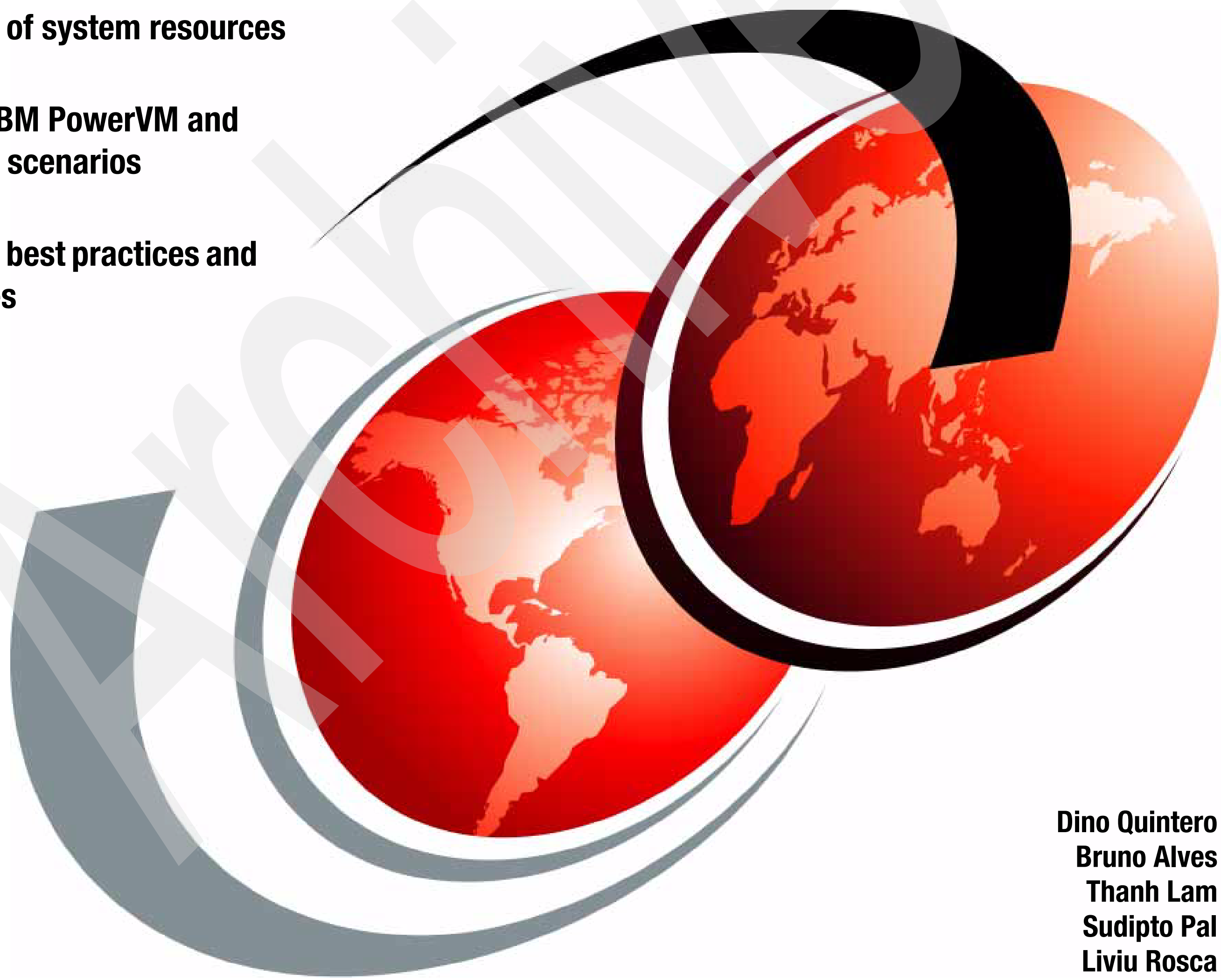
IBM

Exploiting IBM PowerVM Virtualization Features with IBM Cognos 8 Business Intelligence

Demonstrates how to maximize utilization of system resources

Includes IBM PowerVM and Cognos BI scenarios

Illustrates best practices and procedures



Dino Quintero
Bruno Alves
Thanh Lam
Sudipto Pal
Liviu Rosca

Redbooks

International Technical Support Organization

Exploiting IBM PowerVM Virtualization Features with IBM Cognos 8 Business Intelligence

September 2010

SG24-7842-00

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

First Edition (September 2010)

This edition applies to IBM AIX 6.1 TL4, IBM Cognos 8.4, IBM WebSphere Application Server 6.1, IBM DB2 9.7, and Tivoli Directory Server 6.2.

Contents

Notices

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

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

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

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

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

COPYRIGHT LICENSE:

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

Trademarks

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

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

Active Memory™	LoadLeveler®	Redbooks®
AIX 5L™	Power Systems™	Redpaper™
AIX®	POWER6®	Redbooks (logo) ®
Cognos®	POWER7™	System x®
DB2®	PowerVM™	Tivoli®
HACMP™	POWER®	WebSphere®
IBM®	pSeries®	

The following terms are trademarks of other companies:

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

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

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

This IBM® Redbooks® publication addresses topics to leverage the virtualization strengths of the IBM Power platform to solve customer system resource utilization challenges and maximize system throughput and capacity.

This IBM Redbooks publication will help you leverage the strengths of the POWER® platform, provide implementation scenarios with Cognos® 8 Business Intelligence (BI) with the comprehensive set of the IBM PowerVM™ virtualization features, and identify and document best practices for exploiting the IBM PowerVM virtualization features within Cognos BI deployments to maximize utilization of system resources and maximize Cognos throughput and capacity.

This book is targeted toward technical professionals (BI consultants, technical support staff, IT architects, and IT specialists) responsible for providing business intelligence solutions and support for Cognos BI on POWER systems.

The team who wrote this book

Figure 1 Sudipto Pal, Dino Quintero (Project Leader), Liviu Rosca, Bruno Alves, and Thanh Lam

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

Dino Quintero is a Technical Project Leader and a IT Generalist with the International Technical Support Organization in Poughkeepsie, New York. His areas of expertise include enterprise continuous availability planning and implementation, enterprise systems management, virtualization, and clustering solutions. He is currently an Open Group Master Certified IT Specialist - Server Systems. He holds a master's degree in Computing Information Systems and a Bachelor of Science in Computer Science from Marist College.

Bruno Alves is an Cognos Application Developer in the Cognos CoC team of Brazil. He has worked for IBM for two years, and is especially involved with Cognos technology. He is an IBM Certified Solution Expert - Cognos 8 BI and holds a bachelor's degree in Management Information Systems from the Centro Universitário da Cidade.

Thanh Lam is a Cluster System Test Specialist based in Poughkeepsie, New York. For 20 years, he has been working as a Software Engineer in the High Performance Computing area with extensive experience in parallel software products, such as LoadLeveler® Job Scheduling and Resource Management, Message Passing Interfaces in Parallel Environments, programming and debugging tools. He works with hardware platforms including Power Systems™, Blade Centers, and Blue Genes. He joined IBM in 1988, and then started working in the early Scalable Parallel Development Lab and Parallel System Support Programs in Kingston, New York. He has a Doctor of Professional Study in Computing degree from Pace University, White Plains, New York.

Sudipto Pal is a Technical Consultant in the STG ISV enablement team working for IBM Global Services in India. He works with the software group to benchmark several products with IBM virtualization features. He has published several white papers with collateral for WebSphere® Application Server and WebSphere MQ server with virtualization features such as LPAR, WPAR, and AMS. He has experience in coding using C/C++ under the UNIX® environment for several networking protocols, such as LDPA, DNS, DHCP, SNMP.

Liviu Rosca is a Senior PS and IT Specialist with IBM Global Technology Services Romania. He has worked for IBM for eight years on pSeries® and Power systems, AIX®, and HACMP™. His area of expertise include pSeries, AIX, HACMP, networking, security, and telecommunications. He is an IBM Certified AIX 5L™ and HACMP System Administrator, as well as CCNP. He teaches AIX and HACMP classes, and has co-authored six other IBM Redbooks publications.

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

Ella Buslovic, Rich Conway, David Benin, Don Brennan, Scott Vetter
International Technical Support Organization, Poughkeepsie Center

Marius Ileana, Andrei Socoliuc, Cristian Stanciu
IBM Romania

Beth Hoffman, John Mullin, Chris Tulloch, Duane Witherspoon, Godfrey Lee, Scott Masson, Robert Hatfield, Khalid Filali-Adib, Ernest M. Gamble, Ian Robinson, Linda Flanders
IBM US

Mark McFadden, Eric Racine, Claude Tennier
IBM Canada

Luciana Sales Cruz, Johnatton Dos Santos, Leandro Hiroshi Hobo
IBM Brazil

Now you can become a published author, too!

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

  ibm.com/redbooks

- Send your comments in an 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

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBM-Redbooks

- Follow us on twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

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

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

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

- Stay current on recent Redbooks publications with RSS Feeds:

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

1

Introduction to IBM Power Systems and IBM Cognos 8 BI

In this chapter, we introduce the IBM Power Systems servers with and their PowerVM virtualization features, and describe the IBM Cognos 8 Business Intelligence (BI) solution.

This chapter discusses the following topics:

- "IBM Power Systems and PowerVM virtualization features" on page 2
- "Introduction to IBM Cognos 8 Business Intelligence" on page 4

1.1 IBM Power Systems and PowerVM virtualization features

IBM Power Systems and IBM PowerVM virtualization technology allows you to consolidate applications and servers by using partitioning and virtualized system resources to provide a flexible and dynamic IT infrastructure.

PowerVM delivers unprecedented virtualization strength for AIX, IBM i, and Linux® environments running on IBM POWER processor-based systems.

This section provides an overview of the hardware and software features that coexist to exploit PowerVM.

1.1.1 Logical partitioning

Logical partitioning allows you to define logical servers that can accommodate an instance of an operating system. The logical server is allocated various amounts of system resources such as processors, memory, and I/O adapters. An LPAR is not an operating system. Instead, it is a complete virtualized workload, including operating system, applications, and configuration settings.

1.1.2 Workload partitions (WPARs)

WPARs provide a virtualized operating environment to manage separate workloads within a logical partition. Each WPAR is a secured and isolated environment for the hosted application. LPARs provide more flexibility in managing applications and workloads.

1.1.3 Shared processor pools and micropartitioning

Early versions of Power Systems servers allowed you to assign a partition a specific number of whole processors. With later developments, physical processors can be grouped in processor pools. A shared processor pool contains at least one physical processor and can be increased to include the entire installed processing capacity of the system.

The processing power of a shared processor pool can be divided and allocated to multiple logical partitions. This feature is named *micropartitioning*. With micropartitioning a single logical partition can be assigned at least 10% of the processing power corresponding to a physical processor up to the entire capacity of the physical shared processor pool.

Changes to the processing capacity allocated to a logical partition can be done with a granularity of 1% of the processing power that corresponds to a physical processor.

1.1.4 Dynamic reconfiguration

IBM Power Systems allows you to dynamically move system resources between partitions without rebooting the operating system running in the partition. Resources that can be moved between partitions include:

- Processors
- Memory
- I/O adapters

This capability is known as *dynamic reconfiguration* or *dynamic LPAR*.

1.1.5 IBM POWER hypervisor

All capabilities of IBM Power Systems rely on a foundation layer named POWER hypervisor. The hypervisor enables the hardware to be divided among multiple partitions, while ensuring partition isolation and enforcing data security.

1.1.6 Virtual Ethernet

The POWER hypervisor also provides inter-partition communication that enables using virtual Ethernet interfaces.The POWER hypervisor implements the Ethernet transport mechanism as well as an Ethernet switch function that supports virtual LAN (VLAN) capability. VLAN allows secure communication between logical partitions without the need for a physical I/O adapter or cabling.

1.1.7 Virtual SCSI

Another way to increase hardware utilization is by using virtual SCSI. Virtual SCSI provides a means for LPARs without physical I/O hardware to share disks and storage adapters with partitions that have physical I/O hardware. These adapters and disks can then be shared among multiple partitions, thus increasing utilization. A single physical disk I/O adapter and associated disk subsystem can be used by many logical partitions on the same server.

1.1.8 Virtual I/O Server

Virtual SCSI is made possible by the Virtual I/O Server that provides access to storage devices.

The ability to securely share Ethernet bandwidth across multiple partitions increases hardware utilization. On logical partitions, it is also possible to define virtual Ethernet interfaces that can communicate with Shared Ethernet Adapters defined on the Virtual I/O Server. The VIO server acts as a bridge for the network traffic originating from virtual Ethernet networks and directs this traffic to outside physical Ethernet networks.

VIOS owns physical resources and shares them with client LPARs. A physical adapter assigned to a VIOS partition can be shared by one or more other partitions. VIOS eliminates the need for dedicated network adapters, disk adapters, and disk and tape drives in each client partition.

1.1.9 Live Partition Mobility

Live Partition Mobility allows you to migrate partitions that are running AIX and Linux operating systems and their hosted applications from one physical server to another without disrupting the infrastructure services. The migration operation maintains complete system transaction integrity. The migration transfers the entire system environment, including processor state, memory, attached virtual devices, and connected users. For additional details regarding these features, see the *PowerVM Virtualization on IBM System p: Introduction and Configuration Fourth Edition*, SG24-7940.

1.1.10 Active Memory Sharing

Aside from supporting memory dedicated to a logical partition, Power Systems allows you to share memory among logical partitions on a single server. This feature is named Active

Memory™ Sharing and allows you to optimize the overall physical memory usage in the pool. Partitions sharing memory and partitions with dedicated memory can coexist on the same system.

1.1.11 Active Memory Expansion

With POWER7™ systems, a new feature named Active Memory Expansion is available. Active Memory Expansion employs memory compression technology to transparently compress in-memory data, allowing more data to be placed into memory and thus expanding the memory capacity of POWER7 systems.

1.2 Introduction to IBM Cognos 8 Business Intelligence

IBM Cognos 8 Business Intelligence enables decision makers across an organization to solve performance management problems and improve performance. It is an enterprise-class solution that delivers the complete range of BI capabilities (reporting, analysis, dashboarding, business event management, and scorecards) on a single service-oriented architecture (SOA).

IBM Cognos 8 Business Intelligence is a segment within the IBM Information On Demand (IOD) (Figure 1-1 on page 5). The strategy of the IOD focuses on unlocking the business value of information for competitive advantage. The IOD focuses on four key components:

- The ability to manage data and content to reduce the costs associated with managing information, provide controlled accessibility, and address retention and compliance requirements.
- The ability to use data and content as part of the individual business processes and applications across the enterprise, optimizing the performance of applications and improving decision making.
- Establish an accurate, trusted view of information across these processes and applications. Drive more consistent information across the enterprise and support analytic and other requirements to use information coming from different sources. Provide a flexible architecture that can leverage all existing investments. Accurate, trusted information is established through information integration, data warehousing, and master data management.
- IBM Cognos 8 Business Intelligence provides the ability to leverage trusted information to build plans, understand how business is performing, and focus on optimizing performance across the enterprise.

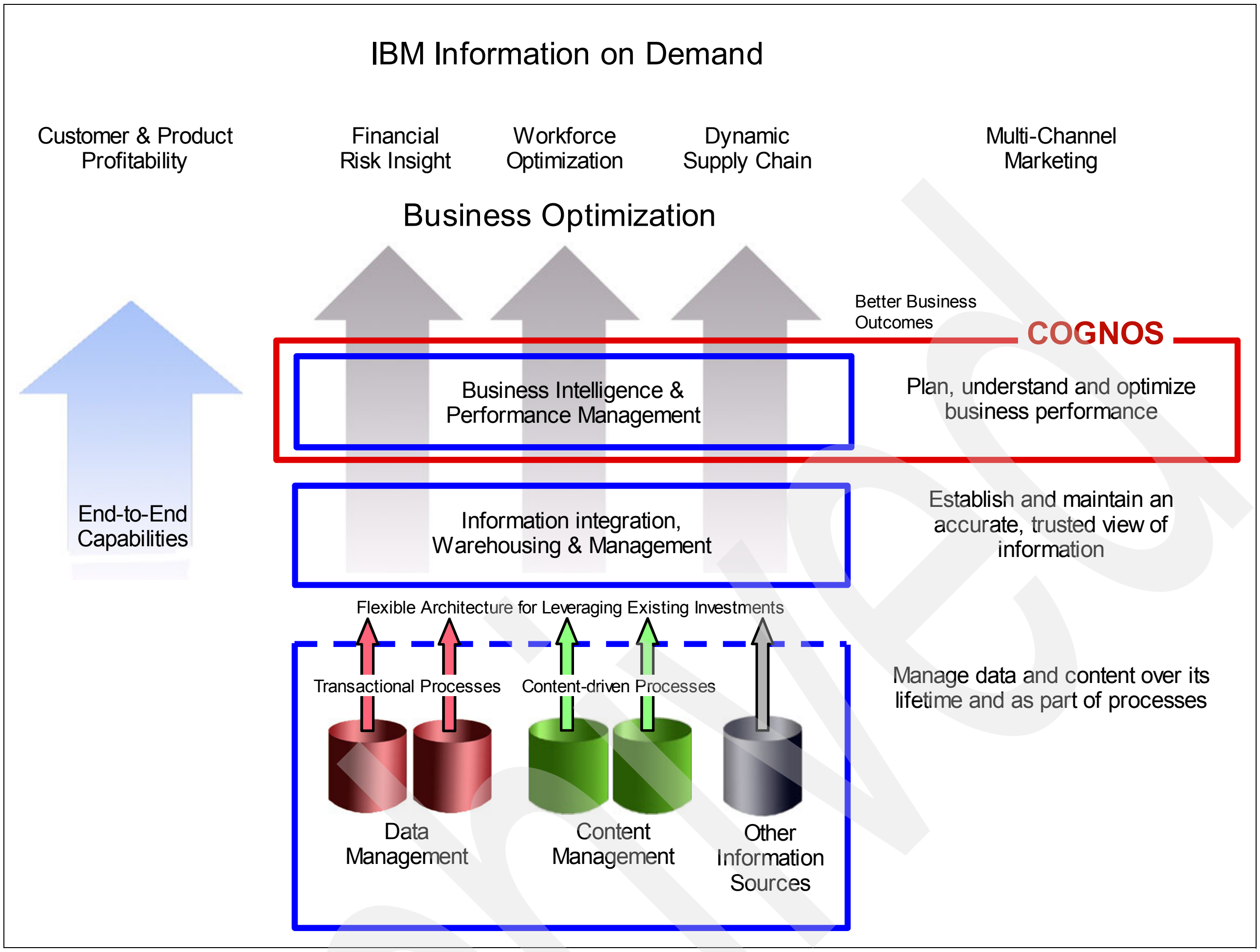


Figure 1-1 IBM Information On Demand (IOD) high-level overview

1.2.1 Role of Cognos in business intelligence

Many factors affect business performance. One of the most critical factors is the decision making that happens in an organization from senior executives to department managers to individual contributors. Decision makers base their decisions on the information available. IBM Cognos BI helps organizations improve performance by empowering the decision makers with the correct information, in the correct format, at the correct time (Figure 1-2). Employees at every level in the company can look at the same report, see where that data comes from, and be confident that the information that they are reading is accurate and appropriate.

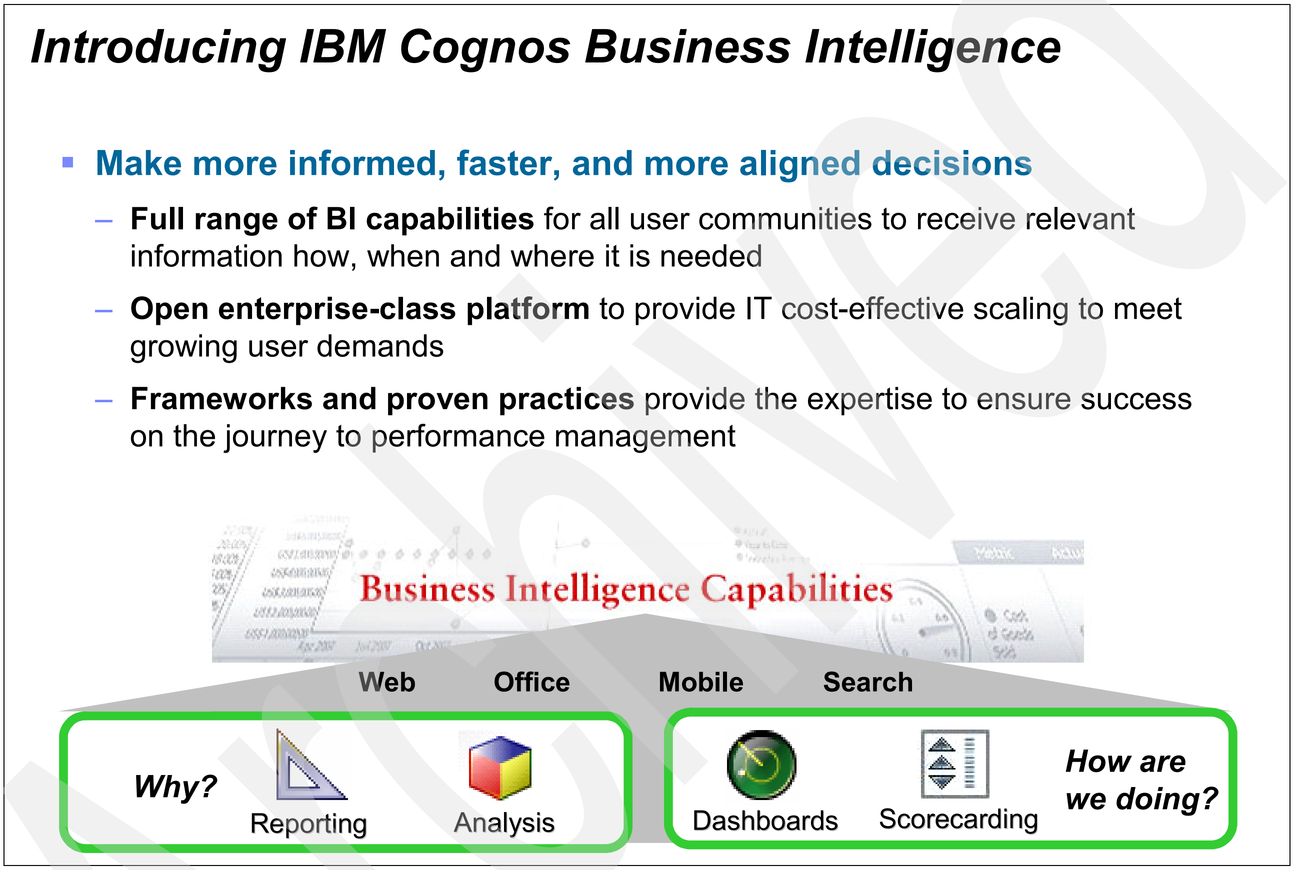


Figure 1-2 IBM Cognos BI

Decision makers throughout the organization need the following information capabilities so that they can find answers to basic questions that drive performance:

- They need to be able to measure and monitor their business and share the current status. This allows them to find answers to the question: How are we doing?
- They need to be able to dig down and determine why the situation is what it is.
- They need the capabilities of reporting and analyzing to make sense of what has happened. That gives decision makers the ability to look at historic data and understand trends, anomalies, and understand why it is that way.

Planning is the common denominator. Planning helps you understand what is going on and sets a forward-looking view of the business, which helps measure and monitor against actual performance. Planning answers the question: What should we be doing?

These integrated capabilities allow you to find answers to the fundamental questions that drive performance. These are the questions that enable you to respond to changes happening in the business and helps make decisions that effectively drive an organization's performance.

1.2.2 IBM Cognos BI features overview

IBM Cognos Business Intelligence delivers a system for performance management that provides the information and the analytic environment where decisions can be made more efficiently and more effectively. IBM Cognos 8 Business Intelligence is a complete set of BI capabilities to help organizations address the vital questions about their performance (Figure 1-3 on page 8):

- Reporting
  - Provides a full breadth of report types
  - Delivers consistent information across all types of report output
  - Can be personalized and targeted
  - Enables collaboration across users and communities and with IT
  - Provides access via email, portal, MS-Office, search, and mobile devices
- Analysis
  - Provides exploration across multiple dimensions of information
  - Performs complex analysis and scenario modeling easily
  - Gets to the *why* behind trends to reveal symptoms and causes
  - Moves from the summary level to detail of information effortlessly
- Dashboards
  - Provides at-a-glance, high-impact views of complex information
  - Helps quickly focus on issues that need attention and action
  - Are highly visual and intuitive
  - Combines information across disparate sources
- Scorecards
  - Provides instant measurement relative to targets and benchmarks
  - Aligns decisions and tactics with strategic initiatives
  - Supports scorecarding methodologies
  - Ensures ownership and accountability

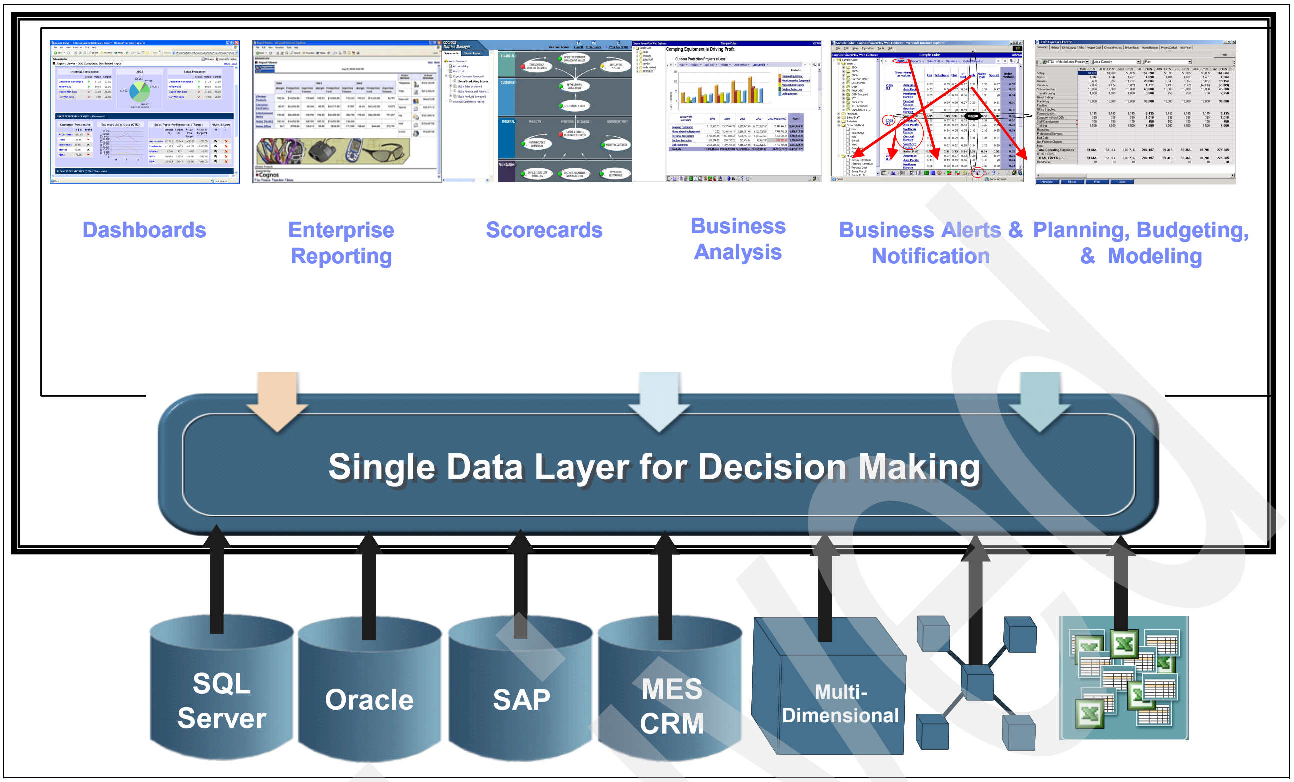


Figure 1-3   All BI capabilities - one source of the truth

From the IT perspective, IBM Cognos 8 Business Intelligence provides the following key features (Figure 1-4 on page 9):

- All capabilities on a single, web services-based SOA.
- It leverages existing infrastructure so that it is not necessary to add duplication of functions, or duplicate security or proprietary application servers. It is an open modern standard that fits clearly into the direction in which the IT architecture is growing, by focusing on business intelligence.
- Predictably performs in complex environments. If BI systems are going to grow in scope across the organization in complexity, amount of information, and number of users, IBM Cognos 8 Business Intelligence architecture can do the heavy lifting. It is able to scale up and out to respond to needs as you add more hardware resources and also provides redundancy so it is reliable, deployable, and easy to get up and running to get information out to users.
- BI capabilities are provided in a zero footprint browser. This means that there is nothing to install or maintain on any person's desktop, such as viewing, query, analysis, and even authoring dashboards. There are no applets to download, and no plug-ins to install or maintain. The value for IT in this approach is that there is no delivering of software to the desktops, which means no installation, no compatibility concerns, no costly upgrade, and the freedom to support a user community not tied to specific PCs. This zero footprint enables broad cost-effective rollouts to hundreds of thousands of users, and does not require a lot of IT resources.
- IBM Cognos 8 Business Intelligence is designed to fit into any web environment. Web servers such as IBM WebSphere, Microsoft® IIS, and Apache are supported. The web infrastructure in place does not need to be changed (for example, firewalls) to ensure security or to provide load-balancing routers that spread requests across server farms. IBM Cognos 8 Business Intelligence fits into this environment with lightweight web

gateways that simply forward incoming requests from the web server through to the application tier.

Figure 1-4 Business intelligence features all in one product

IBM Cognos 8 Business Intelligence architecture is focused on providing open access to data, regardless of where the data resides. IBM Cognos 8 Business Intelligence provides access though a single metadata model. The metadata model provides a consistent business view to the user and also provides a highly productive metadata environment. This environment is built to support the segmentation of models to separate developers and then the ability to bring them back together to create the enterprise model.

IBM Cognos 8 BI is designed to support best practices of building multi-tier models from physical to business. This is critical for enabling IT to model the complex physical environment, and yet translate that into a business model that makes sense to the user and hides the complexity.

To keep track of all the content published to the user as a view, as well as information about reports, metrics, user preferences, and so on, there is a content and metric store where these business views are stored (such as report and analysis definitions, report output, content folders, server and user configuration information). This content store can be implemented with your standard relational database system (RDBMS) including Oracle, DB2®, MSFT SQL Server, and Sybase (Figure 1-5).

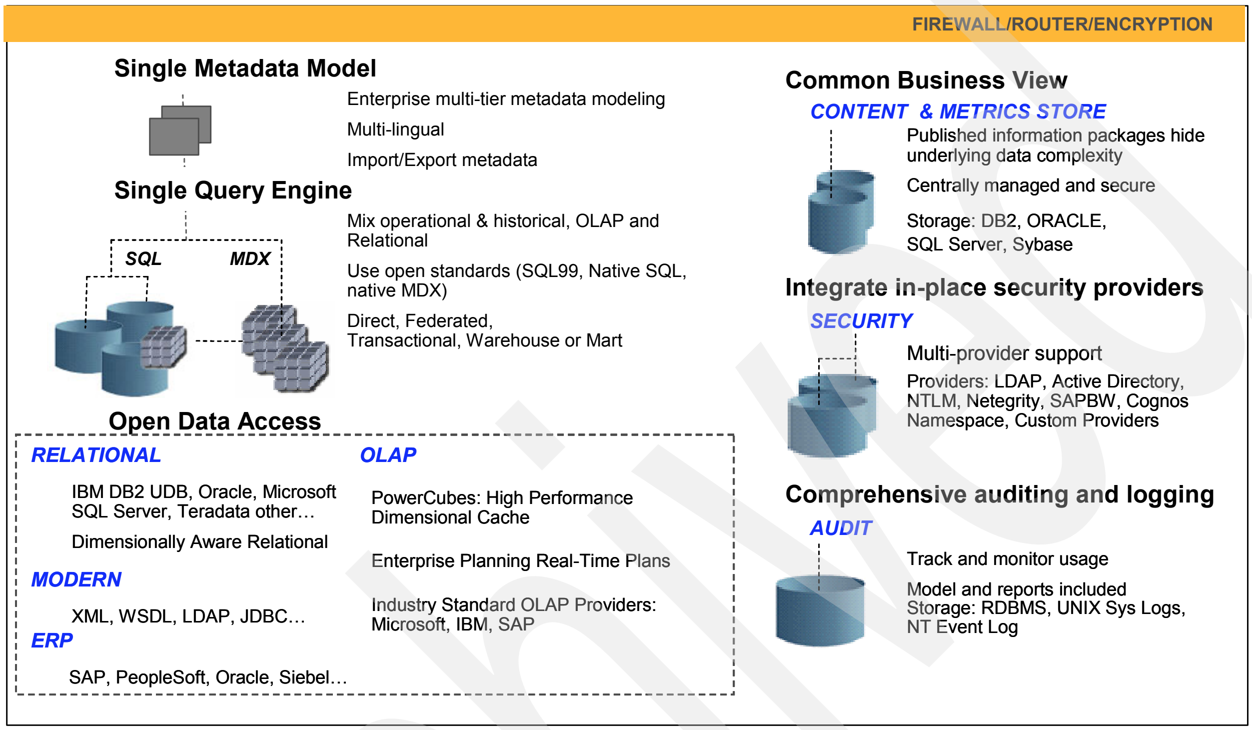


Figure 1-5 Software stack integration

This chapter provided an introduction to the IBM Power Systems virtualization features and a brief introduction to the IBM Cognos 8 BI solution. Chapter 2, “IBM Cognos 8 BI considerations” on page 11, describes the options available when integrating the solution with IBM software stack components (such as Tivoli® Directory Server, WebSphere Application Server, and databases).

2

IBM Cognos 8 BI considerations

In this chapter, we provide details about the IBM Cognos solution from a simple default installation to integrating with other existing servers and services. This chapter discusses the following topics:

2.1 IBM Cognos architecture overview

The IBM Cognos 8 architecture features a consistent, zero footprint, web-based user interface for viewing, creating, and administering reports, analyses, scorecards, and events. It has a common dispatcher and supports leading relational databases as well as Online Analytical Processing (OLAP) and dimensionally modeled relational cubes. It ensures dynamic and automatic load balancing at the application level and provides failover recovery for 24x7 operation. The architecture also provides a single point of administration, as well as web-based delegated administration. It also integrates with your current environment.

The IBM Cognos 8 architecture is a service-oriented architecture where resources on network in a SOA environment are available as independent services that can be accessed without previous knowledge of the platform. This means that the IBM Cognos 8 Business Intelligence (BI) can be deployed to any location in your environment regardless of the underlying technology in place.

2.2 The IBM Cognos BI solution

The Cognos BI software package comes with multiple components that are built for distributed application. Even though the entire Cognos package can be installed on a single machine, its components can be distributed and installed on separate machines. The secret is in the three-tier architecture (for more details see 3.1, “Cognos multi-tiered architecture and components” on page 26):

- Presentation/web tier
- Application tier
- Data tier

The same analogy applies to multiple software packages that complement Cognos. All of these can be installed on only one machine or on many:

- IBM WebSphere Application Server
- IBM HTTP server
- IBM DB2
- IBM Tivoli Directory Server/IBM Tivoli Directory Server as LDAP server

These software packages are not required for Cognos to work. However, they help when workload and demand grow for Cognos. Understanding how these software packages work with Cognos and the three-tier architecture are essential for smooth transitions from a simple installation to more dynamic and adaptable on demand configurations.

2.3 Default installation options

A single out-of-the-box installation makes Cognos simple to start with. The required software or tools are included as default options. Figure 2-1 shows a single server Cognos 8 configuration.

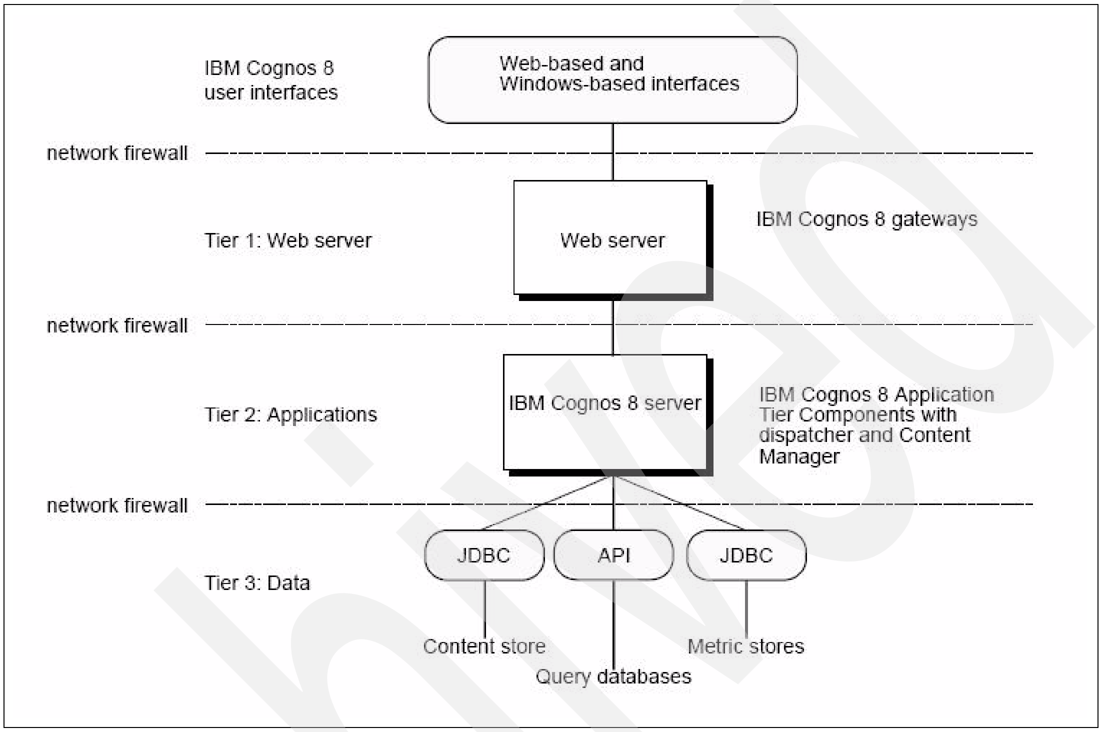


Figure 2-1 Cognos 8 single-server configuration

Below we list the core default components with brief descriptions. Detailed information is provided in the following sections to help the reader understand better how the default components interoperate with other software packages.

- Web server

  Cognos is based on serving information over the web through gateways that are an extension of the web server. Therefore, a web server or HTTP server is mandatory. The gateways are often CGI programs by default. However, the IBM Tivoli Directory Server can be incorporated to provide user directories and access authentication.

- Application server

  To enrich the web user interfaces and data manipulation capabilities, a web application server and the Java™ Virtual Machine or servlet container are also required. Cognos uses Tomcat as an application server by default. A copy of Tomcat is installed with Cognos. On the other hand, Tomcat can be replaced by an external web application server such as the IBM WebSphere Application Server.

- Content database

  If selected at installation, the Apache Derby database can be used as the default content store for Cognos. The content store is required for Cognos metadata. The same database can also be used for the notification database. To overcome Apache Derby's limitations and extend the capability to a full-featured database, the IBM DB2 server can be used as an external database management tool for Cognos.

2.4 Infrastructure considerations

Installing all applications and all components onto one computer is great for a quickstart to evaluate the Cognos solution. Cognos can be tried out or implemented for proof of concept. However, considering all the functionalities and workloads that these applications have to handle, a single server might be quickly overloaded. When the CPU power is maxed out and memory is depleted, upgrading the number of CPUs and the amount of memory on this server might be inevitable.

Taking one step further, for enterprise-scale reporting, analysis, scorecarding, and event notification, a single server is not a viable solution for many reasons, such as:

- The single server becomes a single point of failure. Therefore, instead of putting all the investment of hardware and software into one server, an equivalent amount of funding can be invested for multiple servers.
- In a typical enterprise environment, certain applications such as DB2, Tivoli Directory Server, and WebSphere Application Server, already exist. Therefore, being able to integrate them with Cognos provides an attractive alternative solution.
- Breaking up the applications or components within an application in effect divides the workload into separate servers. If the application is built to be truly modular, each individual component uses less memory footprint as a result.

In distributed computing, increasing the number of servers to handle the same or more workload is called scaling out, as opposed to scaling up a single server. The IBM Power Systems PowerVM capability takes the ideas of distributed computing to a higher level of flexibility and efficiency.

The first step in distributing the applications and the components in Cognos is to understand how they are divided in the first place. Starting from Cognos, its components are based on a multitiered architecture, as described in the following sections.

2.4.1 User authentication using Tivoli Directory Server

By default, Cognos 8 allows anonymous access to the web portal through its default namespace. There is no login prompt when the web browser enters the Cognos home URL. To provide user authentication, install Tivoli Directory Server (TDS) on a system that has a database server. TDS does not have to be on the same system as Cognos 8. Use the Cognos configuration tool to override the access default with an optional LDAP or SAP namespace (Figure 2-2).

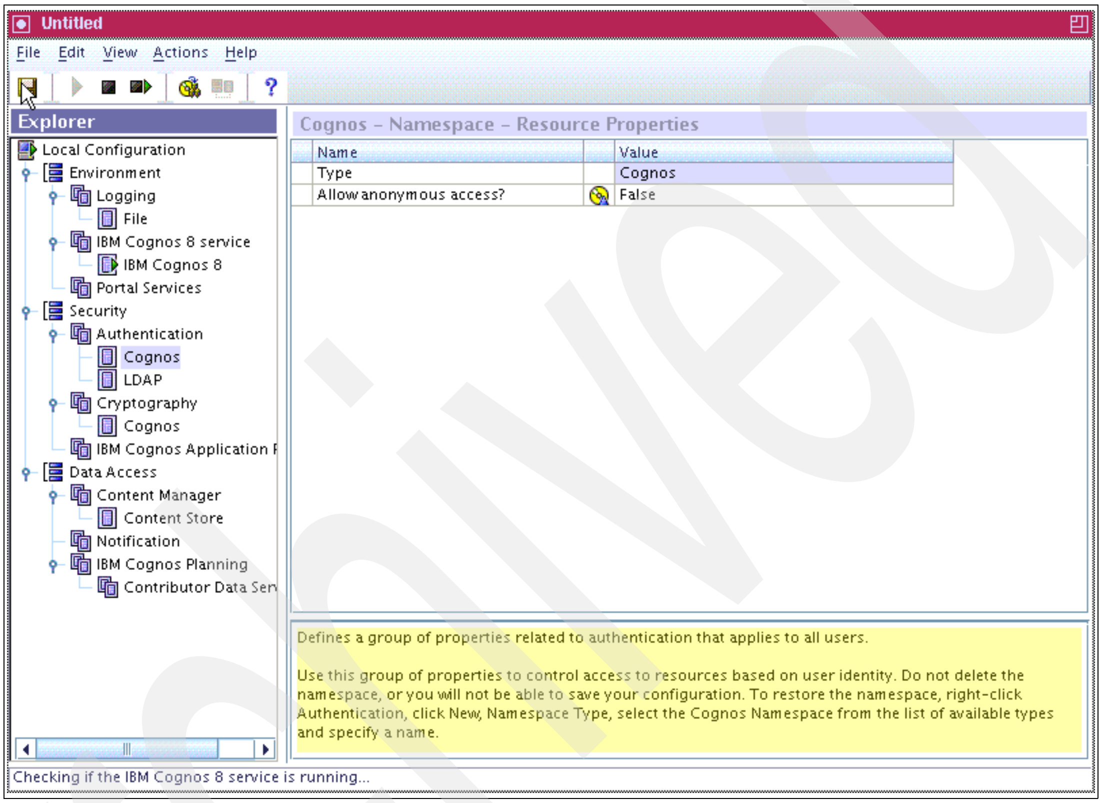


Figure 2-2 Setting anonymous access to false

Before you define the LDAP, choose the namespace from a list of options by right-clicking the **Authentication** menu in the Explorer panel on the left-hand side. In the window shown in Figure 2-3, choose **LDAP** and enter `LDAP` in the Name field. The LDAP menu option displays under the Authentication menu in the Explorer panel.

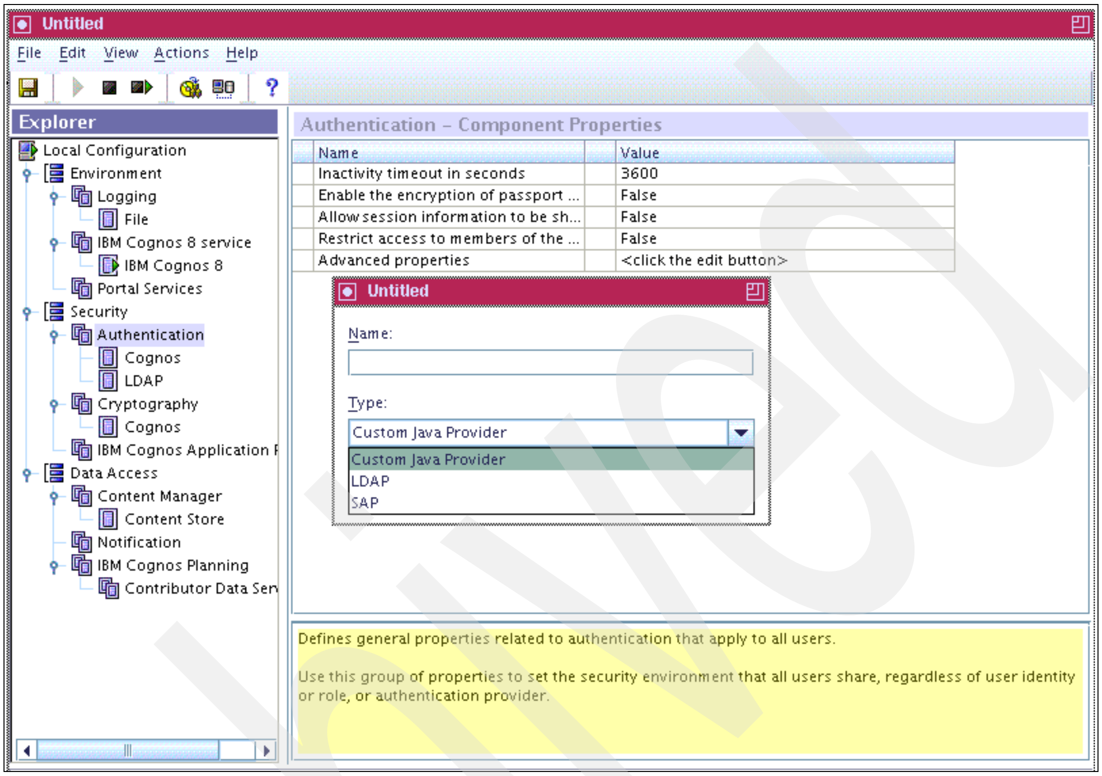


Figure 2-3 Choosing an authentication namespace

Clicking **LDAP** displays the LDAP Namespace Resource Properties on the right-hand side of the panel (Figure 2-4). Enter your own specific host, port, and distinguished name in the three required fields. Our example values are:

- Namespace ID: win32
- Host and port: 192.168.100.198:888
- Base Distinguished Name: ou=Ottawa, o=cognos.com

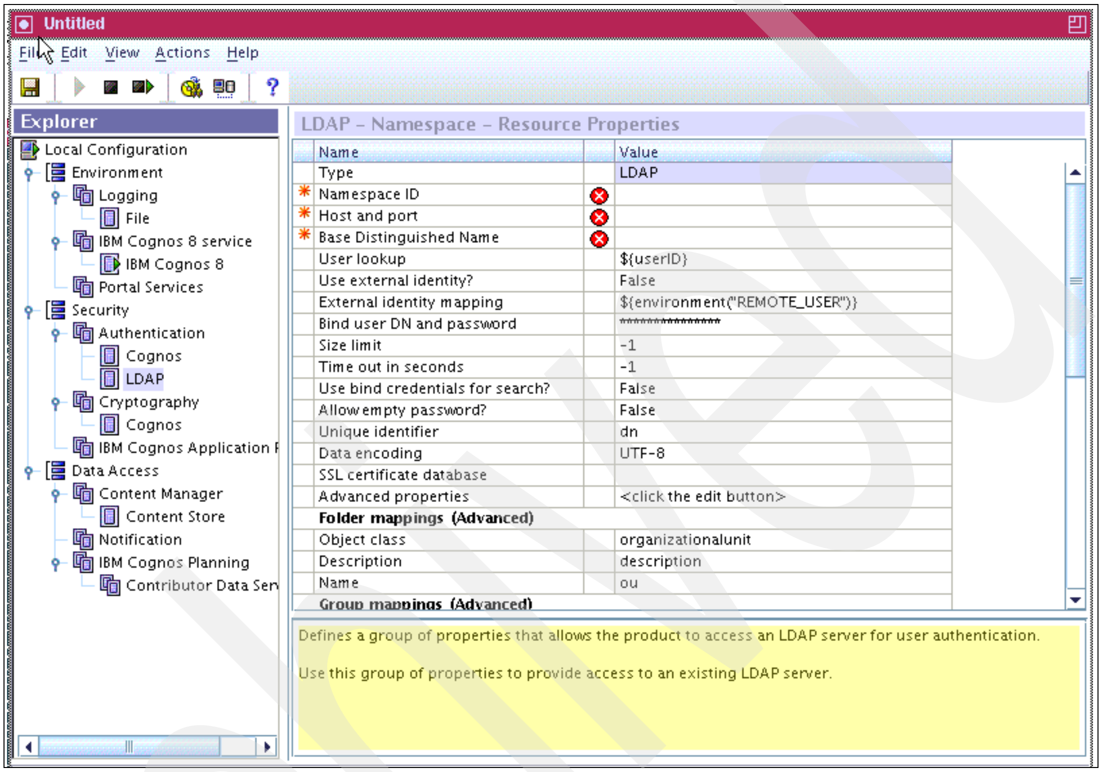


Figure 2-4 LDAP namespace and resource properties

2.4.2 Using WebSphere Application Server

Cognos applications are built based on the Application Archive File formats called Enterprise Archive File (EAR) or Web Archive File (WAR). Using the Cognos Application Wizard GUI, the user can package the applications within Cognos into the archive file, which then can be imported or deployed in other application servers, such as WebSphere Application Server.

Deploying Cognos application consideration

Consider that:

- For customers who already have an application server running, they can deploy the Cognos applications on their application server without additional resource requirements. Tomcat can be disabled and ignored if another application server is used.
- For customers who currently do not have an application server, using Tomcat as the default by Cognos is a good start. However, an external application server provides more options for hosting Cognos applications.

Which Cognos applications to deploy

In the Cognos server distributed configurations, separate components can be installed and operated on separate computers (such as machines, LPARs, or WPARs). Once the distributed components are verified to be functioning, the applications can be packaged and deployed.

> **Note:** The procedures of packaging and deploying the applications include backing up the working *copies* of the applications.

Content Manager (CM) stores and retrieves information from the content store. Multiple CMs can be installed and configured, although only one CM is active at a time.

Application tier components can be installed and configured on the same or separate computers (such as machines, LPARs, or WPARs) with the CM. This option is necessary when there is a high volume of requests or loads.

The Cognos distributed model matches with the ability of creating multiple server instances in the WebSphere Application Server. Each application is packaged into a separate EAR file and deployed separately on the WebSphere Application Server.

Deploying application archive onto WebSphere Application Server

WebSphere Application Server (WAS) creates the environment where Enterprise Applications or Java EE components run. Each application server instance is implemented as a Java Virtual Machine (JVM), in which applications can be deployed. The server instance is configured via a profile. With multiple profiles, the WAS application servers can be set up a stand-alone or distributed. There are many options for creating and managing profiles to meet application or server scaling requirements. Figure 2-5 shows multiple WASs in a server or in multiple servers.

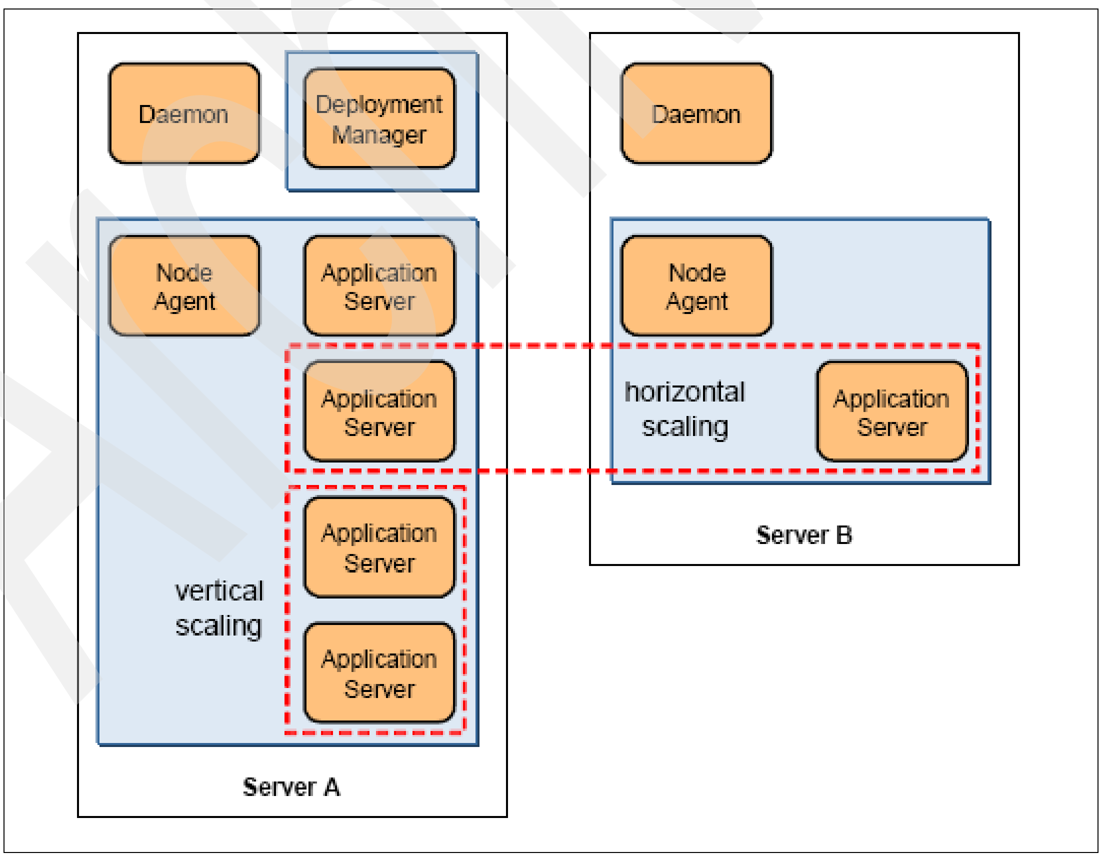


Figure 2-5 Vertical and horizontal scaling with WAS

With PowerVM, the server that runs the WAS does not have to be a physical system. WAS runs on LPAR or WPAR and provides two types of virtual hosts:

- default_host: on ports 80, 9443, and 9080
- admin_host: on ports 9060 and 9043

WAS also works well with the IBM HTTP server, which can be installed and operated on the same server or a remote server (Figure 2-6).

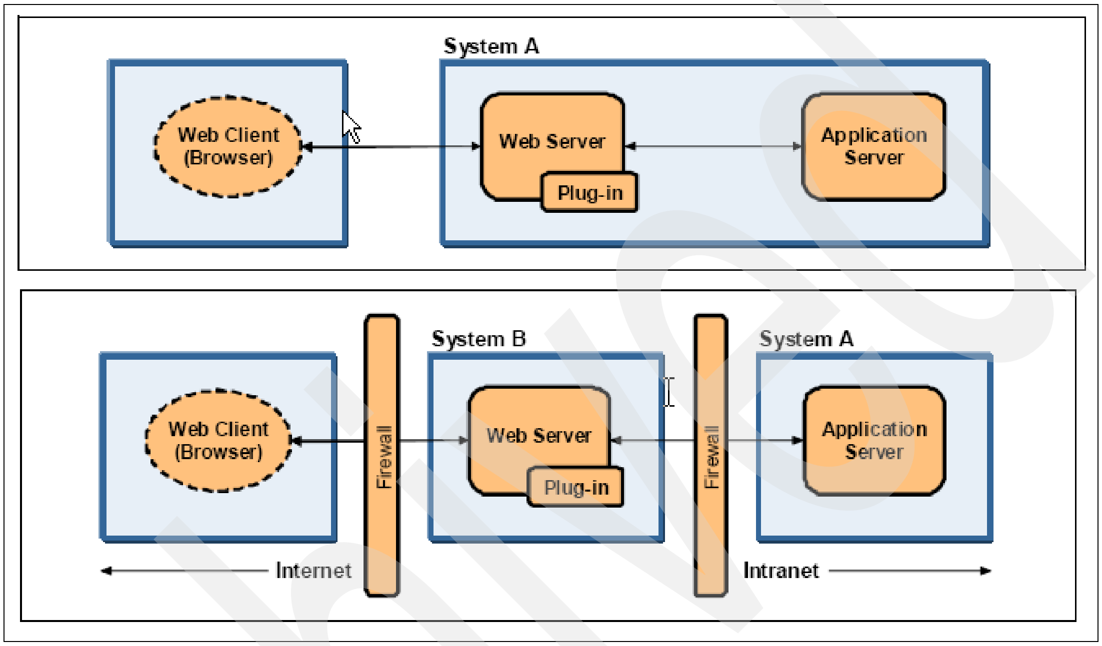


Figure 2-6 Local versus remote web server setup with WAS

The extendability is endless considering WAS federated servers or clustering servers with high availability. For more information see the WebSphere software website:

http://www.ibm.com/software/websphere/

2.4.3 Integrating with external database server

The Cognos 8 data tier contains:

- The content store
- The data sources
- The metric store

A database management server is required as data sources for querying databases from Cognos. The Cognos content store and metric store databases might reside on the same or on separate database servers. As discussed in “DB2 server installation” on page 52, DB2 is used for all database work in this publication.

