Redbooks

ibm.com/redbooks

Oracle on IBM Power Systems

Dino Quintero

Andrew Braid

Frederic Dubois

Alexander Hartmann

Octavian Lascu

Francois Martin

Wayne Martin

Stephan Navarro

Norbert Pistoor

Hubert Savio

Ralf Schmidt-Dannert

Analytics

Power Systems

IBM

Redbooks

IBM Redbooks

Oracle on IBM Power Systems

November 2021

SG24-8485-00

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

First Edition (November 2021)

This edition applies to:
Oracle RAC 19c Database.
AIX 7.1 TL5 SP01.
AIX 7.2 TL2 SP1.
AIX 7.2 Technology Level 4, Service Pack 2
IBM Spectrum Scale 5.0.3.3

Contents

Notices vii
Trademarks viii

Preface ix
Authors. ix
Now you can become a published author, too! xii
Comments welcome. xii
Stay connected to IBM Redbooks xii

Chapter 1. IBM Power Systems and IBM AIX features. 1
1.1 IBM Power Systems 2
1.1.1 Reliability. 2
1.1.2 Availability 3
1.1.3 Serviceability. 3
1.2 IBM AIX. 4

Chapter 2. Architecture design options and choices 5
2.1 Introduction 6
2.2 IBM Power Systems architectural design and RAS capabilities 6
2.2.1 Integrated system design 6
2.2.2 Keeping error detection as a first priority 6
2.2.3 Using technology and design for soft error management 7
2.2.4 Deploying strategic spare capacity to self-heal hardware. 8
2.2.5 Redundancy 8
2.2.6 Spare components 9
2.2.7 Focusing on operating system independence. 9
2.2.8 Building system level RAS rather than just processor and memory RAS. 9
2.2.9 Error reporting and handling 10
2.3 PowerVM partitioning and outages 12

Chapter 3. Oracle Database in hybrid multicloud infrastructure on IBM Power Systems 13
3.1 Overview of cloud computing 14
3.2 Power Systems consumption-based model 15
3.2.1 IBM Private Cloud with Dynamic Capacity 16
3.2.2 IBM Power Systems Virtual Server. 17
3.3 Path to hybrid cloud with IBM Power Systems 19
3.3.1 Step 1: Transform traditional IT to on-premises Private Cloud 19
3.3.2 Step 2: Make available off-premises Infrastructure-as-a-Service and cloud services 21
3.3.3 Step 3: Use hybrid cloud Power Systems. 22
3.4 IBM PowerVC 24
3.5 IBM Terraform and Service Automation 27
3.6 Red Hat Ansible 29
3.7 Other orchestration solutions 30

Chapter 4. Migration best practices 31
4.1 Motivation 32
4.2 Migration plan 33

4.2.1 Analyzing the source environment 33
4.2.2 Evaluate and select a migration method. 34
4.2.3 First test migration 35
4.2.4 Second test migration 35
4.2.5 Live migration 35
4.3 Migration methods. 35
4.3.1 Traditional export /import 35
4.3.2 Oracle Data Pump tool 36
4.3.3 Data transfer over Oracle named network links 37
4.3.4 Logical replication 37
4.3.5 Oracle Data Guard 37
4.3.6 Transportable Database 38
4.3.7 Cross-Platform Transportable Tablespace 39
4.3.8 Full Transportable Export/Import. 39
4.3.9 Cross-Platform Incremental Backup 39
4.4 Combined method for optimized migration 40
4.5 Preferred methods for different scenarios. 42
4.5.1 Components of the Migration Factory Framework 43

Chapter 5. Recommendations for preparing the Oracle software installation 47
5.1 Preparing to install Oracle Database on AIX. 48
5.2 Firmware 48
5.3 AIX 48
5.3.1 X Window System terminal (optional) 49
5.3.2 Unzip (mandatory) 49
5.3.3 Operating system changes (mandatory). 50
5.3.4 Space in /tmp for all partitions (mandatory) 50
5.3.5 Keep AIX updated (best practice). 50
5.4 CPU (best practice). 51
5.4.1 Configuring LPAR profile for Shared Processor LPAR performance (best practice) 51
5.4.2 Checking that Power Saver Mode is set to Maximum Performance Mode (best practice) 51
5.4.3 Activating Spectre and Meltdown (best practice) 52
5.4.4 Using SMT-8 for performance improvement (best practice) 53
5.4.5 Checking LPAR affinity (best practice) 53
5.5 Memory 54
5.5.1 64 K page promotion (Best Practice) 55
5.6 DNS (RAC only; mandatory) 55
5.7 Network best practices 55
5.7.1 Network settings for RAC (best practice) 56
5.7.2 NTPD (RAC only, mandatory). 56
5.8 UNIX users and groups (mandatory) 58
5.8.1 Turning off Oracle Online Patching in your environment if not required (best practice) 59
5.9 Storage 59
5.9.1 LUNs for logical volumes (best practice). 59
5.9.2 Setting the queue depth to allow enough I/O throughput (best practice) 60
5.9.3 Mount options for Oracle binaries (best practice) 60
5.9.4 Standalone with JFS2 61
5.9.5 Standalone with ASM 62
5.9.6 Required file sets for Oracle Database software installation (mandatory) 63
5.10 Installing the Oracle Database 65

5.11 Post-installation tasks 67
5.12 Creating the database (best practices) 68

Chapter 6. Oracle RAC for AIX with IBM Spectrum Scale 69
6.1 Introduction and considerations 70
6.2 POWER9 platform capabilities 70
6.2.1 CPU 70
6.2.2 Memory 70
6.2.3 Networking 71
6.2.4 Storage options 71
6.3 Configuration overview 72
6.3.1 Scenario description 72
6.3.2 Shared storage configuration 73
6.3.3 Networking redundancy overview 74
6.4 AIX platform preparation (excerpt) 75
6.4.1 Operating system parameters 77
6.4.2 Network configuration 80
6.4.3 Users and groups 84
6.4.4 Storage configuration 86
6.5 IBM Spectrum Scale configuration 88
6.6 Deploying Oracle software 96
6.6.1 Preparing for OUI GUI 96
6.6.2 Extracting the Oracle Grid 19c installation archive 97
6.6.3 Extracting the Oracle Database 19c installation archive 97
6.6.4 Installing the Oracle Grid infrastructure 98
6.6.5 Installing the Oracle Database 100

Chapter 7. Monitoring and performance troubleshooting 105
7.1 Monitoring 106
7.2 Server monitoring 106
7.3 Database monitoring 107
7.4 Performance troubleshooting 108
7.4.1 CPU 108
7.4.2 Memory 108
7.4.3 I/O 109

Chapter 8. Oracle Database and storage 111
8.1 AIX I/O stack 112
8.2 AIX I/O queues 113
8.3 Data layout for Oracle Database 115

Chapter 9. Networking options for LPARs running Oracle DB 119
9.1 Introduction 120
9.2 Dedicated physical network adapter in logical partition 120
9.3 Shared Ethernet Adapter in Virtual I/O Server 121
9.4 Multiple RAC clusters sharing servers with SEA-based interconnect 122

Appendix A. Additional material 125
Locating the GitHub material 126

Related publications 127
IBM Redbooks 127
Online resources 127
Help from IBM 127

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

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 jurisdictions 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 provide in any way it believes appropriate without incurring any obligation to you.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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 actual people or business enterprises 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. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

Redbooks (logo) ®
AIX®
FlashCopy®
IBM®
IBM Cloud®
IBM Cloud Pak®
IBM Garage™
IBM Spectrum®
IBM Z®
InfoSphere®
Interconnect®
POWER®
POWER8®
POWER9™
PowerVM®
Redbooks®

The following terms are trademarks of other companies:

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

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

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

Ansible, OpenShift, Red Hat, are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

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

VMware, and the VMware logo are registered trademarks or trademarks of VMware, Inc. or its subsidiaries in the United States and/or other jurisdictions.

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

Preface

This IBM® Redbooks® publication helps readers to understand how Oracle uses the architectural capabilities of IBM Power Systems.

This book delivers a technical snapshot of Oracle on Power Systems by using the general available and supported software and hardware resources to help you guide you to understand:

Why use Oracle on Power Systems?

- Efficiencies and benefits of the Power Systems architecture.
- Strengths of the Power Systems architecture (differentiate the technology and align with Oracle DB requirements, processor architecture helpers, and IBM AIX® on IBM POWER® system).
- Use scenarios that align to the story, and showcase and document step by step the relevant, selected strengths, and system features in the scenarios.

The goal of this publication is to complement Oracle and Power Systems features to deliver general and typical content, including details with a modern publication view of the solution for deploying Oracle (RAC and Single instance) on Power Systems by using theoretical knowledge, hands-on exercises, and documenting the findings by way of sample scenarios.

This publication addresses topics for developers, IT specialists, systems architects, brand specialist, sales team, and anyone looking for a guide about how to implement the best options for Oracle on Power Systems. Moreover, this book provides documentation to transfer the how-to-skills to the technical teams, and solution guidance to the sales team.

This publication complements the documentation that is available at the IBM Documentation, and aligns with the educational materials that are provided by the IBM Garage for Systems Technical Training.

Authors

This book was produced by a team of specialists from around the world working at IBM Redbooks, Poughkeepsie Center.

Dino Quintero is a Power Systems Technical Specialist with Garage for Systems. He has 25 years of experience with IBM® Power Systems technologies and solutions. Dino shares his technical computing passion and expertise by leading teams to develop technical content in the areas of enterprise continuous availability, enterprise systems management, high-performance computing (HPC), cloud computing, artificial intelligence (including machine and deep learning), and cognitive solutions. He also is a Certified Open Group Distinguished IT Specialist. Dino holds a Master of Computing Information Systems degree and a Bachelor of Science degree in Computer Science from Marist College.

Andrew Braid is a technical specialist working in the IBM Oracle Center in Montpellier, France. He worked for Oracle Worldwide Support as part of the E-business Suite Core technologies team and as a team leader for Oracle Premium Support in the UK, specializing in platform migrations and upgrades before working as a production Database Administrator for several large Oracle E-Business Suite clients across Europe. He joined IBM in 2011 to provide support for benchmarks and customers running Oracle Databases on IBM Power Systems.

Frederic Dubois is a Global Competitive Sales Specialist at the IBM Garage™ for Systems at IBM Global Markets in France. He delivers client value by way of his technical, presentation, and writing skills, while supporting brand specific business strategies.

Alexander Hartmann is a Senior IT Specialist working for IBM Systems Lab Services in Germany and is a member of the IBM Migration Factory. He holds a master's degree in Business Informatics from the University of Mannheim, Germany. With more than 25 years of experience with relational databases, he has been working intensively for the last 16 years on all aspects of Oracle Databases, with a focus on migration, and performance-, and license-optimization. In addition to Oracle Database specialties, his areas of expertise include AIX, Linux, scripting, and automation.

Octavian Lascu is an IBM Redbooks® Project Leader and a Senior IT Consultant for IBM Romania with over 25 years of experience. He specializes in designing, implementing, and supporting complex IT infrastructure environments (systems, storage, and networking), including high availability and disaster recovery solutions and high-performance computing deployments. He has developed materials for and taught over 50 workshops for technical audiences around the world. He is the author of several IBM publications.

Francois Martin is a Global Competitive Sales Specialist is responsible for developing brand and product specific solutions that address customer's business needs (industry and business) and deliver client value while supporting brand-specific business strategies. Francois has experience and skills with Power Systems Sales competition. He understands customer situations, sales, and technical sales to tackle competitive proposals. Francois knows of competitor's sales strategy, especially competition against Oracle Launch worldwide sales plays, Enablement session, workshop, Webex for sellers, and BPs Skills and experience from previous assignments in IBM Education, Cloud consolidation, Teaching, Technical Sales Enablement, Performance benchmarks, TCO, AIX, Power systems, Virtualization, and Architecture design and technology.

Wayne Martin is the IBM Systems Solution Manager who is responsible for the technology relationship between IBM and Oracle Corporation for all IBM server brands. He also is responsible for developing the mutual understanding between IBM and Oracle about technology innovations that generate benefits for mutual customers. Wayne has held various technical and management roles at IBM that focused on driving enhancements of ISV software that use IBM mainframe, workstation, and scalable parallel products.

Stephan Navarro is an Oracle for IBM Power Systems Architect at the IBM Garage for Systems at IBM Global Markets in France.

Norbert Pistoor was a Senior Consultant at Systems Lab Services in Germany and a Member of the Migration Factory until his retirement in June 2021. He has more than 20 years of experience with Oracle Databases and more than 10 years with cross-platform database migrations. He contributed several enhancements to standard Oracle migration methods and incorporated them into the IBM Migration Factory Framework. He holds a PhD in physics from University of Mainz, Germany.

Hubert Savio is working in the IT field for over 23 years as a DBA, consultant, IT specialist, and IT architect on international projects, and Danish/Nordics. This background helped him build skills that are required by customers within Linux, UNIX, and SAP/Oracle areas for cloud or on-premises solutions. Hubert has worked with Oracle since late 1996 as far back as releases 6 and 7.3.4. He has a Masters degree in IT from France's Centre d'etudes Supérieures Industrielles of Strasbourg. Hubert is specialized in Oracle Real Application Clusters under AIX.

Ralf Schmidt-Dannert has over 29 years of experience in IT and currently works as an Executive IT Specialist with the IBM Advanced Technology Group, ISV on Power - Oracle team. He has spent most of his career focused on large database environments, designing, testing, and troubleshooting solutions for high performance, high availability, and near zero data loss disaster recovery. Ralf has helped customers in the financial, telecommunications, utility, retail, and manufacturing industries to choose suitable database and infrastructure technologies to meet their business requirements for databases up to a 100 terabytes. Most recently, he has been evaluating and implementing technologies to provide Database as a Service to customers that are running their databases on IBM Power Systems servers (on-premises) or in IBM Power Systems Virtual Server infrastructure (off-premises). This work includes Oracle Database on AIX and open source databases on Linux on Power.

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

Wade Wallace
IBM Redbooks, Poughkeepsie Center

Majidkhan Remtoula
IBM France

Doug Bloom, Charles Graham, Alexander Paul
IBM US

Reinaldo Katahira
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 IBM Redbooks residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, IBM Redbooks
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- 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

IBM Power Systems and IBM AIX features

This chapter describes IBM Power Systems and IBM AIX features that deliver technological capabilities to run Oracle workloads.

This chapter includes the following topics:

- 1.1, “IBM Power Systems” on page 2.
- 1.2, “IBM AIX” on page 4.

1.1 IBM Power Systems

This section highlights some of the key features of IBM Power Systems.

1.1.1 Reliability

From a server hardware perspective, *reliability* is a collection of technologies (such as chipkill memory error detection and correction and dynamic configuration) that enhance system reliability by identifying specific hardware errors and isolating the failing components.

Built-in system failure recovery methods enable cluster nodes to recover, without falling over to a backup node, when problems are detected by a component within a node in the cluster. Built-in system failure recovery is transparent and achieved without the loss or corruption of data. It is also much faster compared to system or application failover recovery (failover to a backup server and recover).

Because the workload does not shift from this node to another, no other node's performance or operation is affected. Built-in system recovery covers applications (monitoring and restart), disks, disk adapters, LAN adapters, power supplies (battery backups) and fans.

From a software perspective, reliability is the capability of a program to perform its intended functions under specified conditions for a defined period. Software reliability is achieved mainly in two ways: infrequent failures (built-in software reliability), and extensive recovery capabilities (self-healing - availability).

IBM's fundamental focus on software quality is the primary driver of improvements in reducing the rate of software failures. As for recovery features, IBM-developed operating systems historically mandated recovery processing in the mainline program and in separate recovery routines as part of basic program design.

As IBM Power Systems become larger, more customers expect mainframe levels of reliability. For some customers, this expectation derives from their previous experience with mainframe systems, which were "downsized" to UNIX servers. For others, this large server is a consequence of having systems that support more users.

The cost that is associated with an outage grows every year; therefore, avoiding outages becomes increasingly important. This issue leads to new design requirements for all AIX-related software.

For all operating system or application errors, recovery must be attempted. When an error occurs, it is not valid to give up and end processing. Instead, the operating system or application must at least try to keep the component that is affected by the error up and running. If that is not possible, the operating system or application makes every effort to capture the error data and automate system restart as quickly as possible.

The amount of effort that is put into the recovery is proportional to the effect of a failure and the reasonableness of "trying again". If recovery is not feasible, the effect of the error is reduced to the minimum suitable level.

Today, many customers require that recovery processing be subject to a time limit. They concluded that rapid termination with quick restart or takeover by another application or system is preferable to delayed success.

However, takeover strategies rely on redundancy that becomes more expensive as systems grow, and in most cases, the main reason for quick termination is to begin a lengthy takeover process as soon as possible. Thus, the focus is now shifting back toward core reliability with quality and recovery features.

1.1.2 Availability

Today's systems feature hot plug capabilities for many subcomponents, from processors to input/output cards to memory. Also, clustering techniques, reconfigurable input and output data paths, mirrored disks, and hot swappable hardware help to achieve a significant level of system availability.

From a software perspective, *availability* is the capability of a program to perform its function whenever it is needed. Availability is a basic customer requirement. Customers require a stable degree of certainty, and require that schedules and user needs are met.

Availability evaluates the percentage of time a system or program can be used by the customer for productive use. Availability is determined by the number of interruptions and the duration of the interruptions. It also depends on characteristics and capabilities, including the ability to perform the following tasks:

- Change program or operating system parameters without rebuilding the kernel and restarting the system.
- Configure new devices without restarting the system.
- Install software or update software without restarting the system.
- Monitor system resources and programs and cleanup or recover resources when failures occur.
- Maintain data integrity in spite of errors.

The AIX operating system includes many availability characteristics and capabilities from which your overall environment benefits.

1.1.3 Serviceability

The focus on serviceability is shifting from providing customer support remotely through conventional methods, such as phone and email, to automated system problem reporting and correction, without user (or system administrator) intervention.

Hot swapping capabilities of some hardware components enhance the serviceability aspect. A service processor with advanced diagnostic and administrative tools further enhances the system serviceability. An IBM System p server's service processor can call home in the service report, which provides detailed information for IBM Service to act upon. This automation not only eases the burden that is placed on system administrators and IT support staff, but it enables rapid and precise collection of problem data.

On the software side, *serviceability* is the ability to diagnose and correct or recover from an error when it occurs. The most significant serviceability capabilities and enablers in AIX are referred to as the *software service aids*. The primary software service aids are error logging, system memory dump, and tracing.

With the advent of next generation UNIX servers from IBM, many hardware reliability-, availability-, and serviceability-related issues (such as memory error detection LPARs, and hardware sensors) were implemented. These features are supported by the relevant software in AIX. These abilities continue to establish AIX as the best UNIX operating system.

1.2 IBM AIX

Many market requirements are available for continuous availability to resolve the following examples of typical customer pain points:

- Too many scheduled outages.
- Service depends on problem recreation and intrusive problem determination.
- System unavailability disrupts customer business.
- Need for reliable protection of customer data.

IBM made AIX robust regarding continuous availability characteristics. This robustness makes IBM UNIX servers the best in the market. IBM AIX continuous availability strategy features the following characteristics:

- Reduce the frequency and severity of (planned and unplanned) AIX system outages.
- Improve serviceability by enhancing AIX failure data capture tools.
- Provide enhancements to debug and problem analysis tools.
- Ensure that all necessary information that involves unplanned outages is provided to correct the problem with minimal customer effort.
- Use of mainframe hardware features for operating system continuous availability that us brought to IBM System p hardware.
- Provide key error detection capabilities through hardware-assist.
- Use other IBM System p hardware aspects to continue transition to “stay-up” designs, which are used for continuous availability.
- Maintain operating system availability in the face of errors while minimizing application effects.
- Use of sophisticated and granular operating system error detection and recovery capabilities.
- Maintain a strong tie between serviceability and availability.
- Provide problem diagnosis from data that is captured at first failure without the need for further disruption.
- Provide service aids that are nondisruptive to the customer environment.
- Provide end-to-end and integrated continuous availability capabilities across the server environment and beyond the base operating system.
- Provide operating system enablement and application and storage use of the continuous availability environment.

2

Architecture design options and choices

This chapter discusses architecture design options and choices to deploy Oracle on IBM Power Systems.

This chapter includes the following topics:

- 2.1, “Introduction” on page 6.
- 2.2, “IBM Power Systems architectural design and RAS capabilities” on page 6.
- 2.3, “PowerVM partitioning and outages” on page 12.

2.1 Introduction

This chapter describes IBM POWER design strengths, RAS capabilities, and how its architectural design delivers scalability, granularity, and recoverability.

2.2 IBM Power Systems architectural design and RAS capabilities

This section describes some of the design philosophies and characteristics that influence Power Systems designs.

2.2.1 Integrated system design

IBM Power Systems use a processor and other hardware components that are designed and manufactured by IBM, including memory buffer components and service processors.

Other components that are not designed or manufactured by IBM are chosen and specified by IBM to meet system requirements. These components are procured for use by IBM by using a rigorous procurement process that is intended to deliver reliability and design quality expectations.

The systems incorporate software layers (firmware) for error detection, fault isolation and support, and virtualization in a multi-partitioned environment. These features include IBM designed and developed service firmware. IBM PowerVM hypervisor also is designed and supported by IBM.

In addition, IBM offers two operating systems that were developed by IBM: AIX and IBM i. Both operating systems come from a code base with a rich history of design for reliable operation.

These components are designed with application availability in mind, including the software layers, which also can use hardware features, such as storage keys that enhance software reliability.

2.2.2 Keeping error detection as a first priority

It is not possible to detect or isolate every potential fault or combination of faults that a server might experience. It is important to invest in error detection, especially for workloads where getting the correct answer is paramount.

Within the IBM POWER9 processor and memory subsystem, an investment was made in error detection and fault tolerance. This investment includes checking for data in memory and caches and validating that data was transferred correctly across busses.

It goes well beyond error detection to include techniques for checking state machine transitions, residue checking for specific operations, and protocol checking to ensure that the transmitted bits are correct, and that the data went when and where it was expected.

When a fault is detected, the primary intent of the RAS design is to prevent reliance on this data. Most of the rest of the RAS characteristics that are discussed in this section describe ways in which disruption because of bad data can be eliminated or minimized. However, cases exist in which avoiding reliance on bad data or calculations means ending the operation.

Although error detection seems like a well-understood and expected goal, it is not always the goal of every possible subsystem design. For example, graphics processing units (GPUs) whose primary purpose is rendering graphics in noncritical applications exist in which a single dropped pixel on a window is of no significant importance, and a solid fault is an issue only if it is noticed.

In general, I/O adapters also can have less hardware error detection capability where they can rely on a software protocol to detect and recover from faults.

2.2.3 Using technology and design for soft error management

In a real sense, detected errors can take several forms. The most obvious is a functional fault in the hardware, such as a silicon defect, or a worn component that failed over time.

Another type of failure is what is broadly classified as a *soft error*. Soft errors are faults that occur in a system and are occasional events that are inherent in the design or temporary faults that are the result of an external cause.

For example, data cells in caches and memory can include a bit-value that is temporarily upset by an external event, such as a cosmic ray-generated particle. Logic in processors cores also can be subject to soft errors in which a latch can flip because of a particle strike or similar event. Busses that transmit data can experience soft errors because of clock drift or electronic noise.

The susceptibility to soft errors in a processor or memory subsystem depends on the design and technology that is used in these devices. This capability is the first line of defense.

Methods for interleaving data so that two adjacent bits in array flipping do not cause undetected multi-bit flips in a data word is another important design technique.

Ultimately when data is critical, detecting soft error events that occur needs to be done immediately, inline to avoid relying on bad data because periodic diagnostics is insufficient to catch an intermittent problem before damage is done.

The simplest approach to detecting many soft error events can be the use of parity protection on data that can detect a single bit flip. However, when such simple single bit error detection is deployed, the effect of a single bit upset is bad data. Discovering bad data without being able to correct it results in the termination of an application (or even a system) if data correctness is important.

To prevent such a soft error from affecting a system, a bit flip must be detected *and* corrected. This process requires more hardware than simple parity. It is common now to deploy a bit correcting error correction code (ECC) in caches that can contain modified data.

However, because such flips can occur in more than caches, such ECC codes are widely deployed in POWER9 processors in critical areas on busses, caches, and so on.

Protecting a processor from more than data errors requires more than ECC checking and correction. CRC checking with a retry capability is used on a number of busses, for example.

POWER processors since POWER6 were designed with sufficient error detection to not notice key typical software upsets that affect calculations, and to notice quickly enough to allow processor operations to be retried. Where retry is successful, as expected for temporary events, system operation continues without application outages.

2.2.4 Deploying strategic spare capacity to self-heal hardware

Techniques that protect against soft errors offer limited protection against solid faults that are caused by a real hardware failure. For example, a single bit error in a cache can be continually corrected by most ECC codes that allow double-bit detection and single bit correction.

However, if a solid fault is continually being corrected, the second fault that occurs typically causes data that is not correctable. This issue results in the need to at least end whatever uses the data.

In many system designs, when a solid fault occurs in something like a processor cache, the management software on the system (the hypervisor or operating system) can be signaled to migrate the failing hardware off the system.

This process is called *predictive deallocation*. Successful predictive deallocation allows for the system to continue to operate without an outage. However, to restore full capacity to the system, the failed component still must be replaced, which results in a service action.

Within POWER, the general philosophy is to go beyond simple predictive deallocation by incorporating strategic sparing or micro-level deallocation of components so that when a hard failure that affects only a portion of the subsystem occurs, full error detection capabilities can be restored without the need to replace the failed part.

Examples include a spare data lane on a bus, a spare bit-line in a cache, having caches that are split up into multiple small sections that can be deallocated, or a spare DRAM module on a DIMM.

2.2.5 Redundancy

Redundancy is generally a means of continuing operation in the presence of specific faults by providing more components or capacity than is needed for system operation, but where a service action is to be taken to replace the failed component after a fault.

Sometimes, redundant components are not actively in use unless a failure occurs. For example, a processor actively uses only one clock source at a time, even when redundant clock sources are provided.

In contrast, if a system is said to have “n+1” fan redundancy, all “n+1” fans normally are active in a system absence a failure. If a fan fail occurs, the system runs with “n” fans. In such a case, power and thermal management code compensate by increasing fan speed or making adjustments according to operating conditions per the power management mode and policy.

2.2.6 Spare components

A *spare component* is similar to highly available spare capacity, except that when a spare is successfully used, the system continues to operate without replacing the component. For example, for voltage regulator output modules, if five output phases are needed to maintain the power that is needed at the specific voltage level, seven can be deployed initially. It takes the failure of three phases to cause an outage.

If on the first-phase failure, the system continues to operate, no call out is made for repair, and the first failing phase is considered spare. After the failure (spare is said to be used), the Voltage Regulator Module (VRM) can experience another phase failure with no outage. This component maintains the required n+1 redundancy. If a second phase fail, a redundant phase fails and a call-out for repair is made.

2.2.7 Focusing on operating system independence

Because Power Systems were designed to support multiple operating systems, the hardware RAS design is intended to allow the hardware to take care of the hardware largely independent of any operating system involvement in the error detection or fault isolation (excluding I/O adapters and devices for the moment.)

To a significant degree, this error handling is contained within the processor hardware. However, service diagnostics firmware (depending on the error) can aid in the recovery. When fully virtualized, specific operating system involvement in such tasks as migrating off a predictively failed component also can be performed transparent to the operating system.

The PowerVM hypervisor can create logical partitions with virtualized processor and memory resources. When these resources are virtualized by the hypervisor, the hypervisor has the capability of deallocating fractional resources from each partition when necessary to remove a component, such as a processor core or logical memory block (LMB).

When an I/O device is directly under the control of the operating system, the device error handling is the device driver’s responsibility. However, I/O can be virtualized through the VIO Server offering; that is, I/O redundancy can be achieved independently of the operating system.

2.2.8 Building system level RAS rather than just processor and memory RAS

IBM builds Power Systems with the understanding that every item that can fail in a system is a potential source of outage.

Although building a strong base of availability for the computational elements, such as the processors and memory is important, it is hardly sufficient to achieve application availability.

The failure of a fan, power supply, voltage regulator, or I/O adapter might be more likely than the failure of a processor module that is designed and manufactured for reliability.

Scale-out servers maintain redundancy in the power and cooling subsystems to avoid system outages that occur because of common failures in those areas. Concurrent repair of these components also is provided.

For the Enterprise system, a higher investment in redundancy is made. For example, the E980 system is designed with the expectation that the system must be generally shielded from the failure of these other components that are incorporating redundancy within the service infrastructure (redundant service processors, redundant processor boot images, and so on). The reliability of the components is emphasized where highly reliable means the component lasts.

This level of RAS investment extends beyond what is expected and often what is seen in other server designs. For example, at the system level, selective sparing includes such elements as a spare voltage phase within a voltage regulator module.

2.2.9 Error reporting and handling

This section focuses on the error handling capabilities of the architecture.

First failure data capture architecture

POWER processor-based systems are designed to handle multiple software environments, including various operating systems. This architecture motivates a design where the reliability and response to faults is not relegated to an operating system.

Further, the error detection and fault isolation capabilities are intended to enable retry and other mechanisms to avoid outages that are caused by soft errors. It also allows for use of self-healing features, which requires a detailed approach to error detection.

This approach is beneficial to systems as they are deployed by users. it also includes benefits in the design, simulation, and manufacturing test of systems.

Including this level of RAS into the hardware cannot be an afterthought. It must be integral to the design from the beginning, as part of an overall system architecture for managing errors. Therefore, during the design process of a processor, IBM places considerable emphasis on developing structures within it specifically for error detection and fault isolation.

Each subsystem in the processor hardware features registers that are devoted to collecting and reporting fault information as they occur. The design for error checking is rigorous and detailed. The value of data is checked generally wherever it is stored. This checking mechanism is true for data that is used in computations, but also nearly any other data structure, including arrays that only are used to store performance and debug data.

Error checkers are derived for logic structures by using various techniques. Such techniques include checking the validity of state-machine transitions, defining and checking protocols for generated commands, performing residue checks for specific computational instructions and by other means. These checks are made to detect faults before the resulting effect propagates beyond the detecting subsystem.

The exact number of checkers and type of mechanisms is not as important as is the fact that the processor is designed for detailed error checking. That is, much more is required than just reporting that a fault occurred at run time.

All of these errors feed a data reporting structure within the processor. Registers are used that collect the error information. When an error occurs, that event typically results in generating an interrupt.

The error detection and fault isolation capabilities maximize the ability to categorize errors by severity and handle faults with the minimum effect as possible.

Processor Runtime Diagnostics

In previous system designs, the dedicated server processor ran code, which is referred to here as *Processor Runtime Diagnostics* (PRD). PRD accesses this information and directs the error management to handle the recovery.

Ideally, this code primarily handles recoverable errors, including orchestrating the implementation of specific “self-healing” features, such as the use of spare DRAM modules in memory, purging and deleting cache lines, and the use of spare processor fabric bus lanes.

Code within a hypervisor controls specific system virtualized functions, especially as it relates to I/O including the PCIe controller and specific shared processor accelerators. Generally, errors in these areas are signaled to the hypervisor.

In addition, a reporting mechanism still exists for what amounts to the more traditional machine-check or checkstop handling.

In an IBM POWER7 generation system, the PRD was said to run and manage most errors whether the fault occurred at run time, or at system IPL time, or after a system-checkstop, which is the descriptive term for entire system termination by the hardware because of a detected error.

In IBM POWER8®, the processor module included a Self-Boot-Engine (SBE). This engine is loaded code on the processors that is intended to bring the system up to the point where the hypervisor can be started. Specific faults in early steps of that IPL process were managed by this code and PRD ran as host code as part of the start process.

In POWER9™ process-based systems, during normal operation the PRD code is run in a special service partition in the system on the POWER9 processors by using the hypervisor to manage the partition. This process has the advantage in systems with a single service processor of allowing the PRD code to run during normal system operation, even if the service processor is faulty.

