V role, refer to your Windows Server 2008 documentation.

## 5.2.2 Network configuration for Hyper-V

Virtual network configuration for Hyper-V is handled by the Virtual Network Manager. This tool can be accessed by going into the Hyper-V Manager (**Start** → **Administrative Tools** → **Hyper-V Manager**) and then clicking the **Virtual Network Manager** option on the Actions pane (the Hyper-V Manager must be connected to the correct server). Figure 5-4 shows the initial menu of the Virtual Network Manager.

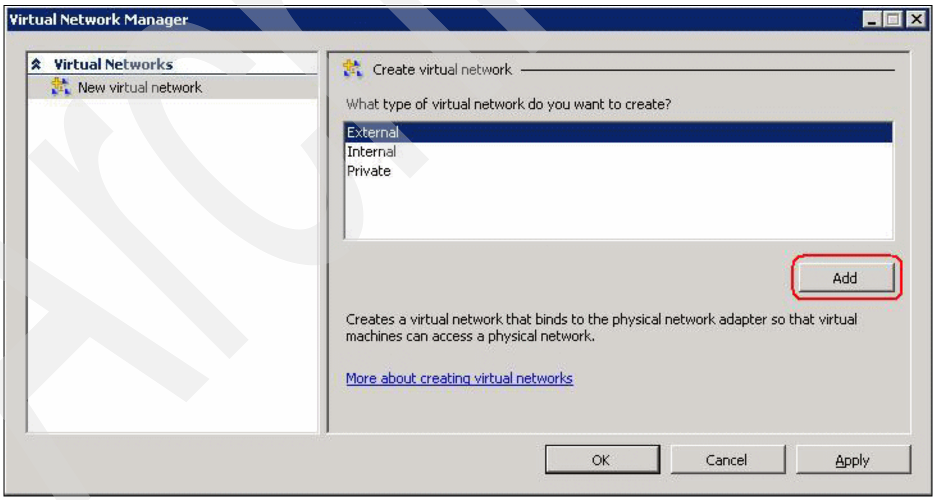



*Figure 5-4 The Virtual Network Manager of Hyper-V (Images are reprinted with permission from Microsoft Corporation.)*

If you have previously added virtual networks, you see a few options on the left-hand side. There is a listing of the current networks available to your machine and an option to create a new virtual network (only the create a new virtual network option will be available if you have not created any virtual networks beforehand). When selected, the new virtual network option displays a list of virtual networks that you can create, labeled external, internal, or private:

- *External* networks allow for communication between the virtual machines residing on the server and the external network. Thus, the virtual machines appear to other machines in the network just like normal physical machines in terms of networking. A specific physical network adapter must be specified for this option. Figure 5-5 illustrates an example of this type of networking.

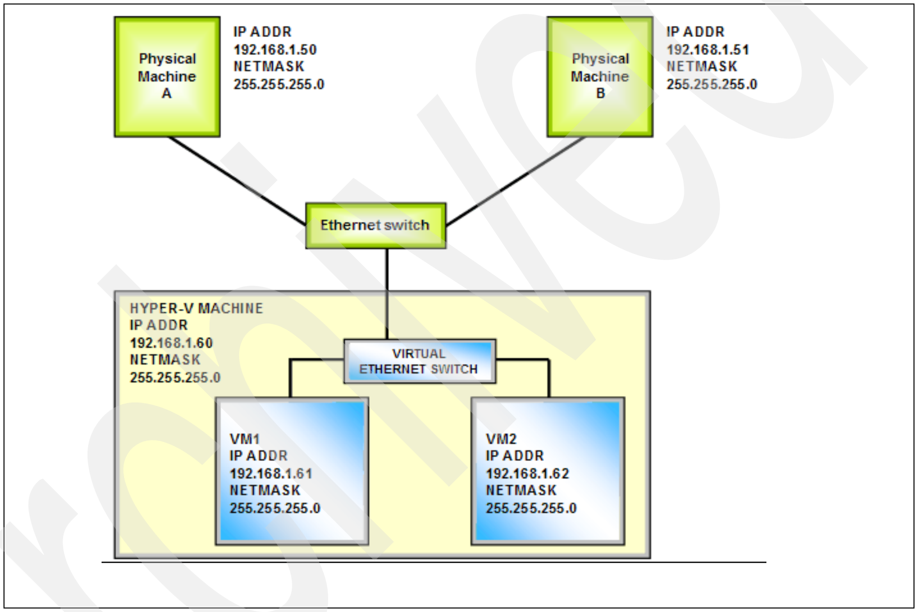



*Figure 5-5 Diagram showing an external network in Hyper-V*

- *Internal* networks allow for communication between the virtual machines residing on the server and the host computer. It does not allow for any communication between the virtual machine and any external machines.
- *Private* virtual machine networks create a virtual network that can only be used by virtual machines on the host machine. The host machine itself does not have any network communications to any of the virtual machines.