Database client and configurations

Cognos works with the DB2 server running on a remote system. A DB2 client is required on the Cognos Content Manager system for connecting to the databases. Figure 2-7 shows the DB2 client connecting to the COGNOS or COGCM content store on server 192.168.100.202 and the GOTEST test database on server 192.168.100.197.

Figure 2-7 Cognos database connections

The database server and port number are configured in the Cognos Content Manager configuration. The Cognos configuration graphical interface shows the Content Manager information. To see it, run the command in Example 2-1.

Example 2-1 Command to show the Content Manager information

```
#/usr/cognos/c8_64/bin64/cogconfig.sh
```

When in the Cognos Configuration Explorer, select the content store under Content Manager. Database - Resource Properties displays on the right (Figure 2-8). There can only be one database server connected to the active Content Manager at a time. The access port number is defined on the DB2 server.

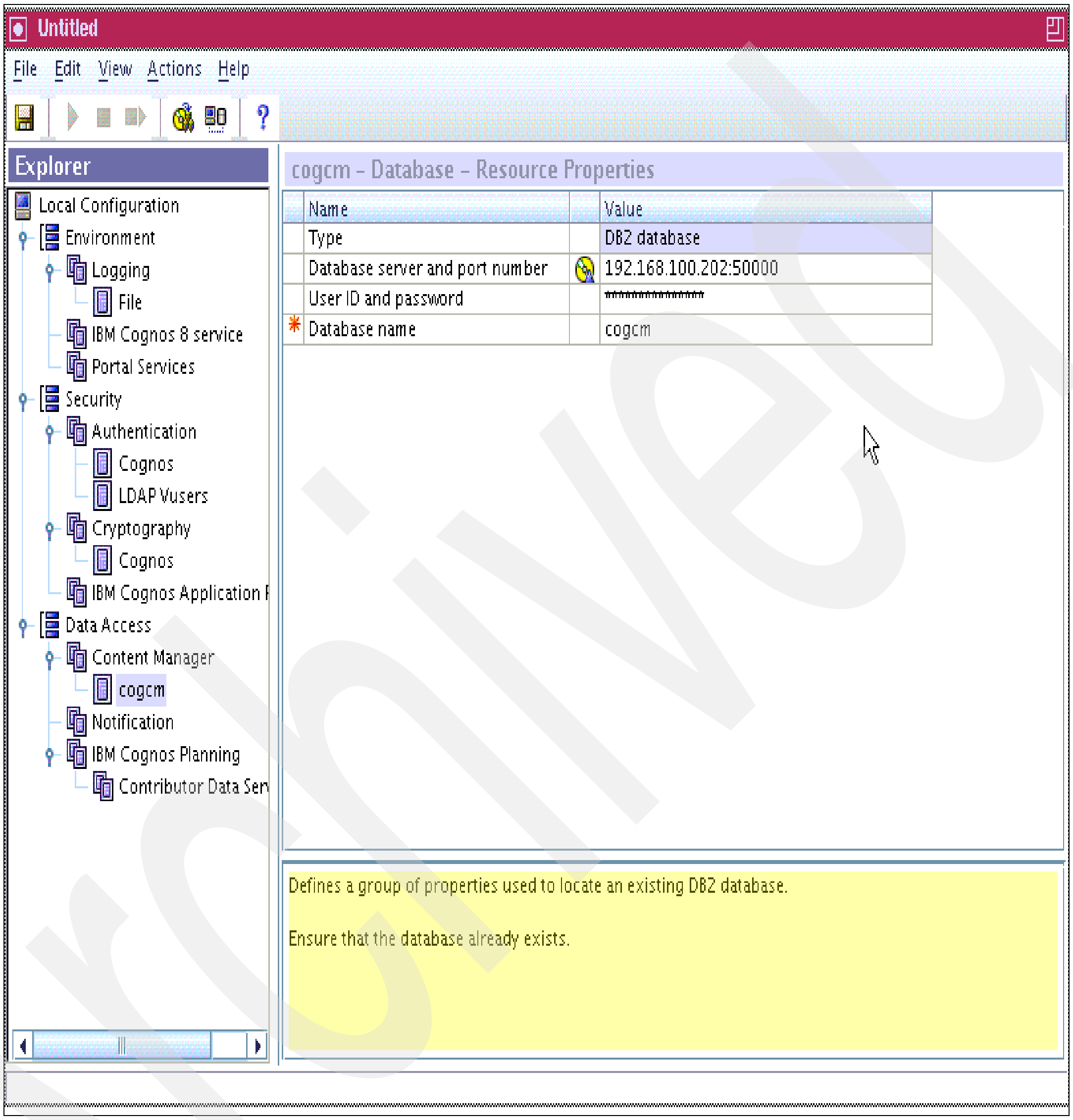


Figure 2-8 Content store configuration

In a distributed environment, the standby Content Managers also specify the same content store database. However, they only take over the Content Manager services when the primary Content Manager is down. Figure 2-9 depicts the connections to the databases in a distributed configuration.

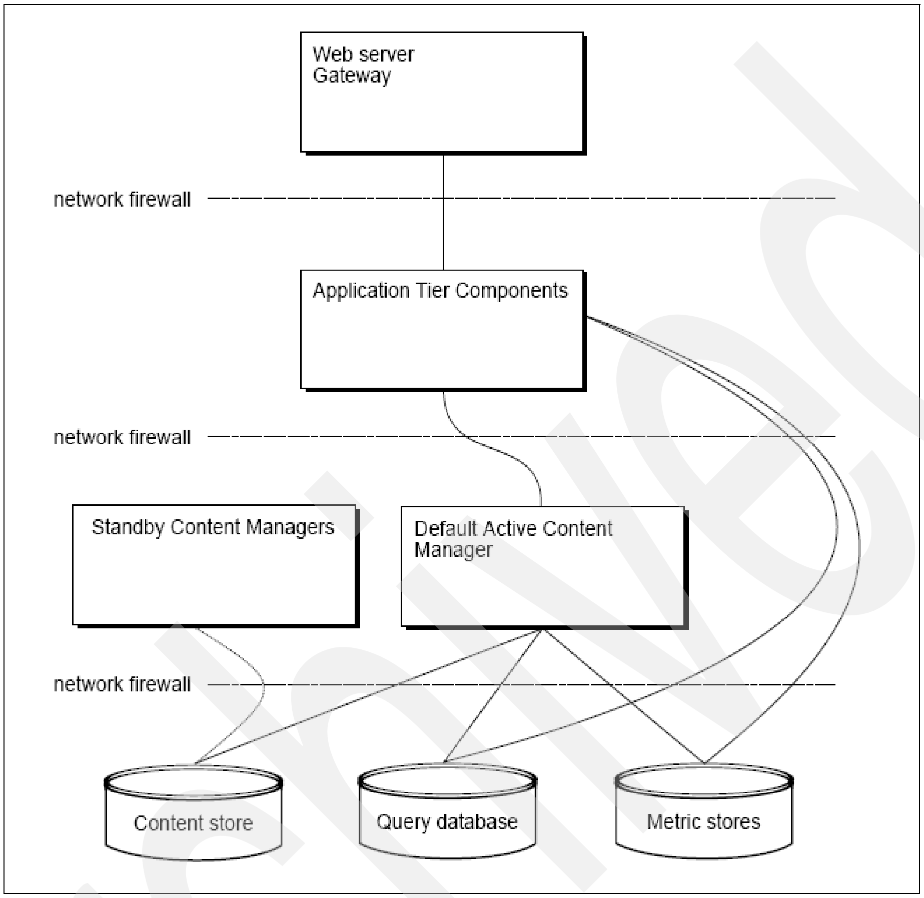


Figure 2-9 Database connection in a distributed configuration

Communication and database connections

This section describes the communication and database connections in a Cognos environment.

Cognos BIBus processes

An important element of communication between Cognos components is the BIBus (BiBusTKServerMain). The BIBus processes are spawned from Java, but run native (C++) code. Hence, there is always a shell associated, which might require environment variables. This behavior might change in other application servers. The Cognos application services especially use the BIBus processes to exchange requests and data with the data layer. Users never interact with the BIBus process directly. Similarly traffic on the highways delivering goods and keeping production flowing, BIBus processes play similar roles in the performance of Cognos applications. Depending on the BIBus processes are:

- Messaging and dispatching
- Log message processing
- Database connection management
- Microsoft .NET Framework interactions
- Port usage

- Request flow processing
- Portal pages

All the processes are for the need to access the data. To carry data, a bus needs memory. Each BIBus process is a multi-threaded process spawned by the Cognos dispatcher. Figure 2-10 shows that a BIBus process contains one high-affinity thread and four low-affinity threads.

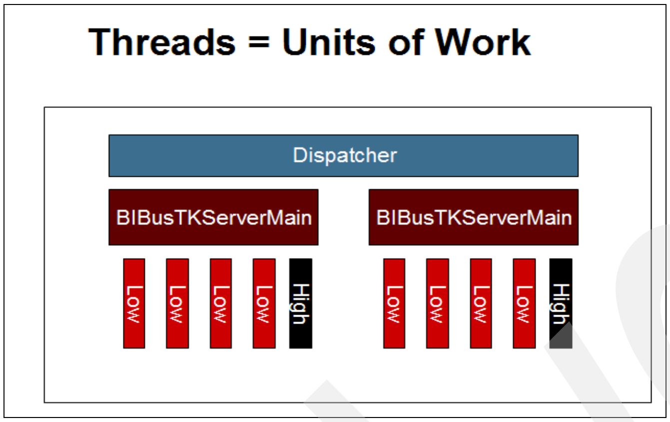


Figure 2-10 Multi-threaded BI bus

Although the Cognos administrator can define the number of BIBus processes using Cognos configuration interfaces, the amount of physical memory decides the number of buses to be operating at the same time. Data moves faster with more buses. However, when the buses crowd up the memory space, congestion can occur and slow down the processes. Remember not to configure more buses than the real memory available.

> **Note:** Cognos applications run in the JVM, which also requires a considerable amount of memory. A JVM usually takes approximately 2 GB to run.

With multiple distributed Cognos servers, the number of BIBuses are set to the same value on each server. The total number of BIBus processes is increased by the number of servers available.

Database connections

Every request to the CM for accessing the content store uses a database connection. CM creates database connections as needed. Connections are put in a pool for purposes of reuse. The Cognos architecture document cites that “The theoretical maximum number of concurrent Content Manager requests equals the number of requests accepted by the Java application server or Tomcat.”

Accessing the query databases also requires database connections. The number of maximum database connections and the duration for which connections are retained are configurable. The Cognos application server has a cleanup thread that examines the connections every minute. A connection is removed after staying inactive for the duration longer than the time-out value.

This chapter provided several installation options for integrating with IBM software stack components. Chapter 3, “IBM Cognos 8 BI features and components” on page 25, describes in more depth the features and components of the IBM Cognos 8 BI solution.

3

IBM Cognos 8 BI features and components

This chapter describes the Cognos architecture, components, topologies, and benefits.

The IBM Cognos 8 architecture was designed for scalability, availability, and openness. It uses platform-independent, industry-proven technology such as Extensible Markup Language (XML), Simple Object Access Protocol (SOAP), and Web Services Definition Language (WSDL). For this reason, IBM Cognos 8 can be integrated with existing technology infrastructures on multiple platforms and supports leading relational databases as well as Online Analytical Processing (OLAP) and dimensionally modeled relational cubes.

This chapter discusses the following topics:

- "Cognos multi-tiered architecture and components" on page 26
- "Request flow" on page 34
- "Affinity connection requests" on page 41
- "Balance requests among dispatchers" on page 43
- "Topologies and benefits" on page 44

3.1 Cognos multi-tiered architecture and components

The IBM Cognos 8 Business Intelligence (BI) is a multi-tiered architecture that can be separated into the following tiers (Figure 3-1 on page 27):

- Tier 1: web server: IBM Cognos 8 gateways

  Web communication in IBM Cognos 8 is typically through gateways that reside on one or more web servers. IBM Cognos 8 supports several types of web gateways including CGI, ISAPI, apache_mod, and servlets, if you are using an application server.

- Tier 2: application server: IBM Cognos 8 servers

  The second tier of the IBM Cognos 8 BI is where the majority of the processing is performed and hosts the IBM Cognos 8 server and its associated services. The components of the application tier are responsible for receiving user requests from the web server and treats them appropriately. It is also the application tier that often has to query the data tier in the form of SQL requests to obtain data.

  – Application tier components: includes the dispatcher and the services managed by the dispatcher

  – Content Manager: stores application data including report specifications, report outputs, security, and configuration

- Tier 3: data content

  The third tier of IBM Cognos 8 BI hosts the content store (application data), query data sources (such as relational data sources), dimensional cubes, files, and other Cognos data sources (such as a metric store).

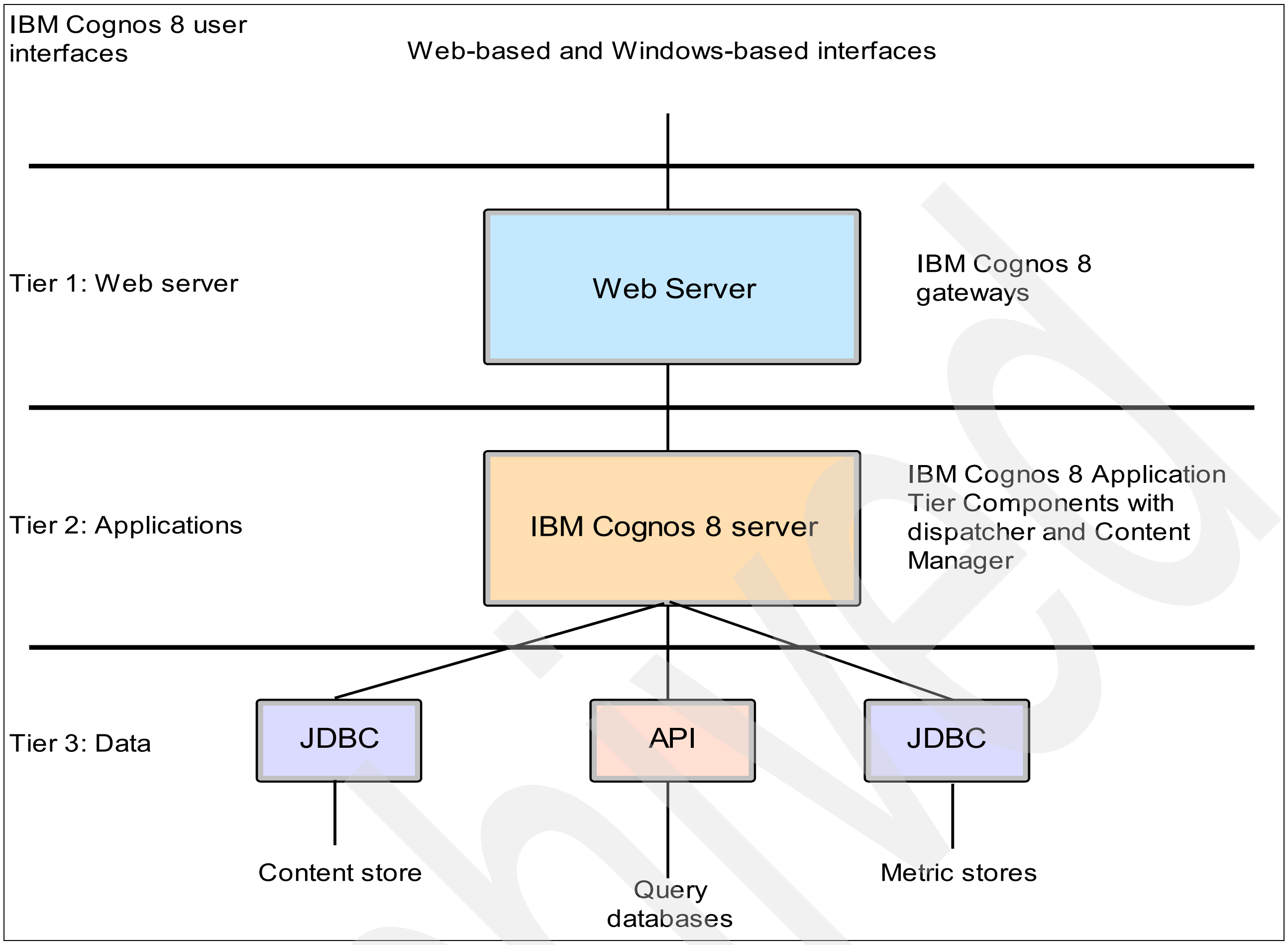


Figure 3-1 IBM Cognos multi-tier architecture overview

3.1.1 IBM Cognos gateway

Web communication in IBM Cognos 8 is typically through gateways that can be installed on one or more web servers. IBM Cognos gateway supports several types of web gateways, including CGI, ISAPI, apache_mod, and servlet, if the web server tier employs an application server.

When an IBM Cognos 8 gateway receives a request, it performs a series of security procedures, including encrypting passwords, extracting information needed to submit the request, attaching environment variables for the web server, adding a default namespace to the request to ensure that the server authenticates the user in the correct namespace, and passing the requests to an IBM Cognos 8 dispatcher for processing.

Installing multiple gateways on the environment provides a better level of service, but does not provide load balancing. Load balancing options are discussed later in this chapter.

3.1.2 Cognos 8 Business Intelligence servlets

The Content Manager servlet and the dispatcher servlet run with the Java servlet container. Both are multithreaded and the threads are isolated from each other. By default, IBM Cognos 8 Business Intelligence uses Apache Tomcat as the default servlet container. The Apache Tomcat already comes with IBM Cognos 8 Business Intelligence. The dispatcher servlet also launches child processes as part of the implementation of Content Manager services (for example, the report service and batch report service shown in Figure 3-2).

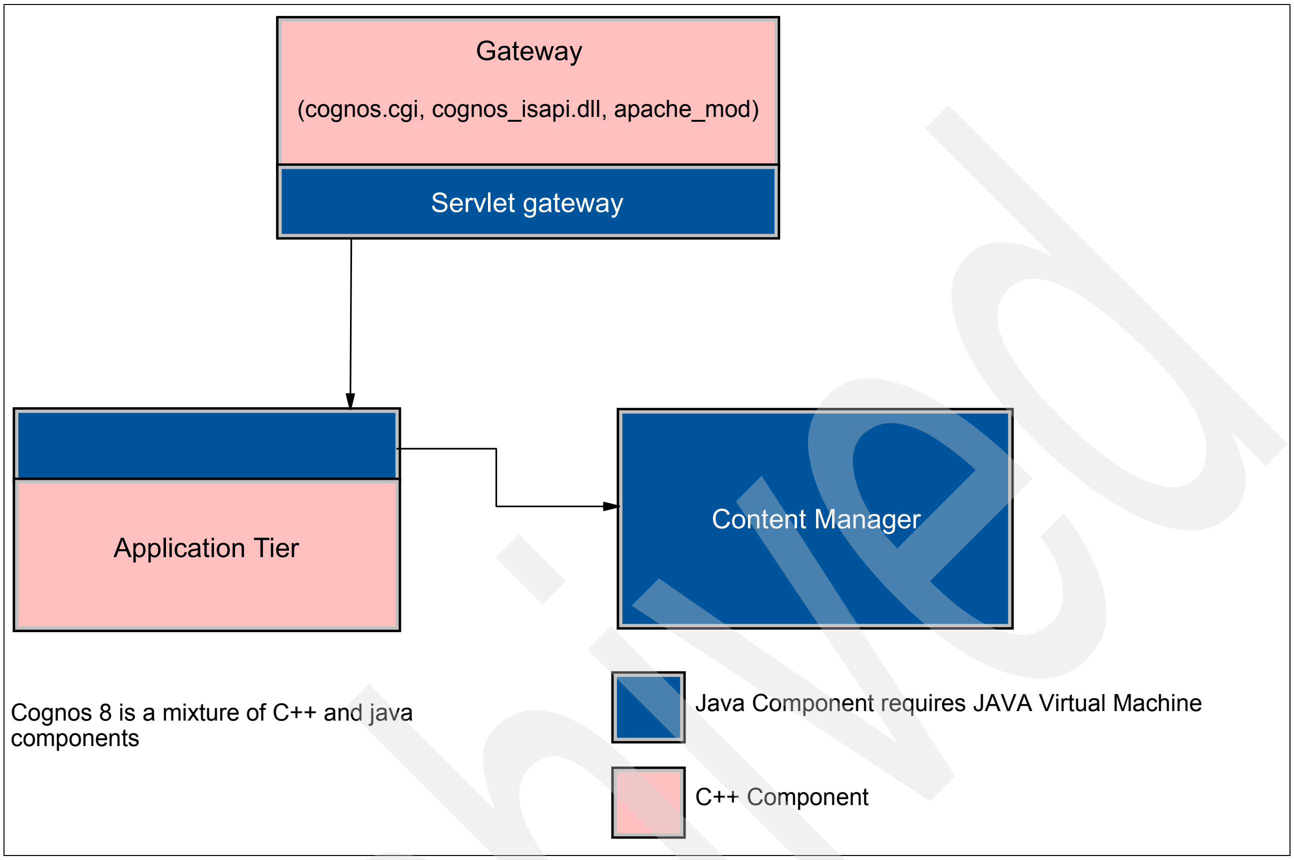


Figure 3-2 Cognos servers

3.1.3 The dispatcher

The dispatcher is a multithreaded Java servlet application that starts all IBM Cognos 8 BI services configured and enabled on a computer, and routes requests from the gateway to local services or a sibling dispatcher servlet. Each dispatcher has a set of associated services, which Table 3-1 describes.

Table 3-1 Dispatcher associated services

Service	Purpose
Agent service	Runs agents. If the conditions for an agent are met when the agent runs, the agent service asks the monitor service to run the tasks.
Batch report service	Manages background requests to run reports and provides output on behalf of the monitor service.
Content Manager service	► Performs object manipulation functions in the content store, such as add, query, update, delete, move, and copy. ► Performs content store management functions, such as import and export.

Service	Purpose
Data movement service	Manages the execution of data movement tasks in IBM Cognos 8. Data movement tasks, such as builds and JobStreams, are created in Data Manager Designer and published to IBM Cognos 8.
Delivery service	Sends emails to an external SMTP server on behalf of other services such as the report service, job service, agent service, or data integration service.
Event management service	Creates, schedules, and manages event objects that represent reports, jobs, agents, content store maintenance, deployment imports and exports, and metrics.
Job service	Runs jobs by signaling the monitor service to run job steps in the background. Steps include reports, other jobs, import, exports, and so on.
Log service	Records log messages generated by the dispatcher and other services. The log service can be configured to record log information in a file, a database, a remote log server, Windows® Event Viewer, or a UNIX system log. The log information can then be analyzed by customers or by Cognos software services, including: ► Security event ► System and application error information ► Selected diagnostic information
Metric Studio service	Provides the metric studio user interface for monitoring and entering performance information.
Migration service	Manages the migration from IBM Cognos Series 7 to IBM Cognos 8.
Monitor service	► Manages the monitoring and execution of tasks that are scheduled, submitted for execution at a later time, or run as a background task. ► Assigns a target service to handle a scheduled task. For example, the monitor service might ask the batch report service to run a report, the job service to run a job, or the agent service to run an agent. ► Creates history objects within the Content Manager and manages failover and recovery for executing entries.
Planning job service	Manages communications with the planning job server subsystem.
Planning web service	Manages communications with Contributor Web and Contributor Add-in for Excel users.
Planning administration console service	Manages communication with the Contributor Administration Console.

Service	Purpose
Planning data service	Manages communications for real-time reporting from Contributor plan data in IBM Cognos 8.
Presentation service	► Transforms generic XML responses from another service into output format, such as HTML or PDF. ► Provides display, navigation, and administration capabilities in IBM Cognos Connection.
Report data service	Manages the transfer of report data between IBM Cognos 8 and applications that consume the data, such as IBM Cognos 8 Go! Office and IBM Cognos 8 Go! Mobile.
Report service	Manages interactive requests to run reports and provides output for a user in IBM Cognos Connection or a studio.
System service	Defines the Business Intelligence Bus API-compliant service used to obtain application-wide IBM Cognos 8 configuration parameters. It also provides methods that normalize and validate locale strings and map locale strings to locales supported by your application.

Note: Services running on each machine might vary depending on the configuration of the environment. Services such as planning job service, planning web server, planning administration console service, and planning data service are used to manage tasks associated with IBM Cognos 8 planning in your environment.

The dispatcher registers with the Content Manager when it starts the first time. Since they have a single point of registration, the dispatcher is aware of all other dispatchers registered and running. The dispatchers available in the environment can be accessed through **Cognos portal** ∅ **Administration** ∅ **Configuration** (Figure 3-3).

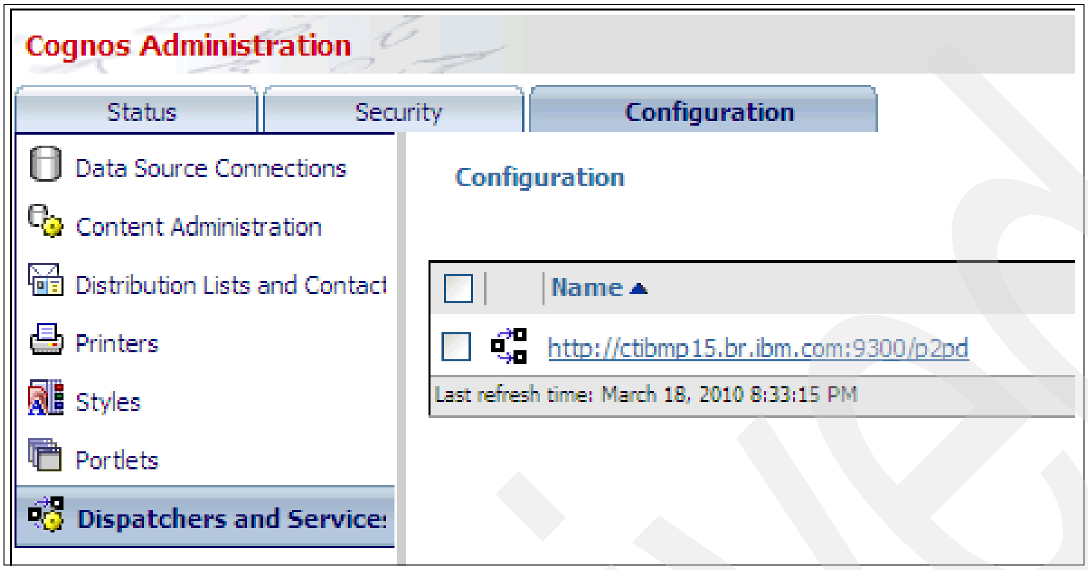


Figure 3-3 Cognos portal administration

The services running on a dispatcher can be located by clicking the selected dispatcher (Figure 3-4).

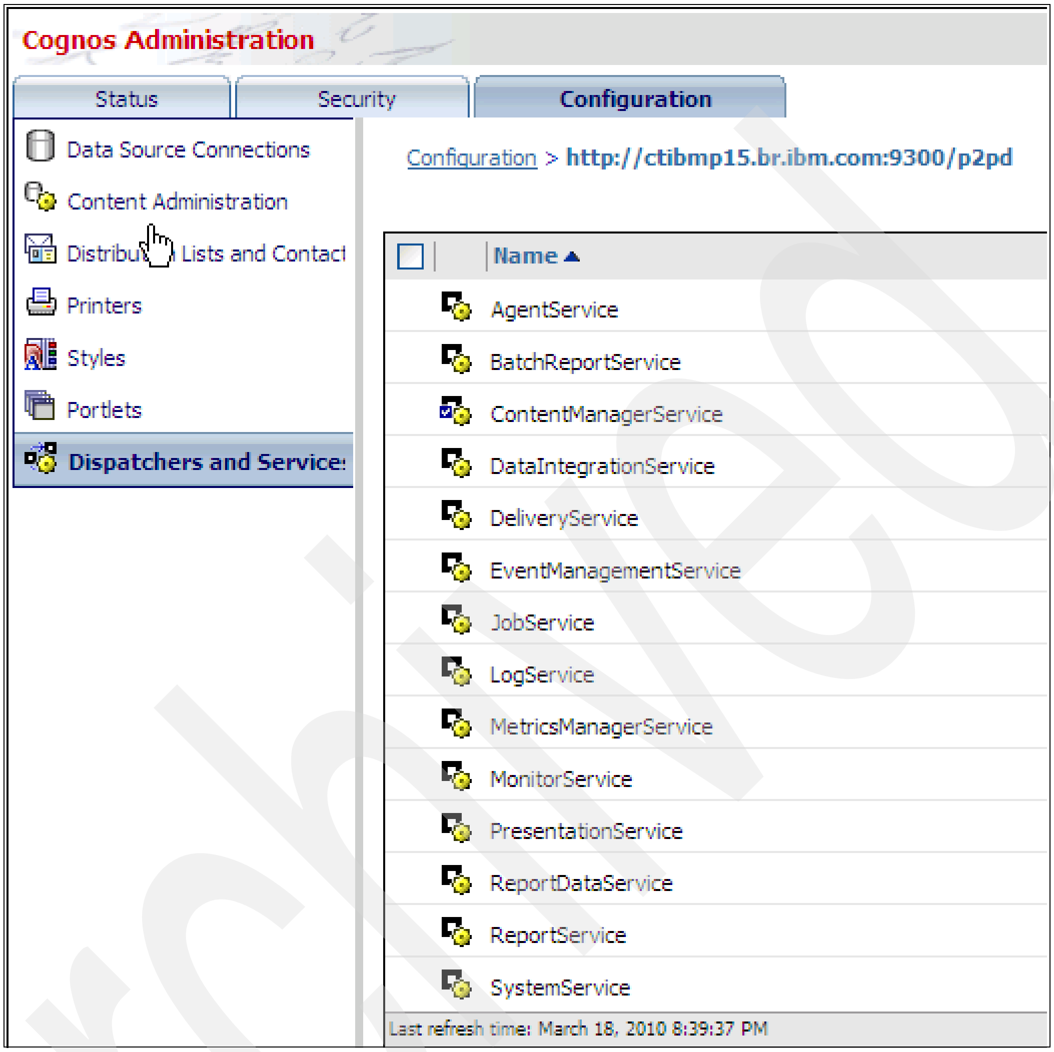


Figure 3-4 Services running with the dispatcher

Also, we can open the Cognos configuration for UNIX (**c8location ∅ bin ∅ cogconfig.sh**) or for Windows (**c8location ∅ bin ∅ cogconfig.bat**) and check what services are enabled on the server (Figure 3-5).

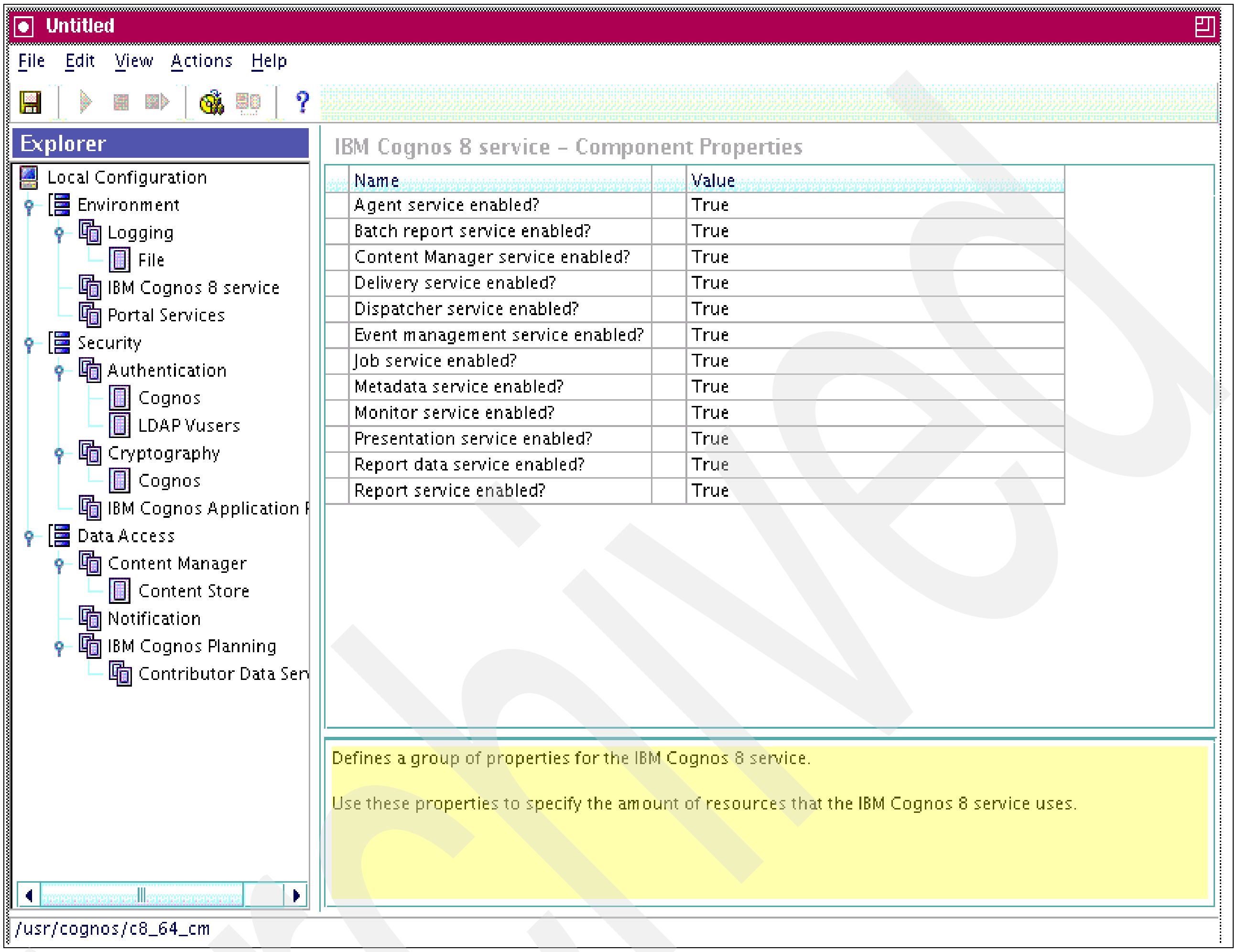


Figure 3-5 Checking the services enabled

3.1.4 Content Manager

The Content Manager service is a Java servlet that manages the storage of application data, including security, configuration data, models, metric packages, agents, personal user information, language information, report specifications, and report output. It is needed to publish packages, retrieve or store report specifications, and manage the Cognos namespace. The tables in the content store are created when the application is started for the first time.

The Content Manager also contains the Cognos Access Manager (CAM) (Figure 3-6), which is the security component used to implement authentication, authorization, and encryption.

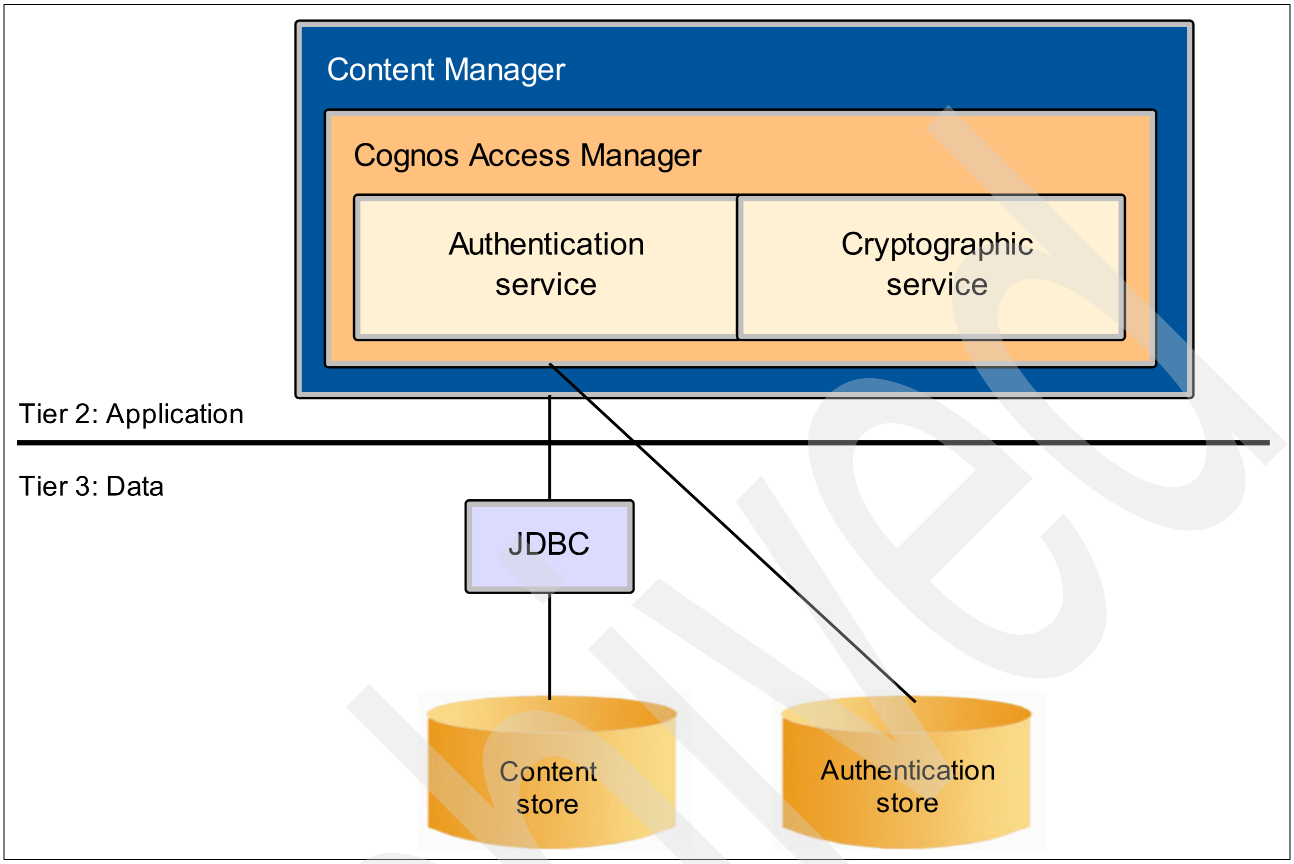


Figure 3-6 Cognos access manager

We might include more than one Content Manager, each on a separate server or on a separate instance of JVM. Just one Content Manager service per computer is active, and one or more Content Manager servers are in standby mode. The standby Content Manager servers are for failover protection only and not for load balance.

3.1.5 Content store

The content store is the relational database that contains the data that IBM Cognos 8 BI needs to operate, such as report specifications, published models, packages, information to other data sources, and security.

3.2 Request flow

For each user request, a separate IBM Cognos 8 service can be used or a separate flow request can be processed. There are six type of requests:

- Access IBM Cognos 8.
- View a report or analysis.
- Run a scheduled task.
- Open the IBM Cognos Connection folder.
- Run an agent.

It is important to understand the request flow processing so that you can identify which services are being used to provide the user response. For example, if most of the requests for reports in the environment are for report views, it is expected that you will have high CPU and memory usage at the machine running the Content Manager service.

For the purpose of this book, we cover the request processing that does the majority of processing of IBM Cognos 8 Business Intelligence.

3.2.1 User authentication

The user authentication process is:

1. The user attempts to access the IBM Cognos Connection Welcome page from a web browser, sending a request to the IBM Cognos gateway.
2. The IBM Cognos gateway accepts the request and sends it to a dispatcher.
3. The dispatcher notes that there is no passport attached to the request and sends the request to the Content Manager.
4. The Content Manager sends the request to the Cognos access manager.
5. Anonymous access is disabled in this IBM Cognos system, so the access manager sends the request back to the Content Manager with a fault attached. The fault contains information about what is needed to log on. For example, if there are multiple namespaces, the user might be required to select a namespace. If there is only one namespace, the user might be required to provide a user ID and password.
6. The Content Manager returns the request with the attached fault to the dispatcher.
7. The dispatcher sends the request to the presentation service.
8. The presentation service creates the appropriate logon page for the user and returns the page through the dispatcher and the gateway to the user.
9. The user enters the required information, such as a user ID and password. The information is attached to the original request and sent through the gateway to the dispatcher.
10. The dispatcher sends the request to the Content Manager.
11. The Content Manager sends the request to the Cognos access manager.
12. If all the required information is correct, the access manager issues a passport, attaches it to the original request, and sends the request back to the Content Manager.
13. The Content Manager sends the request to a dispatcher.

14. The dispatcher processes the request and sends it to the presentation service (Figure 3-7).

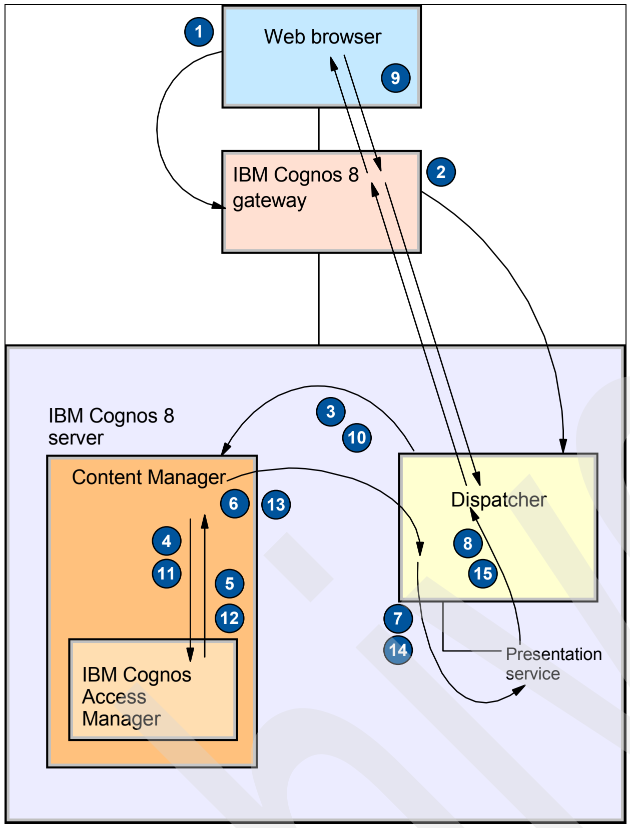


Figure 3-7 User authentication request flow

3.2.2 Portal navigation

The portal navigation process is:

1. The user clicks a folder to open it, and the request goes through the gateway and the dispatcher to the presentation service.
2. The presentation service sends the request to the Content Manager.
3. The Content Manager checks with the access manager to determine whether access is granted.
4. The Content Manager sends the folder contents or an error message to the presentation service.

5. The presentation service formats the appropriate HTML page and sends it through the dispatcher and the gateway to the user (Figure 3-8).

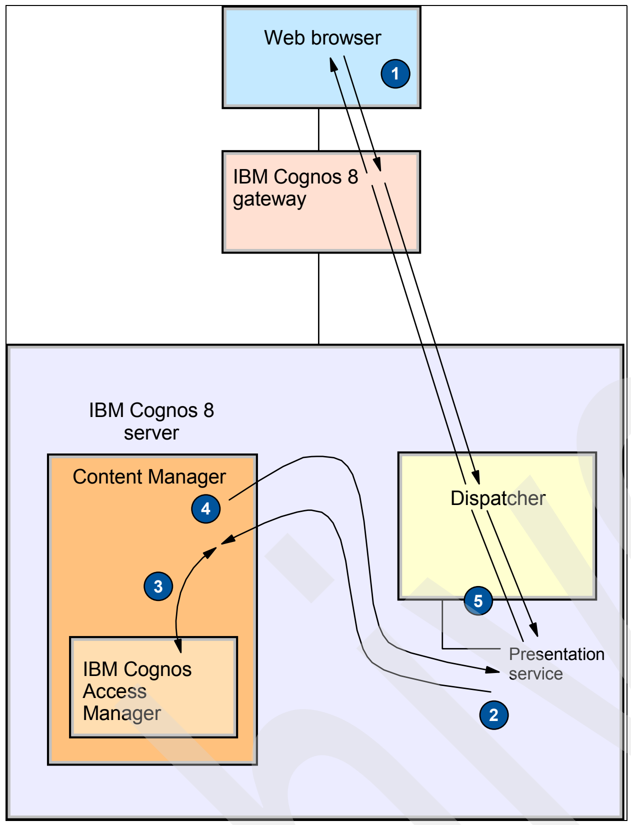


Figure 3-8 Portal navigation flow

3.2.3 Report viewing

The report viewing process is:

1. The user clicks a report to view it, and the request goes through the gateway and the dispatcher to the presentation service.
2. The presentation service returns the Cognos report viewer through the dispatcher and gateway to the browser. The report viewer has an HTML frame that shows the report page. The data view of the page contains a URL to a Content Manager object to be displayed.
3. The browser sends the URL through the gateway and the dispatcher to the content manager.
4. The Content Manager checks with the access manager to see whether the user has view privileges for the report. To do this, the Content Manager sends the access manager the access control lists for the report, the attempted action (read), and the user information.

5. The Cognos access manager determines that the user can perform the action, and the Content Manager sends the report in the form of an HTML page or PDF document through the dispatcher and gateway to the user. When serving a PDF report, byte serving is used (Figure 3-9).

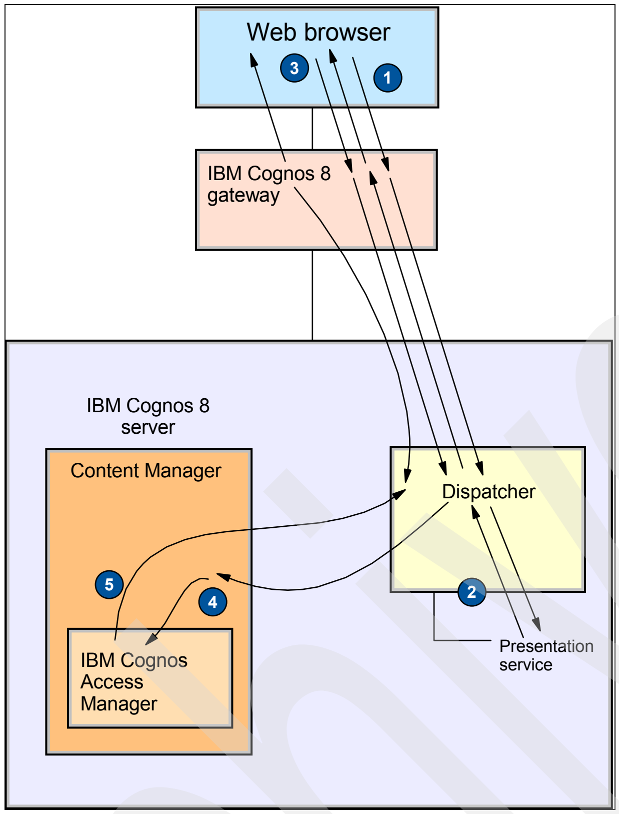


Figure 3-9 Report view flow

3.2.4 Request flow for report execution (low affinity request)

When an user runs an HTML report through the Cognos connection, the following occurs:

1. The user clicks a report to run it, and the request goes through the gateway and the dispatcher to the presentation service.
2. The presentation service sends the request to the report service.
3. The report service requests the report and metadata from the Content Manager.
4. The Content Manager sends the report XML specifications and metadata to the report service. The Content Manager refetches metadata only when IBM Cognos is stopped and restarted or the model is updated and republished.
5. The report service returns one of these results to the presentation service:
   - An error page
   - A not ready page
   - A page of an HTML report
6. The presentation service sends one of these results through the dispatcher and gateway to the browser:
   - An error page
   - A wait or cancel page
   - A page of a completed HTML report in the report viewer interface

When a user runs a PDF report through the Cognos connection, the following occurs:

1. The user clicks a report to run it, and the request goes through the gateway and the dispatcher to the presentation service.
2. The presentation service sends the request to the report service.
3. The report service requests the report XML specification and metadata from the Content Manager.
4. The Content Manager sends the report XML specification and metadata to the report service.
5. The Content Manager refetches metadata only when IBM Cognos is stopped and restarted or the model is updated and republished.
6. The report service returns one of these results to the presentation service:
   - An error page
   - A not ready page
   - A PDF report
7. The presentation service sends the PDF result to the Content Manager for storage in the user's session temporary area. Byte serving is used.

   The wait/cancel page polls every few seconds to see whether the report is complete. When it is complete, the report viewer HTML frame content is replaced with the PDF. A request is sent through the gateway and the dispatcher to the Content Manager and back with the PDF (Figure 3-10).

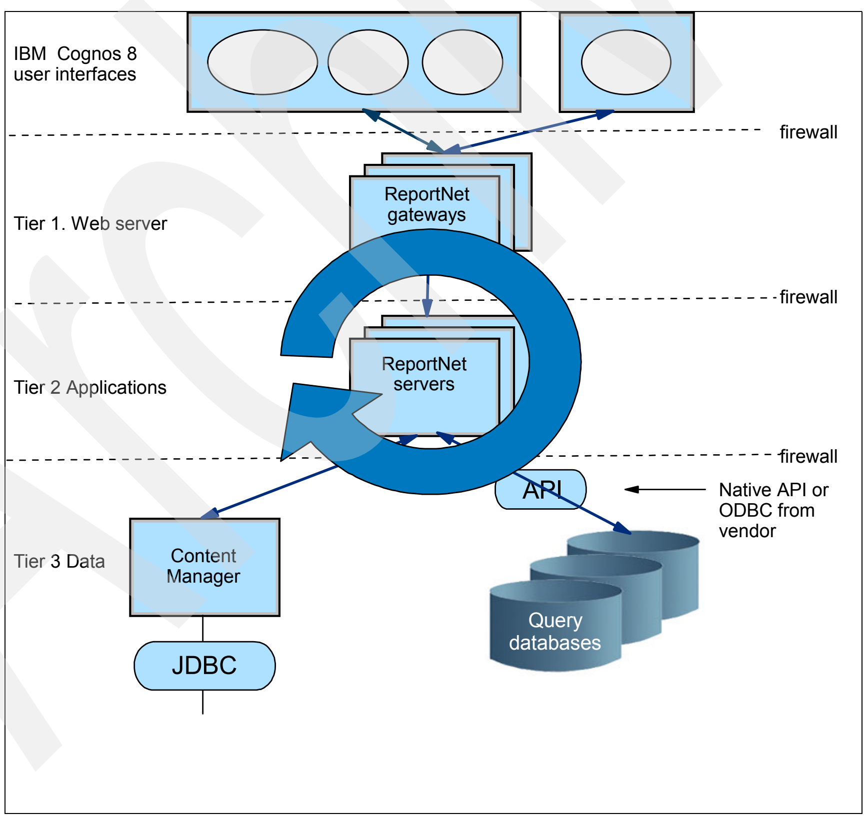


Figure 3-10 Request flow for report execution (low-affinity request)

3.2.5 Request flow for report execution (high-affinity request)

The request flow for report execution (high-affinity request) is:

1. An user submits an HTML navigation request (next page) through his browser.
2. An IBM Cognos gateway receives the request and routes it to a dispatcher.
3. The dispatcher attempts to assign the request to the IBM Cognos report server (BIBus) that processed the open report request using a high-affinity connection.
4. If available, the dispatcher assigns the request to a high-affinity connection of the same BIBus. If unavailable, the dispatcher assigns the request to another BIBus and re-executes the initial report request to get the report to the same state as before so that the high-affinity request can be processed.
5. The report server accesses the report specification (metadata) from memory and the previously generated SQL and shifts the cursor to build the next page of the report.
6. The report server retrieves additional data from the query database if applicable.
7. The report server generates the page according to report specification details (layout, conditions, and so on).
8. The report server delivers content to the gateway through the dispatcher.
9. The user receives the next page in his browser (Figure 3-11).

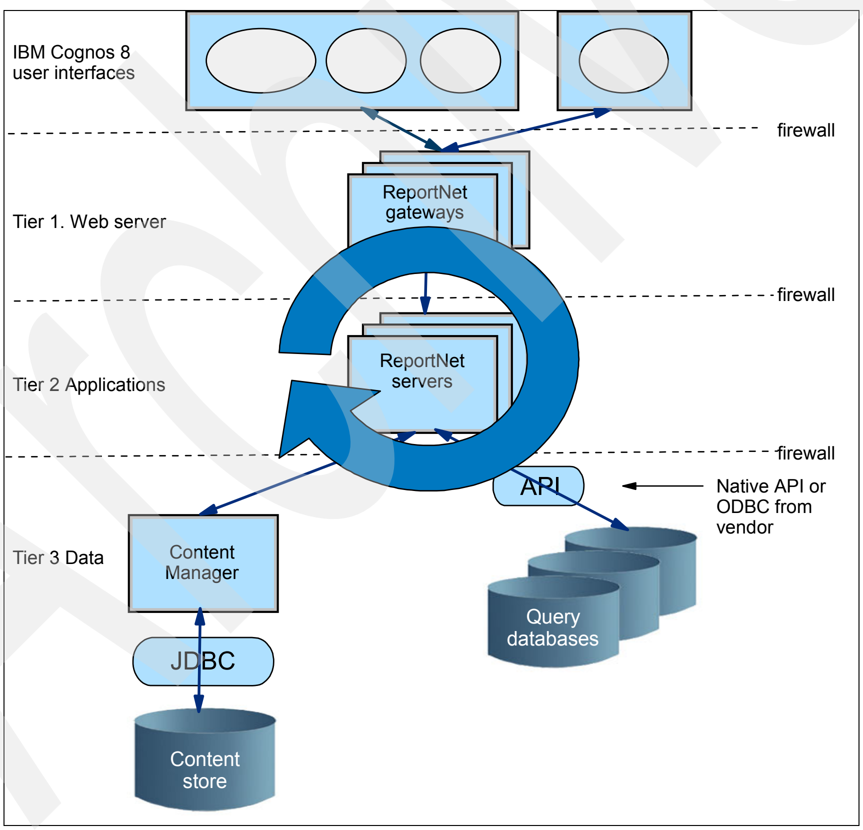


Figure 3-11 Request flow for report execution (high-affinity request)

3.3 Affinity connection requests

This section describes the affinity connection requests.

3.3.1 High-affinity request

High-affinity requests benefit from the work done from previous requests processed. This means that IBM Cognos 8 Business Intelligence routes the new request to the same process and server that was initially invoked (if it is available) regardless of the load balancing used, thus attempting to optimize resource consumption. High-affinity requests apply to the report service only and not to the batch report service. Examples of high-affinity requests are:

- Running the report again
- HTML report navigation
  - Top/bottom
  - Next page
  - Page down
  - Page up
- Report delivery
  - Email
  - Save
  - Save as
  - Print

For example, when a page up command is executed while reading a report, the command is executed more efficiently using the report process that displayed the existing page. When the same process is used, the system scrolls the existing database cursor, retrieves the data, and renders the page. The information is cached by the report process and a subsequent high-affinity request avoids overhead.

High-affinity requests can be processed by a low-affinity connection if all high-affinity connections are currently in use. If the specific server is not available or busy, the initial report request is executed on a separate BIBus process (perhaps on a different server) as a low-affinity request to reestablish the state necessary to execute the high-affinity request.

3.3.2 Low-affinity request

A low-affinity request has no affinity for a report process. This means that these requests are forwarded to any report process on any IBM Cognos 8 Business Intelligence servers running in your environment. They do not attempt to use the same process and server that invoked the request. For a low-affinity request, both the report services and batch report services are capable of handling requests. Examples of low-affinity requests are:

- Administrative
  - Test data source connections.
  - Add objects.
  - Refresh.
- Report Authoring
  - Metadata retrieval
  - Query validation
- Running reports (interactive and batch only)
  - Querying
  - Processing
  - Prompting

Each service enabled on the IBM Cognos can be accessed and configured thorough the administration portal (Figure 3-12). We can define the number of high-affinity and low-affinity requests.