Service diagnostics in the POWER9 design

In a POWER7 generation server, PRD and other service code was run within the dedicated service processor that is used to manage these systems. The dedicated service processor-managed IPL process that was used to start the hardware and bring the servers up to the state where the hypervisor can run. The dedicated service processor also managed to run the PRD code during normal operation.

In the rare event that a system outage resulted from a problem, the service processor accessed the basic error information that identified the type of fault that occurred. It also accessed considerable information about the state of the system hardware, including the arrays and data structures that represent the state of each processing unit in the system, and debug and trace arrays that can be used to further understand the root cause of faults.

Even if a severe fault caused system termination, this access provided the means for the service processor to determine the root cause of the problem, deallocate the failed components, and allow the system to restart with failed components removed from the configuration.

POWER8 gained a System-Boot-Engine, which allowed processors to run code and boot by using the POWER8 processors to speed up the process and provide for parallelization across multiple nodes in the high-end system. During the initial stages of the IPL process, the boot engine code handled specific errors and the PRD code ran as an application as later stages if necessary.

In POWER9, the changed so that during normal operation the PRD code runs in a special hypervisor-partition under the management of the hypervisor. This change has the advantage of continuing to allow the PRD code to run, even if the service processor is nonfunctional, which is important in non-redundant environments.

If a fault occurred, the code running fails, and the hypervisor can restart the partition.

The system service processors also are still monitored at run time by the hypervisor code and can report errors if the service processors are not communicating.

2.3 PowerVM partitioning and outages

The PowerVM® hypervisor provides logical partitioning, which allows multiple instances of an operating system to run in a server. At a high level, a server with PowerVM runs with a single copy of the PowerVM hypervisor, regardless of the number of nodes or partitions.

The PowerVM hypervisor uses a distributed model across the server's processor and memory resources. In this approach, some individual hypervisor code threads can be started and ended as needed when a hypervisor resource is required. Ideally, when a partition must access a hypervisor resource, a core that was running the partition then runs a hypervisor code thread.

Specific faults that might affect a PowerVM thread result in a system outage if these faults occur. This fault can be by stopping PowerVM or by the hardware determining that the system must checkstop for PowerVM integrity.

The design cannot be viewed as a physical partitioning approach. Multiple independent PowerVM hypervisors do not run in a system. If for fault isolation purposes, it is wanted to have multiple instances of PowerVM and hence multiple physical partitions, separate systems can be used.

Not designing a single system to have multiple physical partitions reflects the belief that the best availability can be achieved if each physical partition runs in separate hardware. Otherwise, a concern exists that when resources for separate physical partitions come together in a system, even IBM POWER9 Processor-based Systems RAS with redundancy, some common access point exists and the possibility a “common mode” fault that affects the entire system.

3

Oracle Database in hybrid multicloud infrastructure on IBM Power Systems

This chapter describes the POWER cloud portfolio and how to create a secure and reliable hybrid multicloud infrastructure on IBM Power Systems for Oracle workloads.

This chapter includes the following topics:

- 3.1, “Overview of cloud computing” on page 14
- 3.2, “Power Systems consumption-based model” on page 15
- 3.3, “Path to hybrid cloud with IBM Power Systems” on page 19
- 3.4, “IBM PowerVC” on page 24
- 3.5, “IBM Terraform and Service Automation” on page 27
- 3.6, “Red Hat Ansible” on page 29
- 3.7, “Other orchestration solutions” on page 30

3.1 Overview of cloud computing

Cloud computing is on-demand access to computing resources, applications, servers (physical and virtual), data storage, development tools, networking capabilities, and more. It transforms IT infrastructure into a utility, whether on-premises (private) or off-premises.

> **Note:** Oracle features specific considerations regarding what they support as a Public Cloud. For more information, see the following resources:
>
> - *Licensing Oracle Software in the Cloud Computing Environment*
> - *Oracle Database Support for Non-Oracle Public Cloud Environments* (MyOracleSupport Doc ID 2688277.1)

Consider the following points:

- *Off-premises* means that the service provider owns, manages, and assumes all responsibility for the data centers, hardware, and infrastructure on which its customers' workloads run. Typically, it provides high-bandwidth network connectivity to ensure high performance and rapid access to applications and data.
- *Private cloud* combines many of the benefits of cloud computing-including elasticity, scalability, and ease of service delivery with the access control, security, and resource customization of on-premises infrastructure.

Cloud is a capability, not a destination

Cloud computing changed how enterprise IT is delivered. In addition to on-demand access by way of services, it transformed the financial model from CAPEX to OPEX with part or complete resources billing on a consumption model or monthly subscription fee.

The following motivators drive enterprises to switch from traditional IT to a cloud computing model to run enterprise infrastructure more effectively and expand the business:

- Lower IT costs: Offload some or most of the costs and effort of purchasing, installing, configuring, and managing your own on-premises infrastructure.
- Improve agility and time-to-value: Use enterprise applications in minutes, instead of waiting weeks or months for IT to respond to a request, purchase and configure supporting hardware, and install software.
- Scale more easily and cost-effectively: Instead of purchasing excess capacity that sits unused during slow periods, you can scale capacity up and down in response to spikes and dips in traffic.

Many companies choose hybrid cloud to establish a mix of public and private cloud resources. This mixture includes a level of orchestration between them that gives an organization the flexibility to choose the optimal cloud for each application or workload. Organizations also can move workloads freely between the two clouds as circumstances change.

IBM Power Systems provides a cloud-ready platform that integrates with most of the needed tools and software to enable customers to implement cloud and Hybrid Multicloud.

3.2 Power Systems consumption-based model

A Power Systems consumption-based model (see Figure 3-1) includes the following features:

- Capacity upgrade on-demand: Power Systems provides features and functions to match available resources, CPU, and memory, to varying workload characteristics (for increases over time or to handle temporary spikes.
- Elastic capacity on-demand
- Utility capacity on-demand
- Trial capacity on-demand
- Power Systems Enterprise Pool

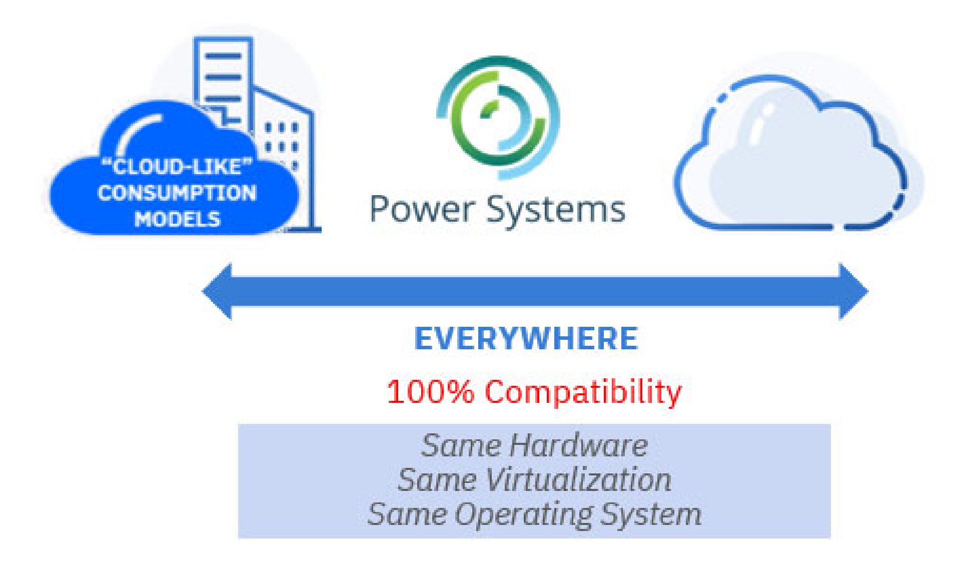


Figure 3-1 IBM Power Systems everywhere

In today's fast-paced business environment, pay-as-you-go or consumption-based infrastructure models are more requested by clients.

The following packaged solutions are available from IBM to address this requirement:

- On-premises: IBM Private Cloud with Dynamic Capacity
- Off-premises: hIBM Power Systems Virtual Server

These solutions are described next.

3.2.1 IBM Private Cloud with Dynamic Capacity

The IBM Private Cloud Solution with Shared Utility Capacity featuring Power Enterprise Pool 2.0 lowers IT acquisition costs and delivers a by-the-minute, pay-per-use consumption model in an on-premises environment.

The base capacity that a customer must purchase is as low as 1 core and 256 GB. Users can buy capacity credits for resource usage beyond the base capacity. They also can add multiple systems to the pool, as shown in Figure 3-2.

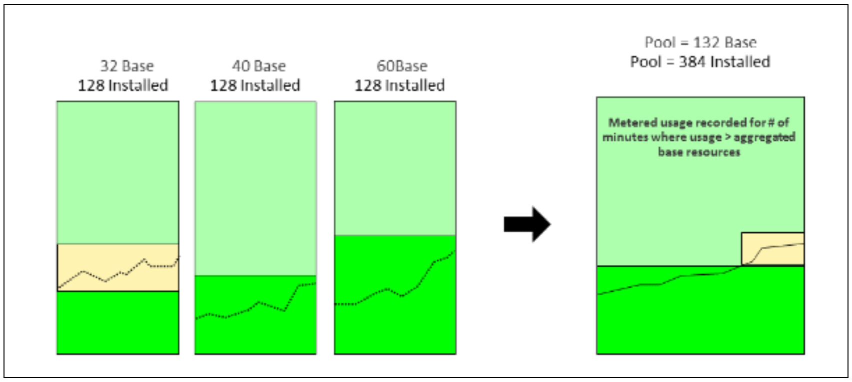


Figure 3-2 Pool usage with metered charges

When resource usage exceeds the aggregated base of the pool, capacity credits are debited in real time, based on by-the-minute resource consumption. Clients no longer need to worry about over-provisioning capacity to support growth.

For more information, see *IBM Power Systems Private Cloud with Shared Utility Capacity: Featuring Power Enterprise Pools 2.0*, SG24-8478.

All LPARs that are running Oracle DB on Power Systems can be assigned to a capped pool of processors to benefit from CPU sharing mechanisms that are provided by the hypervisor (shared processor pool). LPARs that require more CPU resource can get more capacity from other LPARs in that pool that cede idle CPU cycles to the pool. Overall CPU capacity is hard limited by the shared CPU pool.

An VM/LPAR can be limited to the maximum number of cores or processors to which it has access. This technology is consistent with Oracle's hard partitioning guidelines for Oracle licensing terms and conditions.

Power Enterprise Pools 2.0 has no effect on shared processor pools or LPARs capping mechanism. It relies on real CPU consumption and does not interact with resources that are assigned to a specific LPAR. That is, Power Enterprise Pool 2.0 is not a technology to reduce software licensing costs, but optimize hardware acquisition costs.

3.2.2 IBM Power Systems Virtual Server

IBM Power Systems Virtual Server is an enterprise hosted virtual infrastructure offering with access to over 200 IBM Cloud services. It is deployed in dedicated environments that are colocated with IBM Cloud but it is *not* on IBM Cloud. It is identical to customer on-premises environments and is composed of the same fully certified stack of components; therefore, it is fully supported by Oracle.

The IBM Power Virtual Server environment consists of Power Systems servers, PowerVM Hypervisor, and AIX operating systems that are certified for Oracle DB 12c, 18c, and 19c and other Oracle SW Products (Application, Middleware). This same stack is used by tens of thousands of customers in their current IT environment.

Oracle publishes their certifications of the PowerVM hypervisor, its features, AIX 7.1 and 7.2, and confirms support of these features at the following websites:

- Certified Virtualization and Partitioning Technologies for Oracle Database and RAC Product Releases
- My Oracle Support

Oracle is fully committed to support their products on certified stacks.

Per the Oracle Software Technical Policies document[1]:

> *Technical support is provided for issues (including problems you create) that are demonstrable in the currently supported release(s) of an Oracle licensed program, running unaltered, and on a certified hardware, database and operating system configuration, as specified in your order or program documentation*

The environment uses LPARs and adheres to Oracle's hard partitioning guidelines, if LPM is not used with those LPARs running Oracle software. Oracle licensing terms and conditions are always based on the contract between the customer and Oracle. For more information, see the *Oracle Partitioning Policy* document.

The customer is responsible to comply with the terms of the licensing contract with Oracle (regardless of the chosen deployment option chosen) on-premises or IBM Power Systems Virtual Server.

[1] `https://www.oracle.com/us/support/library/057419.pdf`

AIX/Power Systems provide choice and flexibility and prevent client lock-in to a specific vendor. Customers can select the correct location to deploy their Oracle Workloads while using the value-add of Power Systems for Oracle. Such choice allows them to take advantage of their investments in software and tailored databases options licenses by moving, migrating, or building Oracle environments to another location and switching back per business requirements, as shown in Figure 3-3.

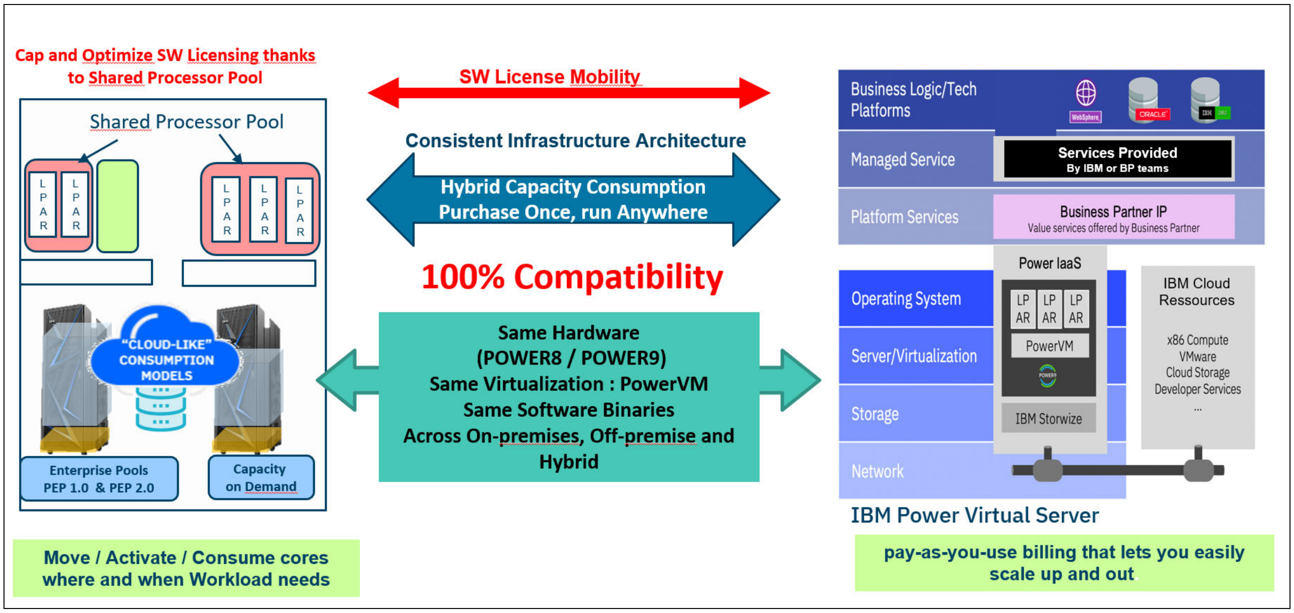

Cap and Optimize SW Licensing thanks to Shared Processor Pool
Shared Processor Pool
LPAR
LPAR
LPAR
LPAR
LPAR
"CLOUD-LIKE" CONSUMPTION MODELS
Enterprise Pools PEP 1.0 & PEP 2.0
Capacity on Demand
Move / Activate / Consume cores where and when Workload needs
SW License Mobility
Consistent Infrastructure Architecture
Hybrid Capacity Consumption
Purchase Once, run Anywhere
100% Compatibility
Same Hardware
(POWER8 / POWER9)
Same Virtualization : PowerVM
Same Software Binaries
Across On-premises, Off-premise and Hybrid
Business Logic/Tech Platforms
WebSphere
ORACLE
IBM DB2
Managed Service
Services Provided
By IBM or BP teams
Platform Services
Business Partner IP
Value services offered by Business Partner
Power IaaS
Operating System
Server/Virtualization
Storage
Network
LP AR
LP AR
LP AR
PowerVM
POWER9
IBM Storwize
IBM Cloud Ressources
x86 Compute
VMware
Cloud Storage
Developer Services
...
IBM Power Virtual Server
pay-as-you-use billing that lets you easily scale up and out

Figure 3-3 Hybrid POWER Cloud infrastructure for Oracle

3.3 Path to hybrid cloud with IBM Power Systems

The path to hybrid cloud begins with a solid foundation of infrastructure and hardware management capabilities whatever the location, as shown in Figure 3-4.

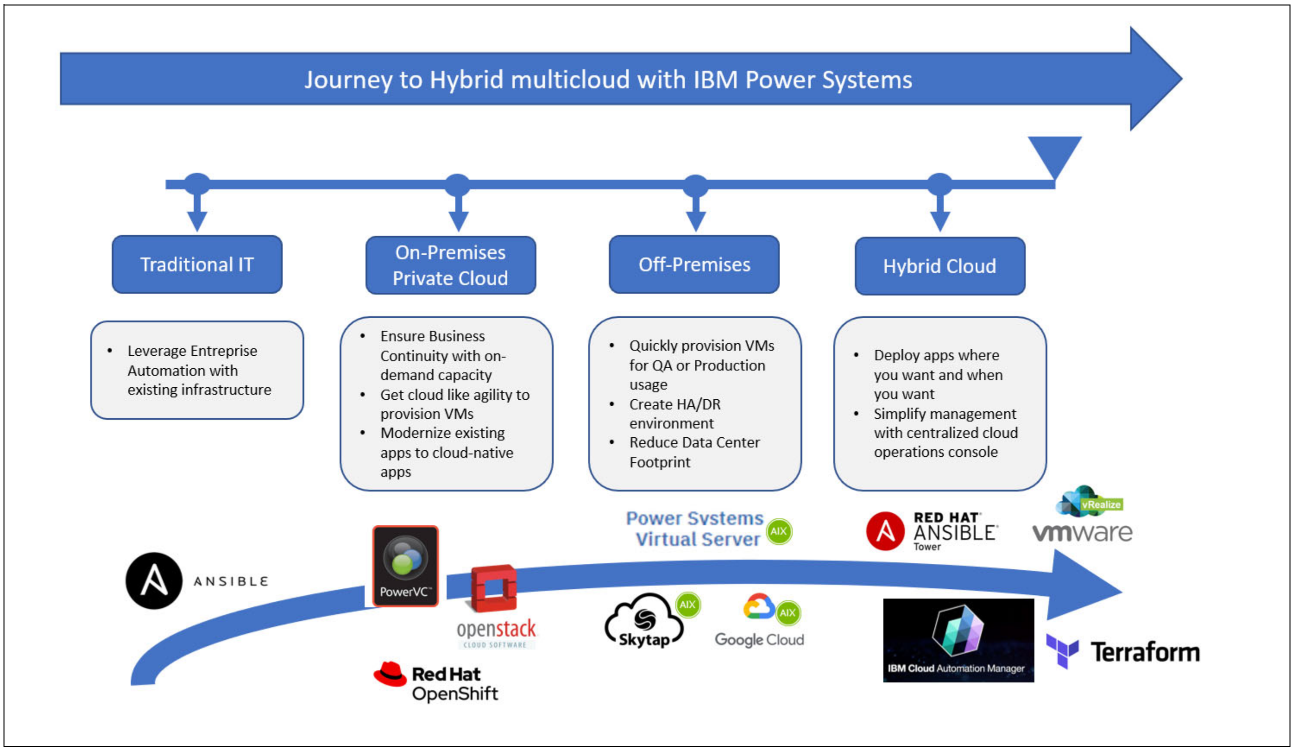

Journey to Hybrid multicloud with IBM Power Systems
Traditional IT
On-Premises Private Cloud
Off-Premises
Hybrid Cloud
Leverage Entreprise Automation with existing infrastructure
Ensure Business Continuity with on-demand capacity
Get cloud like agility to provision VMs
Modernize existing apps to cloud-native apps
Quickly provision VMs for QA or Production usage
Create HA/DR environment
Reduce Data Center Footprint
Deploy apps where you want and when you want
Simplify management with centralized cloud operations console
ANSIBLE
PowerVC
openstack
Red Hat OpenShift
Power Systems Virtual Server
AIX
Skytap
Google Cloud
RED HAT ANSIBLE Tower
vRealize
vmware
IBM Cloud Automation Manager
Terraform

Figure 3-4 Journey to hybrid multicloud

This hybrid multicloud design requires a unified cloud management solution. Power Systems is compatible with leading cloud orchestration technologies, including Red Hat Ansible, Chef, Puppet, and VMWare vRealize. Those single panes of glass solutions help and centralize deployment and management of environments wherever they run.

The flexibility is available to move to the cloud at your own pace.

This section addresses the steps to move from a traditional Power Systems IT landscape to a hybrid (on-premises + off-premises) POWER infrastructure.

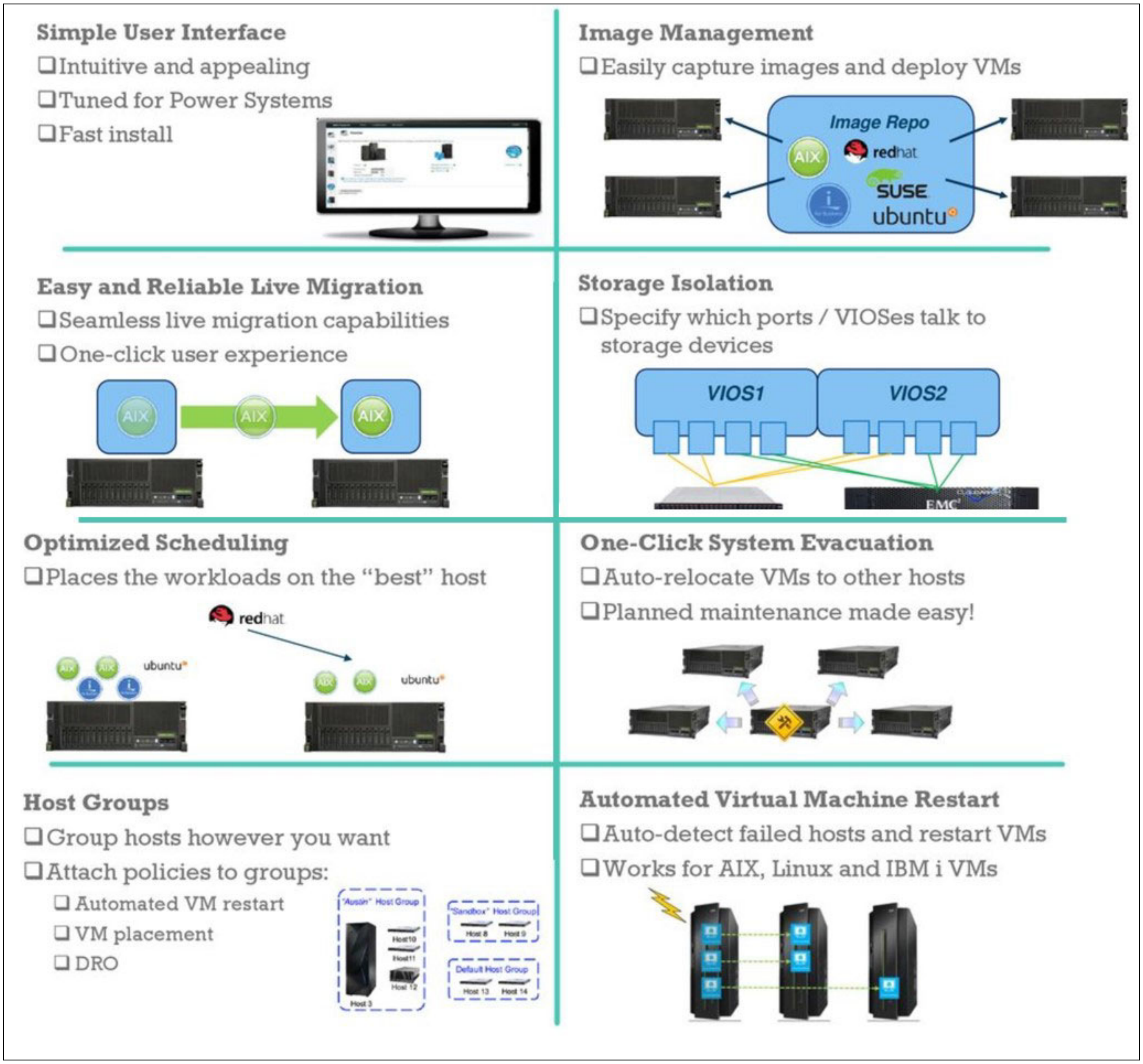
3.3.1 Step 1: Transform traditional IT to on-premises Private Cloud

IBM Power Virtualization Center (PowerVC) is the critical element in your cloud move. Built on OpenStack, it provides comprehensive virtualization and Cloud Management for IBM Power Systems servers to rapidly spin up sets of standardized VMs and then shut them down when finished.

Combined with on-demand capacity, clients realize cloud-like agility to provision virtual machines (VMs) that is based on templates that can incorporate software to accelerate deployment and installation of the complete stack. AIX, IBM i, or Linux on POWER VMs can easily be captured as new images and used by users to build new environments faster by way of the PowerVC portal.

For more information, see *IBM PowerVC, see IBM PowerVC Version 2.0 Introduction and Configuration*, SG24-8477.

IBM PowerVC features are shown in Figure 3-5.



Simple User Interface

❑ Intuitive and appealing

❑ Tuned for Power Systems

❑ Fast install

Image Management

❑ Easily capture images and deploy VMs

Easy and Reliable Live Migration

❑ Seamless live migration capabilities

❑ One-click user experience

Storage Isolation

❑ Specify which ports / VIOSes talk to storage devices

Optimized Scheduling

❑ Places the workloads on the "best" host

One-Click System Evacuation

❑ Auto-relocate VMs to other hosts

❑ Planned maintenance made easy!

Host Groups

❑ Group hosts however you want

❑ Attach policies to groups:

- ❑ Automated VM restart
- ❑ VM placement
- ❑ DRO

Automated Virtual Machine Restart

❑ Auto-detect failed hosts and restart VMs

❑ Works for AIX, Linux and IBM i VMs

Figure 3-5   Examples of IBM PowerVC features

Within this transformation of traditional IT, organizations also are modernizing traditional software to containerized applications to help improve operational efficiency, cloud integration from multiple vendors, and to build a more unified cloud strategy.

Red Hat OpenShift is fully enabled and supported on IBM Power Systems to rapidly build, deploy, and manage cloud-native applications.

IBM Cloud Paks are lightweight, enterprise-grade, modular cloud solutions, that integrate a container platform, containerized IBM middleware and open source components, and common software services for development and management.

IBM introduced IBM Cloud Paks to address those application transformation needs and offer a faster, more reliable way to build, move, and manage on the cloud.

Thanks to Red Hat OpenShift, which is paired with IBM Cloud Paks, customers gain enterprise-ready, containerized software solutions for an open, faster, and more secure way to move core business applications to any cloud (starting by on-premises private cloud).

For more information about Red Hat OpenShift and IBM Cloud Pak on IBM Power Systems, see *Red Hat OpenShift V4.X and IBM Cloud Pak on IBM Power Systems Volume 2*, SG24-8486.

3.3.2 Step 2: Make available off-premises Infrastructure-as-a-Service and cloud services

Companies are lead to take advantage of off-premises infrastructure for several reasons: a need for test and development environments, review a new features or version, use a new location for Disaster Recovery (DR) purposes, or switch to an Operational Expenses Model (OpEx), which reduces data center footprint.

The same technology stack as on-premises allows users to develop and customize those new environments and return them to on-premises (or vice versa) without risks or massive effort.

Some of these use cases described next.

Reducing testing time by migrating custom AIX on-premises to off-premises

Building up a separate and secure infrastructure on-premises for a test project involves more investments and can slow down the project because of lack of resources and challenges to cede resources from existing environments. Provisioning an AIX environment off-premises is now easy and cost effective with the IBM Power Systems Virtual Server offering. Clients can evaluate or test new application releases, features, and operating system versions.

Custom on-premises environments can be migrated to off-premises infrastructure by using one of the following methods:

- Export/ or import AIX image into OVA format
- AIX MKSYSD backup and restore

Our example shows how to export a custom AIX image from on-premises Power Systems infrastructure to IBM Power Systems Virtual Server by using PowerVC and IBM Cloud Object Storage. PowerVC images that are built with custom content can be exported in OVA format into an IBM Cloud Object Storage bucket to be imported into the boot image catalog of your Power Virtual Server Instance.

This process allows you to quickly build environments based on your own customized AIX Image. The process can be easily reversed to migrate an AIX LPAR built-in Power Virtual Server on which customization was done to back on-premises.

For more information, see the following videos:

- Oracle Database on IBM Power Virtual Server - Part 1
- Oracle Database on IBM Power Virtual Server - Part 2

Off-premises infrastructure for Disaster Recovery or production migration

Clients can now avoid the need to build their own data center for DR purpose and choose to subscribe to Power Virtual Server or an alternative off-premises Power Systems infrastructure provider, or start migrating production environments.

Clients can subscribe for the minimal virtual server configuration and expand the configuration depending on the needs at the time of failover or workload needs. IBM Infrastructure as a Service (IaaS) management expertise and experience is used to provide the services that relate to the hardware and infrastructure. Applications, databases, and operating systems layers remain under customer or out-sourced management control.

3.3.3 Step 3: Use hybrid cloud Power Systems

Hybrid multicloud implies an IT infrastructure that uses a mix of on-premises and private cloud and off-premises from multiple providers. From a single pane of glass, you deploy your applications and databases where and when you want.

IBM Cloud Paks include solutions, such as IBM Multicloud Manager, that allow customers to adopt VMs and containers in a hybrid multicloud environment while using your infrastructure investments. This multicloud self-service management platform empowers developers and administrators to meet business demands.

For more information, see this web page.

This platform allows you to efficiently manage and deliver services through end-to-end automation. It also enables developers to build applications that are aligned with enterprise policies and use open source Terraform to manage and deliver cloud infrastructure as code.

With Red Hat Ansible Tower, users can now define one master workflow that ties different areas of IT together, which is designed to cover a hybrid infrastructure without being stopped at specific technology silos. Red Hat Ansible ensures your cloud deployments work seamlessly across off-premises, on-premises, or hybrid cloud as easily as you can build a single system.

Figure 3-6 on page 23 shows the high-level hybrid multicloud reference architecture that is inclusive of the major industry hardware platforms; IBM Power Systems, IBM Z®, and x86.