We recommend, as a normal best practice, that if virtual networks require external access to the network, that multiple network adapters be installed in the machine, and that at least one network adapter is free only for remote administration of the server (that is, not bound to an external virtual network).

### 5.2.3 Disk configuration for Hyper-V

From a Hyper-V role installation perspective, we recommend that sufficient physical disk and I/O bandwidth be present on the server to serve the different virtual machines' needs. These requirements combined with the knowledge of the expected number of virtual machines and the workloads that will exist on the machine will affect the disks, storage controllers, and RAID configurations that you choose.

## 5.3 Creation and setup of virtual machines on Hyper-V

Once the Hyper-V role has been set up appropriately on the server, the next step is to create and set up the various virtual machines on the system. This section discusses this aspect of the process, as well as recommended settings for virtual hardware for each virtual machine.

### 5.3.1 Creating a virtual machine

In order to create a virtual machine on Hyper-V, follow these steps:

1. Click **Start** → **Administrative Tools** → **Hyper-V Manager** to open the Hyper-V Manager window.
2. Make sure that you are connected to the local computer in the Hyper-V Manager window. If needed, on the Actions pane, click **Connect to Server** and select **Local Computer**, then click **OK**.

3. On the Actions pane, select **New** → **Virtual Machine**. This opens the New Virtual Machine Wizard, as shown in Figure 5-6.

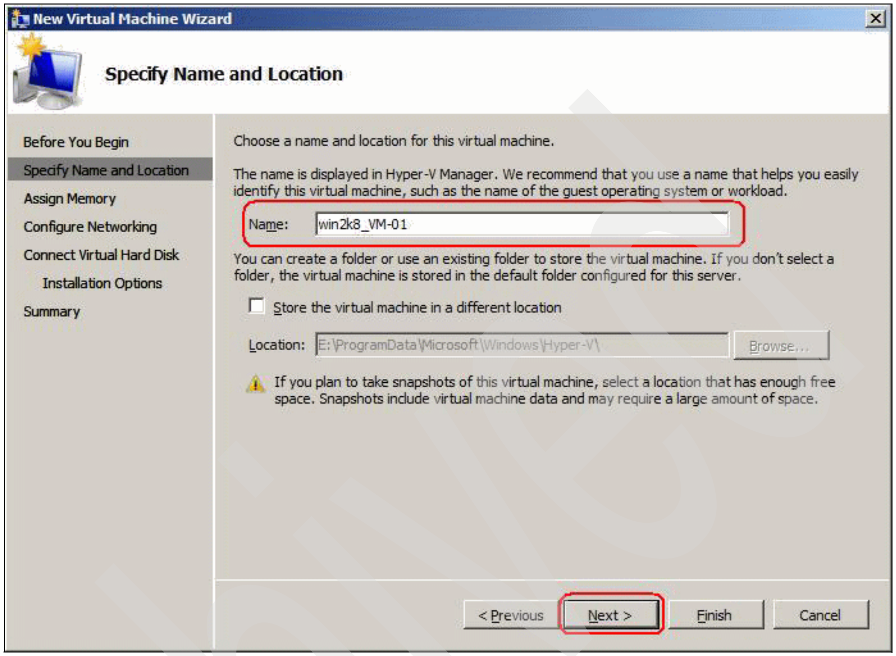



*Figure 5-6 The New Virtual Machine Wizard of Hyper-V (Images are reprinted with permission from Microsoft Corporation.)*

4. Click **Next** to continue to the Specify Name and Location menu where you set the name and location of the virtual machine. Click **Next** to continue.
5. In the Assign Memory menu, you can assign the amount of memory required for the virtual machine. Set up the appropriate amount of memory for the type of workload that you will be doing on this virtual machine. See "Allocating memory for the virtual machine" on page 119 for more information. Click **Next** to continue.
6. In the Configure Network menu, you can assign the specific network connection that you would like to use with the virtual machine. The Virtual Network Manager (see 5.2.2, "Network configuration for Hyper-V" on page 114) determines the type of network connection (external, internal, or private) that it is. Click **Next** to continue.
7. Next is the Connect Virtual Hard Disk menu, where you can choose to create a new virtual hard disk, use an existing virtual hard disk, or attach a virtual hard disk later. See "Disk configuration considerations for the virtual machine" on page 119 for more information about the ramifications of this choice. We recommend that virtual hard disks be configured at a later time (this is due to

the fact that if a disk is created here it must be a dynamic expanding disk type). Click **Next** to continue.