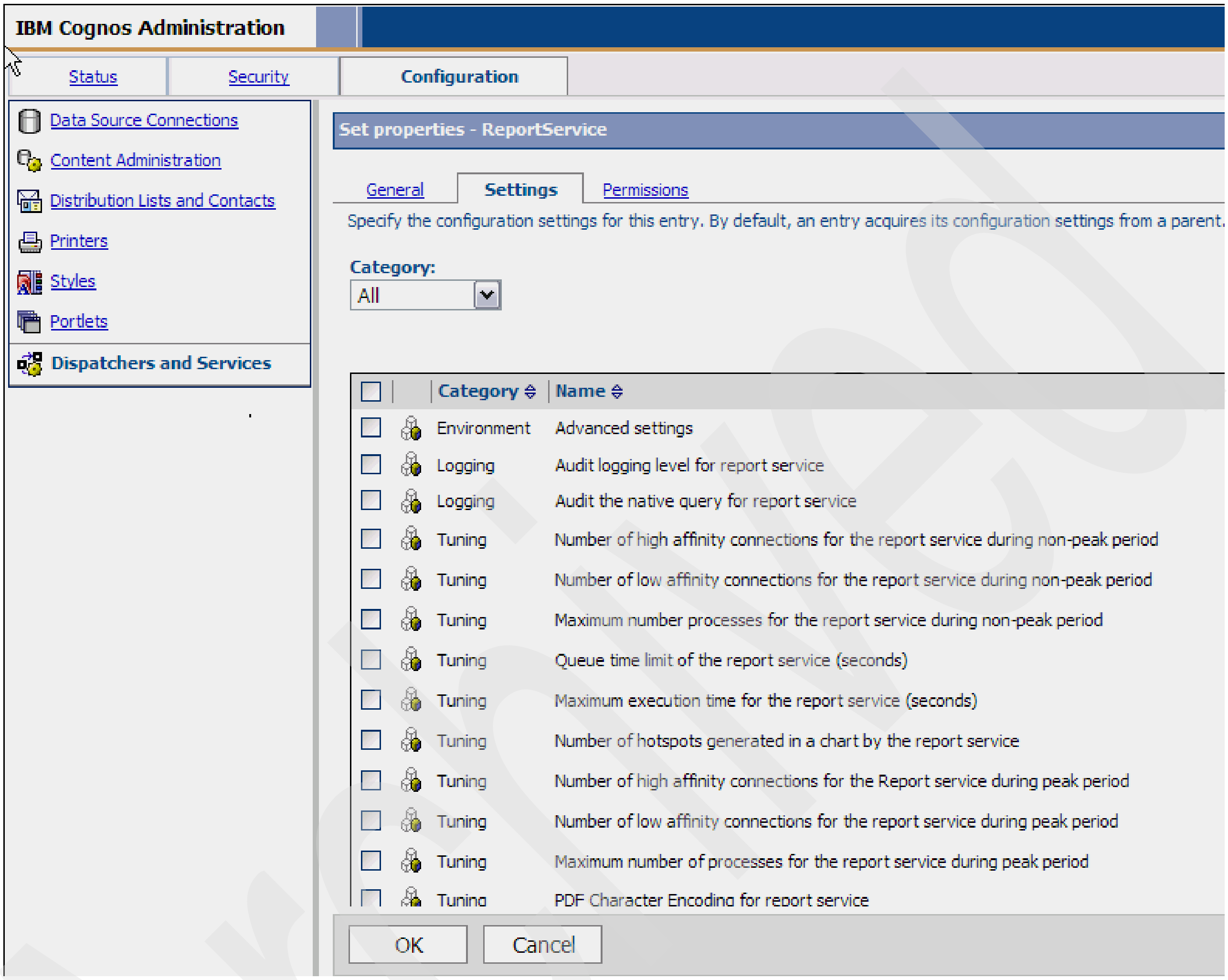


Figure 3-12 Report service details

The number of high-affinity or low-affinity connections available is the product of the values provided for:

- The maximum number of processes setting
- The number of high-affinity or low-affinity request connections

The IBM Cognos 8 BI report process manages the allocation of high-affinity and low-affinity connections, for example, giving the following scenario:

- Two CPUs allocated to report service
- Two report services per CPU equals four report services
- Two low-affinity connections configured per report service equals eight low-affinity connections (Table 3-2)

Table 3-2 Low-affinity and high-affinity connections per report process

Report processes	Low affinity connection	High affinity connection	Total low affinity connection	Total high affinity connection
2	4	1	8	2

If we have four HTML report requests executed at the same time, we have two report services spawned and four low-affinity connections, consuming a total of eight connections available.

3.4 Balance requests among dispatchers

In a distributed environment, we can specify the proportion of requests that each dispatcher is able to process by changing their processing capacity.

If all servers in your environment are equal in processing capacity (number of CPUs, memory, clock rates), set the default processing capacity to 1 and the load balancing mode to weighted round robin (Figure 3-13). With these settings, the requests are distributed equally to all servers.

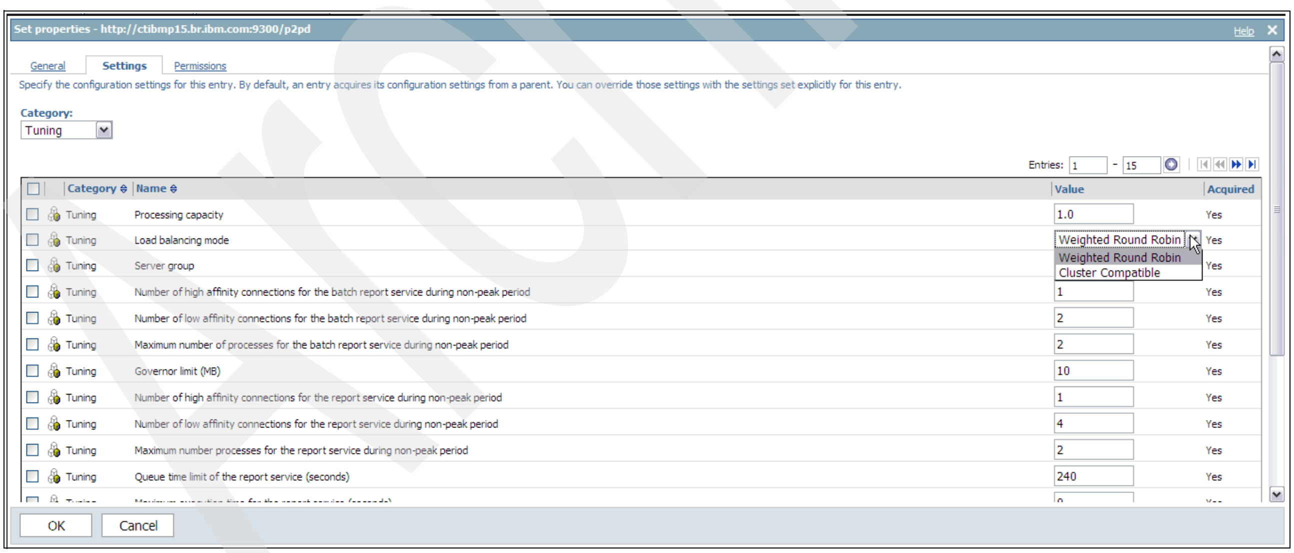


Figure 3-13 Setting the load balancing mode

The load balancing mode can be set to:

- Weighted round robin

  The dispatcher distributes the requests in a weighted round fashion according to the processing capacity.

- Cluster compatible

  The cluster compatible directs the dispatcher to execute requests locally if the service is available (on the same server as the dispatcher) and the request has not been directed to a specific server (high-affinity and absolute-affinity requests). Only use it where the load balancing has been performed already, for example, by the IBM WebSphere web server.

3.5 Topologies and benefits

This section describes topologies and benefits of IBM Cognos 8 BI.

3.5.1 IBM Cognos 8 BI on single computer

IBM Cognos 8 Business Intelligence components can be installed in a single environment. However, most of the environments have implemented application servers, web servers, and databases. Thus, we show this approach of installation only as a reference model for the most simplistic installation and configuration perspective. However, installing all components in a single environment is a good idea just for test or proof-of-concept purposes.

By installing all IBM Cognos 8 Business Intelligence on a single machine (Figure 3-14), a web server installation on the same machine is needed, since the Cognos gateway and web server must be installed on the same machine. If the environment that you are using has an application server instead of the web server, it is possible to move the Cognos 8 Business Intelligence gateway and the dispatcher to the application server. This last approach does not require a web server.

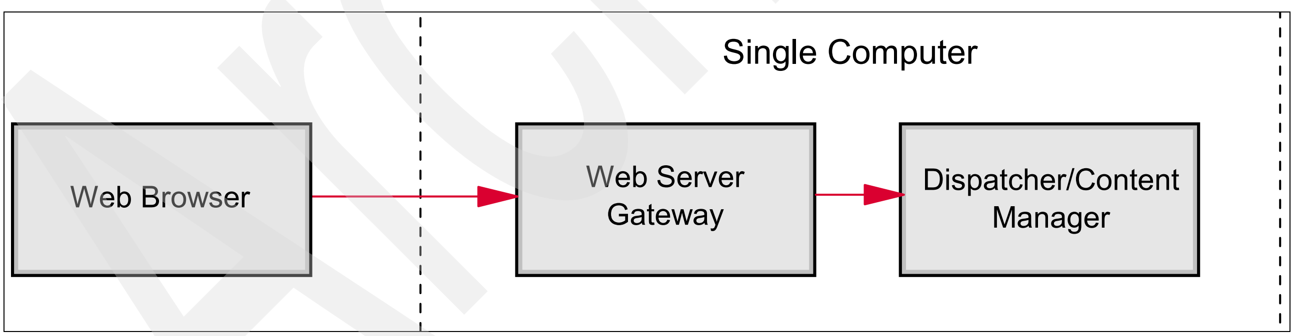


Figure 3-14 Deploying all components on a single machine

> **Note:** Modeling tools such as Framework Manager and Metrics Studio do not run under UNIX environments.

3.5.2 Installing the gateways on separate computers

The first approach to increase performance is to distribute the IBM Cognos 8 Business Intelligence in your environment. The IBM Cognos gateway can be installed on a separate computer from the other components (application tier). If multiple web servers are in place in the environment, it is possible to install the IBM Cognos gateway on each web server. This provides a better level of service since you can manage the requests through each web server. Figure 3-15 provides a simple diagram of this topology.

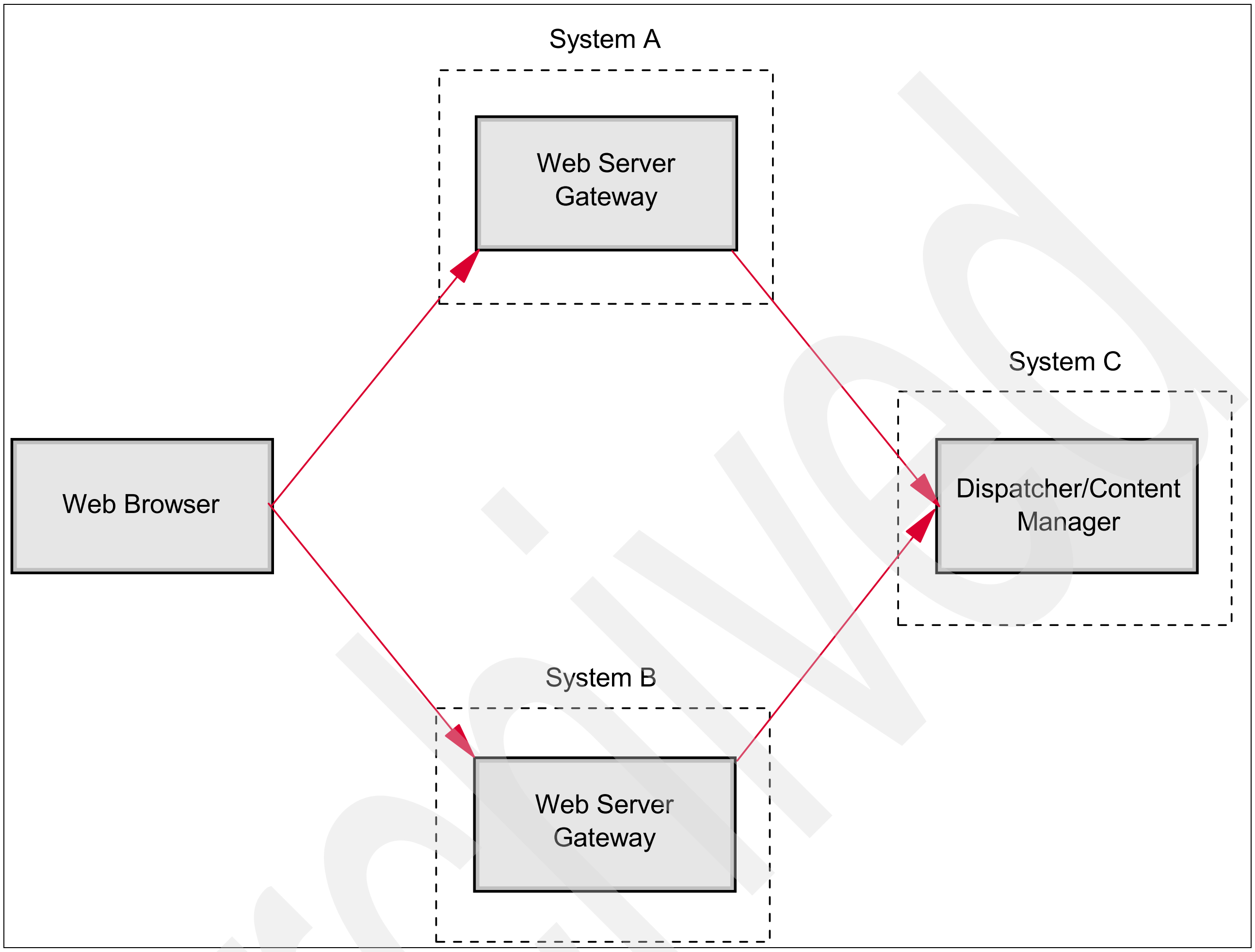


Figure 3-15 Installing the gateways on separate servers

In Figure 3-15, there are two web servers running the IBM Cognos 8 BI gateway. Using multiple web servers to manage incoming requests provides a better level of service. In this scenario, where the IBM Cognos gateway component is installed on the same computer as the web server, your web server manages the core web services. There is no load balancing mechanism at the level of the gateway in this implementation. To process requests, each web server/gateway has a separate URL, eliminating the advantage of having multiple instances of the gateway and achieving load balancing. You can use an external mechanism as a router or a network dispatcher to achieve load balancing at the level of the dispatcher.

3.5.3 Balancing the request load

In an IBM Cognos 8 environment, load balancing can be achieved at the gateway or report services level. Figure 3-16 shows multiple instances of each component installed. If the environment has multiple web servers in place, it is possible to consider using the existing servers to deploy the IBM Cognos gateway. Using multiple web servers provides a better level of service by managing the incoming requests, but does not provide load balancing for the IBM Cognos gateway service. At the dispatcher services level load balance is automatic in a distributed environment. When more than one report service is available, the dispatcher distributes the requests to all servers registered with the Content Manager.

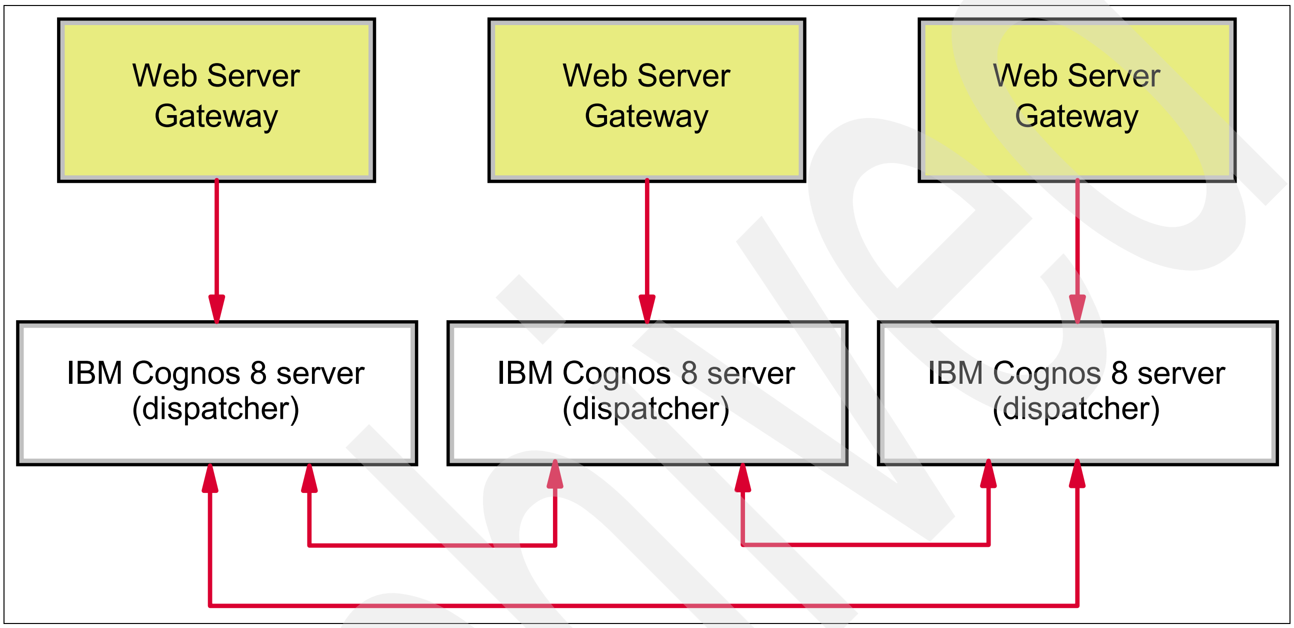


Figure 3-16 Multiple instances of the gateway and dispatcher installed

This chapter provided a description of the separate IBM Cognos 8 BI solution components and features. Chapter 4, “Environment installation and implementation” on page 47, describes how to configure the test environment with the IBM Power Systems virtualization features, the IBM software stack, IBM storage, and the IBM Cognos 8 BI solution.

4

Environment installation and implementation

This chapter describes several approaches to deploying the Cognos server. We start with the typical or reference deployment, which is easy to configure and easy to manage. In this configuration, a single Cognos installation is configured with all required services.

Later in this chapter, we explore the advanced and distributed implementation. Under the distributed deployment, separate Cognos services are installed in separate logical partitions (LPARs) to achieve higher throughput. The advanced configuration also enables us to leverage the IBM Power Systems' PowerVM virtualization features to achieve better scalability and performance.

The topics explored in this chapter are:

4.1 Overview

The deployment of the IBM Cognos Business Intelligence server requires various supporting software:

- A database engine
- An application server
- A web server

Cognos is built on a service-oriented architecture (SOA) for the benefit of integrating with other databases, applications, or web servers. In this scenario, we used a wide variety of IBM software solution offerings such as IBM DB2, IBM WebSphere Application server, IBM HTTP server, and the IBM Tivoli Directory Server (TDS).

The Cognos server itself is a combination of services, such as:

- Gateway service
- Dispatcher service
- Report service
- Content Manager

These services and the previously mentioned software stack components can be installed and configured under a single operating system (OS) image, or each component can be installed on separate OS image. In our environment, we use an AIX 6.1 TL 4 operating system hosted on several Power Systems LPARs.

In our scenario, the IBM HTTP server is an integral part of the Cognos gateway installation.

The DB2 server is installed on an LPAR with dedicated resources. Apart from the recommended configuration from Cognos, any additional tuning for performance optimization for DB2 is beyond the scope of this book. Also, the DB2 server partition is not part of any migration activity using Live Partition Mobility or Live Application Mobility. IBM TDS enables access authentication for the Cognos user directories. Users can securely access the Cognos content store and query database with login identification and a password, as opposed to the default anonymous access.

Under a typical installation, all components of Cognos are installed on the same LPAR. During a distributed deployment, certain major functionality components, such as the gateway server, report server, and Content Manager, can be deployed in separate systems or LPARs.

4.2 Reference deployment of Cognos server with one component per logical partition

In this section we explore the reference deployment of Cognos. The Content Manager and the dispatcher (with all Cognos services) are installed in a single LPAR. The HTTP server with the Cognos gateway server are installed in separate LPARs, and the DB2 server is installed in separate LPAR. To leverage the Power Systems' PowerVM virtualization features, we have one VIO server configured. Certain virtualization features, for example, Virtual I/O adapter, micro partitioning, Shared Ethernet Adapter (SEA), and Virtual LAN (VLAN), are used in this deployment.

All of these components are installed on the same POWER6® server. Figure 4-1 shows the topology of the installation.

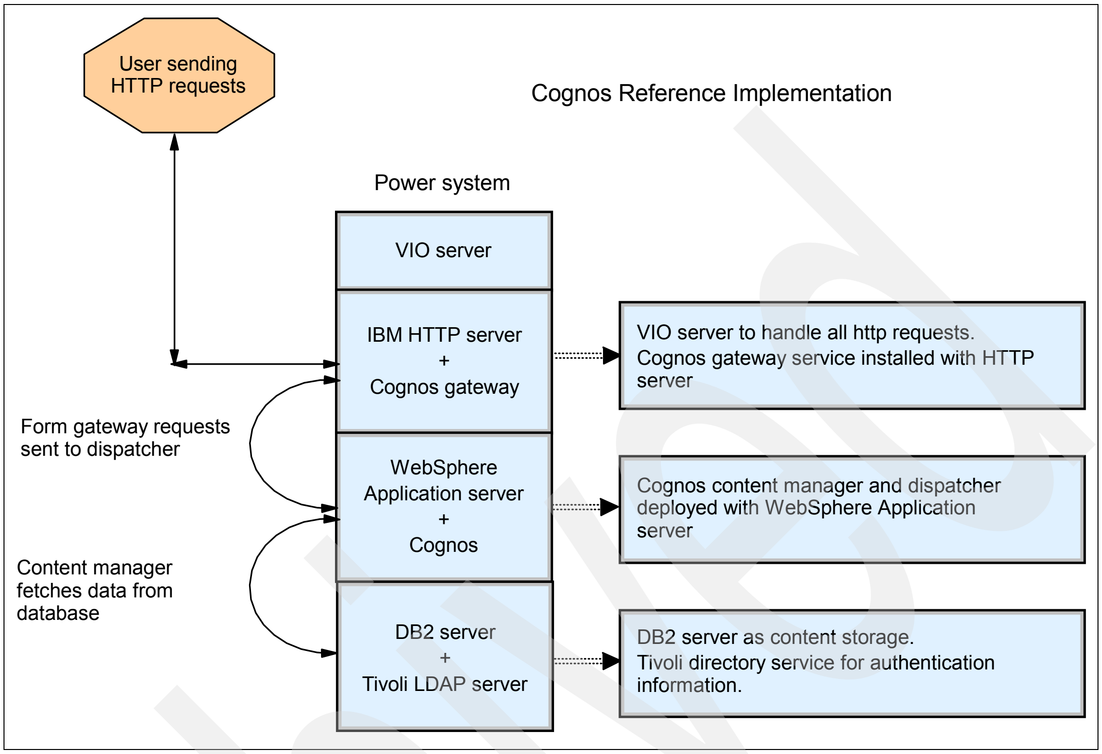


Figure 4-1 Functional diagram of reference implementation

4.2.1 IBM Power Systems servers

The POWER6 570 server with the PowerVM Enterprise License enabled is configured with eight CPUs and 48 GB of memory. With local SCSI devices, SAN devices, and a physical Ethernet adapter, this server has the capability to enable Active Memory Sharing and Live Partition Mobility. Figure 4-2 shows the hardware configuration for the reference deployment.

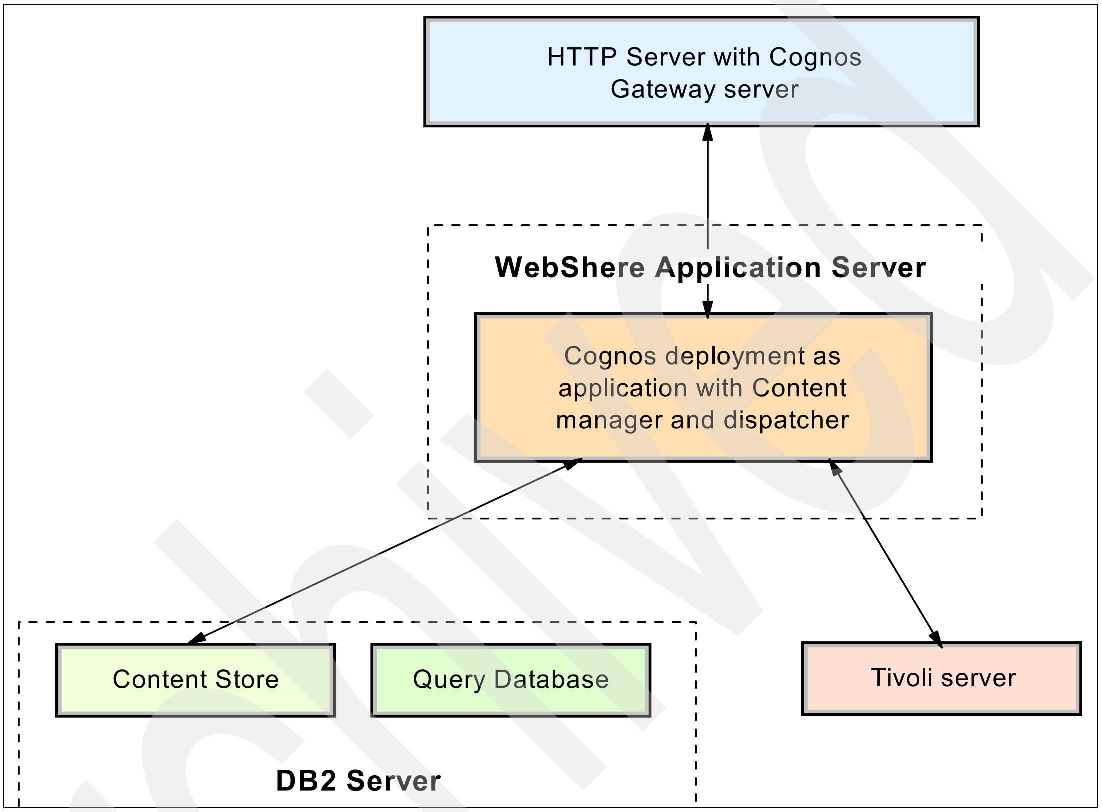


Figure 4-2 Diagram of Cognos Content Manager and dispatcher server in a single LPAR

4.2.2 Virtual I/O server

To explore several virtualization features of PowerVM, one VIO server is created with all the physical adapters. Virtual SCSI adapters, Virtual LAN (VLAN), and a Shared Ethernet Adapter (SEA) are configured to set up client LPARs.

This VIO server is also configured with Active Memory Sharing (AMS) to leverage memory virtualization features. Under the AMS configuration, a memory pool of 16 GB is created with the required number of paging devices. Both local SCSI and SAN devices are used to create these paging devices.

4.2.3 Gateway server

Cognos services are accessed through a web-based interface. A web server is required to handle all these HTTP requests. A typical web server can be a *servlet* server with a WebSphere application server or a *cgi-bin* server with an HTTP server.

We deploy the IBM HTTP server with the Cognos gateway service to set up the *cgi-bin* web server. Both of these components are installed on separate LPARs. During peak periods, the web server receives many HTTP requests. For high availability and stability, redundant web servers can be configured. From the gateway server, all requests are sent to the Cognos dispatcher.

4.2.4 Cognos server with the WebSphere Application Server

This is the application tier in the Cognos deployment. Several Cognos services such as dispatcher, Content Manager, and report service are installed in this tier. By default, Cognos is shipped with the Tomcat application server. However, Cognos services can be installed with any other application server. Important services under the Cognos application tier include:

- Dispatcher

  Receives all the requests from the Cognos gateway server, then based on the requested service sends it to the corresponding server or it connects to the Content Manager for database interaction.

- Content Manager

  Interface to access the content stored in the database. The Content Manager is responsible for finding all metadata from the database called the content store. The content store can be created on the database from content data. The JDBC driver (type-2) is used for the connectivity with the Content Manager.

- Report service

  This is one of the primary services that Cognos provides. Several reports are generated using this service.

Note: In our environment, we deploy Cognos with the WebSphere application server.

Both servers described above need a good amount of processing power (CPU and memory) to deliver higher throughput. Each of these LPARs is configured with adequate CPU and memory to meet the throughput requirements. These LPARs are connected to the same VLAN through a virtual Ethernet adapter using virtual SCSI devices.

4.2.5 Content storage and content database with an IBM DB2 server

Data is important to generate separate reports. In this case, a DB2 server is installed as the content storage database and the query database. The content store is used to store metadata for separate reports, and it is only accessed via the Content Manager.

The query database contains all the data to generate several graphs. The data is accessed from the report server.

4.2.6 Deployment reference model for Cognos

The Cognos deployment reference model primarily has three components:

- The database
- The Cognos server with the application tier
- The gateway server

Installation of each of these components is independent of each other. However, the configuration of the Content Manager needs the content storage available, so the database server should be deployed before the Content Manager is configured. Similarly, the gateway server needs the Content Manager available for retrieval of the security certificates. Follow the sequence given in Figure 4-3.

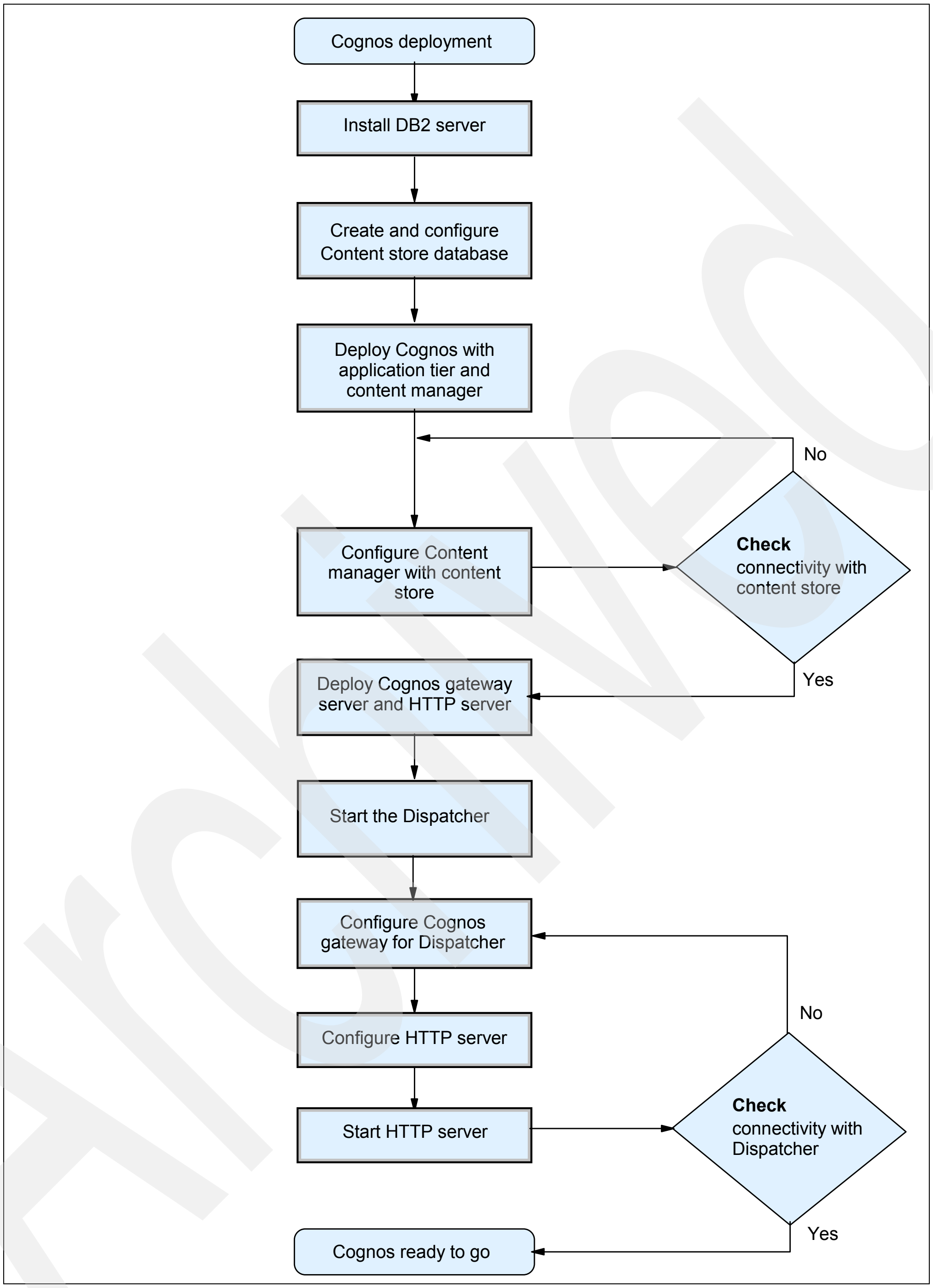


Figure 4-3 Flowchart for Cognos deployment with other software components

DB2 server installation

This section highlights the necessary steps to configure the DB2 server installation.

DB2 server installation

DB2 Server Version 9.7 is used as the database for Cognos deployment. The same DB2 server is used as the content store and the query database. The installation of the DB2 server can be done from a Java-based GUI installer named **db2setup**, which we used for server deployment. Apart from the X-manager, the **db2setup** installer requires a compatible web browser such as Firefox or Mozilla to start the installation wizard. With the help of a response file, the installer can be used in silent mode.

As the installation process starts, we need to select the desired server version (Figure 4-4). The default installation path for the DB2 server is `/opt` and can occupy up to 1800 MB of disk space.

Figure 4-4 DB2 server installation startup screen to select server version

The DB2 administrative server is an integral component of the DB2 database server, and it is configured during installation. This administrative service requires a special user and group that are also configured during the installation.

A DB2 instance is a logical database server environment. One user and administrative group is created to access this environment (Figure 4-5).

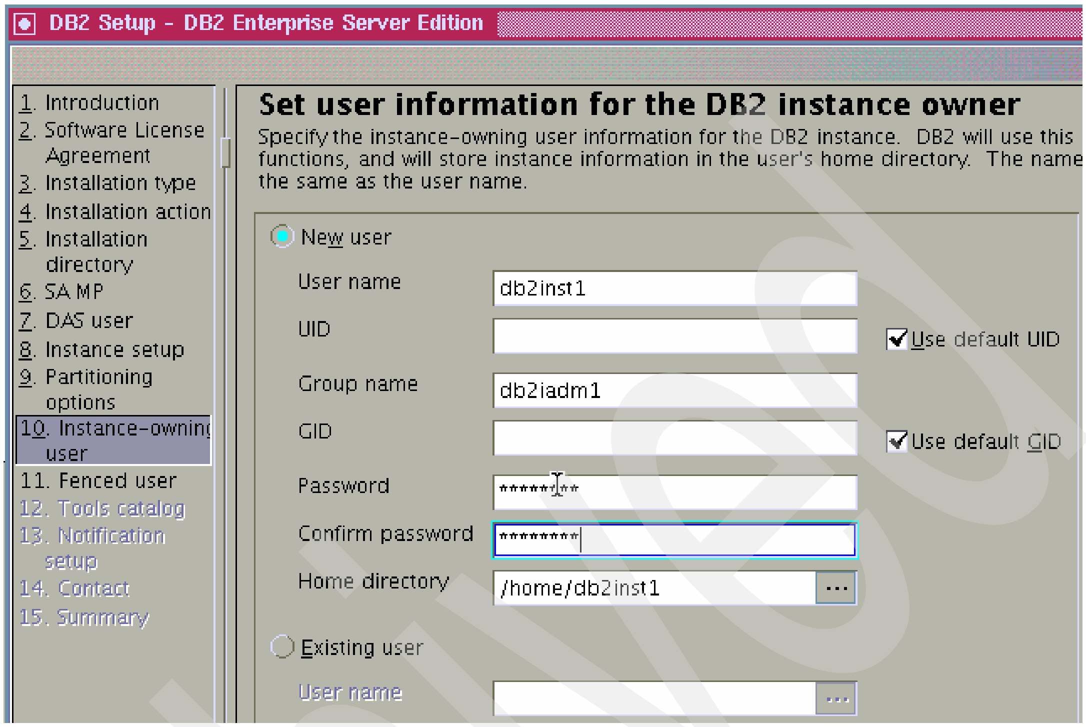


Figure 4-5 DB2 instance creation

After successful installation, we get the port number to be used by this instance to communicate with the DB2 server. As shown in Figure 4-6, the DB2 instance “db2inst1” with port 50000 is used for database connectivity. Also, user “db2inst1” is created for the AIX environment with the required database libraries. The DB2 installation can be verified using the **db2start** command from this instance (Example 4-1).

Example 4-1 db2start command

```
$ db2start
03/18/2010 21:55:18     0   0   SQL1063N  DB2START processing was successful.
SQL1063N  DB2START processing was successful.
$
```

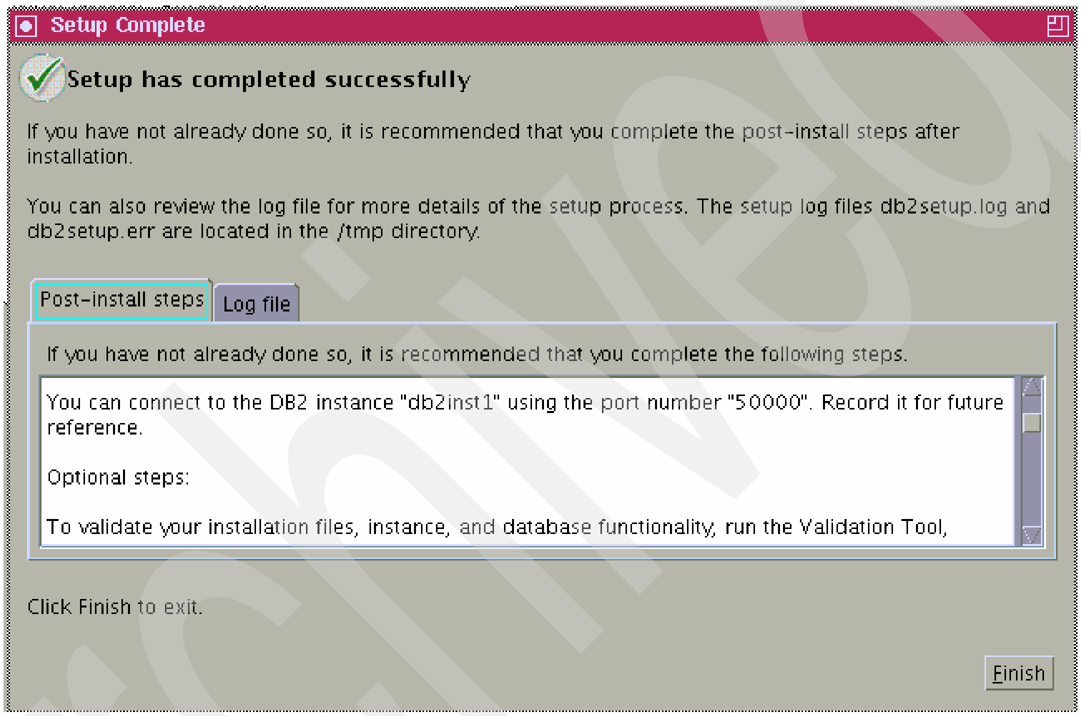


Figure 4-6 Installation log with user name and port number

Content store configuration

The Content Manager needs a database to store all metadata and cryptographic information. The Content Manager can only be activated if this database is available and configured properly. Example 4-2 lists the db2 commands used to create and configure the content store.

In Example 4-2, we create a content store with the name *cm* from the database instance “db2inst1”. Each Cognos deployment must have at least one active Content Manager with its unique content store database. For each Cognos deployment, there is one distinct content store that cannot be shared.

Example 4-2 db2 commands to create and configure content store

```
CREATE DATABASE cm ALIAS cm USING CODESET UTF-8 TERRITORY US

CHANGE DATABASE cm COMMENT WITH 'CM SINGLE'

CONNECT TO cm

UPDATE DATABASE CONFIGURATION USING APPLHEAPSZ automatic
```

```
UPDATE DATABASE CONFIGURATION USING LOCKTIMEOUT 240 DEFERRED

CONNECT RESET

CONNECT TO cm

CREATE BUFFERPOOL COG04KBP IMMEDIATE SIZE 250 PAGESIZE 4 K

CREATE BUFFERPOOL COG08KBP IMMEDIATE SIZE 250 PAGESIZE 8 K

CREATE BUFFERPOOL COG16KBP IMMEDIATE SIZE 250 PAGESIZE 16 K

CREATE BUFFERPOOL COG32KBP IMMEDIATE SIZE 250 PAGESIZE 32 K

CONNECT RESET

CONNECT TO cm

CREATE SYSTEM TEMPORARY TABLESPACE COGSYSTMP IN DATABASE PARTITION GROUP
IBMTEMPGROUP PAGESIZE 16 K MANAGED BY SYSTEM USING
('/home/db2inst1/db2inst1/NODE0000/cm/COGSYSTMP01') EXTENTSIZE 16 PREFETCHSIZE 16
OVERHEAD 10.5 TRANSFERRATE 0.14 BUFFERPOOL COG16KBP

CREATE USER TEMPORARY TABLESPACE COGUSRTMP IN DATABASE PARTITION GROUP
IBMDEFAULTGROUP PAGESIZE 32 K MANAGED BY SYSTEM USING
('/home/db2inst1/db2inst1/NODE0000/cm/COGUSRTMP01') EXTENTSIZE 32 PREFETCHSIZE 32
OVERHEAD 10.5 TRANSFERRATE 0.33 BUFFERPOOL COG32KBP

CREATE REGULAR TABLESPACE COGUSRDAT IN DATABASE PARTITION GROUP IBMDEFAULTGROUP
PAGESIZE 8 K MANAGED BY SYSTEM USING
('/home/db2inst1/db2inst1/NODE0000/cm/COGUSRDAT01') EXTENTSIZE 8 PREFETCHSIZE 8
OVERHEAD 10.5 TRANSFERRATE 0.33 BUFFERPOOL COG08KBP DROPPED TABLE RECOVERY ON

CREATE REGULAR TABLESPACE COGUSRLOG IN DATABASE PARTITION GROUP IBMDEFAULTGROUP
PAGESIZE 4 K MANAGED BY SYSTEM USING
('/home/db2inst1/db2inst1/NODE0000/cm/COGUSRLOG01') EXTENTSIZE 4 PREFETCHSIZE 4
OVERHEAD 10.5 TRANSFERRATE 0.33 BUFFERPOOL COG04KBP DROPPED TABLE RECOVERY OFF

ALTER TABLESPACE TEMPSPACE1 PREFETCHSIZE 16 OVERHEAD 24.100000 TRANSFERRATE
0.900000

ALTER TABLESPACE USERSPACE1 PREFETCHSIZE 16 OVERHEAD 24.100000 TRANSFERRATE
0.900000

CONNECT RESET
```

DB2 client installation

The DB2 client needs to be installed on several remote systems to access the database. The Content Manager needs this client to access the content store, and the report server needs this client to access the query database. This client comes with the server package, and it is installed using the same installer (Figure 4-4 on page 53). A database instance is also created for the database client. To access the database at the server side, we need to catalog it (Example 4-3).

Example 4-3 catalog commands from db2 client

```
catalog TCPIP NODE cm remote 192.168.100.202 SERVER 50000
catalog DATABASE cm at node cm with "CM single”
```

Once the client is connected to the database, we can verify it using the **`list application`** command from the server database instance. In Example 4-4, we find that the client 192.168.100.144 is accessing the database CM using the db2 instance named DB2INST1.

Example 4-4 Verifying client connection from server side

```
db2 => list application
DB2INST1 db2jcc_applica 829       192.168.100.244.45996.100318191 CM       1
```

Tivoli Directory Server installation

For Cognos user access and authentication, Tivoli Directory Server (TDS) V6.2 is used. The software comes as *installp* packages. Figure 4-7 lists the file sets.

```
p5701db2(root)/INST/tdsV6.2/installp/ppc> installp -l -d .
  Fileset Name                Level                     I/U Q Content
  ====================================================================
gskjs.rte                   7.0.4.14                    I  N usr
#   AIX Certificate and SSL Java only Base Runtime
gsksa.rte                   7.0.4.14                    I  N usr
#   AIX Certificate and SSL Base Runtime ACME Toolkit
gskta.rte                   7.0.4.14                    I  N usr
#   AIX Certificate and SSL Base Runtime ACME Toolkit
idsldap.clt32bit62.rte      6.2.0.0                     I  N usr,root
#   Directory Server - 32 bit Client
idsldap.clt64bit62.rte      6.2.0.0                     I  N usr,root
#   Directory Server - 64 bit Client
idsldap.clt_max_crypto32bit62.rte 6.2.0.0                     I  N usr
#   Directory Server - 32 bit Client (SSL)
idsldap.clt_max_crypto64bit62.rte 6.2.0.0                     I  N usr
#   Directory Server - 64 bit Client (SSL)
idsldap.cltbase62.adt       6.2.0.0                     I  N usr
#   Directory Server - Base Client
idsldap.cltbase62.rte       6.2.0.0                     I  N usr,root
#   Directory Server - Base Client
idsldap.cltjava62.rte       6.2.0.0                     I  N usr
#   Directory Server - Java Client
idsldap.ent62.rte           6.2.0.0                     I  N usr
#   Directory Server - Entitlement

   idsldap.msg62.en_US         6.2.0.0                     I  N usr
#   Directory Server - Messages - U.S. English (en)

#   Directory Server - Messages - for supporting multi-languages

  idsldap.srv64bit62.rte      6.2.0.0                     I  N usr
#   Directory Server - 64 bit Server
idsldap.srv_max_cryptobase64bit62.rte 6.2.0.0                     I  N usr
#   Directory Server - base Server (SSL)
idsldap.srvbase64bit62.rte  6.2.0.0                     I  N usr,root
#   Directory Server - Base Server
idsldap.srvproxy64bit62.rte 6.2.0.0                     I  N usr,root
#   Directory Server - 64 bit Proxy Server
idsldap.webadmin62.rte      6.2.0.0                     I  N usr
#   Directory Server - Web Administration
idsldap.webadmin_max_crypto62.rte 6.2.0.0                     I  N usr
#   Directory Server - Web Administration (SSL)
```

Figure 4-7 Tivoli directory server installp images

Note: The DB2 server is required on the same machine on which TDS is going to be installed. Otherwise, the installation fails because remote database connectivity is not supported. This database server can be in the same LPAR or a separate LPAR from the Cognos database server.

The **smitty** command can be used to install all the packages. Figure 4-8 shows a snapshot of the SMIT screen.

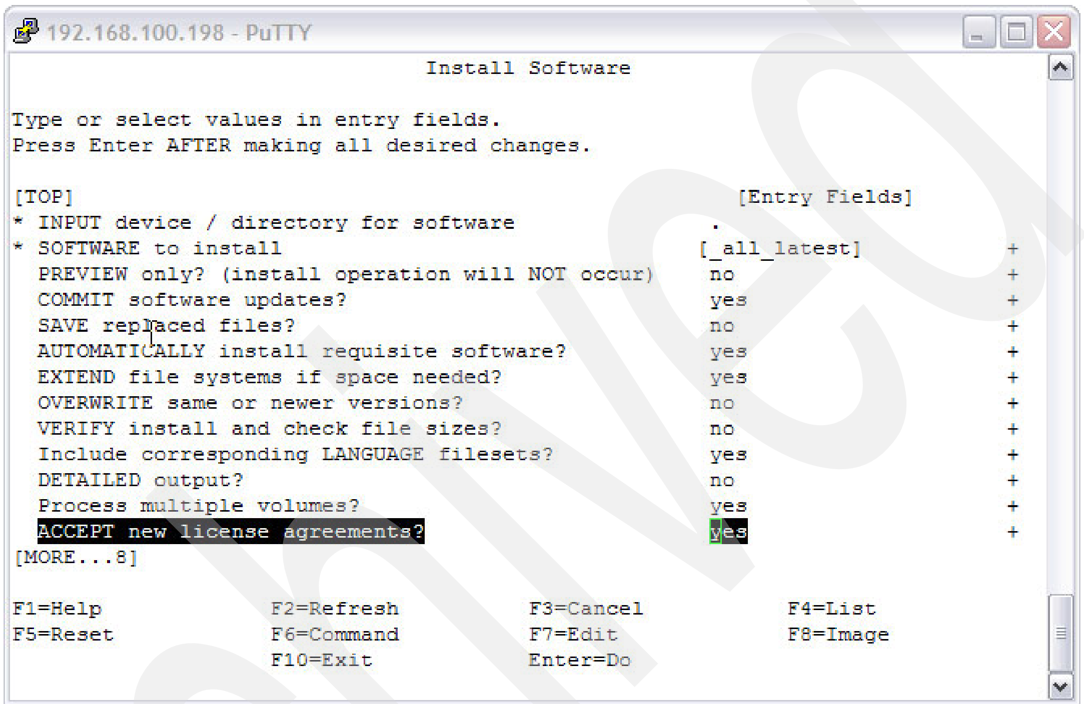


Figure 4-8 Installing TDS V6.2 using smitty

After the installation completes, run the Tivoli Directory Server configuration tool (Example 4-5) to start the server and configure LDAP.

Example 4-5 Running the Tivoli configuration to start the server and configure LDAP services

```
# /opt/IBM/ldap/V6.2/sbin/idsxcfg
```

Note: The command **idsxcfg** requires XWindows or Xserver to run.

Figure 4-9 shows the all-in-one configuration interface tool.

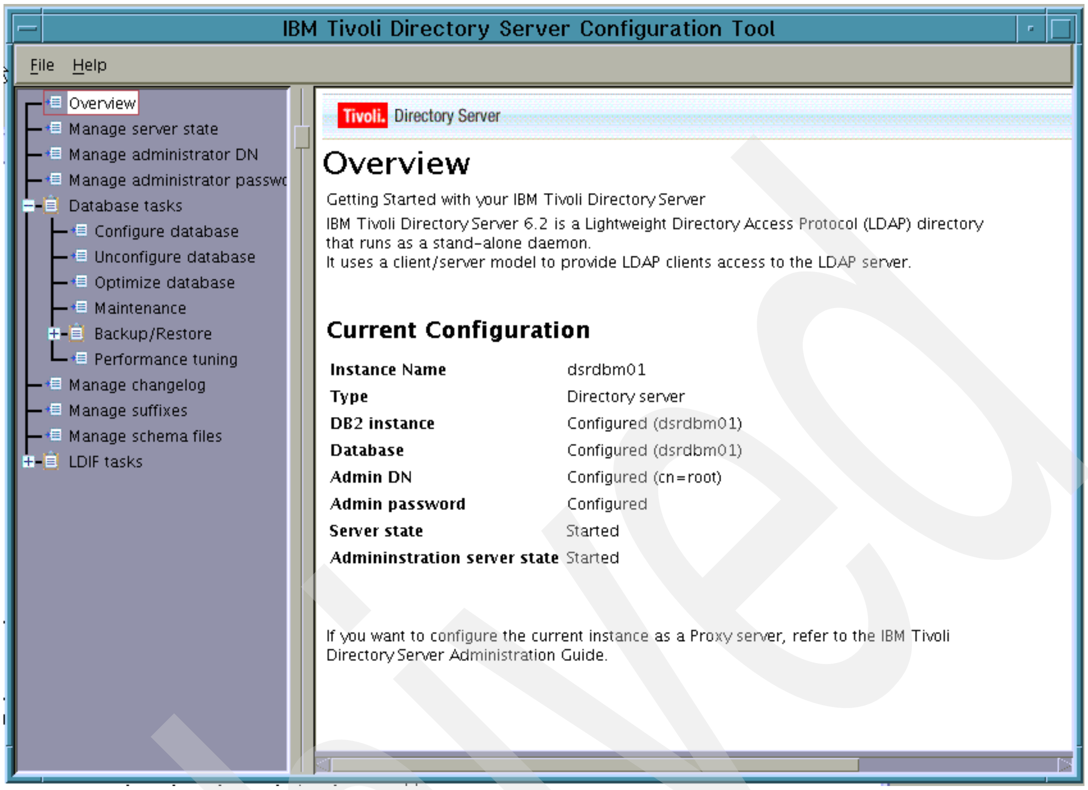


Figure 4-9 IBM Tivoli Directory Server configuration tool

Clicking the **Manage server state** menu option on the left-hand side opens the panel on the right-hand side for starting the server if it is not running and vice versa. This panel provides the server status before any action is carried out (Figure 4-10). The information presented includes the TDS instance name and a short description, the state of the directory server and the administration server, and the start time and elapsed time if the server has already started.

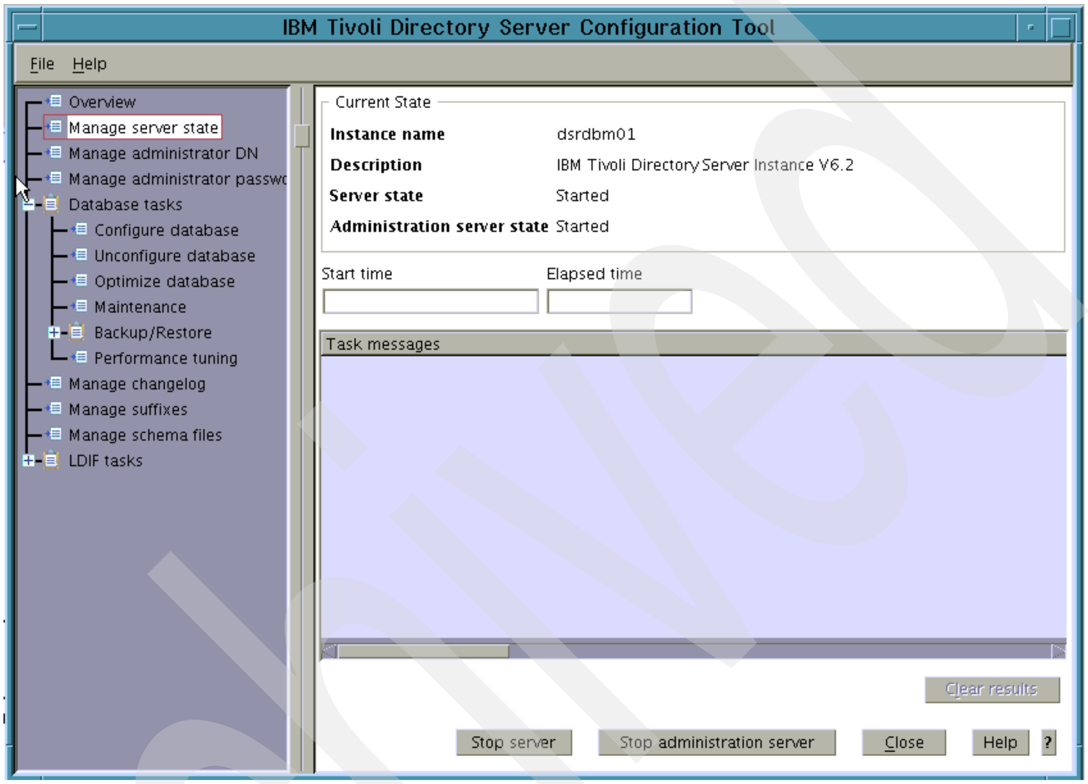


Figure 4-10 Managing TDS state

Once the server is configured, the remaining task is to create user IDs or import them from a LDIF file.

Cognos deployment

Cognos deployment starts with the installation through a GUI-based installer called issetup. The LPAR must have an X-manager and either Mozilla or Firefox installed. Under the reference implementation, a single Cognos installation contains the Content Manager, dispatcher, and other services. Thus, during installation only these components are selected (Figure 4-11). The installation requires around 1.5 GB of disk space.

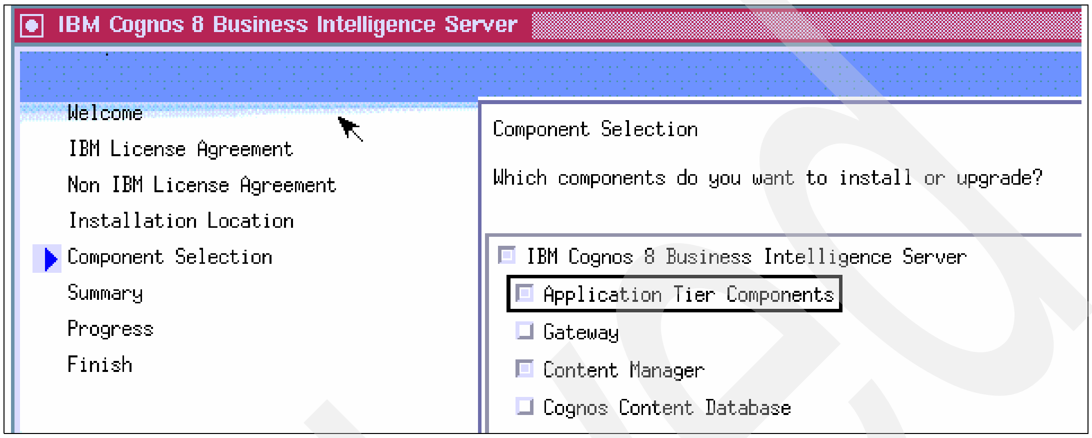


Figure 4-11 Cognos installer with components selected for reference installation

After successful installation, copy the `bcprov-jdkxx-yyy.jar` from the Cognos installed location to the Java library path. As given in Example 4-6, the `bcprov-jdk14-134.jar` is copied to JAVA_HOME, which is set to `/usr/java5_64/jre`. This jar file supports the cryptography algorithm for security certificates. Also, we need to append `<cognos_install_path>/bin64` (for 64 bit Cognos) and `<cognos_install_path>/cgi-bin` to LD_LIBRARY_PATH and LIBPATH environment variables.