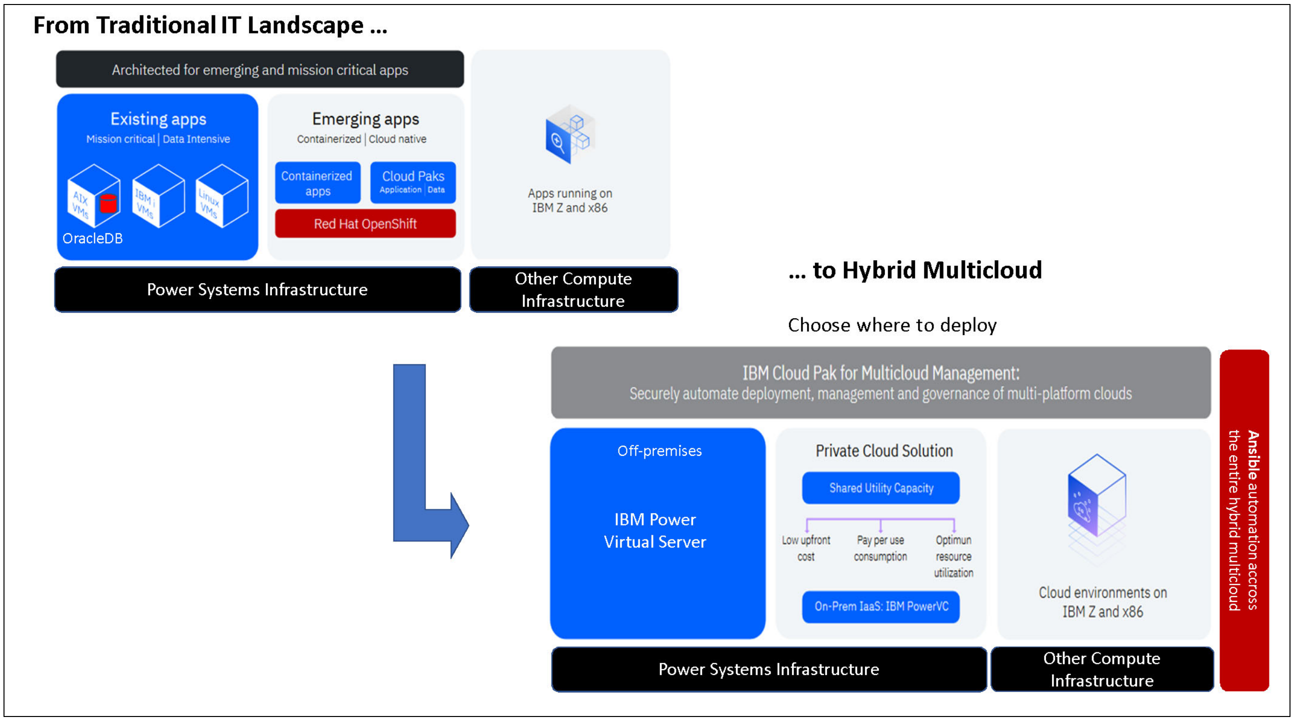

From Traditional IT Landscape ...
Architected for emerging and mission critical apps
Existing apps
Mission critical | Data Intensive
AIX VMs
IBM i VMs
Linux VMs
OracleDB
Emerging apps
Containerized | Cloud native
Containerized apps
Cloud Paks
Application | Data
Red Hat OpenShift
Apps running on IBM Z and x86
Power Systems Infrastructure
Other Compute Infrastructure
... to Hybrid Multicloud
Choose where to deploy
IBM Cloud Pak for Multicloud Management:
Securely automate deployment, management and governance of multi-platform clouds
Off-premises
IBM Power Virtual Server
Private Cloud Solution
Shared Utility Capacity
Low upfront cost
Pay per use consumption
Optimun resource utilization
On-Prem IaaS: IBM PowerVC
Cloud environments on IBM Z and x86
Power Systems Infrastructure
Other Compute Infrastructure
Ansible automation accross the entire hybrid multicloud

Figure 3-6 High-level reference architecture

Power Systems is deigned to economically scale mission-critical, data-intensive applications (VM-based or containerized) by delivering industry leading reliability to run them and reduce the cost of operations with built-in virtualization to optimize capacity utilization. It also provides flexibility and choice to deploy those containerized applications or VM in the cloud of your choice.

The architecture that is shown in Figure 3-6 applies to VMs (logical partitions) that are running AIX, which allows you to automate the deployment and management of the Oracle Database on AIX across the management stack.

Other orchestration tools are available, such as VMWare vRealize. IBM's partnership with VMware provides clients with the vRealize Suite to unify applications and infrastructure management across IBM Power Systems, x86 servers, and IBM Z environments.

The following sections describe how to take advantage of key components of the process to move to Hybrid Cloud on POWER for your Oracle Database. Examples of automated Oracle Database environment deployments are illustrated.

3.4 IBM PowerVC

IBM PowerVC is an advanced virtualization and cloud Cloud Management offering. Built on OpenStack, it provides simplified virtualization management and cloud deployments for IBM AIX, IBM i, and Linux VMs that are running on IBM Power Systems. The offer is designed to build a private cloud on the Power Systems servers and improve administrator productivity.

For more information about the process to build an Oracle DB as a Service (DBaaS) offering on AIX/Power Systems by developing a reference image with Oracle Database and Oracle Grid Infrastructure installed, see this IBM Support web page.

A post-provisioning step then updates the Oracle configuration to obtain the IP address, host name, and database hostname that are defined during the LPAR Creation process by way of PowerVC.

> **Note:** Implementation details and all developed scripts to implement Oracle Database as a Service are included. You can reuse them as-is to start your private cloud journey and then customize and enhance them as per your own constraints and requirements.
>
> For more information, see this web page.

Users of the DBaaS service can select from a set of deployable images, such as Oracle Database 12c or 19c, with JFS2 or with ASM. Then, customization parameters are provided, such as a database name or required storage capacity by way of the PowerVC interface. The DBaaS service administrator also can define constraints and any specific approval requirements as needed before a request is fulfilled.

After the user request is approved, all later steps are fully automated.

IBM Cloud® PowerVC Manager sends the deployment request to PowerVC with the provided customization parameters. IBM PowerVC then does the bulk of the work. It creates the LPAR by way of the HMC, allocates and maps storage, and then, starts the LPAR.

In the LPAR, Cloud-Init evaluates the provided parameters after start and adjusts the hostname, network setting, database name and any other customization accordingly (see Figure 3-7).

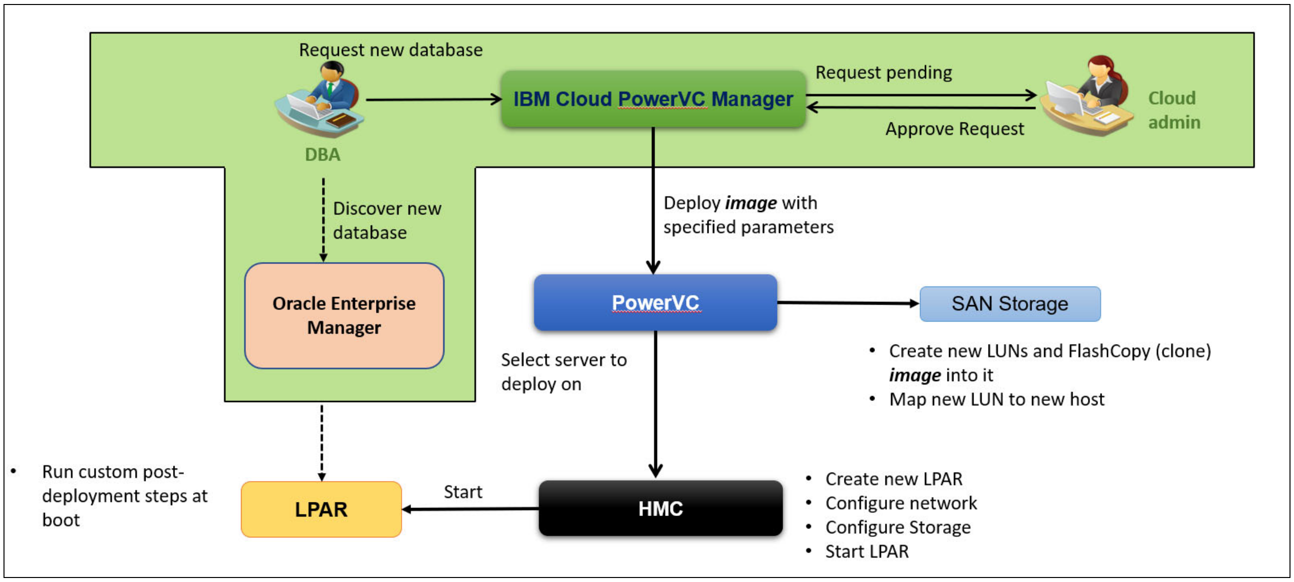
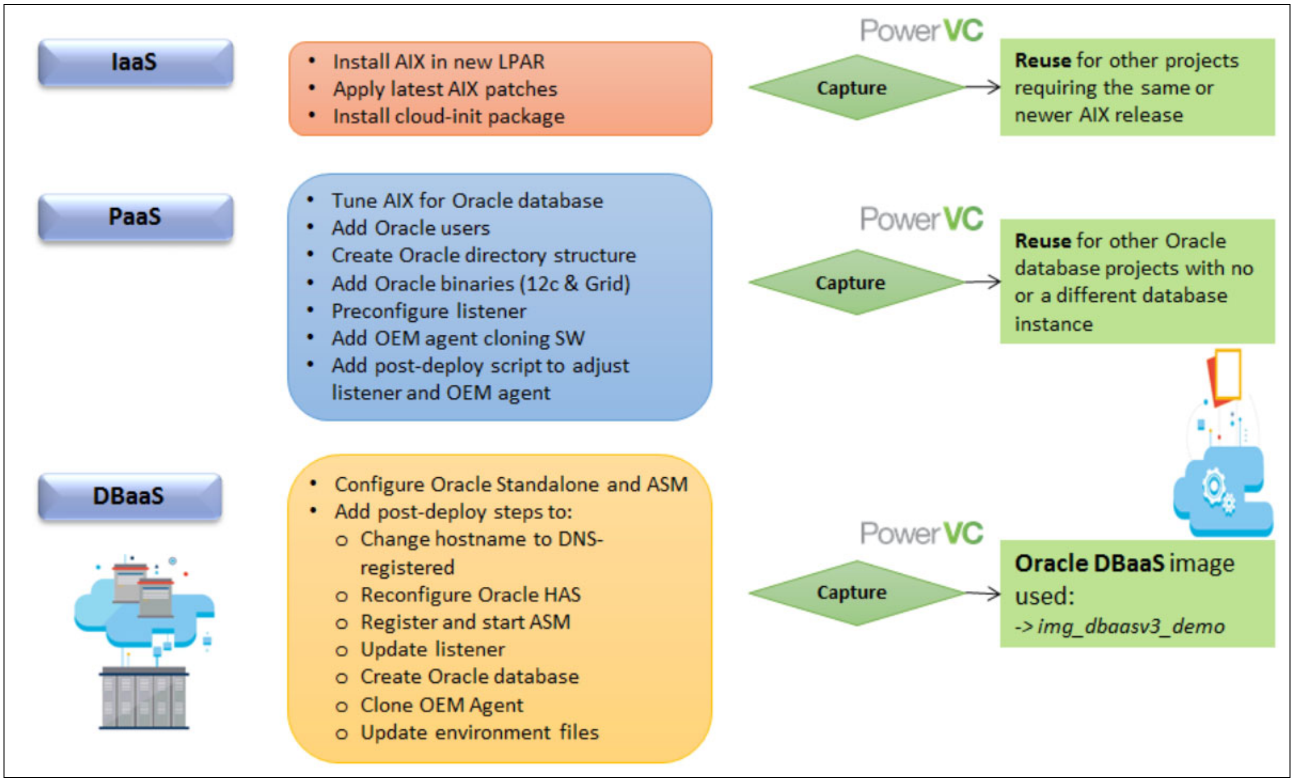

Request new database
IBM Cloud PowerVC Manager
Request pending
Approve Request
Cloud admin
DBA
Discover new database
Deploy *image* with specified parameters
Oracle Enterprise Manager
PowerVC
SAN Storage
Select server to deploy on
• Create new LUNs and FlashCopy (clone) *image* into it
• Map new LUN to new host
• Run custom post-deployment steps at boot
LPAR
Start
HMC
• Create new LPAR
• Configure network
• Configure Storage
• Start LPAR

Figure 3-7 Deployment of Oracle with IBM PowerVC

The deployment is then completed by running the post-deployment scripts that were included in the deployment image at capture time. Cloud-Init is the technology that takes user inputs and configures the operating system and software on the deployed virtual machine.

PowerVC relies on capturing and restoring images. This first step is part of the process to convert traditional on-premises Power Systems infrastructure to a private cloud and offer services and templates to users.

Figure 3-8 shows the high-level preparation steps that are taken to provide different service level IaaS, PaaS, and DBaaS, each with further refined images that provide enhanced functions.



- Install AIX in new LPAR
- Apply latest AIX patches
- Install cloud-init package

Reuse for other projects requiring the same or newer AIX release

- Tune AIX for Oracle database
- Add Oracle users
- Create Oracle directory structure
- Add Oracle binaries (12c & Grid)
- Preconfigure listener
- Add OEM agent cloning SW
- Add post-deploy script to adjust listener and OEM agent

Reuse for other Oracle database projects with no or a different database instance

- Configure Oracle Standalone and ASM
- Add post-deploy steps to:
  - Change hostname to DNS-registered
  - Reconfigure Oracle HAS
  - Register and start ASM
  - Update listener
  - Create Oracle database
  - Clone OEM Agent
  - Update environment files

Oracle DBaaS image used:
-> *img_dbaasv3_demo*

Figure 3-8 High-level image preparation steps

This PowerVC requires to maintain image sets and handle a large set of images to offer wide flexibility and choice regarding the operating system version and software stack version combination. However, PowerVC can be convenient when large databases must be cloned, for example. The capture and restore process saves time and avoids reinstalling the software stack and exporting and importing the Oracle DB.

To increase agility and choice in services to offer to users, you add an orchestration and a decomposition solution. This addition brings you to next level of Hybrid Cloud because such orchestration tools apply to on-premises and off-premises environments.

3.5 IBM Terraform and Service Automation

IBM Terraform and Service Automation (previously named IBM Cloud Automation Manager) runs on top of IBM Cloud Pak® for Multicloud Management. It automates provisioning infrastructure, databases, and applications across multiple cloud environments with optional workflow orchestration (see Figure 3-9).

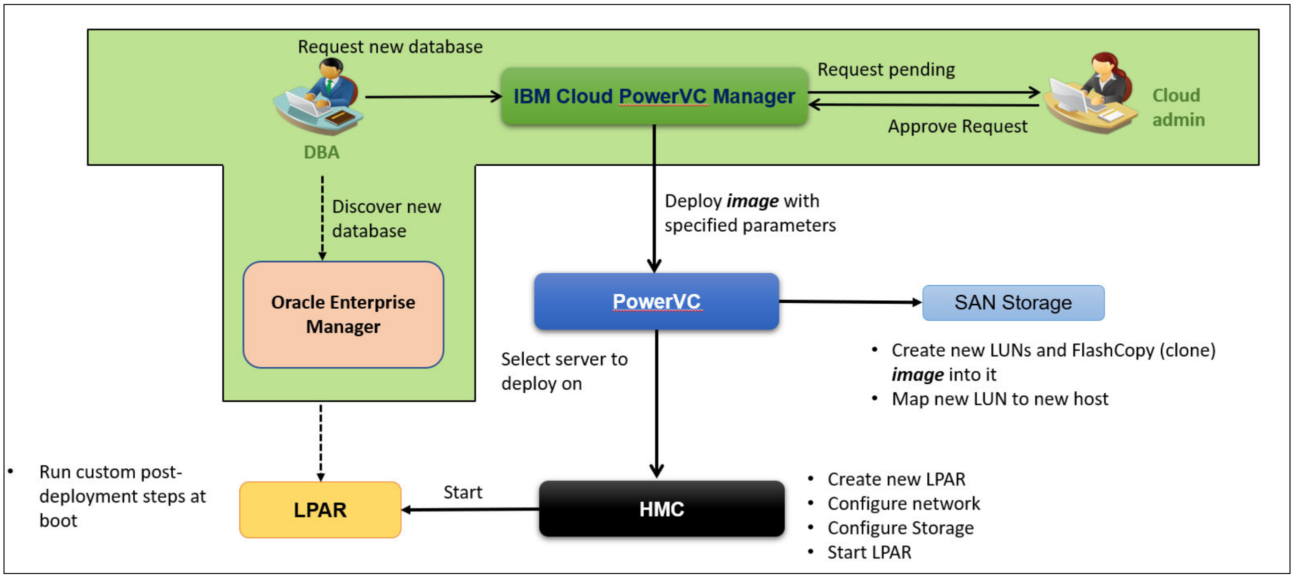

Request new database
DBA
IBM Cloud PowerVC Manager
Request pending
Approve Request
Cloud admin
Discover new database
Deploy *image* with specified parameters
Oracle Enterprise Manager
PowerVC
SAN Storage
Select server to deploy on
• Create new LUNs and FlashCopy (clone) *image* into it
• Map new LUN to new host
• Run custom post-deployment steps at boot
LPAR
Start
HMC
• Create new LPAR
• Configure network
• Configure Storage
• Start LPAR

Figure 3-9 Deployment of Oracle with IBM PowerVC

For more information about how to use IBM Terraform and Service Automation (instead of PowerVC) to provide the control point and user interface for DBaaS for an Oracle Database, while reusing existing PowerVC image, see this IBM Support web page.

GitHub source code is at this web page.

This image approach for provisioning limits the operating system and database versions you offer to users because it increases the number of images to build and maintain. Decomposing the steps to build a deployed Oracle Database environment into several steps results in longer deployment times, but allows the reuse of parts in other services and provides higher flexibility and customization to the user of the DBaaS offering.

Terraform is an Open Source tool that was created by HashiCorp to handle provisioning of cloud and infrastructure resources. These resources are described to Terraform by using a high-level configuration language that is stored in configuration files or templates. From these templates, Terraform generates an execution plan that describes its process and then runs it to build the described infrastructure.

Figure 3-10 shows the decomposition of an Oracle DBaaS on AIX service to deploy an Oracle Database on-premises or in an IBM Power Systems Virtual Server with the capability to customize each of the steps based on user input.

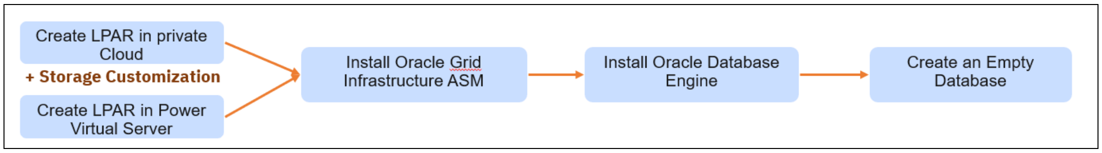

Create LPAR in private Cloud
+ Storage Customization
Create LPAR in Power Virtual Server
Install Oracle Grid Infrastructure ASM
Install Oracle Database Engine
Create an Empty Database

Figure 3-10 Dividing an Oracle DBaaS to deploy it on an IBM Power System Virtual Server

Assets of the related sample source code and scripts that illustrate this modified approach to provide a DBaaS service are available in a public GitHub repository.

Important: You must decide in advance what you want to include in the image and what you want to make customizable. For instance, we decided to include in the base AIX image the Oracle user and group creation in addition to applying our AIX and Oracle best practices settings to have a ready to use AIX image for Oracle. This operating system configuration and customization alternatively can be implemented by way of a post-deploy scripted template that is responsible to set AIX and Oracle prerequisites and best practices.

The first part of this DBaaS provisioning workflow is creating the LPAR. Depending on what the user selects, the corresponding template is called to create the LPAR on Power Systems on-premises or in Power Virtual Server (off-premises) by using an image with a specific version of AIX and all Oracle Prerequisites and best practices set.

The number and size of extra volumes that host the Oracle DB was integrated to this LPAR Creation Template to customize to DBA requirements for this new environment the storage layout.

The second part of this provisioning is about installing Oracle Grid Infrastructure and the configuration of Oracle ASM with custom disks settings. Oracle Home Path can be set by default or modified by the user in addition to Oracle ASM Instance Password and Oracle software version to install. This example offers the installation of an Oracle 18c or 19c version.

A similar template was developed to install the Oracle Database Engine and offer similar customization and choice; for example, Oracle Home Path of the Oracle DB installation directory and Oracle DB Version to install.

Last is the creation of an Oracle Database Standalone Instance. We defined the following a set of input and output parameters for each template:

- Input parameters define the customization that you want to bring and offer regarding the execution of each template. The list is not exhaustive and you can easily extend and replace some parameters that we defined to customize the Oracle software component installation.
- Output parameters are the result of creation tasks during the execution of the template and can be used by other templates that are run after it.

More parameters can be set for more customization and fit with your own requirements, constraints, and needs.

For more information about Terraform and IBM Power Systems Virtual Server code, see IBM Documentation.

The example that is available at this web page shows a new Oracle Database that is created on-premises or off-premises in IBM Power Systems Virtual Server. It also shows the orchestration of those independent Terraform templates combined to create such DBaaS for the user. It also addresses the deployment of an application tier followed by its database tier.

3.6 Red Hat Ansible

Red Hat Ansible is a simple automation language (based on Python which serves as the basis for Red Hat Ansible and uses YAML as its configuration syntax), is easy-to-learn, and is self-documenting. It can be used to update programs and configuration on hundreds of servers at once, but the process remains the same.

Red Hat Ansible is agentless and does not require agents to be installed on your target servers. It connects through the secured SSH protocol to run its tasks.

Red Hat Ansible is an open source software and easy to install by way of Yum. It can be supported by a subscription from Red Hat.

IBM created an extensive set of Red Hat Ansible modules for the Power Systems user community, ranging from operating system management to Cloud Management and everything in between. You can use those modules to codify key maintenance and operational tasks for AIX and the software stack so that you can focus on business priorities (see Figure 3-11).

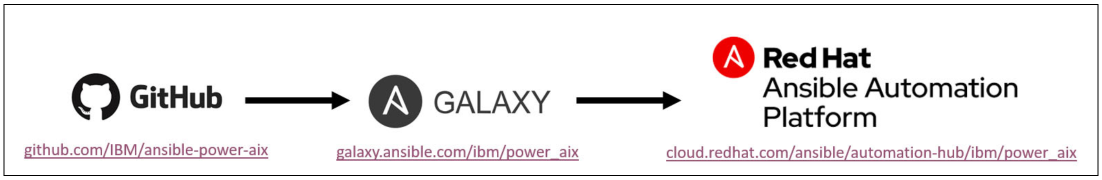

GitHub
GALAXY
Red Hat
Ansible Automation Platform
github.com/IBM/ansible-power-aix
galaxy.ansible.com/ibm/power_aix
cloud.redhat.com/ansible/automation-hub/ibm/power_aix

Figure 3-11 Red Hat Ansible Automation Platform

A notable entry to rapidly jump-start into your Red Hat Ansible project can be achieved with Red Hat Ansible Galaxy, as shown in Figure 3-11. Galaxy is a no cost site for finding, downloading, and sharing community developed roles.

For more information about the Community Ansible Collection for IBM Power Systems, see this web page.

You can download the Supported Red Hat Ansible Collection for IBM Power Systems from Automation Hub (Red Hat Ansible subscription required). For more information, see the following resources:

- Ansible Automation Platform Certified Content
- Red Hat Automation Hub (log in required)

The Red Hat Ansible experience is identical across POWER and x86 servers. The same steps can be repeated in the IBM Power Systems Virtual Server, public clouds environments, and on-premises. Some Red Hat Ansible modules or playbooks can be platform or operating system specifics. Then, customers can develop their own playbook to build a platform-independent management solution.

In the following section, we describe an Oracle DBaaS on AIX and Power Systems that uses Red Hat Ansible modules. The example relies on the similar workflow that is used with the IBM Terraform and Service Automation example, as shown in Figure 3-12 on page 30.

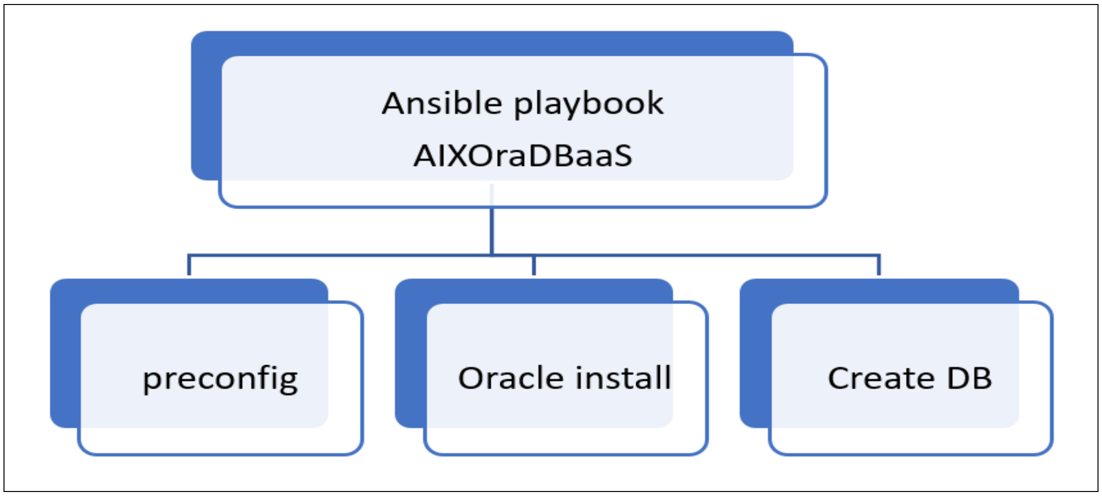


Figure 3-12 Oracle on AIX and Power Systems Red Hat Ansible Playbook example

This Red Hat Ansible playbook assumes that the AIX LPAR is created. It can be extended by creating the AIX LPAR by way of another Red Hat Ansible module. Such corresponding Red Hat Ansible module is in charge of the use of OpenStack APIs with PowerVC or Power Virtual Server APIs to create the AIX LPAR that hosts the Oracle Database.

The deployment of an Oracle Database was decomposed into several Red Hat Ansible modules to allow flexibility and re-use on those independent modules into other playbooks:

- `preconfig`: This role performs AIX configuration tasks that are needed for Oracle installation. It sets all AIX and Oracle best practices and prerequisites.
- `oracle_install`: This role performs the Oracle binary installation.
- `oracle_createdb`: This role creates the database instance by using the Oracle `dbca` utility and custom parameters, such as DB password and DB SID.

Note: For more information about a playbook and roles project on development, see this web page.

You can reuse them as-is to start the first Oracle Deployment on AIX by using Red Hat Ansible and then, customize and enhance them per your own requirements.

3.7 Other orchestration solutions

Other Infrastructure as Code tools or other orchestration solutions, such as Puppet, Chef, and VMWare vRealize, also are available.

Deploying Oracle DBaaS with vRealize Automation 7.2, vRealize Orchestrator and IBM PowerVC illustrates the deployment of an Oracle DBaaS with vRealize Automation and vRealize Orchestration and IBM PowerVC. It uses most of the concepts and deployment logic that were introduced with PowerVC and IBM Terraform Automation and Services examples.

4

Migration best practices

Migration is a topic that eventually must be considered. Even when performing only a platform refresh while staying on the same platform, the database must be moved. When changing the platform, for example, from HP-UX or Solaris to AIX, more steps than just copying the data files are necessary.

This chapter describes different methods to complete a migration. It also presents other issues that must be considered and describes the arguments for and against the different methods.

This chapter includes the following topics:

- 4.1, “Motivation” on page 32.
- 4.2, “Migration plan” on page 33.
- 4.3, “Migration methods” on page 35.
- 4.4, “Combined method for optimized migration” on page 40.
- 4.5, “Preferred methods for different scenarios” on page 42.

4.1 Motivation

Migrating a database appears to be a straightforward process. However, when analyzed critically, it can become a complex undertaking when one or more of the following conditions exist:

- Source database is large
- Source and target operating systems are different
- Source and target systems are geographically remote
- Migration window time frame is limited or nonexistent
- Downtime to perform test migrations is difficult or impossible to allocate

Migration methods can be sorted into the following categories:

- Logical data migration:
  - Transferring the logical objects, such as tables, that constitute the source database onto the target system.
  - Only table data is transferred; indexes are rebuilt on the target.
  - Conversions (for example, endianness change) is completed dynamically.
- Physical data migration:
  - Transferring the operating system level objects (for example, files in the file system or UNIX *raw* disk partitions) that constitute the source database onto the target system.
  - This form of data encapsulation involves a *byte-by-byte* copy from the source to the target.
  - Some (or all if endianness change occurred) data files must be converted before they can be used on the new platform. Therefore, you cannot restore a HP-UX backup on an AIX target.

During migration, you need downtime for the application. Depending on the size of the database (that is, the amount of data that must be copied) and the available downtime window, you must carefully plan the migration.

The migration process does not include that only the database is moved to the new platform. It also includes at least one (but can be more than) the following tasks:

- Analyze the source environment
- Evaluate and select a migration method
- First test migration plus application test
- Second test migration plus application test
- Live migration

Plan for ample preparation time. Especially if several databases and departments are involved because migration is not a task; rather, it is a project.

4.2 Migration plan

This section describes the migration procedure.

4.2.1 Analyzing the source environment

To plan the migration, you must understand your source environment. Gather at least the following information:

- List of all databases, including:
  - Size
  - Version, installed patches
  - Stakeholders, available downtime
- Platform (source and target):
  - Operating system, release
  - Are source and target the same or different endianness
- Infrastructure:
  - Available network connectivity between source and target
  - Storage space availability (for target but also for possible interim copies)
  - Available other *source platform* servers that can be used to create database mirrors to assist in the actual migration

> **Note:** Key aspects also include rollback options and point of no return in the migration process.

Endianness

Endianness refers to the order in which the processor stores and interprets data. Systems are big-endian or little-endian. *Big-endian* systems store the most significant byte of multi-byte data at the first memory address and the least significant byte at the last memory address. *Little-endian* system store data the opposite way (see Figure 4-1).

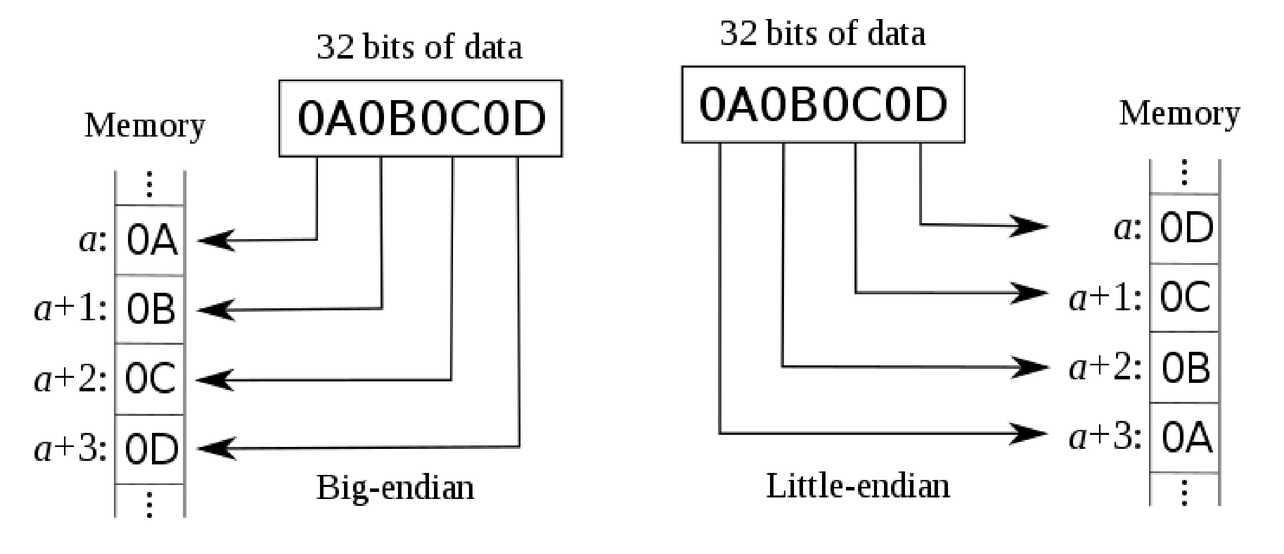


Figure 4-1 Big versus Little endian, Source: Wikipedia.org

Oracle gives you a list of which endianness is used for each platform, as listed in Table 4-1.