8. On the Installation Options menu, you can choose the method of installation of the OS for the virtual machine. Once finished, click **Next** to continue.
9. Review your choices for the virtual machine on the summary page. When finished, click **Finish** to end the wizard, and the system begins creating the virtual machine.

## 5.3.2 Additional configuration of the virtual machine

Once the virtual machine has been created, there may be additional virtual hardware configuration that is necessary in order to satisfy your requirements. In order to edit these settings, enter the Hyper-V Manager and then select the VM that you would like to manipulate. Right-click this entry and then in the pop-up menu select **Settings**.

### Setting the number of VCPUs in the virtual machine

In order to change the settings concerning the virtual CPUs (VCPUs) on the VM, click **Processor** on the left side of the Settings window to view the options. (In order to change the VCPU settings of a VM, the VM must be turned off.)

By default, each virtual machine that is created in Hyper-V has one VCPU allocated. In order to process the workload that we would expect on the VM, we recommend changing this value to an appropriate number. Each virtual machine can be configured with one, two, or four VCPUs in Hyper-V, up to the limit of physical CPUs on the system or the limits imposed by Hyper-V for a specific guest OS. Thus, if your machine only has two physical CPUs, you are unable to create a VM with four VCPUs in Hyper-V.

With respect to sizing the correct number of VCPUs needed for your workload, existing guides and knowledge from the physical realm also apply to the virtual one. If engaging in server consolidation, most likely existing data on the hardware usage from the workload is already known, and therefore should be leveraged to determine the appropriate sizing.

Resource control for the virtual machine is also set on this page. These settings enable you to balance the resources for this VM versus others that are running on the system. You are able to set the reserves and limits to each VM, and if resources are in contention then the relative weight setting will be a factor in the hypervisor’s determination of resource allocation.

## Allocating memory for the virtual machine

The memory that is allocated for the virtual machine has already been set in the steps leading up to the creation of the VM. However, if you wish to change the memory settings you can do so by clicking **Memory** on the left side of the Settings window. (In order to change the memory settings of a VM, the VM must be turned off.)

In general (as in physical systems also), 4–8 gigabytes (GB) of memory per allocated processor core should suffice for most systems using DB2.

## Disk configuration considerations for the virtual machine

In order to manipulate the virtual hard disks for your virtual machine, click the appropriate virtual disk controller (either IDE or SCSI) on the left side of the Settings window, and click the **Add** button. The window now displays settings that you can now set for the hard disk for the VM. Two main types of hard disk media are presented:

- A virtual hard disk (.vhd) file
- A physical hard disk

### *Virtual hard disk (.vhd) file*

In order to add a virtual hard disk file, click **New** in the middle of the window to bring up the New Virtual Hard Disk Wizard.

1. Click **Next** to bring up the Choose Disk Type menu. Here you can choose from one of three different types of virtual hard disk:
   - Dynamically Expanding
   - Fixed Size
   - Differencing

   See “Differences between virtual hard disk types” on page 120 for a description of these different types as well as a recommendation for different scenarios. Click **Next** to continue.
2. In the Specify Name and Location menu, type in a name for the virtual hard disk and specify a location on the machine’s disk where the virtual hard disk will reside.
3. In the Configure Disk menu, you can choose to create a brand new virtual hard disk or copy the contents of an existing physical disk. If brand new, you must enter a size for the virtual hard disk. Click **Next** and then **Finish** to exit the wizard.

### *Differences between virtual hard disk types*

In step 1 in “Virtual hard disk (.vhd) file” on page 119, you are able to choose from three different types of virtual hard disks for the virtual machine. The following are descriptions of each:

- A dynamically expanding virtual hard disk is one where the virtual hard disk file increases in size based upon the insertion of data onto the virtual disk. When data is deleted from the disk, the .vhd file will not automatically shrink, but can be manually shrunk through a *compact* command in the Edit Virtual Hard Disk Wizard.
- A fixed sized virtual hard disk is one where the virtual hard disk file will be created initially at the set size, and this size will be maintained whether data is inserted or deleted from the virtual hard disk. This pre-allocation in space on the physical disk allows storage I/O operations to be faster than using dynamic expanding virtual hard disks.
- A differencing virtual hard disk produces a child virtual hard disk to store changes made to a parent virtual hard disk. The parent is not altered while the child virtual hard disk grows as changes to the datastored on the virtual hard disk are made.