Example 4-6 Copying the jar file for cryptography support

```
cp /usr/cognos/c8_64/bin64/jre/1.5.0/lib/ext/bcprov-jdk14-134.jar
/usr/java5_64/jre/lib/ext
export LD_LIBRARY_PATH
=/usr/cognos/c8_64/bin64:/usr/cognos/c8_64/cgi-bin:$LD_LIBRARY_PATH
```

The DB2 client setup is another prerequisite before the Content Manager is configured with the content storage. Details of the DB2 client installation are provided in “DB2 server installation” on page 52.

When all these configurations steps are completed, start the dispatcher and the Content Manager configuration using the cogconfig.sh script. This opens an interface (Figure 4-12) in which to set several cognos parameters.

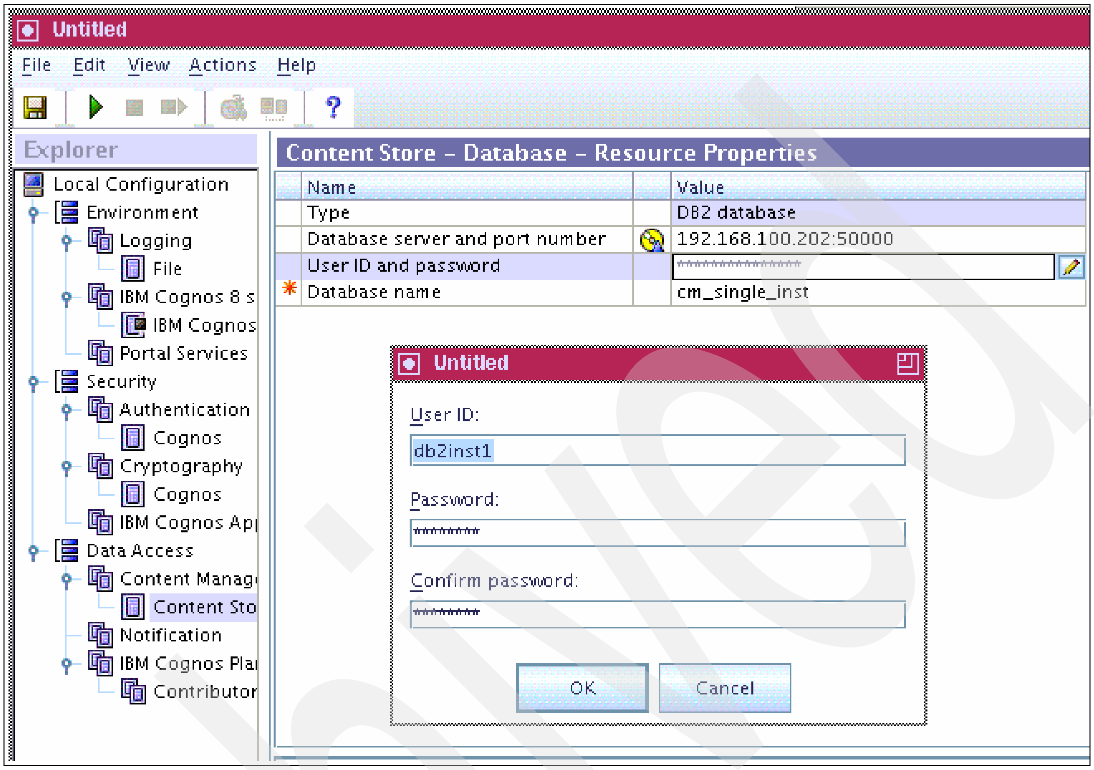


Figure 4-12 Configuring the Content Manager with the content storage parameters

Content Manager configuration

Under the Content Manager, we find *content store*. In the right-hand pane, enter the DB2 server and JDBC port number, followed by the DB2 server instance name with the password. In our environment, db2inst1 is the instance name and the content storage name is “cmsingle”.

The Content Manager uses the JDBC driver to communicate with the DB2 server. `db2jcc.jar` and `db2jcc_license_cu.jar` are the files containing the required drivers. Copy these two files to the `lib` directory under Cognos deployment path. The driver files are shipped with the DB2 server or client installation. The Cognos 8 servers require 32-bit JDBC drivers for DB2. Thus, the CLASSPATH variable needs to be configured so that the 32-bit JDBC drivers can be found first.

Example 4-7 JDBC driver configuration for Cognos Content Manager

```
cp /opt/IBM/db2/V9.7/java/db2jcc.jar /usr/cognos/c8_64/webapps/p2pd/WEB-INF/lib
cp /opt/IBM/db2/V9.7/java/db2jcc_license_cu.jar
/usr/cognos/c8_64/webapps/p2pd/WEB-INF/lib

ADD THE CLASSPATH DETAILS TO CHOW ONLY 32 bit DB2 LIBS are used to make DB2
Content Manager V8.4 and its APIs work correctly, the path to 32-bit libraries
must be listed before the path to the 64-bit libraries in LD_LIBRARY_PATH or
LIBPATH..
```

After all the properties are set and before saving the new Cognos profile, check for connectivity. Select **Content Store** and right-click to select the **test** option. The Content Manager is able to contact the content store database (Figure 4-13).

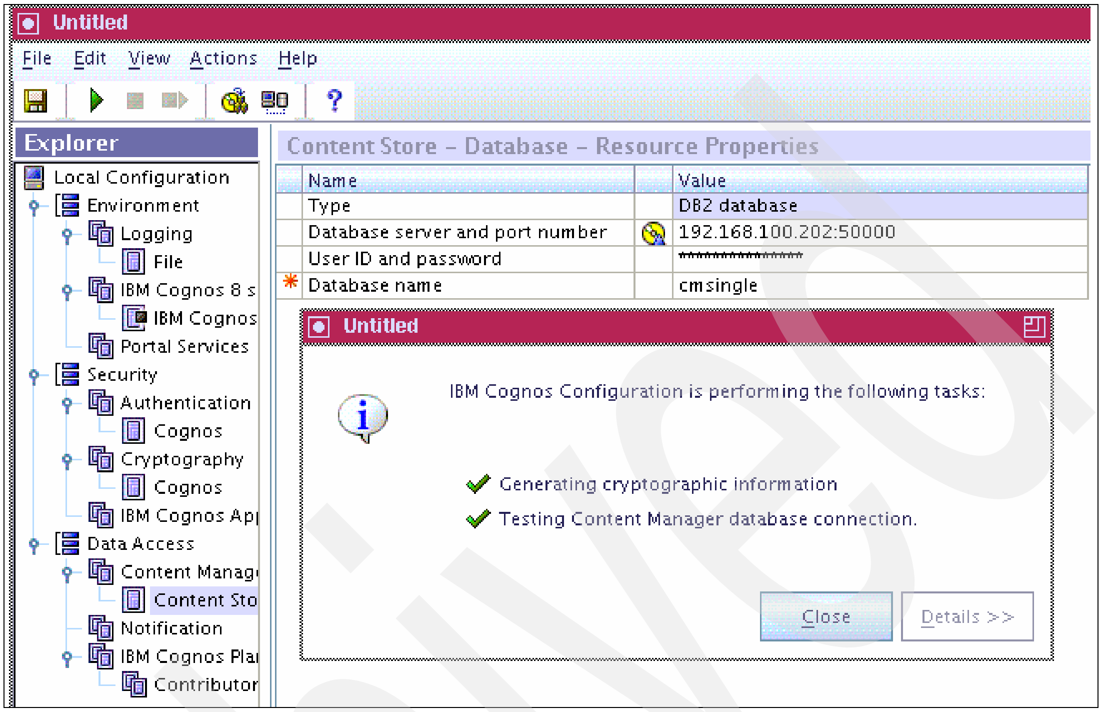


Figure 4-13 Testing Content Manager connectivity with content store database

Dispatcher configuration

The dispatcher configuration parameters are available under the Environment section. In this deployment, both the dispatcher and the Content Manager are located on the same system. We can specify localhost for the Content Manager and several dispatcher URIs. By default, Cognos is deployed with the Tomcat application server with default port 9300. We need to provide the IP address and port number for the web server, for example, the gateway server. In our case, we have a web server configured on 192.168.100.243 listening on HTTP port 80 (Figure 4-14).

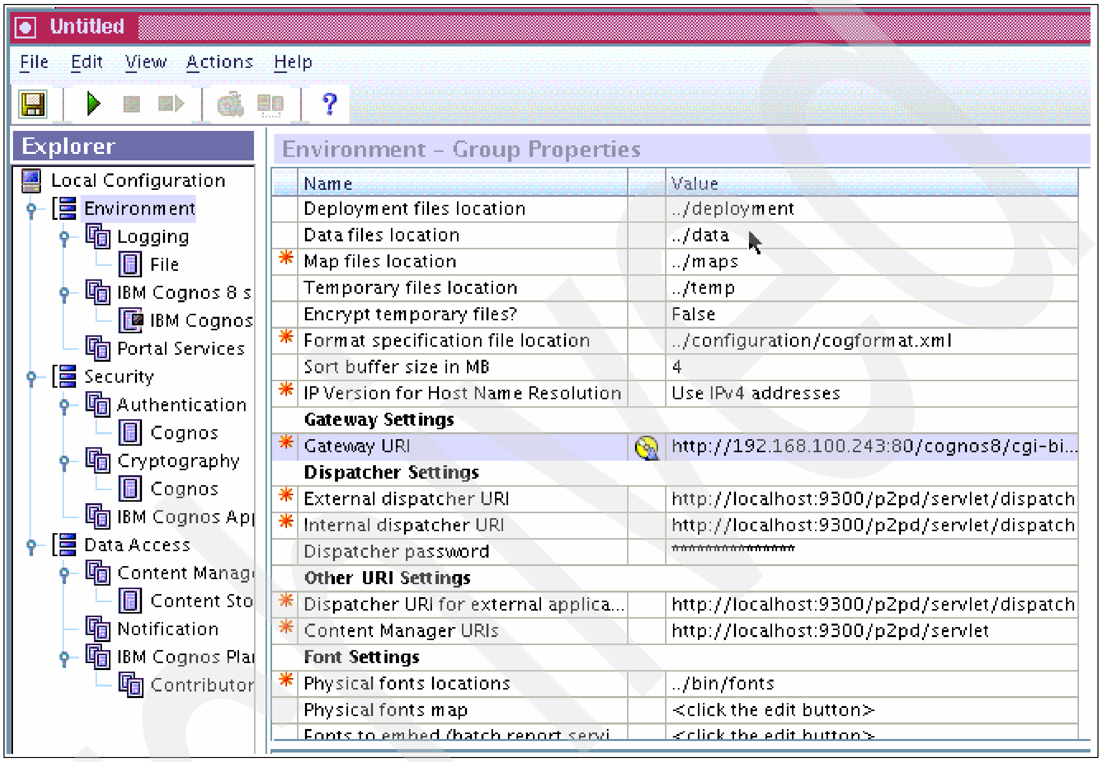


Figure 4-14 Cognos dispatcher configuration

Save all changes before starting the Cognos service. The Cognos service is available under IBM Cognos 8 service and is specific to the deployment with Tomcat, but it is not required for WebSphere Application Server. During Cognos startup, it performs a series of operations such as checking for Content Manager database connectivity, checking the status for configuration setup, and finally starting the service. Figure 4-15 shows the details.

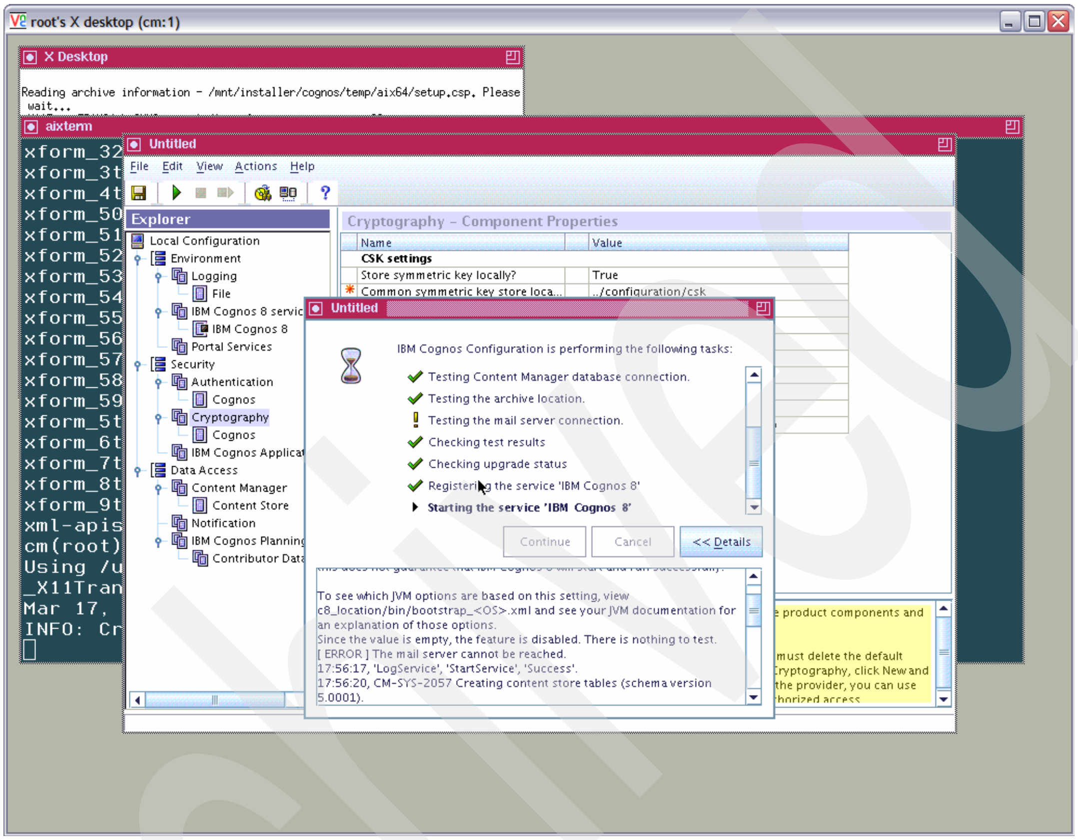


Figure 4-15 Cognos service startup process

Once the Cognos server is started from the IBM Cognos 8 service property, we can check that several services are running as part of the server. Figure 4-16 shows the list of services.

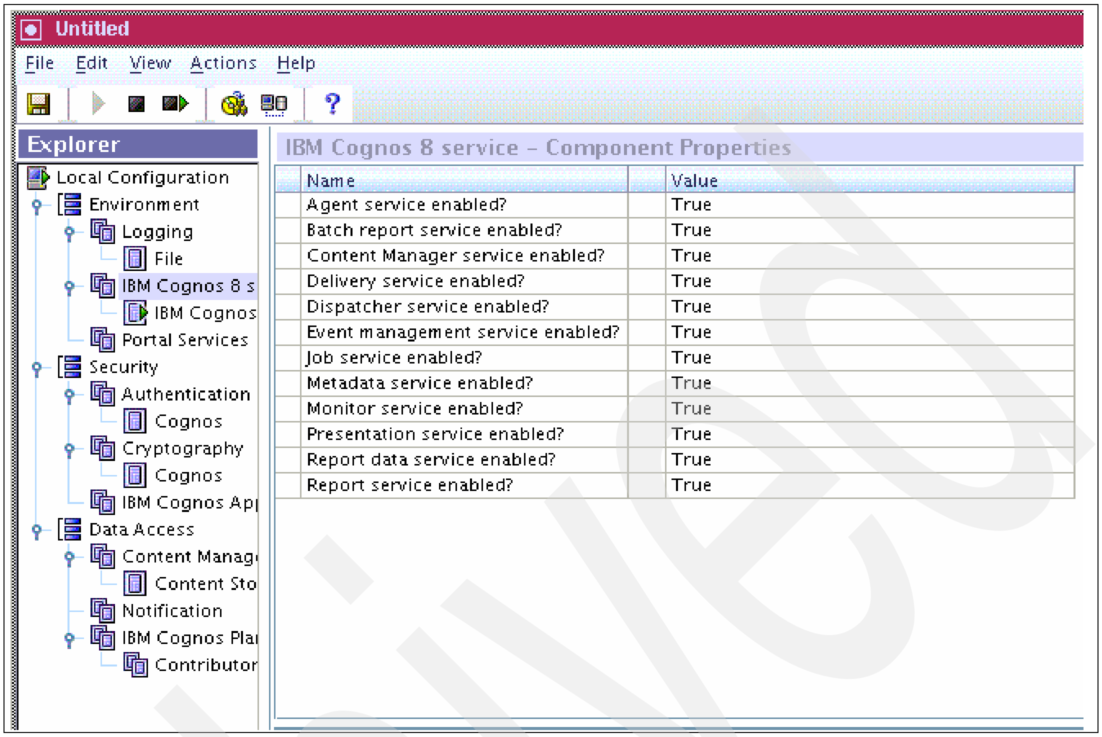


Figure 4-16 Cognos services configured under a deployment

To disable or enable one or more services, set the property accordingly. Then the configuration is saved once again, and the Cognos server is restarted for the changes to take effect. Under a single server deployment, we need all the services running. However, in distributed deployment, we can start or stop selected services for a particular Cognos server service. For more information see 4.3, “Distributed deployment of Cognos server” on page 74.

We check the installation status for dispatcher and Content Manager from their respective URLs. For the dispatcher use:

`http://<cognos_server_ip>:<application server port>/p2pd/servlet/dispatch`

For the Content Manager use:

`http://<cognos_server_ip>:<application server port>/p2pd/servlet/`

If both URLs respond, all the services are configured properly (Example 4-8).

Example 4-8 Dispatcher and Content Manager URLs to check status

```
Dispatcher: http://192.168.100.244:9300/p2pd/servlet/dispatch
Content manager: http://192.168.100.244:9300/p2pd/servlet/dispatch
```

Cognos configuration with IBM WebSphere Application Server

Cognos distribution file sets are packaged with the Tomcat application server. However, to achieve better performance and high scalability, we need an advanced application server such as IBM WebSphere Application Server (WAS). This section provides the configuration details for implementing Cognos with the IBM WebSphere Application Server.

The integration starts with the installation of the WebSphere Application Server. WAS is installed on the same LPAR where we implemented the Cognos Content Manager or dispatcher or any other deployment installed. Also, the Content Manager is configured with the content store database. The database client must be configured if this Cognos deployment needs to query the database. Before migrating to the WebSphere Application Server, we confirm that the Cognos deployment is working properly with the default Tomcat configuration.

Once migrated to the WebSphere Application Server, all cryptographic information is regenerated. Back up all existing cryptographic information by exporting the **Cognos Configuration ∅ File ∅ Export As** to a XML file (Figure 4-17). Click **Yes** to stop the Cognos service after the export is completed successfully.

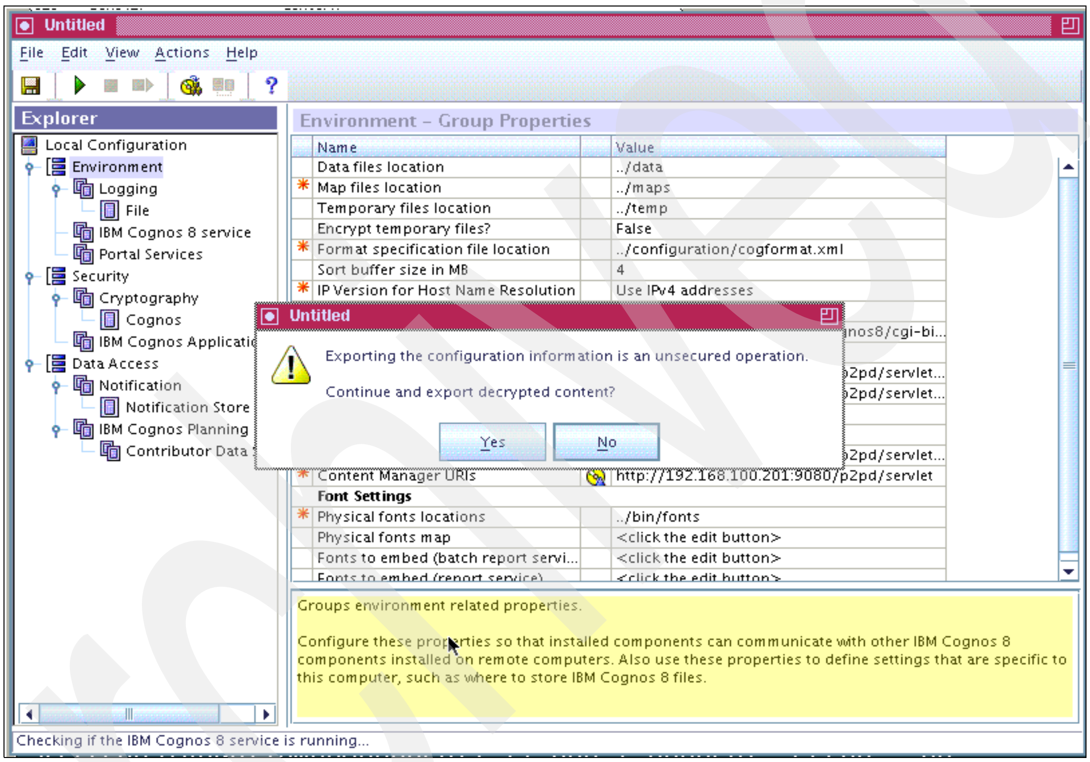


Figure 4-17 Exporting Cognos configuration before migrating to WebSphere Application Server

From the directory called `configuration` in the Cognos installation path, take a backup of all files and directories:

- `cogstartup.xml`
- `caSerial`
- `cogconfig.prefs`
- `coglocale.xml`
- `csk`
- `encryptkeypair`
- `signkeypair`

Delete the `caSerial` and `cogconfig.prefs` files and the three directories:

- `csk`
- `encryptkeypair`
- `signkeypair`

Replace `cogstartup.xml` with the exported property file.

The WebSphere Application Server comes with a java distribution, and the JAVA_HOME path must be set to point to this java distribution. The LD_LIBRARY_PATH and LIBPATH environment variables must be appended with the *bin64* and the *cgi-bin* path from the Cognos installed location. Now Cognos is started with the Java from the WebSphere distribution. Thus `bcprov-jdkxx-yyy.jar` is copied to the new Java path to provide encryption capability required for Cognos.

Start the Cognos configuration using `cogconfig.sh` to create the EAR file that is deployed on the WebSphere Application Server. The EAR file is generated using **Action ∅ Build Application Files**. This creates a `p2pd.ear` file (Figure 4-18). The context root value entered in the wizard must be the same as the one deployed to the application server. For IBM Cognos 8, the default context root and application directory name is *p2pd*.

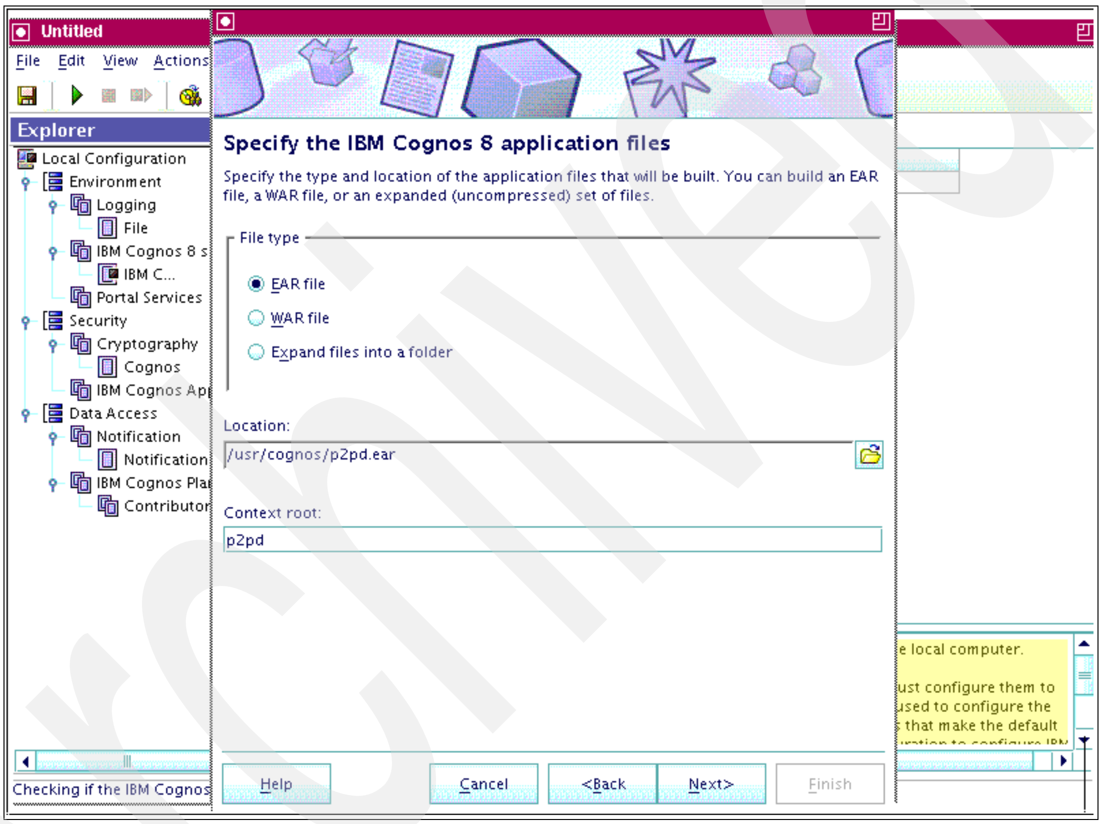


Figure 4-18 Generate p2pd.ear file to be deployed on WebSphere Application server

After the EAR file is created, the dispatcher URL is set with the IP address and host name of the server. Also, we enter the WAS port number configured for the application. In our case this is port 9080. Delete the Tomcat service under **Environment ∅ IBM Cognos 8 service**, right-click **IBM Cognos 8**, and select **Delete operation**. This service is not required for the WebSphere Application Server. Save the new Cognos configuration. New cryptographic keys are generated using the JVM packaged with WebSphere.

Now it is time to deploy the EAR file to the WebSphere Application Server. From the admin console select **Install New Application** and select the path for the `p2pd.ear` from the remote file system (Figure 4-19). We go with the same context root as exported (p2pd). Mostly we go with all the default values during the EAR deployment. Once completed, we find that the IBM Cognos 8 application has been added to the WebSphere Application Server. Save the new configuration and restart the WebSphere Application Server.

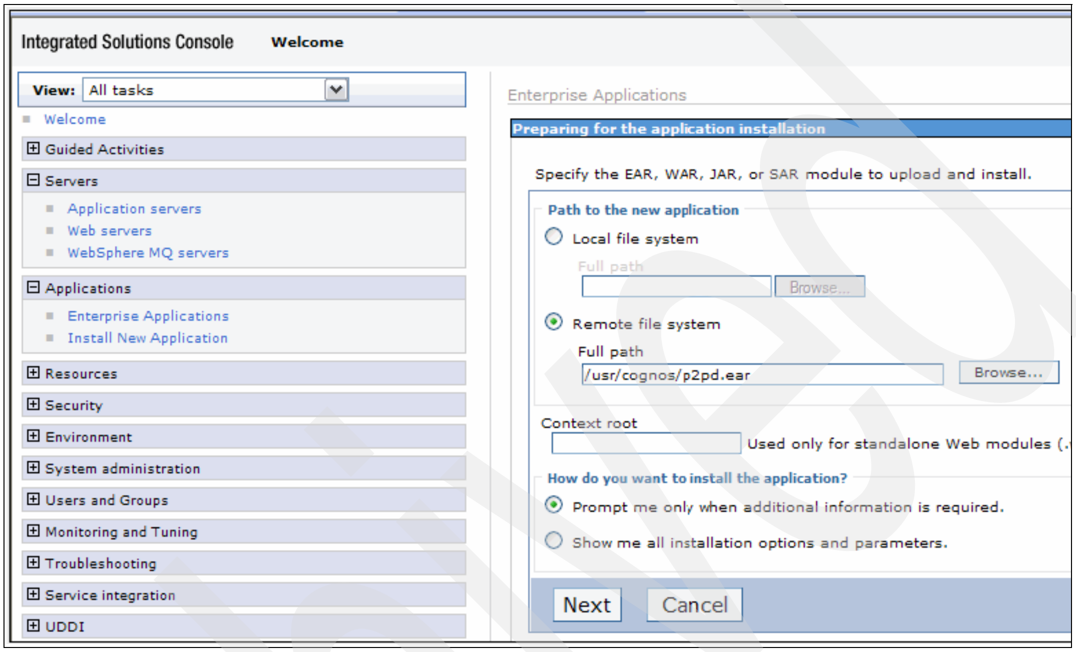


Figure 4-19 Deploying the p2pd.ear to the WebSphere Application Server

The dispatcher status with the WebSphere Application Server is verified from the server log. In the `SystemOut.log` file we get the following message (Figure 4-20):

```
The dispatcher is ready to process requests message
```

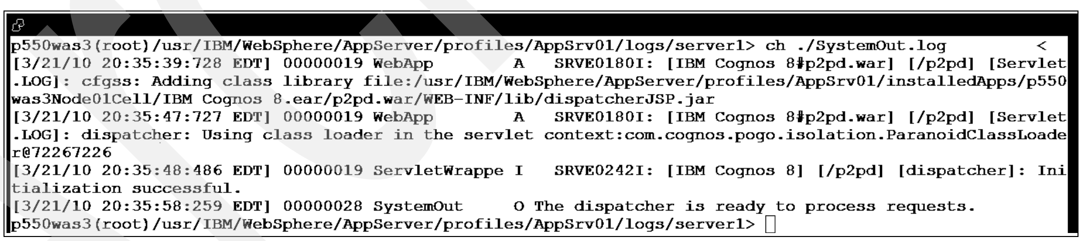


Figure 4-20 Dispatcher status from WebSphere server log file

You can check the status from the dispatcher URL:

`http://<server_IP_address>:<port>/p2pd/servlet/dispatch`

Also, you can check the status from WebSphere Application Server's admin console (Figure 4-21).

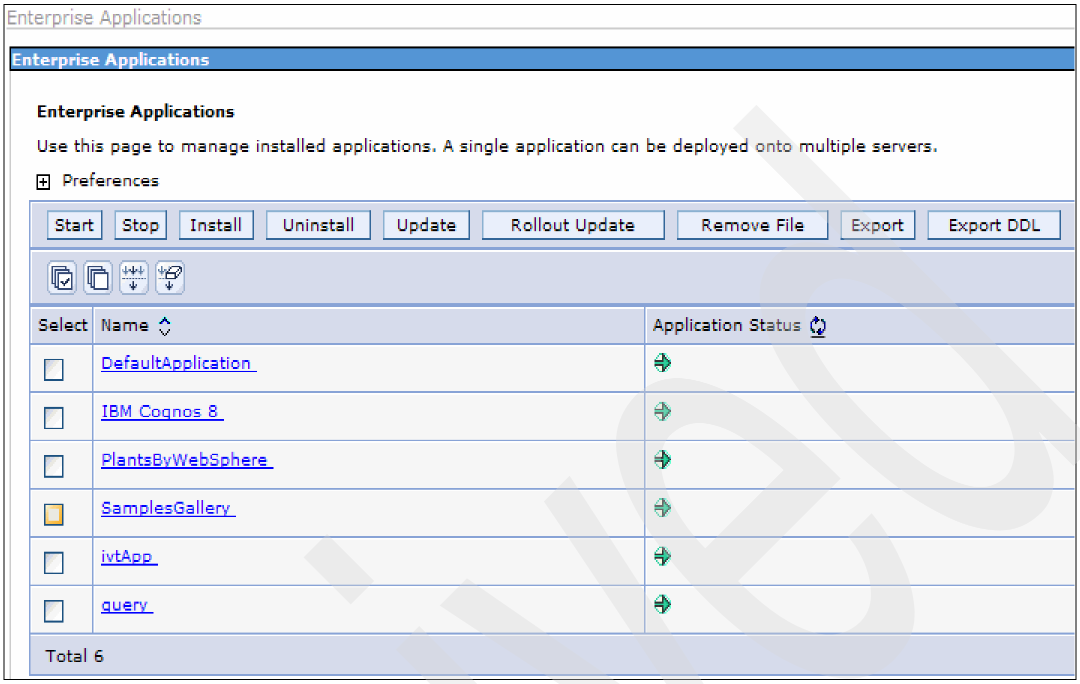


Figure 4-21 Deployment status of IBM Cognos 8 from the WebSphere admin console

Gateway configuration

The gateway server is the interface to the user and receives HTTP requests. We can configure two separate classes of gateways:

- One with the *servlet* server
- The other with the *cgi-bin* server

We use the *cgi-bin* server implementation in this exercise. The gateway server is composed of the IBM HTTP server and the Cognos gateway service.

The IBM HTTP server

The IBM HTTP server is available as a package with its own installer. Once installed, the HTTP server is started using a script called apachectl located in the `bin` directory from the installed path. On a successful start, we see a daemon called httpd. The default port for the httpd daemon is 80. On startup, the HTTP server reads all the configuration properties from the `httpd.conf` file located in the "`conf`" directory in the HTTP installation path. We create virtual directories for the Cognos gateway server (Example 4-9).

Example 4-9 Configuration for the HTTP server with virtual directories for Cognos

```
ScriptAlias /cognos8/cgi-bin "/usr/cognos/c8_64/cgi-bin"

<Directory "/usr/cognos/c8_64/cgi-bin/">
    AllowOverride None
    Options +ExecCGI
    Order allow,deny
    Allow from all
</Directory>

Alias /cognos8 "/usr/cognos/c8_64/webcontent"
```

```
<Directory "/usr/cognos/c8_64/webcontent">
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
```

The Cognos gateway server

Only the Cognos gateway server component is installed in this particular LPAR (Figure 4-22).

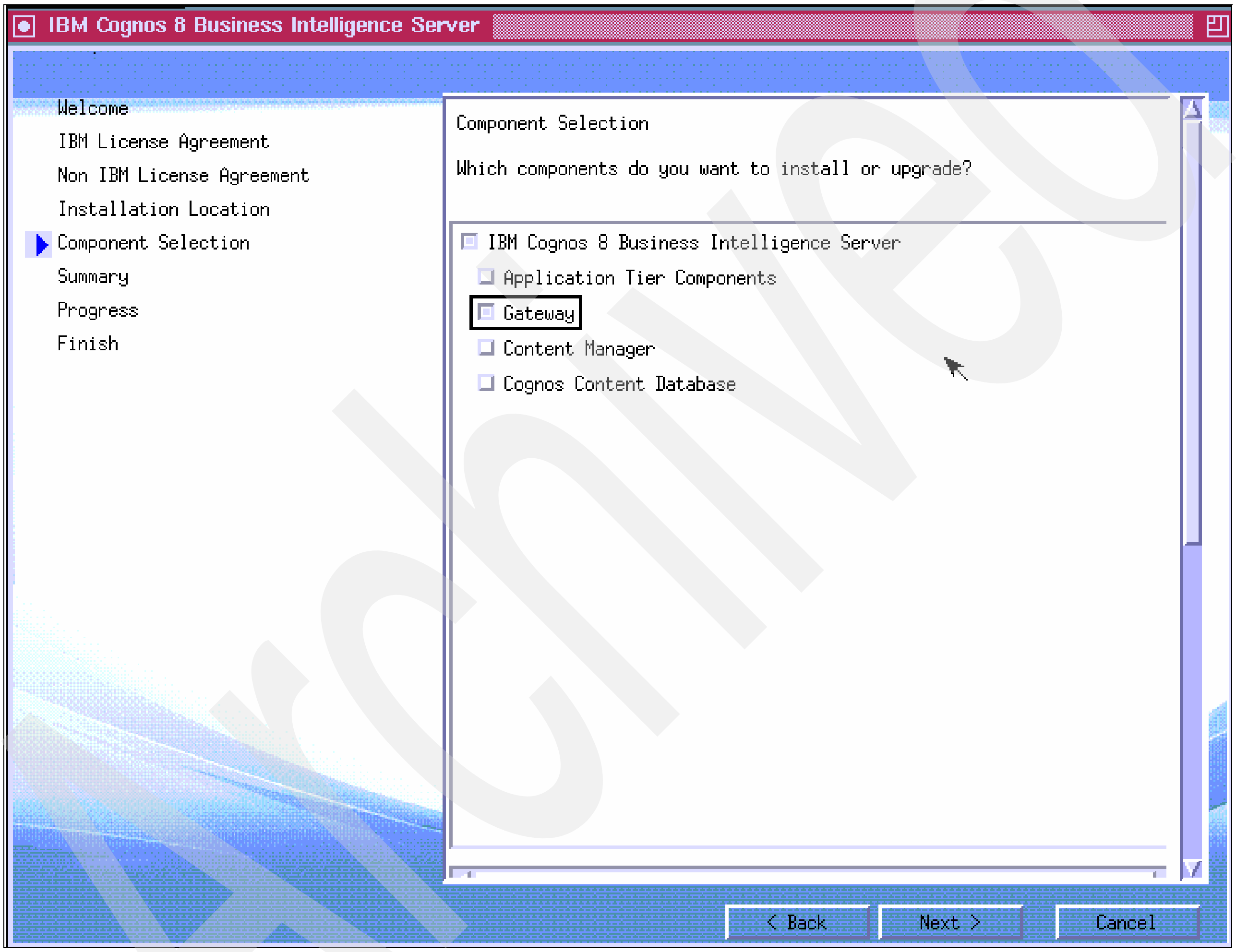


Figure 4-22   Cognos gateway server installation

Once the installation is completed successfully, we configure the Cognos gateway with the URL of the dispatcher (Figure 4-23).

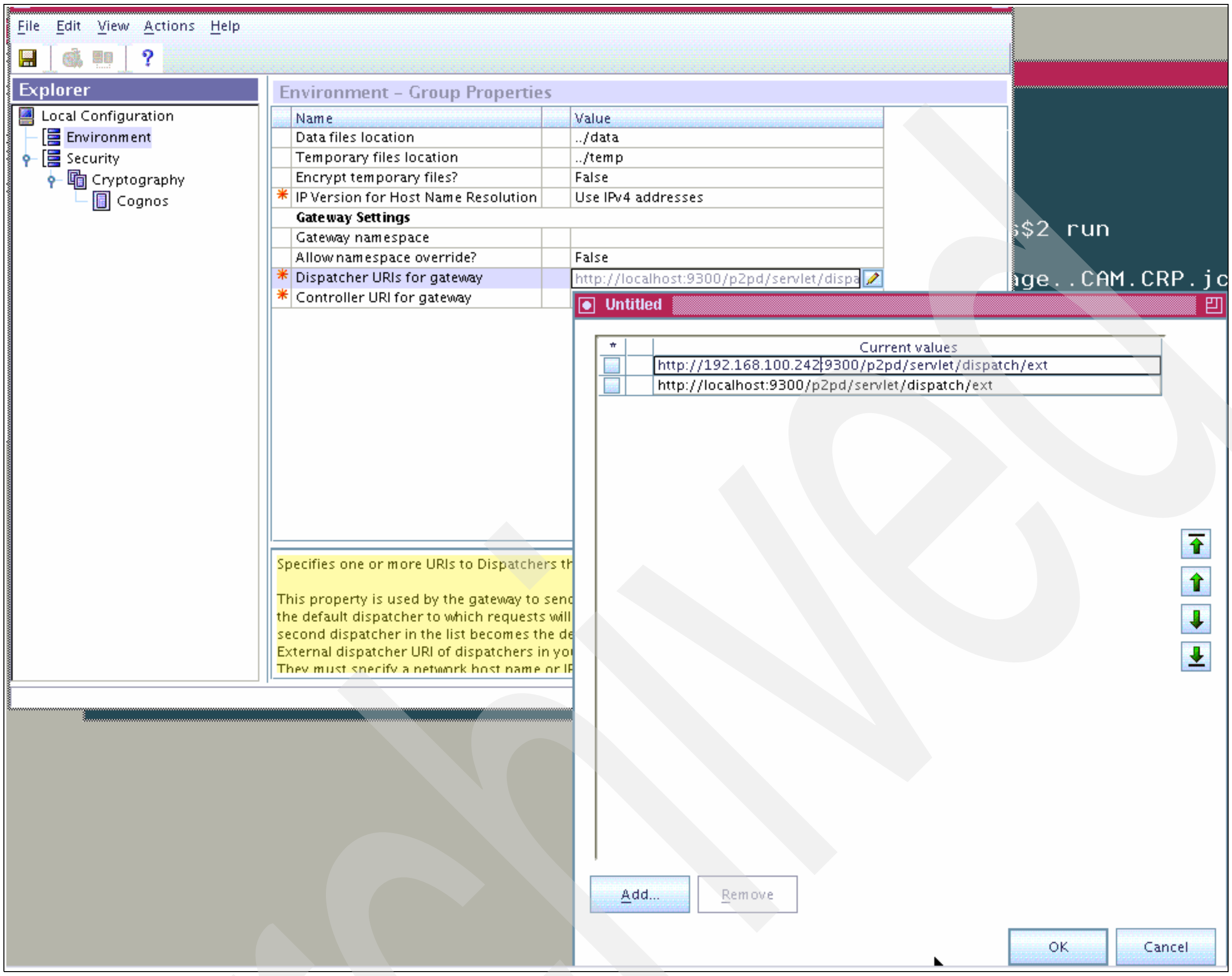


Figure 4-23 Configuring the Cognos gateway with the dispatcher URL

Also we need to set the same password that we set for the Content Manager security certificate (Figure 4-24).

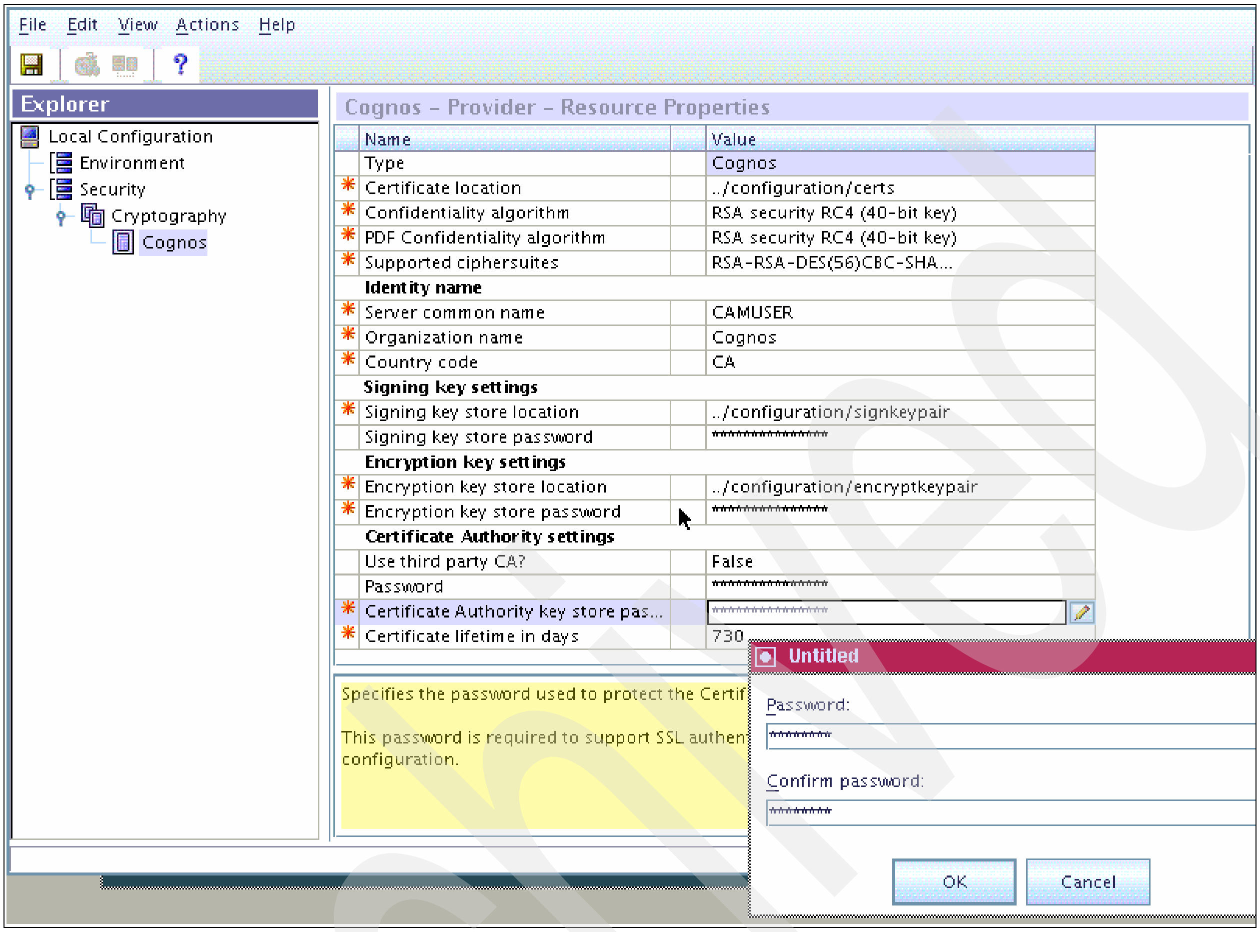


Figure 4-24 Setting the security certificate password in the gateway server

Save all the changes before the HTTP server is started using the **apachectl start** command. With this, the deployment of Cognos is completed.

4.3 Distributed deployment of Cognos server

Under a typical deployment of Cognos, all the functionalities of Cognos reporting are part of single deployment. A single deployment of Cognos needs to perform several tasks, such as request handle, communication with the database, data parsing, report processing, and so on. Resource availability is a major issue for all these services to produce good throughput. However, multiple instances of Cognos can be implemented with a specific service activated. With a distributed deployment, we can achieve high availability and redundancy. Each of these components is installed under separate AIX instances. There can be several combinations based on redundancy of separate services but primarily is divided into three major components:

- Content Manager
- Report service
- Dispatcher with other services

Since the report service is the most active service in Cognos, we have two instances for the report service. Overall, there are four Cognos servers. Apart from these, we have one LPAR with a gateway service and another with a DB2 server.

Figure 4-25 shows the functional diagram for this distributed deployment.

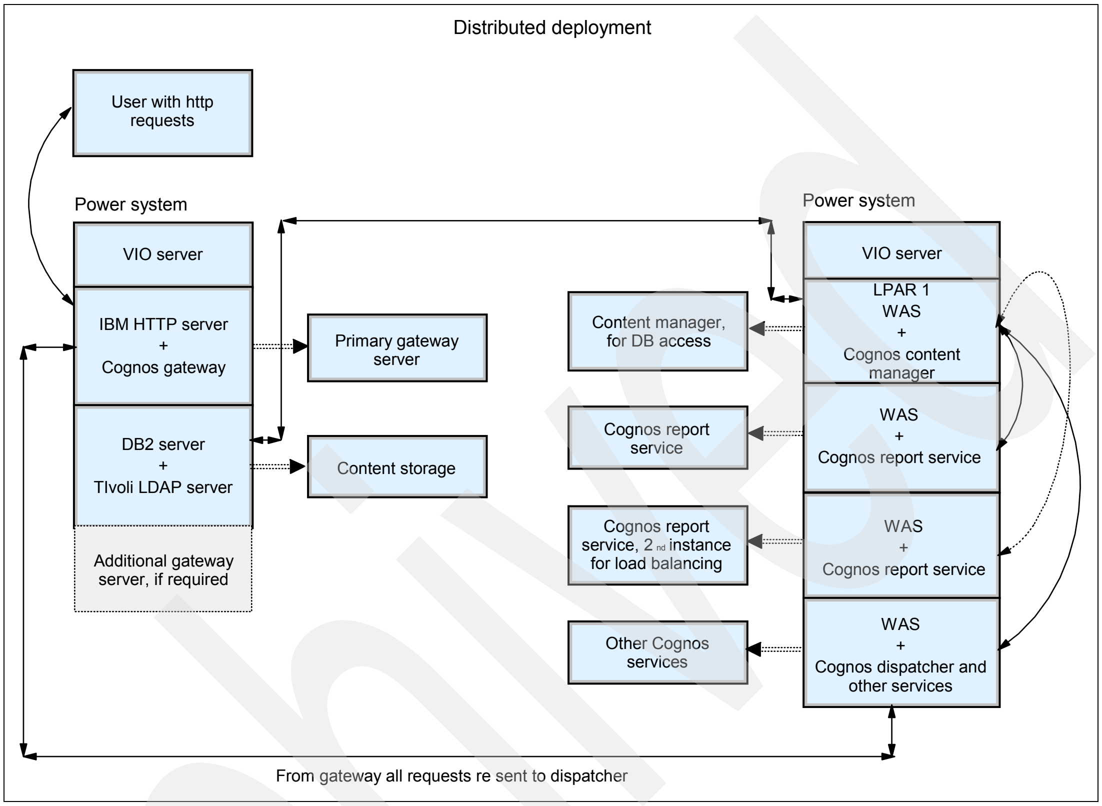


Figure 4-25 Functional diagram for a distributed deployment of the Cognos services

4.3.1 IBM Power Systems servers

We used two POWER6 systems with eight CPUs each. Both systems are connected to local storage and external storage. Also, they are in the same network. We used two systems so that each LPAR can be configured with the desired amount of system resources. Figure 4-26 shows the topology.

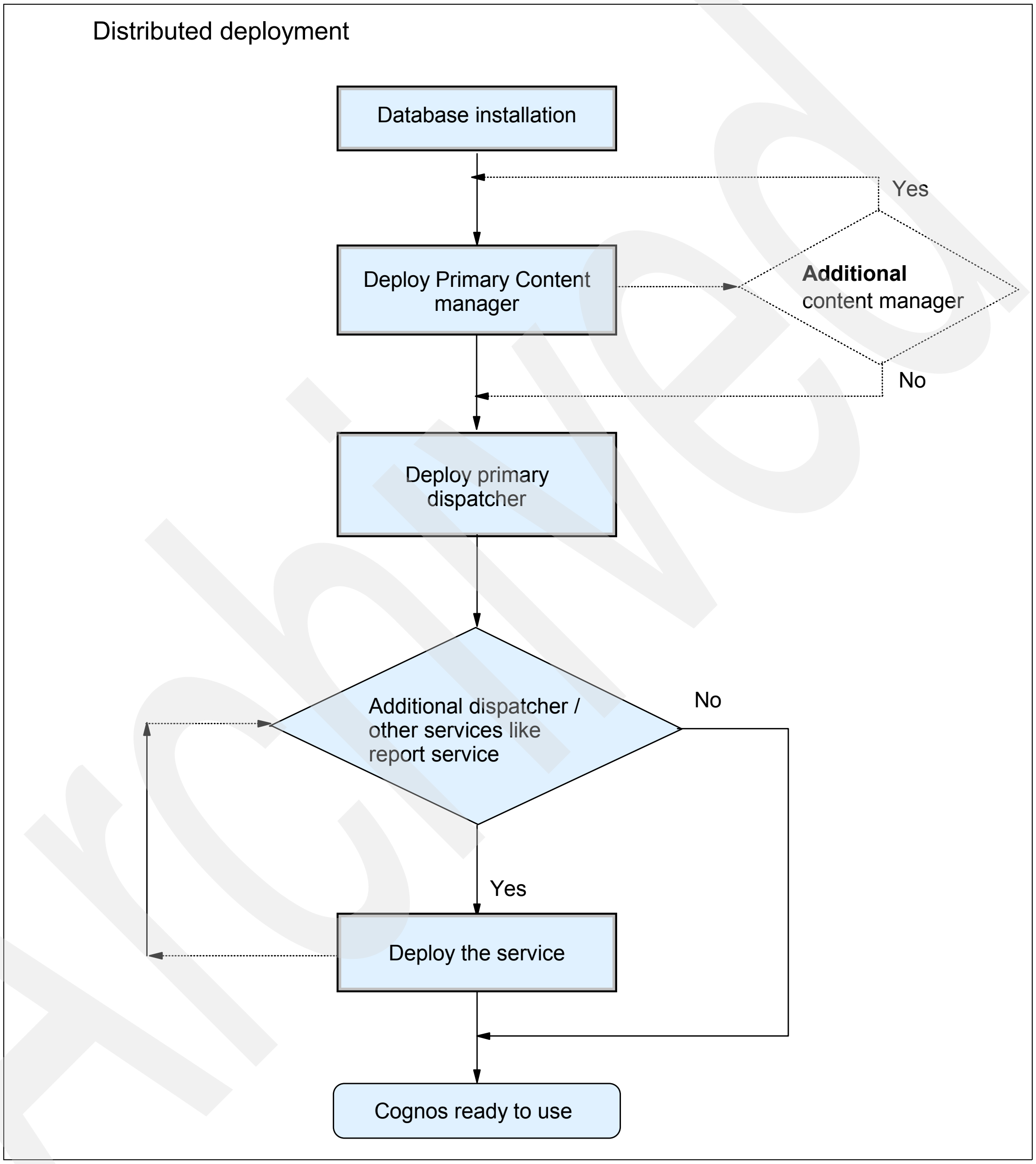


Figure 4-26 Topology for a distributed Cognos deployment

4.3.2 Cognos deployment

As shown in Figure 4-25 on page 75, the application tier is distributed under four separate LPARs. Each Cognos service is deployed with a WebSphere Application Server. We use the same gateway and database server from the reference installation. The following components are installed during the Cognos deployment:

- Content Manager

  Only one Content Manager component is selected during the Cognos installation (Figure 4-27). This deploys a Content Manager service. This component is responsible for communication with the content storage. Other Cognos services such as the dispatcher and the report service connect to this Content Manager to retrieve data from the content store.

  The Content Manager is configured with the server IP, JDBC port number, database instances ID, and content store database name (Figure 4-12 on page 63). Check database connectivity before all the information is saved (Figure 4-13 on page 64).

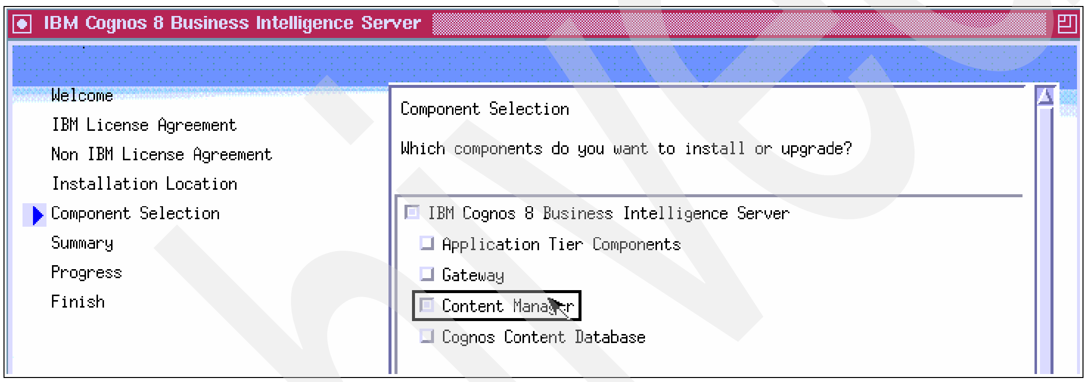


Figure 4-27 Installation of the Content Manager only

- Report server LPARs

  Configure Cognos installations such that only the report service is enabled so that requests for other services are routed to the *other services* LPAR.

- Dispatcher for other services

  From the gateway server, requests are routed to the dispatcher, where they are processed and sent to the service requested. Within the dispatcher, there are a few more services, such as a presentation service, parsing service, and so forth, configured in the same deployment. Unlike the dispatcher, these services are not that actively used. The report service is to be disabled in this install.

> **Note:** In practice, the effectiveness of adding additional dispatcher and report service LPARs is limited by current scalability limitations with the Content Manager service itself.

4.4 Active Memory Sharing and Cognos deployment

Active Memory Sharing (AMS) is part of the IBM PowerVM Enterprise Edition. It is available on IBM POWER6 processor systems with updated firmware (340_75) and later. IBM AIX 6.1 TL3 operating system supports Active Memory Sharing. A group of partitions is configured with to increase memory utilization with the ability to move, or flow memory from one partition to another. A key benefit of AMS is the capability to configure systems with less physical memory. The configuration depends greatly on the workloads and performance requirements. For example, in dedicated memory, if a workload requires more memory at peak performance, then physical memory must be allocated for the duration of the application's execution. In Active Memory Sharing with the same workload, memory can be saved because there is no need to allocate the maximum physical memory.

During peak workloads, Cognos Content Manager and report server might need an additional amount of memory. Also, there can be multiple report servers, the environment can be configured with Active Memory Sharing, and, depending on the requirement of each logical partition, they can pull in the required amount of memory to meet their workload requirements.

In our environment, Active Memory Sharing is a feature of the logical partition that has the Cognos deployment. The Cognos deployment and configuration process remains the same as mentioned in earlier sections. The configuration for Active Memory Sharing is provided in details in Chapter 6, "Experiences with Live Partition Mobility and Active Memory Expansion with Cognos 8 BI components" on page 139.