Table 4-1 Endianness in Oracle

select platform_name from v$transportable_platform where endian_format = 'Big';	select platform_name from v$transportable_platform where endian_format = 'Little';
► Solaris[tm] OE (32-bit) ► Solaris[tm] OE (64-bit) ► AIX-Based Systems (64-bit) ► HP-UX (64-bit) ► HP-UX IA (64-bit) ► IBM zSeries Based Linux ► Apple Mac operating system ► IBM POWER Based Linux	► Microsoft Windows IA (32-bit) ► Linux IA (32-bit) ► HP Tru64 UNIX ► Linux IA (64-bit) ► HP Open VMS ► Microsoft Windows IA (64-bit) ► Linux x86 64-bit ► Microsoft Windows x86 64-bit ► Solaris Operating System (x86) ► Solaris Operating System (x86-64) ► HP IA Open VMS

4.2.2 Evaluate and select a migration method

Several methods are available to migrate an Oracle Database. For logical and physical data transfer, you can make the following distinctions:

- Methods where all data is transferred during downtime.
- Methods with replication of data while the application can be up and running, which means during downtime. you do not have data transfer over the network. However, you might still need to complete conversions.
- Incremental methods in which an initial online copy is done while the application is still up and running, followed by arbitrary incremental online copies where only data changed since last copy is transferred, with a final incremental copy during downtime.

The choice is always a consideration between effort (preparation, planning, performing the migration), tool cost (especially because logical replication methods are often costly), size, and available downtime and network speed.

You must script all migration steps to reproduce them (first and second test and live migration), especially when migrating a complex environment, a higher number of databases or critical systems.

For a hardware refresh while staying on the same platform, you also must plan your approach.

> **Note:** You must be aware that the use of Live Partition Mobility to move an LPAR to a new Power System is technically feasible and simple. However, Oracle Licensing terms state that when you use Live Partition Mobility, you must purchase Oracle licenses for all CPU cores in the old and new server. Therefore, this method often cannot be used for migration.

4.2.3 First test migration

The reason to perform test migrations is not only to test whether the application works after moving to the new platform. The first test migration also allows you to verify the migration approach, test the scripts that are developed for the migration, and assess the timing for migration operations. If the needed downtime is too large, you must optimize or reconsider your migration method or workflow.

If any issues are revealed during the first test migration (error in scripts that must be fixed, errors in the migrated database during application testing, and so on) you fix the scripts and perform the second test migration.

Depending on the migration method that is used, you might need downtime for your source system, which also applies to all future test migrations.

4.2.4 Second test migration

The second test migration proves that the lessons learned from the first test migration result in a smooth migration with no need for script adjustments and within the time that is assessed during the first test migration. Also, extensive application testing (functional and non-functional, such as performance) must be done late in this stage. A complete backup and restore cycle also is part of the tests and verification of any high availability and disaster recovery functions.

If any issues are revealed, consider repeating this step that is, the third test migration) to avoid surprises during live migration as much as possible.

4.2.5 Live migration

This migration is the final move to the new platform.

Applications must be shut down. After the migration, you must (depending on the allotted time) run basic or comprehensive functional application tests before you decide whether to proceed. It is important to start your first full backup at this point.

4.3 Migration methods

This section presents an overview of different migration methods. This list is not exhaustive because many tools are available that support migrations. However, most tools are based on methods that are described here or use a combination of different methods.

If not stated otherwise, you need downtime for the *entire* process.

4.3.1 Traditional export /import

This method is a logical data transfer method. The **exp**/**imp** command-line tools connect from the client over the network to the database. Data is written to a file on the client host/LPAR. The transfer speed is limited and no build-in parallelism is available. Only table data is transferred; indexes must be re-created.

You also must create an empty target database, including all needed table spaces. The **Imp** command imports only the application schema or data into the new database. SYS/SYSTEM level objects are *not* transferred.

For traditional export, the following optimizations are possible:

- You can write the export to an UNIX named pipe (FIFO) and directly run the import to read from this pipe. Therefore, no temporary storage is needed for the dump, no extra copy of the dump files over network is needed (however, data is transferred over the network to and from the `exp/imp` client), and import can start before export is completed.
- You can configure the **exp** command to export only one table. By using this feature, scripts can be created that can be run in parallel. Therefore, several `exp/imp` processes can run at the same time, even though `exp/imp` has no direct implementation of parallelism. However, you need good planning to take care of foreign Key constraints or you must import without constraints and after all tables are finished.
- At the end of the **imp** process, indexes are created. You can skip this step and create scripts that start index creation (in parallel) when each table is finished with data transfer.

If small databases are being migrated, this migration method is easy and quick to set up. For any reasonably sized database (that is, for medium to large size databases), this approach is too slow.

4.3.2 Oracle Data Pump tool

Starting with Oracle 10g, Oracle Data Pump is available as an export/import tool. It also is a logical data transfer method.

The **expdp/impdp** command line tools connect from the client over the network to the database. However, they start only the export/import process.

Data is written to a directory on the database server that is defined in the database. It also is possible to run the **impdp** command in a way that the export is done directly over a database link between the target and source database, which saves time and space to perform the migration.

Also, the transfer speed is considerably higher compared to `exp/imp`. Build-in parallelism is available. as with the traditional export /import method, only table data is transferred; indexes are re-created.

An empty target database also must be created, including all needed table spaces. SYS/SYSTEM level objects are *not* transferred.

With Oracle Data Pump, you cannot use UNIX named pipes (FIFO) to run import while export is still running. Data is written to a file system on the database server (into several files when parallelism is used).

However, you can write data to an NFS file system, which is exported from the target database server. Therefore, no extra copying of the dump after export finishes is needed; import can start immediately.

Oracle Data Pump can be used with manageable effort for medium-sized databases when adequate downtime is available.

4.3.3 Data transfer over Oracle named network links

Networks are available that are identified to Oracle for a specific purpose. In Oracle, you can define network links between databases, which allows data to be migrated by using the following approach:

1. Create an empty target database with all needed users and table spaces.
2. Create a database link from target to source (name, for example, *source*).
3. Repeat for each table:
   a. On target, run a statement, such as the following example:

      ```
      create table x [ tablespace yyy / storage clause ] as select * from
      x@source;
      ```

   b. Run statements to create indexes.
4. After all tables are transferred, create (foreign key + other) constraints and grants that are defined on the source.

Specific data types (for example, LONG/LONG RAW) cannot be queried by using a database link. If such data types exist for a table, you can combine them with an `exp/imp` or Oracle Data Pump for those tables.

This approach also can be classified into logical data transfer methods.

4.3.4 Logical replication

Several tools are available that can be used to perform logical replication, such as Oracle GoldenGate or IBM InfoSphere® CDC.

Changes to the database are extracted from the log files and (depending on configuration) transferred (modified or unmodified) to another database, or skipped. By first creating a database copy as of time stamp `x` (which can be done, for example, by using a time consistent data pump dump) and then transferring all changes, the target database can be put (and kept) in sync while the source database is still in use.

This process results in a reduced downtime migration because only replication must be stopped and the application must be configured to access the target database. However, the tools are mostly expensive.

Note: Oracle Data Guard running as a Logical Standby cannot be used in this context because to set it up, you first must create a Physical Standby that later can be converted into a Logical Standby. Therefore, the constraints of setting up an Oracle Data Guard mirror still apply.

4.3.5 Oracle Data Guard

This function is part of Oracle Enterprise Edition. It allows you to create a 1:1 mirror of your database. Because source and target databases must (for most operating systems) be the same, Oracle Data Guard often cannot be used for cross-platform migrations. However, when refreshing hardware within the same architecture (for example, moving from POWER8 to POWER9), this method can be used to create and synchronize a copy of the database online.

For migration, only sync must be disabled, the mirror must be “activated”, and the application must be pointed to the new database.

My Oracle Support, Document 413484.1 provides an overview of which Cross-Platform combinations are supported. The following combinations are the most important:

- Solaris (SPARC): AIX
- Windows (x86): Windows (x86-64), Windows (Itanium), Linux (x86), and Linux (x86-64)
- Linux (x86-64): Solaris (x86-64)
- Solaris (x86): Solaris (x86-64)

If source and target are the same or one of these combinations, Oracle Data Guard is a good method to use to perform the migration.

Another use of Oracle Data Guard is to obtain o generate a copy of the database that can be used for test migrations. Because the source database must be down for migration, it can be difficult to perform test migrations for databases that must be highly available.

However, if you first create an Oracle Data Guard mirror for migration, you can stop replication, activate the copy database, and then, perform migrations steps.

4.3.6 Transportable Database

This method is preferred if the source and target are different platforms but use the same endianness. The basic approach includes the following steps:

1. Shut down the source database.
2. Copy data files from source to target. (For more information about how to copy files between different ASM instances, see My Oracle Support, Document 785580.1).
3. Convert some of the data files. All data files that contain local rollback segments that must be converted. Many times, those data files are only the SYSTEM and UNDO table spaces, which often are small. Therefore, conversion time is relatively low.
4. Create a database (plus control files) from the copied and converted data files.

For more information about this approach, see this Oracle white paper.

Note: The procedure that is used in this document creates a convert script (`convert_mydb.rman`), which includes all data files, not only those files that include rollback segments. Edit the script to remove any surplus data files.

Another optimization is to allocate more than one disk channel so that all data files are converted in parallel. This process saves overall conversion time.

The following SQL query is used to identify the data files with rollback segments:

```
select distinct(file_name)
from dba_data_files a, dba_rollback_segs b
where a.tablespace_name=b.tablespace_name;
```

For more information about this approach, see My Oracle Support, Document 732053.1.

One optimization that can be done when the target database uses a file system (not ASM) is to create a Data Guard mirror in which the file system from the target server that finally contains the data files is NFS-exported to the server that is running the Data Guard instance.

By using this approach, all data files are at their final location after the Data Guard mirror is in sync. This result eliminates the time that is needed to copy the data files from source to target.

4.3.7 Cross-Platform Transportable Tablespace

This method is similar to Transportable Database, but can be used even when the source and target platform use different endianness. The basic approach includes the following tasks:

1. NFS-mount the source datafile directory to the target. If source database files are in ASM, you can set up a Data Guard mirror that stores the data files on the file system of an interim server.
2. Create an empty target database.
3. Restart source database in READ ONLY mode.

   This time, all the data files must be converted. However, as an optimization, files are read from the NFS mount so that the data transfer and conversion are running at the same time. You also can allocate several disk channels to convert several files in parallel.
4. Export/Import table spaces into the target database.
5. Perform Post-Transport actions, as described in see this Oracle white paper.

4.3.8 Full Transportable Export/Import

Full Transportable Export/Import is a new feature of Oracle 12c (and later). It is based on Transportable Tablespace; however, the number of extra manual steps that are involved was reduced, which makes it easier to perform the migration.

The high-level necessary steps are similar to the steps that are described in chapter 4.3.7, “Cross-Platform Transportable Tablespace”.

For more information about this approach, see this Oracle white paper.

4.3.9 Cross-Platform Incremental Backup

All methods involving endianness conversion share the issue that conversion must be performed during downtime. If the database is large and the downtime window is short, not enough time might be available to perform the conversion.

Even when data files are on the target platform (by using a DataGuard mirror with NFS) and several channels are used to convert data files in parallel, you still might not realize the needed throughput because it is limited by CPU and storage performance.

Imagine having a 20 TB database and a downtime window of 2 hours. Even if all actions are scripted, this maximum of 80 minutes are available for conversion. Therefore, each second (that is, `20*1024/(60*80) = 4.3 GB`) must be read and written, for a total of 8.6 GBps on average over the entire conversion time.

It might be a challenge to achieve this result.

Cross-Platform Incremental Backup is a method to perform most of the transfer and conversion work while the source database is still in use. This method includes the following steps:

1. Create an empty target database.
2. Take a Level 0 backup of the source database, and restore that data files on target in correct endianness format.
3. Repeat the following steps as often as needed:
   a. Take incremental backups.

   b. Transfer to target.
   c. Convert to correct endianness format.
   d. Apply to destination data files.

4. During downtime:

   a. Set source database to READ ONLY mode.
   b. Perform one last incremental backup, conversion, and apply.
   c. Export/Import metadata.
   d. Set target database to READ WRITE mode.

The needed downtime mostly depends on the number of data changes that were made to the source database since the last incremental backup. If fewer changes were done, the process takes less time, especially if compared to a full conversion during downtime.

The time that is needed for the incremental backups can be further reduced by using Block Change Tracking. By using this method, only the data blocks that changed since the previous backup are read.

For more information about this approach, see My Oracle Support, Doc ID 2471245.1.

Cross-Platform Incremental backup is not a complete migration method, but it can be combined with other methods, such as Transportable Database, Cross-Platform Transportable Tablespace, or Full Transportable Export/Import.

4.4 Combined method for optimized migration

Consider the following scenario:

1. An endianness change is necessary (for example, migrating from Linux to AIX).
2. No downtime of source database possible during test migrations.
3. Target environment must be protected by a Data Guard mirror (which is set up during migration also, not after).
4. All databases are version 12.1or later.

To address those requirements, we used the following approach:

1. We choose Cross-Platform Incremental Backup to take care of endianness change and to have a low downtime for large databases.
2. To perform test migrations without interrupting the source database, we set up a Data Guard mirror of the source database. Migration tests can then be performed from that database. Redo log apply must be stopped. We created a guaranteed restored point to flashback the Data Guard mirror after test migration so it can be synced again. Then, we activated and open the mirror. Migration is then performed from there on.
   After migration is finalized, the mirror can be flashed back to the restore point and resynced (assuming all archive logs can be kept).
3. On the target environment, an empty database must be created. The empty database is used to set up a mirror with Data Guard. The restore and incremental updates are applied to both target servers.
4. The final migration step is performed with Full Transportable Export/Import (FTEX). The import of metadata only must be done in the primary target database because it is to be replicated to the standby database.

Unfortunately, the scripts that are provided by Oracle (My Oracle Support, Doc ID 2471245.1) do not support running against a Data Guard Standby database.[1]

Prepare your environment (including scripts as described in My Oracle Support, Doc ID 2471245.1) by using the IBM Migration Factory enhanced versions (see Figure 4-2):

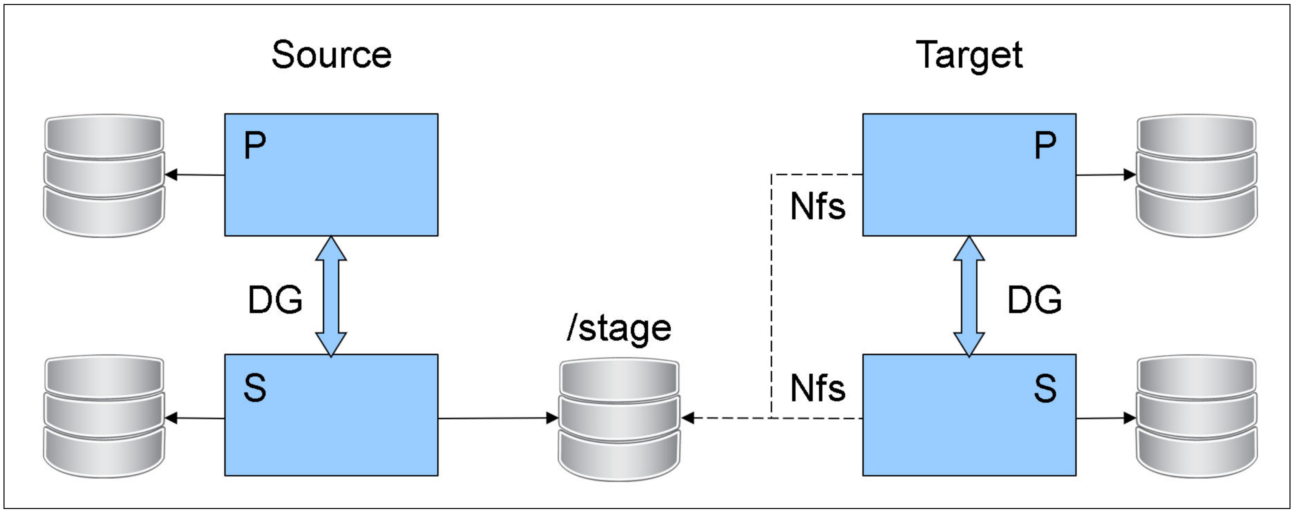


Figure 4-2 Combined method

1. Setup:
   a. Run the **/stage** command to hold the storage for the backup files and shared configuration files.
   b. On the source side. create standby storage and wait for sync. Activate block change tracking.
   c. On the target, create an empty database and Data Guard mirror.
2. Prepare and rollforward:
   a. On source (standby), run **xttdriver.pl --backup** (script from My Oracle Support).
   b. On target (primary + standby), run **xttdriver.pl --restore**.
   c. Repeat this process as many times as needed to keep the target as closely in sync with the source as possible.
3. Transport (downtime is necessary if final migration, or stop sync to standby for test migration):
   - Source:
     - Alter tablespaces to read only.
     - **xttdriver.pl --backup** (last time).

       ```
       (FTEX): expdp system/xxx full=y transportable=always [version=12]
       ```

[1] But we found a way to enhance them to make this scenario possible by creating a version that will run certain commands against the target primary database instead but gets the critical data from the source standby database by way of a database link. Contact the Systems Lab Services Migration Factory for more details about this approach: https://www.ibm.com/blogs/systems/tag/ibm-systems-lab-services-migration-factory/

- Target:
  - `xttdriver.pl --restore` (last time).
  - Drop tablespaces to prepare for import of data files.

    ```
    (FTEX): impdp system/xxx ... transport_datafiles=‘...‘,‘...‘ ...
    ```

Using NFS mounts where suitable avoids extra copy steps.

4.5 Preferred methods for different scenarios

The different migration methods that were described in 4.5, “Preferred methods for different scenarios” on page 42 feature different scopes to where they can be applied. The following important criteria for selection of the migration method are used:

- Platform and operating system of source and target
- Endianness of source and target
- Database version
- Downtime requirements

Table 4-2 lists some suggestions on which migration methods can be recommended for different scenarios. However, some issues might need to be considered based on each case’s specifics.

Table 4-2 Preferred migration methods

Scenario	Preferred method	Possible enhancements	Typical database downtimes
Same platform (or supported exception).	Data Guard (DG) between source and target (9i+).	N/A	Several minutes.
Different platform / same endianness.	Transportable Database (TDB) (10.2+).	Source standby (on NFS from target if on file system); Incremental Backup; target standby.	Less than one hour with enhancements.

Different platform / different endianness.	Full Transportable Export/Import (FTEX) (12c+) /
	Cross Platform Transportable Tablespaces (XTTS) (10g+).
		Source standby;
		Incremental Backup;
		target standby.
			Less than one hour with enhancements.

Nothing else works or database is small (<1 TB).	Data Pump (10 g+).	N/A	Data transfer time (parallel).
Last resort.	export/import.	N/A	Data transfer time (nonparallel).

Often, we do not prefer methods that are not includes in the table, but logical replication might be suitable in cases where small downtimes are required (less than one hour) and other methods do not work.

Migration Framework for large projects

As we described previously, even migrating a single database can be a complex task in some environments. Because most migration projects must process many databases (sometimes hundreds), you can face more challenges, including the following examples:

- Many migrations must be planned and performed according to an agreed schedule.
- Different migration methods must be used for different groups of databases.
- Each migration method involves complex tasks, possibly spanning several days each.
- Overall project duration requires several migrations to be performed in parallel.
- Migration tasks and results must be documented.
- Short downtimes add pressure.

To deal with these challenges and to make a structured approach to complex migration projects, we developed the Migration Factory Framework.

We used the Migration Factory Framework (including some earlier versions) successfully for several large migration projects. However, it is still a work in progress that we intend to enhance and improve further over time.

4.5.1 Components of the Migration Factory Framework

The Migration Factory Framework is a set of tools that are used to improve the migration project experience compared to a standard copy-and-paste, command-line approach.

This approach is based on Red Hat Ansible playbooks; therefore, it does not need any agent programs to be installed on the source or target environments except for python. It runs on a Linux workstation (called the Control Workstation) with Red Hat Ansible and Python3 that is installed that can access the source and target machines by way of SSH as an unprivileged user.

This user then needs the access to run commands on the source or target machines as oracle/grid through sudo with no password per standard Red Hat Ansible practice.

The Migration Factory Framework includes the following main components:

- Configurable Menu program
- Collection of Menu configurations for different migration workflows
- Collection of Red Hat Ansible Playbooks for the migration steps
- Helper programs to configure the Red Hat Ansible inventory files

These main components are described next.

Menu program

The Menu program is the front-end part of the Migration Factory Framework. It is used by the migrators to perform the steps in the migration workflow as it is defined in the Menu configuration files. It is a python3 program with a simple character-based user interface.

Each migration includes a line in a database list file that contains information, such as database name, migration type (trial or live), which menu file to use (for example, which workflow), and more. When starting the program, the user is prompted to choose a database to be migrated. If the database is found in the database list file, the corresponding line is locked exclusively for this user.

If another user attempts to work on the same database while the line is locked, it does not work.

The user is then shown a section from the workflow with the next available step, which is marked with an arrow and a few choices, including to run the next step (see Example 4-1).

Example 4-1 Menu Program after start

```
[np@akoya migration_factory]$ ./mfg_menu2.py

            __
 _ __ ___  / _| __ _
| '_ ` _ \| |_ / _` |Welcome to the mfg migration menu
| | | | | |  _| (_| |
|_| |_| |_|_|  \__, |Version 2.1 (2020-06-23)
               |___/

Which is your dblist file? mfg.local

Which database do you want to work with? cdb1/pdb1

Your project is HUGO6

{'#project': 'HUGO6', 'db_name': 'cdb1/pdb1', 'migration_type': 'T', 'menu_file':
'mfg_menu_FTEX_backup', 'TID_list': '1,2,3,4,5,6', 'TID_times':
'202006081601,202006081602,202006081603,202006081604,202006081605,202006081606',
'db_version': '12.2.0.1'}

Do you want to continue (yes/no)? y

>>>>> INFO: Locking project HUGO6 for user np... [OK]

--- Main Menu

 --->       1 A   a  -- Start new Task A
 ....       2 A   m  Setup source reference and target
 ....       3 B   a  -- Start new Task B
 ....       4 B   m  Prepare target database
 [more]

*** TRIAL *** Project: HUGO6 - max: 13 - last: 0 - result: 0 - next: 1

1) next    2) last    3) info    4) quit    #?
```

If a step is run successfully, the arrow advances to the next step in the workflow.

If a step fails, the failure is indicated by a crossed out arrow, and the workflow does not advance. The problem must be fixed and then the failed step resubmitted (however, this is an exception).

The steps in the workflow also can be grouped into larger units, which are called *tasks*. In the database list file, each task can be associated with a specific date and time when this task must be run.

Although this process does not occur automatically, when the user starts a new task in the Menu program, the date and time are checked, and the user must specifically agree whether the task must be run in advance. This check is useful for the beginning of the downtime, for example, or when the workflow must pause to allow for some manipulation outside of the workflow.

Each task also includes a task ID that is associated to it, which must be entered by the user before a new task starts. This ID can be used, to synchronize the workflow to a change management system, for example. It also prevents the user from entering a new task accidentally by quickly progressing through the menu (although this is strongly discouraged because the user is expected to read through the output of each step carefully to spot any errors that are not caught by the logic in the playbook).

The user also czn decide to quit the menu after each step. In this case, the status of the migration is stored in a status file and the migration is unlocked in the database list. Another user (or the same user) later can start the Menu program and, after choosing the database, automatically continue at the same step where the last user quit the menu.

Users also can work at different migrations at the same time by starting multiple copies of the Menu program in different terminal windows and then, choosing different databases from the database list file.

Menu configurations

The Menu program workflows are fully configurable. You can define the menu structure, including submenus, tasks, and the individual steps, as shown in Example 4-2. For each step, a corresponding Red Hat Ansible playbook is available that is started when the step is run. We included sample workflows for some available migration methods and we plan to create a more comprehensive collection.

Example 4-2 Menu configuration file for FTEX_backup workflow

```
#
#:t:task:[!:]
#:m:menu "Menu Text":
#:a|aT|aL:a|c:action "Action Text":
#

:m:MAIN "Main Menu":
{

:t:A:
:m:install "Setup source reference and target":
{
:a:a:xtt_setup "Setup scripts for xtt":
}

:t:B:
:m:prep_target "Prepare target database"
{
:a:a:ftex_drop_ts_target "Drop tablespaces in target database"
}

:t:C:
:m:increment "Incremental backup and restore"
{
:a:a:xtt_backup_reference "Take incremental backup from reference node"
:a:a:xtt_transfer "Transfer backup info to target"
:a:a:xtt_restore "Apply incremental backup to target"
}

:t:D:
:m:conv2snap "Convert source standby to snapshot standby"
{
:aT:a:dg_convert_snapshot "Convert source standby to snapshot standby"
}

:t:E:!:
:m:ftex "Perform FTEX migration
{
:aL:a:ftex_ts_ro_primary "Set tablespaces to read only on primary"
:aT:a:ftex_ts_ro_reference "Set tablespaces to read only on reference node"
:a:a:xtt_backup_reference "Take final backup from reference node"
:a:a:xtt_transfer "Transfer backup info to target"
:a:a:xtt_restore "Apply final incremental backup to target"
:a:a:ftex_drop_ts_target "Drop tablespaces in target database"
:a:a:ftex_apply "Perform FTEX migration"
}

:t:F:
:m:conv2phys "Convert source standby to physical standby"
{
:aT:a:dg_convert_physical "Convert source standby to physical standby"
}

}
```

Example 4-2 on page 45 shows the menu configuration file for the workflow of the “Combined Method”, as discussed in 4.4, “Combined method for optimized migration” on page 40.

Red Hat Ansible playbooks

Each step in the migration workflow is associated with Red Hat Ansible playbook. The correspondence is defined by way of the action items in the Menu configuration files. A set of Red Hat Ansible playbooks is available for some migration methods and we plan to make playbooks available for more migration methods or variations.

When a playbook is run through the menu program, all output is displayed in the window and checked by the user for errors that occurred but were not caught by the logic of the playbook. For later reference (and as a documentation aid), all output also is collected in log files in the respective project directory.

Helper programs

Helper programs are available to help the preparation of the inventory file for the Red Hat Ansible playbooks. When a Red Hat Ansible playbook runs, it needs an inventory file that contains the host names of the machines that are used in the migration and many variables that are needed by the playbooks.

The helper programs guides you through the process of creating the inventory file by asking some basic questions about the migration and then, providing you with a template or sample inventory file that might need some adaptation to your specific environment.

5

Recommendations for preparing the Oracle software installation

In this chapter, we provide suggestions for installing the Oracle software.

This chapter includes the following topics:

- 5.1, “Preparing to install Oracle Database on AIX” on page 48.
- 5.3, “AIX” on page 48.
- 5.4, “CPU (best practice)” on page 51.
- 5.5, “Memory” on page 54.
- 5.6, “DNS (RAC only; mandatory)” on page 55.
- 5.7, “Network best practices” on page 55.
- 5.8, “UNIX users and groups (mandatory)” on page 58.
- 5.9, “Storage” on page 59.
- 5.12, “Creating the database (best practices)” on page 68.

5.1 Preparing to install Oracle Database on AIX

The following configurations are available for installing Oracle on AIX:

- Standalone with JFS2 file systems
- Standalone with Automatic Storage Management (ASM)
- Oracle Real Application Clusters (RAC) with IBM Spectrum® Scale file systems (formerly known as GPFS)
- RAC with ASM

The simplest installation is a stand-alone Oracle install with JFS2 file systems. ASM requires the installation of Grid Infrastructure that has the additional overhead of an ASM database. When correctly configured the performance difference between JFS2 and ASM is negligible.

Implementing Oracle is a team effort with tasks for the DBA and the Systems, Storage, and Network administrators. Getting the prerequisites correct is key to a successful installation.

5.2 Firmware

Your firmware must be as up to date as possible. Consider the following points:

- Check the current firmware level by using the **`prtconf | grep "Firmware"`** AIX command from your LPAR.

  Example output:

  ```
  # prtconf | grep "Firmware"
  Platform Firmware level: VL940_027
  Firmware Version: IBM,FW940.00 (VL940_027)
  ```

- The latest firmware level is available at this IBM Support Fix Central web page.

5.3 AIX

The 19c database software is certified against AIX 7.1 and 7.2. The following minimum levels are required:

- 7.1 TL5 SP01
- 7.2 TL2 SP1

Note: Unlike older versions, Oracle 19c is not certified on AIX 6.1.

For best performance and reliability, it is recommended to install the latest server firmware and AIX TL and SP levels.

The Oracle installation documentation covers most tasks that are required to install Grid Infrastructure and the Oracle Database software.

The Oracle documentation does *not* cover all the best practices. Some steps are omitted that we cover in this publication for completeness or to highlight their importance.

Each point is marked as optional, best practice, or mandatory.

5.3.1 X Window System terminal (optional)

Unless you prepared a silent installation response file, you need an xWindows terminal.

One commonly used tool is TightVNC server. TightVNC server is not installed on AIX by default. However, you can find it as part of the AIX Toolbox for Linux Applications, which is available at no cost to download at this web page.

You also must install the TightVNC viewer or another VNC viewer on your desktop to access the session. Complete the following steps:

1. Connect as the `oracle` or `grid` user and start the VNC server. You are prompted for a password if this connection is the first time that VNC server was used.
2. Export the DISPLAY variable with the value that is indicated by the vncserver.
3. Run the **`xhost +`** command to allow external connections to the vncserver.
4. Connect to the vncserver by using the vncviewer you installed that uses the same host and port that the vncserver is running on. Enter the password that you set.

5.3.2 Unzip (mandatory)

Check the version that is installed by using the following command:

```
# rpm -qa unzip
```

The `unzip` must be version 6. You cannot work around this requirement with an old version of `unzip` because it cannot handle the file size. You cannot use **`jre`** to perform the extraction because the permissions are incorrect on the files and the `runInstaller` does not work.

`Unzip` also is included in the AIX Toolbox for Linux Applications. If you do not want to install the full toolbox, you can download the rpm for `unzip` 6 from this web page.

As of this writing, the latest version that was written is `unzip-6.0-3.aix6.1.ppc.rpm`.

Upgrade or install by using the following command:

```
# rpm -Uvh unzip-6.0-3.aix6.1.ppc.rpm
Preparing...                          ################################# [100%]
Updating / installing...
   1:unzip-6.0-3                      ################################# [ 50%]
Cleaning up / removing...
   2:unzip-5.51-1                     ################################# [100%]
```

`Ulimit` must be set for `root`, `grid`, and `oracle` users. Without the correct values, copying the binaries onto the file system can fail because the zip file size exceeds the default `ulimit` maximum file size. Setting the values to unlimited helps to avoid any issues, as shown in the following example:

```
chuser threads='-1' nproc='-1' fsize='-1' data='-1' rss='-1' nofiles='-1' root
```

Run the same command for `oracle` and `grid`.

5.3.3 Operating system changes (mandatory)

Update max processes by using the command:

```
chdev -l sys0 -a maxuproc=16384
```

By default, the input/output completion port (IOCP) is set to `Defined`. To enable IOCP, set IOCP to `Available` and perform the following commands as `root`:

```
mkdev -l iocp0
chdev -l iocp0 -P -a autoconfig='available'
```

You must restart the system to make the changes permanent.

Verify the settings by using the **`lsdev`** command to confirm that the IOCP status is set to `Available`:

```
# lsdev | grep iocp
iocp0 Available I/O Completion Ports
```

> **Note:** If IOCP was not defined, an AIX restart is required. Without this setting, the Oracle installer fails, but the Grid Infrastructure that is installed does not check the status of IOCP; however, your ASM disks are not available.

5.3.4 Space in /tmp for all partitions (mandatory)

At least 5 GB must be available in `/tmp`, or the `runinstaller` fails=.

5.3.5 Keep AIX updated (best practice)

You must keep focus on the AIX lifecycle and update it to remain supported and benefiting from fixes regarding your software stack certification matrix and your change management process. To check your operating system version or level, use the following command:

```
# oslevel -s
7200-03-03-1914
```

The output format is shown in the following example:

```
Base Level – Technology Level – Service Pack – Build sequence identifier.
```

> **Note:** You cannot apply updates that have a lower build sequence identifier.

If you update AIX, it is recommended to relink Oracle Home binaries. This process done as the `oracle` user by using the command **`relink all`**.