### *Physical (pass-through) hard disks*

Instead of using a virtual hard disk file that sits on the host machine’s file system, you can map the disk for the VM directly to a physical hard disk or a LUN on a SAN. This then bypasses any overhead that is caused by the host machine’s file system. In order to set up a pass-through disk for your virtual machine:

1. Click **Start** → **Administrative Tools** → **Computer Management**. This brings up the Computer Management window, with multiple choices on the left side. Expand the **Storage** category, and then click **Disk Management**.
2. Look for the disk that you would like to attach to the virtual machine. It must not have a partition or a file system on it, so that it can be used as a raw device. Right-click the disk information on the left side to see a pop-up menu, then click **Initialize Disk**. Select either an MBR or GPT partition style and then click **OK**.
3. Ensure that the disk is placed in an *offline* state. If it is currently online due to the initialize disk process, then right-click the disk information on the left and choose **Offline**.
4. Return to the Settings window for the virtual machine configuration. Select the appropriate IDE or SCSI controller and click **Add**. Select **Physical hard disk** and you should be able to see the physical disk that you want to use in the drop-down menu.
5. Click **OK** to finish adding the disk.

### *Disk configuration recommendations for virtual machines*

Due to the many different options available for the disk configuration for a Hyper-V virtual machine, you must take into account the scenario where the virtual machine will be used and configure the disk accordingly. The following are some recommendations:

- The disk used for the storage of data is separated from the disk used for the installation of the OS in the virtual machine. The primary boot virtual hard disk must be using the virtual IDE controller for the OS to boot and runs in a slower emulated mode until such time that the Integration services are started. Separation of the OS and individual data disks (such as table space and log disks) is also recommended in a physical implementation, and applies in a virtual machine implementation as well.
- We highly recommend that Integration services be installed on the virtual machine (a must if you are using the virtual SCSI controller). The installation of Integration services allows for enlightened drivers to be used, allowing the use of the VMBus directly by virtual devices (see 5.1.1, “Architecture” on page 110, for more information) and therefore increasing I/O performance.
- In scenarios where the virtual machines are being used in a test or development environment, we recommend using virtual hard disks that are dynamic (whether it is using dynamically expanding or differencing virtual hard disks). Both types allow for the most efficient use of the storage capacity that is available. Differencing disks can be used in a scenario where a master copy of the hard disk must be maintained, and only rolling in the changes as needed to the primary virtual hard disk. As both of these types requires allocation of physical disk space as you work on the VM, it is slightly slower than fixed sized virtual hard disks or pass-through disks.
- In scenarios where the virtual machines are being used in a production environment, we highly recommend using fixed sized virtual hard disks or pass-through physical disks for the highest I/O performance. Fixed sized disks do not require you to allocate space on the disk every time that you insert more data onto the drive, and the pass-through disks allow for the minimum overhead of the virtualization layer on the disk I/O. However, due to their fixed size, these disk types will take up space on the physical disks that may be wasted, and pass-through disks will decrease the mobility of the virtual machine should you employ quick migration in your infrastructure. (The use of pass-through disks requires that both Hyper-V host machines have access to the LUN.)
- Finally, the locations of any virtual hard disk files should be contained in separated physical disks so as to not have the disks compete for I/O bandwidth. For example, if two virtual disks are being accessed at the same time and they reside on the same physical disk or LUN, each disk will only receive a maximum of 50% of the total I/O bandwidth. It is advisable to look at

your specific consolidation scenario to make sure that there is a minimum of competition for I/O bandwidth among the physical disks.

## 5.4 DB2 in Hyper-V environments

Once the virtual machine has been created in the Hyper-V environment, we can proceed with the operating system and DB2 install and configuration.

### 5.4.1 Supported OS platforms for DB2 on Hyper-V

Table 5-1 describes the DB2 support for Hyper-V at the time of publication of this book.

*Table 5-1 DB2 supported environments for Hyper-V*

| Hypervisor | Architecture | Minimum guest OS (Windows) | Minimum guest OS (Linux) | Minimum DB2 level |
|---|---|---|---|---|
| MS Windows 2008 SP2 Hyper-V | X64 System with Intel VT or AMD-V | MS Windows Server 2008 SP2 | Not supported | DB2 9.5 FP4<br>DB2 9.7 GA |

Note that the Quick Migration feature of Hyper-V is not supported at the time of publication.

For the latest information about DB2-supported environments for virtualization, visit the following Web site:

http://www.ibm.com/developerworks/wikis/display/im/DB2+Virtualization+Support

### 5.4.2 Installation of operating system and DB2

Once the virtual machine hardware configuration is complete, the operating system and DB2 can be installed inside. For more information about the installation of Windows Server 2008, refer to your Windows Server 2008 documentation.