4.5 Live Partition Mobility and Cognos deployment

Live Partition Mobility, a key virtualization feature in IBM PowerVM, first became available for IBM Power Systems with POWER6 processors. Live Partition Mobility allows AIX or Linux logical partitions to be migrated from one system to another. Applications running on one LPAR can be migrated during scheduled system maintenance. Live Partition Mobility can also be used to aid server migrations and server consolidations and to balance workload across a set of servers. The application does not need to stop during migration, which ensures service availability at all times.

Cognos deployment and configuration exploit this feature. Thus, Cognos continues to provide services even during migration and peak workloads.

Hardware and logical partitioning configuration with live partition mobility details are given in Chapter 6, "Experiences with Live Partition Mobility and Active Memory Expansion with Cognos 8 BI components" on page 139.

4.5.1 Live Partition Mobility with reference implementation

Under our implementation, we have three servers:

- The gateway server
- The Cognos server with an application tier
- The DB2 server

Only the Cognos server is migrated. During this migration, the Cognos server runs a very high workload, which in turn creates several HTTP and database sessions. After migration, the Cognos server reaches the same throughput level as mentioned in previous sections.

4.5.2 Live Partition Mobility with a distributed Cognos deployment

Under our distributed deployment, there are separate LPARs assigned to the dispatcher, Content Manager, and report server, apart from the gateway and the database servers. Using Live Partition Mobility (LPAR), LPARs hosting the Content Manager and the report server can be migrated to a separate machine. Both the Cognos servers can be migrated one by one or simultaneously. Cognos server performance remains the same even after migration, including all live sessions and transactions being restored properly at the end of the migration.

4.6 Cognos deployment with workload partition

Workload partition (WPAR) provides a virtualized operating environment to manage separate workloads within a logical partition. Each WPAR is a secured and isolated environment for the hosted application. LPARs provide more flexibility in managing applications and workloads. Based on a WPAR profile, they can be divided in two types:

- System WPARS
- Application WPARs

A system WPAR has its own AIX operating system instance, can have a dedicated writable file system, and can share the global environment's `/usr` and `/opt` directories in read-only mode unless a private copy is specified. System WPAR provides a complete networking capability, making it possible to remotely log in to a system WPAR.

An application WPAR is reserved for an application. It uses the global file system and resources. After the application processes are completed, the WPAR is stopped. All the file systems in an application WPAR are shared with the global environment.

We used a system WPAR to deploy Cognos components. The Cognos server in each WPAR is configured with separate services. Except for minor but important differences, deployment and configuration of separate server components are mostly similar in a WPAR environment.

4.6.1 WebSphere Application Server and HTTP server deployment

The installer used for the WebSphere Application Server or the HTTP server checks for disk space availability as a pre-installation validation. Under the WPAR environment, disk space is shared across the global environment and the workload partition environment. Thus, disk space validation should be skipped.

For the deployment, instead of using the **setup** command, use the install script with additional parameters (Example 4-10).

Example 4-10 WebSphere Application Server and HTTP server installation using the install script

```
./install -W checklateprereqs.active=False -W
lateprereqsfailedpanelInstallWizardBean.active=False -W
checklateprereqs.prereqsPassed=True -W
calculatediskspaceInstallWizardBean.active=False
```

4.6.2 Advantages of using WPARs

Advantages of using WPARs are:

- Better management of system resources

  At times users need to have separate AIX operating system (OS) images to run several software services. In a traditional setup using LPARs, multiple OS images can be installed. However, each OS environment cannot consume all the resources allocated. Therefore, at a certain point there is the need to buy additional hardware to extend the services, though there are still resources underutilized across various partitions that cannot be consolidated.

  In a WPAR environment, we create a large partition. Then multiple system WPARs are configured to provide separate OS images for several services. Also, each WPAR can be configured with a given amount of processing power, memory, and with its own writable file system. To extend the service, a new WPAR is created using the remaining resources from the LPAR.

- Flexibility in process and application monitoring

  With a LPAR environment, all the processes run under the same OS, which determines the priority and scheduling for each process, so it is difficult to ensure a certain amount of processing power and memory to a particular process.

  However, this can be achieved using WPARs, which also help to monitor the resource utilization by various processes.

- High availability of software service through live application mobility

  Live application mobility is a feature with which to migrate a WPAR with active processes to a separate LPAR. This is very useful for extending service availability during scheduled maintenance. This also helps to achieve better load balancing.

- Reduced system administration effort

  System administration efforts can be reduced significantly through WPARs. Most often multiple LPARs are running with the same OS level. To upgrade to the next technology level, the same operation needs to be performed to all the partitions. However, with WPARs, this process needs to be done to a single instance (the global instance), and all other WPARs get it by default.

  Also, WPARs can be created and removed easily compared to LPAR. For additional information about WPARs and the WPAR manager, see the IBM Redbooks publication *Workload Partition Management in IBM AIX Version 6.1*, SG24-7656.

4.6.3 How to create and manage a system WPAR

A new WPAR is created using the **mkwpar** or **smitty** command with the appropriate parameters. A WPAR name, IP address, file system type, and location are several of the important parameters required to create a system WPAR. The parameters can be provided either via the command line or through smitty. Also, the information can be saved in a specification file, and the same file can be referred during WPAR creation. Example 4-11 shows a sample specification file.

Example 4-11 System WPAR specification file

```
general:
        name = "wpar_cm"
        hostname = "cmtest"
        directory = "/wpars/wpar_cm"
        privateusr = "yes"
```

```
        vg = "rootvg"
        devices = "/etc/wpars/devexports"
        auto = "yes"
        preserve = "no"
        copy_nameres = "yes"

mount:
        dev = "/wpar_nfs/wpar_cm"
        mountopts = "bg,intr"
        vfs = "nfs"
        directory = "/"
        host = "192.168.100.199"

mount:
        dev = "/wpar_nfs/wpar_cm/home"
        mountopts = "bg,intr"
        vfs = "nfs"
        directory = "/home"
        host = "192.168.100.199"

mount:
        dev = "/wpar_nfs/wpar_cm/var"
        mountopts = "bg,intr"
        vfs = "nfs"
        directory = "/var"
        host = "192.168.100.199"
mount:
        dev = "/wpar_nfs/wpar_cm/tmp"
        mountopts = "bg,intr"
        vfs = "nfs"
        directory = "/tmp"
        host = "192.168.100.199"

mount:
        dev = "/wpar_nfs/wpar_cm/usr"
        vfs = "nfs"
        directory = "/usr"
        mountopts = "rw"
        host = "192.168.100.199"

mount:
        dev = "/wpar_nfs/wpar_cm/opt"
        vfs = "nfs"
        mountopts = "rw"
        directory = "/opt"
        host = "192.168.100.199"

mount:
        dev = "/proc"
        directory = "/proc"
        vfs = "namefs"
        mountopts = "rw"

security:
        secfile = "/etc/wpars/secattrs"
```

```
network:
        broadcast = "192.168.100.255"
        interface = "en0"
        netmask = "255.255.255.0"
        address = "192.168.100.224"
```

4.6.4 Live application mobility with WPARs

This is a feature to migrate a WPAR from one LPAR to another with active processes. Live application mobility is a very useful feature for addressing several scenarios such as scheduled system outage, maintenance, and load balancing.

Certain special configurations are required to support live application mobility, as discussed in this section

NFS server for source and destination for LPARs

A NFS Server V3 or V4 needs to be configured. However, it is a good idea to configure a separate volume group on a separate server. The following sections describe the NFS server configuration instructions.

Modifying the /etc/exports file and exporting all the directories

Through the NFS server, a directory is shared across several client systems. The rules are configured in the /etc/exports file. Each line in this file contains a directory name, which is shared across several OS images. Thus, all systems can access this directory, read-write access, and so on. Example 4-12 provides a typical configuration. The **exports** command exports all the directories in the `/etc/exports` file.

Example 4-12 Example of /etc/exports configuration

```
/PW9408/nfs -public,sec=sys,rw,root=*
/wpar_nfs -public,sec=sys,rw,root=*
```

Starting the NFS server

The NFS server is started from smitty or using the **infest** command.

Adding the client information in the /etc/hosts file or in the DNS server

The NFS server must be able to resolve all the NFS clients by name. The client name and IP addresses must be available in the DNS server. If a DNS server is not available, then all the client information is added in the `/etc/hosts` file.

Mounting the exported file system on the clients

The **mount** command imports the file system on a client partition.

WPAR configuration for live application mobility

The prerequisites to enable a WPAR for live application mobility are:

- The WPAR is created on the NFS drive. However, the base directory must point to the `/wpars` directory (Example 4-13).
- WPAR must be *checkpointable* enable.

Example 4-13 WPAR configuration for live application mobility

```
general:
   name = "wpar_app"
   checkpointable = "yes"
   hostname = "apptest"
   privateusr = "yes"
   directory = "/wpars/wpar_app"

mount:
   dev = "/wpar_nfs/wpar_app/var"
   directory = "/var"
   vfs = "nfs"
```

- The LPAR has the WPAR manager client or MCR file set installed.

Live application mobility

Live application mobility can be performed using the WPAR manager interface. Otherwise, the **movewpar** command is executed to perform the migration. Migration starts from the source LPAR where the **movewpar** command is executed with the WPAR name (Example 4-14). A key is generated and is passed to the **movewpar** command on the destination LPAR along with the WPAR name, and the source LPAR IP or host name.

Example 4-14 movewpar command to perform a live application mobility

```
Movwpar command to be executed on source LPAR
movewpar -s -o /tmp/migratoin_server -l debug wpar_app

Movwpar command to be executed on destination LPAR
movewpar -b -a -k 4bcd85ae00008278 -o /tmp/migration_log -l debug wpar_app
192.168.100.199
```

During migration a checkpoint is taken, and all running processes are suspended until the end of the migration. The duration of the checkpoint is very small, and that is held at the last phase of the migration. During the checkpoint certain transactions might experience a time-out error. This risk can be mitigated if the application and server in the WPAR are tuned with higher time-out values and retry of timed-out transaction. At the end of the migration, the same migrating WPAR's profile is deleted from the source LPAR, recreated on the destination LPAR, and the WPAR is resumed with all the processes.

Figure 4-28 shows the output of the **movewpar** command on the source LPAR.

```
p5701db2(root)/> movewpar -s -o /tmp/migratoin_server -l debug wpar_app
Connection key:
4bcd85ae00008278
mcr: Migration server started successfully for wpar wpar_app
p5701db2(root)/>
```

Figure 4-28 movewpar output on source LPAR

The key is highlighted in the output. This key is passed to the **movewpar** command on the destination LPAR (Figure 4-29).

```
cognos(root)/> movewpar -b -a -k 4bcd85ae00008278 -o /tmp/migration_log -l debug wpar_app 192.168.100.198
mkwpar: 0960-002 /mnt_wpar/wpar_app already exists.
mkwpar: Creating file systems...
 /
 /home
 /opt
 /proc
 /tmp
 /usr
 /var
Workload partition wpar_app created successfully.
mkwpar: 0960-390 To start the workload partition, execute the following as root: startwpar [-v] wpar_app
Starting workload partition wpar_app.
Mounting all workload partition file systems.
Loading workload partition.
Module MCRK successfully loaded
Exporting workload partition devices.
Starting workload partition subsystem cor_wpar_app.
0513-059 The cor_wpar_app Subsystem has been started. Subsystem PID is 368832.
mcr: wpar wpar_app migrated successfully
cognos(root)/>
```

Figure 4-29 movewpar output from destination LPAR

The **lswpar** is the command to monitor different phases of migration. For a running WPAR, the state is A for active. Once the **movewpar** command is executed, its state turns into T for transitional. The M for moving is the state when the checkpoint-restart operation is in progress. Example 4-15 shows all these states.

Example 4-15 Separate states for WPAR during live application mobility

```
p5701db2(root)/> lswpar
Name      State  Type  Hostname  Directory        RootVG WPAR
-----------------------------------------------------------------
httptest  A      S     httptest  /wpars/httptest  no

p5701db2(root)/> lswpar
Name      State  Type  Hostname  Directory        RootVG WPAR
-----------------------------------------------------------------
httptest  T      S     httptest  /wpars/httptest  no

p5701db2(root)/> lswpar
Name      State  Type  Hostname  Directory        RootVG WPAR
-----------------------------------------------------------------
httptest  M      S     httptest  /wpars/httptest  no
```

This chapter provided the procedures to implement the test environment for the IBM Cognos8 BI solution in an IBM Power Systems server environment. Chapter 5, “Exploiting IBM Power Systems PowerVM virtualization features with IBM Cognos 8 BI” on page 87, describes in more detail how to leverage the IBM Power System PowerVM virtualization features.

5

Exploiting IBM Power Systems PowerVM virtualization features with IBM Cognos 8 BI

This chapter provides information about how to exploit the IBM Power Systems PowerVM virtualization features with the IBM Cognos solution. This chapter covers the following topics:

5.1 Overview

Power Systems servers such as POWER6 and POWER7 provide a single virtualization solution with multiple capabilities for efficiently utilizing system and hardware resources. All these virtualization capabilities together are called PowerVM. In this section, we describe how these advanced features can be used for improving performance of various Cognos BI services and components. Virtualization environments, architectures, and operations, each of which can help Cognos running certain workloads, are discussed in detail. For example:

- Micro-partitioning
  - CPU
  - Memory shared pools
  - Capacity on demand
- Tunables at different levels
  - Power Systems specific
  - AIX
  - Cognos
- Active Memory Sharing (AMS) with multiple LPARs
- Virtual I/O Server (VIOS)
- Virtual Ethernet
- Workload Partitions (WPAR)
- Live Application Mobility and Live Partition Mobility

5.2 Testing methodology

The test objective is to determine the optimal configuration for IBM Cognos 8 Business Intelligence (BI) while taking advantage of the unique PowerVM virtualization features.

Testing scenarios reflect the typical Business Intelligence user workloads, which include a set of tests that represent most of the use cases for Business Intelligence applications. The used cases run simultaneously with concurrent simulated-user sessions to fully stress all applications and environmental server components.

5.3 User expectation versus performance factors

The scenarios used in this chapter are based on various virtualization technologies. Each scenario provides information regarding the virtualization capability and about the way it affects particular Cognos components. If any pain points are recognized, we expect to solve them with the recommended methods. Performance factors are looked at from the following perspectives:

- Delay factor

  Although a user might think he is using the web browser alone, he is actually competing with other users at any point in time. The number of users sending requests simultaneously to Cognos affects the response time that a user sees. A longer delay is a sign of poor performance in general.

- Transaction factor

  The number of transactions or queries is a variation of the number of competing users. Only in this case, requests are repetitive and multiple events like traffic flows indicate how fast the Cognos dispatchers are forwarding the request messages and the responding data.

- Capacity factor

  The amount of data sent at one time is also a significant factor in how a request can be handled. This often is related to memory and network performance, in addition to the crunching power of the CPUs. Cognos might show signs of hanging or crashing.

Understanding the performance factors helps to pinpoint the hidden problems. When someone complains that his BI website is slow, it is possible that one of the factors mentioned above is at fault. Then one of the following scenarios might come in handy. Keep in mind that every performance situation is different. It usually takes more than one try or method to improve the conditions or fix the problems. To help in observing the changes and effects in different situations, we set up a test environment for these experiments.

5.3.1 Test environment setup

The goals for setting up the test environment include:

- Simulating user access
- Generating workload

Using this test environment, we are able to observe the responses of the Cognos components and the utilization of system resources. The test results and analyses provide feedback data and information indicating the need to change test workload or system resources.

Test systems

After looking into the test software available, we decided on the software tool that is recommended by the Cognos development team. The test systems are independent of the Cognos and other servers. Five System x® machines are used as the test controller and generators. Figure 5-1 shows the test environment.

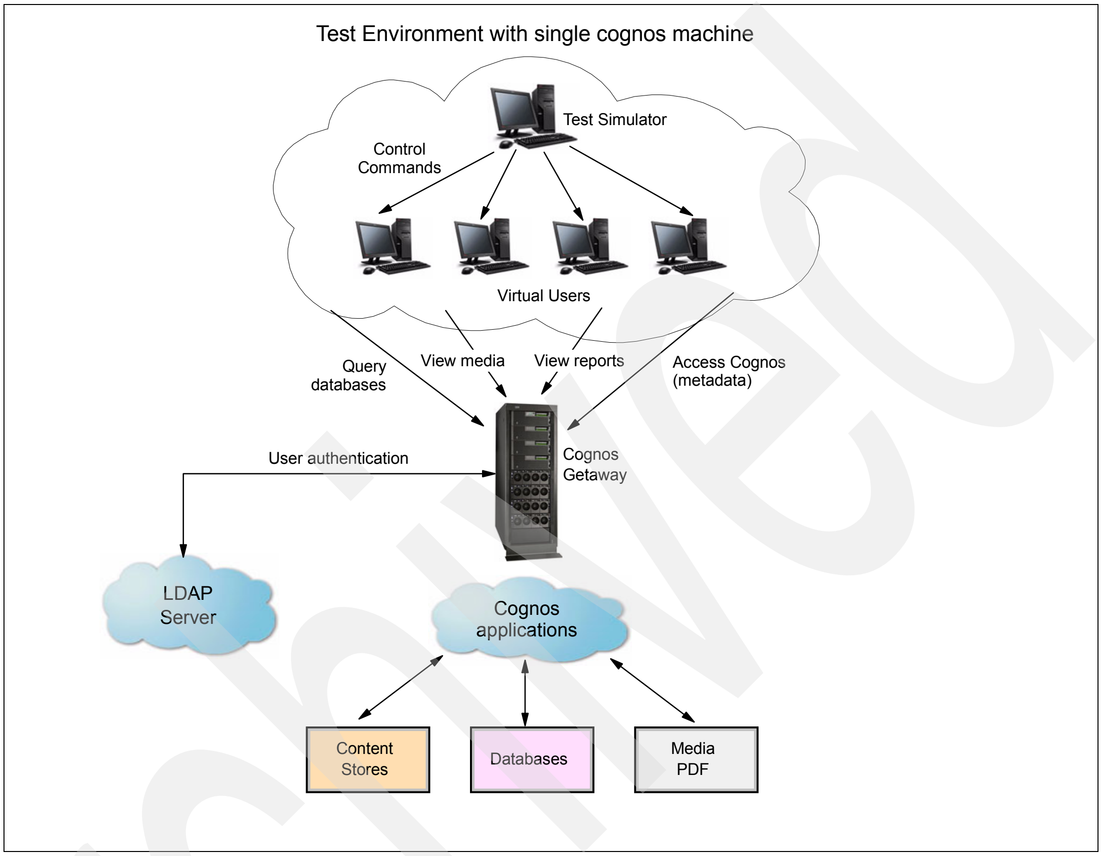


Figure 5-1 Test environment

Regardless of what changes in the Cognos and other server configurations, the test systems are kept unchanged throughout all the tests. We are interested in the following test attributes:

- Number of users accessing Cognos at a time
- Types of transactions and requests sent to Cognos

This test environment is non-intrusive and provides realistic workload stressing Cognos as a whole application or separate distributed components.

During the test run, the test tools displayed live statistics when the queries and transactions were going on. The following analytic characteristics are of interest to us:

- Maximum running users
- Average hits per second
- Response time from certain transactions

Simulating users

The test controller can be set up to generate a fixed number of virtual users. At test time, the virtual users run the scripts provided. A script in turn goes through the logon and authentication processes just like a real user does.

When generating users, the tool applies ramp-up and ramp-down times. This is especially for the test run that creates thousands of users. In reality, not all users start accessing their Cognos website at the same time. Ramping up is for staggering user queries at the beginning of the test run. We particularly pay attention to this because it has a significant effect on Cognos responses. The interval between starting two users can be set in seconds. Certain test runs generate two users every six seconds. Most of the test runs generate one user every second. Similar ramp-down intervals are applied at the end of the run.

To simulate real users, the tool also allows a setting for thinking time. The value zero means no thinking time, which is set to all of the users in our tests. For each test, a fixed number of users is set. Then, based on the test result, the number of users is increased or decreased in the next test run.

The capability of using multiple generators provides another means of controlling test loads. The total workload is divided across all generators according to designated percentages.

Generating workload

Testing with the correct type of workload is important. We use a collection of scripts that make up an overall workload mimicking a typical business intelligent workload. As depicted in Figure 5-2, the scripts are grouped into four categories with various distributions from each group of users.

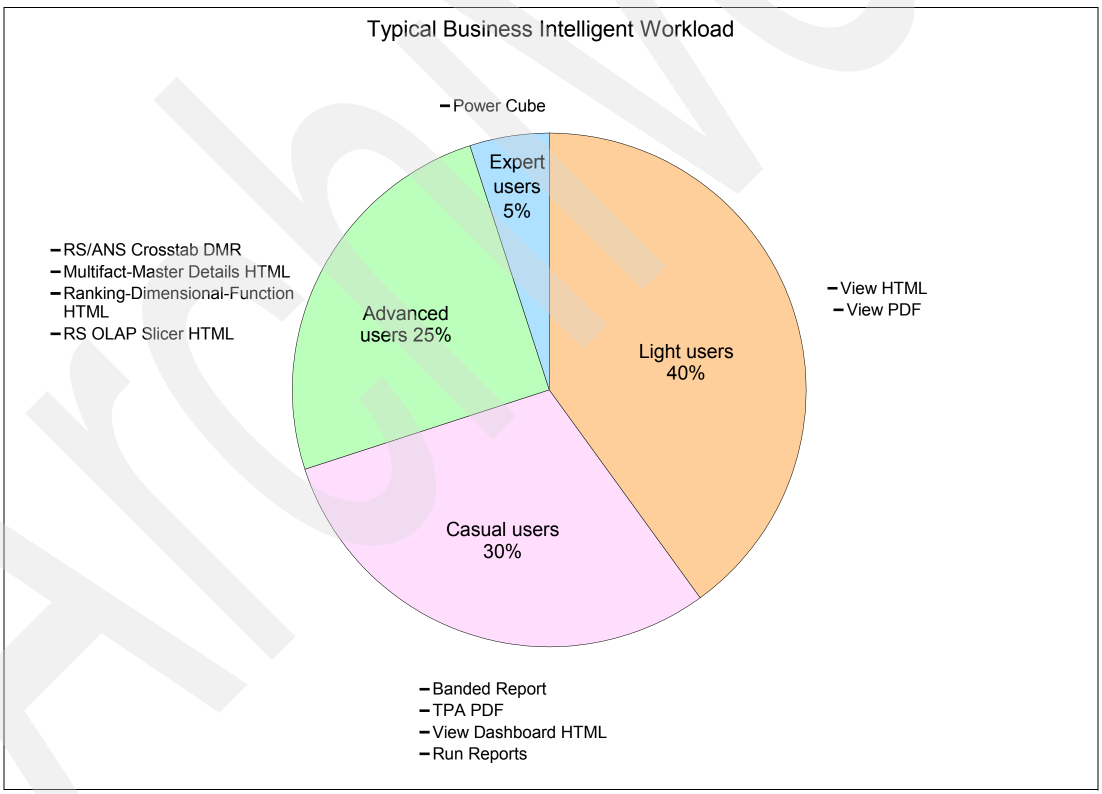


Figure 5-2 Typical Business Intelligent workload

The categories of users based on their typical workload are:

- Light users: make up the majority of the workload by viewing HTML and PDF files from the Cognos websites. There is no database query involved.
- Casual users: view the Cognos Dashboard and run reports, which might also generate PDF files.

- Advanced users: need more processing power for data analysis and viewing details of the data models.
- Expert users: only represent a small population of the users. However, the PowerCube models require the most complex database transactions and data manipulations.

In a typical business intelligent set of scenarios, the scripts send a mix of transaction requests to Cognos. Figure 5-3 shows the list of transactions that can be monitored.

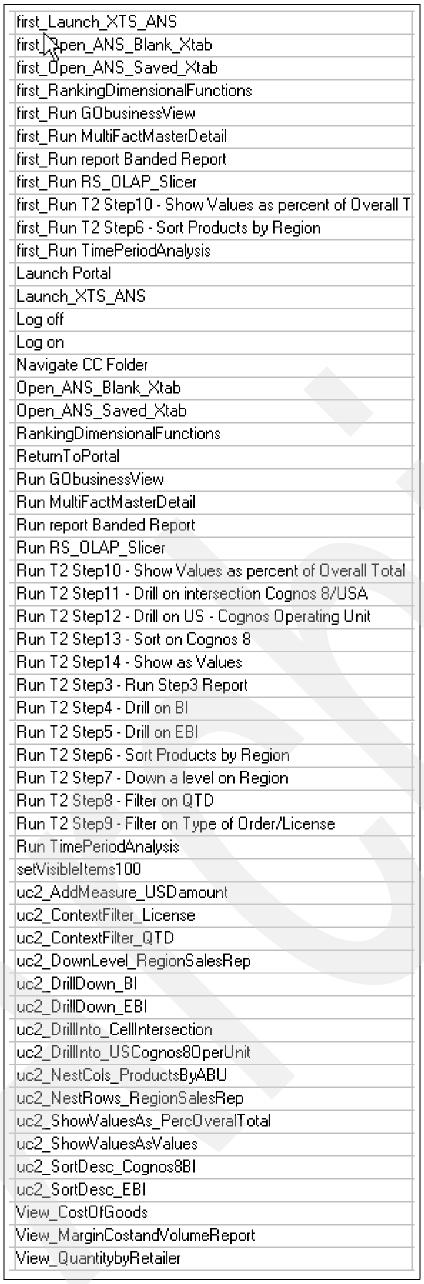


Figure 5-3 List of transactions

The test tool also allows specifying workload characteristics such as:

- Browser caching: We set browser caching to clear the cache for every transaction so that the query data is fetched from the Cognos databases every time.
- Run time iterations: We set runtime iterations to 20 iterations to extend a script's run time.
- Running a user as a process or a thread: We set running user to thread. For the same available amount of system resources, running threads is more efficient than processes.

> **Note:** The above are just the key settings. The test tool allows many other settings that we do not list here.

5.3.2 Test observation

This section discusses the test observation characteristics of interest to our case study.

Maximum running users

After starting a test run, we watch for the increment of the number of virtual users during ramp-up time. Since we defined either one user every second or every three seconds, the number of users increases steadily at that rate. If the increment stops or many failures of creating users occurs in the middle of the ramp-up time, the test has encountered problems. On the other hand, the systems could possibly have reached maximum users limit.

Hits per second

This is the number of hits that the virtual users put on the web server in a second. Since the browser caching is cleared every time, the hits per second show the workload on the Cognos servers associated with user queries. A test workload with good performance shows that hits per second increase along with the user ramp-up time. In contrast, a test workload that starts with high hits per second then decreases with time is bad.

Average and peak response times

For each transaction, we want to see the average response time as low as possible. If a peak response time is abnormally high, then there is a bottleneck somewhere. Since every transaction has a different response time value, we base our observation on user expectation. For example, if a transaction takes more than two seconds, then it is not acceptable to the user.

5.3.3 Quick fix with PowerVM features

When we observe one of the factors described in 5.3.2, "Test observation" on page 93, we are able to bring the stalled performance up to the next level by using one or more of the Power Systems PowerVM virtualization features.

The most intriguing and very quick change is to reconfigure and add one or more CPUs to the LPAR that needs more processing power. In the case of all Cognos components running on a single LPAR, increasing one CPU doubles the maximum number of users. Average hits per second also increases as a result. Redistributing CPUs from an available share pool is made possible by using logical partitioning and micro partitioning features of PowerVM. Existing processors are redeployed as opposed to buying and installing new hardware.

Another system resource that can be redistributed from a share pool is physical memory. By changing the partition profile, memory can be moved to the Cognos application server and dispatcher where the BI bus processes require most of the memory. We noticed that at least

8 GB is required. Furthermore, to have five BI bus processes available for optimal performance, we added 2 GB for each BI bus processes. A feature in POWER6 and POWER7 is to enable dynamic memory sharing called Active Memory Sharing. See 4.4, “Active Memory Sharing and Cognos deployment” on page 77.

The Cognos distributed architecture also works well with PowerVM. LPARs can be added, then one or more Cognos components can be moved to the new LPAR. The POWER6 and POWER7 feature called Live Application Migration (LAM) allows moving an application running in a WPAR to a more powerful partition with a slight delay, but no disruption.

Another PowerVM feature is the capability of moving an LPAR between POWER6 and POWER7 systems. We have been able to migrate the Cognos partition from a POWER6 system to a POWER7 system while running the test workload. Test result analysis showed only a quick drop of CPU usage, then back to normal or higher processing in the case of migrating to a more powerful system.

Moving Cognos partitions or workloads from POWER7 to POWER6 for maintenance reasons is also possible. We observed similar positive experiences, which highlights the flexibility in configurations and operations.

Besides the workload test tool, we also use the AIX performance monitoring tool **nmon**. Data generated from **nmon** is saved in a file for later analysis. This is especially good for the Live Application Mobility experiments.

5.3.4 Power Systems virtualization with AMS

This section introduces Active Memory Sharing technology and shows how it can be used in Power Systems based environments.

5.3.5 Introduction

In addition to the traditional way in which logical partitions are individually assigned dedicated amounts of memory, Active Memory Sharing allows the creation of a memory pool that can be shared among multiple logical partitions.

The physical memory of an IBM Power System can be assigned to logical partitions either in a dedicated mode or a shared mode. A single partition can have either dedicated or shared memory.

At the system level, the administrator has the capability to assign physical memory to a logical partition and physical memory to a memory pool that is shared by other logical partitions.

The decision is based on several considerations that include performance expectations, global resource usage, optimization, and cost. Typically, a single system is configured with both dedicated and shared resources.

5.3.6 Dedicated versus shared memory

The following sections describe the differences between dedicated and shared memory.

Memory management in dedicated mode

Dedicated memory partitions have system physical memory reserved based on their configuration. Memory size is a multiple of the system *logical memory block size*. The memory is allocated to the logical partition as soon as the partition is activated.

The system administrator optimizes the available memory distribution among multiple logical partitions. If a logical partition performance is adversely impacted by memory constraints while other logical partitions have unused memory, the administrator can manually react by issuing a dynamic memory reconfiguration. Reconfiguration is then subject to free memory availability and administrator reaction time.

Memory management in shared mode

Shared memory partitions are configured with a *logical memory space* that is a multiple of the system logical memory block size, but physical memory is allocated by the Power Hypervisor from the shared memory pool based on the logical partition runtime memory requirements. Memory allocation is made with a very fine granularity that depends on the hypervisor page size, which is currently 4 KB. When using shared memory mode, it is the system that automatically decides the optimal distribution of physical memory among logical partitions and adjusts the memory assignment based on demand for memory pages. The administrator just reserves physical memory for the shared memory pool and assigns logical partitions to the pool. The administrator also configures the amount of memory in the LPAR's configuration, but how much memory is actually being used depends on the memory activity in the LPARs. Active Memory Sharing can be exploited to increase memory utilization on the system either by decreasing the system memory requirement or by allowing the creation of additional logical partitions on an existing system.

The benefits of Active Memory Sharing are that it:

- Increases the utilization rate of existing memory
- Balances workloads and memory resources between partitions running various applications
- Consolidates your environment by accommodating more partitions on consolidated systems
- Changes configurations in a very simple and secure manner with limited administrator intervention
- Optimizes the usage of global system resources and adjusts the infrastructure to accommodate new systems
- Reduces costs

5.3.7 Shared Memory Pool

This section describes shared memory pools in more detail.

How it works

Active Memory Sharing allows for over-commitment of memory resources. Since partition logical memory is mapped to system physical memory depending on logical partition memory demand, the sum of all logical partition logical memory can exceed the shared memory pool size.

Each logical partition is allowed to use all assigned logical memory. When the cumulative usage of physical memory reaches the pool size, the hypervisor can transparently steal memory from a shared memory partition and assign it to another shared memory partition. If the removed memory page contains data, it is stored on a paging device and the memory page content is cleared before it is assigned to another logical partition. If the newly assigned page contained data, it is restored from the disk device.

Since paging disk activity has a cost in terms of logical memory access time, the hypervisor keeps track of memory usage to steal memory that is likely not be used in the near future. The shared memory partition's operating system cooperates with the hypervisor by providing hints about page usage and by freeing memory pages to limit hypervisor paging.

When the hypervisor decides that paging activity has to be performed, it sends a request to the paging Virtual I/O server to copy a specific memory page belonging to a specific logical partition to or from the corresponding paging device. The paging Virtual I/O server performs the action and then notifies completion to the hypervisor. Multiple paging requests might be issued at the same time.

Definition and deletion

A system enabled for the Active Memory Sharing feature can be configured with a single shared memory pool. The shared memory pool can be created using either the Hardware Management Console (HMC) or the Integrated Virtualization Manager (IVM).

The memory in the pool is reserved upon creation and it is used exclusively by the shared memory partitions. Dedicated memory partitions cannot access the shared memory pool, even if no shared memory partitions are defined or activated.

The size of the shared memory pool can be dynamically changed at any time by using the HMC or IVM interfaces. The pool can grow up to the maximum system memory available.

If you have inactive shared memory partitions, the shared memory pool cannot be deleted. However, you can either remove the LPARs that are configured with shared memory or you have to re-activate them using dedicated memory before deleting the shared memory pool.

If no shared memory pool is available on the system, it is not possible to define any new shared memory partitions. Once the pool is available, it is designed to support up to 128 shared memory partitions.

5.3.8 Paging devices

The shared memory pool configuration requires the definition of a set of *paging devices* that are used to store excess memory pages on temporary storage devices. Access to the paging devices associated with a shared memory partition is provided by a paging Virtual I/O server on the same system. At the time of pool creation, the paging Virtual I/O server that provides the paging service to the pool must be identified.

Each shared memory partition requires a dedicated paging device in order to be started. Paging device selection is made when a shared memory partition is activated, based on the availability and the size of the maximum logical memory of the logical partition. If no suitable paging device is available, activation fails with an error message providing the required size of the paging device.

Paging devices can be dynamically added or removed from the shared memory pool configuration. Device deletion is allowed only if it is not assigned to any running logical partition. If the logical partition is activated after device removal, a new paging device is selected from the available set.

A separate paging device is required for each active shared memory partition, and it can be any one of the following:

- A logical volume
- Locally attached storage
- SAN-attached storage
- iSCSI-attached storage

5.3.9 Virtual I/O server

When the hypervisor needs to free memory pages in the shared memory pool, the content of the memory must be stored on a paging device to be restored later when the data is accessed again. This activity is referred to as *paging activity* and is performed by the Virtual I/O server defined for paging in the shared memory pool's configuration.

The paging Virtual I/O server is designed to handle up to 128 shared memory partitions. The paging Virtual I/O server must be configured with dedicated memory since it is providing services to the pool itself. On HMC-managed systems, the user can assign up to two paging VIOS partitions to a shared memory pool to provide access to the paging devices.

5.3.10 Partition requirements

To be defined as a shared memory partition, the logical partition must meet the following requirements:

- Use shared processors.
- Use virtual I/O, including:
  - Virtual Ethernet adapters
  - Virtual SCSI adapters
  - Virtual Fibre Channel adapters
  - Virtual serial adapters
- The minimum level of the AIX operating system running in the logical partition is Version 6.1 TL 03.

Attention: Shared memory partitions do not allow dedicated adapters.

5.3.11 Logical and physical memory

The memory section of a logical partition's profile has been enhanced to allow selection of either a dedicated memory or a shared memory. The memory assigned to the logical partition is then defined as either dedicated memory or logical memory.

On a shared memory partition, two parameters define the memory configuration:

Logical memory Quantity of memory that the operating system manages and can access. Logical memory pages that are in use might be backed up by either physical memory or a pool's paging device.

Memory weight Relative number used by the hypervisor to prioritize the physical memory assignment from the shared memory pool to the logical partition. A higher value increases the probability that more physical memory is assigned to the logical partition.

As with dedicated memory partitions, shared memory partitions have minimum, desired, and maximum memory configuration values. In the case of a shared memory partition, these values control the logical memory size of a partition. If a logical partition's memory requirement changes with time, it is possible to dynamically modify the size of the logical memory assigned, provided that the minimum and maximum logical memory limits are satisfied.

Logical memory content placement is under the control of the hypervisor, which decides whether it must be stored in physical memory or in a paging space device.

5.4 Usage examples

The goal of memory sharing is to optimize the usage of the memory pool by assigning the physical memory to the logical partitions that need it most at a specific point in time. This optimization can be used either to reduce global memory requirements of logical partitions or to allow logical partitions to increase their memory footprint during peak memory demand periods.

Multiple memory-sharing scenarios are possible, depending on the overcommitment type of physical memory. Depending on your workload, they might provide significant advantages to overall performance.

5.4.1 Logical memory overcommitment

In a logical overcommitment scenario, the memory sizing of the shared memory partitions is made taking into account memory demands throughout an interval, such as a day, and making sure that the global requirement of physical memory never exceeds the physical memory in the pool.

In this configuration, it is possible to optimize existing physical memory on the system or to reduce the global memory needs.

Existing memory optimization

Consider a set of logical partitions that have been already sized and are using dedicated memory. The logical partitions might be changed into shared logical partitions in a memory pool that contains the same amount of memory that the previously dedicated partitions occupied. The logical memory assigned to each shared memory partition is configured to be larger than the size in dedicated mode.

This new configuration does not change the global memory requirements, and every logical partition can have the same amount of physical memory that it had previously. However, memory allocation is highly improved, because unexpected memory demands due to unplanned peak demands of one logical partition can be satisfied by the pool by assigning extra pages automatically. This differs from a dedicated memory configuration, in which the new memory must be statically assigned to only a few selected logical partitions.

Reduced memory needs

Good knowledge of the physical memory requirements of the partitions over time allows you to configure systems with a reduced memory configuration. For example, two logical partitions are known to require 8 GB each at peak time, but their concurrent requirement never exceeds 10 GB. The shared memory pool can be defined with 10 GB available and each logical partition is configured with 10 GB of logical memory. On a dedicated memory configuration, 16 GB of memory is required instead of the 10 GB required with the shared memory setup.

The memory footprint required to host a set of shared memory partitions might be greatly reduced when shared memory partitions do not actively use all their memory at the same time. The hypervisor monitors memory demand on each shared memory partition and is able to meet the memory demand.

Logical overcommitment might be a good opportunity for workloads that have the following characteristics:

- Overlap peaks and valleys in memory usage (for example, night and day activities or applications accessed by users in different time zones).
- Low average memory residency requirements.
- Do not have sustained loads such as retail headquarters and university environments.
- Failover and backup logical partitions that are used for redundancy, which require resources only when the primary server goes down. Resources do not have to be dedicated to a redundant server.

5.4.2 Physical memory overcommit

Physical overcommitment occurs when the sum of all logical memory that is being referenced at a point in time exceeds the physical memory in the shared memory pool. The hypervisor must then make frequent use of the paging devices to back up the active memory pages.

In this scenario, memory access times vary depending on whether logical pages are available in physical memory or on a paging device. The rate of hypervisor page faults determines application performance and throughput, but all logical partitions are allowed to work.

Not all workloads might be affected by memory latency, and overcommitment allows the creation of a larger number of logical partitions than with a dedicated memory configuration.

Example configurations where physical overcommit might be appropriate are:

- Workloads that use a lot of file cache. The operating system can control cache size according to hypervisor requirements in order to limit hypervisor paging.
- Workloads that are less sensitive to memory latency, such as file servers, print servers, and Content Manager network applications.
- Logical partitions that are inactive most of the time.

5.4.3 Practical scenario

We used the Active Memory Sharing feature of the IBM Power System PowerVM in the following environment:

- An IBM Power System server based on the POWER7 processors
- Firmware level 01ZL710_042
- HMC Version 7 Release 7.1.0 with MH01214
- Virtual I/O Server Version 2.1.0.1-FP22
- AIX 6.1 TL 4

We created a Cognos business environment with separate system workload partitions dedicated to various Cognos components as follows:

- Partition vios_p7_2 for a VIO server
- Partition cm for Cognos Content Manager
- Partition dispatcher for Cognos dispatcher
- Partition gw1 for Cognos gateway server 1
- Partition gw2 for Cognos gateway server 2
- Partition <rep n> for Cognos report server 1
- Partition <rep n> for Cognos report server 2

All partitions have been configured to use dedicated memory. All disks and communication interfaces are accessible via a single VIO server. We change partition profiles to use a shared memory pool later.

The name, ID, maximum amount of memory, and memory access mode of all partitions are displayed from the HMC using the **lshwres** command (Example 5-1). Note that we use dedicated memory for the VIO server.

Example 5-1 Displaying partition name, ID, maximum memory, and memory access mode

```
hscroot@hmcdq1:~> lshwres -r mem --level lpar -m p7_2 -F lpar_name lpar_id
curr_max_mem mem_mode --header
lpar_name lpar_id curr_max_mem mem_mode
rep2 7 4096 ded
rep1 6 4096 ded
gw2 5 4096 ded
gw1 4 4096 ded
cm 3 4096 ded
dispatcher 2 4096 ded
vios_p7_2 1 32768 ded
```

To configure the partitions to use shared memory pools we take the following steps:

1. We check the amount of physical memory available for logical partitions on the system and the number of memory pools that can be defined (Example 5-2). The command is issued from the HMC and shows that there are 219,392 MB of physical memory available. Only one memory pool can be defined. Also note that the amount of memory available for logical partitions is lower than the total amount of memory installed on the system. The difference is used by the hypervisor.

   Example 5-2 Checking the memory available and the maximum number of memory pools

   ```
   hscroot@hmcdq1:~> lshwres -m p7_2 -r mem --level sys -F configurable_sys_mem
   curr_avail_sys_mem max_mem_pools --header
   configurable_sys_mem curr_avail_sys_mem max_mem_pools
   262144 219392 1
   ```

2. We verify whether the system is capable of using the Active Memory Sharing feature using the command in Example 5-3.

   Example 5-3 Verifying that the system is Active Memory Sharing capable

   ```
   hscroot@hmcdq1:~> lssyscfg -r sys -m p7_2 -F active_mem_sharing_capable
   1
   ```

   A returned value of 1 means that the system is AMS-capable.

3. We verify whether there is a shared memory pool already defined using the command in Example 5-4.

   Example 5-4 Verifying whether there is a shared memory pool defined

   ```
   hscroot@hmcdq1:~> lshwres -m p7_2 -r mempool
   No results were found.
   ```

   There is no shared memory pool defined on the system.

4. We create a 16 GB shared memory pool from the HMC (Example 5-5). The pool is assigned to a VIO server named vios_p7_2, the same VIO server that serves the logical partitions. We then list the attributes of the shared memory pool. Also note that the amount of memory that can be shared among the partitions is less than the size of the entire memory pool. The difference of 256 MB of memory is used by the hypervisor for memory pool management.

Example 5-5   Creating and verifying the shared memory pool

```
hscroot@hmcdq1:~> chhwres -r mempool -m p7_2 -o a -a
"pool_mem=16384,max_pool_mem=16384","paging_vios_names=vios_p7_2"
hscroot@hmcdq1:~> lshwres -m p7_2 -r mempool
curr_pool_mem=16384,curr_avail_pool_mem=16128,curr_max_pool_mem=16384,pend_pool
_mem=16384,pend_avail_pool_mem=16128,pend_max_pool_mem=16384,sys_firmware_pool_
mem=256,paging_vios_names=vios_p7_2,paging_vios_ids=1
```

5. We modify partition profiles to used the newly created shared memory pool. With respect to memory, all client partitions have identical configurations. In Figure 5-4 we show what the HMC graphical interface looks like when configuring shared memory for the rep1 partition.

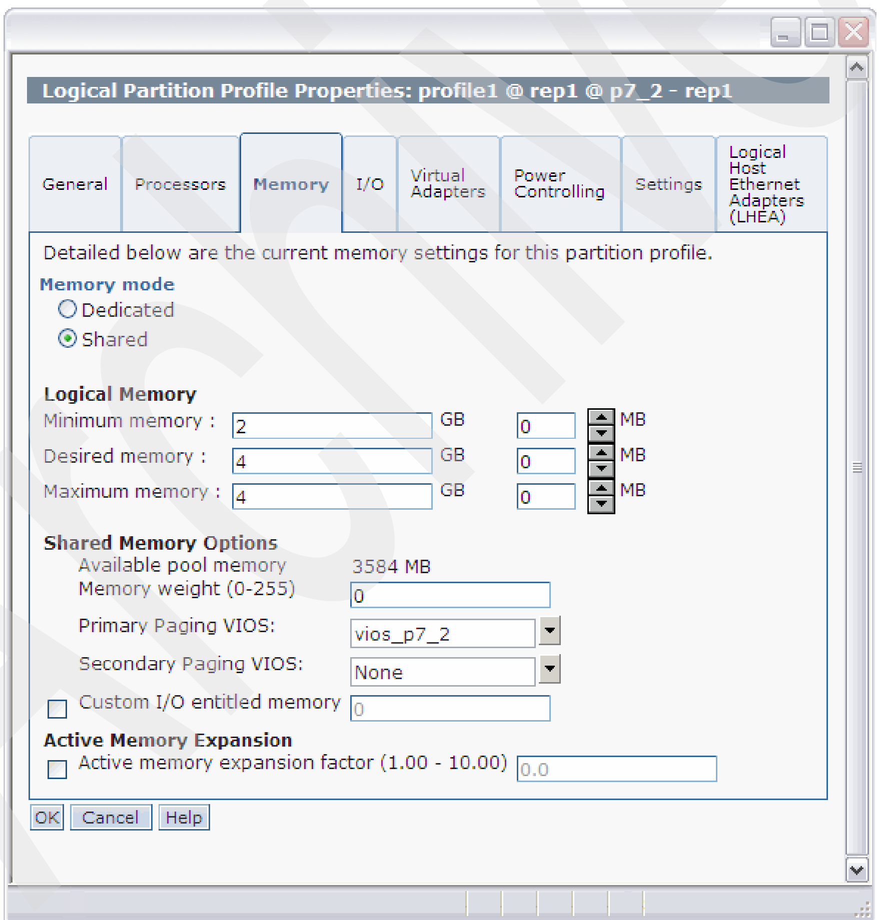


Figure 5-4   Configuring a partition to use shared memory pool

6. We add the hdisk35 disk as the first paging device assigned to the shared memory pool (Example 5-6). We then list the attributes of the paging device, the name, the size, and the physical location of the disk. Also note that the device status is inactive and has not been assigned yet to any partition.

Example 5-6 Adding the first paging device to the shared memory pool

```
hscroot@hmcdq1:~> chhwres -r mempool -m p7_2 --rsubtype pgdev -o a -p vios_p7_2
--device hdisk35
hscroot@hmcdq1:~> lshwres -m p7_2 -r mempool --rsubtype pgdev
device_name=hdisk35,paging_vios_name=vios_p7_2,paging_vios_id=1,size=30720,typ
e=phys,state=Inactive,phys_loc=U78A0.001.DNWHFY4-P1-C5-T1-W201300A0B811A6
62-L1B000000000000,is_redundant=0,lpar_id=none
```

7. The definition of the paging device is added to the VIO server. Paging device attributes and status can also be verified from the VIO server (Example 5-7). Note that at this time the paging device has not been assigned to any partition yet and its status is inactive.

Example 5-7 Listing paging devices and their attributes from the VIO server

```
$ lsdev -dev vrmpage*
name             status      description
vrmpage0         Defined     Paging Device - Disk
$ lsdev -dev vrmpage0 -attr
attribute       value              description                   user_settable

LogicalUnitAddr 0x8100000000000000 Logical Unit Address          False
aix_tdev        hdisk35            Target Device Name            False
client_reserve  no                 Client Reserve                True
partition_id    0                  Client Partition ID           False
redundant_usage no                 Redundant Usage               True
vasi_drc_name                      VASI DRC Name                 True
vrm_state       inactive           Virtual Real Memory State     True
vtd_handle      0x200001275aaad5d3 Virtual Target Device Handle False
```

8. To illustrate how AMS functions, we gradually reduce the amount of the shared memory pool to a smaller size as the memory is physically overcommitted. We used the commands shown in Example 5-8. Note that 256 MB of memory is still being used by the hypervisor for memory pool management. Also note that we modified only the *current* shared memory pool size and kept the maximum memory pool size of 16 GB. This might be useful in case we decide to decrease back the size of the pool.

Example 5-8 Gradually reducing the size of the shared memory pool

```
hscroot@hmcdq1:~>chhwres -r mempool -m p7_2 -o s -a "pool_mem-=256"
hscroot@hmcdq1:~>chhwres -r mempool -m p7_2 -o s -a "pool_mem-=8192"
hscroot@hmcdq1:~> chhwres -r mempool -m p7_2 -o s -a "pool_mem-=4096"
hscroot@hmcdq1:~> lshwres -m p7_2 -r mempool
curr_pool_mem=3840,curr_avail_pool_mem=3584,curr_max_pool_mem=16384,pend_pool_m
em=3840,pend_avail_pool_mem=3584,pend_max_pool_mem=16384,sys_firmware_pool_mem=
256,paging_vios_names=vios_p7_2,paging_vios_ids=1
```

9. We start the partition named rep1, which has partition ID 6. Notice that the paging device hdisk35 is now assigned partition ID 6, and its status changed to active (Example 5-9).

Example 5-9 Paging device status changes to available

```
$ lsdev -dev vrmpage0 -attr
attribute       value                          description
user_settable

LogicalUnitAddr 0x8100000000000000             Logical Unit Address          False
aix_tdev        hdisk35                        Target Device Name            False
client_reserve  no                             Client Reserve                True
partition_id    6                              Client Partition ID           False
redundant_usage no                             Redundant Usage               True
vasi_drc_name   U8233.E8B.100291P-V1-C32776    VASI DRC Name                 True
vrm_state       active                         Virtual Real Memory State     True
vtd_handle      0x200001275aaad5d3             Virtual Target Device Handle False
```

10. We now define the paging devices for the remaining partitions (Example 5-10). Note that when trying to assign a paging device that is already in use, you get an error. Remember that we started only one partition and note that only hdisk35 is active and assigned to the rep1 logical partition, which has ID 6. All other paging devices are inactive and unassigned.

Example 5-10 Defining paging devices for the remaining partitions

```
hscroot@hmcdq1:~> chhwres -r mempool -m p7_2 --rsubtype pgdev -o a -p vios_p7_2 --device hdisk34
hscroot@hmcdq1:~> chhwres -r mempool -m p7_2 --rsubtype pgdev -o a -p vios_p7_2 --device hdisk33
hscroot@hmcdq1:~> chhwres -r mempool -m p7_2 --rsubtype pgdev -o a -p vios_p7_2 --device hdisk33
HSCLA48A The add paging space device operation failed:

HSCLA46E The paging space device hdisk33 on Virtual I/O Server partition vios_p7_2 is already in the
shared memory pool.

hscroot@hmcdq1:~> chhwres -r mempool -m p7_2 --rsubtype pgdev -o a -p vios_p7_2 --device hdisk32
hscroot@hmcdq1:~> chhwres -r mempool -m p7_2 --rsubtype pgdev -o a -p vios_p7_2 --device hdisk31
hscroot@hmcdq1:~> chhwres -r mempool -m p7_2 --rsubtype pgdev -o a -p vios_p7_2 --device hdisk30
hscroot@hmcdq1:~> lshwres -m p7_2 -r mempool --rsubtype pgdev
device_name=hdisk35,paging_vios_name=vios_p7_2,paging_vios_id=1,size=30720,type=phys,state=Active
,phys_loc=U78A0.001.DNWHFY4-P1-C5-T1-W201300A0B811A662-L1B000000000000,is_redundant=0,lpar_name
=rep1,lpar_id=6
device_name=hdisk34,paging_vios_name=vios_p7_2,paging_vios_id=1,size=30720,type=phys,state=Inact
ive,phys_loc=U78A0.001.DNWHFY4-P1-C5-T1-W201300A0B811A662-L1A000000000000,is_redundant=0,lpar_id
=none
device_name=hdisk33,paging_vios_name=vios_p7_2,paging_vios_id=1,size=30720,type=phys,state=Inact
ive,phys_loc=U78A0.001.DNWHFY4-P1-C5-T1-W201300A0B811A662-L19000000000000,is_redundant=0,lpar_id
=none
device_name=hdisk32,paging_vios_name=vios_p7_2,paging_vios_id=1,size=30720,type=phys,state=Inact
ive,phys_loc=U78A0.001.DNWHFY4-P1-C5-T1-W201300A0B811A662-L18000000000000,is_redundant=0,lpar_id
=none
device_name=hdisk31,paging_vios_name=vios_p7_2,paging_vios_id=1,size=30720,type=phys,state=Inact
ive,phys_loc=U78A0.001.DNWHFY4-P1-C5-T1-W201300A0B811A662-L17000000000000,is_redundant=0,lpar_id
=none
device_name=hdisk30,paging_vios_name=vios_p7_2,paging_vios_id=1,size=30720,type=phys,state=Inact
ive,phys_loc=U78A0.001.DNWHFY4-P1-C5-T1-W201300A0B811A662-L16000000000000,is_redundant=0,lpar_id
=none
```

11. We start all remaining partitions and verify the paging device new attributes and status from the VIO server (Example 5-11). At this time all paging devices are available and, for example, hdisk30 is assigned and being used by a partition with ID 7.

Example 5-11 Verifying paging device status from the VIO server

```
$ lsdev -dev vrmpage*
name             status      description
vrmpage0         Available   Paging Device - Disk
vrmpage1         Available   Paging Device - Disk
vrmpage2         Available   Paging Device - Disk
vrmpage3         Available   Paging Device - Disk
vrmpage4         Available   Paging Device - Disk
vrmpage5         Available   Paging Device - Disk
$ lsdev -dev vrmpage5 -attr
attribute       value                          description
user_settable

LogicalUnitAddr 0x8600000000000000             Logical Unit Address           False
aix_tdev        hdisk30                        Target Device Name             False
client_reserve  no                             Client Reserve                 True
partition_id    7                          Client Partition ID          False
redundant_usage no                             Redundant Usage                True
vasi_drc_name   U8233.E8B.100291P-V1-C32775 VASI DRC Name                  True
vrm_state       active                       Virtual Real Memory State    True
vtd_handle      0x200001275aaad5d8             Virtual Target Device Handle False
```

12. We can also display the current shared memory pool status using the graphical interface from the HMC (Figure 5-5).

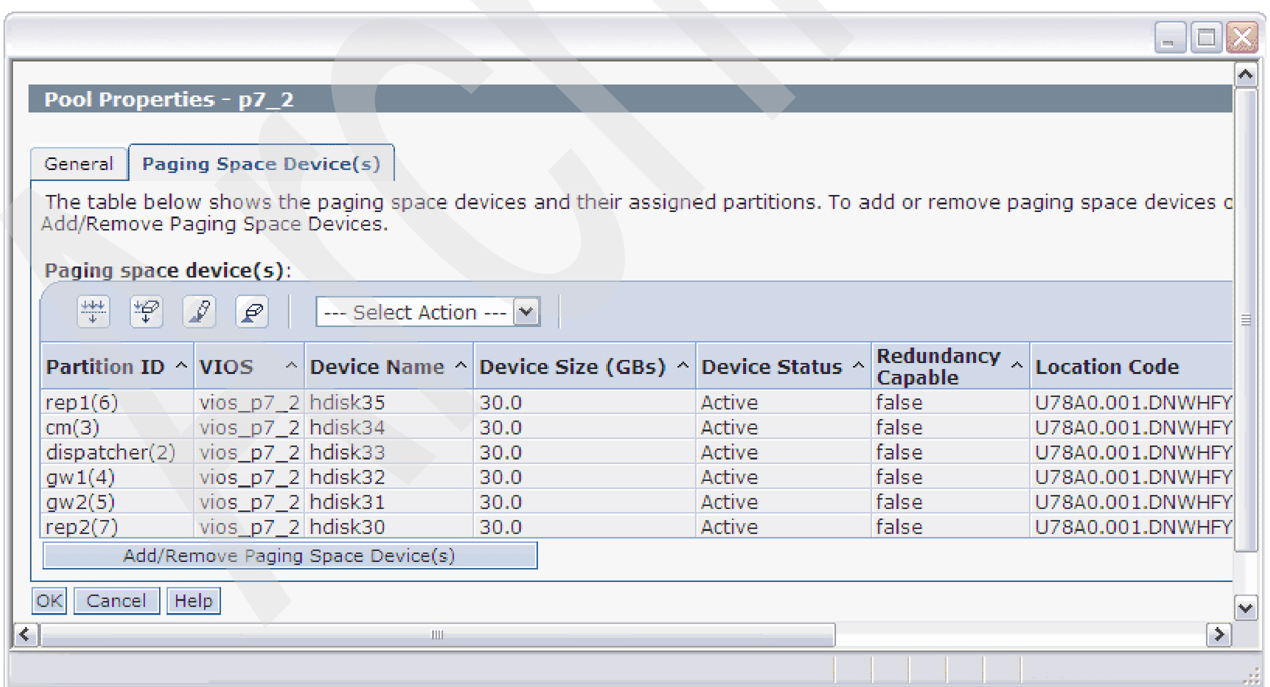


Figure 5-5 Displaying shared memory pool status using graphical interface from HMC

5.4.4 Monitoring memory performance in AMS environments

The **vmstat** command has been changed to allow monitoring memory performance in AMS environments. The following fields have been added:

mmode Shows as shared if the partition is running in shared memory mode.

mpsz Shows the size of the shared memory pool.

hpi Shows the number of hypervisor page-ins for the partition. A hypervisor page-in occurs if a page is being referenced that is not available in real memory because it has been paged out by the hypervisor previously.

hpit Shows the time spent in hypervisor paging in milliseconds for the partition.

pmem Shows the amount of physical memory backing the logical memory, in gigabytes.

loan Shows the amount of the logical memory in gigabytes that is loaned to the hypervisor. The amount of loaned memory can be influenced through the vmo ams_loan_policy tunable.