The latest AIX Technology Level (TL) or Service Pack (SP) can be downloaded from this IBM Fix Central web page (search for “AIX Technology Levels” to display the available levels).

5.4 CPU (best practice)

The preferred configuration for processors is dedicated if you are looking for maximum performance or dedicated donating if you want to reduce license costs by sharing capacity.

5.4.1 Configuring LPAR profile for Shared Processor LPAR performance (best practice)

If you use shared processors, correctly set and adjust entitled capacity (EC) and virtual processor (VP) settings (rule of thumb: up to 30% gap range between EC and VP settings) to mitigate CPU folding activity.

Check by using the **lparstat -i** AIX command output:

```
# lparstat -i
Type                                        : Shared-SMT-8
Mode                                        : Uncapped
Entitled Capacity                           : 1.50
Online Virtual CPUs                         : 2
Maximum Virtual CPUs                        : 2
Minimum Virtual CPUs                        : 2
Desired Virtual CPUs                        : 2
```

These lines are extract from the **lparstat** output that shows only the values that are discussed here.

5.4.2 Checking that Power Saver Mode is set to Maximum Performance Mode (best practice)

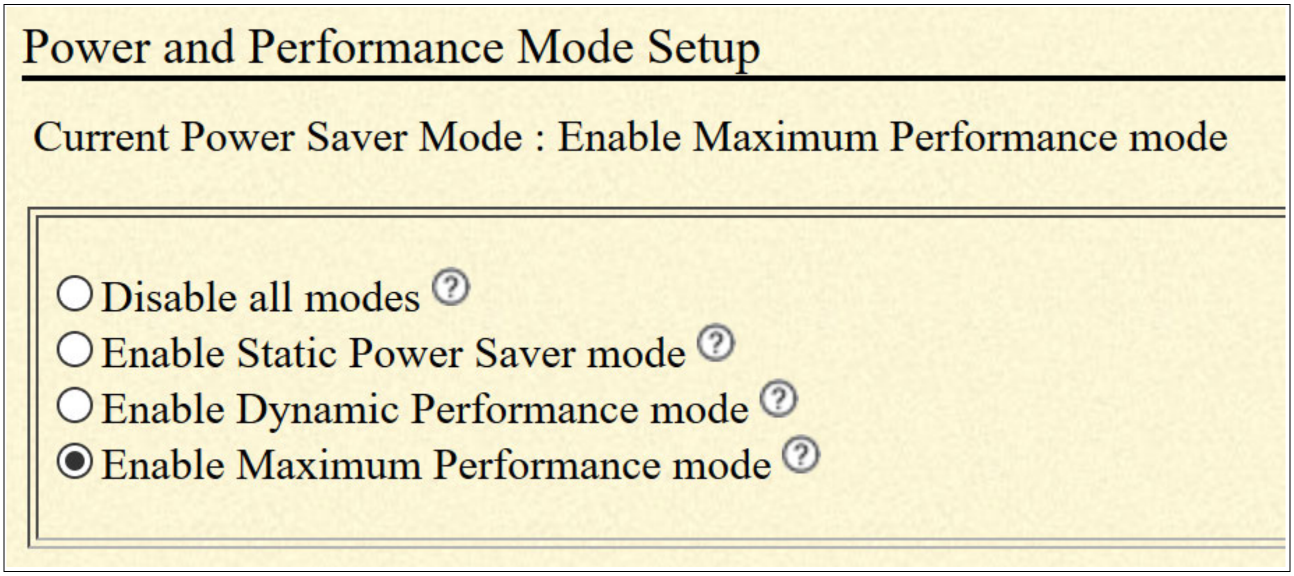
This process is done by using the HMC (from the Advanced System Management menu [ASM] or the command-line interface [CLI]). This setting is the default setting from S924 model to E980.

To access the Advanced System Management menu from the HMC, complete the following steps:

1. Select the server and then, **Operations** → **Launch Advanced System Management (ASM)**.

   You are prompted for an administrator login and password.

2. Click **ASM menu** → **System Configuration** → **Power Management** → **Power and Performance Mode Setup (or Power Mode Setup for POWER8)**. Select the **Enable Maximum Performance mode** option if it is not yet selected, as shown in Figure 5-1 on page 52 (or, select **Enable Fixed Maximum Frequency mode for POWER8**).



Power and Performance Mode Setup

Current Power Saver Mode : Enable Maximum Performance mode

○ Disable all modes

○ Enable Static Power Saver mode

○ Enable Dynamic Performance mode

● Enable Maximum Performance mode

Figure 5-1 Power and Performance Mode Setup

3. Alternatively, SSH to the HMC console, as shown in the following example:

   ```
   CLI: lspwrmgmt -m <managed system name>  -r sys | cut -d, -f4,5,
   ```

   If it is not set, run the following command:

   ```
   chpwrmgmt -m <managed system name> -r sys -o enable -t fixed_max_frequency
   ```

   Example output:

   ```
   lspwrmgmt -m SERVER1  -r sys | cut -d, -f4,5
   curr_power_saver_mode=Enabled,curr_power_saver_mode_type=max_perf
   ```

5.4.3 Activating Spectre and Meltdown (best practice)

The security fix for Spectre and Meltdown is activated by default on POWER9.

The following options are available:

- 0: Speculative execution fully enabled (*lowest* level of protection).
- 1: Speculative execution controls to mitigate user-to-kernel side-channel attacks.
- 2: Speculative execution controls to mitigate user-to-kernel and user-to-user side-channel attacks (*highest* level of protection).

The Speculative execution fully enabled option is described in the IBM Support documentation[1]:

> This optional mode is designed for systems where the hypervisor, operating system, and applications can be fully trusted. Enabling this option can expose the system to CVE-2017-5753, CVE-2017- 5715, and CVE-2017-5754. This includes any partitions that are migrated (by using Live Partition Mobility) to this system. This option has the least possible impact on the performance at the cost of possible exposure to both User accessible data and System data.

If your entire environment is protected against Spectre and Meltdown vulnerabilities, you can disable the security on the POWER9 frame by using the ASM interface from the HMC.

[1] https://www.ibm.com/support/pages/how-disableenable-spectremeltdown-mitigaton-power9-systems

Removing the overhead of this protection can reduce execution time by as much as 6%.

> **Note:** This change is possible only when the server is powered off.

Complete the following steps to enable this change:

1. Connect to the HMC and then, click **Operations** → **Launch Advanced System Management (ASM)**.

   You are prompted for administrator login and password credentials.

2. Click **ASM menu** → **System Configuration** → **Speculative Execution Control** and then, choose the **Speculative execution fully enabled** option.
3. Click **Save Settings**.

5.4.4 Using SMT-8 for performance improvement (best practice)

The default SMT for POWER8 was SMT4; for POWER9, the default is SMT8. The performance of SMT8 on the POWER9 chip is significantly better and can be your starting point for SMT. If you are migrating from SMT4 on POWER8, testing SMT4 *and* SMT8 on POWER9 is suitable.

Check SMT by using the **`lparstat | grep smt`** command.

The output of the command is shown in the following example:

```
System configuration: type=Shared mode=Uncapped smt=8 lcpu=16 mem=16384MB psize=16
ent=1.50
```

SMT can be changed by using the **`smtctl –t #SMT`** command.

Best practice for enabling Processor Implementation Mode

Check Processor Implementation Mode from the LPAR profile on the HMC or from AIX by using the **`prtconf | grep "Processor Implementation Mode"`** command.

The output of the command is shown in the following example:

```
Processor Implementation Mode: POWER 9
```

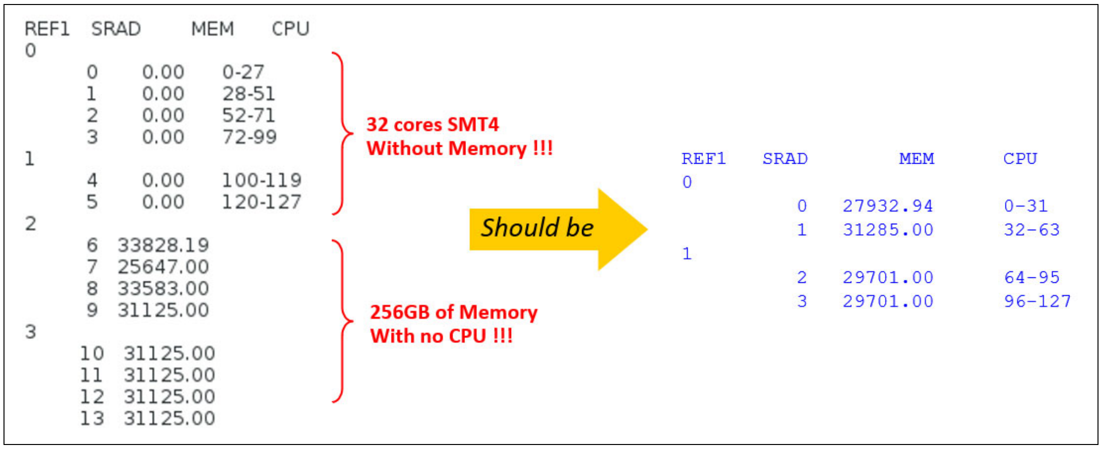
> **Note:** POWER 9 mode is possible with AIX 7.2 only.
>
> If at some time you must use live partition mobility to a POWER8 server, remain on POWER8 mode.

5.4.5 Checking LPAR affinity (best practice)

LPAR affinity is less of an issue on POWER9 but it is still worth checking when creating your partitions.

The core and memory affinity map must be closely aligned as much as possible for optimal performance. Check the affinity by using the **`lssrad -va`** command from the AIX instance that is running on your LPAR. This command reports the logical map view.

The output of the command is shown in Figure 5-2.



```
REF1  SRAD      MEM     CPU
0
       0     0.00     0-27
       1     0.00     28-51
       2     0.00     52-71
       3     0.00     72-99
1
       4     0.00     100-119
       5     0.00     120-127
2
       6  33828.19
       7  25647.00
       8  33583.00
       9  31125.00
3
      10  31125.00
      11  31125.00
      12  31125.00
      13  31125.00
```

32 cores SMT4
Without Memory !!!

256GB of Memory
With no CPU !!!

Should be

```
REF1    SRAD         MEM         CPU
0
           0    27932.94       0-31
           1    31285.00       32-63
1
           2    29701.00       64-95
           3    29701.00       96-127
```

Figure 5-2 Output of lssrad - va command

As shown on the left side in Figure 5-2, the CPUs and memory are not aligned. In this case, consider shutting down the frame and restarting the LPARs (starting with the largest) to align the CPU and memory allocation. The result looks more like the example that is shown on the right side of Figure 5-2.

> **Tip:** With a shared processor system that is running RAC, it is suggested to set the **`vpm_xvcpus`** parameter from **`schedo`** to **`2`** to avoid RAC node evictions under a light database workload conditions (**`schedo -p -o vpm_xvcpus=2`**).

5.5 Memory

The installation document suggests that the minimum RAM for a database installation is 1 GB; however, 2 GB is recommended unless you use Grid Infrastructure. In that instance, 8 GB is recommended.

From the testing that we performed, it is clear that the database software can be installed with a small amount of memory; however, dbca fails with less than 8 GB of RAM (+swapspace). Therefore, we recommend 8 GB of RAM as a minimum without Gird Infrastructure and 24 GB with it.

In the environment we created to validate this document, we allocated 8 GB, but the key factor in defining the memory for a partition is the memory that is required for the SGA and PGA or memory target of the database instance for the workload that you are running.

Oracle recommends the following memory allocations for the swap space:

- 1 GB - 2 GB: 1.5 times the size of the RAM
- 2 GB - 16 GB: Equal to the size of the RAM
- More than 16 GB: 16 GB

On AIX, we do not want to perform any swapping. For a production environment, it is recommended to allocate sufficient swap space to avoid a crash because of a lack of swap space that occurs as a result of a lack of memory.

5.5.1 64 K page promotion (Best Practice)

Unlike older versions, 19c automatically uses 64 K pages, except for the listener.

For more information, see the LDR_CNTRL Settings in MOS Note, *USE OF AIX LDR_CNTRL ENVIRONMENTAL SETTINGS WITH ORACLE* (Doc ID 2066837.1).

If you are installing RAC, see this white paper.

5.6 DNS (RAC only; mandatory)

If you are installing RAC, you must have a Single Client Access Name (SCAN) for the cluster with the following characteristics:

- Three static IP addresses that are configured on the domain name server (DNS) before installation so that the three IP addresses are associated with the name that is provided as the SCAN. All l three addresses are returned in random order by the DNS to the requester.
- Configured before installation in the DNS to resolve to addresses that are not used.
- Given addresses on the same subnet as all other public IP addresses, VIP addresses, and SCAN addresses in the cluster.
- Given a name that conforms with the RFC 952 standard, which allows alphanumeric characters and hyphens (-), but does not allow underscores (_) with a maximum of 15 characters.

It is not sufficient to add the addresses to `/etc/hosts`. Oracle must detect them by using **`nslookup`**. You must add the DNS details in the `resolv.conf` file on your server.

5.7 Network best practices

To correctly set up the 10 Gb network attributes for performance, we suggest the following best practices:

- Enable TCP enhancements as specified by RFC 1323.

  Set `rfc1323=1` system wide by using the **`no`** command or on a per-interface basis by using the **`chdev`** command.

  Check by using the following command:

  ```
  # no -Fa | grep rfc1323
  rfc1323 = 1
  ```

  or

  ```
  lsattr -El <interface name> -a rfc1323
  ```

- Increase the TCP Socket Buffer Space for Receiving and for Sending setting from 16 K (the default) to 256 K (`tcp_recvspace=262144` and `tcp_sendspace=262144` system wide or per interface basis).

  Check by using the command **`no -Fa | egrep "tcp_recvspace|tcp_sendspace"`** command (if system wide) or **`lsattr -El <interface name> -a tcp_recvspace -a tcp_sendspace`**.

  Note: This parameter is depends on the `sb_max` parameter.

- Enable Ethernet flow control: `flow_ctrl=yes`. Use the **`lsattr -El <interface name> -a flow_ctrl`** command to check and to prevent rebroadcast that results in network congestion.

> **Note:** To use the flow control as efficiency as possible, it must be enabled on all network components (including the network switch).

- Set `sb_max=4194304` by using the following command:

```
no –o sb_max=4 x (udp_recvspace or tcp_recvspace whichever is the bigger) -p
```

5.7.1 Network settings for RAC (best practice)

For IBM RAC Interconnect®, increasing UDP Socket Buffer Space for Receiving and for Sending from 64 K default to 1M (`udp_recvspace=1064960` and `udp_sendspace= 106496`, is allowed only at the system wide level). Check by using the following command:

```
no -Fa |  egrep “udp_recvspace|udp_sendspace”
```

Modify the parameter by using the following command:

```
no –o udp_recvspace=1064960 -p
```

Enable jumbo frames (`jumbo_frame=yes`) on a per-interface basis by using the **`lsattr -El <interface name> -a jumbo_frame`** command.

> **Note:** The network switch must be jumbo frame capable with jumbo frame support enabled.

Virtual Ethernet

Enable largesend for `mtu_bypass=on` per interface basis by using the **`lsattr -El <interface name> -a mtu_bypass`** command.

Set the following parameter to 4096 by using the **`chdev`** command on the Virtual Ethernet adapter (inherited from SEA):

```
min_buff_tiny=max_buff_tiny=min_buff_small=max_buff_small=4096
```

5.7.2 NTPD (RAC only, mandatory)

This section describes the Network Time Protocol (NTP) that is used to synchronize the clocks of computers over a network.

Network Time Protocol

The NTP protocol is formalized by RFCs that are released by the IETF.

The **`ntpd`** daemon maintains the system time of day in synchronism with internet-standard time servers by exchanging messages with one or more configured servers at designated poll intervals.

Under ordinary conditions, **`ntpd`** adjusts the clock in small steps so that the timescale is effectively continuous. Under conditions of extreme network congestion, the round-trip delay jitter can exceed three seconds and the synchronization distance (which is equal to one-half the round-trip delay plus error budget terms) can become large.

The **ntpd** algorithms discard sample offsets that exceed 128 ms, unless the interval during which no sample offset is less than 128 ms exceeds 900s. The first sample that is taken after that first-time interval, no matter what the offset, steps the clock to the indicated time. In practice, this algorithm reduces the false alarm rate where the clock is stepped in error to a vanishingly low incidence.

As the result of this behavior, after the clock is set, it rarely stays more than 128 ms, even under extreme cases of network path congestion. Sometimes, especially when **ntpd** is first started, the error might exceed 128 ms. With RAC, this behavior is unacceptable. If the **-x** option is included on the command line, the clock is never stepped and only slew corrections are used.

Note: The **-x** option ensures that time is not set back, which is the key issue.

The **-x** option sets the threshold to 600 s, which is well within the accuracy window to set the clock manually.

To configure the daemon NTP, complete the following steps:

1. Edit the `/etc/ntp.conf` file and add the following lines:

   ```
   # Broadcast client, no authentication.
   #
   broadcastclient
   driftfile /etc/ntp.drift
   tracefile /etc/ntp.trace
   authenticate no
   OPTIONS="-x"
   ```

2. Run **/usr/sbin/xntpd -x** to start the NTP daemon (remember the **x** option).
3. To start the **ntp** daemon automatically at restart with **-x** (slewing option), update the `/etc/rc.tcpip` file by uncommenting the following line and adding the **–x** options:

   ```
   # Start up Network Time Protocol (NTP) daemon
   startsrc /usr/sbin/xntpd "$src_running" "–x"
   # lssrc –a | grep ntp
   ```

   To start the **ntp** daemon manually with **–x** (slewing option), enter:

   ```
   {iccxx:root}/ # startsrc -s xntpd -a "-x"
   ```

After the cluster is installed, you can ignore the following message in one of the alertORA.log of the cluster:

```
[ctssd(2687544)]CRS-2409:The clock on host rac1 is not synchronous with the mean
cluster time. No action has been taken as the Cluster Time Synchronization Service
is running in observer mode.
```

Note: RAC also has its own time sync method, but this method is overridden by NTPD.

5.8 UNIX users and groups (mandatory)

Consider the following points:

- You must create the following users and groups for all Oracle installations:

```
mkgroup -A id=54421 oinstall
mkgroup -A id=54322 dba
mkgroup -A id=54323 oper
```

- The following groups are optional and can be used for different roles if you want to limit access:

```
mkgroup -A id=54324 backupdba #(for backup and restore)
mkgroup -A id=54325 dgdba #(for dataguard)
mkgroup -A id=54326 kmdba #(for encryption key management)
```

- The following groups are required for ASM:

```
mkgroup -A id=54327 asmdba
mkgroup -A id=54328 asmoper
mkgroup -A id=54329 asmadmin
```

- The following group is new with 19c and is for RAC administration:

```
mkgroup -A id=54330 racdba
```

- Create the Oracle user:

```
mkuser id='54321' pgrp='oinstall' groups=dba,oper,asmdba home='/home/oracle'
oracle
```

- For a stand-alone database with ASM, create the `grid` user with the following groups:

```
mkuser id='54322' pgrp='oinstall' groups=dba,asmadmin,asmdba,oper,asmoper
 home='/home/grid' grid
```

- For RAC, create the `grid` user with the following groups:

```
mkuser id='54322' pgrp='oinstall'
groups=dba,asmadmin,asmdba,racdba,oper,asmoper home='/home/grid' grid
```

- The following directories are suggested by Oracle in the installation documents:

```
mkdir -p /u01/app/19.0.0/grid
mkdir -p /u01/app/grid
mkdir -p /u01/app/oracle
mkdir -p /u01/app/oraInventory
mkdir -p /u01/app/oracle/product/19.0.0/dbhome_1
chown -R grid:oinstall /u01
chown oracle:oinstall /u01/app/oracle
chmod -R 775 /u01/
```

- Ulimit must be positioned for the `root`, `grid`, and `oracle` users. Setting the values to `unlimited` avoids any issues:

```
chuser threads='-1' nproc='-1' fsize='-1' data='-1' rss='-1' nofiles='-1'
oracle
chuser threads='-1' nproc='-1' fsize='-1' data='-1' rss='-1' nofiles='-1' grid
```

- The umask for the user is `022`.
- The `.profile` for Oracle user:
- If the user is going to be the owner of the listener, see 5.5.1, “64 K page promotion (Best Practice)” on page 55.

5.8.1 Turning off Oracle Online Patching in your environment if not required (best practice)

Update your oracle AIX user `.profile` file by using the following parameter:

```
export MPROTECT_TXT=OFF
```

This parameter prevents the CPU from skyrocketing in case of memory page claims under certain circumstances.

5.9 Storage

In this section, we discuss high-level storage issues. For more information, see Chapter 8, “Oracle Database and storage” on page 111.

5.9.1 LUNs for logical volumes (best practice)

When creating file systems or ASM disks to store your Oracle Database data files, control files, redo logs, archive logs, and so on, set multiple I/O queues by configuring several LUNs.

We recommend the use of eight a multiple of eight LUNs to improve I/O flow. We recommend limiting the size of the LUNs to 512 GB.

ASM tuning (best practice)

Increase the SGA of the +ASM instance to a minimum of 3 GB. Calculate the exact size as described next.

The following initialization parameters are adjusted for ASM:

- Add 16 to the value of processes.
- Add 600 KB to the value of large pool size.
- Add the aggregate of the values that are returned by the following queries to the shared pool size:
  - `SELECT SUM(bytes)/(1024*1024*1024) FROM V$DATAFILE;`
  - `SELECT SUM(bytes)/(1024*1024*1024) FROM V$LOGFILE a, V$LOG b WHERE a.group#=b.group#;`
  - `SELECT SUM(bytes)/(1024*1024*1024) FROM V$TEMPFILE WHERE status=’ONLINE’;`
- For disk groups that use external redundancy, every 100 GB of space needs 1 MB of extra shared pool, plus 2 MB.
- For disk groups that use normal redundancy, every 50 GB of space needs 1 MB of extra shared pool, plus 4 MB.
- For disk groups that use high redundancy, every 33 GB of space needs 1 MB of extra shared pool, plus 6 MB.

> **Note:** Normal practice is to use external redundancy on AIX because the redundancy is provided by the storage.

- Set the AU size to at least 4 MB, the default of 1 MB is too small (list of value 1, 2, 4, 8, 16, 32, or 64 MB). ASM Allocation Unit size must be a multiple of the underlying storage block size and larger AU size is recommended for big LUN size. Also, larger AU sizes provide performance advantages for data warehouse workloads.

5.9.2 Setting the queue depth to allow enough I/O throughput (best practice)

To check the current `queue_depth` value for an HDisk device, use the following command:

```
# lsattr -El hdisk1 –a queue_depth

queue_depth 64 Queue DEPTH True+
```

To check the range of settings that is allowed by the driver, run the following command:

```
# lsattr -Rl hdisk1 -a queue_depth

1...256 (+1)
```

To set a new value with your own device and value, use the following command:

```
chdev -l hdisk1 -a queue_depth=256 –P
```

The value of `max_transfer` must be set to a minimum of `0x100000`. Its value can be checked by using the **`lsattr`** command. The value can be set by using the **`chdev`** command in the same way.

The value of the algorithm is set to the `shortest_queue` again by using the **`chdev`** and **`lsattr`** commands.

5.9.3 Mount options for Oracle binaries (best practice)

Mount Oracle binaries JFS2 repository with **`'noatime,rw'`** options to turn off access-time update on i-node.

Check the options column from the **`lsfs -q /<repository>`** command.

Example output:

```
# lsfs -q /u01
Name            Nodename   Mount Pt               VFS   Size    Options    Auto
Accounting
/dev/oralv   --           /u01 jfs2  1002438656 noatime,rw yes  no
  (lv size: 1002438656, fs size: 1002438656, block size: 4096, sparse files: yes,
inline log: yes, inline log size: 512, EAformat: v1, Quota: no, DMAPI: no, VIX:
yes, EFS: no, ISNAPSHOT: no, MAXEXT: 0, MountGuard: no)
```

For the Oracle Database, set multiple I/O queues by configuring several LUNs to improve I/O flow, particularly if the queue depth value is low.

When hosting the Oracle Database on JFS2, create a scalable volume group (for example, a VG) with Physical Partition (PP) size set to 32 MB:

```
mkvg -S -y'<VG name>' -s '32' '-f' <space separated hdisk list>
```

Create separate JFS2 file systems for the following configuration files:

- Data files (4 K default).
- Redolog files (512 bytes)

- Control files (512 bytes).

All data files are spread across the LUN set (**-e'x**' flag) by using the **noatime** option and inline JFS2 logging:

- For each Logical Volume (LV):

  ```
  mklv -y'<LV name>' -t'jfs2' -e'x' <VG name> <LV size in PP unit>
  ```

- Check LV spreading by using the following command:

  ```
  lslv <LV name> | grep "INTER-POLICY" which is tagged as "maximum“
  ```

  Sample output:

  ```
  INTER-POLICY:      maximum               RELOCATABLE:    yes
  ```

- For data files:

  ```
  crfs -v jfs2 -d'<LV name>' -m'<JFS2 mount point>' -A'yes' -p'rw' -a
  options='noatime' -a agblksize='4096' -a logname='INLINE' -a isnapshot='no’.
  ```

- For redolog and control files:

  ```
  crfs -v jfs2 -d'<LV name>' -m'<JFS2 mount point>' -A'yes' -p'rw' -a
  options='noatime' -a agblksize='512' -a logname='INLINE' -a isnapshot='no'.
  ```

5.9.4 Standalone with JFS2

This section describes options for stand-alone JFS2 file systems.

Mount options (best practice)

Mount the Oracle binaries JFS2 repository with **'noatime,rw'** options to turn off access-time update on i-node.

Check the options column from the **lsfs -q /<repository>** command.

Example output:

```
# lsfs -q /oracle
Name            Nodename   Mount Pt               VFS   Size    Options    Auto
Accounting
/dev/oralv     --        /oracle                jfs2  1002438656 noatime,cio,rw
yes  no
  (lv size: 1002438656, fs size: 1002438656, block size: 4096, sparse files: yes,
inline log: yes, inline log size: 512, EAformat: v1, Quota: no, DMAPI: no, VIX:
yes, EFS: no, ISNAPSHOT: no, MAXEXT: 0, MountGuard: no)
```

For the Oracle Database, set multiple I/O queues by configuring several LUNs to improve I/O flow, particularly if the queue depth value is low.

When hosting the Oracle Database on JFS2, create a scalable Volume Group (VG) with Physical Partition (PP) size set to 32 MB:

```
mkvg -S -y'<VG name>' -s '32' '-f' <space separated hdisk list>
```

Create separate JFS2 file systems for the following configuration files:

- Data files (4K default)
- Redolog files (512 bytes)
- Control files (512 bytes)

All data files are spread across the LUN set (**-e'x'** flag) with **noatime** option and inline JFS2 logging:

- For each Logical Volume (LV):

  ```
  mklv -y'<LV name>' -t'jfs2' -e'x' <VG name> <LV size in PP unit>
  ```

- Check LV spreading with the command:

  ```
  lslv <LV name> | grep "INTER-POLICY" which is tagged as "maximum“
  ```

  Sample output:

  ```
  INTER-POLICY:      maximum              RELOCATABLE:    yes
  ```

- For data files:

  ```
  crfs -v jfs2 -d'<LV name>' -m'<JFS2 mount point>' -A'yes' -p'rw' -a
  options='noatime' -a agblksize='4096' -a logname='INLINE' -a isnapshot='no’.
  ```

- For redolog and control files:

  ```
  crfs -v jfs2 -d'<LV name>' -m'<JFS2 mount point>' -A'yes' -p'rw' -a
  options='noatime' -a agblksize='512' -a logname='INLINE' -a isnapshot='no'.
  ```

5.9.5 Standalone with ASM

This section describes how to configure ASM.

Prepare the disks for ASM (mandatory)

The disks must be visible if you perform an **lspv** but no volume group is required because this issue is handled by ASM. If a volume group is created, ASM cannot see the disks:

```
# lspv
hdisk0          00cff3b40a6acc17                    rootvg          active
hdisk1          00c00980e0eb69e0                    oravg           active
hdisk2          none                                None
```

The hdisk2 is available to use for ASM. You see that the Physical Volume Identifier (PVID) and volume group are both set to none. If not, ASM cannot see the disk.

If the PVID is not showing as none, reset it by using the following command:

```
chdev -l hdisk2 -a pv=clear
```

Set the reserve_policy to no_reserve by using the following command:

```
chdev -l hdisk2 -a reserve_policy=no_reserve
```

You can check the value of hdisk2 by using the following command:

```
lsattr -El hdisk2 -a reserve_policy
reserve_policy no_reserve Reserve Policy True+
```

The permissions on the rhdisks are to 066. These rhdisks are owned by the grid user and the group asmadmin. If you must rename the devices, you can do so by using the **rendev** command.

Installer (mandatory)

The grid infrastructure installer requests 79 GB of free space to install GI but needs only less than 5 GB at this stage because the binaries are unpacked on the disk.

If ASM fails to see your disks, review the steps in 5.3.3, “Operating system changes (mandatory)” on page 50) because the permissions and the ownership all can prevent the installer from seeing the disks.

The **runfixup.sh** function fixes any minor configuration issues that are prerequisites, but it does not apply the recommended best practices.

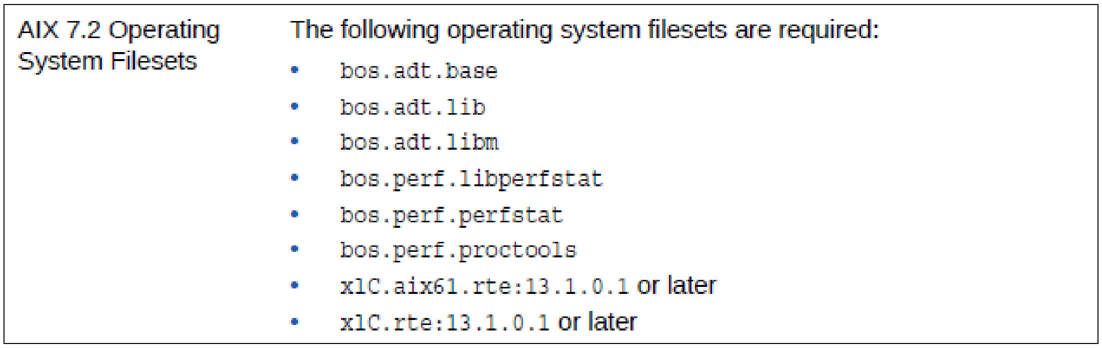
During the grid infrastructure installation process, you can ignore the space and swap space requirements.

5.9.6 Required file sets for Oracle Database software installation (mandatory)

To check whether the necessary file sets are installed, run the following commands:

```
lslpp -l bos.adt.base bos.adt.lib bos.adt.libm bos.perf.perfstat
bos.perf.libperfstat bos.perf.proctools
lslpp -l|grep xlC
```

The list of file sets is shown in Figure 5-3.



AIX 7.2 Operating System Filesets	The following operating system filesets are required: • `bos.adt.base` • `bos.adt.lib` • `bos.adt.libm` • `bos.perf.libperfstat` • `bos.perf.perfstat` • `bos.perf.proctools` • `xlC.aix61.rte:13.1.0.1` or later • `xlC.rte:13.1.0.1` or later

Figure 5-3 AIX 7.2 operating system file sets

If you are running AIX 7.2 TL 2 SP1, this documentation also recommends installing APAR: IJ06143.