For more information about the installation of DB2 on Windows, refer to the following links in the DB2 Information Center:

- DB2 9.5

  http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.qb.server.doc/doc/t0052773.html

- DB2 9.7

  http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.qb.server.doc/doc/t0052773.html

## 5.4.3 Configuration of DB2 in Hyper-V environments

Once the virtual machine has been set up correctly based on the aforementioned best practices from both physical and virtual environments, the configuration of the database in a virtual machine is not unlike one that is set up in a physical installation.

### Use of autonomics with DB2

Various autonomic features have been introduced throughout the different versions of DB2. These autonomic technologies can result in ease of use as well as performance enhancements, as DB2 can adjust to the virtual computing resources that it is given.

#### *Self Tuning Memory Manager*

Self Tuning Memory Manager (STMM) was introduced in DB2 9 and simplifies the task of memory management and configuration for the database by automatically configuring memory settings based upon available resources and workload needs. By default STMM is turned on with many different memory consumers within newly created databases since DB2 9.1. Because of this, the system will automatically adjust the memory configuration based on need, thereby increasing performance by using the memory resources at hand in the most efficient way. For more information about STMM, refer to the DB2 Information Center's topic on self-tuning memory:

http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.perf.doc/doc/c0024366.html

#### *Configuration advisor*

When creating new databases, the configuration advisor can be used to obtain recommendations for the initial values of certain configuration parameters for the database such as buffer pool sizes. These recommendations are based upon input that is given by the user as well as system information that is automatically detected. By default, the configuration advisor is used on any new databases created with the CREATE DATABASE command. For more information about the configuration advisor in DB2, refer to the DB2 Information Center's topic on the configuration advisor:

http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.dbobj.doc/doc/c0052481.html

### *Automatic maintenance*

In order to ease administration and maintain peak performance without manual intervention, automatic maintenance is available for DB2 databases. Specifically, automatically run reorganizations of tables or indexes and automatic statistics collection are important to maintain performance of the database, reducing fragmentation of the storage disks and providing correct profiling data for the DB2 optimizer, respectively. These maintenance activities can be set into either online or offline maintenance windows, a time period that the user can define for running these automated tasks. For more information about automatic maintenance, refer to the DB2 Information Center's topic on automatic maintenance:

http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.dbobj.doc/doc/c0021757.html

## 5.5 Conclusion

Microsoft Windows Server 2008 Hyper-V technology brings forth virtualization benefits without the need for the purchase of other third-party software. Both DB2 9.5 and 9.7 are supported on Hyper-V and can be successfully deployed in such an environment given the aforementioned best practices.

Here we summarize the best practices for running DB2 with Hyper-V:

- Dedicate the machine to only the Hyper-V role if possible to reduce the needs of resources and software updates and maintenance.
- Use multiple network adapters to decrease network resource contention, and dedicate one network adapter for remote administration of the server and virtual machines.
- Use dynamic virtual hard disks when in test and development environments to increase efficient use of storage, while using fixed sized virtual hard disks when in production environments to increase I/O performance.
- Once the OS is set up correctly, make sure to install the Integration services for Hyper-V in the virtual machine for better performance.
- Ensure that DB2 autonomic technologies are turned on in the virtual machine, as this provides many benefits, including performance and ease-of-use enhancements.

A

# Extension of virtualization technologies

Virtualization is becoming more popular due to its increasing ease of efficiently utilizing and managing resources. With virtualization additional uses have emerged as an extension of this technology. The most notable are:

- Virtual appliances (VAs)
- Cloud computing

In this appendix we provide a general overview of virtual appliances and cloud computing. Along with this overview, a brief discussion is included about how DB2 can take advantage of these two new emerging virtualization technologies.

# Virtual appliances

The word *appliance* is defined as an *instrument or device designed to perform a specific task*. When people first hear the word appliance, they commonly think of a household appliance such as a refrigerator or a stove. These two devices have been specifically designed and built to handle food. For example, the refrigerator is used to keep food cold, which is mainly to keep the food from spoiling before consumption.

The word appliance is now being used in the computing world. The first reference to the word appliance was used to identify a computer server that was dedicated to perform a specific task. These appliances are built using dedicated hardware components. These components were either specifically built for the server or configured from existing components to use in the server. This type of appliance is loosely referred to as a hardware or computer appliance.

The term *hardware appliance* is now more broadly used to refer a complete solution that includes the defined hardware stack and the software stack. Early on when the hardware appliance used a nonstandard hardware stack, the operating system and applications had to be customized to be used on the server. Now software stacks are created to be used on industry-standard hardware. This type of software stack is referred to as a software appliance. The software appliance typically includes the setup and configuration operations as part of the appliance. Evolving from the software appliance with the use of virtual machines are virtual appliances.