In Example 5-12, we show how to use the **vmstat** command to monitor memory performance on the rep1 partition.

Example 5-12 Using vmstat to monitor memory performance

```
rep1(root)/> vmstat -h -t 1

System configuration: lcpu=8 mem=4096MB ent=2.00 mmode=shared mpsz=16.00GB

kthr    memory              page              faults            cpu              hypv-page          time
----- ----------- ------------------------ ------------ ----------------------- ------------------------- --------
 r  b   avm   fre  re  pi  po  fr   sr  cy  in   sy  cs us sy id wa    pc    ec   hpi  hpit   pmem   loan hr mi se
 0  0 221131 520960   0   0   0   0    0   0  12    91 166  0  0 99  0  0.01   0.6     0     0   0.64   1.05 13:51:24
 0  0 221134 520957   0   0   0   0    0   0   6    22 163  0  0 99  0  0.01   0.5     0     0   0.64   1.05 13:51:25
 0  0 221134 520957   0   0   0   0    0   0   6    19 169  0  0 99  0  0.01   0.5     0     0   0.64   1.05 13:51:26
 0  0 221134 520957   0   0   0   0    0   0   5    17 158  0  0 99  0  0.01   0.5     0     0   0.64   1.05 13:51:27
 0  0 221134 520957   0   0   0   0    0   0   2    27 169  0  0 99  0  0.01   0.5     0     0   0.64   1.05 13:51:28
 0  0 221133 520958   0   0   0   0    0   0   3    87 169  0  0 99  0  0.01   0.5     0     0   0.64   1.05 13:51:29
 0  0 221133 520958   0   0   0   0    0   0   4    35 174  0  0 99  0  0.01   0.5     0     0   0.64   1.05 13:51:30
 0  0 221133 520958   0   0   0   0    0   0   3    20 186  0  0 99  0  0.02   0.9     0     0   0.64   1.05 13:51:31
 0  0 221132 520959   0   0   0   0    0   0   6    31 173  0  0 99  0  0.01   0.5     0     0   0.64   1.05 13:51:32
 0  0 221132 520959   0   0   0   0    0   0   6    39 169  0  0 99  0  0.01   0.5     0     0   0.64   1.05 13:51:33
 0  0 221132 520959   0   0   0   0    0   0  16    18 165  0  0 99  0  0.01   0.5     0     0   0.64   1.05 13:51:34
```

It is mandatory to correlate any performance measurements on logical partitions using shared memory pools with the performance of the VIO server that manages the paging devices associated with the shared memory pool,

Example 5-13 shows using the **viostat** command to monitor disk performance on hdisk35 (paging device for partition rep1).

Example 5-13 Monitoring disk performance in the VIO server

```
$  viostat -time -disk hdisk35 1
System configuration: lcpu=8 drives=38 paths=66 vdisks=29

Disks:        % tm_act     Kbps      tps    Kb_read   Kb_wrtn  time
hdisk35          0.0       0.0       0.0          0         0  18:00:31
hdisk35          0.0       8.0       2.0          8         0  18:00:32
hdisk35          0.0       0.0       0.0          0         0  18:00:33
hdisk35          0.0       0.0       0.0          0         0  18:00:34
hdisk35          0.0       0.0       0.0          0         0  18:00:35
```

```
hdisk35          0.0       0.0       0.0          0         0  18:00:36
hdisk35          0.0       0.0       0.0          0         0  18:00:37
hdisk35          0.0       0.0       0.0          0         0  18:00:38
```

To improve performance, the enhanced collection of disk statistics is usually disabled. To enabled it, you must run the following command on the VIO server:

```
chdev -dev sys0 -attr iostat=true
```

Once you have enabled enhanced statistics collection, you can get additional information regarding disk activity with the **viostat** command (Example 5-14).

Example 5-14 Displaying enhanced performance disk data on the VIO server

```
$ viostat -time -extdisk hdisk35 1
System configuration: lcpu=8 drives=38 paths=66 vdisks=29

hdisk35        xfer:  %tm_act      bps      tps      bread      bwrtn      time
                         0.0      0.0      0.0        0.0        0.0   17:57:38
               read:      rps  avgserv  minserv  maxserv   timeouts      fails
                         0.0      0.0      0.0      0.0          0          0
              write:      wps  avgserv  minserv  maxserv   timeouts      fails
                         0.0      0.0      0.0      0.0          0          0
              queue:  avgtime  mintime  maxtime  avgwqsz    avgsqsz     sqfull
                         0.0      0.0      0.0      0.0        0.0        0.0
--------------------------------------------------------------------------------
hdisk35        xfer:  %tm_act      bps      tps      bread      bwrtn      time
                         1.0     4.1K      1.0       4.1K        0.0   17:57:39
               read:      rps  avgserv  minserv  maxserv   timeouts      fails
                         1.0      4.9      4.9      4.9          0          0
              write:      wps  avgserv  minserv  maxserv   timeouts      fails
                         0.0      0.0      0.0      0.0          0          0
              queue:  avgtime  mintime  maxtime  avgwqsz    avgsqsz     sqfull
                         0.0      0.0      0.0      0.0        0.0        0.0
--------------------------------------------------------------------------------
```

5.5 Using VIO server features for fast software deployment on multiple partitions

This section shows how the VIO server features can be used to quickly install and deploy an entire Power-based infrastructure. We show how you can install the AIX operating system in a partition defined in your environment. The same method can be used for fast and consistent deployment of any software packages across multiple partitions.

This scenario is seen very often: logical partitions have a basic configuration, do not require any physical adapters, and are served by a VIO server. The system administrator has to quickly create new partitions, install the operating system, and deploy software.

5.5.1 Installing the AIX operating system on a newly defined partition

In our scenario, we use a p550 POWER6 system that already has a VIO server partition up and running. We show how to install the AIX operating system on the first logical partition defined on this system. We assume that:

- Partition name and profiles have been created.
- The partition uses only shared storage and virtual Ethernet adapters.
- The partition uses a single virtual client SCSI adapter.
- The VIO server uses a single server SCSI adapter to provide access to all storage devices required for the partition.
- One virtual Ethernet adapter has been created already for the client partition.
- One virtual Ethernet adapter has been created already for the VIO Server partition.

While additional adapters can be added later to the logical partition, the minimum requirements are fairly simple and easy to meet.

In Example 5-15 on page 109, we offer an example of quickly installing the AIX operating system on an LPAR as follows:

1. Log in on the VIO server and verify the inventory of existing disks.
2. Identify the *vhost* adapter corresponding to the logical partition. Since we have defined a single server SCSI adapter there is only one vhost adapter that corresponds to the logical partition. You can use the **`lsmap`** command to list all SCSI server adapters. In our scenario, the SCSI server adapter that corresponds to our new partition is vhost0.
3. We assign an entire shared disk as a backing device for the SCSI server adapter using the **`mkvdev`** command. In our scenario we use hdisk29.
4. The **`lsmap`** command shows the mapping between the server adapter, the target adapter, and the backing device.
5. We create a virtual optical target device using the **`mkvdev`** command. The name of the virtual optical target device is vtopt0.
6. You can use the **`lsvopt`** command to display all virtual optical target devices available and any media loaded. The device has no media loaded yet.
7. The **`lsmap`** command shows the mapping between the server adapter, the target adapter and its backing device, and the virtual optical device adapter and its backing device. vtopt0 has also been mapped to vhost0, and there is no backing device associated with vtopt0 yet.
8. We use the **`lssp`** command to verify the inventory of storage pools. There is only one storage pool, named rootvg, on the system. The rootvg volume group of the VIO server acts as a predefined storage pool. We could use another shared drive to define a separate storage pool. However, for the sake of simplicity we use the predefined storage pool. The storage pool has 285,696 MB, of which 260,608 is free.
9. We use the **`lsrep`** command to verify the existence of any media repository. So far there is no repository defined.
10. We use the **`mkrep`** command to define a 10 GB repository located in the rootvg storage pool. The repository is backed by a logical volume named VMLibrary that is part of the VIO server rootvg volume group. The logical volume contains a file system that is mounted under `/var/vio/VMLibrary`.

11. We used the **lsrep** command to show the properties of the newly created repository. We also notice that out of the initial 260,608 MB of free space, we have now only 250,368 MB free.

12. We use the **ftp** command to transfer a file with an AIX operating system image in the home directory of the VIO server padmin user. We use an image file named `aix610.iso`, as you can see from the output of the **ls** command in Example 5-15 on page 109.

13. We use the **mkvopt** command to create a virtual media named aix61 from the `aix610.iso` file.

14. The media is loaded in the repository, as you can see from the output of **lsrep** command. The name, size, and access permissions are also displayed. It is worth noticing that the amount of free space available in the repository is now 6519 MB.

15. The **ls** command shows that a file named `aix61` was created in the `/var/vio/VMLibrary` file system that backs up the repository.

16. We use the **loadopt** command to load the virtual media named aix61 in the virtual CD-ROM device named vtopt0.

17. The **lsvopt** command shows the virtual CD-ROM devices as well the virtual medias that are currently loaded.

18. We use the **lsmap** command again and notice that /var/vio/VMLibrary/aix61 is shown as a backing device for the vtopt0 virtual optical adapter.

19. We activate our client partition, boot into SMS mode, and notice that the virtual CD-ROM device is available as a boot device (Figure 5-6).

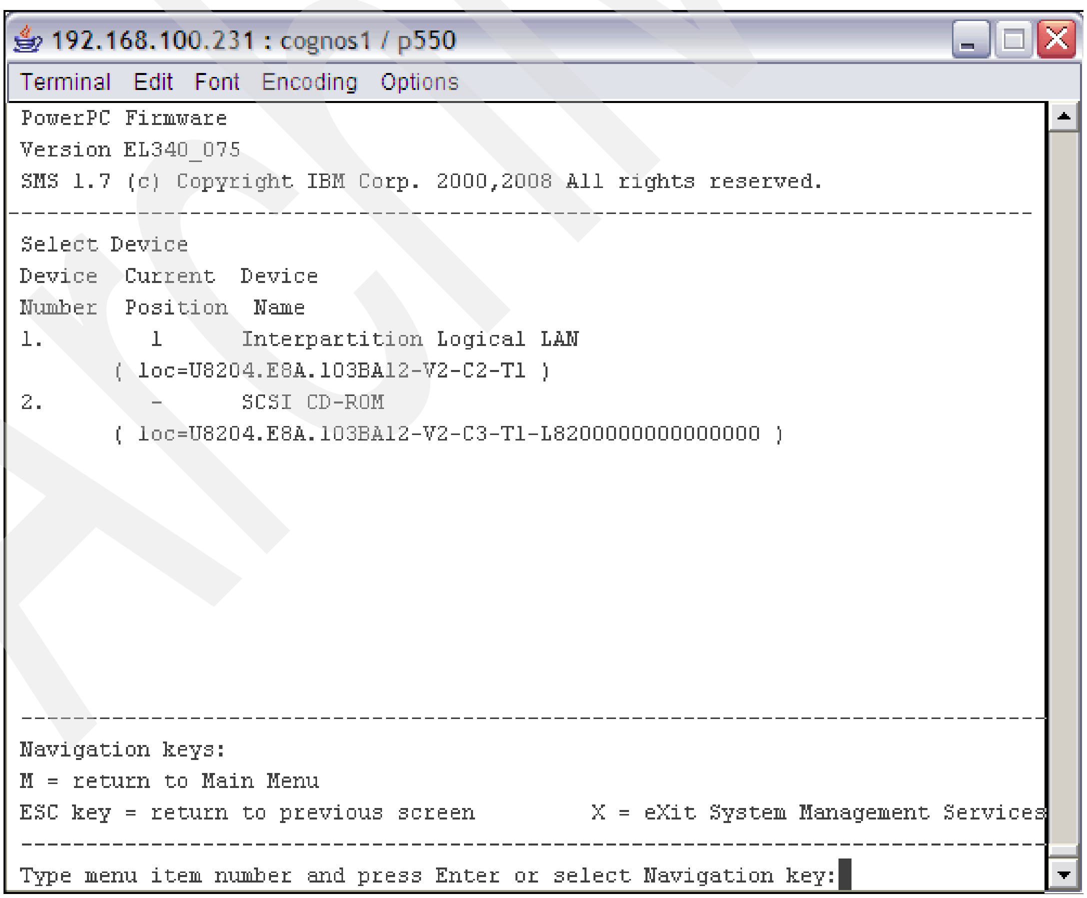


Figure 5-6 Virtual CDROM is available as a boot device for a client LPAR

20. We proceed with the installation of AIX as though we booted from physical media.

Example 5-15 Installing AIX on an LPAR using VIO server features

```
padmin@viosp550:/home/padmin>lspv
NAME            PVID                                VG              STATUS
hdisk0          0003ba121138d8fd                    rootvg          active
hdisk1          none                                None
hdisk2          none                                None
hdisk3          none                                None
hdisk4          none                                None
hdisk5          none                                None
hdisk10         none                                None
hdisk11         none                                None
hdisk12         none                                None
hdisk13         none                                None
hdisk14         none                                None
hdisk15         none                                None
hdisk16         none                                None
hdisk17         none                                None
hdisk18         none                                None
hdisk19         none                                None
hdisk20         none                                None
hdisk21         none                                None
hdisk22         00c2a9c04db6d962                    None
hdisk23         none                                None
hdisk24         none                                None
hdisk25         none                                None
hdisk26         none                                None
hdisk27         none                                None
hdisk28         none                                None
hdisk29         none                                None

padmin@viosp550:/home/padmin>lsmap -all
SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost0          U8204.E8A.103BA12-V1-C12                     0x00000000

VTD                   NO VIRTUAL TARGET DEVICE FOUND

padmin@viosp550:/home/padmin>mkvdev -vdev hdisk29 -vadapter vhost0
vtscsi0 Available

padmin@viosp550:/home/padmin>lsmap -all
SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost0          U8204.E8A.103BA12-V1-C12                     0x00000000

VTD                   vtscsi0
Status                Available
LUN                   0x8100000000000000
Backing device        hdisk29
Physloc
U78A0.001.DNWGLVC-P1-C5-T2-W201300A0B811A662-L13000000000000
```

```
$ mkvdev -fbo -vadapter vhost0
vtopt0 Available

padmin@viosp550:/home/padmin>lsvopt
VTD             Media                                   Size(mb)
vtopt0          No Media                                     n/a

padmin@viosp550:/home/padmin>lsmap -all
SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost0          U8204.E8A.103BA12-V1-C12                     0x00000000

VTD                   vtopt0
Status                Available
LUN                   0x8200000000000000
Backing device
Physloc

VTD                   vtscsi0
Status                Available
LUN                   0x8100000000000000
Backing device        hdisk29
Physloc
U78A0.001.DNWGLVC-P1-C5-T2-W201300A0B811A662-L13000000000000

padmin@viosp550:/home/padmin>lssp
Pool              Size(mb)   Free(mb)  Alloc Size(mb)    BDs Type
rootvg              285696     260608             512      0 LVPOOL

$ lsrep
The DVD repository has not been created yet.

padmin@viosp550:/home/padmin>mkrep -sp rootvg -size 10G
Virtual Media Repository Created
Repository created within "VMLibrary" logical volume

padmin@viosp550:/home/padmin>lsrep
Size(mb) Free(mb) Parent Pool         Parent Size      Parent Free
   10198    10198 rootvg                   285696           250368

padmin@viosp550:/home/padmin>ls -l
total 7534520
-rw-r-----    1 padmin   staff    3857645568 Mar 11 10:06 aix610.iso
drwxrwxr--    2 root     staff           256 Sep 19 19:09 config
-rw-r--r--    1 root     staff          1870 Mar 11 09:58 ioscli.log
-rw-r--r--    1 padmin   staff          8402 Mar 09 17:28 smit.log
-rw-r--r--    1 padmin   staff           300 Mar 09 17:28 smit.script
-rw-r--r--    1 padmin   staff           776 Mar 09 17:28 smit.transaction
drwxr-xr-x    3 padmin   staff           256 Oct 23 23:33 tivoli

padmin@viosp550:/home/padmin>mkvopt -name aix61 -file /home/padmin/aix610.iso

padmin@viosp550:/home/padmin>lsrep
```

```
Size(mb) Free(mb) Parent Pool         Parent Size      Parent Free
   10198     6519 rootvg                   285696           250368

Name                                   File Size Optical         Access
aix61                                       3679 None            rw

padmin@viosp550:/home/padmin>ls -l /var/vio/VMLibrary
total 7534464
-rw-------    1 root     staff    3857645568 Mar 11 10:09 aix61
drwxr-xr-x    2 root     system          256 Mar 11 09:52 lost+found

$ loadopt -disk aix61 -vtd vtopt0

padmin@viosp550:/home/padmin>lsvopt
VTD             Media                                   Size(mb)
vtopt0          aix61                                       3679

padmin@viosp550:/home/padmin>lsmap -all
SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost0          U8204.E8A.103BA12-V1-C12                     0x00000000

VTD                   vtopt0
Status                Available
LUN                   0x8200000000000000
Backing device        /var/vio/VMLibrary/aix61
Physloc

VTD                   vtscsi0
Status                Available
LUN                   0x8100000000000000
Backing device        hdisk29
Physloc
U78A0.001.DNWGLVC-P1-C5-T2-W201300A0B811A662-L13000000000000
```

5.5.2 Installing software packages on a newly defined partition

Following the AIX installation, the virtual CD-ROM is available to the partition and can be accessed from the AIX operating system just like a regular physical device, as shown by the **lsdev** command. The only difference is that the device is virtual (Example 5-16).

Example 5-16 Virtual CD-ROM

```
root@cognos1:/>lsdev -C|grep cd
cd0        Available       Virtual SCSI Optical Served by VIO Server
```

If you want to deploy any additional software packages, you can create additional virtual media and store it in your repository. You can unload and load them on the virtual optical device available on the partition.

5.5.3 Deploying software packages on multiple partitions

If you want to have the same media available simultaneously to multiple partitions, you can load the virtual media on multiple virtual CD-ROMs. However, you need to protect media integrity by setting the access privileges to read only.

If you already have the media loaded onto a virtual optical device (vtopt0 in our scenario) and try to load it onto an additional optical virtual device (vtopt1), you get an error similar to the one shown below:

```
$ loadopt -disk aix61 -vtd vtopt1
The specified virtual disk is not read-only, and cannot be loaded into multiple
devices simultaneously
```

To change the access mode, unload the virtual media first. If you try to change the access mode while the media is still loaded, you get an error similar to the following:

```
$ chvopt -name aix61 -access ro
Unable to perform operation while the virtual DVD is loaded.
```

You have to unload the media from the virtual CD-ROM, change the access node to read only, and then you can load it onto other virtual CD-ROMs using the commands shown in Example 5-17.

Example 5-17 Media access

```
$ unloadopt -vtd vtopt0
$ chvopt -name aix61 -access ro
$ loadopt -disk aix61 -vtd vtopt0
$ loadopt -disk aix61 -vtd vtopt1
```

Note: If you want to have the same media available simultaneously to multiple partitions, you can load the virtual media on multiple virtual CD-ROMs. However, you need to protect media integrity by setting the access privileges to read only. If you try to load the media on multiple devices you will get an error similar to that shown below:

```
$ loadopt -disk aix61 -vtd vtopt1
```

The specified virtual disk is not read only and cannot be loaded into multiple devices simultaneously.

To change the access mode you have to unload the virtual media first. If you try to change the access mode while the media is still loaded you will get the following error:

```
$ chvopt -name aix61 -access ro
```

Unable to perform operation while the virtual DVD is loaded: You have to unload the media from the virtual CDROM, change the access node to read-only and then you can load it on all virtual CDROMs using the following commands:

```
$ unloadopt -vtd vtopt0
$ chvopt -name aix61 -access ro
```

5.6 Live application and partition mobility with Cognos BI components

In this chapter, we provide an overview of Live Partition Mobility with a high-level description of its features. We also show how Live Partition Mobility can be used in a practical business environment that includes Cognos components.

This section contains the following topics:

5.6.1 Overview of Live Partition Mobility

Live Partition Mobility allows you to migrate partitions that are running AIX and Linux operating systems along with their hosted applications from one physical server to another without disrupting application services. The migration maintains complete system transactional integrity. The migration transfers the entire system environment, including processor state, processor registers, memory, attached virtual devices, operating system, running applications, and connected users.

Live Partition Mobility is a feature of the PowerVM Enterprise Edition offering. The Live Partition Mobility is offered starting with POWER6 technology-based systems.

The migration can be performed either with a powered-off or an active partition as follows:

Inactive migration The logical partition is powered off and moved to the destination system.

Active migration The migration of the partition is performed while the operating system and the application are up and running.

Live Partition Mobility is a very powerful feature of IBM PowerVM. It can be combined with other virtualization technologies available with PowerVM such as logical partitions, Live Workload Partitions, and Active Memory Sharing to create a completely virtualized IT environment.

The benefits that Live Partition Mobility brings to your business are that it:

- Restructures the infrastructure without disrupting the service.
- Avoids the need to stop business services due to maintenance operations or modifications to existing systems.
- Meets challenging service-level agreements (SLAs) because it allows you to be proactive move running partitions and applications from one server to another.
- Balances workloads and resources by moving applications from heavily loaded systems to servers that have spare capacity.
- Consolidates your servers by providing an easy path to move applications from individual, stand-alone servers to consolidated systems.
- Changes configurations in a very simple and secure manner with limited administrator intervention.
- Optimizes the usage of global system resources and adjusts the infrastructure to accommodate new systems.

5.6.2 Capabilities of Live Partition Mobility

A partition migration operation can be performed either when a partition is powered off (inactive) or when a partition is providing a service (active). A logical partition can be migrated between two POWER6s or later technology-based systems if the destination system has enough resources to accommodate the migrating partition. There is no restriction with respect to the number of processing units or memory size, either for inactive or for active migrations.

Inactive migration

Inactive migration moves the definition of a powered-off logical partition from one system to another along with its network and disk configuration. No additional change in network or disk setup is required and the partition can be activated as soon as migration is completed.

The inactive migration procedure handles the reconfiguration of involved systems as follows:

- A new partition is created on the destination system. The partition retains the configuration present on the source system.
- Network access and disk data is preserved and made available to the new partition.
- On the source system, the partition configuration is removed and all involved resources are released.

When the services provided by the partition cannot be interrupted, its relocation can be performed without disrupting the service using active migration.

Active migration

By using active migration, a running partition is moved from a source system to a destination system with no disruption of partition operation or user service. For example, a partition that is hosting a live production database with normal user activities can be migrated to a second system with no loss of data, no loss of connectivity, and no effect on the running transactions.

An active migration performs the same operations as an inactive migration except that the operating system, the applications, and the services that they provide are not stopped during the process. The content of the physical memory of the logical partition is copied from system to system, allowing the transfer to be imperceptible to the users.

During an active migration, the applications continue to handle their normal workload. Disk data transactions, active network connections, user contexts, and the complete environment are migrated without any loss, and migration can be activated any time on any production partition.

No limitation exists with respect to a partition CPU resources and memory configuration, and multiple migrations can be executed concurrently. Both inactive and active migrations might involve partitions with any processing unit and memory size configuration.

5.6.3 Architecture of Live Partition Mobility

Live Partition Mobility has specific requirements. They can be divided in several classes as discussed in this section.

Hardware requirements

The hardware requirements are:

- Live Partition Mobility can be controlled by the Hardware Management Console (HMC) or the Integrated Virtualization Manager (IVM). Live Partition Mobility requires HMC Version 7 Release 3.2 or later. For more information about the Hardware Management Console, see:

  http://www.software.ibm.com/webapp/set2/sas/f/hmc/home.html

- Live Partition Mobility requires POWER6 or POWER7 technology-based systems with the PowerVM Enterprise Edition. Live Partition Mobility compliments other PowerVM virtualization features such as Virtual I/O Server, Virtual SCSI, Virtual Ethernet, and Shared Ethernet Adapters.
- Both source and destination systems must have specific firmware levels. Although there is a minimum required firmware level, each system may have a separate level of firmware. The level of the source system firmware must be compatible with the destination firmware. You can find details about firmware compatibility at:

  http://www.software.ibm.com/webapp/set2/sas/f/pm/migrate.html

Infrastructure requirements

The following infrastructure requirements are important to consider:

- The mobile partition network and disk access must be virtualized by using one or more Virtual I/O servers. At least one Virtual I/O server at release level 1.5.1.1 or later must be installed on both the source and the destination systems. For more information about the Virtual I/O server see:

  http://www.software.ibm.com/webapp/set2/sas/f/vios/download/home.html

- Network infrastructure: The migrating partition uses virtual LANs (VLANs) for network access. The VLANs must be bridged to a physical network using a Shared Ethernet Adapter residing in the Virtual I/O server partition. Both the source and the target system must have an appropriate Shared Ethernet Adapter environment to host a mobile partition. The existing LAN infrastructure must be configured so that migrating partitions can continue to communicate after the migration is completed.
- Storage infrastructure: The operating system, applications, and data on the mobile partition must reside on virtual storage on an external storage subsystem. An external storage should be accessible by both source and destination systems.

  For more details about the supported storage systems, see the Virtual I/O server data sheet for VIOS:

  http://www.software.ibm.com/webapp/set2/sas/f/vios/documentation/datasheet.html

- No physical adapters may be used by the mobile partition during the migration.

Figure 5-7 shows a basic infrastructure enabled for Live Partition Mobility using a single HMC. Each system is configured with a single Virtual I/O server partition. The mobile partition has only virtual access to network and disk resources. The Virtual I/O server on the destination system is connected to the same network and is configured to access the same disk space used by the mobile partition. For illustration purposes, the device numbers are all shown as zero, but in practice, they can vary considerably.

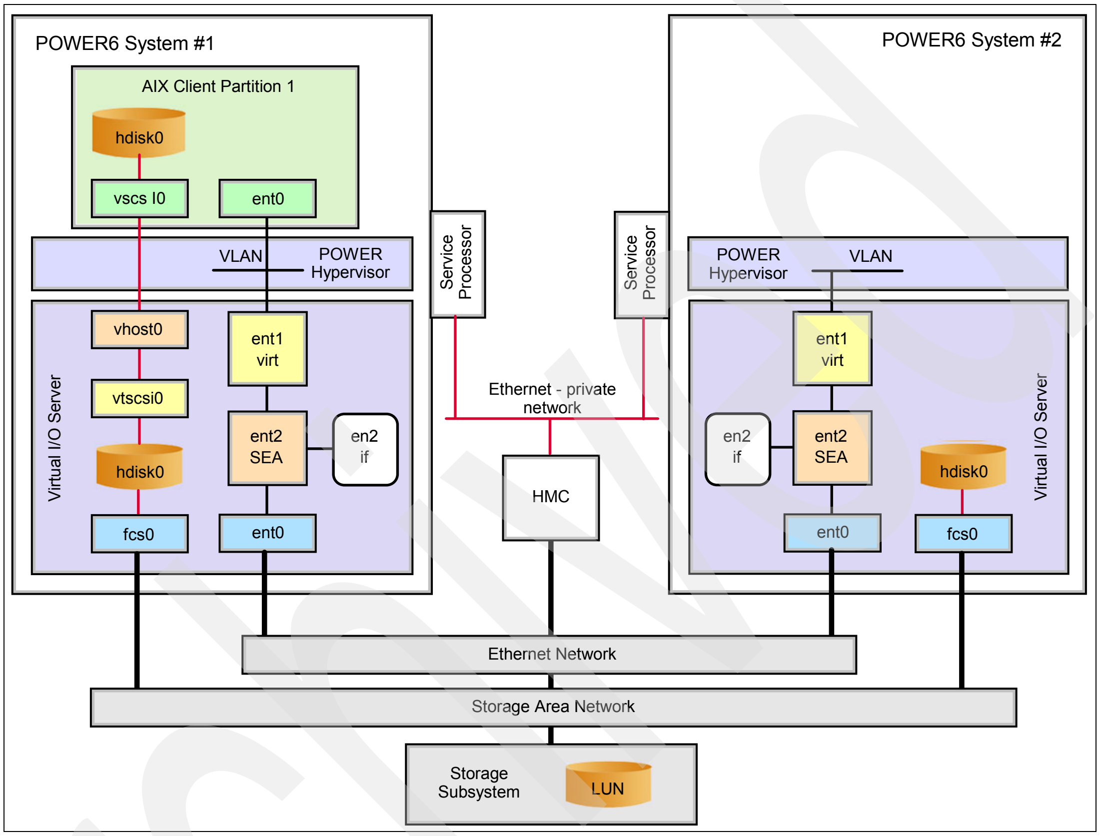


Figure 5-7 Basic infrastructure used for Live Partition Mobility

Operating system requirements

Mobile partitions should run AIX or Linux. Live Partition Mobility is supported for partitions running AIX 5.3 Level 5300-07 or later.

Kernel extensions can be used to register and notify migrations. The notification mechanism uses the standard dynamic reconfiguration mechanism.

Application requirements

Most applications do not require any changes to work with Live Partition Mobility. However, there are applications, such as those that use processor and memory affinity or processor binding, that need to be aware of the migration. They might recognize and adjust to changes made to the hardware environment by using standard AIX dynamic reconfiguration notification infrastructure.

5.6.4 Practical scenario using Live Partition Mobility

This section shows how we used the Live Partition Mobility feature of IBM PowerVM in an environment based on IBM POWER technology-based servers managed by a HMC. The mobile partition uses dedicated memory, and all disks and communication interfaces are accessible via a single VIO server.

We implemented the following environment:

- One IBM POWER6 server named p550 with one VIO server named VIOS_p550.
- One IBM POWER7 server named p7_1 with one VIO server named vios_p7_1
- HMC Version 7 Release 7.1.0 with MH01214.
- Virtual I/O Server Version 2.1.0.1-FP22
- AIX 6.1 TL 4

We created a Cognos distributed business environment with separate logical partitions dedicated to various Cognos components. The partition named p550was3 hosted by p550 has WebSphere Application Server installed to run one Cognos server. We migrated the partition while it was running to the p7_1 system using the following stages:

1. We verify the hardware infrastructure requirements.
2. We verify the VIO server requirements.
3. We verify partition requirements.
4. We migrate the partition.

Verifying hardware infrastructure requirements

To verify the hardware requirements:

1. We verify the current HMC model, version release, and service pack level using the **lshmc** command (Example 5-18).

Example 5-18 Verifying HMC requirements

```
hscroot@hmcdq1:~> lshmc -v
"vpd=*FC ????????
*VC 20.0
*N2 Fri Mar 19 13:45:12 EDT 2010
*FC ????????
*DS Hardware Management Console
*TM eserver xSeries 336 -[7310CR3]-
*SE 103297B
*MN IBM
*PN Unknown
*SZ 1059430400
*OS Embedded Operating Systems
*NA 10.0.0.1
*FC ????????
*DS Platform Firmware
*RM V7R7.1.0.0
"
```

2. We verify that HMC manages both source and target systems using the **`lssyscfg`** command (Example 5-19).

> **Note:** There is no requirement that the two systems are managed by the same HMC as long as the HMC has a minimum version of HMC V7R3.4.0.

Example 5-19 Verifying that the HMC manages both source and target systems

```
hscroot@hmcdq1:~> lssyscfg -r sys -F name type_model state
p570 9117-MMA Operating
p7_2 8233-E8B Operating
p7_1 8233-E8B Operating
p550 8204-E8A Operating
```

3. We verify that there is an RMC connection available between the HMC, the source and destination VIO servers, and the mobile partition using the **`lspartition`** command (Example 5-20). Note the *active* status for all partitions involved in the migration.

Example 5-20 Verifying RMC connectivity between HMC, VIO servers, and mobile partition

```
hscroot@hmcdq1:~> lspartition -dlpar
<#0> Partition:<2*9117-MMA*102A9C0, , 192.168.100.199>
       Active:<1>, OS:<AIX, 6.1, 6100-04-03-1009>, DCaps:<0xc5f>, CmdCaps:<0x1b, 0x1b>, PinnedMem:<1024>
<#1> Partition:<3*9117-MMA*102A9C0, , 192.168.100.198>
       Active:<1>, OS:<AIX, 6.1, 6100-04-03-1009>, DCaps:<0xc5f>, CmdCaps:<0x1b, 0x1b>, PinnedMem:<768>
<#2> Partition:<1*9117-MMA*102A9C0, , 192.168.100.237>
       Active:<1>, OS:<AIX, 6.1, 6100-04-00-0000>, DCaps:<0x79f>, CmdCaps:<0x1b, 0x1b>, PinnedMem:<768>
<#3> Partition:<2*8233-E8B*10029BP, , 192.168.100.245>
       Active:<1>, OS:<AIX, 6.1, 6100-04-00-0000>, DCaps:<0x79f>, CmdCaps:<0x1b, 0x1b>, PinnedMem:<848>
<#4> Partition:<1*8204-E8A*103BA12, , 192.168.100.238>
       Active:<1>, OS:<AIX, 6.1, 6100-04-00-0000>, DCaps:<0x79f>, CmdCaps:<0x1b, 0x1b>,
PinnedMem:<768>
<#5> Partition:<5*9117-MMA*102A9C0, , 192.168.100.200>
       Active:<1>, OS:<AIX, 6.1, 6100-04-03-1009>, DCaps:<0xc5f>, CmdCaps:<0x1b, 0x1b>, PinnedMem:<768>
<#6> Partition:<4*9117-MMA*102A9C0, , 192.168.100.197>
       Active:<1>, OS:<AIX, 6.1, 6100-04-03-1009>, DCaps:<0xc5f>, CmdCaps:<0x1b, 0x1b>, PinnedMem:<768>
<#7> Partition:<1*8233-E8B*100291P, , 192.168.100.240>
       Active:<1>, OS:<AIX, 6.1, 6100-04-00-0000>, DCaps:<0x79f>, CmdCaps:<0x1b, 0x1b>, PinnedMem:<860>
<#8> Partition:<3*8233-E8B*10029BP, , 192.168.100.206>
       Active:<1>, OS:<AIX, 6.1, 6100-04-03-1009>, DCaps:<0xc5f>, CmdCaps:<0x1b, 0x1b>, PinnedMem:<768>
<#9> Partition:<1*8233-E8B*10029BP, , 192.168.100.205>
       Active:<1>, OS:<AIX, 6.1, 6100-04-03-1009>, DCaps:<0xc5f>, CmdCaps:<0x1b, 0x1b>,
PinnedMem:<768>
<#10> Partition:<4*8233-E8B*10029BP, , 192.168.100.207>
       Active:<1>, OS:<AIX, 6.1, 6100-04-03-1009>, DCaps:<0xc5f>, CmdCaps:<0x1b, 0x1b>, PinnedMem:<1024>
<#11> Partition:<6*8233-E8B*100291P, , 192.168.100.246>
       Active:<1>, OS:<AIX, 6.1, 6100-04-03-1009>, DCaps:<0xc5f>, CmdCaps:<0x1b, 0x1b>, PinnedMem:<768>
<#12> Partition:<7*8233-E8B*100291P, , 192.168.100.247>
       Active:<1>, OS:<AIX, 6.1, 6100-04-03-1009>, DCaps:<0xc5f>, CmdCaps:<0x1b, 0x1b>, PinnedMem:<768>
<#13> Partition:<5*8233-E8B*100291P, , 192.168.100.244>
       Active:<1>, OS:<AIX, 6.1, 6100-04-03-1009>, DCaps:<0xc5f>, CmdCaps:<0x1b, 0x1b>, PinnedMem:<768>
<#14> Partition:<4*8233-E8B*100291P, , 192.168.100.243>
       Active:<1>, OS:<AIX, 6.1, 6100-04-03-1009>, DCaps:<0xc5f>, CmdCaps:<0x1b, 0x1b>, PinnedMem:<768>
<#15> Partition:<2*8233-E8B*100291P, , 192.168.100.242>
       Active:<1>, OS:<AIX, 6.1, 6100-04-03-1009>, DCaps:<0xc5f>, CmdCaps:<0x1b, 0x1b>, PinnedMem:<768>
<#16> Partition:<4*8204-E8A*103BA12, , 192.168.100.203>
```

```
       Active:<1>, OS:<AIX, 6.1, 6100-04-03-1009>, DCaps:<0xc5f>, CmdCaps:<0x1b, 0x1b>,
PinnedMem:<768>
<#17> Partition:<6*8204-E8A*103BA12, , 192.168.100.202>
       Active:<1>, OS:<AIX, 6.1, 6100-04-03-1009>, DCaps:<0xc5f>, CmdCaps:<0x1b, 0x1b>, PinnedMem:<768>
<#18> Partition:<2*8204-E8A*103BA12, , 192.168.100.201>
       Active:<1>, OS:<AIX, 6.1, 6100-04-03-1009>, DCaps:<0xc5f>, CmdCaps:<0x1b, 0x1b>, PinnedMem:<768>
<#19> Partition:<3*8233-E8B*100291P, , 192.168.100.241>
       Active:<1>, OS:<AIX, 6.1, 6100-04-03-1009>, DCaps:<0xc5f>, CmdCaps:<0x1b, 0x1b>, PinnedMem:<768>
```

4. We verify that systems have machine type, model, and microcode levels required to support Live Partition Mobility using the **lslic** command (Example 5-21).

Example 5-21 Verifying system type, model and microcode level

```
hscroot@hmcdq1:~> lslic -m p550
lic_type=Managed
System,management_status=Enabled,disabled_reason=,activated_level=75,installed_
level=75,accepted_level=75,ecnumber=01EL340,mtms=8204-E8A*103BA12,deferred_leve
l=None,platform_ipl_level=75,curr_level_primary=75,curr_ecnumber_primary=01EL34
0,curr_power_on_side_primary=temp,pend_power_on_side_primary=temp,temp_level_pr
imary=75,temp_ecnumber_primary=01EL340,perm_level_primary=75,perm_ecnumber_prim
ary=01EL340,update_control_primary=HMC

hscroot@hmcdq1:~> lslic -m p7_1
lic_type=Managed
System,management_status=Enabled,disabled_reason=,activated_level=39,installed_
level=39,accepted_level=39,ecnumber=01ZL710,mtms=8233-E8B*10029BP,deferred_leve
l=None,platform_ipl_level=39,curr_level_primary=39,curr_ecnumber_primary=01ZL71
0,curr_power_on_side_primary=temp,pend_power_on_side_primary=temp,temp_level_pr
imary=39,temp_ecnumber_primary=01ZL710,perm_level_primary=39,perm_ecnumber_prim
ary=01ZL710,update_control_primary=HMC
```

5. We verify logical memory block size for both systems and ensure that the logical memory block (LMB) size is the same on the source and destination systems using the **lshwres** command (Example 5-22).

Example 5-22 Verifying the logical memory block size

```
hscroot@hmcdq1:~> lshwres -r mem -m p550 --level sys -F mem_region_size
256
hscroot@hmcdq1:~> lshwres -r mem -m p7_1 --level sys -F mem_region_size
256
```

6. We verify that both the source and the target system support Live Partition Mobility using the **lssyscfg** command (Example 5-23).

Example 5-23 Verifying source and target system support for Live Partition Mobility

```
hscroot@hmcdq1:~> lssyscfg -r sys -m p550 -F active_lpar_mobility_capable
1
hscroot@hmcdq1:~> lssyscfg -r sys -m p7_1 -F active_lpar_mobility_capable
1
```

7. We verify that both VIO servers are configured as mover service partitions, and their clocks are synchronized using the **lssyscfg** command (Example 5-24).

Example 5-24 Verify the VIO servers

```
hscroot@hmcdq1:~> lssyscfg -r lpar -m p550 --filter "lpar_names=VIOS_p550" -F
msp time_ref --header
msp time_ref
1 1
hscroot@hmcdq1:~> lssyscfg -r lpar -m p7_1 --filter "lpar_names=vios_p7_1" -F
msp time_ref --header
msp time_ref
1 1
```

8. We verify the existing partition names and IDs on the target system. We also verify the parameters of the mobile partition on the source system. We can do both operations using the **lssyscfg** command. Note that partition p550was3 has ID 4, its type is AIX, it is up and running, it has been activated with the profile named profile, it has the redundant error path disabled, its clock is synchronized, its current processor compatibility mode is POWER6, it does not belong to any workgroup, and its name is not used on the target system (Example 5-25).

Example 5-25 Verify the existing partition names and IDs on the target system

```
hscroot@hmcdq1:~> lssyscfg -r lpar -m p550 --filter "lpar_names=p550was3" -F
name lpar_id os_version state default_profile redundant_err_path_reporting
time_ref curr_lpar_proc_compat_mode work_group_id --header
name lpar_id os_version state default_profile redundant_err_path_reporting
time_ref curr_lpar_proc_compat_mode work_group_id
p550was3 4 "AIX 6.1 6100-04-03-1009" Running profile 0 1 POWER6 none

hscroot@hmcdq1:~> lssyscfg -r lpar -m p7_1 -F name lpar_id
p7was3 4
p7was2 3
vios_p7_1 2
p7was1 1
```

9. We verify that the p550was3 mobile partition is not started with a profile that uses barrier synchronization registers or huge pages using the **lssyscfg** command (Example 5-26).

Example 5-26 Verifying that the mobile partition has started

```
hscroot@hmcdq1:~> lssyscfg -r prof -m p550 --filter "lpar_names=p550was3" -F
max_num_huge_pages bsr_arrays
0 0
```

10. We verify the processor compatibility modes that are supported by the target system using the **lssyscfg** command (Example 5-27).

Example 5-27 Verifying the processor compatibility modes

```
hscroot@hmcdq1:~> lssyscfg -r sys -m p7_1 -F lpar_proc_compat_modes
"default,POWER6,POWER6+,POWER7"
```

Verifying VIO server requirements

To verify the VIO server requirements for our environment:

1. We verify the current release of the source Virtual I/O server using the **ioslevel** command (Example 5-28).
2. We verify the virtual adapters defined on the system using the **lsmap** command. The vhost2 virtual adapter is assigned to a partition with ID 4.
3. We verify that the source Virtual I/O Server uses only physical disks as backing devices for virtual adapter vhost4 using the **lsmap** command (Example 5-28). The p550was3 (ID 4) partition is running and has only one physical disk named hdisk31. The mobile partition must not use logical volumes or storage pools.
4. We verify that hdisk31 is located on shared storage and has reserve policy set to no_reserve using the **lsdev** command (Example 5-28). Note the unique ID of hdisk31.
5. We verify that the virtual SCSI server adapter is not required for the VIO server and only the client SCSI adapter of the p550was3 mobile partition can connect to the server adapter.

Example 5-28 Verifying source VIO server version and storage settings

```
$ uname -n
viosp550
$ ioslevel
2.1.2.0
$ lsmap -all|grep vhost
vhost0          U8204.E8A.103BA12-V1-C11                      0x00000002
vhost1          U8204.E8A.103BA12-V1-C12                      0x00000000
vhost2      U8204.E8A.103BA12-V1-C14            0x00000004
vhost3          U8204.E8A.103BA12-V1-C17                      0x00000006

$ lsmap -vadapter vhost2
SVSA            Physloc                                       Client Partition ID
--------------- --------------------------------------------- ------------------
vhost2          U8204.E8A.103BA12-V1-C14                      0x00000004

VTD                   disk1_p550was3
Status                Available
LUN                   0x8100000000000000
Backing device        hdisk31
Physloc               U78A0.001.DNWGLVC-P1-C5-T2-W201300A0B811A662-L15000000000000
$ lsdev -type disk|grep hdisk31
hdisk31          Available   MPIO Other DS4K Array Disk
$ lsdev -dev hdisk31 -attr|grep reserve
reserve_policy  no_reserve
Reserve Policy                   True
$ lsdev -dev hdisk31 -attr|grep unique
unique_id       3E213600A0B8000291B080000BC4B04BCFE140F1815      FAStT03IBMfcp
Unique device identifier         False
```

6. We verify that the source VIO server uses a Shared Ethernet Adapter (SEA) adapter. The SEA interface uses a physical PCI Ethernet adapter, ent0, to communicate with the external environment, and the ent2 virtual interface to communicate with the p550was3 partition (Example 5-29) with the **lsdev** commands.

Example 5-29 Verifying source VIO server networking requirements

```
$ uname -n
viosp550
$ lsdev -type adapter|grep ent
ent0         Available   2-Port 10/100/1000 Base-TX PCI-X Adapter (14108902)
ent1         Available   2-Port 10/100/1000 Base-TX PCI-X Adapter (14108902)
ent2         Available   Virtual I/O Ethernet Adapter (l-lan)
ent3         Available   Shared Ethernet Adapter
$ lsdev -dev ent3 -attr
attribute     value    description
user_settable

accounting    disabled Enable per-client accounting of network statistics          True
ctl_chan               Control Channel adapter for SEA failover                    True
gvrp          no       Enable GARP VLAN Registration Protocol (GVRP)               True
ha_mode       disabled High Availability Mode                                      True
jumbo_frames  no       Enable Gigabit Ethernet Jumbo Frames                        True
large_receive no       Enable receive TCP segment aggregation                      True
largesend     0        Enable Hardware Transmit TCP Resegmentation                 True
netaddr       0        Address to ping                                             True
pvid          1        PVID to use for the SEA device                              True
pvid_adapter  ent2     Default virtual adapter to use for non-VLAN-tagged packets  True
qos_mode      disabled N/A                                                         True
real_adapter  ent0     Physical adapter associated with the SEA                    True
thread        1        Thread mode enabled (1) or disabled (0)                     True
virt_adapters ent2     List of virtual adapters associated with the SEA (comma separated) True
```

7. We verify the current release of the destination Virtual I/O server and the existing SCSI server adapters using the **ioslevel** and **lsmap** commands (Example 5-30). The destination VIO server has a different release.

Example 5-30 Verifying destination VIO server version and storage settings

```
$ uname -n
viosp71
$ ioslevel
2.1.2.10-FP-22
$ lsmap -all|grep vhost
vhost0          U8233.E8B.10029BP-V2-C3                      0x00000001
vhost1          U8233.E8B.10029BP-V2-C4                      0x00000003
vhost2          U8233.E8B.10029BP-V2-C5                      0x00000004
vhost4          U8233.E8B.10029BP-V2-C20                     0x00000000
```

8. We verify that the destination VIO server uses a SEA adapter. The SEA interface uses a physical HEA adapter, ent0, to communicate with the external environment and the ent1 virtual interface to communicate with logical partitions (Example 5-31) with the **lsdev** commands. The system also has PCI Ethernet adapters, but they are not used. Also note the content of the VIO server ARP table.

Example 5-31 Verifying destination VIO server networking requirements

```
$ uname -n
viosp71
$ lsdev -type adapter|grep ent
ent0          Available   Logical Host Ethernet Port (lp-hea)
ent1          Available   Virtual I/O Ethernet Adapter (l-lan)
ent2          Available   Shared Ethernet Adapter
ent3             Available   4-Port 10/100/1000 Base-TX PCI-X Adapter (14101103)
ent4             Available   4-Port 10/100/1000 Base-TX PCI-X Adapter (14101103)
ent5             Available   4-Port 10/100/1000 Base-TX PCI-X Adapter (14101103)
ent6             Available   4-Port 10/100/1000 Base-TX PCI-X Adapter (14101103)
$ lsdev -dev ent2 -attr
attribute     value    description
user_settable

accounting    disabled Enable per-client accounting of network statistics          True
ctl_chan               Control Channel adapter for SEA failover                    True
gvrp          no       Enable GARP VLAN Registration Protocol (GVRP)               True
ha_mode       disabled High Availability Mode                                      True
jumbo_frames  no       Enable Gigabit Ethernet Jumbo Frames                        True
large_receive no       Enable receive TCP segment aggregation                      True
largesend     0        Enable Hardware Transmit TCP Resegmentation                 True
netaddr       0        Address to ping                                             True
pvid          1        PVID to use for the SEA device                              True
pvid_adapter  ent1     Default virtual adapter to use for non-VLAN-tagged packets  True
qos_mode      disabled N/A                                                         True
real_adapter  ent0     Physical adapter associated with the SEA                    True
thread        1        Thread mode enabled (1) or disabled (0)                     True
virt_adapters ent1     List of virtual adapters associated with the SEA (comma separated) True
$ uname -n
viosp71
$ lstcpip -arp

  ? (192.168.100.60) at 0:9:6b:6e:0:e7 [ethernet] stored in bucket 8

  ? (192.168.100.231) at 0:14:5e:81:4d:49 [ethernet] stored in bucket 30


There are 2 entries in the arp table.
```

Verifying partition requirements

To verify the partition requirements:

1. We verify that the partition does not have required physical adapters or any virtual serial adapters other than the two default adapters with IDs 0 and 1.
2. We verify that the migrating partition has only virtual adapters and that the adapters are available using the **lsdev** command (Example 5-32). The p550was3 partition has only one virtual SCSI adapter and one virtual Ethernet adapter.
3. We verify from the partition side that there is an RMC connection available between the HMC and the mobile partition. The p550was3 partition is managed by the HMC that manages the entire environment, as shown by the **lsrsrc** command in Example 5-32. Also note the IP and MAC addresses associated to this partition as shown by the **netstat** command.
4. We verify the number, type, and frequency of the processors used by the mobile partition, as shown with the **lsdev**, **lsattr**, and **lscfg** commands in Example 5-32.

Example 5-32 Verifying partition requirements

```
p550was3(root)/> lsdev -Cc adapter
ent0   Available  Virtual I/O Ethernet Adapter (l-lan)
vsa0   Available  LPAR Virtual Serial Adapter
vscsi0 Available  Virtual SCSI Client Adapter
p550was3(root)/> netstat -in
Name  Mtu   Network     Address          ZoneID    Ipkts Ierrs    Opkts Oerrs
Coll
en0   1500  link#2     ae.c2.24.7d.e7.4      -   151810     0     2759     0
0
en0   1500  192.168.100 192.168.100.203      -   151810     0     2759     0
0
lo0   16896 link#1                           -    11531     0    11535     0
0
lo0   16896 127        127.0.0.1             -    11531     0    11535     0
0
lo0   16896 ::1                              0    11531     0    11535     0
0

p550was3(root)/> lsrsrc IBM.ManagementServer
Resource Persistent Attributes for IBM.ManagementServer
resource 1:
        Name             = "192.168.100.231"
        Hostname         = "192.168.100.231"
        ManagerType      = "HMC"
        LocalHostname    = "192.168.100.203"
        ClusterTM        = "9078-160"
        ClusterSNum      = ""
        ActivePeerDomain = ""
        NodeNameList     = {"p550was3"}

p550was3(root)/> lsdev -C|grep proc
proc0       Available 00-00 Processor
p550was3(root)/> lsattr -El proc0
frequency   4204000000     Processor Speed       False
smt_enabled true           Processor SMT enabled False
smt_threads 2              Processor SMT threads False
state       enable         Processor state       False
```

```
type        PowerPC_POWER6 Processor type        False
p550was3(root)/> lscfg -vpl proc0
  proc0              Processor

  PLATFORM SPECIFIC

  Name:  PowerPC,POWER6
    Node:  PowerPC,POWER6@0
    Device Type:  cpu
```

Migrating the mobile partition

To migrate the mobile partition, we took the following steps:

1. We saved for future reference the configuration of the mobile partition given by the **lssyscfg** command (Example 5-33). A few points of interest include:
   - Physical and virtual processor configuration, dedicated or shared and entitlements (minimum, maximum, and desired)
   - Memory configuration (minimum, maximum, and desired)
   - Virtual adapter configuration
   - Virtual device mappings

Example 5-33 Active configuration of p550was partition before migration

```
hscroot@hmcdq1:~> lssyscfg -r prof -m p550 --filter "lpar_names=p550was3"
name=profile,lpar_name=p550was3,lpar_id=4,lpar_env=aixlinux,all_resources=0,min
_mem=1024,desired_mem=4096,max_mem=4096,min_num_huge_pages=0,desired_num_huge_p
ages=0,max_num_huge_pages=0,mem_mode=ded,proc_mode=shared,min_proc_units=1.0,de
sired_proc_units=1.0,max_proc_units=2.0,min_procs=1,desired_procs=1,max_procs=2
,sharing_mode=uncap,uncap_weight=128,shared_proc_pool_id=0,shared_proc_pool_nam
e=DefaultPool,io_slots=none,lpar_io_pool_ids=none,max_virtual_slots=10,"virtual
_serial_adapters=0/server/1/any//any/1,1/server/1/any//any/1",virtual_scsi_adap
ters=2/client/1/VIOS_p550/14/1,virtual_eth_adapters=4/0/1//0/1/ETHERNET0,hca_ad
apters=none,boot_mode=norm,conn_monitoring=0,auto_start=0,power_ctrl_lpar_ids=n
one,work_group_id=none,redundant_err_path_reporting=0,bsr_arrays=0,lhea_logical
_ports=none,lhea_capabilities=none,lpar_proc_compat_mode=default,electronic_err
_reporting=null,virtual_fc_adapters=none
```

2. We verify the available slots on the target system and list possible and suggested mappings of virtual SCSI adapters used by the mobile partition using the **lslparmigr** command (Example 5-34).

Example 5-34 Listing possible and suggested mappings of virtual SCSI adapters

```
hscroot@hmcdq1:~> lslparmigr -r virtualio -m p550 -t p7_1 --filter
lpar_names=p550was3
"available_vios_virtual_slots=""vios_p7_1/2/6-19,21-24""",possible_virtual_scsi
_mappings=2/vios_p7_1/2,suggested_virtual_scsi_mappings=2/vios_p7_1/2/14,possib
le_virtual_fc_mappings=none,suggested_virtual_fc_mappings=none
```

3. We verify the valid source and destination mover service partitions that can be used for the migration using the **lslparmigr** command (Example 5-35). Since both the source and the target systems have only one VIO server partition, they are selected automatically.

Example 5-35 Listing the possible source and destination mover service partitions

```
hscroot@hmcdq1:~> lslparmigr -r msp -m p550 -t p7_1 --filter
"lpar_names=p550was3"
source_msp_name=VIOS_p550,source_msp_id=1,dest_msp_names=vios_p7_1,dest_msp_ids
=2,ipaddr_mappings=192.168.100.238//2/vios_p7_1/192.168.100.245/
```

4. We verify that we do not exceed the maximum number of simultaneous migration (Example 5-36).

Example 5-36 Verifying the number of current migrations

```
hscroot@hmcdq1:~> lslparmigr -r sys -m p550 -F num_active_migrations_supported
num_active_migrations_in_progress
80 0
```

5. We used the graphical user interface to migrate the partition. We selected **p7_1** as a target system from a set of potential target systems (Figure 5-8).

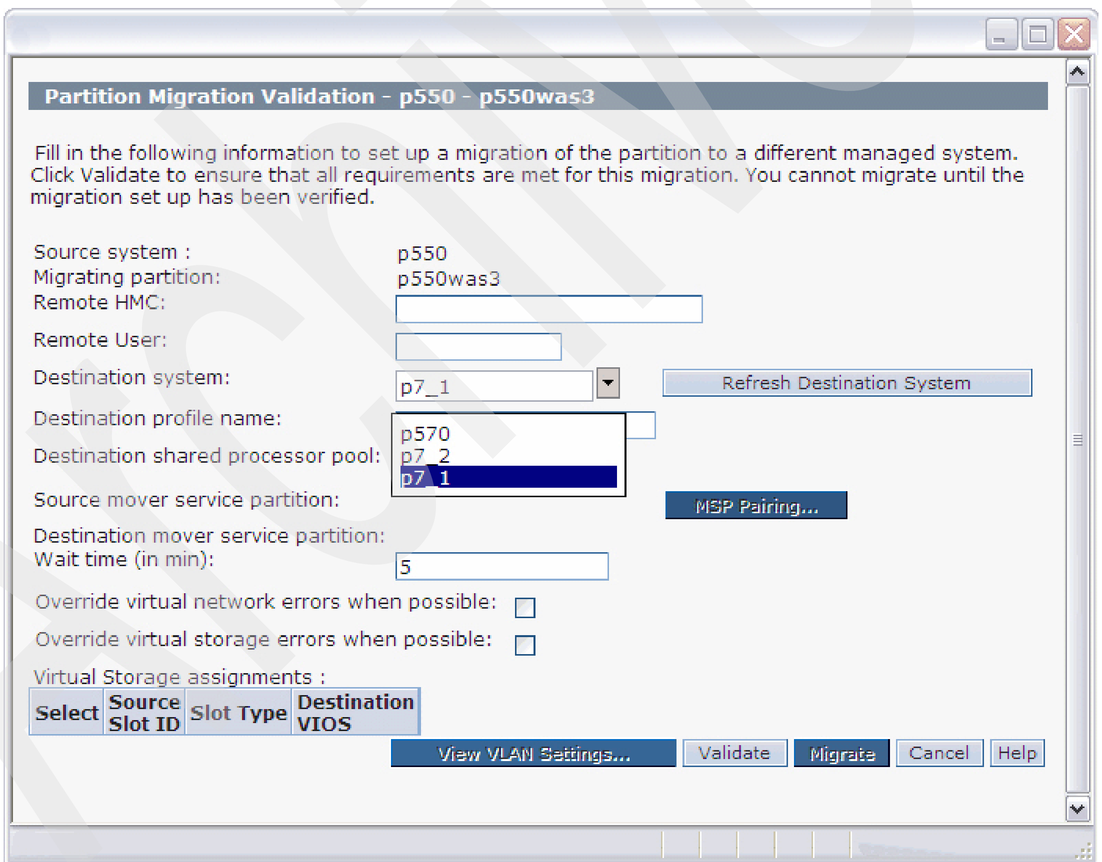


Figure 5-8 Selecting the target system to validate partition migration

6. We validate and start the migration (Figure 5-9). In our scenario there is only one Virtual I/O server partition per system, so source and destination partitions cannot be changed. The target system has only one shared processor pool, which is selected by default.