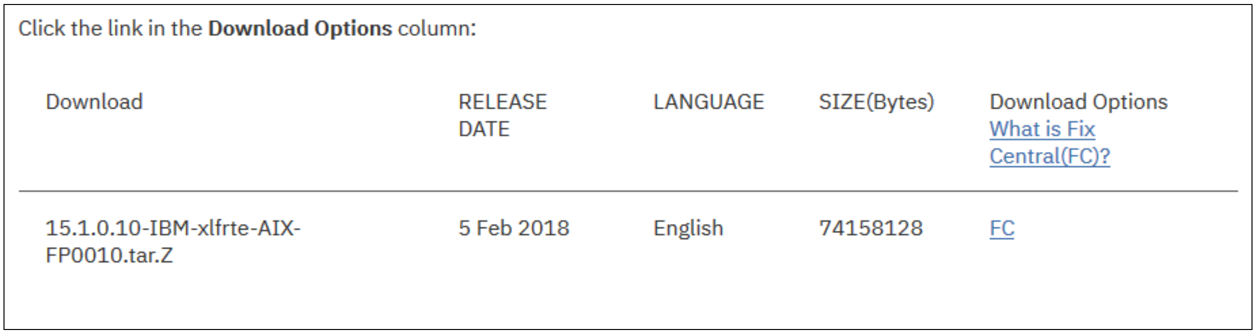
The following package also is required for 19C but is not in the documentation:

```
Xlfrte.aix61-15.1.0.9
```

If you do not install this package, you get the error PRVF-7532: Package “xlftre.aix61” is missing.

The required package is available at this IBM Support web page.

At tis web page, scroll down to find the section that is shown in Figure 5-4.



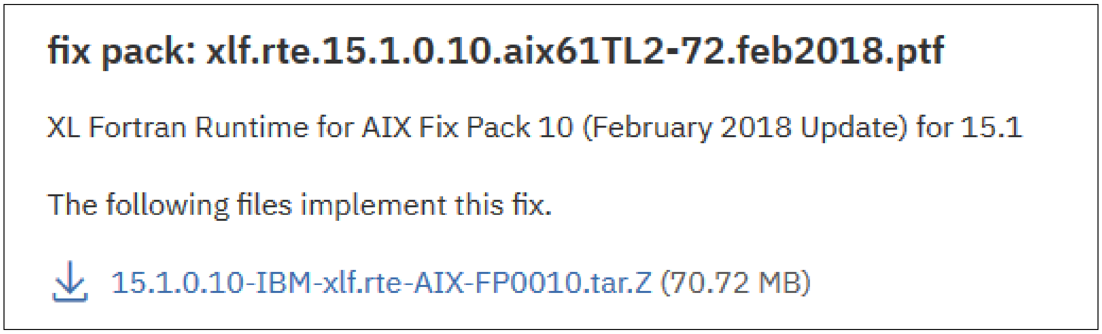
Click the link in the **Download Options** column:

Download	RELEASE DATE	LANGUAGE	SIZE(Bytes)	Download Options What is Fix Central(FC)?
15.1.0.10-IBM-xlfrte-AIX-FP0010.tar.Z	5 Feb 2018	English	74158128	FC

Figure 5-4 Link in the Download Options column

When you click **What is ix Central (FC)?**, you are required to log in by using your IBM ID credentials.

Scroll down to find the download link (see Figure 5-5).



fix pack: xlf.rte.15.1.0.10.aix61TL2-72.feb2018.ptf

XL Fortran Runtime for AIX Fix Pack 10 (February 2018 Update) for 15.1

The following files implement this fix.

15.1.0.10-IBM-xlf.rte-AIX-FP0010.tar.Z (70.72 MB)

Figure 5-5 fix pack: xlf.rte.15.1.0.10.aix61TL2-72.feb2018.ptf

Transfer the package to the server, unpack and install it by running the following command:

```
smit install_latest
```

Run the **smitty** command to show you that the following files were installed:

```
File:
    I:xlfrte                       15.1.0.0
    S:xlfrte                       15.1.0.10
    I:xlfrte.aix61                 15.1.0.0
    S:xlfrte.aix61                 15.1.0.10
```

Note: This document does not cover the requirements for NLS language settings.

File system for database installation (mandatory)

Oracle recommends 100 GB of space for the installation of the Oracle home to allow space for patching.

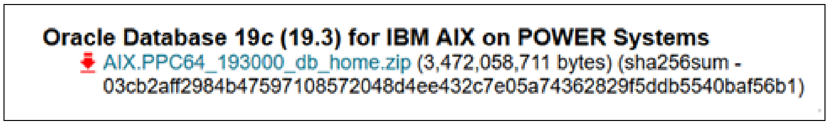
In the documentation, the minimum is stated as approximately 11 GB. You need more space for staging the zip and with the log files that are created in the Oracle Home, you need approximately 15 GB as a minimum to have a usable environment without any patching. In our test installation, the final size of the installation alone was 12.6 GB without the zip file or any database logs or trace files created.

Download and documentation links

The installation documentation is at this web page.

The software is available for download at this web page.

The base version is 19.3 (see Figure 5-6).



Oracle Database 19c (19.3) for IBM AIX on POWER Systems

AIX.PPC64_193000_db_home.zip (3,472,058,711 bytes) (sha256sum - 03cb2aff2984b47597108572048d4ee432c7e05a74362829f5ddb5540baf56b1)

Figure 5-6 Oracle Database version

Setting the user and group IDs to be the same helps if you need to transfer files or data between servers.

5.10 Installing the Oracle Database

Complete the following steps to install the Oracles Database:

1. Download the Oracle Database 19c (19.3) for IBM AIX on POWER Systems software from this web page.
2. Copy the database installation file `AIX.PPC64_193000_db_home.zip` to a staging area on the server where it is to be installed or NFS mount a disk to the server to make the zip file available.
3. Create the file system where the Oracle Home is to be created.
4. Set the permissions on the file system so that the Oracle user has read, write, and execute permission, as shown in the following example:

   ```
   chown oracle.oinstall /u01
   chmod 755 /u01
   ```

   Three directories are required for a stand-alone installation. One directory is used for inventory, one directory for the ORACLE_BASE, and one directory for the ORACLE_HOME.

   > **Note:** ORACLE_HOME used to be a subdirectory of ORACLE_HOME; however, it no longer is used as a subdirectory.

   In this scenario, we create a file system `/u01` for all the Oracle files:

   – `/u01/app/oraInventory` was created for the inventory files.

   – `/u01/app/oracle` was created for the Oracle Base.

   – `/u01/app/product` was created for the ORACLE_HOME to be created as `/u01/app/product/19c`.

   We chose to use JFS2 file systems to store our data files.
5. Create the ORACLE_HOME directory and cd to it.

   From this directory, start the **`unzip`** command for the zip file as the `oracle` user.

   Oracle recommends performing the **`-q`** option:

   ```
   unzip -q AIX.PPC64_193000_db_home.zip
   ```

6. As the `root` user, run the **`rootpre.sh`** script, which is found in the `ORACLE_HOME/clone` directory.
7. For ease of access for the runInstaller, the VNCserver is started as the user who is the owner of the Oracle binaries and from the directory that becomes the ORACLE HOME.

   You can check that the display is working correctly by using xclock. If the clock appears, cancel it and you are ready to run the runInstaller.
8. Choose the following options within the installer:
   - Setup software only. Click **Next**.
   - Single instance database installation. Click **Next**.
   - Enterprise Edition. Click **Next**.

   You can install Standard Edition, but you are limited to the use of two sockets and only 16 threads.
9. Indicate the location of your Oracle base directory (for example, we chose `/u01/app/oracle`). Then, click **Next**.

   Note: Unlike older versions of Oracle, the Oracle Home is not created within the Oracle Base; it must be in a separate directory.

   Complete the following steps:

   a. Indicate the Inventory directory (for our example, we chose `/u01/app/oraInventory`). Click **Next**.

   b. Accept the default settings for the Privileged Operating System groups. Click **Next**.

   c. We chose not to allow the scripts that require root privilege to be run automatically. Click **Next**.

   At this stage, the installer checks the prerequisites and configuration. No APARs are listed as missing.
10. The installer warns that not enough swap space is available. Ignore the warning and click **Next**.
11. The Summary Installation window prompts you to click Install. Click **Install**.
12. Run the **`orainstRoot.sh`** and the **`root.sh`** scripts as `root`.
13. On completion, click **OK**.

    The Finish window is displayed that includes the message: `The registration of Oracle Database was successful.`
14. Click **Close** to finish.

5.11 Post-installation tasks

You are now ready to create your database by using `dbca`. The `dbca` utility is found in the `$ORACLE_HOME/bin` directory.

For this example, we created the `/oradata` file system that is owned by the `oracle` user for the data files and redo logs.

> **Note:** This process does not follow best practices; it is included here for demonstration purposes only.

Your redo logs are on a separate file system with a block size of 512 bytes. The data files must be in a file system that is mounted in cio. Alternatively, you can set the parameter `filesystemio_options` to `setall` (see “Related publications” on page 127).

Complete the following steps to create the database:

1. Accept the default **Create a database** option. Click **Next**.
2. Choose the **Advanced** option. Click **Next**.
3. Choose **Oracle Single Instance database** and the **General Purpose or Transaction Processing** template. These selections are used for most non-data warehouse environments. click **Next**.
4. Choose your Oracle SID. Click **Next**.
5. Chose the database storage attributes option and change the database file location from the oracle base directory to the mount point chosen for your data files (we often use `/oradata`). Click **Next**.
6. If you choose to define a Fast Recovery Area, it must be large enough to contain your RMAN backup files and archive log files. If you do not define such an area, create your backup location later. Click **Next**.
7. In this Specify Network Configuration Details page. you define a listener. Because the database is picked up by the default listener upon starting, we do not need to configure this section. Click **Next**.
8. Unless required, Oracle Database Vault and Oracle Label Security are not needed. Click **Next**.

   The next page features several tabs that are used for defining memory, character sets, connection mode, and sample schemas. Unless you must meet specific requirements, you can ignore all of the tabs, except memory.

   Automatic memory management is again the default setting that is used (unlike 18c). If other databases are running in the same partition, they are not considered in the proposed memory size. The System Global Area (SGA) rarely extends to its full memory allocation.

   > **Note:** The SGA of all database on a partition cannot exceed 70% of the available memory on the server or partition.

9. We do not recommend the use of Memory Target. Although it is a useful tool if you do not know the required memory size, assigning SGA and PGA targets avoids memory resize operations that can affect performance. Click **Next**.
10. By default, the database ID is configured to include Enterprise Manager database express. Clear this option if you do not want it to be installed. Click **Next**.

11. The Database User Credentials page opens, in which you can set SYS and SYSTEM passwords. If this environment is a demonstration or test environment, you can set a simple password (although Oracle warns that such a password does not match the expected standard). Click **Next**.
12. Accept the default **Create database** option. Click **All Initialization Parameters** to change the database parameters that are written to the `spfile`.
13. For the initialization parameters, click **Show advanced parameters**. The `filesystemio_options` is then set to `setall`. Click the option to include it in `spfile` and then, click **Close**.
14. Modify the redo log definition in the Customize Storage window. The default size of the redo logs is 200 M, which must be increased for log groups 1, 2, and 3.

    The recommendation from Oracle is to switch redo logs 3 times each hour. This switch avoids waits on too many switches that can affect performance but might be too infrequent for some critical workloads.

    On most of our benchmark testing, we set a redolog size of 1 GB - 10 GB, depending on the intensity of the workload.
15. The default location for the redo logs is in the same place as the data files. Change this location if you created a separate file system as recommended in the best practices documentation.
16. Click **Apply** whenever you change a value or your changes are lost. After all redo log changes are applied, click **OK**.
17. Click **Finish** to start the database creation process.

The progress page now appears as the database is created.

After the creation process completes, you are presented with another opportunity to change Password Management options or you can close the `dbca` utility.

5.12 Creating the database (best practices)

Set the parameter `filesystemio_options=setall` in your `init.ora` or `spfile`. This parameter allows the Oracle Database instance to access files by way of cio and bypass the file system cache. For more information, see Chapter 8, “Oracle Database and storage” on page 111.

For memory allocation, we recommend setting SGA and PGA targets. Setting memory targets can result in frequent memory resize operations, which degrades performance.

As a rule, the total memory that is taken by the SGA and PGA must not exceed 70% of the memory that is allocated on the machine. This requirement allows enough memory for the Oracle processes and operating system.

6

Oracle RAC for AIX with IBM Spectrum Scale

IBM Spectrum Scale (formerly known as General Parallel File System) provides the shared storage infrastructure that is required for Oracle RAC deployments.

In this chapter, we describe the deployment of Oracle RAC 19c on a two node cluster that is based on IBM Advanced Executive Interactive (AIX) and PowerVM on IBM POWER9 servers that use IBM Spectrum Scale as shared storage infrastructure.

We also introduce design considerations for such deployment configuration, and management-related tasks.

> **Disclaimer:** Consider the following points:
>
> - In this scenario, we describe the experience in our test environment. The procedures that are described in this chapter are not intended to replace any official product documentation; official product documentation *always* must be followed.
> - The test environment that we used provides only an example for specific tasks that must be performed for deploying and verifying the configuration. Because of the flexibility of the IBM Power Systems platform, your environment can differ significantly from our test environment.
> - Your platform design must consider all recommendations for functions, performance, availability, and disaster recovery that are provided by the products guidelines.

This chapter includes the following topics:

- 6.1, “Introduction and considerations” on page 70.
- 6.2, “POWER9 platform capabilities” on page 70.
- 6.3, “Configuration overview” on page 72.
- 6.4, “AIX platform preparation (excerpt)” on page 75.
- 6.5, “IBM Spectrum Scale configuration” on page 88.
- 6.6, “Deploying Oracle software” on page 96.

6.1 Introduction and considerations

The IBM AIX operating system is known for leadership performance, security, and reliability for running mission critical, enterprise class applications. Together with IBM POWER9 platform, the combination provides unmatched availability[1], virtualization capabilities, disaster recovery capabilities, and scalability (scale-up and scale-out).

The choice of deploying Oracle RAC on IBM POWER9 with AIX and IBM Spectrum Scale provides a strong combination of performance, scalability, and availability for your mission critical databases and applications. The combination (Oracle RAC, Power, AIX, and Spectrum Scale) goes back many years, starting with Oracle RAC 9i[2] to the most recent announcement for Oracle RAC 19c[3].

6.2 POWER9 platform capabilities

In this section, we describe the IBM POWER9 capabilities, focusing on workload management and virtualization.

For more information about IBM POWER9 systems, see this IBM Redbooks web page.

6.2.1 CPU

POWER9 provides advanced processor capabilities, including the following examples that we consider the most relevant for our deployment:

- High frequency super scalar Reduced Instruction Set Computing (RISC) architecture.
- Parallel execution of multiple threads (8-way Simultaneous Multi-Threading - SMT).
- Optimized execution pipeline and integrated accelerators.
- Highly granular allocation capabilities (0.01 processor increments).
- Dynamic CPU sparing.

For more information, see *IBM POWER Virtualization Best Practices Guide*.

6.2.2 Memory

Memory management for POWER9 servers includes the following features:

- Dynamic memory allocation (add/remove): Also known as DLPAR Memory, this memory is the traditional memory management that is based on PowerVM and Reliable Scalable Technology (RSCT) feature.
- Available Active Memory Expansion (priced feature): Provides in-memory data compression, which provides expanded memory capacity for your system. This feature relies on-chip compression feature of the Power processor.
- Active Memory Sharing: Provides physical pool of memory sharing among a group of partitions. This feature is implemented in PowerVM.

[1] Availability is a combination of hardware and software working in synergy for providing the uptime required by your applications' service level agreement. Combined with the AIX operating system and management software, the POWER9 platform can provide "five nines" uptime (99.999% availability). For more information, see: https://www.ibm.com/it-infrastructure/us-en/resources/power/five-nines-power9.

[2] For latest support matrix, see: https://www.oracle.com/database/technologies/tech-generic-unix-new.html

[3] See http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/FLASH10907

6.2.3 Networking

IBM PowerVM provides advanced networking virtualization options for LPARs, including the following examples:

- Virtual Ethernet that uses Shared Ethernet Adapter in the VIO server for external network access.
- Dedicated physical Ethernet adapters (standard Ethernet or RDMA capable).
- SR-IOV logical ports (dedicated or shared).
- Virtual Network Interface Card (vNIC) client that uses shared SR-IOV logical ports in the VIO Server.

Availability for LPAR access to external network can be provided by various techniques and combinations of thereof, including the following examples:

- Redundant VIO Server configuration with SEA failover (with optional dual Ethernet adapters and link aggregation in each VIO Server).
- Dual Ethernet adapters (physical, virtual, SR-IOV, or vNIC) configuration in LPAR with network interface backup configuration in AIX.
- Redundant VIO Servers with vNIC failover.

6.2.4 Storage options

Disk storage can be provided to LPARs for dedicated and shared volumes by way of:

- Dedicated Fibre Channel adapters
- Virtual Fibre Channel (NPIV)
- Virtual SCSI

Storage availability is provided by one or a combination of the following methods:

- Redundant Fibre Channel (physical or virtual) with AIX MPIO at LPAR level.
- RAID configuration in the storage subsystems.
- Redundant VIO servers for external storage access.
- Redundant SAN fabrics.
- Operating system level mirroring for data protection in AIX.

6.3 Configuration overview

In this section, we provide a high-level description of the platform that we used to deploy the scenario that is described in this chapter.

6.3.1 Scenario description

In our test environment, we deployed a two-node cluster with the following configuration elements:

- Two POWER9 LPARs (`node1`, `hostname=aop93cld24`, and `node2`, `hostname=aop93cl093`), on two different systems. Each LPAR includes the following features:
  - Four CPUs
  - 24 GB of RAM
  - AIX 7.2 Technology Level 4, Service Pack 2
- Storage access

  Virtual Fibre Channel (NPIV) protected by AIX multipathing with dual VIO Servers configuration and redundant SAN fabrics.
- Storage volumes:
  - Internal (nonshared): Two LUNs (100 GB each); Internal storage is used for AIX rootvg and for Oracle binaries[4] (grid and database).
  - Shared:
    - Three 10-GB LUNs for tiebreaker disks (Spectrum Scale cluster management).
    - Eight 50-GB LUNs for data (for shared Oracle storage).
- Networking (see Example 6-1):
  - Public network: Virtual I/O Ethernet that is protected by SEA failover, and identified on both nodes as `ent0`.
  - Private network: vNIC adapter that is protected by vNIC failover, and identified on both nodes as `ent2`.

Example 6-1 Network interface configuration on both nodes

```
# lsdev -Cc adapter |grep ent|egrep -v "fcs|scsi"
ent0   Available      Virtual I/O Ethernet Adapter (l-lan) <-- Public
ent1   Available      Virtual I/O Ethernet Adapter (l-lan)
ent2   Available      Virtual NIC Client Adapter (vnic)<-- Private
```

[4] Oracle Database binaries also can be deployed on shared storage; however, for availability during upgrades, we decided not to use shared storage for database binaries.

A high-level diagram of our test environment is shown in Figure 6-1. `Node1` and `Node2` are LPARs in two distinct IBM POWER9 servers.

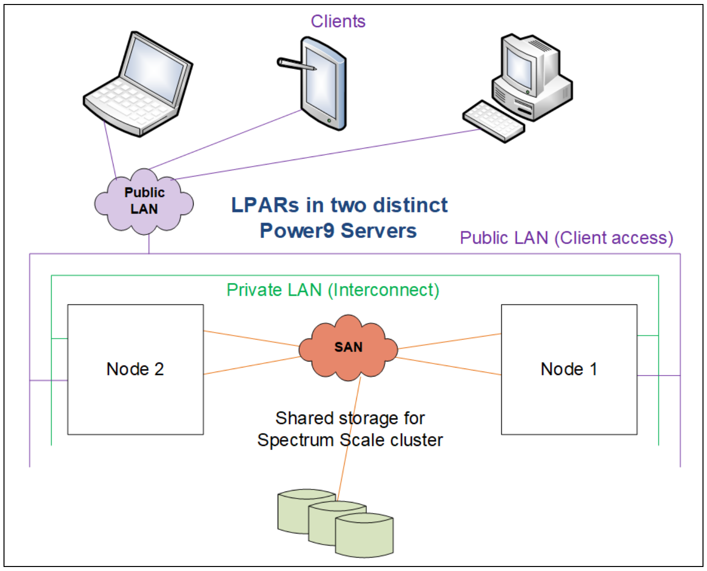

Clients
Public LAN
LPARs in two distinct Power9 Servers
Public LAN (Client access)
Private LAN (Interconnect)
Node 2
SAN
Node 1
Shared storage for Spectrum Scale cluster

Figure 6-1 Cluster diagram (high level)

6.3.2 Shared storage configuration

For optimal redundancy and availability, storage is provided from three external storage subsystems by way of redundant (dual) SAN fabric (see Figure 6-2).

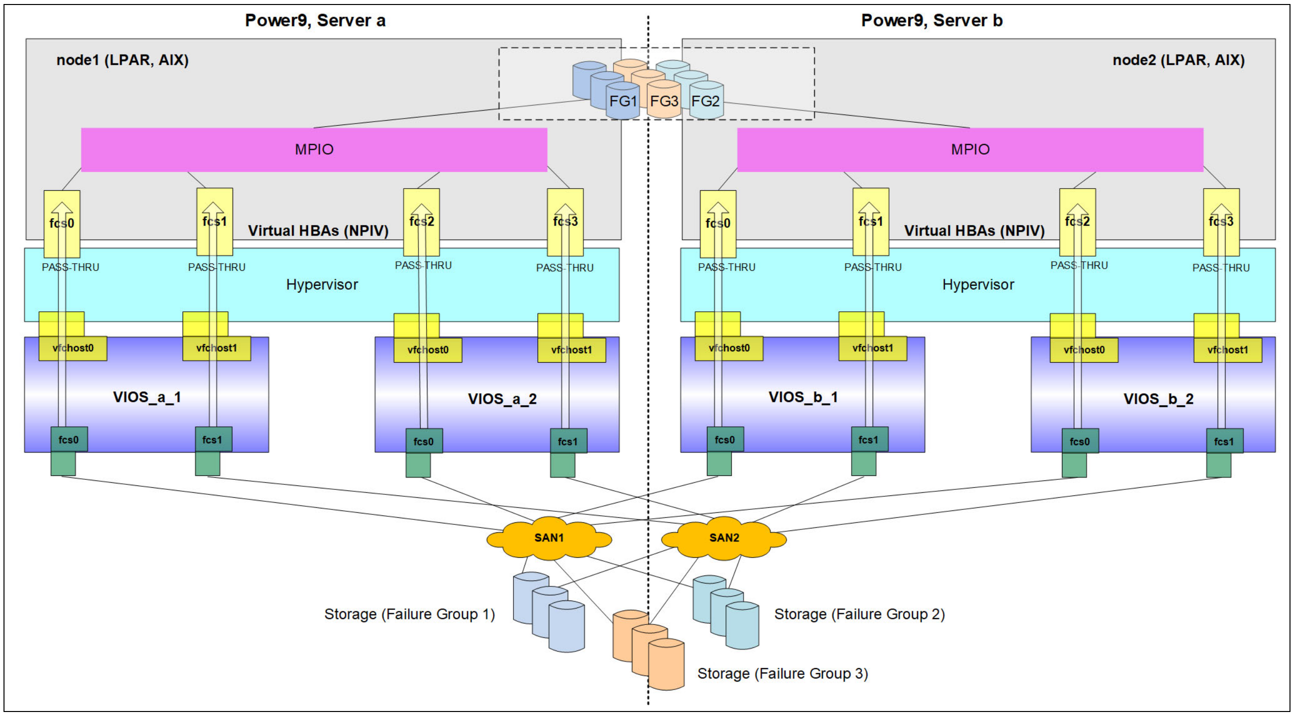


Figure 6-2 Storage configuration (shared storage)

6.3.3 Networking redundancy overview

Networking configuration consists of two separate network segments: one used for client access (public LAN) and one used for Oracle Interconnect (private LAN). A high-level networking diagram is shown in Figure 6-3.

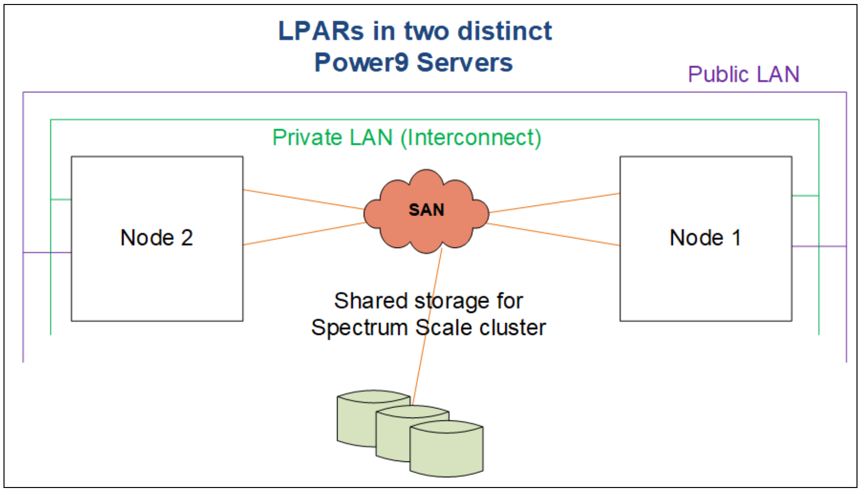


Figure 6-3 High-level networking diagram

Redundancy is provided for both network segments:

- Public LAN uses Virtual I/O Ethernet, which is protected with dual VIO Servers and Shared Ethernet Adapter failover. For more information, see this IBM Support web page.

  Figure 6-4 shows the SEA failover configuration for node1 (similar configuration for `node2`).

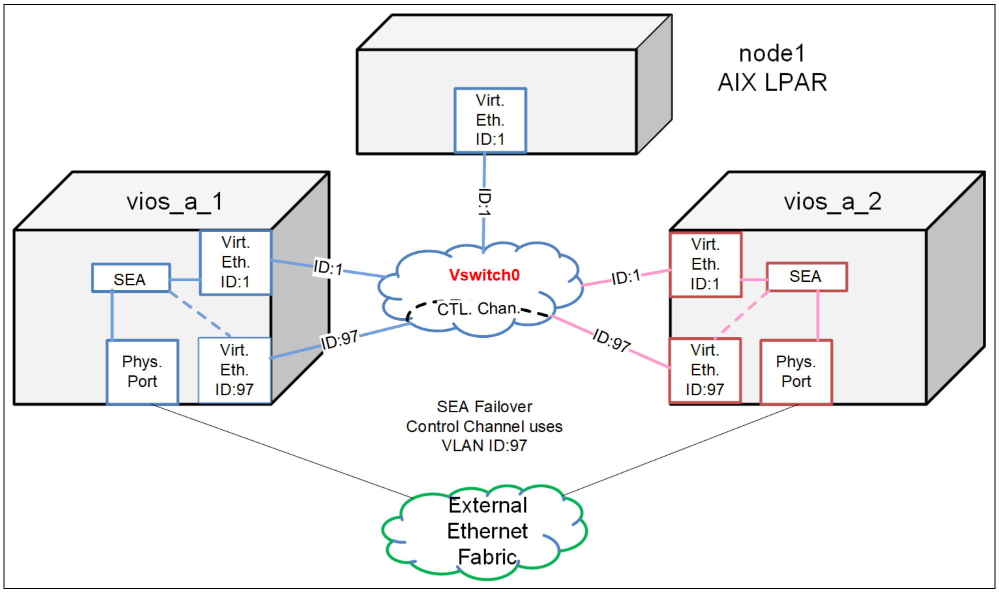


Figure 6-4 Public LAN redundancy - SEA failover

- Private LAN uses virtual Network Interface Card (vNIC), which are protected with dual VIO Servers and vNIC failover. A configuration diagram for `node1` is shown in Figure 6-5 (`node2` uses a similar configuration). For more information and recommendations, see this IBM Support web page. The private LAN is configured as `ent2` on both nodes.

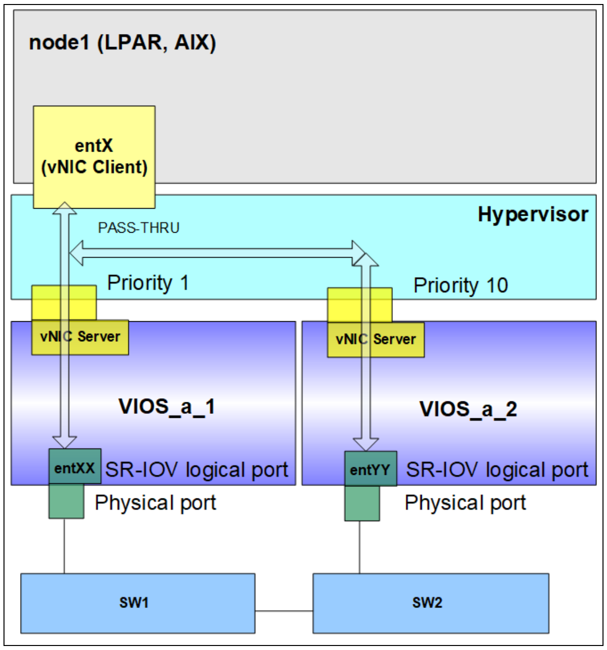


Figure 6-5 Interconnect redundancy: vNIC failover

6.4 AIX platform preparation (excerpt)

Note: This section covers cluster nodes configuration information.

The two-node cluster configuration starts by installing the base operating systems and required packages for deploying Oracle RAC solution (Oracle Clusterware and Oracle Database).

The AIX installation (AIX 7.2 TL4 SP2) procedure is beyond the scope of this document. Use the installation method of your choice to deploy the operating system and other packages that are required for your environment.

For more information about the required operating system (AIX operating system) and packages, see the following Oracle documents, which are available at this Oracle Database 19c web page:

- *Grid Infrastructure Installation and Upgrade Guide - 19c for IBM AIX on POWER Systems (64-Bit)*, E96274-03
- *Oracle Database Installation Guide - 19c for IBM AIX on POWER Systems (64-bit)*, E96437-04

Tools for distributed commands (optional)

Because we are working on a cluster that consists of two AIX LPARs (nodes), some of the configuration parameters must be set on both nodes.

Although we can run all commands on each node, we can optionally use the distributed systems management toolkit for AIX that allows us to run commands from a single node onto multiple nodes (in our case, two nodes).

The packages that are used for this purpose are named `dsm.core` and `dsm.dsh`. These packages provide the tools to run commands and copy files on multiple nodes by using a single point of control (one of the nodes in the cluster).

> **Tip:** The distributed systems management tools can be installed on only one of the nodes in your cluster.

Example 6-2 shows the distributes systems management configuration. We use the **`dsh`** command to run commands on both nodes in the cluster. Because the remote command uses **`/usr/bin/ssh`**, the Secure Shell (**`ssh`**) must be configured to run commands on all nodes without prompting the user for a password. Make sure that your nodes are configured for password-less remote **`ssh`** command execution.

> **Important:** Password-less remote commands execution with **`ssh`** also is required for Spectrum Scale configuration and Oracle RAC installation and configuration.
>
> Per Oracle Grid installation instructions, we also configured the SSH server by using the following parameter:
>
> `LoginGraceTime 0`[a]
>
> This parameter is set in **`/etc/ssh/sshd_config`**.

[a] After editing the `/etc/ssh/sshd_config`, the SSH daemon (server) must be restarted.

Example 6-2 Distributed commands configuration (optional)

```
►  Distributes systems management AIX filesets:
  Fileset                     Level  State  Type  Description (Uninstaller)
  ----------------------------------------------------------------------------
  dsm.core             7.2.4.0    C     F    Distributed Systems Management Core
  dsm.dsh              7.2.4.1    C     F    Distributed Systems Management Dsh
........................
►  Environment variables for distributes commands and file copy:
export DSH_NODE_LIST=/<path_to_file>/NODES.txt
export DSH_PATH=$PATH
export DSH_NODE_RSH=/usr/bin/ssh
export DSH_NODE_RCP=/usr/bin/scp
.......................
►  Sample NODES.txt node files (names or IP addresses):
<node1> # cat /<path_to_files>/NODES.txt
aop93cld24
aop93cl093
```

6.4.1 Operating system parameters

For more information about configuring the AIX system parameters, see Chapter 3 of the Oracle manual *Grid Infrastructure Installation and Upgrade Guide - 19c for IBM AIX on POWER Systems (64-Bit)*, E96274-03.