A virtual appliance, in the general sense, is a preconfigured virtual machine image that has integrated the application software, operating system, and virtual resources. Essentially, a virtual appliance can be broken down into two parts:

- A software appliance
- A virtual machine

A virtual machine, which is created from virtual resources, still requires the operating system and application software to be installed before it can be fully utilized. At a minimum, the operating system must be installed and configured before the virtual machine can be functional. This usually involves manually installing, configuring, and testing both the operating system and the application software in the virtual machine environment. The same method is used for a dedicated or stand-alone server, which in comparison would be considered a non-virtual machine. With the introduction of software appliances, the software stack, which could also include the operating system, is pre-packaged and pre-configured into one image. This image is then used to deploy the software solution to the intended server, which can be either a dedicated server or a virtual machine.

By extending the software appliance concept to encompass the virtual machine, a virtual machine image can be generated. This image creation includes the virtual machine setup along with the configuration of the operating system and application software. The use of a virtual appliance simplifies many aspects of offering a complete solution. The aspects that are simplified include installation, configuration, development, deployment, and maintenance.

## Installation and configuration

A virtual appliance consists of a virtual machine image that contains the application software and operating system, along with the virtual resources for the targeted virtual machine. This also includes the setup and configuration of the application software, operating system, and the virtual resources.

Without using a virtual appliance the virtual machine would first have to be manually set up and configured, which includes the virtual resources. After this the operating system still must be manually installed and configured. Once the virtual machine is operational the application software also must be manually installed and configured to complete the solution. Manually installing and configuring each component of the software stack can be time consuming and tedious, especially when deploying more than one instance of the software solution. Also, since the underlying physical hardware is abstracted in the virtual machine, the operating system and the application software are configured using the virtual resources. This lessens the time to configure the software stack since there is no dependence on the underlying hardware or different hardware device drivers. So a virtual appliance simplifies the setup of the virtual resources along with installation and configuration of the operating system and application software.

## Development and deployment

As mentioned above, a virtual appliance simplifies the installation and configuration of a software solution. This simplification helps ease the development, packaging, and deployment of the solution. As a result, the benefits of using virtual appliances is realized more during the deployment phase of solution.

In the development phase, instead of spending time configuring multiple environments for a specific instance of a solution, this time can be spent configuring one instance of a solution that can be deployed to multiple environments. While time is spent in the development phase to set up and configure the virtual resources and then to install and configure the software stack, this is the only time that these tasks are performed. Once the solution is completed the virtual machine image can be packaged, which can then be deployed to multiple environments that can use the virtual machine image. The deployment of the virtual appliance only requires that the virtual machine image be applied to the environment, with no additional need to set up or configure the environment.

To further assist in the packaging and the deployment of virtual appliances a standard has been established. The Distributed Management Task Force (DMTF) has published an Open Standard: Open Virtualization Format (OVF). Top contributing companies in the field of virtualization technologies, including IBM, created the Open Standard in a joint effort. The DMTF describes the specification as "an open, secure, portable, efficient, and extensible format for the packaging and distribution of software to be run in virtual machines." For more information about the OVF Specification the complete document can be downloaded from the DMTF Web site:

http://www.dmtf.org/standards/published_documents/DSP0243_1.0.0.pdf

## Maintenance

When using virtual appliances the software stack is treated as a single unit, as opposed to the individual software components. Since the software stack has been deployed as a single unit within the virtual appliance, updates and patches can be applied in the same manner. This eliminates the need to manually update each individual software component in the software stack. Also, testing of the updates and patches only must be performed once since the software stack is not dependent on the underlying physical hardware, which can be different from server to server where the virtual appliance is deployed. This is in contrast to software appliances where updates and patches would need to be tested on each hardware server where the solution is deployed.

In summary, a virtual appliance is a preconfigured virtual machine. The use of virtual appliances helps ease the installation and configuration of the application software and operating system with the virtual resources within the virtual machine. This allows a software solution to be developed, packaged, and delivered more easily than using traditional methods to deploy a solution. At the same time the virtual appliance is less complicated to maintain and deploy.

# DB2 virtual appliances

Users of DB2 in a virtualized environment can greatly benefit from the use of virtual appliances. These VAs offer prebuilt environments with the OS and DB2 installed and configured, providing many benefits including:

- Rapid deployment of a set configuration in a data center environment
- A prepared image with all configurations necessary for demonstrations and trials every time
- A stand-alone and complete development environment on a desktop without affecting other applications and tools on the machine

## Extensions of DB2 virtual appliances

DB2 virtual appliances in their standard form are very useful for many different scenarios. However, there will be instances where these VAs can be improved upon for specific deployments or tasks. For example, an independent software vendor (ISV) may wish to include their software product that works on top of DB2 in a demonstration VA. In these cases of repackaging, DB2 virtual appliances can be extended to include such additional software, easing the work required to build a virtual appliance, and bringing forth all the benefits of virtualization and appliances very quickly.