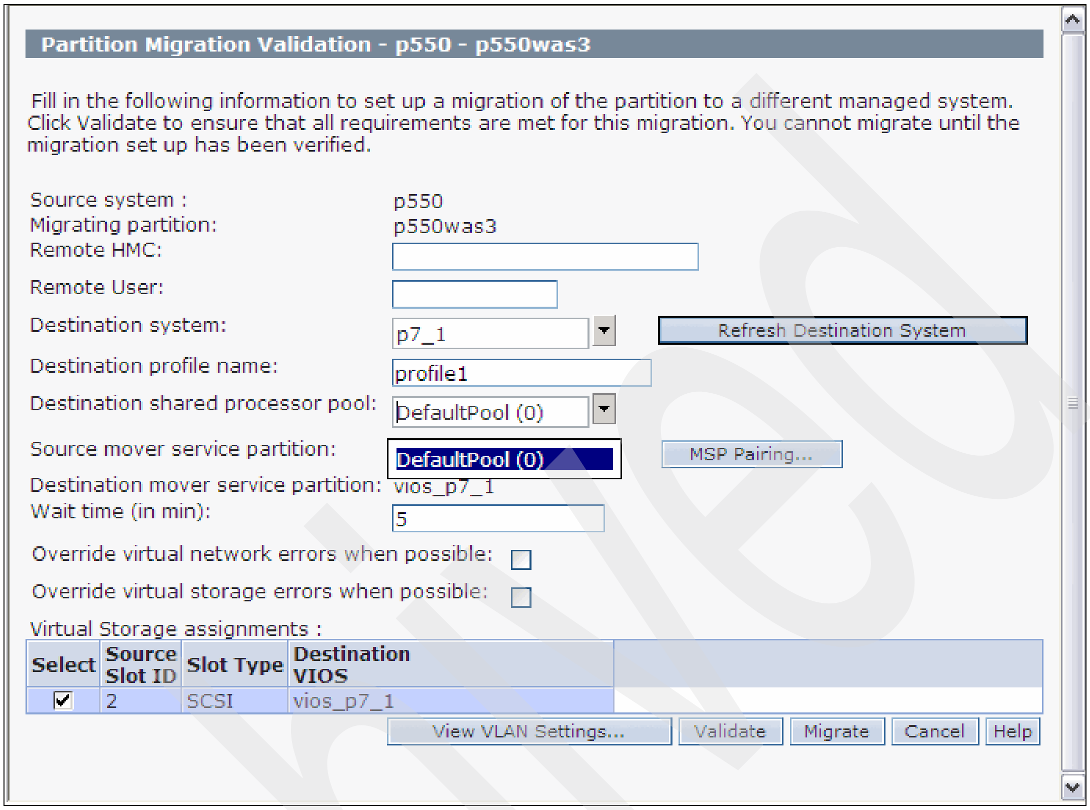


Figure 5-9 Validating and starting the migration

7. The migration progress window changes its status and goes through various intermediary states until migration is complete (Figure 5-10, Figure 5-11 on page 128, and Figure 5-12 on page 128).

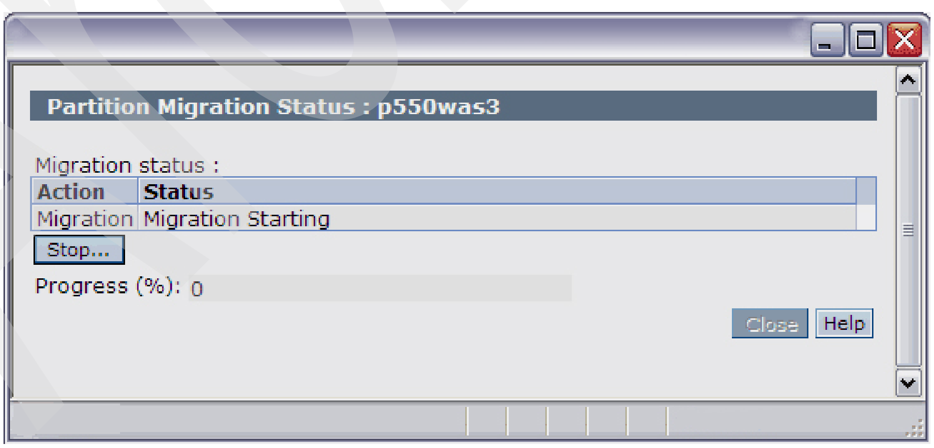


Figure 5-10 Migration initial status

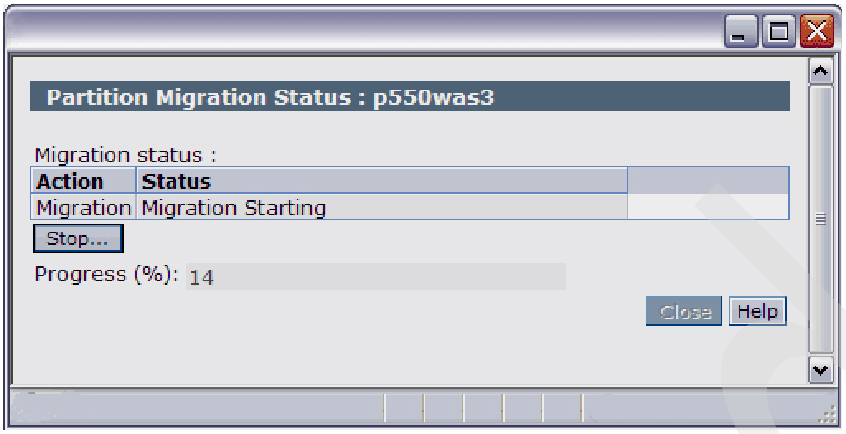


Figure 5-11   Migration intermediary status

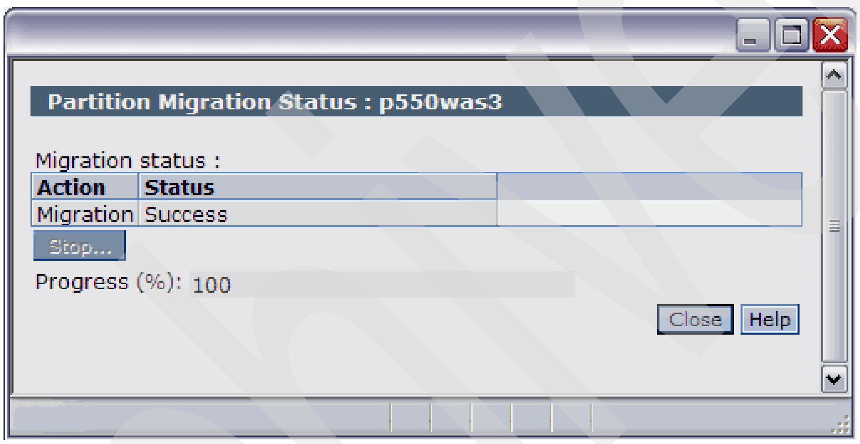


Figure 5-12   Migration successfully completed

8. Once the partition migration status window indicates that the migration is 100% complete, the mobile partition is up and running on the target system. The mobile partition has been deleted from the source system and now resides on the destination system, as shown with the **lssyscfg** command. The partition retained its name and has been assigned an ID of 5 (Example 5-37).

Example 5-37   Partition deleted from the source system and moved on the target system

```
hscroot@hmcdq1:~> lssyscfg -r lpar -m p550 -F name lpar_id
VIOS_p550 1
p550was1 2
p550was2 6
hscroot@hmcdq1:~> lssyscfg -r lpar -m p7_1 -F name lpar_id
p7was3 4
p7was2 3
vios_p7_1 2
p7was1 1
p550was3 5
```

9. Once the migration has completed, the mappings on the source VIO server are updated. In our scenario, the definition of vhost 2 has been removed from the source VIO server (Example 5-38).

Example 5-38 Updating source VIO server mappings after partition migration

```
$ uname -n
viosp550
$ lsmap -all|grep vhost
vhost0          U8204.E8A.103BA12-V1-C11                     0x00000002
vhost1          U8204.E8A.103BA12-V1-C12                     0x00000000
vhost3          U8204.E8A.103BA12-V1-C17                     0x00000006
```

10. Once the migration has completed, the mappings on the target VIO server are updated. In our scenario the definition of vhost 5 has been added to the target VIO server (Example 5-39). The new virtual adapter vhost5 is assigned to the partition with an ID of 5. The shared disk is now seen as hdisk29. The unique ID of hdisk29 is identical to the unique ID of hdisk31 from the source system. The name of the virtual target device has been preserved.

Example 5-39 Updating target VIO server mappings after partition migration

```
$ lsmap -all|grep vhost
vhost0          U8233.E8B.10029BP-V2-C3                      0x00000001
vhost1          U8233.E8B.10029BP-V2-C4                      0x00000003
vhost2          U8233.E8B.10029BP-V2-C5                      0x00000004
vhost4          U8233.E8B.10029BP-V2-C20                     0x00000000
vhost5          U8233.E8B.10029BP-V2-C14                     0x00000005
$ lsmap -vadapter vhost5
SVSA            Physloc                                      Client Partition ID
--------------- -------------------------------------------- ------------------
vhost5          U8233.E8B.10029BP-V2-C14                     0x00000005

VTD                   disk1_p550was3
Status                Available
LUN                   0x8100000000000000
Backing device        hdisk29
Physloc               U78A0.001.DNWHFSP-P1-C1-T1-W201300A0B811A662-L15000000000000
$ lsdev -dev hdisk29 -attr|grep unique
unique_id       3E213600A0B8000291B080000BC4B04BCFE140F1815      FAStT03IBMfcp
Unique device identifier        False
```

11. The mobile partition profile has been moved on the target system (Example 5-40). Before migration the partition had only one profile. Since we specified a new profile named profile1 when we started the migration, the partition definition now includes also the newly created profile.

Example 5-40 Current configuration of p550was partition after migration

```
hscroot@hmcdq1:~> lssyscfg -r prof -m p7_1 --filter "lpar_names=p550was3"
name=profile,lpar_name=p550was3,lpar_id=5,lpar_env=aixlinux,all_resources=0,min
_mem=1024,desired_mem=4096,max_mem=4096,min_num_huge_pages=0,desired_num_huge_p
ages=0,max_num_huge_pages=0,mem_mode=ded,mem_expansion=null,proc_mode=shared,mi
n_proc_units=1.0,desired_proc_units=1.0,max_proc_units=2.0,min_procs=1,desired_
procs=1,max_procs=2,sharing_mode=uncap,uncap_weight=128,shared_proc_pool_id=0,s
hared_proc_pool_name=DefaultPool,io_slots=none,lpar_io_pool_ids=none,max_virtua
l_slots=10,"virtual_serial_adapters=0/server/1/any//any/1,1/server/1/any//any/1
```

```
",virtual_scsi_adapters=2/client/1/p7was1/14/1,virtual_eth_adapters=4/0/1//0/1/
ETHERNET0,hca_adapters=none,boot_mode=norm,conn_monitoring=0,auto_start=0,power
_ctrl_lpar_ids=none,work_group_id=none,redundant_err_path_reporting=0,bsr_array
s=0,lhea_logical_ports=none,lhea_capabilities=none,lpar_proc_compat_mode=defaul
t,electronic_err_reporting=null,virtual_fc_adapters=none
name=profile1,lpar_name=p550was3,lpar_id=5,lpar_env=aixlinux,all_resources=0,mi
n_mem=1024,desired_mem=4096,max_mem=4096,min_num_huge_pages=null,desired_num_hu
ge_pages=null,max_num_huge_pages=null,mem_mode=ded,mem_expansion=null,proc_mode
=shared,min_proc_units=1.0,desired_proc_units=1.0,max_proc_units=2.0,min_procs=
1,desired_procs=1,max_procs=2,sharing_mode=uncap,uncap_weight=128,shared_proc_p
ool_id=0,shared_proc_pool_name=DefaultPool,io_slots=none,lpar_io_pool_ids=none,
max_virtual_slots=10,"virtual_serial_adapters=0/server/1/any//any/1,1/server/1/
any//any/1",virtual_scsi_adapters=2/client/2/vios_p7_1/14/1,virtual_eth_adapter
s=4/0/1//0/1/ETHERNET0,hca_adapters=none,boot_mode=norm,conn_monitoring=0,auto_
start=0,power_ctrl_lpar_ids=none,work_group_id=none,redundant_err_path_reportin
g=null,bsr_arrays=null,lhea_logical_ports=none,lhea_capabilities=none,lpar_proc
_compat_mode=default,electronic_err_reporting=null,virtual_fc_adapters=none
```

12. The partition AIX error log now contains two entries pertaining to the migration (Example 5-41). Informational entries are added when the partition migration is started and when partition migration is completed.

Example 5-41 Notification messages in the AIX mobile partition error log

```
LABEL:          CLIENT_PMIG_DONE
IDENTIFIER:     A5E6DB96

Date/Time:       Sun Mar 21 11:21:54 EDT 2010
Sequence Number: 31
Machine Id:      0003BA12D900
Node Id:         p550was3
Class:           S
Type:            INFO
WPAR:            Global
Resource Name:   pmig

Description
Client Partition Migration Completed

---------------------------------------------------------------------------
LABEL:          CLIENT_PMIG_STARTED
IDENTIFIER:     08917DC6

Date/Time:       Sun Mar 21 11:21:50 EDT 2010
Sequence Number: 30
Machine Id:      00E19BB54C00
Node Id:         p550was3
Class:           S
Type:            INFO
WPAR:            Global
Resource Name:   pmig

Description
Client Partition Migration Started
```

13. We verify the number, type, and frequency of the processors used by the mobile partition, as shown by the **lsdev**, **lsattr**, and **lscfg** commands in Example 5-42. Note that the processor type and frequency changed.

Example 5-42 Verifying the number, type, and frequency of processor after migration completed

```
p550was3(root)/> lsdev -C|grep proc
proc0      Available 00-00 Processor
p550was3(root)/> lsattr -El proc0
frequency   3000000000    Processor Speed       False
smt_enabled true          Processor SMT enabled False
smt_threads 2             Processor SMT threads False
state       enable        Processor state       False
type        PowerPC_POWER Processor type        False
p550was3(root)/> lsattr -El proc0
frequency   3000000000    Processor Speed       False
smt_enabled true          Processor SMT enabled False
smt_threads 2             Processor SMT threads False
state       enable        Processor state       False
type        PowerPC_POWER Processor type        False
p550was3(root)/> lscfg -vpl proc0
  proc0             Processor

  PLATFORM SPECIFIC

  Name:  PowerPC,POWER
    Node:  PowerPC,POWER6@0
    Device Type:  cpu
```

5.7 Active Memory Expansion

This section presents a new feature available on power7 systems: Active Memory Expansion. The section provides an overview of the Active Memory Expansion features as well as guidance for planning and deploying this feature in Power Systems based server environments.

5.7.1 Overview

Active Memory Expansion is a new feature available for IBM POWER7 systems that allows expansion of a system's effective memory capacity. Active Memory Expansion uses memory compression to compact data residing in memory. Thus, more data can be placed in memory and the memory capacity of POWER7 systems can be enlarged.

Active Memory Expansion can be enabled at the logical partition level. The compression of data residing in memory data is managed by the operating system, and this compression is transparent to applications and users.

From the entire amount of data stored in memory, the operating system identifies the data eligible for compression and stores it in a compressed pool. The remaining data remains uncompressed. The operating system dynamically adjusts the amount of memory being compressed based on the current workload and the configuration of the logical partition. The operating system moves data between the compressed and uncompressed memory pools based on the application requests. At the time of writing, AIX is the only operating system that supports Active Memory Expansion.

Because Active Memory Expansion is based on memory compression, it is expected that additional CPU resources will be utilized when Active Memory Expansion is being used. The amount of additional CPU resources used varies based on the system workload characteristics and the level of memory expansion being used. So prior to using Active Memory Expansion (AME), careful planning is required to ensure that the partition workload gets the maximum benefit from AME.

The benefits of using the Active Memory Expansion feature are that it:

- Improves overall system utilization
- Helps to increase system throughput
- Consolidates memory-constrained environments

5.7.2 Active Memory Expansion versus Active Memory Sharing

Both Active Memory Expansion and Active Memory Sharing are memory virtualization technologies supported on Power Systems servers. However, each feature provides a different set of capabilities.

Active Memory Expansion enables you to extend the amount of logical memory available to a partition by compressing the data residing in memory. *AME scope* is contained within the limits of the logical partition. AME *increases* the amount of logical memory a partition perceives. When using AME, the amount of logical memory that a partition may access is *variable*. The AME mechanism is controlled by the operating system.

Active Memory Sharing enables you to improve usage of the physical memory existing on the system by allowing sharing of a physical memory pool among multiple logical partitions. AMS scope is contained within the limits of the server on which partitions are defined. Active Memory Sharing *does not* increase the amount of logical memory that a partition perceives that it may address. When using AMS, the amount of logical memory that a partition may access is *fixed*. AMS controls the distribution of physical memory among multiple partitions. The amount of physical memory that can be addressed by partitions using AMS is controlled by the Hypervisor.

Since Active Memory Expansion and Active Memory Sharing provide different capabilities, they can be used independently or together.

5.7.3 Active Memory Expansion requirements and usage

Active Memory Expansion is supported on all IBM POWER7 systems. To enable AME you need the following minimum requirements:

- HMC V7R7.1.0.0
- Firmware level 710
- AIX 6.1 TL4 SP2

AME planning tool

Prior to starting to use AME, you need to determine how an individual partition can benefit from AME. You can use the Active Memory Expansion planning tool (**amepat**) that is shipped with the AIX media. The AME planning tool helps monitor memory usage when the partition is running a particular workload and identifies data that can be compressed while the workload is being run. The AME planning tool runs on any partition running AIX 6.1, so you can use it to estimate the hardware requirements before you migrate your partitions to POWER7 systems.

The tool generates a detailed report containing possible AME configurations that can be used for that particular workload. The report also includes an estimated CPU usage for all AME configurations and recommends an initial AME configuration to be used for further reference. To get accurate results, run the tool throughout the peak usage of your partition workload. Based on the data collected, the AME planning tool suggests the amount of memory to be allocated for the partition and a memory expansion factor.

Configuring a partition to use AME consists of enabling the partition profile to use AME and specifying the memory expansion factor — the only parameter that can be configured when using AME. Partition physical memory size multiplied by the memory expansion factor gives the amount of logical memory allocated to the partition.

Once you have activated the partition to use AME you have to monitor partition performance associated with a specific memory expansion factor. The memory expansion factor can be further adjusted for even better performance.

5.7.4 Practical scenario using AME

This section shows how we used the Active Memory Expansion feature in an environment based on IBM POWER technology-based servers managed by the HMC.

We used the following environment:

- One logical partition named p550was1 hosted by an IBM POWER6 server named p550. The partition is migrated later to a POWER7 system.
- HMC Version 7 Release 7.1.0 with MH01214.
- AIX 6.1 TL 4 SP3.

The partition is assigned three CPUs and 28 GB of memory to run a typical Cognos workload that lasts 60 minutes. Before migrating the partition to a POWER7 system, we run the **amepat** tool for 60 minutes to assess memory and CPU requirements (Example 5-43).

Example 5-43 Running amepat tool

```
p550was1(root)/> amepat 60

Command Invoked                : amepat 60

Date/Time of invocation        : Tue Mar 23 18:57:48 EDT 2010
Total Monitored time           : 1 hrs 7 mins 48 secs
Total Samples Collected        : 9

System Configuration:
---------------------
Partition Name                 : p550was1
Processor Implementation Mode  : POWER6_in_P6_mode
Number Of Logical CPUs         : 6
Processor Entitled Capacity    : 3.00
Processor Max. Capacity        : 3.00
True Memory                    : 28.00 GB
SMT Threads                    : 2
Shared Processor Mode          : Enabled-Uncapped
Active Memory Sharing          : Disabled
```

```
Active Memory Expansion       : Disabled

System Resource Statistics:                Average            Min
Max
---------------------------               -----------        -----------
-----------
CPU Util (Phys. Processors)               0.93 [ 31%]        0.24 [  8%]
2.51 [ 84%]
Virtual Memory Size (MB)                  6146 [ 21%]        3119 [ 11%]
6924 [ 24%]
True Memory In-Use (MB)                   7398 [ 26%]        4332 [ 15%]
8197 [ 29%]
Pinned Memory (MB)                        1712 [  6%]        1710 [  6%]
1714 [  6%]
File Cache Size (MB)                      1237 [  4%]        1198 [  4%]
1259 [  4%]
Available Memory (MB)                    21654 [ 76%]       20877 [ 73%]
24682 [ 86%]

Active Memory Expansion Modeled Statistics:
-------------------------------------------
Modeled Expanded Memory Size   :  28.00 GB
Average Compression Ratio      :   2.37

Expansion    Modeled True      Modeled              CPU Usage
Factor       Memory Size       Memory Gain          Estimate
---------    -------------     ------------------   -----------
     1.04         27.00 GB         1.00 GB [  4%]   0.00 [  0%]
     1.20         23.25 GB         4.75 GB [ 20%]   0.00 [  0%]
     1.37         20.50 GB         7.50 GB [ 37%]   0.00 [  0%]
     1.53         18.25 GB         9.75 GB [ 53%]   0.00 [  0%]
     1.70         16.50 GB        11.50 GB [ 70%]   0.00 [  0%]
     1.87         15.00 GB        13.00 GB [ 87%]   0.00 [  0%]
     2.04         13.75 GB        14.25 GB [104%]   0.32 [ 11%]

Active Memory Expansion Recommendation:
---------------------------------------
The recommended AME configuration for this workload is to configure the LPAR
with a memory size of 13.75 GB and to configure a memory expansion factor
of 2.04.  This will result in a memory gain of 104%. With this
configuration, the estimated CPU usage due to AME is approximately 0.32
physical processors, and the estimated overall peak CPU resource required for
the LPAR is 2.83 physical processors.

NOTE: amepat's recommendations are based on the workload's utilization level
during the monitored period. If there is a change in the workload's utilization
level or a change in workload itself, amepat should be run again.

The modeled Active Memory Expansion CPU usage reported by amepat is just an
estimate.  The actual CPU usage used for Active Memory Expansion may be lower
or higher depending on the workload.
```

`Amepat` reports that for the same workload, the amount of memory assigned to the partition can be reduced from 28 Gb to 13.75 GB at the expense of using 0.32 additional physical processors. Thus, we could save 28 - 13.75 = 14.25 GB of memory on the target system.

5.8 Tunables

The section provides information about tuning parameters that we use to tune the features in the hardware and software within our testing environment.

5.8.1 In AIX

In this section we discuss the parameters for AIX.

The AIX user limits

The primary optimization approach requires tuning of thread, memory, and lock management because the critical processing is implemented in a multithreaded, distributed architecture. Optimized thread handling within AIX ensures that Cognos Report Server (BI Bus) processes can most efficiently use the available processor resources.

Example 5-44 shows the AIX user limits. The ulimits must be set for the user account that is used to run Cognos 8 BI. These resource-limit settings are needed so that IBM Cognos 8 processes can run with maximum available resources on the system. Set the stack to 4,194,304 KB to allow IBM Cognos 8 to use the maximum addressable memory for a 32-bit process. Set all other parameters to unlimited. The limits can be edited in the `etc/security/limits` file (Example 5-44).

Example 5-44 AIX user limits

```
*
*   Attribute        Value
*   ==========    ============
*   fsize_hard    set to fsize
*   cpu_hard      set to cpu
*   core_hard         -1
*   data_hard         -1
*   stack_hard      8388608
*   rss_hard          -1
*   nofiles_hard      -1
*
* NOTE:  A value of -1 implies "unlimited"
*

default:
        fsize = -1
        core = -1
        cpu = -1
        data = -1
        rss = -1
        stack = 4194304
        nofiles = -1
```

Tuning memory allocation settings

IBM Cognos 8 BI server processes are multithreaded and are resource-intensive under load, issuing a large number of small allocation requests to AIX. The configuration of MALLOCTYPE=buckets provides an optional buckets-based extension of the default memory allocator. Setting MALLOCTYPE to buckets improves performance for IBM Cognos 8 BI because this instructs the memory allocator to accommodate threads that issue large numbers of small allocation requests, matching the requirements of IBM Cognos 8 processes. When the malloc buckets are enabled, allocation requests that fall within a predefined range of block sizes are processed by malloc buckets. All other requests are processed in the usual manner by the default memory allocator. For more information about the Malloc buckets, refer to:

http://publib16.boulder.ibm.com/doc_link/en_US/a_doc_lib/aixprggd/genprogc/malloc_buckets.htm

MALLOCMULTIHEAP=heaps:n (where n is two times the number of processors) is set for all IBM Cognos 8 servers.

With malloc multiheap capability enabled, the malloc subsystem creates a fixed number of heaps for its use. It begins to use multiple heaps after the second thread is started (process becomes multithreaded). Each memory-allocation request is serviced using one of the available heaps. The malloc subsystem can then process memory allocation requests in parallel, as long as the number of threads simultaneously requesting service is less than or equal to the number of heaps. If the number of threads simultaneously requesting service exceeds the number of heaps, additional simultaneous requests are serialized. Unless this occurs on an ongoing basis, the overall performance of the malloc subsystem should be significantly improved when multiple threads are making calls to the malloc subroutine in a multiprocessor environment. For more information about this setting, see:

http://publib16.boulder.ibm.com/doc_link/en_US/a_doc_lib/aixprggd/genprogc/malloc_multiheap.htm

Tuning AIX thread handling

IBM Cognos 8 BI server processes perform most of the time data-processing activity in the processor-kernel mode. Using the AIX environment variable setting AIXTHREAD_MINKTHREADS=32, the number of kernel threads that are available to IBM Cognos 8 processes is raised to 32 from the default of eight threads. This provides the IBM Cognos 8 server processes with more kernel threads and, hence, improves the efficiency of IBM Cognos 8 processor utilization. This especially applies to multiprocessor systems such as POWER6 processor. In addition to increasing the number of kernel threads, setting the ratio of user threads to kernel threads also improves IBM Cognos 8 performance. Using the AIX environment-variable setting of AIXTHREAD_MNRATIO=1:1 provides a 1-to-1 matching of user threads to kernel threads, which is a good setting for applications that use a large number of threads.

Given the large number of threads that Cognos 8 creates, thread-lock contention can cause requests to queue up, thereby degrading the efficiency of the processor utilization under default AIX configurations. Use of the AIX AIXTHREAD_MUTEX_FAST=ON enables the use of the optimized mutex-locking mechanism and results in improved Cognos 8 performance because of more efficient processor utilization.

5.8.2 In Cognos

A default installation of IBM Cognos 8 Business Intelligence contains all components in a single install location. To achieve high application load and application availability, each of the three IBM Cognos 8 Business Intelligence components can be deployed into multiple install locations either on a single physical Power server within multiple LPARs or across multiple physical servers.

The decision to deploy multiple IBM Cognos 8 server instances on a single physical server or across multiple physical servers can be driven by physical server availability, application availability, or resource availability. In a virtualized deployment, physical server resources can be shared between the IBM Cognos 8 Business Intelligence components.

When scaling an IBM Cognos 8 Business Intelligence deployment, capacity can be increased using the PowerVM features to:

- Add a new instance of the Cognos report server.
- Add CPU to an existing server.
- Add memory resources between the LPARs.
- Enable AMS between components.
- Enable AME to increase memory resources where memory resources are not available.

This chapter provided details about how to exploit IBM Power Systems PowerVM virtualization features with the IBM Cognos 8 BI solution. Chapter 6, “Experiences with Live Partition Mobility and Active Memory Expansion with Cognos 8 BI components” on page 139, goes into more detail about how to exploit additional virtualization features of the IBM Power System servers, particularly the IBM POWER 7 servers.

6

Experiences with Live Partition Mobility and Active Memory Expansion with Cognos 8 BI components

This chapter shows how to use Live Partition Mobility and Active Memory Expansion features in an environment based on IBM Power Systems servers managed by the Hardware Management Console (HMC). We use logical partitions running typical Cognos workloads on a POWER6 system, which later are migrated to a POWER7 system.

This chapter covers the following topics:

In this chapter, we discuss the following steps:

- We monitor the performance on both partitions before they are migrated to a POWER7 system.
- We use the amepat tool to assess the amount of memory required for partitions after their migration to a POWER7 system.
- We migrate the partitions.
- We monitor the performance on both partitions after they are migrated to a POWER7 system.
- We use the amepat tool to assess how memory is used after partitions are migrated to POWER7.

6.1 Using amepat to assess memory requirements

This section shows how to use the Active Memory Expansion features in the following environment:

- One logical partition named p550was1 hosted by an IBM POWER6 server named p550. The partition is migrated later to a POWER7 system named p7_1.
- One logical partition named p550was3 hosted by an BM POWER6 server named p550. The partition is migrated later to a POWER7 system named p7_1.

Both partitions are assigned three CPUs and 28 GB of memory to run typical Cognos workloads that last 60 minutes. Before migrating the partition to a POWER7 system, we run the **amepat** tool on the p550was1 partition for 60 minutes to assess the memory and CPU requirements (Example 6-1).

Example 6-1 Running amepat tool on p550was1

```
p550was1(root)/> amepat 60

Command Invoked                : amepat 60

Date/Time of invocation        : Tue Mar 23 18:57:48 EDT 2010
Total Monitored time           : 1 hrs 7 mins 48 secs
Total Samples Collected        : 9

System Configuration:
---------------------
Partition Name                 : p550was1
Processor Implementation Mode  : POWER6_in_P6_mode
Number Of Logical CPUs         : 6
Processor Entitled Capacity    : 3.00
Processor Max. Capacity        : 3.00
True Memory                    : 28.00 GB
SMT Threads                    : 2
Shared Processor Mode          : Enabled-Uncapped
Active Memory Sharing          : Disabled
Active Memory Expansion        : Disabled

System Resource Statistics:              Average            Min
Max
---------------------------             -----------        -----------
-----------
CPU Util (Phys. Processors)             0.93 [ 31%]        0.24 [  8%]
2.51 [ 84%]
Virtual Memory Size (MB)                6146 [ 21%]        3119 [ 11%]
6924 [ 24%]
True Memory In-Use (MB)                 7398 [ 26%]        4332 [ 15%]
8197 [ 29%]
Pinned Memory (MB)                      1712 [  6%]        1710 [  6%]
1714 [  6%]
File Cache Size (MB)                    1237 [  4%]        1198 [  4%]
1259 [  4%]
Available Memory (MB)                  21654 [ 76%]       20877 [ 73%]
24682 [ 86%]
```

```
Active Memory Expansion Modeled Statistics:
-------------------------------------------
Modeled Expanded Memory Size   :  28.00 GB
Average Compression Ratio      :   2.37

Expansion    Modeled True      Modeled              CPU Usage
Factor       Memory Size       Memory Gain          Estimate
---------    -------------     ------------------   -----------
     1.04         27.00 GB         1.00 GB [  4%]   0.00 [  0%]
     1.20         23.25 GB         4.75 GB [ 20%]   0.00 [  0%]
     1.37         20.50 GB         7.50 GB [ 37%]   0.00 [  0%]
     1.53         18.25 GB         9.75 GB [ 53%]   0.00 [  0%]
     1.70         16.50 GB        11.50 GB [ 70%]   0.00 [  0%]
     1.87         15.00 GB        13.00 GB [ 87%]   0.00 [  0%]
     2.04         13.75 GB        14.25 GB [104%]   0.32 [ 11%]

Active Memory Expansion Recommendation:
---------------------------------------
The recommended AME configuration for this workload is to configure the LPAR
with a memory size of 13.75 GB and to configure a memory expansion factor
of 2.04.  This will result in a memory gain of 104%. With this
configuration, the estimated CPU usage due to AME is approximately 0.32
physical processors, and the estimated overall peak CPU resource required for
the LPAR is 2.83 physical processors.

NOTE: amepat's recommendations are based on the workload's utilization level
during the monitored period. If there is a change in the workload's utilization
level or a change in workload itself, amepat should be run again.

The modeled Active Memory Expansion CPU usage reported by amepat is just an
estimate.  The actual CPU usage used for Active Memory Expansion may be lower
or higher depending on the workload.
```

Amepat reports that for the same workload, the amount of memory assigned to the partition can be reduced from 28 Gb to 13.75 GB at the expense of using 0.32 additional physical processors. Thus, we could save 28 - 13.75 = 14.25 GB of memory on the target POWER7 system.

Before migrating the partition to a POWER7 system, we run the **amepat** tool on the p550was3 partition for 60 minutes to assess memory and CPU requirements (Example 6-2).

Example 6-2 Running the amepat tool on p550was3

```
p550was3(root)/> amepat 60

Command Invoked               : amepat 60

Date/Time of invocation       : Tue Mar 23 18:57:48 EDT 2010
Total Monitored time          : 1 hrs 7 mins 48 secs
Total Samples Collected       : 9

System Configuration:
---------------------
Partition Name                : p550was3
Processor Implementation Mode : POWER6_in_P6_mode
Number Of Logical CPUs        : 6
```

```
Processor Entitled Capacity    : 3.00
Processor Max. Capacity        : 3.00
True Memory                    : 28.00 GB
SMT Threads                    : 2
Shared Processor Mode          : Enabled-Uncapped
Active Memory Sharing          : Disabled
Active Memory Expansion        : Disabled

System Resource Statistics:               Average           Min
Max
---------------------------              -----------       -----------
-----------
CPU Util (Phys. Processors)              0.93 [ 31%]       0.24 [  8%]
2.51 [ 84%]
Virtual Memory Size (MB)                 6146 [ 21%]       3119 [ 11%]
6924 [ 24%]
True Memory In-Use (MB)                  7398 [ 26%]       4332 [ 15%]
8197 [ 29%]
Pinned Memory (MB)                       1712 [  6%]       1710 [  6%]
1714 [  6%]
File Cache Size (MB)                     1237 [  4%]       1198 [  4%]
1259 [  4%]
Available Memory (MB)                   21654 [ 76%]      20877 [ 73%]
24682 [ 86%]

Active Memory Expansion Modeled Statistics:
-------------------------------------------
Modeled Expanded Memory Size   :  28.00 GB
Average Compression Ratio      :   2.37

Expansion    Modeled True      Modeled              CPU Usage
Factor       Memory Size       Memory Gain          Estimate
---------    -------------     ------------------   -----------
     1.04         27.00 GB         1.00 GB [  4%]   0.00 [  0%]
     1.20         23.25 GB         4.75 GB [ 20%]   0.00 [  0%]
     1.37         20.50 GB         7.50 GB [ 37%]   0.00 [  0%]
     1.53         18.25 GB         9.75 GB [ 53%]   0.00 [  0%]
     1.70         16.50 GB        11.50 GB [ 70%]   0.00 [  0%]
     1.87         15.00 GB        13.00 GB [ 87%]   0.00 [  0%]
     2.04         13.75 GB        14.25 GB [104%]   0.32 [ 11%]

Active Memory Expansion Recommendation:
---------------------------------------
The recommended AME configuration for this workload is to configure the LPAR
with a memory size of 13.75 GB and to configure a memory expansion factor
of 2.04.  This will result in a memory gain of 104%. With this
configuration, the estimated CPU usage due to AME is approximately 0.32
physical processors, and the estimated overall peak CPU resource required for
the LPAR is 2.83 physical processors.

NOTE: amepat's recommendations are based on the workload's utilization level
during the monitored period. If there is a change in the workload's utilization
level or a change in workload itself, amepat should be run again.

The modeled Active Memory Expansion CPU usage reported by amepat is just an
```

```
estimate.  The actual CPU usage used for Active Memory Expansion may be lower
or higher depending on the workload.
```

Amepat reports that for the same workload, the amount of memory assigned to the partition can be reduced from 28 Gb to 13.75 GB at the expense of using 0.32 additional physical processors. Thus, we could save 28 - 13.75 = 14.25 GB of memory on the target POWER7 system.

6.1.1 Migrating live logical partitions from POWER6 to POWER7

In this section, we migrate live logical partitions from POWER6 to POWER7. We perform the following steps:

1. We verify that both VIO servers and both mobile partitions have their clocks synchronized using the **lssyscfg** command (Example 6-3).

Example 6-3 Verifying VIO servers and partitions

```
hscroot@hmcdq1:~> lssyscfg -r lpar -m p550 --filter "lpar_names=VIOS_p550" -F
time_ref
1
hscroot@hmcdq1:~> lssyscfg -r lpar -m p550 --filter "lpar_names=p550was1" -F
time_ref
1
hscroot@hmcdq1:~> lssyscfg -r lpar -m p550 --filter "lpar_names=p550was3" -F
time_ref
1
hscroot@hmcdq1:~> lssyscfg -r lpar -m p7_1 --filter "lpar_names=vios_p7_1" -F
time_ref
1
```

2. We verify the current processor mode on the source system for both migrating partitions (Example 6-4).

Example 6-4 Verify the processor mode

```
hscroot@hmcdq1:~> lssyscfg -r lpar -m p550  --filter "lpar_names=p550was1"  -F
curr_lpar_proc_compat_mode
POWER6
hscroot@hmcdq1:~> lssyscfg -r lpar -m p550  --filter "lpar_names=p550was3"  -F
curr_lpar_proc_compat_mode
POWER6
```

3. We verify the current partition IDs (Example 6-5).

Example 6-5 Verify partition IDs

```
hscroot@hmcdq1:~> lssyscfg -r lpar -m p550 --filter "lpar_names=p550was3" -F
name lpar_id
p550was3 6
hscroot@hmcdq1:~> lssyscfg -r lpar -m p550 --filter "lpar_names=p550was1" -F
name lpar_id
p550was1 8
```

4. We verify the processor compatibility modes supported by the target system using the **lssyscfg** command:

```
hscroot@hmcdq1:~> lssyscfg -r sys -m p7_1 -F lpar_proc_compat_modes
"default,POWER6,POWER6+,POWER7"
```

5. We verify the existing SCSI and Ethernet adapters for both migrating partitions (Example 6-6).

Example 6-6 Verify existing SCSI and Ethernet adapters

```
hscroot@hmcdq1:~> lssyscfg -r prof -m p550  --filter "lpar_names=p550was1,
profile_names=profile" -F virtual_scsi_adapters
3/client/1/VIOS_p550/11/1
hscroot@hmcdq1:~> lssyscfg -r prof -m p550  --filter "lpar_names=p550was1,
profile_names=profile" -F virtual_eth_adapters
2/0/1//0/1/ETHERNET0
hscroot@hmcdq1:~> lssyscfg -r prof -m p550  --filter "lpar_names=p550was3,
profile_names=profile1" -F virtual_scsi_adapters
2/client/1/VIOS_p550/17/1
hscroot@hmcdq1:~> lssyscfg -r prof -m p550  --filter "lpar_names=p550was3,
profile_names=profile1" -F virtual_eth_adapters
4/0/1//0/1/ETHERNET0
```

Note that p550was1 partition is activated with a profile named *profile* and has a client SCSI adapter with and ID of 3, which is assigned to server SCSI adapter with an ID of 11 and a virtual Ethernet adapter with 2.

Also note that the p550was3 partition is activated with a profile named profile1 and has a client SCSI adapter with and ID of 2, which is assigned to a server SCSI adapter with an ID of 17 and a virtual Ethernet adapter with 4.

We perform the following steps:

1. We verify the possible and suggested mappings for migration (Example 6-7).

Example 6-7 Verify the mapping before migration

```
hscroot@hmcdq1:~> lslparmigr -r virtualio -m p550 -t p7_1 --filter
lpar_names=p550was1
"available_vios_virtual_slots=""vios_p7_1/2/6-19,21-24""",possible_virtual_scsi
_mappings=3/vios_p7_1/2,suggested_virtual_scsi_mappings=3/vios_p7_1/2/11,possib
le_virtual_fc_mappings=none,suggested_virtual_fc_mappings=none
hscroot@hmcdq1:~> lslparmigr -r virtualio -m p550 -t p7_1 --filter
lpar_names=p550was3
"available_vios_virtual_slots=""vios_p7_1/2/6-19,21-24""",possible_virtual_scsi
_mappings=2/vios_p7_1/2,suggested_virtual_scsi_mappings=2/vios_p7_1/2/17,possib
le_virtual_fc_mappings=none,suggested_virtual_fc_mappings=none
```

   The suggested_virtual_scsi_mappings=3/vios_p7_1/2/11 means that SCSI client adapter 3 will be assigned to SCSI server adapter ID 11 that belongs to vios_p7_1, which has partition ID 2.

2. We initiate the migration for both partitions simultaneously using the graphical interface.

3. We verify the number of pending migrations (Example 6-8).

Example 6-8 Verify the pending migrations

```
hscroot@hmcdq1:~> lslparmigr -r lpar -m p550
name=VIOS_p550,lpar_id=1,migration_state=Not Migrating
name=p550was1,lpar_id=8,migration_state=Not Migrating
name=p550was2,lpar_id=7,migration_state=Not Migrating
```

4. The partition AIX error log contains one entry stating that the partition ID has changed (Example 6-9).

Example 6-9 Notification messages in the AIX mobile partition error log

```
---------------------------------------------------------------------------
LABEL:          PARTID_CHANGE
IDENTIFIER:     60AFC9E5

Date/Time:       Wed Mar 31 04:03:37 EDT 2010
Sequence Number: 179
Machine Id:      00C2A9C04C00
Node Id:         p550was3
Class:           O
Type:            INFO
WPAR:            Global
Resource Name:   sys0

Description
Partition ID changed since last boot.

---------------------------------------------------------------------------
```

5. We verify the current processor mode on the destination system for both migrating partitions (Example 6-10).

Example 6-10 Verifying processor mode

```
hscroot@hmcdq1:~> lssyscfg -r lpar -m p7_1 --filter "lpar_names=p550was1"  -F
curr_lpar_proc_compat_mode
POWER6
hscroot@hmcdq1:~> lssyscfg -r lpar -m p7_1 --filter "lpar_names=p550was3"  -F
curr_lpar_proc_compat_mode
POWER6
```

6. We verify the new partition IDs (Example 6-11).

Example 6-11 Verifying the new partition ID

```
hscroot@hmcdq1:~> lssyscfg -r lpar -m p7_1 --filter "lpar_names=p550was3" -F
name lpar_id
p550was3 6
hscroot@hmcdq1:~> lssyscfg -r lpar -m p7_1 --filter "lpar_names=p550was1" -F
name lpar_id
p550was1 8
```

7. We verify the new SCSI and Ethernet adapters for both migrating partitions (Example 6-12).

Example 6-12 Verifying new SCSI and Ethernet adapters

```
hscroot@hmcdq1:~> lssyscfg -r prof -m p7_1 --filter "lpar_names=p550was1,
profile_names=profile" -F virtual_scsi_adapters
3/client/1/VIOS_p550/11/1
hscroot@hmcdq1:~> lssyscfg -r prof -m p7_1 --filter "lpar_names=p550was1,
profile_names=profile" -F virtual_eth_adapters
2/0/1//0/1/ETHERNET0
hscroot@hmcdq1:~> lssyscfg -r prof -m p7_1 --filter "lpar_names=p550was3,
profile_names=profile1" -F virtual_scsi_adapters
2/client/1/VIOS_p550/17/1
hscroot@hmcdq1:~> lssyscfg -r prof -m p7_1 --filter "lpar_names=p550was3,
profile_names=profile1" -F virtual_eth_adapters
4/0/1//0/1/ETHERNET0
```

6.1.2 Using amepat to assess memory usage after migration to POWER7

This section shows how to use the Active Memory Expansion features in the following environment:

- One logical partition named p550was1 initially hosted by an BM POWER6 server and migrated later to a POWER7 system
- One logical partition named p550was3 initially hosted by an BM POWER6 server and migrated later to a POWER7 system

Both partitions are now assigned three CPUs and 28 GB of memory to run the same typical Cognos workloads that last 60 minutes. We run the **amepat** tool on both partitions, p550was1 and p550was3, for 60 minutes to assess how memory and CPU are used after migration (Example 6-13 and Example 6-14 on page 147).

Example 6-13 Running the amepat tool on p550was1 after migration to POWER7

```
p550was1(root)/> amepat 60

Command Invoked              : amepat 60

Date/Time of invocation      : Tue Mar 23 18:57:48 EDT 2010
Total Monitored time         : 1 hrs 7 mins 48 secs
Total Samples Collected      : 9

System Configuration:
---------------------
Partition Name               : p550was1
Processor Implementation Mode : POWER6_in_P6_mode
Number Of Logical CPUs       : 6
Processor Entitled Capacity  : 3.00
Processor Max. Capacity      : 3.00
True Memory                  : 28.00 GB
SMT Threads                  : 2
Shared Processor Mode        : Enabled-Uncapped
Active Memory Sharing        : Disabled
Active Memory Expansion      : Disabled
```

```
System Resource Statistics:                  Average            Min
Max
---------------------------                -----------        -----------
-----------
CPU Util (Phys. Processors)                0.93 [ 31%]        0.24 [  8%]
2.51 [ 84%]
Virtual Memory Size (MB)                   6146 [ 21%]        3119 [ 11%]
6924 [ 24%]
True Memory In-Use (MB)                    7398 [ 26%]        4332 [ 15%]
8197 [ 29%]
Pinned Memory (MB)                         1712 [  6%]        1710 [  6%]
1714 [  6%]
File Cache Size (MB)                       1237 [  4%]        1198 [  4%]
1259 [  4%]
Available Memory (MB)                     21654 [ 76%]       20877 [ 73%]
24682 [ 86%]

Active Memory Expansion Modeled Statistics:
-------------------------------------------
Modeled Expanded Memory Size   :  28.00 GB
Average Compression Ratio      :   2.37

Expansion    Modeled True      Modeled              CPU Usage
Factor       Memory Size       Memory Gain          Estimate
---------    -------------     ------------------   -----------
     1.04         27.00 GB         1.00 GB [  4%]   0.00 [  0%]
     1.20         23.25 GB         4.75 GB [ 20%]   0.00 [  0%]
     1.37         20.50 GB         7.50 GB [ 37%]   0.00 [  0%]
     1.53         18.25 GB         9.75 GB [ 53%]   0.00 [  0%]
     1.70         16.50 GB        11.50 GB [ 70%]   0.00 [  0%]
     1.87         15.00 GB        13.00 GB [ 87%]   0.00 [  0%]
     2.04         13.75 GB        14.25 GB [104%]   0.32 [ 11%]

Active Memory Expansion Recommendation:
---------------------------------------
The recommended AME configuration for this workload is to configure the LPAR
with a memory size of 13.75 GB and to configure a memory expansion factor
of 2.04.  This will result in a memory gain of 104%. With this
configuration, the estimated CPU usage due to AME is approximately 0.32
physical processors, and the estimated overall peak CPU resource required for
the LPAR is 2.83 physical processors.

NOTE: amepat's recommendations are based on the workload's utilization level
during the monitored period. If there is a change in the workload's utilization
level or a change in workload itself, amepat should be run again.

The modeled Active Memory Expansion CPU usage reported by amepat is just an
estimate.  The actual CPU usage used for Active Memory Expansion may be lower
or higher depending on the workload.
```

Example 6-14 Running the amepat tool on p550was3 after migration to POWER7

```
p550was3(root)/> amepat 60

Command Invoked                : amepat 60
```

```
Date/Time of invocation       : Tue Mar 23 18:57:48 EDT 2010
Total Monitored time          : 1 hrs 7 mins 48 secs
Total Samples Collected       : 9

System Configuration:
---------------------
Partition Name                : p550was3
Processor Implementation Mode : POWER6_in_P6_mode
Number Of Logical CPUs        : 6
Processor Entitled Capacity   : 3.00
Processor Max. Capacity       : 3.00
True Memory                   : 28.00 GB
SMT Threads                   : 2
Shared Processor Mode         : Enabled-Uncapped
Active Memory Sharing         : Disabled
Active Memory Expansion       : Disabled

System Resource Statistics:              Average            Min
Max
---------------------------            -----------        -----------
-----------
CPU Util (Phys. Processors)            0.93 [ 31%]        0.24 [  8%]
2.51 [ 84%]
Virtual Memory Size (MB)               6146 [ 21%]        3119 [ 11%]
6924 [ 24%]
True Memory In-Use (MB)                7398 [ 26%]        4332 [ 15%]
8197 [ 29%]
Pinned Memory (MB)                     1712 [  6%]        1710 [  6%]
1714 [  6%]
File Cache Size (MB)                   1237 [  4%]        1198 [  4%]
1259 [  4%]
Available Memory (MB)                 21654 [ 76%]       20877 [ 73%]
24682 [ 86%]

Active Memory Expansion Modeled Statistics:
-------------------------------------------
Modeled Expanded Memory Size   :  28.00 GB
Average Compression Ratio      :   2.37

Expansion    Modeled True      Modeled              CPU Usage
Factor       Memory Size       Memory Gain          Estimate
---------    -------------     ------------------   -----------
     1.04         27.00 GB         1.00 GB [  4%]   0.00 [  0%]
     1.20         23.25 GB         4.75 GB [ 20%]   0.00 [  0%]
     1.37         20.50 GB         7.50 GB [ 37%]   0.00 [  0%]
     1.53         18.25 GB         9.75 GB [ 53%]   0.00 [  0%]
     1.70         16.50 GB        11.50 GB [ 70%]   0.00 [  0%]
     1.87         15.00 GB        13.00 GB [ 87%]   0.00 [  0%]
     2.04         13.75 GB        14.25 GB [104%]   0.32 [ 11%]

Active Memory Expansion Recommendation:
---------------------------------------
The recommended AME configuration for this workload is to configure the LPAR
with a memory size of 13.75 GB and to configure a memory expansion factor
```

```
of 2.04.  This will result in a memory gain of 104%. With this
configuration, the estimated CPU usage due to AME is approximately 0.32
physical processors, and the estimated overall peak CPU resource required for
the LPAR is 2.83 physical processors.

NOTE: amepat's recommendations are based on the workload's utilization level
during the monitored period. If there is a change in the workload's utilization
level or a change in workload itself, amepat should be run again.

The modeled Active Memory Expansion CPU usage reported by amepat is just an
estimate.  The actual CPU usage used for Active Memory Expansion may be lower
or higher depending on the workload.
```

6.2 Implementing Live Partition Mobility and Active Memory Sharing with Cognos

This section shows how to use Live Partition Mobility and Active Memory Sharing features in an environment based on IBM POWER technology-based servers managed by a HMC.

We use logical partitions running typical Cognos workloads that are migrated between two Power Systems named p550 and p570. You can use the steps depicted in this scenario for active migration of partitions between any two POWER6 or POWER7 systems.

The scenario contains the following steps:

1. We list the shared memory pool existing in the source system (Example 6-15).

Example 6-15   Listing the shared memory pools

```
hscroot@hmcdq1:~> lshwres -m p550 -r mempool
curr_pool_mem=8192,curr_avail_pool_mem=7936,curr_max_pool_mem=8192,pend_pool_me
m=8192,pend_avail_pool_mem=7936,pend_max_pool_mem=8192,sys_firmware_pool_mem=25
6,paging_vios_names=VIOS_p550,paging_vios_ids=1
```

2. We verify the existing paging devices on the source system. There are three paging devices, one of which is active and used by a logical partition named p550was2. The partition has ID 7 and the device used for paging is hdisk19 (Example 6-16).

Example 6-16   Verifying existing paging devices

```
hscroot@hmcdq1:~> lshwres -m  p570 -r mempool --rsubtype pgdev
device_name=hdisk16,paging_vios_name=VIOS_p570,paging_vios_id=1,size=6144,type=
phys,state=Inactive,phys_loc=U789D.001.DQDZCWY-P1-C6-T1-W201300A0B811A662-L4000
000000000,is_redundant=0,lpar_id=none
device_name=hdisk18,paging_vios_name=VIOS_p570,paging_vios_id=1,size=6144,type=
phys,state=Inactive,phys_loc=U789D.001.DQDZCWY-P1-C6-T1-W201300A0B811A662-L6000
000000000,is_redundant=0,lpar_id=none
device_name=hdisk19,paging_vios_name=VIOS_p570,paging_vios_id=1,size=6144,type=
phys,state=Active,phys_loc=U789D.001.DQDZCWY-P1-C6-T1-W201300A0B811A662-L700000
0000000,is_redundant=0,lpar_name=p550was2,lpar_id=7
```

3. We can collect the information regarding paging devices from VIOS level as well (Example 6-17).

Example 6-17 Collecting information on paging devices via the VIO server

```
$ uname -n
viosp570
$ lsdev -dev vrmpage*
name            status      description
vrmpage0        Defined     Paging Device - Disk
vrmpage1        Defined     Paging Device - Disk
vrmpage2        Available   Paging Device - Disk
$ lsmap -ams -all
Paging          Stream ID          Client ID
--------------- ------------------ ---------------
vrmpage0        0x20000122127eaf75 0

Status              inactive
Redundancy Usage    no
Backing device      hdisk16
Pool ID             0
VASI
Pager
VBSD

Paging          Stream ID          Client ID
--------------- ------------------ ---------------
vrmpage1        0x20000122127eaf76 0

Status              inactive
Redundancy Usage    no
Backing device      hdisk18
Pool ID             0
VASI
Pager
VBSD

Paging          Stream ID          Client ID
--------------- ------------------ ---------------
vrmpage2        0x20000122127eaf77 7

Status              active
Redundancy Usage    no
Backing device      hdisk19
Pool ID             0
VASI                vasi4
Pager               pager4
VBSD                vbsd4

listing attributes of a paging device active

$ lsdev -dev vrmpage2 -attr
attribute       value                  description
user_settable

LogicalUnitAddr 0x8300000000000000     Logical Unit Address         False
```

```
aix_tdev        hdisk19                Target Device Name          False
client_reserve  no                     Client Reserve              True
partition_id    7                      Client Partition ID         False
redundant_usage no                     Redundant Usage             True
vasi_drc_name   U9117.MMA.102A9C0-V1-C6 VASI DRC Name              True
vrm_state       active                 Virtual Real Memory State   True
vtd_handle      0x20000122127eaf77     Virtual Target Device Handle False
```

4. We verify the paging devices that exist on the target system. There are four paging devices, none of which is being used (Example 6-18).

Example 6-18 Verifying the paging devices in the target system

```
hscroot@hmcdq1:~> lshwres -m p550 -r mempool --rsubtype pgdev
device_name=hdisk2,paging_vios_name=VIOS_p550,paging_vios_id=1,size=286102,type=
phys,state=Inactive,phys_loc=U78A0.001.DNWGLVC-P2-D3,is_redundant=0,lpar_id=none
device_name=hdisk3,paging_vios_name=VIOS_p550,paging_vios_id=1,size=286102,type=
phys,state=Inactive,phys_loc=U78A0.001.DNWGLVC-P2-D6,is_redundant=0,lpar_id=none
device_name=hdisk35,paging_vios_name=VIOS_p550,paging_vios_id=1,size=30720,type
=phys,state=Inactive,phys_loc=U78A0.001.DNWGLVC-P1-C5-T2-W201300A0B811A662-L190
00000000000,is_redundant=0,lpar_id=none
device_name=hdisk36,paging_vios_name=VIOS_p550,paging_vios_id=1,size=30720,type
=phys,state=Inactive,phys_loc=U78A0.001.DNWGLVC-P1-C5-T2-W201300A0B811A662-L1A0
00000000000,is_redundant=0,lpar_id=none
```

5. We list existing partition profiles for the partition named p550was2 (Example 6-19).

Example 6-19 Listing the existing partition profiles

```
hscroot@hmcdq1:~> lssyscfg -r prof -m p570 --filter "lpar_names=p550was2" -F
name
profile
ams_profile
```

6. We list the active partition profile for the partition named p550was2 (Example 6-20).

Example 6-20 Listing the active partition profile

```
hscroot@hmcdq1:~> lssyscfg -r lpar -m p570 --filter "lpar_names=p550was2" -F
curr_profile
ams_profile
```

7. We list memory settings of the active partition profile (Example 6-21).

Example 6-21 Listing memory settings

```
hscroot@hmcdq1:~> lssyscfg -r prof -m p570 --filter
"lpar_names=p550was2,profile_names=ams_profile" -F lpar_name min_mem
desired_mem max_mem mem_mode --header
lpar_name min_mem desired_mem max_mem mem_mode
p550was2 1024 4096 4096 shared
```

8. We list the migration status of all partitions on the source system. The migration has not been started yet (Example 6-22).