The following parameters were set in our environment:

- Example 6-3 shows the AIX operating system level on our cluster nodes.

Example 6-3 Operating system level

```
# dsh oslevel -s|dshbak -c
HOSTS -------------------------------------------------------------------------
aop93cl093, aop93cld24
-------------------------------------------------------------------------------
7200-04-02-2028
```

- Example 6-4 shows the kernel bit mode on our nodes.

Example 6-4 Kernel and hardware BITMODE (64 bit)

```
# dsh getconf KERNEL_BITMODE |dshbak -c
HOSTS -------------------------------------------------------------------------
aop93cl093, aop93cld24
-------------------------------------------------------------------------------
64
# dsh getconf HARDWARE_BITMODE |dshbak -c
HOSTS -------------------------------------------------------------------------
aop93cl093, aop93cld24
-------------------------------------------------------------------------------
64
```

- Example 6-5 shows the real memory that we configured on our nodes.

Example 6-5 Memory configuration on our cluster nodes

```
# dsh 'lparstat -i |grep Memory|egrep
"Online|Maximum|Minimum|Mode|Desired"'|dshbak -c
HOSTS -------------------------------------------------------------------------
aop93cl093, aop93cld24
-------------------------------------------------------------------------------
Online Memory                              : 24576 MB
Maximum Memory                             : 65536 MB
Minimum Memory                             : 8192 MB
Memory Mode                                : Dedicated
Desired Memory                             : 24576 MB
```

- Example 6-6 displays the AIX operating system packets that we installed on both nodes in preparation for Oracle Clusterware and Oracle Database installation.

Tip: If AIX toolbox for Linux packages is installed (or other open source packages) we recommend that you set the AIX MANPATH variable (in /etc/environment) to include the path to the man pages of the installed products (for example, add /opt/freeware/man to your current MANPATH).

Example 6-6 AIX operating system prerequisites

```
# dsh "lslpp -L bos.adt.base bos.adt.lib bos.adt.libm bos.perf.libperfstat
bos.perf.perfstat bos.perf.proctools xlC.* xlfrte.* rsct.basic.rte
rsct.compat.clients.rte openssh.*"|dshbak -c
HOSTS -------------------------------------------------------------------------
aop93cl093, aop93cld24
-------------------------------------------------------------------------------
  Fileset                      Level  State  Type  Description (Uninstaller)
  ----------------------------------------------------------------------------
  bos.adt.base               7.2.4.1    C     F    Base Application Development
                                                   Toolkit
  bos.adt.lib                7.2.4.0    C     F    Base Application Development
                                                   Libraries
  bos.adt.libm               7.2.3.0    C     F    Base Application Development
                                                   Math Library
  bos.perf.libperfstat       7.2.4.1    C     F    Performance Statistics Library
                                                   Interface
  bos.perf.perfstat          7.2.4.0    C     F    Performance Statistics
                                                   Interface
  bos.perf.proctools         7.2.4.1    C     F    Proc Filesystem Tools
  openssh.base.client   7.5.102.1801    C     F    Open Secure Shell Commands
  openssh.base.server   7.5.102.1801    C     F    Open Secure Shell Server
  openssh.man.en_US     7.5.102.1801    C     F    Open Secure Shell
                                                   Documentation - U.S. English
  openssh.msg.en_US     7.5.102.1801    C     F    Open Secure Shell Messages -
                                                   U.S. English
  rsct.basic.rte             3.2.5.2    C     F    RSCT Basic Function
  rsct.compat.clients.rte    3.2.5.0    C     F    RSCT Event Management Client
                                                   Function
  xlC.aix61.rte             16.1.0.3    C     F    IBM XL C++ Runtime for AIX 6.1
                                                   and later
  xlC.cpp                    9.0.0.0    C     F    C for AIX Preprocessor
  xlC.msg.en_US.cpp          9.0.0.0    C     F    C for AIX Preprocessor
                                                   Messages--U.S. English
  xlC.msg.en_US.rte         16.1.0.3    C     F    IBM XL C++ Runtime
                                                   Messages--U.S. English
  xlC.rte                   16.1.0.3    C     F    IBM XL C++ Runtime for AIX
  xlC.sup.aix50.rte          9.0.0.1    C     F    XL C/C++ Runtime for AIX 5.2
  xlfrte.aix61              15.1.2.0    C     F    XL Fortran runtime environment
                                                   for AIX 6.1, and AIX 7.1
  xlfrte.msg.en_US          15.1.2.0    C     F    XL Fortran runtime en_US
                                                   messages
```

- Example 6-7 shows the I/O completion ports for our cluster nodes.

Example 6-7 I/O completion ports

```
# dsh lsdev -l iocp0 |dshbak -c
HOSTS -------------------------------------------------------------------------
aop93cl093, aop93cld24
-------------------------------------------------------------------------------
iocp0 Available  I/O Completion Ports
```

- Example 6-8 shows the VMM[5] tuning parameters on our cluster nodes.

Example 6-8 Virtual memory manager (VMM) parameters - same on both cluster nodes

```
# vmo -L minperm%
NAME                      CUR    DEF    BOOT   MIN    MAX    UNIT           TYPE
     DEPENDENCIES
--------------------------------------------------------------------------------
minperm%                  3      3      3      1      100    %% memory         D
--------------------------------------------------------------------------------
# vmo -L maxperm%
NAME                      CUR    DEF    BOOT   MIN    MAX    UNIT           TYPE
     DEPENDENCIES
--------------------------------------------------------------------------------
maxperm%                  90     90     90     1      100    %% memory         D
     minperm%
     maxclient%
--------------------------------------------------------------------------------
# vmo -L maxclient%
NAME                      CUR    DEF    BOOT   MIN    MAX    UNIT           TYPE
     DEPENDENCIES
--------------------------------------------------------------------------------
maxclient%                90     90     90     1      100    %% memory         D
     maxperm%
     minperm%
--------------------------------------------------------------------------------
# vmo -L strict_maxclient
NAME                      CUR    DEF    BOOT   MIN    MAX    UNIT           TYPE
     DEPENDENCIES
--------------------------------------------------------------------------------
strict_maxclient          1             1                    boolean           d
     strict_maxperm
--------------------------------------------------------------------------------
# vmo -L strict_maxperm
NAME                      CUR    DEF    BOOT   MIN    MAX    UNIT           TYPE
     DEPENDENCIES
--------------------------------------------------------------------------------
strict_maxperm            0             0                    boolean           d
     strict_maxclient
--------------------------------------------------------------------------------
```

[5] VMM - Virtual Memory Manager (AIX kernel)

- Example 6-9 shows the AIX CPU folding[6] configuration on our nodes.

Example 6-9 AIX CPU folding

```
# schedo -L| grep "vpm_xvcpus"
vpm_xvcpus                2      0      0      -1     1536   processors        D
# schedo -L vpm_xvcpus
NAME                      CUR    DEF    BOOT   MIN    MAX    UNIT           TYPE
     DEPENDENCIES
--------------------------------------------------------------------------------
vpm_xvcpus                2      0      0      -1     1536   processors        D
--------------------------------------------------------------------------------
```

- Example 6-10 shows the maximum user processes and block size allocation (system parameters) on our cluster nodes.

Example 6-10 sys0 maxuproc and ncargs

```
# dsh 'lsattr -El sys0|egrep "maxuproc|ncargs"'|dshbak -c
HOSTS -------------------------------------------------------------------------
aop93cl093, aop93cld24
-------------------------------------------------------------------------------
maxuproc     16384        Maximum number of PROCESSES allowed per user     True
ncargs        1024        ARG/ENV list size in 4K byte blocks              True
```

6.4.2 Network configuration

For more information about configuring AIX networking, see Chapter 4 of the Oracle manual *Grid Infrastructure Installation and Upgrade Guide - 19c for IBM AIX on POWER Systems (64-Bit)*, E96274-03.

The network configuration for our test environment consists of the following parameter:

- Network tuning options parameters, as shown in Example 6-11.

Example 6-11 AIX network tuning options

```
# no -L udp_sendspace
--------------------------------------------------------------------------------
NAME                      CUR    DEF    BOOT   MIN    MAX    UNIT           TYPE
     DEPENDENCIES
--------------------------------------------------------------------------------
udp_sendspace             64K    9K     64K    4K     8E-1   byte              C
     sb_max
--------------------------------------------------------------------------------
# no -L udp_recvspace
--------------------------------------------------------------------------------
NAME                      CUR    DEF    BOOT   MIN    MAX    UNIT           TYPE
     DEPENDENCIES
--------------------------------------------------------------------------------
udp_recvspace             640K   42080  640K   4K     8E-1   byte              C
     sb_max
--------------------------------------------------------------------------------
# no -L tcp_sendspace
--------------------------------------------------------------------------------
```

[6] See https://www.ibm.com/support/pages/aix-virtual-processor-folding-misunderstood

```
NAME                      CUR    DEF    BOOT   MIN    MAX    UNIT           TYPE
     DEPENDENCIES
--------------------------------------------------------------------------------
tcp_sendspace             64K    16K    64K    4K     8E-1   byte              C
     sb_max
--------------------------------------------------------------------------------
# no -L tcp_recvspace
--------------------------------------------------------------------------------
NAME                      CUR    DEF    BOOT   MIN    MAX    UNIT           TYPE
     DEPENDENCIES
--------------------------------------------------------------------------------
tcp_recvspace             64K    16K    64K    4K     8E-1   byte              C
     sb_max
--------------------------------------------------------------------------------
# no -L rfc1323
--------------------------------------------------------------------------------
NAME                      CUR    DEF    BOOT   MIN    MAX    UNIT           TYPE
     DEPENDENCIES
--------------------------------------------------------------------------------
rfc1323                   1      0      1      0      1      boolean           C
--------------------------------------------------------------------------------
# no -L sb_max
--------------------------------------------------------------------------------
NAME                      CUR    DEF    BOOT   MIN    MAX    UNIT           TYPE
     DEPENDENCIES
--------------------------------------------------------------------------------
sb_max                    4M     1M     4M     4K     8E-1   byte              D
--------------------------------------------------------------------------------
# no -L ipqmaxlen
--------------------------------------------------------------------------------
NAME                      CUR    DEF    BOOT   MIN    MAX    UNIT           TYPE
     DEPENDENCIES
--------------------------------------------------------------------------------
ipqmaxlen                 512    100    512    100    2G-1   numeric           R
--------------------------------------------------------------------------------
```

- Network interfaces, as shown in Example 6-12. For our test environment, we use `ent0` for Public LAN and `ent2` for Private LAN (RAC Interconnect).

Example 6-12 Network interfaces (both nodes)

```
<node1>, <node2> # lsdev -Cc adapter |grep ent|egrep -v "fcs|scsi"
ent0   Available      Virtual I/O Ethernet Adapter (l-lan)
ent1   Available      Virtual I/O Ethernet Adapter (l-lan)
ent2   Available      Virtual NIC Client Adapter (vnic)
```

- The IP network configuration for our nodes, as shown in Example 6-13. We use the **netstat -i** command to display configuration on both nodes. You can use a distributed shell (for example, **dsh**) for issuing commands on both nodes from a single terminal. For more information about **dsh** configuration, see “Tools for distributed commands (optional)” on page 76.

Example 6-13 Network Interfaces IP addresses

```
<node1> (aop93cld24)
-------------------------------------------------------------------------------
Name   Mtu   Network     Address                Ipkts     Ierrs        Opkts     Oerrs  Coll
en0    1500  link#2      fa.c4.d.80.1b.20        18310834     0          9740857     0     0
en0    1500  129.xx.xx   aop93cld24              18310834     0          9740857     0     0
en1    1500  link#3      fa.16.3e.62.8b.d             621     0              861     0     0
en1    1500  10.10.1     node1-ic                     621     0              861     0     0
en2    1500  link#4      62.66.43.90.1a.7        28566468     0         30714671     0     0
en2    1500  10.20.1     node1-ic2               28566468     0         30714671     0     0
lo0    16896 link#1                               5343022     0          5343022     0     0
lo0    16896 127         loopback                 5343022     0          5343022     0     0
lo0    16896 loopback                             5343022     0          5343022     0     0

<node2> (aop93cl093)
-------------------------------------------------------------------------------
Name   Mtu   Network     Address                Ipkts     Ierrs        Opkts     Oerrs  Coll
en0    1500  link#2      fa.f6.ea.66.14.20        12474675     0         11019992     0     0
en0    1500  129.xx.xx   aop93cl093               12474675     0         11019992     0     0
en1    1500  link#3      fa.16.3e.23.22.6c             505     0              404     0     0
en1    1500  10.10.1     node2-ic                      505     0              404     0     0
en2    1500  link#4      62.66.4a.82.88.7         30714349     0         28337844     0     0
en2    1500  10.20.1     node2-ic2                30714349     0         28337844     0     0
lo0    16896 link#1                                3607774     0          3607774     0     0
lo0    16896 127         loopback                  3607774     0          3607774     0     0
lo0    16896 loopback                              3607774     0          3607774     0     0
```

- The host names and basic network configuration, as shown in Example 6-14.

Example 6-14 Host name and default gateway

```
<node1> (aop93cld24)
-------------------------------------------------------------------------------
authm         65536                           Authentication Methods             True
bootup_option no                              Use BSD-style Network Configuration True
gateway                                       Gateway                            True
hostname      aop93cld24.pbm.ihost.com        Host Name                          True
rout6                                         IPv6 Route                         True
route         net,-hopcount,0,,0,129.xx.xx.xxx Route                             True

<node2> (aop93cl093)
-------------------------------------------------------------------------------
authm         65536                           Authentication Methods             True
bootup_option no                              Use BSD-style Network Configuration True
gateway                                       Gateway                            True
hostname      aop93cl093.pbm.ihost.com        Host Name                          True
rout6                                         IPv6 Route                         True
route         net,-hopcount,0,,0,129.xx.xx.xxx Route                             True
```

- Network name resolution for our test environment is configured for the use of static IP addresses that are resolved to IP labels locally (**/etc/hosts**), and the use of Domain Name Server (DNS, which is configured in **/etc/resolv.conf**), in this specific order. The network name resolution order is set in **/etc/netsvc.conf**.

Important:

- All nodes in your cluster *must* resolve the network names identically (same IP labels for same IP addresses, and in the same order).
- Oracle cluster configuration requires three IP addresses resolvable by DNS for SCAN-VIP[a] configuration. The SCAN-VIP addresses are used for client access and are dynamically assigned to nodes in the cluster by Oracle Clusterware.

[a] SCAN - VIP - Single Cluster Access Name Virtual IP which is configured as a single entry in your DNS to be resolved to three IP addresses in random fashion.

Time synchronization

Generally, time synchronization between cluster nodes is required in any cluster environment.

In our test environment, we use Network Time Protocol (NTP) client on *both* nodes, which is configured in `/etc/ntp.conf`, as shown in Example 6-15.

Example 6-15 Sample /etc/ntp.conf

```
broadcastclient
driftfile /etc/ntp.drift
tracefile /etc/ntp.trace
logfile /var/log/ntp.log
server 0.north-america.pool.ntp.org
server 1.north-america.pool.ntp.org
server 2.north-america.pool.ntp.org
```

After the changes are made in your environment, you must restart the `xntpd` service to pick up the changes (**`stopsrc -s xntpd; startsrc -s xntpd`**). To check the `xntpd`, you can use the command that is shown in Example 6-16.

Example 6-16 Checking xntp daemon data

```
# lssrc -ls xntpd
 Program name:    /usr/sbin/xntpd
 Version:         3
 Leap indicator:  00 (No leap second today.)
 Sys peer:        ntp2.wiktel.com
 Sys stratum:     2
 Sys precision:   -18
 Debug/Tracing:   DISABLED
 Root distance:   0.039841
 Root dispersion: 0.002563
 Reference ID:    69.89.207.199
 Reference time:  e32c1fa6.fb94e000  Sat, Oct 10 2020  7:55:18.982
 Broadcast delay: 0.003906 (sec)
 Auth delay:      0.000122 (sec)
 System flags:    bclient pll monitor filegen
 System uptime:   923400 (sec)
 Clock stability: 0.000107 (sec)
 Clock frequency: 0.000000 (sec)
 Peer: ntp2.wiktel.com
      flags: (configured)(sys peer)
      stratum:  1, version: 3
      our mode: client, his mode: server
 Peer: lofn.fancube.com
```

```
      flags: (configured)(sys peer)
      stratum:  2, version: 3
      our mode: client, his mode: server
 Peer: test.diarizer.com
      flags: (configured)(sys peer)
      stratum:  2, version: 3
      our mode: client, his mode: server
Subsystem        Group           PID         Status
 xntpd           tcpip           6750682     active
```

Check that both nodes in the cluster use the same NTP configuration by using the **`dsh date`** command, as shown in Example 6-17.

Example 6-17   Checking time synchronization on cluster nodes

```
# dsh date
aop93cld24: Sat Oct 10 08:13:30 EDT 2020
aop93cl093: Sat Oct 10 08:13:30 EDT 2020
```

6.4.3 Users and groups

For more information about configuring the users and groups for Oracle Grid and Oracle Database, see Chapter 5 of the Oracle manual *Grid Infrastructure Installation and Upgrade Guide - 19c for IBM AIX on POWER Systems (64-Bit)*, E96274-03.

Consider the following points:

- We created users and groups, as shown in Example 6-18.

Example 6-18   Users and groups (for both cluster nodes)

```
# dsh cat /etc/passwd |dshbak -c
HOSTS -------------------------------------------------------------------------
aop93cl093, aop93cld24
-------------------------------------------------------------------------------
..........< snippet >..............
grid:!:1100:2000::/home/grid:/usr/bin/ksh
oracle:!:1000:2000::/home/oracle:/usr/bin/ksh

# dsh cat /etc/group |dshbak -c
HOSTS -------------------------------------------------------------------------
aop93cl093, aop93cld24
-------------------------------------------------------------------------------
..........< snippet >..............
dba:!:1000:grid,oracle
oinstall:!:2000:grid,oracle
asmadmin:!:1020:grid,oracle
asmdba:!:1021:grid,oracle
asmoper:!:1022:grid,oracle
racdba:!:1023:oracle
```

- The created users feature the properties and capabilities that are shown in Example 6-19.

Example 6-19   Users' attributes and capabilities (both cluster nodes)

```
# dsh lsuser -f -a capabilities fsize cpu data stack core rss nofiles
oracle|dshbak -c
```

```
HOSTS -------------------------------------------------------------------------
aop93cl093, aop93cld24
-------------------------------------------------------------------------------
oracle:
        capabilities=CAP_NUMA_ATTACH,CAP_BYPASS_RAC_VMM,CAP_PROPAGATE
        fsize=-1
        cpu=-1
        data=-1
        stack=-1
        core=2097151
        rss=-1
        nofiles=-1

# dsh lsuser -f -a capabilities fsize cpu data stack core rss nofiles grid|dshbak
-c
HOSTS -------------------------------------------------------------------------
aop93cl093, aop93cld24
-------------------------------------------------------------------------------
grid:
        capabilities=CAP_NUMA_ATTACH,CAP_BYPASS_RAC_VMM,CAP_PROPAGATE
        fsize=-1
        cpu=-1
        data=-1
        stack=-1
        core=2097151
        rss=-1
        nofiles=-1
```

- We set the log on password for `grid` and `oracle` users and created a file in the users' home directory, as shown in Example 6-20.

Example 6-20 Environment variables for grid and oracle users

```
##For both node1 and node2
# su - grid
$ cat ~/.gridenv
export ORACLE_HOME=/u02/app/19.0.0/grid
export PATH=$PATH:${ORACLE_HOME}/bin
$ exit

## For node1:
# su - oracle
$ cat ~/.oraenv
export ORACLE_HOME=/u02/app/oracle/product/19.0.0/dbhome_1
export ORACLE_SID=itsodb1
export PATH=$PATH:${ORACLE_HOME}/bin
$ exit

##For node2:
# su - oracle
$ cat ~/.oraenv
export ORACLE_HOME=/u02/app/oracle/product/19.0.0/dbhome_1
export ORACLE_SID=itsodb2
export PATH=$PATH:${ORACLE_HOME}/bin
$ exit
```

6.4.4 Storage configuration

In our test environment, we use the following local storage configuration:

- Spectrum Scale (shared file system) is used for deploying Oracle Grid infrastructure-related files (Oracle Cluster Repository, OCR files and Oracle Cluster Ready Services, and CRS Vote disks), and for database files.

  For more information about Spectrum Scale configuration, see 6.5, “IBM Spectrum Scale configuration” on page 88.

- Local storage (not shared between cluster nodes) is used for deploying Oracle Grid binaries and Oracle Database binaries. This configuration allows updating Oracle software on one node at a time.

In this section, we describe the local (non-shared) storage configuration.

> **Important:** The storage configuration must be adjusted to fit your configuration needs. For more information, see the following publications:
>
> - Chapter 7 of the Oracle manual *Grid Infrastructure Installation and Upgrade Guide - 19c for IBM AIX on POWER Systems (64-Bit)*, E96274-03.
> - Chapter 1 of the Oracle manual *Oracle Database Installation Guide - 19c for IBM AIX on POWER Systems (64-bit)*, E96437-04.

The local storage configuration consists of two non-shared volumes (LUNs), each 100 GB, on each cluster node. One disk volume is used for `rootvg` (AIX operating system) and the second volume is used for Oracle binaries:

- Local (non-shared) volumes are shown in Example 6-21.

Example 6-21 Volume groups in our cluster (both nodes)

```
# dsh 'lsvg -p rootvg oravg'|dshbak -c
HOSTS -------------------------------------------------------------------------
aop93cld24, aop93cl093
-------------------------------------------------------------------------------
rootvg:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
hdisk3            active            799         630         159..109..42..160..160
oravg2:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
oradisk2          active            799         78          00..00..00..00..78
```

- Paging space also is configured per Oracle requirements, as shown in Example 6-22. Per Oracle requirements, 16 GB of swap space is required as a minimum for systems with real memory that is larger than 16 GB.

  > **Note:** You must correctly size the memory and paging space for your environment to avoid the “out of paging space” error in AIX.

Example 6-22 Paging space available on cluster nodes

```
# dsh lsps -s |dshbak -c
HOSTS -------------------------------------------------------------------------
aop93cl093, aop93cld24
-------------------------------------------------------------------------------
Total Paging Space   Percent Used
```

```
      18176MB               0%
```

- Local file systems that are used for our environment are shown in Example 6-23. The /u02 file system is used for Oracle binaries (typically, /u01 is used as the name of the file system; however, this name is only a convention).

Example 6-23 Local file systems

```
# dsh "lsfs |grep jfs2"|dshbak -c
HOSTS -------------------------------------------------------------------------
aop93cld24, aop93cl093
-------------------------------------------------------------------------------
/dev/hd4        --        /                      jfs2  3145728 --        yes  no
/dev/hd1        --        /home                  jfs2  262144  --        yes  no
/dev/hd2        --        /usr                   jfs2  6029312 --        yes  no
/dev/hd9var     --        /var                   jfs2  5242880 --        yes  no
/dev/hd3        --        /tmp                   jfs2  11534336 --        yes  no
/dev/hd11admin  --        /admin                 jfs2  262144  --        yes  no
/dev/hd10opt    --        /opt                   jfs2  3932160 --        yes  no
/dev/livedump   --        /var/adm/ras/livedump  jfs2  524288  --        yes  no
/dev/fslv00     --        /inst_repo             jfs2  6291456 rw        yes  no
/dev/fslv01     --        /REPO                  jfs2  41943040 --        yes  no
/dev/fslv05     --        /u02                   jfs2  188743680 rw        yes  no
```

- Oracle binaries local file system configuration is shown in Example 6-24.

Tip: Example 6-24 shows that a LV is used for JFS2 logging operations. You also can configure your JFS2 file systems with inline logging.

Example 6-24 /u02 file system (Oracle binaries)

```
# dsh lsvg -l oravg2|dshbak -c
HOSTS -------------------------------------------------------------------------
aop93cl093, aop93cld24
-------------------------------------------------------------------------------
oravg2:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
loglv02             jfs2log    1       1       1    open/syncd    N/A
fslv05              jfs2       720     720     1    open/syncd    /u02

# dsh df -Pg /u02 |dshbak -c
HOSTS -------------------------------------------------------------------------
aop93cld24
-------------------------------------------------------------------------------
Filesystem    GB blocks      Used Available Capacity Mounted on
/dev/fslv05       90.00     43.89     46.11      49% /u02

HOSTS -------------------------------------------------------------------------
aop93cl093
-------------------------------------------------------------------------------
Filesystem    GB blocks      Used Available Capacity Mounted on
/dev/fslv05       90.00     46.82     43.18      53% /u02
```

- Directory structure that was prepared for Oracle installation is shown in Example 6-25.

Example 6-25 Directory structure for Oracle Grid and Oracle Database

```
##Oracle Grid
```

```
# dsh mkdir -p /u02/app/19.0.0/grid
# dsh mkdir -p /u02/app/grid
# dsh mkdir -p /u02/app/oracle
# dsh chown -R grid:oinstall /u02
# dsh chown oracle:oinstall /u02/app/oracle
# dsh chmod -R 775 /u02/

##Oracle Database
# dsh mkdir -p /u02/app/oracle
# dsh mkdir -p /u02/app/oraInventory
# dsh chown -R oracle:oinstall /u02/app/oracle
# dsh chown -R oracle:oinstall /u02/app/oraInventory
# dsh chmod -R 775 /u02/app
```

> **Note:** Because we do not use ASM in our environment, we do not prepare a shared storage for this purpose.

6.5 IBM Spectrum Scale configuration

In this section, we describe the configuration of the IBM Spectrum Scale file system that is to be used for the Oracle RAC Database.

> **Note:** The Spectrum Scale configuration that is shown in this section is provided for your reference. Your configuration can differ, depending on your deployment requirements.
>
> For more information about IBM Spectrum Scale installation and configuration, this IBM Documentation web page.
>
> For more information about the latest Spectrum Scale Frequently Asked Questions, see this web page.
>
> It is beyond the scope of this document to describe configuring external devices, such as PowerVM VIO Servers and NPIV[a], Storage Area Network, and external storage subsystems configuration.

[a] N-Port ID Virtualization (Fibre Channel virtualization)

Spectrum Scale configuration consist of the following tasks:

- Installing Spectrum Scale packages on AIX.
- Configuring:
  - Network for Spectrum Scale cluster (network interfaces, name resolution, and so on).
  - Secure Shell for password-less remote command execution.
  - External storage and verifying accessibility.
  - Persistent reservation for the shared disks.
- Creating:
  - File that contains the nodes description for the cluster.
  - Spectrum Scale cluster by using the nodes descriptor file.
  - File that contains the disks description and adding the network shared disks to the cluster (by using the disks descriptor file).
- Applying the Spectrum Scale license to the cluster nodes (as required).

- Starting the Spectrum Scale daemon on all nodes and changing the cluster quorum to use node with tiebreaker disks.
- Adding the file systems to the cluster.
- Checking the configuration and verifying the file systems.

Configuration steps

We completed the following steps:

1. Spectrum Scale software was installed on our nodes, as shown in Example 6-26.

   We also set the path variable ($PATH) to include the Spectrum Scale binaries path.

Example 6-26 Spectrum Scale packages on our cluster nodes

```
# dsh lslpp -L gpfs.*|dshbak -c
HOSTS -------------------------------------------------------------------------
aop93cl093, aop93cld24
-------------------------------------------------------------------------------
  Fileset                      Level  State  Type  Description (Uninstaller)
  ----------------------------------------------------------------------------
  gpfs.adv                   5.0.3.3    C     F    GPFS Advanced Features
  gpfs.base                  5.0.3.3    C     F    GPFS File Manager
  gpfs.compression           5.0.3.2    C     F    GPFS Compresson Libraries
  gpfs.crypto                5.0.3.3    C     F    GPFS Cryptographic Subsystem
  gpfs.docs.data             5.0.3.3    C     F    GPFS Server Manpages and
                                                   Documentation
  gpfs.ext                   5.0.3.3    C     F    GPFS Extended Features
  gpfs.gskit               8.0.50.86    C     F    GPFS GSKit Cryptography
                                                   Runtime
  gpfs.license.adv           5.0.3.0    C     F    IBM Spectrum Scale Advanced
                                                   Edition License
  gpfs.msg.en_US             5.0.3.3    C     F    GPFS Server Messages - U.S.
                                                   English

# echo $PATH
/usr/bin:/etc:/usr/sbin:/usr/ucb:/usr/bin/X11:/sbin:/usr/java7_64/jre/bin:/usr/jav
a7_64/bin:/usr/lpp/mmfs/bin:/u02/app/19.0.0/grid/bin
```

2. We use the network configuration that is shown in Example 6-27 for our Spectrum Scale cluster.

   For our cluster configuration, we used `en0`.

   Because Spectrum Scale is used as back-end storage for Oracle RAC, we configured the local name resolution (`/etc/hosts`) for Spectrum Scale.

Example 6-27 Network used for Spectrum Scale

```
# dsh ifconfig en0|dshbak -c
HOSTS -------------------------------------------------------------------------
aop93cld24
-------------------------------------------------------------------------------
en0:
flags=1e084863,814c0<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64B
IT,CHECKSUM_OFFLOAD(ACTIVE),LARGESEND,CHAIN>
        inet XXX.XX.XX.123 netmask 0xffffff00 broadcast XXX.XX.XX.255
         tcp_sendspace 262144 tcp_recvspace 262144 rfc1323 1
```

```
HOSTS --------------------------------------------------------------------------
aop93cl093
--------------------------------------------------------------------------------
en0:
flags=1e084863,814c0<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64B
IT,CHECKSUM_OFFLOAD(ACTIVE),LARGESEND,CHAIN>
        inet XXX.XX.XX.93 netmask 0xffffff00 broadcast XXX.XX.XX.255
         tcp_sendspace 262144 tcp_recvspace 262144 rfc1323 1
```

3. We tested the secure shell password-less access between Spectrum Scale cluster nodes, as shown in Example 6-28. For more information about the SSH configuration, see *IBM Spectrum Scale Version 5.0.3: Concepts, Planning, and Installation Guide*, SC27-9567.

Example 6-28 Testing password-less remote command execution

```
# hostname
aop93cld24.pbm.ihost.com
# ssh aop93cl093 ssh localhost ssh aop93cld24 ssh localhost date
Sat Oct 10 10:48:37 EDT 2020
```

4. We use the disks for Spectrum Scale that are shown in Example 6-29 on page 90. We configured these disks with SCSI-3 Persistent reservation (only one of the disks that are shown as an example).

Notes: Consider the following points:

- Spectrum Scale changes the `PR_key_value` upon cluster services startup.
- During Network Shared Disk (NSD) configuration step, Spectrum Scale identifies the disk devices uniquely on both nodes, even if the disk device names (as seen by the AIX operating system) are not the same.
- For consistency, we recommend that the disk devices in AIX use the same names on both nodes. This suggestion can be achieved by identifying the disks on both systems based on their UUID (**`lsattr -El hdiskYY |grep unique_id`** or **`lscfg -vpl hdiskYY|grep hdiskYY`**) and then, by using the AIX **`rendev`** command.