Virtual appliances are available for all editions of DB2. To attain the DB2 Express-C versions of the VA, visit the following virtualization portal Web site:

`http://www.ibm.com/developerworks/wikis/display/im/IBM+Virtual+Appliances`

For other editions of DB2 virtual appliances, contact your local IBM representative or e-mail askdata@ca.ibm.com.

# Cloud computing

Cloud computing is a computing model that provides a hosted service over the internet using a connected device. It is an evolving model that has been built upon grid computing, utility computing, and software as a service (SaaS). The service that is offered allows scalable access to computer resources. All of the computer resources are invisible to the users of the service since the infrastructure is managed by a provider. Also, the use of the computer resources is charged on demand by the provider to the users of the hosted service.

The main aspect of cloud computing is the hosting of the physical infrastructure. This infrastructure is made up of the computing resources that provide the hosted services. Instead of managing the infrastructure locally, the infrastructure is managed remotely, or off-site, by a third-party provider, which eliminates the initial cost of the physical computer resources, along with the cost and time to manage and maintain the infrastructure since it is taken over by a third-party provider.

In a sense cloud computing is another form of virtualization, since now the computer resources are abstracted or *virtualized* to the users of the service. The hosted service is provided using any type of connected device through the internet. In fact, the name cloud computing is derived from how the internet is depicted as a cloud in diagrams. The more familiar services using cloud computing are offered using a Web portal or client software. However, now services are being offered that allow the user to create a virtual machine or virtual appliance that can be hosted remotely using virtualization in the cloud.

The use of virtualization allows the virtual image to be created and activated. At the same time, virtualization allows the provisioning of the computer resources used in the infrastructure. This is all within the scope of the cloud computing paradigm. The computer resources offered in the cloud include CPU and memory, but can also include storage, all of which will be available on demand as capacity needs change. As capacity needs change, the provider can facilitate these changes using virtualization within the infrastructure to allocate or deallocate resources dynamically. There are two types of distinct clouds:

- A public cloud
- A private cloud

The most common cloud is a public cloud, or external cloud. A *public cloud* is a cloud that is offered to the public by a third-party provider, where the resources are shared among the users of the hosted service. The use of the resources can be charged based on the utilization. A *private cloud*, or an internal cloud, is very similar to a public cloud, and is based on clouds being built within a private network. The service provided in a private cloud are usually offered to select users. But using a private cloud means that the infrastructure is not hosted by a third-party provider, which goes against the traditional meaning of cloud computing. So hybrid clouds, or virtual clouds, are used to build private clouds using public clouds.

## Cloud computing and DB2

DB2 and other IBM Software Group products can be found in various cloud offerings such as the IBM Cloud and on Amazon Web Services (AWS) Elastic Compute Cloud (EC2). Both of these infrastructures use virtualization as the main component in abstracting the hardware resources away from the customer, and the images that are contained in these clouds are based on the fundamentals of virtual appliances.

For more information about DB2 products on AWS EC2, visit the following Web sites:

http://aws.amazon.com/ibm/
http://aws.amazon.com/solutions/featured-partners/ibm/
http://www.ibm.com/developerworks/spaces/cloud/

For more information about IBM Cloud and other IBM technologies for cloud computing, visit:

http://www.ibm.com/ibm/cloud/

# 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 publications

For information about ordering these publications, see “How to get Redbooks publication” on page 136. Note that some of the documents referenced here may be available in softcopy only.

- *PowerVM Virtualization on IBM System p: Introduction and Configuration Fourth Edition*, SG24-7940

## Other publications

These publications are also relevant as further information sources.

### IBM DB2 for Linux, UNIX, and Windows manuals

- *Administrative API Reference,* SC27-2435
- *Administrative Routines and Views,* SC27-2436
- *Call Level Interface Guide and Reference, Volume 1,* SC27-2437
- *Call Level Interface Guide and Reference, Volume 2,* SC27-2438
- *Command Reference,* SC27-2439
- *Data Movement Utilities Guide and Reference,* SC27-2440
- *Data Recovery and High Availability Guide and Reference,* SC27-2441
- *Database Administration Concepts and Configuration Reference,* SC27-2442
- *Database Monitoring Guide and Reference,* SC27-2458
- *Database Security Guide,* SC27-2443
- *DB2 Text Search Guide,* SC27-2459
- *Developing ADO.NET and OLE DB Applications,* SC27-2444
- *Developing Embedded SQL Applications,* SC27-2445
- *Developing Java Applications,* SC27-2446