Example 6-22 Listing the migration status of the partitions

```
hscroot@hmcdq1:~> lslparmigr -r lpar -m p570
name=p550was2,lpar_id=7,migration_state=Not Migrating
name=cognos,lpar_id=5,migration_state=Not Migrating
name=was,lpar_id=4,migration_state=Not Migrating
name=db2,lpar_id=3,migration_state=Not Migrating
name=http,lpar_id=2,migration_state=Not Migrating
name=VIOS_p570,lpar_id=1,migration_state=Not Migrating
```

9. We list the potential virtual SCSI adapters that are assigned following the migration (Example 6-23).

Example 6-23 Listing the potential virtual SCSI adapters

```
hscroot@hmcdq1:~> lslparmigr -r virtualio -m p570 -t p550 --filter
lpar_names=p550was2 -F suggested_virtual_scsi_mappings
3/VIOS_p550/1/18
```

Tip: It is always a good practice to verify the suggested mappings. It allows you to keep track of slot numbers when the Power systems contain multiple partitions.

10. We validate the migration and receive only an informational warning related to the creation of a migration profile (Example 6-24).

Example 6-24 Validating the migration

```
hscroot@hmcdq1:~> migrlpar -o v -m p570 -t p550 -p p550was2

Warnings:
HSCLA295 As part of the migration process, the HMC will create a new migration
profile containing the partition's current state.  The default is to use the
current profile, which will replace the existing definition of this profile.
While this works for most scenarios, other options are possible.  You may
specify a different existing profile, which would be replaced with the current
partition definition, or you may specify a new profile to save the current
partition state.
```

11. We decide to create a new profile named ams_profile_p550 that includes the intended settings for the migrating partition on the target system (Example 6-25).

Example 6-25 Creating a new profile

```
hscroot@hmcdq1:~> migrlpar -o v -m p570 -t p550 -p p550was2 -n ams_profile_p550
hscroot@hmcdq1:~> echo $?
0
```

12. We migrate the partition (Example 6-26).

Example 6-26 Migrating the partition

```
hscroot@hmcdq1:~> migrlpar -o m -m p570 -t p550 -p p550was2 -n ams_profile_p550
hscroot@hmcdq1:~> echo $?
0
```

13. During the migration, you can check the migration status using the **lslparmigr -r lpar -m p570 -F name migration_state** command.

   The migration status has one of the following values:

   – Not Migrating
   – Migration Starting
   – Migration In Progress
   – Migration Complete

1. We verify whether migration completed successfully (Example 6-27).

Example 6-27 Verifying the migration

```
hscroot@hmcdq1:~> lslparmigr -r lpar -m p570
name=p550was2,lpar_id=7,migration_state=Migration
Complete,migration_type=active,dest_sys_name=p550,dest_lpar_id=2,source_msp_na
me=VIOS_p570,source_msp_id=1,dest_msp_name=VIOS_p550,dest_msp_id=1,remote_manag
er=unavailable,remote_user=unavailable,bytes_transmitted=0,bytes_remaining=7780
3520
name=cognos,lpar_id=5,migration_state=Not Migrating
name=was,lpar_id=4,migration_state=Not Migrating
name=db2,lpar_id=3,migration_state=Not Migrating
name=http,lpar_id=2,migration_state=Not Migrating
name=VIOS_p570,lpar_id=1,migration_state=Not Migrating
```

2. We verify the status and the ID of the migrating partition on the target system. The partition had partition ID 7 on the source system. Since on the target system this ID is already assigned, the migrating partition had to be assigned a new ID of 2 (Example 6-28).

Example 6-28 Verifying the status of the migrated partition

```
hscroot@hmcdq1:~> lssyscfg -r lpar -m p550 -F name lpar_id state
p550was1 7 Running
p550was3 5 Running
VIOS_p550 1 Running
p550was2 2 Running
```

3. We verify the list of profiles associated with the migrating partition, and see that the list has been updated to include the profile defined during migration (Example 6-29).

Example 6-29 Verifying the profiles listings

```
hscroot@hmcdq1:~> lssyscfg -r prof -m p550 --filter "lpar_names=p550was2" -F
name
profile
ams_profile
ams_profile_p550
```

4. We verify that the profile defined during the migration has been activated on the target system (Example 6-30).

Example 6-30 Verifying that the profile defined is active

```
hscroot@hmcdq1:~> lssyscfg -r lpar -m p550 --filter "lpar_names=p550was2" -F
curr_profile
ams_profile_p550
```

5. We list memory parameters of the newly created profile (Example 6-31_.

Example 6-31 Listing the memory parameters of the new profile

```
hscroot@hmcdq1:~> lssyscfg -r prof -m p550 --filter
"lpar_names=p550was2,profile_names=ams_profile_p550" -F lpar_name min_mem
desired_mem max_mem mem_mode --header
lpar_name min_mem desired_mem max_mem mem_mode
p550was2 1024 4096 4096 shared
```

6. We list the paging devices on the target system, and see that the first of the smallest disks that can meet partition memory requirements has been selected (Example 6-32).

Example 6-32 Listing paging devices on the target system

```
hscroot@hmcdq1:~> lshwres -m p550 -r mempool --rsubtype pgdev
device_name=hdisk2,paging_vios_name=VIOS_p550,paging_vios_id=1,size=286102,type=
phys,state=Inactive,phys_loc=U78A0.001.DNWGLVC-P2-D3,is_redundant=0,lpar_id=none
device_name=hdisk3,paging_vios_name=VIOS_p550,paging_vios_id=1,size=286102,type=
phys,state=Inactive,phys_loc=U78A0.001.DNWGLVC-P2-D6,is_redundant=0,lpar_id=none
device_name=hdisk35,paging_vios_name=VIOS_p550,paging_vios_id=1,size=30720,typ
e=phys,state=Active,phys_loc=U78A0.001.DNWGLVC-P1-C5-T2-W201300A0B811A662-L1900
0000000000,is_redundant=0,lpar_name=p550was2,lpar_id=2
device_name=hdisk36,paging_vios_name=VIOS_p550,paging_vios_id=1,size=30720,type
=phys,state=Inactive,phys_loc=U78A0.001.DNWGLVC-P1-C5-T2-W201300A0B811A662-L1A0
00000000000,is_redundant=0,lpar_id=none
```

7. We verify the virtual SCSI adapters of the new profile against the estimation made prior to migration (Example 6-33).

Example 6-33 Verifying the virtual SCSI adapters in the new profile

```
hscroot@hmcdq1:~> lssyscfg -r prof -m p550  --filter "lpar_names=p550was2,
profile_names=ams_profile_p550" -F virtual_scsi_adapters
3/client/1/VIOS_p550/18/1
```

8. We verify that the paging device used on the source system has been released (Example 6-34).

Example 6-34 Verifying the paging device

```
hscroot@hmcdq1:~> lshwres -m p570 -r mempool --rsubtype pgdev
device_name=hdisk16,paging_vios_name=VIOS_p570,paging_vios_id=1,size=6144,type=
phys,state=Inactive,phys_loc=U789D.001.DQDZCWY-P1-C6-T1-W201300A0B811A662-L4000
000000000,is_redundant=0,lpar_id=none
device_name=hdisk18,paging_vios_name=VIOS_p570,paging_vios_id=1,size=6144,type=
phys,state=Inactive,phys_loc=U789D.001.DQDZCWY-P1-C6-T1-W201300A0B811A662-L6000
000000000,is_redundant=0,lpar_id=none
device_name=hdisk19,paging_vios_name=VIOS_p570,paging_vios_id=1,size=6144,type=
phys,state=Inactive,phys_loc=U789D.001.DQDZCWY-P1-C6-T1-W201300A0B811A662-L7000
000000000,is_redundant=0,lpar_id=none
```

This chapter showed our experiences in a test environment while using IBM Power Systems Live Partition Mobility and Active Memory Expansion in conjuction with the IBM Cognos 8 Business Intelligence solution. Chapter 7, “Tips and tricks” on page 155, provides hints and tips discovered while implementing and testing the business intelligence solution.

7

Tips and tricks

This chapter provides useful tips from our experience with the deployment and testing of the solution. This chapter discusses the following topics:

7.1 WebSphere application server profile is corrupted

After Cognos is installed with the WebSphere Application Server, the service starts when the WebSphere server starts. If for some reason the WebSphere application server profile gets corrupted, then both WebSphere Application Server and the Cognos server cannot be started. There are various reasons for a profile corruption. This can be because of a file system failure or from accidentally deleting important configuration or executable files.

The answer to this issue is to either create a new profile or restore the existing profile. With a new profile, the WebSphere Application Server starts in another port. After the dispatcher starts in another port, the entire Cognos deployment needs to be reconfigured accordingly. Regenerate the EAR deployment files and redeploy them. This approach takes much more effort and more time.

Another option is to restore the same profile and to start the Cognos service with the previous port number. The corrupted profile is deleted and a new profile is created with the same name on the same path. Example 7-1 shows the commands.

Example 7-1 Commands to restore the corrupted WebSphere Application Server profile

```
manageprofiles.sh -delete –templatePath
                    /usr/IBM/WebSphere/AppServer/profileTemplates/default
                    -profileName AppSrv01 –profilePath
                    /usr/IBM/WebSphere/AppServer/profiles/AppSrv01

manageprofiles.sh -create –templatePath
                    /usr/IBM/WebSphere/AppServer/profileTemplates/default
                    -profileName AppSrv01 –profilePath
                    /usr/IBM/WebSphere/AppServer/profiles/AppSrv01
```

Verify that the WebSphere Application Server is listening on the previously assigned port. After the profile is restored, applications are installed again. The Cognos deployment is done through the EAR file, which is deployed as an application. The EAR file creates and deploys processes that were described in previous sections.

7.2 Content storage or query database log grows and occupies entire file system

The DB2 database log can be configured in several ways. It can be configured to retain all different transaction details. But in this approach, the DB2 log grows with every transaction. At some point, it can even occupy the entire file system space. Also, a larger log has an impact on DB2 performance. To resolve this issue, DB2 tuning can be performed.

7.3 Checking the disk space

As a good practice, periodically monitor file system disk space over time. Sometimes due to several reasons, file system usage may reach up to 100% (which means that there is no more space left to store new files or data). Applications show unpredictable behavior if critical file systems, such as /usr, /, or /home, reach 100% utilization.

The AIX operating system provides tools called df and du to find the file system utilization. The **df** command shows the usage of any file system (Example 7-2). In this example, we find that 78% of the /usr file system space is occupied.

Example 7-2 Output from df command

```
cognos(root)/> df -g
Filesystem    GB blocks      Free %Used    Iused %Iused Mounted on
/dev/hd4           2.00      1.13   44%    15591     6% /
/dev/hd2           7.00      1.56   78%    87876    18% /usr
/dev/hd9var        1.00      0.75   26%     8528     5% /var
/dev/hd3           1.00      0.99    1%       32     1% /tmp
/dev/hd1           1.00      0.99    2%      152     1% /home
/dev/hd11admin      0.12      0.12    1%        5     1% /admin
/proc                 -         -    -         -     -  /proc
/dev/hd10opt       2.00      1.33   34%    12150     4% /opt
/dev/livedump      0.25      0.25    1%        4     1% /var/adm/ras/livedump
/dev/lv00          1.91      0.46   77%       26     1% /download
/dev/lv01          2.39      0.83   66%     1754     1% /INST
192.168.100.199:/PW9408/nfs     23.88      9.97   59%    25315     2% /mnt
cognos(root)/>
```

The **du** command is used to fetch the size of each file from the specified directory. In Example 7-3, the -k flag is used to get the file size by the unit in Kilobytes.

Example 7-3 du command to find size of each file from /tmp

```
cognos(root)/> du -k /tmp/
0       /tmp/.X11-unix
1876    /tmp/.oslevel.datafiles
0       /tmp/.workdir.205030.487482_1
0       /tmp/.workdir.643302.647390_1
8       /tmp/javasharedresources
2116    /tmp/
```

At a certain time, a directory may contain many subdirectories. Also, each subdirectory can have more subdirectories and files inside. In such cases, instead of each file, we can find the size of each subdirectory with all the files and sub-directories inside it. This is very useful to find the path that contains a very big file such as a log file. In Example 7-4, we find that the size of the /usr/IBM/HTTServer subdirectory is 2231.09 megabytes. After looking at several subdirectories in this path, the size of the /usr/IBM/HTTPServer/logs/ is 1841.01 megabytes, so this particular directory contains a large file size.

Example 7-4 du command to find the path with a large file

```
cognos(root)/> du -sm /usr/IBM/*
2231.09 /usr/IBM/HTTPServer/
cognos(root)/> du -sm /usr/IBM/HTTPServer/*
216.43  /usr/IBM/HTTPServer/Plugins/
2.05    /usr/IBM/HTTPServer/bin/
0.20    /usr/IBM/HTTPServer/build/
0.00    /usr/IBM/HTTPServer/cgi-bin/
2.32    /usr/IBM/HTTPServer/codeset/
0.22    /usr/IBM/HTTPServer/conf/
0.20    /usr/IBM/HTTPServer/error/
0.05    /usr/IBM/HTTPServer/example_module/
14.46   /usr/IBM/HTTPServer/gsk7/
```

```
0.26    /usr/IBM/HTTPServer/htdocs/
0.93    /usr/IBM/HTTPServer/icons/
1.22    /usr/IBM/HTTPServer/include/
125.27  /usr/IBM/HTTPServer/java/
1.05    /usr/IBM/HTTPServer/lafiles/
3.46    /usr/IBM/HTTPServer/lib/
0.57    /usr/IBM/HTTPServer/license/
1841.01 /usr/IBM/HTTPServer/logs/
0.07    /usr/IBM/HTTPServer/man/
3.67    /usr/IBM/HTTPServer/modules/
1.12    /usr/IBM/HTTPServer/properties/
0.89    /usr/IBM/HTTPServer/readme/
15.64   /usr/IBM/HTTPServer/uninstall/
0.00    /usr/IBM/HTTPServer/version.signature
cognos(root)/>
```

7.4 DB2 installation using the response file

The DB2 server and client can be installed using a GUI-based installer or can be done from the command using a silent approach. The **db2setup** is the installer and can be used with a response file. The command is **db2setup -r <response file>**. Example 7-5 shows a sample response file for DB2 server installation.

Example 7-5 Sample response file DB2 server installation

```
PROD                           = ENTERPRISE_SERVER_EDITION
FILE                           = /opt/ibm/db2/V9.5
LIC_AGREEMENT                  = ACCEPT
INTERACTIVE                    = NONE            ** NONE, YES, MACHINE
INSTALL_TYPE                   = TYPICAL         ** TYPICAL, COMPACT, CUSTOM
LANG                           = EN              ** English (en_US)
LANG                           = EN              ** Portuguese - Brazil (pt_BR)
INSTANCE                       = db2inst1
db2inst1.NAME                  = db2inst1        ** real name of the instance
db2inst1.GROUP_NAME            = db2iadm1        ** char(30) no spaces
db2inst1.HOME_DIRECTORY        = /home/db2inst1  ** char(64) no spaces. Valid for
root install only
db2inst1.PASSWORD               = root ** char(8). Valid for root install only
db2inst1.AUTOSTART              = YES            ** YES or NO
db2inst1.START_DURING_INSTALL   = YES        ** YES or NO
db2inst1.SVCENAME               = db2c_db2inst1   ** BLANK or char(14). Reserved
for root install only
db2inst1.PORT_NUMBER            = 50000          ** 1024 - 65535, Reserved for
root install only
db2inst1.FCM_PORT_NUMBER        = 60000          ** 1024 - 65535, the beginning
port number. The ports i
n range [FCM_PORT_NUMBER, FCM_PORT_NUMBER+MAX_LOGICAL_NODES] is reserved during
install. Valid for root i
nstall only.
db2inst1.FENCED_USERNAME        = db2fenc1 ** char(8) no spaces, no upper case
letter
db2inst1.FENCED_GROUP_NAME      = db2fadm1 ** char(30)  no spaces
DAS_USERNAME                   = db2das ** char(8)no spaces, no upper case letter
DAS_GROUP_NAME                  = db2asgrp ** char(30)  no spaces
```

```
DAS_HOME_DIRECTORY               = /home/db2das ** char(64) no spaces
DAS_PASSWORD                     = ** char(8)
DB2COMM                          = TCPIP ** BLANK or TCPIP
```

Example 7-6 shows a sample response file for a DB2 client installation.

Example 7-6 Sample response file for a DB2 client installation

```
PROD                           = CLIENT
FILE                           = /opt/ibm/db2/V9.7
LIC_AGREEMENT                  = ACCEPT
INTERACTIVE                    = NONE
INSTALL_TYPE                   = TYPICAL
LANG                           = EN
LANG                           = EN
INSTANCE                       = db2inst1
db2inst1.NAME                  = db2inst1
db2inst1.GROUP_NAME            = db2iadm1
db2inst1.HOME_DIRECTORY          = /home/db2inst1
db2inst1.PASSWORD                = root
db2inst1.START_DURING_INSTALL    = YES
```

7.5 Starting a NFS-based WPAR from various LPARs

A WPAR created on an NFS file system can be started from several LPARs. However, only one instance of the WPAR can be started at a time. The prerequisites are:

- The NFS server is mounted and accessible from all the LPARs where the WPARs are started.
- All the LPARs have the same operating system level.

For example, a WPAR named wpar_cm is created on an LPAR named lpar1. wpar_cm is created on a NFS path /mnt/wpar_cm. The same WPAR is started from another LPAR named lpar2, which also has access to the same NFS server and path. One option is live application mobility, as explained in previous chapters. The second option is using the **mkwpar** command with the -p option:

1. On lpar1, stop the WPAR using the **stopwpar** command.
2. Mount the desired NFS file system on the second LPAR (lpar2).
3. Check and update the `/etc/filesystems` with the mount group details for the WPAR (Example 7-7).

   Example 7-7 /etc/filesystems configuration for WPAR

```
/mnt_wpar/wpar_cm:
        dev             = /wpar_nfs/wpar_cm
        vfs             = nfs
        nodename        = 192.168.100.199
        mount           = false
        type            = wpar_cm
        options         = bg,intr
        account         = false

/mnt_wpar/wpar_cm/home:
```

```
        dev             = /wpar_nfs/wpar_cm/home
        vfs             = nfs
        nodename        = 192.168.100.199
        mount           = false
        type            = wpar_cm
        options         = bg,intr
        account         = false

/mnt_wpar/wpar_cm/opt:
        dev             = /wpar_nfs/wpar_cm/opt
        vfs             = nfs
        nodename        = 192.168.100.199
        mount           = false
        type            = wpar_cm
        options         = rw
        account         = false

/mnt_wpar/wpar_cm/proc:
        dev             = /proc
        vfs             = namefs
        mount           = false
        type            = wpar_cm
        options         = rw
        account         = false
/mnt_wpar/wpar_cm/tmp:
        dev             = /wpar_nfs/wpar_cm/tmp
        vfs             = nfs
        nodename        = 192.168.100.199
        mount           = false
        type            = wpar_cm
        options         = bg,intr
        account         = false
/mnt_wpar/wpar_cm/usr:
        dev             = /wpar_nfs/wpar_cm/usr
        vfs             = nfs
        nodename        = 192.168.100.199
        mount           = false
        type            = wpar_cm
        options         = rw
        account         = false

/mnt_wpar/wpar_cm/var:
        dev             = /wpar_nfs/wpar_cm/var
        vfs             = nfs
        nodename        = 192.168.100.199
        mount           = false
        type            = wpar_cm
        options         = bg,intr
        account         = false
```

4. On lpar2, create the wpar_cm using the **mkwpar -p wpar_cm** command.

In Cognos

Before trying the following settings, refer to 3.3, “Affinity connection requests” on page 41 to understand the concepts of affinity connections.

- Maximum report services processes

  Based on Cognos testing with Cognos 8, it may be advantageous to reset the maximum number of interactive services to 2 per processor (possibly 3 if you have fast processors).

  This recommended value is based on the number of processors (CPUs) available on your system. Therefore, a system with 16 processors has this value set as a starting point in the range of 32 (2 process x 16 processors) to 48 (3 processes x 16 processors).

- Queue time limit

  The queue time limit refers to the maximum amount of time that a request waits for a report service connection when all connections are in use. The request is queued until a connection is available or up to the configured number of seconds dictated by the queue time limit setting. If no connection is available within the configured time limit, the user is notified that the request cannot be completed because no resources are available within the given time constraints. The default is 240 seconds.

  Our lab testing of Cognos 8 indicated that report servers are optimized when the requests are spread across multiple processes not concentrated all on a single process. More processes with fewer connections performs better than a small number of report processes with many connections. If the system is not being fully utilized, add more processes before adding additional connections within a process.

7.6 Tunables recommended from our experiences

The section discusses tunables that can help the systems’ performance drawn from experiences and behaviors seen during our testing.

7.6.1 In AIX

In this section we discuss the tunables for AIX.

Increasing maximum number of processes allowed per user

The default value set in AIX at around 512 or less is low, considering the HTTP server, for example. Each request coming to the server is run by a process. An HTTP server can receive thousands of requests at the same time. Once the maximum number of processes is reached, subsequent requests fail. Check for the current value setting by using the **lsattr** command, then issue the **chdev** command to change the value (Example 7-8).

Example 7-8 Command checking maximum number of processes allowed per user

```
cognos(root)/> lsattr -El sys0 -a maxuproc
maxuproc 10000 Maximum number of PROCESSES allowed per user True
cognos(root)/> chdev -l sys0 -a maxuproc=50000
sys0 changed
cognos(root)/> lsattr -El sys0 -a maxuproc
maxuproc 50000 Maximum number of PROCESSES allowed per user True
```

The **smitty** command can also be used for listing and changing maxuproc or other attribute values (Figure 7-1).

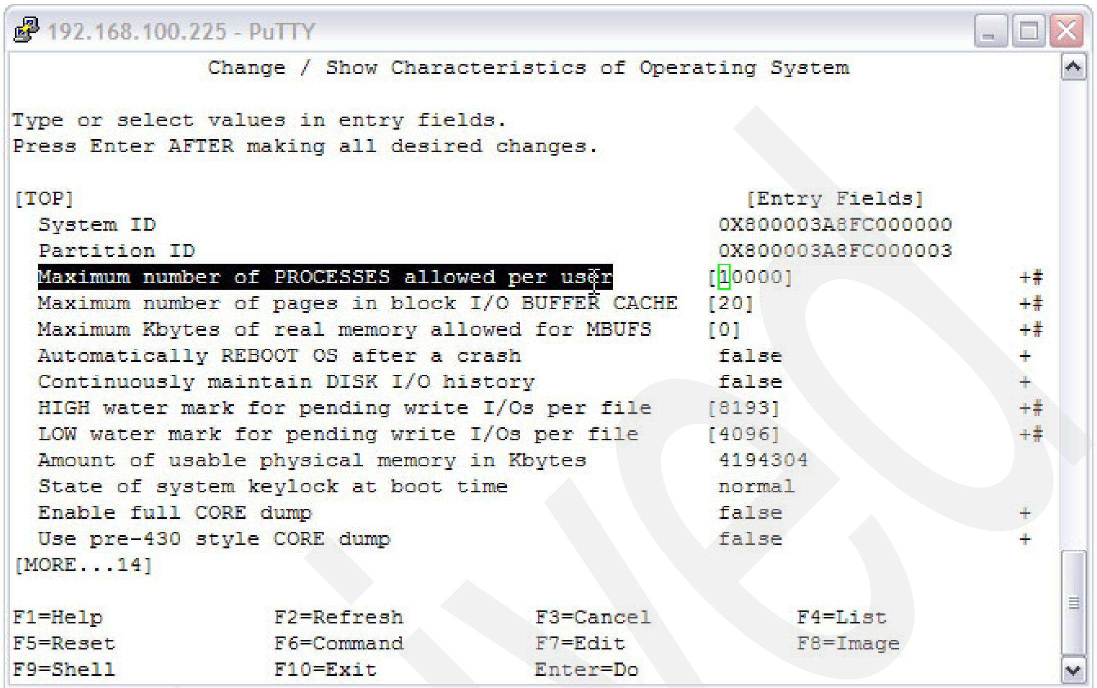


Figure 7-1 Changing AIX system attribute for maximum number of processes

7.6.2 In DB2

In this section we dicsuss tunables for DB2.

Using large buffer pool and temporary user tablespace

For deploying a database that contains large tables and data sizes, a large buffer pool and a temporary user tablespace may be required. These are defined when the Cognos content store database is created. To create buffer pools, connect to the Cognos database, and then run the DB2 command shown in Example 7-9.

Example 7-9 Create bufferpool with 32 K PAGESIZE

```
CREATE BUFFERPOOL COG32KBP IMMEDIATE SIZE 250 PAGESIZE 32 K
```

Then the 32 K BUFFERPOOL can be used to create the user temporary tablespace (Example 7-10).

Example 7-10 Creating the user temporary space

```
CREATE USER TEMPORARY TABLESPACE COGUSRTMP
   IN DATABASE PARTITION GROUP IBMDEFAULTGROUP PAGESIZE 4 K MANAGED BY SYSTEM
   USING ('D:\DB2\NODE0000\COGNOS\COGUSRTMP01') EXTENTSIZE 8 PREFETCHSIZE 8
   OVERHEAD 10.5 TRANSFERRATE 0.33 BUFFERPOOL COG32KBP
@
COMMENT ON TABLESPACE COGUSRTMP IS 'Cognos User Temp'
```

Setting max database connections

Depending on the number of queries to the databases, the maximum number of database connections can be set accordingly using the DB2 command shown in Example 7-11.

Example 7-11 Setting maximal database connections

```
update DBM cfg using MAX_CONNECTIONS 2500
   MAX_COORDAGENTS 2500
```

7.6.3 Tuning to improve HTTP server performance

The HTTP server receives several requests for reports. The higher the number of users, the more hits the server receives. The HTTP server is configured to handle a higher number of HTTP requests. The configuration file for the HTTP server is `httpd.conf`, which contains various tuning parameters. ServerLimit, MaxClients, and ThreadsPerChild are three parameters controlling the HTTP server limit to receive concurrent requests. The HTTP server starts a daemon called httpd, which is a root-owned process. httpd starts a number of child processes, and each of these process starts a number of threads.

- ThreadsPerChild is the limit of the number of threads that each server spawns per child process.
- ServerLimit is the hard limit of the number of active child processes from the HTTP server. Set this parameter to be higher than MaxClients.
- MaxClients is the maximum number of clients that can be served by all processes and threads. Set MaxClients to be low than ServerLimit multiplied by ThreadPerChild.

7.7 Tips for Tivoli Directory Server (TDS)

This section provides suggestions for dealing with certain behaviors encountered while deploying TDS.

7.7.1 An alternative way to installing TDS V6.2

Besides the installp images, the TDS software package also provides a Java program to guide the user through the installation steps. Example 7-12 shows the **`install_tds.bin`** command.

Example 7-12 Installing TDS V6.2 using Java program

```
# /INST/tdsV6.2/tds/install_tds.bin
```

install_tds.bin requires XWindows to run. Once started, enter required values in the fields and follow the steps of the InstallShield Wizard (Figure 7-2).

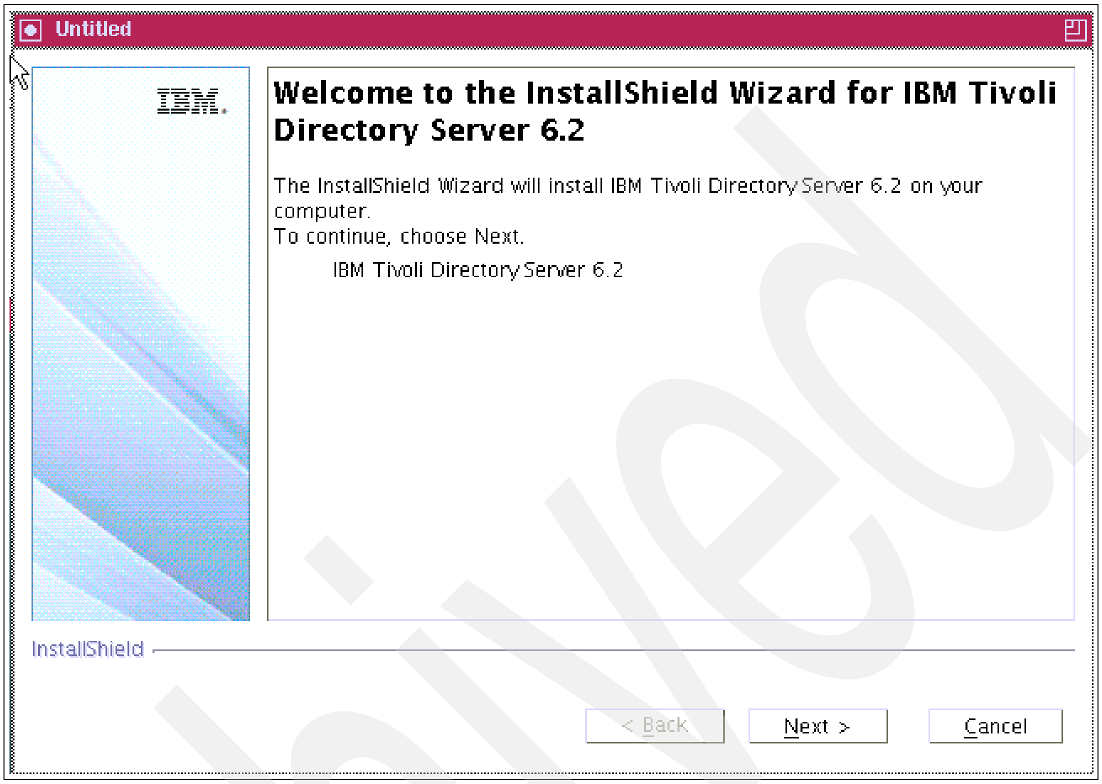


Figure 7-2 Starting the InstallShield Wizard

7.7.2 The default directory server instance can not be determined

If the **idsxcfg** command fails with the following error message, it is because the default instance does not exist:

```
GLPCFG100E The default directory server instance could not be determined. Use the
-I option to specify the instance, or set the IDS_LDAP_INSTANCE environment
variable before launching idsxcfg.
```

The problem may have occurred when the Tivoli Directory Server was installed, but was not able to create the default instance.

Check list

To help determine why the default directory server instance cannot be determined:

1. TDS requires a local database server. A remote database is not supported.
2. If there is a DB2 instance locally, it may be just a client instance.
3. Check the TDS install log for errors.

7.7.3 Creating the database instance for TDS V6.2

If the installation fails to create the default instance, it can be due to the database for the instance failing to create. The configuration tool has an option to create the database. Start **idsxcfg** and click the configure database menu option under the database tasks in the explorer panel.

Note: The yellow triangle with the exclamation mark indicates that an action is needed.

On the right-hand panel, enter the instance name, password, and database name (Example 7-13).

Example 7-13 Creating a database instance

```
Instance name: dsrdbm01
Password: *******
Database ma,e” dsrdbm01
```

After the instance is created successfully, the exclamation mark goes away and more menu options are displayed under the database tasks (Figure 7-3).

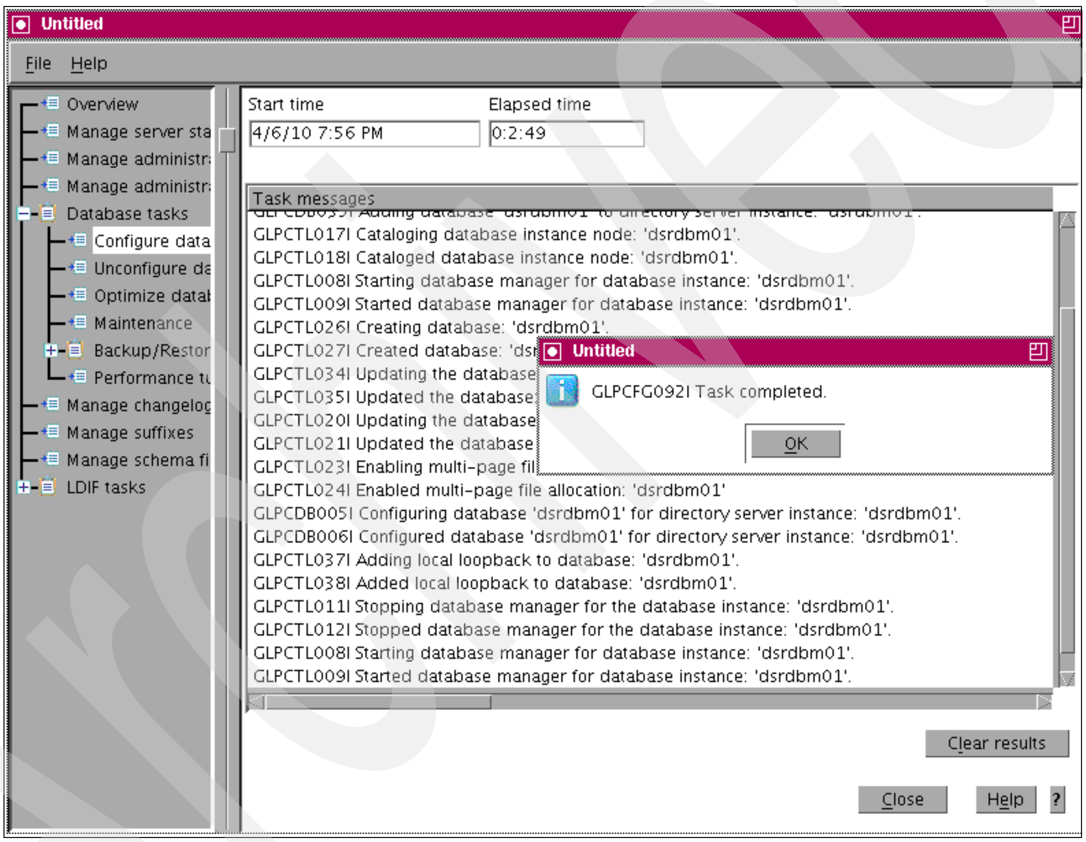


Figure 7-3 Creating a database for an instance with idsxcfg

7.7.4 idsxcfg requires Xwindows or Xserver

idsxcfg is a Java program that generates traces of the classes on the execution path (Example 7-14).

Example 7-14 idsxcfg cannot connect to X11 window

```
p5701db2(root)/opt/IBM/ldap/V6.2/sbin> ./idsxcfg
Mar 31, 2010 3:57:09 PM com.ibm.ldap.idsconsole.XCfgApp startClient
INFO: Starting client
_X11TransTRANS(ibmSHMConnect) () can't connect: errno = 68
Exception in thread "main" java.lang.InternalError: Can't connect to X11 window
```

```
server using ':1.0' as the value of the DISPLAY variable.
        at sun.awt.X11GraphicsEnvironment.initDisplay(Native Method)
        at
sun.awt.X11GraphicsEnvironment.access$000(X11GraphicsEnvironment.jav:95)
        at sun.awt.X11GraphicsEnvironment$1.run(X11GraphicsEnvironment.java:184
        at java.security.AccessController.doPrivileged(AccessController.java:19)
        at sun.awt.X11GraphicsEnvironment.<clinit>(X11GraphicsEnvironment.java:73)
        at java.lang.J9VMInternals.initializeImpl(Native Method)
        at java.lang.J9VMInternals.initialize(J9VMInternals.java:194)
        at java.lang.Class.forNameImpl(Native Method)
        at java.lang.Class.forName(Class.java:130)
        at
java.awt.GraphicsEnvironment.getLocalGraphicsEnvironment(GraphicsEnvronment.java:1
01)
        at sun.awt.motif.MToolkit.<clinit>(MToolkit.java:137)
        at java.lang.J9VMInternals.initializeImpl(Native Method)
        at java.lang.J9VMInternals.initialize(J9VMInternals.java:194)
        at java.lang.Class.forNameImpl(Native Method)
        at java.lang.Class.forName(Class.java:130)
        at java.awt.Toolkit$2.run(Toolkit.java:864)
        at java.security.AccessController.doPrivileged(AccessController.java:19)
        at java.awt.Toolkit.getDefaultToolkit(Toolkit.java:847)
        at javax.swing.UIManager.initialize(UIManager.java:1296)
        at javax.swing.UIManager.maybeInitialize(UIManager.java:1279)
        at javax.swing.UIManager.getDefaults(UIManager.java:590)
        at javax.swing.UIManager.getColor(UIManager.java:624)
        at
com.sun.java.swing.plaf.motif.MotifBorders$BevelBorder.<init>(MotifBrders.java:78)
        at
com.sun.java.swing.plaf.motif.MotifLookAndFeel.initComponentDefaultsMotifLookAndFe
el.java:243)
        at
javax.swing.plaf.basic.BasicLookAndFeel.getDefaults(BasicLookAndFeeljava:132)
        at javax.swing.UIManager.setLookAndFeel(UIManager.java:473)
        at javax.swing.UIManager.setLookAndFeel(UIManager.java:513)
        at com.ibm.ldap.idsconsole.XCfgApp.startClient(XCfgApp.java:246)
        at com.ibm.ldap.idsconsole.XCfgApp.main(XCfgApp.java:166)
```

The error messages come out early on top of the class dump. If the message displays as follows, it is due to a problem with the environment variable $DISPLAY:

```
Exception in thread "main" java.lang.InternalError: Can't connect to X11
windowserver using ':1.0' as the value of the DISPLAY variable.
```

Check list

To help determine whether the Xwindows or Xserver is missing:

1. Check the value of $DISPLAY.
2. Check to see whether XWindow is running.
3. If running remotely, check the X Server.

7.7.5 Importing a large LDIF file containing user IDs

We have an LDIF file that contains 10,000 user IDs. The file size is about 60 MB, and we have had a hard time importing it. There is no clear indicator of what the errors are. To resolve the problems we encountered, the file was broken down into small chunks for importing. It is also a good idea to try just one or a few user IDs first to validate the format. The LDIF file is in ASCII text format, which can be viewed and edited manually. However, remember to use the least destructive editing method when breaking up the file (Figure 7-4).

```
192.168.100.198 - PuTTY
p5701db2(root)/tmp> ls -l test.ldif
-rw-r-----    1 root     system     59014351 Mar 17 20:05 test.ldif
p5701db2(root)/tmp> head -32 test.ldif > head.ldif
p5701db2(root)/tmp> cat head.ldif
# Define members of each group.
# entry-id: 1673
dn: uid=user1673,ou=Ottawa, o=cognos.com
objectclass: organizationalPerson
objectclass: top
objectclass: person
objectclass: inetOrgPerson
userPassword: {SHA}K6Nc4OdyKQJjxz3miqEkuIlIKAA=
sn: user1673
cn: user1673

# Define members of each group.
# entry-id: 1674
dn: uid=user1674,ou=Ottawa, o=cognos.com
objectclass: organizationalPerson
objectclass: top
objectclass: person
objectclass: inetOrgPerson
userPassword: {SHA}K6Nc4OdyKQJjxz3miqEkuIlIKAA=
sn: user1674
cn: user1674

# Define members of each group.
# entry-id: 1675
dn: uid=user1675,ou=Ottawa, o=cognos.com
objectclass: organizationalPerson
objectclass: top
objectclass: person
objectclass: inetOrgPerson
userPassword: {SHA}K6Nc4OdyKQJjxz3miqEkuIlIKAA=
sn: user1675
cn: user1675
p5701db2(root)/tmp>
```

Figure 7-4 Extracting the top three user IDs in an LDIF file

7.8 Tips for accessing the Cognos portal

This section provides tips for challenges encountered while accessing the Cognos portal.

The dispatcher is still initializing problem

If Cognos has multiple dispatchers defined, the first dispatcher on the list has to be up in order for the web browser to access the portal. Otherwise, a message similar to the one in Figure 7-5 is displayed.

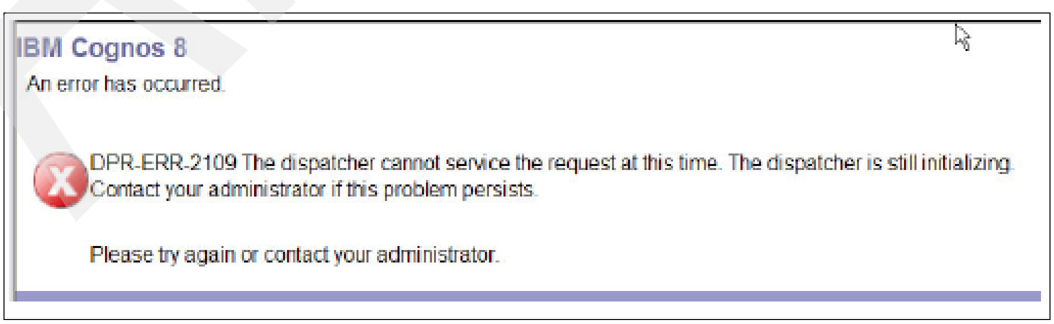


Figure 7-5 The dispatcher is still initializing

To check the dispatcher configuration in Cognos, run the **cogconfig.sh** command and click the environment menu in the explorer window on the left (Figure 7-6).

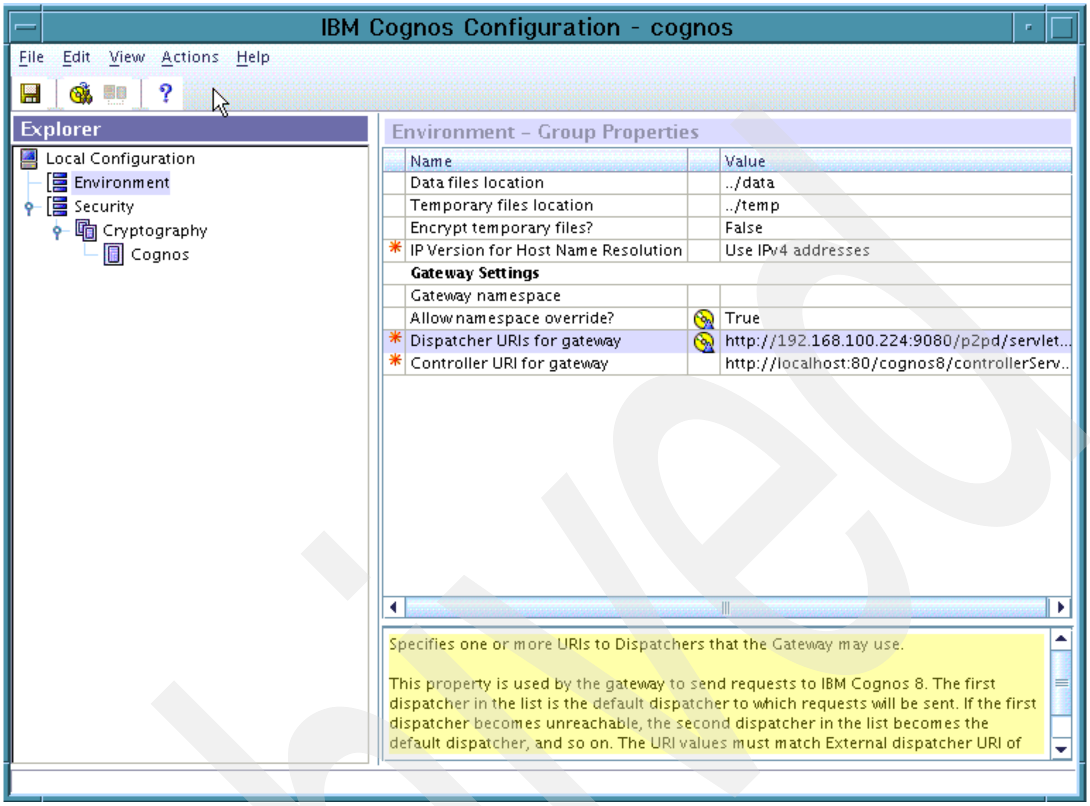


Figure 7-6 Dispatcher configuration in Cognos configuration environment

Then click the edit button on the dispatcher line on the right-hand side of the window. The dispatcher URLs for the gateway window allow changing the order of the dispatchers or adding or removing the dispatcher (Figure 7-7).

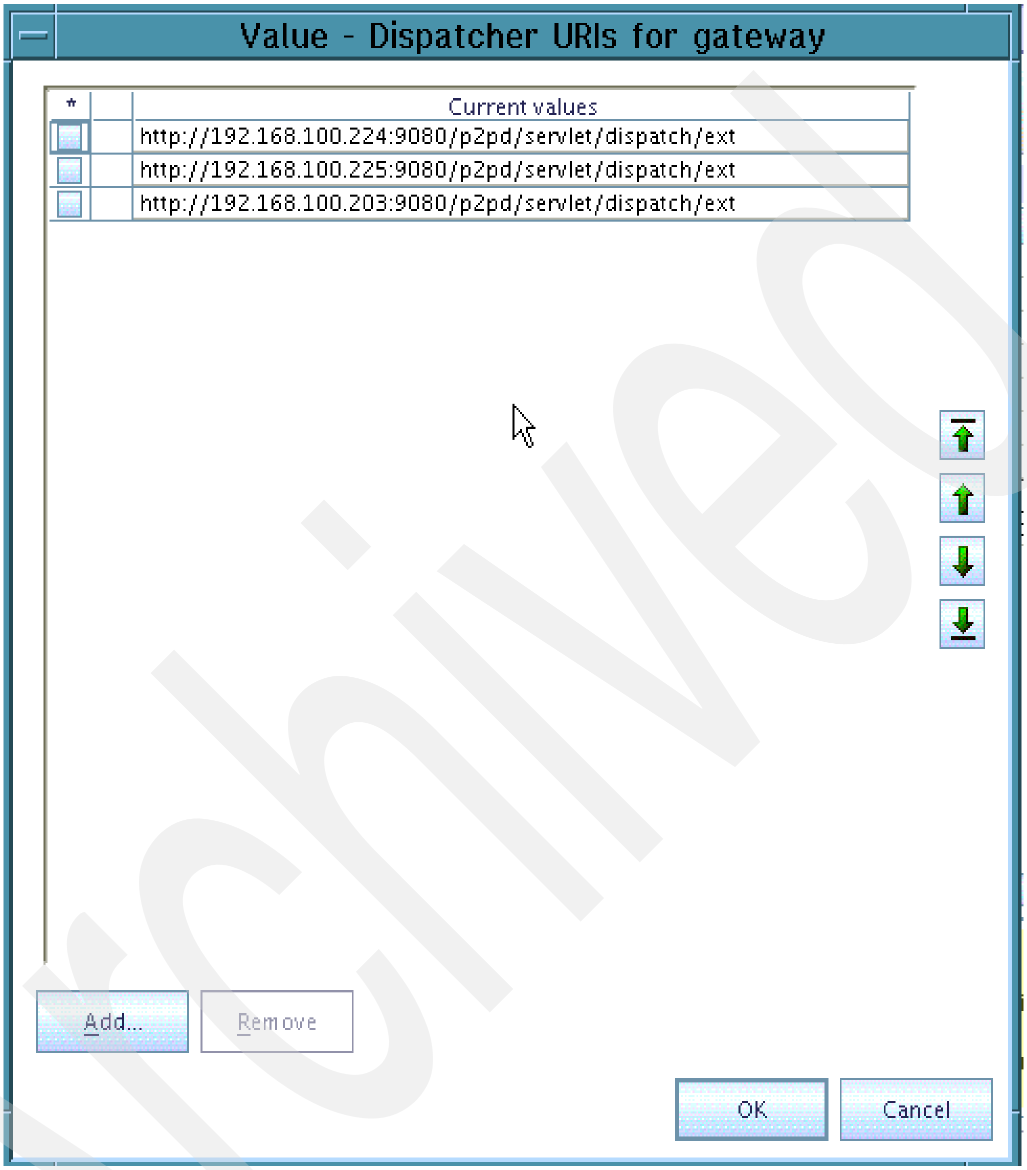


Figure 7-7 Changing dispatcher URLs for the gateway

7.9 Tips for starting the WebSphere Application Server (WAS)

This section provides suggestions for when encountering challenges while starting WAS.

7.9.1 Resolving the host name is required

If WAS cannot start after installation, one of the problems is that it cannot resolve the host name. The WAS `startServer.log` contains the error message shown in Figure 7-8.

```
[4/20/10 16:17:10:672 EDT] 0000000a WsServerImpl E WSVR0009E: Error occurred
during startup com.ibm.ws.exception.ConfigurationError:
com.ibm.ws.exception.ConfigurationError: javax.naming.ConfigurationException:
Cannot get canonical host name for server.
```

Figure 7-8 startServer.log error message on host name

The **host** command can be used to validate the host name or its IP address. Example 7-15 displays the error message 0827-803. A quick way to get around the problem is to add the host name and its IP address into the `/etc/hosts` file (Example 7-15).

Example 7-15 Problem with resolving host name

```
USER@cmtest / >host 192.168.100.225
host: 0827-803 Cannot find address 192.168.100.225.

USER@cmtest / >tail /etc/hosts
# line are not interpreted by routines which search this file.  Blank
# lines are allowed.

# Internet Address      Hostname        # Comments
# 192.9.200.1           net0sample      # ethernet name/address
# 128.100.0.1           token0sample    # token ring name/address
# 10.2.0.2              x25sample       # x.25 name/address
# 2000:1:1:1:209:6bff:feee:2b7f         ipv6sample      # ipv6 name/address
127.0.0.1               loopback localhost      # loopback (lo0) name/address
192.168.100.224 cmtest

USER@cmtest / >echo "192.168.100.225 apptest" >> /etc/hosts

USER@cmtest / >tail -2 /etc/hosts
192.168.100.224 cmtest
192.168.100.225 apptest

USER@cmtest / >host 192.168.100.225
apptest is 192.168.100.225
```

7.9.2 Cannot stop or start WAS

After issuing the **stopServer.sh** command, check that the server process has gone by using the **ps** command. If the server process is still there, **startServer.sh** cannot start WAS. To resolve the problem, terminate the process using the **kill** command (Example 7-16).

Example 7-16 Checking for the running WAS server process

```
USER@cmtest / >ps -ef | grep server
    root 49807474  3145788   0 17:42:58  pts/0  0:00 grep server
    root 30933114        1 250 17:04:04  pts/1 32:57
/usr/IBM/WebSphere/AppServer_1/java/bin/java -Declipse.security
-Dwas.status.socket=52351 -Dosgi.install.area=/usr/IBM/WebSphere/AppServer_1
```

```
-Dosgi.configuration.area=/usr/IBM/WebSphere/AppServer_1/profiles/AppSrv01/configu
ration -Djava.awt.headless=true -Dosgi.framework.extensions=com.ibm.cds
-Xshareclasses:name=webspherev61_%g,groupAccess,nonFatal -Xscmx50M
-Xbootclasspath/p:/usr/IBM/WebSphere/AppServer_1/java/jre/lib/ext/ibmorb.jar:/usr/
IBM/WebSphere/AppServer_1/java/jre/lib/ext/ibmext.jar -classpath
/usr/IBM/WebSphere/AppServer_1/profiles/AppSrv01/properties:/usr/IBM/WebSphere/App
Server_1/properties:/usr/IBM/WebSphere/AppServer_1/lib/startup.jar:/usr/IBM/WebSph
ere/AppServer_1/lib/bootstrap.jar:/usr/IBM/WebSphere/AppServer_1/lib/j2ee.jar:/usr
/IBM/WebSphere/AppServer_1/lib/lmproxy.jar:/usr/IBM/WebSphere/AppServer_1/lib/urlp
rotocols.jar:/usr/IBM/WebSphere/AppServer_1/deploytool/itp/batchboot.jar:/usr/IBM/
WebSphere/AppServer_1/deploytool/itp/batch2.jar:/usr/IBM/WebSphere/AppServer_1/jav
a/lib/tools.jar -Dibm.websphere.internalClassAccessMode=allow -Xms50m -Xmx256m
-Dws.ext.dirs=/usr/IBM/WebSphere/AppServer_1/java/lib:/usr/IBM/WebSphere/AppServer
_1/profiles/AppSrv01/classes:/usr/IBM/WebSphere/AppServer_1/classes:/usr/IBM/WebSp
here/AppServer_1/lib:/usr/IBM/WebSphere/AppServer_1/installedChannels:/usr/IBM/Web
Sphere/AppServer_1/lib/ext:/usr/IBM/WebSphere/AppServer_1/web/help:/usr/IBM/WebSph
ere/AppServer_1/deploytool/itp/plugins/com.ibm.etools.ejbdeploy/runtime
-Dderby.system.home=/usr/IBM/WebSphere/AppServer_1/derby
-Dcom.ibm.itp.location=/usr/IBM/WebSphere/AppServer_1/bin
-Djava.util.logging.configureByServer=true
-Duser.install.root=/usr/IBM/WebSphere/AppServer_1/profiles/AppSrv01
-Djavax.management.builder.initial=com.ibm.ws.management.PlatformMBeanServerBuilde
r -Dwas.install.root=/usr/IBM/WebSphere/AppServer_1
-Dpython.cachedir=/usr/IBM/WebSphere/AppServer_1/profiles/AppSrv01/temp/cachedir
-Djava.util.logging.manager=com.ibm.ws.bootstrap.WsLogManager
-Dserver.root=/usr/IBM/WebSphere/AppServer_1/profiles/AppSrv01
-Djava.security.auth.login.config=/usr/IBM/WebSphere/AppServer_1/profiles/AppSrv01
/properties/wsjaas.conf
-Djava.security.policy=/usr/IBM/WebSphere/AppServer_1/profiles/AppSrv01/properties
/server.policy com.ibm.wsspi.bootstrap.WSPreLauncher -nosplash -application
com.ibm.ws.bootstrap.WSLauncher com.ibm.ws.runtime.WsServer
/usr/IBM/WebSphere/AppServer_1/profiles/AppSrv01/config cmtestNode01Cell
cmtestNode01 server1

USER@cmtest / >kill -9 30933114
```

This chapter provided hints and tips that we discovered while implementing the business intelligence solution on IBM Power System servers.

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" on page 174. Note that several of the documents referenced here might be available in softcopy only.

- *Leveraging IBM Cognos 8 BI for Linux on IBM System z*, SG24-7812
- *PowerVM Virtualization Active Memory Sharing*, REDP-4470
- *IBM System p Advanced POWER Virtualization (PowerVM) Best Practices*, REDP-4194
- *PowerVM Virtualization on IBM System p: Introduction and Configuration Fourth Edition*, SG24-7940
- *WebSphere Application Server V7: Concepts, Planning and Design*, SG24-7708
- *WebSphere Application Server V7 Administration and Configuration Guide*, SG24-7615
- *WebSphere Application Server V6.1: Technical Overview*, REDP-4191

Other publications

These publications are also relevant as further information sources:

- *Cognos Product Development: Configuring WebSphere & Distributed Cognos ReportNet,* Document ID: DO-04111001
- *IBM Cognos 8 Business Intelligence version 8.4 Product Documentation*

  http://publib.boulder.ibm.com/infocenter/c8bi/v8r4m0/index.jsp
- *IBM Cognos 8 v4 Business Intelligence Information Center*

  http://publib.boulder.ibm.com/infocenter/c8bi/v8r4m0/index.jsp
- *Tivoli Directory Server documentation*

  http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?toc=/com.ibm.IBMDS.doc/toc.xml

Online resources

These websites are also relevant as further information sources:

- Cognos 8 Business Intelligence (BI) on IBM AIX best practices: Optimizing and scaling Cognos 8 BI on IBM AIX

  http://http://www.ibm.com/partnerworld/wps/servlet/ContentHandler/whitepaper/aix/v6r1_cognos/methods
- High Availability and Application tier for Cognos 8

  http://http://www.ibm.com/developerworks/data/library/cognos/page67.html
- Cognos 8 BI Architecture and Deployment Guide (Navigate to the Documentation page and select the Cognos 8 Business Intelligence 8.4 product.)

  http://support.cognos.com
- Cognos 8 BI Administration and Security Guide (Navigate to the Documentation page and select the Cognos 8 Business Intelligence 8.4 product.)

  http://support.cognos.com
- Installation and configuration guide for Linux on System z (Navigate to the Documentation page and select the Cognos 8 Business Intelligence 8.4 product.)

  http://support.cognos.com

How to get Redbooks

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

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

A

B

C

W

IBM Redbooks

Exploiting IBM PowerVM Virtualization Features with IBM Cognos 8 Business Intelligence

Exploiting IBM PowerVM Virtualization Features with IBM Cognos 8 Business Intelligence

Demonstrates how to maximize utilization of system resources

Includes IBM PowerVM and Cognos BI scenarios

Illustrates best practices and procedures

This IBM Redbooks publication addresses topics to leverage the virtualization strengths of the IBM Power platform to solve customer system resource utilization challenges and maximize system throughput and capacity.

This publication will help you leverage the strengths of the Power platform. It provides implementation scenarios with Cognos 8 Business Intelligence (BI) with the comprehensive set of the IBM PowerVM virtualization features and identifies and documents best practices for exploiting the IBM PowerVM virtualization features within Cognos BI deployments to maximize utilization of system resources and maximize Cognos throughput and capacity.

This publication is targeted at technical professionals (BI consultants, technical support staff, IT architects, and IT specialists) responsible for providing business intelligence solutions and support for Cognos BI on Power Systems.

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

SG24-7842-00 ISBN 0738434582