- *Developing Perl, PHP, Python, and Ruby on Rails Applications*, SC27-2447
- *Developing User-defined Routines (SQL and External)*, SC27-2448
- *Getting Started with Database Application Development*, GI11-9410
- *Getting Started with DB2 Installation and Administration on Linux and Windows*, GI11-9411
- *Globalization Guide*, SC27-2449
- *Installing DB2 Servers*, GC27-2455
- *Installing IBM Data Server Clients*, GC27-2454
- *Message Reference Volume 1*, SC27-2450
- *Message Reference Volume 2*, SC27-2451
- *Net Search Extender Administration and User's Guide*, SC27-2469
- *SQL Procedural Languages: Application Enablement and Support*, SC23-9838
- *Partitioning and Clustering Guide*, SC27-2453
- *pureXML Guide*, SC27-2465
- *Query Patroller Administration and User's Guide*, SC27-2467
- *Spatial Extender and Geodetic Data Management Feature User's Guide and Reference*, SC27-2468
- *SQL Procedural Language Guide*, SC27-2470
- *SQL Reference, Volume 1*, SC27-2456
- *SQL Reference, Volume 2*, SC27-2457
- *Troubleshooting and Tuning Database Performance*, SC27-2461
- *Upgrading to DB2 Version 9.7*, SC27-2452
- *Visual Explain Tutorial*, SC27-2462
- *What's New for DB2 Version 9.7*, SC27-2463
- *Workload Manager Guide and Reference*, SC27-2464
- *XQuery Reference*, SC27-2466
- *Installing and Configuring DB2 Connect Personal Edition*, SC27-2432
- *Installing and Configuring DB2 Connect Servers*, SC27-2433
- *DB2 Connect User's Guide*, SC27-2434
- *Information Integration: Administration Guide for Federated Systems*, SC19-1020-02

- *Information Integration: ASNCLP Program Reference for Replication and Event Publishing*, SC19-1018-04
- *Information Integration: Configuration Guide for Federated Data Sources*, SC19-1034-02
- *Information Integration: SQL Replication Guide and Reference*, SC19-1030-02
- *Information Integration: Introduction to Replication and Event Publishing*, SC19-1028-02


## Online resources

These Web sites are also relevant as further information sources:

### *DB2*

- DB2 Information Center

  `http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/`
- Database and Information Management home page

  `http://www.ibm.com/software/data/`
- DB2 Technical Support

  `http://www.ibm.com/software/data/db2/support/db2_9/`
- DB2 Product Family Library

  `http://www.ibm.com/software/data/db2/library/`
- DB2 developerWorks

  `http://www.ibm.com/developerworks/db2/`
- DB2 for Linux

  `http://www.ibm.com/software/data/db2/linux/`
  `http://www.ibm.com/software/data/db2/linux/validate/`
- DB2 Universal Database V9 Application Development

  `http://www.ibm.com/software/data/db2/ad/`
- Planet DB2

  `http://www.planetdb2.com/`

### *IBM Power Systems*

`http://www-03.ibm.com/systems/power/`

### *Other*

- VMware

  http://www.vmware.com/
- Microsoft Hyper-V

  http://www.microsoft.com/virtualization/

## How to get Redbooks publication

You can search for, view, or download Redbooks publications, Redpapers publications, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks publications, at this Web site:

**ibm.com**/redbooks

## Help from IBM

IBM Support and downloads

**ibm.com**/support

IBM Global Services

**ibm.com**/services

# Index

IBM Redbooks

DB2 Virtualization

(0.2,Äùspine)
0.17,Äù<->0.473,Äù
90<->249 pages

IBM®

Redbooks®

# DB2 Virtualization

**Learn setting up and configuring DB2 on PowerVM, VMware, and Hyper-V**

**Leverage virtualization technologies**

**See best practices**

Server virtualization technologies are becoming more popular to help efficiently utilize resources by consolidating servers. IBM, the first company that developed and made available the virtual technology in1966, offers advanced, powerful, reliable, and cost-saving virtualization technologies in various hardware and software products including DB2 for Linux, UNIX, and Windows. This IBM Redbooks publication discusses and describes using IBM DB2 9 with server virtualization.

We start with a general overview of virtualization and describe industry server virtualization technologies available in the market. Following the virtualization concept, we describe in detail the setup, configuration, and management of DB2 with three leading server virtualization technologies: IBM System p with PowerVM, VMware, and Hyper-V.

In addition, we discuss the DB2 features and functions that can take advantage of using server virtualization. These features are put into practice when describing how to set up DB2 with the three virtualization technologies discussed in this book. This book also includes a list of best practices from various tests performed while using these virtualization technologies. These best practices can be used as a guideline or a reference when setting up DB2 using these virtualization technologies.

**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



SG24-7805-00 ISBN 0738433438