Example 6-29 Disks to be used for Spectrum Scale

```
# dsh 'lspv |grep none'|dshbak -c
HOSTS --------------------------------------------------------------------------
aop93cld24, aop93cl093
--------------------------------------------------------------------------------
hdisk6          none                                None
hdisk7          none                                none
hdisk8          none                                none
hdisk9          none                                none
hdisk10         none                                none
hdisk11         none                                none
hdisk12         none                                none
hdisk13         none                                none
hdisk14         none                                none
hdisk15         none                                none
hdisk16         none                                none

##For all disks, on both nodes, we execute the following command:
# chdev -l hdiskYY -a reserve_policy=PR_shared -a PR_key_value=8888
# dsh 'lsattr -El hdisk16 |egrep "reserve_policy|PR_key"' |dshbak -c
HOSTS -------------------------------------------------------------------------
aop93cld24, aop93cl093
-------------------------------------------------------------------------------
PR_key_value    8888                    Persistant Reserve Key Value     True+
reserve_policy  PR_shared               Reserve Policy                   True+
```

5. We created the nodes descriptor file that is shown in Example 6-30 for Spectrum Scale Cluster definition.

Example 6-30 Spectrum Scale nodes descriptor file

```
# cat gpfs_nodes
aop93cld24:quorum-manager:
aop93cl093:quorum-manager:
```

Example 6-31 shows the Spectrum Scale cluster creation command.

Example 6-31 Spectrum Scale cluster creation

```
# mmcrcluster -N gpfs_nodes -r /usr/bin/ssh -R /usr/bin/scp -A
```

6. We applied the *Spectrum Scale license* in our cluster, as shown in Example 6-32.

Example 6-32 Applying the Spectrum Scale license

```
# mmchlicense server --accept -N gpfs_nodes

##Listing the license
# mmlslicense

 Summary information
---------------------
Number of nodes defined in the cluster:                         2
Number of nodes with server license designation:                2
Number of nodes with FPO license designation:                   0
Number of nodes with client license designation:                0
Number of nodes still requiring server license designation:     0
Number of nodes still requiring client license designation:     0
This node runs IBM Spectrum Scale Advanced Edition
```

7. We created the disks descriptor files that are shown in Example 6-33 for our cluster and added the NSDs to the cluster configuration. We also created the following files:
   - `gpfs_tie_disks`, which was used to define NSDs that were used for cluster tiebreaker configuration.
   - `gpfs_data_disks_oradata`, which was used to define NSDs that were used for Oracle shared data file system.

Example 6-33 Disks descriptor files

```
# cat gpfs_tie_disks
%nsd: device=/dev/hdisk6
      nsd=tie1
      usage=descOnly
      failureGroup=3
%nsd: device=/dev/hdisk7
      nsd=tie2
      usage=descOnly
      failureGroup=3
%nsd: device=/dev/hdisk8
      nsd=tie3
      usage=descOnly
      failureGroup=3

# cat gpfs_data_disks_oradata
%nsd: device=/dev/hdisk9
      nsd=data11
      servers=aop93cld24,aop93cl093
      usage=dataAndMetadata
      failureGroup=1
      pool=system
%nsd: device=/dev/hdisk10
      nsd=data12
      servers=aop93cld24,aop93cl093
      usage=dataAndMetadata
      failureGroup=1
      pool=system
%nsd: device=/dev/hdisk11
      nsd=data13
      servers=aop93cld24,aop93cl093
      usage=dataAndMetadata
      failureGroup=1
      pool=system
%nsd: device=/dev/hdisk12
      nsd=data14
      servers=aop93cld24,aop93cl093
      usage=dataAndMetadata
      failureGroup=1
      pool=system
%nsd: device=/dev/hdisk13
      nsd=data21
      servers=aop93cld24,aop93cl093
      usage=dataAndMetadata
      failureGroup=2
      pool=system
%nsd: device=/dev/hdisk14
      nsd=data22
      servers=aop93cld24,aop93cl093
      usage=dataAndMetadata
      failureGroup=2
      pool=system
%nsd: device=/dev/hdisk15
      nsd=data23
      servers=aop93cld24,aop93cl093
      usage=dataAndMetadata
      failureGroup=2
      pool=system
%nsd: device=/dev/hdisk16
      nsd=data24
      servers=aop93cld24,aop93cl093
      usage=dataAndMetadata
      failureGroup=2
      pool=system
```

Example 6-34 shows the NSD creation and verification in our cluster. Currently, the NSDs are not assigned to any file system.

Example 6-34 NSD creation

```
# mmcrnsd -F gpfs_tie_disks
# mmcrnsd -F gpfs_data_disks_oradata

##We list the NSDs:
# mmlsnsd

 File system   Disk name    NSD servers
---------------------------------------------------------------------------
 (free disk)   data11       (directly attached)
 (free disk)   data12       (directly attached)
 (free disk)   data13       (directly attached)
 (free disk)   data14       (directly attached)
 (free disk)   data21       (directly attached)
 (free disk)   data22       (directly attached)
 (free disk)   data23       (directly attached)
 (free disk)   data24       (directly attached)
 (free disk)   tie1         (directly attached)
 (free disk)   tie2         (directly attached)
 (free disk)   tie3         (directly attached)
```

8. Example 6-35 shows the cluster started and configured with NSD tiebreaker. Note the (*) after the Quorum value (1) in the last **mmgetstate** output command. This asterisk indicates that the Spectrum Scale cluster used nodes with tiebreaker disks as the cluster quorum.

Example 6-35 Cluster configuration with tiebreaker disks

```
# mmstartup -a
.......

# mmgetstate -aL

 Node number  Node name    Quorum  Nodes up  Total nodes  GPFS state   Remarks
----------------------------------------------------------------------------------
       1      aop93cld24      2       2          2       active       quorum node
       2      aop93cl093      2       2          2       active       quorum node

# mmchconfig tieBreakerDisks=”tie1;tie2;tie3”
........
# mmlsconfig tieBreakerDisks
tiebreakerDisks tie1;tie2;tie3

# mmgetstate -aL

 Node number  Node name   Quorum  Nodes up  Total nodes  GPFS state   Remarks
----------------------------------------------------------------------------------
       1      aop93cld24    1*        2          2       active       quorum node
       2      aop93cl093    1*        2          2       active       quorum node
```

Example 6-36 shows the cluster configuration. Note the following parameters:

```
worker1Threads 48
workerThreads 512
prefetchThreads 72
```

These parameters are recommended for deploying Oracle RAC database configurations as per Oracle's Doc ID 2587696.1.

Example 6-36 Cluster configuration

```
# mmlsconfig
Configuration data for cluster racgpfs.pbm.ihost.com:
-----------------------------------------------------
clusterName racgpfs.pbm.ihost.com
clusterId 2032712214958416533
autoload yes
dmapiFileHandleSize 32
minReleaseLevel 5.0.3.0
ccrEnabled yes
cipherList AUTHONLY
usePersistentReserve yes
failureDetectionTime 10
worker1Threads 48
workerThreads 512
prefetchThreads 72
minQuorumNodes 1
tiebreakerDisks tie1;tie2;tie3
adminMode central

File systems in cluster racgpfs.pbm.ihost.com:
----------------------------------------------
(none)
```

Example 6-37 shows the file system creation and activation. For Oracle RAC deployments, check the My Oracle Support Document Doc ID 2587696.1 for Spectrum Scale file system parameters recommendations.

> **Tip:** The file system we created contains a single pool that is named `system`. The `system` pool contains disks for data and metadata. The data and metadata disks are divided into two failure groups (1 and 2).
>
> Also, data and metadata mirroring is configured for all files. The two copies of each data and metadata block are stored in separate failure groups.
>
> In addition to the two failure groups that are used for data and metadata, the `system` pool also contains a disk that is used as a file system descriptor quorum (failure group 3).

We created the file system first and then, we added one disk to be used as file system descriptor quorum. We used the following NSD descriptor file for this purpose:

```
# cat gpfs_tie_oradata_disk
%nsd: device=/dev/hdisk6
      nsd=tie1
      usage=descOnly
      failureGroup=3
```

> **Important:** Specific Spectrum Scale file system parameters, such as the file system block size (and others) cannot be changed after the file system is created. These parameters must be set at file system creation.

Example 6-37 File system creation and configuration

```
# mmcrfs oradata -F gpfs_data_disks_oradata -A yes -B 1M -D posix -E no -M2 -m2
-R2 -r2 -S no -T /oradata

# mmadddisk oradata -F gpfs_tie_oradata_disk

# mmmount all -a

# mmlsmount all -L

File system oradata is mounted on 2 nodes:
  XXX.XX.XX.123   aop93cld24
  XXX.XX.XX.93    aop93cl093
```

> **Tip:** For more information about the parameters that are used when the file system is created, see the man page of the **`mmcrfs`** command.

Example 6-38 shows the file system disk availability.

> **Note:** The distribution of the three file system descriptors is one per failure group.

Example 6-38 File system disk availability

```
# mmlsdisk oradata -L
disk         driver   sector     failure holds    holds                                    storage
name         type       size       group metadata data  status        availability disk id pool         remarks
------------ -------- ------ ----------- -------- ----- ------------- ------------ ------- ------------ ---------
data11       nsd         512           1 yes      yes   ready         up                 1 system        desc
data12       nsd         512           1 yes      yes   ready         up                 2 system
data13       nsd         512           1 yes      yes   ready         up                 3 system
data14       nsd         512           1 yes      yes   ready         up                 4 system
data21       nsd         512           2 yes      yes   ready         up                 5 system        desc
data22       nsd         512           2 yes      yes   ready         up                 6 system
data23       nsd         512           2 yes      yes   ready         up                 7 system
data24       nsd         512           2 yes      yes   ready         up                 8 system
tie1         nsd         512           3 no       no    ready         up                 9 system        desc
Number of quorum disks: 3
Read quorum value:      2
Write quorum value:     2
```

9. We configured the directory structure and permissions that are shown in Example 6-39 for deploying Oracle Clusterware and one Oracle RAC database. Oracle Clusterware Registry (OCR) and Vote files are deployed in the `/oradata/crs_files2` directory; the Oracle data files are deployed in the `/oradata/itsodb` directory.

Example 6-39 Directory structure in shared file system

```
# ls -all /oradata
total 1075
drwxr-xr-x    8 root     oinstall     262144 Sep 29 18:13 .
drwxr-xr-x   32 root     system         4096 Oct 08 12:57 ..
```

```
drwxr-xr-x    2 root     system         4096 Sep 29 16:45 .mmSharedTmpDir
dr-xr-xr-x    2 root     system         8192 Dec 31 1969  .snapshots
drwxr-xr-x    2 grid     oinstall       8192 Oct 10 09:28 crs_files2
drwxrwxr-x    4 oracle   oinstall       4096 Sep 29 22:20 itsodb
```

6.6 Deploying Oracle software

In this section, we provide information about the Oracle Grid and Oracle Database configuration we implemented in our two node cluster.

> **Important:** We do not provide step-by-step Oracle Grid and Oracle Database installation instructions because this process is documented by Oracle and guided by the Oracle Universal Installer (OUI). Instead, we focus on the configuration parameters we selected for our deployment.
>
> Although specific actions can be performed on a single node during the installation and configuration of Oracle software, some tasks must be performed on both nodes.

6.6.1 Preparing for OUI GUI

In preparation for our deployment (we chose to use the Oracle Universal Installer GUI), we deployed a basic VNC configuration. We installed and configured the VNC server packages on `node1` only.

> **Tip:** In our test environment, we used an older version of VNC. You can use the graphics server of your choice (for example, `tightvnc`).

The VNC configuration files can be found in each user's home directory in the `~/.vnc` folder. Example 6-40 shows the VNC server that is configured for `grid` and `oracle` users.

Example 6-40 VNC stared for grid and oracle users (node1)

```
# rpm -qa |grep vnc
vnc-3.3.3r2-6.ppc
# ps -aef |grep vnc |egrep “grid|oracle”
grid  9372104        1   0   Sep 29      -  0:00 Xvnc :2 -desktop X -httpd
/opt/freeware/vnc/classes -auth /home/grid/.Xauthority -geometry 1024x768 -depth 8 -rfbwait
120000 -rfbauth /home/grid/.vnc/passwd -rfbport 5902 -nolisten local -fp
/usr/lib/X11/fonts/,/usr/lib/X11/fonts/misc/,/usr/lib/X11/fonts/75dpi/,/usr/lib/X11/fonts/100dpi
/,/usr/lib/X11/fonts/ibm850/,/usr/lib/X11/fonts/Type1/

oracle 28508496        1   0   Sep 29      -  0:00 Xvnc :3 -desktop X -httpd
/opt/freeware/vnc/classes -auth /home/oracle/.Xauthority -geometry 1024x768 -depth 8 -rfbwait
120000 -rfbauth /home/oracle/.vnc/passwd -rfbport 5903 -nolisten local -fp
/usr/lib/X11/fonts/,/usr/lib/X11/fonts/misc/,/usr/lib/X11/fonts/75dpi/,/usr/lib/X11/fonts/100dpi
/,/usr/lib/X11/fonts/ibm850/,/usr/lib/X11/fonts/Type1/

# netstat -an|grep LISTEN|grep 590
tcp4       0      0  *.5902                *.*                    LISTEN
tcp4       0      0  *.5903                *.*                    LISTEN
```

6.6.2 Extracting the Oracle Grid 19c installation archive

We use an NFS mounted file system (node1 only) as a repository for the Oracle 19c grid_home.zip and db_home.zip installation packages:

```
# mount |grep -i nfs
XXX.XX.XX.52 /stage/oracle    /mnt1           nfs3   Sep 29 17:29
# ls -l /mnt1/db/19.3
total 20354432
-rwxr-xr-x    1 root     system   3472058711 Mar 24 2020
AIX.PPC64_193000_db_home.zip
-rwxr-xr-x    1 root     system   3604875591 Mar 24 2020
AIX.PPC64_193000_grid_home.zip
```

We log on as grid and extract the Oracle 19c grid_home.zip installation archive in the directory that is shown in Example 6-41.

Example 6-41 Extracting the grid_home.zip (node1 only)

```
$ mkdir -p /u02/app/19.0.0/grid <this will create the directory, if not already
done so>
$ cd /u02/app/19.0.0/grid

$ unzip  /mnt1/db/19.3/AIX.PPC64_193000_grid_home.zip
.........
```

6.6.3 Extracting the Oracle Database 19c installation archive

We use an NFS mounted file system (node1 only) as a repository for the Oracle 19c grid_home.zip and db_home.zip installation packages:

```
# mount |grep -i nfs
XXX.XX.XX.52 /stage/oracle    /mnt1           nfs3   Sep 29 17:29
# ls -l /mnt1/db/19.3
total 20354432
-rwxr-xr-x    1 root     system   3472058711 Mar 24 2020
AIX.PPC64_193000_db_home.zip
-rwxr-xr-x    1 root     system   3604875591 Mar 24 2020
AIX.PPC64_193000_grid_home.zip
```

We log on as oracle and extract the Oracle 19c db_home.zip installation archive in the directory that is shown in Example 6-42.

Example 6-42 Extracting the db_home.zip (node1 only)

```
# su - oracle
$ mkdir -p /u02/app/oracle/product/19.0.0/dbhome_1
$ cd /u02/app/oracle/product/19.0.0/dbhome_1

$ unzip  /mnt1/db/19.3/AIX.PPC64_193000_db_home.zip
.........
```

6.6.4 Installing the Oracle Grid infrastructure

To install the Oracle Grid Infrastructure, we followed the instructions that are provided in *Oracle Grid Infrastructure Installation and Upgrade Guide 19c for IBM AIX on POWER Systems (64-Bit)*, E96274-03.

> **Note:** Before starting the installation process, check the My Oracle Support document *INS-06006 GI RunInstaller Fails If OpenSSH Is Upgraded to 8.x* (Doc ID 2555697.1).
>
> You can "pre-patch" the 19.3 binaries before running the installer with an Oracle Release Update, which contains the fix for this issue. In this case, you do not need *INS-06006 GI RunInstaller Fails If OpenSSH Is Upgraded to 8.x* (Doc ID 2555697.1).
>
> The Patch ID that contains the fix for this issue is 32545008, which is part of the April 2021 Release Update (RU):
>
> ```
> gridSetup.sh -applyRU /<Staging_Path>/grid/32545008
> ```
>
> Where the `<Staging_Path>` is the staging directory where the RU from Apr 2021 was extracted (in our test environment this. this directory is `/u02/stage`).

At the end of the Oracle Grid installation process, we checked the results by using the following procedure:

1. We logged on to `node1` as `root` and checked the path variable:

   ```
   # echo $PATH
   /usr/bin:/etc:/usr/sbin:/usr/ucb:/usr/bin/X11:/sbin:/usr/java7_64/jre/bin:/usr/
   java7_64/bin:/usr/lpp/mmfs/bin:/u02/app/19.0.0/grid/bin
   ```

2. From the VNC GUI as the `grid` user, we started the installation process:

   ```
   $ /u02/app/19.0.0/grid/gridSetup.sh
   ```

   We then followed the OUI menus and instructions. This installation does not use or create an Oracle ASM configuration because ASM is not required for Oracle RAC with Spectrum Scale deployments.

3. Example 6-43 shows the Oracle Grid infrastructure that was installed, configured, and running. Because we used shared file system (Spectrum Scale) for Oracle Clusterware Repository (OCR) and Vote files, ASM (although installed) is *not* configured and has no `ONLINE` resources.

   > **Note:** In our test configuration, we used only one SCAN VIP. Oracle recommends that three SCAN VIP addresses (configured in DNS) are used for standard deployments.

Example 6-43 Cluster Ready Services

```
# crsctl status resource -t
--------------------------------------------------------------------------------
Name           Target  State        Server                   State details
--------------------------------------------------------------------------------
Local Resources
--------------------------------------------------------------------------------
ora.LISTENER.lsnr
               ONLINE  ONLINE       aop93cld24               STABLE
ora.net1.network
               ONLINE  ONLINE       aop93cld24               STABLE
ora.ons
```

```
               ONLINE  ONLINE       aop93cld24               STABLE
--------------------------------------------------------------------------------
Cluster Resources
--------------------------------------------------------------------------------
ora.ASMNET1LSNR_ASM.lsnr(ora.asmgroup)
      1        OFFLINE OFFLINE                               STABLE
      2        OFFLINE OFFLINE                               STABLE
      3        OFFLINE OFFLINE                               STABLE
ora.LISTENER_SCAN1.lsnr
      1        ONLINE  ONLINE       aop93cld24               STABLE
ora.aop93cld24.vip
      1        ONLINE  ONLINE       aop93cld24               STABLE
ora.asm(ora.asmgroup)
      1        OFFLINE OFFLINE                               STABLE
      2        OFFLINE OFFLINE                               STABLE
      3        OFFLINE OFFLINE                               STABLE
ora.asmnet1.asmnetwork(ora.asmgroup)
      1        OFFLINE OFFLINE                               STABLE
      2        OFFLINE OFFLINE                               STABLE
      3        OFFLINE OFFLINE                               STABLE
ora.cvu
      1        ONLINE  ONLINE       aop93cld24               STABLE
ora.qosmserver
      1        ONLINE  ONLINE       aop93cld24               STABLE
ora.scan1.vip
      1        ONLINE  ONLINE       aop93cld24               STABLE
--------------------------------------------------------------------------------
```

The Oracle Clusterware Repository and Vote files are shown in Example 6-44.

Example 6-44 OCR and Vote files (located in shared file system)

```
# ocrcheck -config
Oracle Cluster Registry configuration is :
         Device/File Name         : /oradata/crs_files2/ocr1
         Device/File Name         : /oradata/crs_files2/ocr2
         Device/File Name         : /oradata/crs_files2/ocr3

# crsctl query css votedisk
##  STATE    File Universal Id                File Name Disk group
--  -----    -----------------                --------- ---------
 1. ONLINE   89a2a5a443554ff0bffc97a475b39007 (/oradata/crs_files2/vdsk1) []
 2. ONLINE   e327ca40f9084fd5bff12c0daad9f606 (/oradata/crs_files2/vdsk2) []
 3. ONLINE   06bffdfc14614f74bff47c440ae7b93e (/oradata/crs_files2/vdsk3) []
Located 3 voting disk(s).
```

6.6.5 Installing the Oracle Database

To install the Oracle Database, we followed the instructions that are provided in *Oracle Database Installation Guide - 19c for IBM AIX on POWER Systems (64-bit)*, E96437-04.

During the Oracle Database software installation process, we chose to install software. The instance and database was created later by using the Oracle `dbca` utility.

We completed the following steps:

1. From the VNC terminal window, we logged on as `oracle` and started the Oracle Database 19c Installer:

   ```
   $ cd /u02/app/oracle/product/19.0.0/dbhome_1
   $ ./runInstaller
   ```

2. We chose **Set Up Software Only**, clicked **Next**, and then, and selected **Oracle Real Application Clusters database installation**.
3. We followed the instructions and finalized the software installation. We checked the new Oracle Clusterware resources that were created, as shown in Example 6-45. At this time, the database was yet created; therefore, the instance resources are `OFFLINE`.

Example 6-45 Oracle Database resource

```
# crsctl status resource -t
--------------------------------------------------------------------------------
Name           Target  State        Server                   State details
--------------------------------------------------------------------------------
Local Resources
--------------------------------------------------------------------------------
ora.LISTENER.lsnr
               ONLINE  ONLINE       aop93cl093               STABLE
               ONLINE  ONLINE       aop93cld24               STABLE
ora.net1.network
               ONLINE  ONLINE       aop93cl093               STABLE
               ONLINE  ONLINE       aop93cld24               STABLE
ora.ons
               ONLINE  ONLINE       aop93cl093               STABLE
               ONLINE  ONLINE       aop93cld24               STABLE
--------------------------------------------------------------------------------
Cluster Resources
--------------------------------------------------------------------------------
ora.ASMNET1LSNR_ASM.lsnr(ora.asmgroup)
      1        OFFLINE OFFLINE                               STABLE
      2        OFFLINE OFFLINE                               STABLE
      3        OFFLINE OFFLINE                               STABLE
ora.LISTENER_SCAN1.lsnr
      1        ONLINE  ONLINE       aop93cld24               STABLE
ora.aop93cl093.vip
      1        ONLINE  ONLINE       aop93cl093               STABLE
ora.aop93cld24.vip
      1        ONLINE  ONLINE       aop93cld24               STABLE
ora.asm(ora.asmgroup)
      1        OFFLINE OFFLINE                               STABLE
      2        OFFLINE OFFLINE                               STABLE
      3        OFFLINE OFFLINE                               STABLE
ora.asmnet1.asmnetwork(ora.asmgroup)
```

```
      1        OFFLINE OFFLINE                              STABLE
      2        OFFLINE OFFLINE                              STABLE
      3        OFFLINE OFFLINE                              STABLE
ora.cvu
      1        ONLINE  ONLINE       aop93cld24              STABLE
ora.itsodb.db
      1        OFFLINE OFFLINE                              STABLE
      2        OFFLINE OFFLINE                              STABLE
ora.qosmserver
      1        ONLINE  ONLINE       aop93cld24              STABLE
ora.scan1.vip
      1        ONLINE  ONLINE       aop93cld24              STABLE
--------------------------------------------------------------------------------
```

4. After the database software was installed, we started the Database Configuration Assistant from the VNC terminal window:

   ```
   $ /u02/app/oracle/product/19.0.0/dbhome_1/dbca
   ```

5. We selected **Create a Database**, and then, followed the instructions. We chose to configure a database named `itsodb`, with two instances (`itsodb1/node1` and `itsodb2/node2`). The shared file system location for this database is `/oradata/itsodb`.
6. For our installation, we selected **Sample schemas** and **Oracle Enterprise Manager (EM) database express**.
7. After the installation completed, we checked the configuration by using the commands that are shown in Example 6-46.

Example 6-46 Clusterware showing database status

```
# crsctl status resource -t
--------------------------------------------------------------------------------
Name           Target  State        Server                   State details
--------------------------------------------------------------------------------
Local Resources
--------------------------------------------------------------------------------
ora.LISTENER.lsnr
               ONLINE  ONLINE       aop93cl093              STABLE
               ONLINE  ONLINE       aop93cld24              STABLE
ora.net1.network
               ONLINE  ONLINE       aop93cl093              STABLE
               ONLINE  ONLINE       aop93cld24              STABLE
ora.ons
               ONLINE  ONLINE       aop93cl093              STABLE
               ONLINE  ONLINE       aop93cld24              STABLE
--------------------------------------------------------------------------------
Cluster Resources
--------------------------------------------------------------------------------
ora.ASMNET1LSNR_ASM.lsnr(ora.asmgroup)
      1        OFFLINE OFFLINE                              STABLE
      2        OFFLINE OFFLINE                              STABLE
      3        OFFLINE OFFLINE                              STABLE
ora.LISTENER_SCAN1.lsnr
      1        ONLINE  ONLINE       aop93cl093              STABLE
ora.aop93cl093.vip
      1        ONLINE  ONLINE       aop93cl093              STABLE
ora.aop93cld24.vip
```

```
      1        ONLINE  ONLINE       aop93cld24               STABLE
ora.asm(ora.asmgroup)
      1        OFFLINE OFFLINE                               STABLE
      2        OFFLINE OFFLINE                               STABLE
      3        OFFLINE OFFLINE                               STABLE
ora.asmnet1.asmnetwork(ora.asmgroup)
      1        OFFLINE OFFLINE                               STABLE
      2        OFFLINE OFFLINE                               STABLE
      3        OFFLINE OFFLINE                               STABLE
ora.cvu
      1        ONLINE  ONLINE       aop93cl093               STABLE
ora.itsodb.db
      1        ONLINE  ONLINE       aop93cld24               Open,HOME=/u02/app/o
                                                             racle/product/19.0.0
                                                             /dbhome_1,STABLE
      2        ONLINE  ONLINE       aop93cl093               Open,HOME=/u02/app/o
                                                             racle/product/19.0.0
                                                             /dbhome_1,STABLE
ora.qosmserver
      1        ONLINE  ONLINE       aop93cl093               STABLE
ora.scan1.vip
      1        ONLINE  ONLINE       aop93cl093               STABLE
--------------------------------------------------------------------------------
```

8. We also checked the LISTENER configuration, as shown in Example 6-47.

Example 6-47 Check the LISTENER configuration

```
##As “grid” user
$ srvctl config listener -all
Name: LISTENER
Type: Database Listener
Network: 1, Owner: grid
Home: <CRS home>
  /u02/app/19.0.0/grid on node(s) aop93cl093,aop93cld24
End points: TCP:1521
Listener is enabled.
Listener is individually enabled on nodes:
Listener is individually disabled on nodes:

##As “oracle” user on node1
$ lsnrctl status LISTENER

LSNRCTL for IBM/AIX RISC System/6000: Version 19.0.0.0.0 - Production on
10-OCT-2020 16:09:40

Copyright (c) 1991, 2020, Oracle.  All rights reserved.

Connecting to (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1521))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                  TNSLSNR for IBM/AIX RISC System/6000: Version 19.0.0.0.0
- Production
Start Date                08-OCT-2020 09:13:29
Uptime                    2 days 6 hr. 56 min. 11 sec
```

```
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File   /u02/app/19.0.0/grid/network/admin/listener.ora
Listener Log File
/u02/app/19.0.0/grid_base/diag/tnslsnr/aop93cld24/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=LISTENER)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=129.40.93.123)(PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=129.40.93.22)(PORT=1521)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcps)(HOST=aop93cld24)(PORT=5500))(Security=(my_wa
llet_directory=/u02/app/oracle/product/19.0.0/dbhome_1/admin/itsodb/xdb_wallet))(P
resentation=HTTP)(Session=RAW))
Services Summary...
Service "8939c30fdc8c01b2e0530af11a191106" has 1 instance(s).
  Instance "itsodb1", status READY, has 2 handler(s) for this service...
Service "b07f99e1d79401e2e05381285d5dd6c5" has 1 instance(s).
  Instance "itsodb1", status READY, has 2 handler(s) for this service...
Service "itsodb" has 1 instance(s).
  Instance "itsodb1", status READY, has 2 handler(s) for this service...
Service "pdb" has 1 instance(s).
  Instance "itsodb1", status READY, has 2 handler(s) for this service...
```

7

Monitoring and performance troubleshooting

This chapter discusses monitoring and performance troubleshooting aspects of Oracle on Power Systems.

This chapter includes the following topics:

- 7.1, “Monitoring” on page 106.
- 7.2, “Server monitoring” on page 106.
- 7.3, “Database monitoring” on page 107.
- 7.4, “Performance troubleshooting” on page 108.

7.1 Monitoring

We recommend that clients monitor the server *and* database.

7.2 Server monitoring

Nmon is the main tool for collecting performance data on AIX to allow us to view and record operating system performance.

For daily monitoring in a production environment, we recommend that **nmon** is started and includes the following options:

```
nmon -s60 -c1440 -f -d -V -^ -L -A -M
```

Note: The 60s capture interval for continuous monitoring might be too short as it generates too much data for analysis. AIX by default runs topas_nmon collection. You can disable it if **nmon** is used instead.

Where:

- -s: The snapshot interval (in seconds) that defaults to 2. For a production system, 60 is sufficient for daily performance monitoring. The reason for reducing the frequency is to reduce the size of the output file. The effect on performance is not significant.
- -c 1440: Gives 24 hours of capture.
- -f: Specifies that the output is in spreadsheet format.
- -d: Includes the Disk Service Time section in the view.
- -V: Includes disk volume group section.
- -^: Includes the Fibre Channel (FC) sections.
- -L: Includes the large page analysis section.
- -A: Includes the Asynchronous I/O section in the view.
- -M: Include memory page size-specific data.

The command can be added to **crontab** to run at midnight each night to produce a daily nmon monitoring file. These files can be useful as a reference if performance issues occur.

If you are investigating a specific issue that runs for a shorter period, you can decrease the interval length by using the **-s** parameter to capture more detailed performance data. In benchmarks, we often set the capture interval length to 10 seconds and run the **nmon** command during the test workload that we want to monitor.

Other options are available that can be useful to activate if you are investigating a specific issue; for example, if you are investigating a CPU spike, it is useful to add the **-T** option. This option includes the top processes and their call parameters in the output.

For more information about the full documentation for the nmon monitoring tool, see this IBM Documentation web page.

We recommend that you collect nmon data on your AIX LPARs and on all involved Virtual I/O Servers.

After you capture the nmon data, it can be analyzed by using several tools (most often with the nmon analyzer tool). The nmon tool is an excel spreadsheet that turns the raw data into a readable report. For more information, see this IBM Developer web page.

For more information about alternative post-processing tools for nmon data, see this hweb page.

7.3 Database monitoring

Many third-party tools are available that can be used to monitor the database. The main tool that is discussed in this section is the AWR report from Oracle. It is the most useful tool for investigating a database performance issue after the event occurred.

> **Note:** Although Enterprise Manager Cloud Control allows the same data to be reviewed in real time, this topic is not included here.

To create AWR reports, you must acquire the Oracle software license for the chargeable add-on option Diagnostic and Tuning Pack. Then, you activate the Server Manageability Pack option by using the following command by way of Oracle’s SQL*Plus tool:

```
ALTER SYSTEM SET
control_management_pack_access='DIAGNOSTIC+TUNING' SCOPE=BOTH;
```

The AWR report provides operating system data for the server, but the numbers are not always 100% accurate for the following reasons:

- The operating system level CPU data is for the server or partition and not for the individual database instance.
- Oracle does not record DB CPU use correctly for AIX. You never see 90% CPU usage for a database on AIX when SMT2, SMT4, or SMT8 are used because Oracle does not include time that is spent waiting on the CPU from threads other than the first thread on each core.

  > **Note:** Oracle includes system CPU statistics that show USR/SYS/WAIT as reported by AIX. Where Oracle can be misleading is in reporting the CPU time that is used by the DB. The higher the SMT level, the higher the discrepancy between reported CPU that is used by Oracle and the actual CPU that is used.

- The %DB time can add up to over 100. Despite this issue, knowing the proportion of activity in relation to other events is still a valuable metric.
- I/O statistics must be compared with the same statistics from the AIX hdisk layer and the storage system. The observed values should closely match. If a significant discrepancy is observed, bottlenecks exist somewhere in the I/O stack or SAN infrastructure that must be investigated.

The AWR report is based on the delta between two metric snapshots in time. The longer the time between those two snapshots, the more difficult it is to determine what occurred. For example, the reported CPU utilization data is the same for a process that ran at 100% CPU for 6 minutes or 10% CPU for 1 hour when the AWR report covers a 1-hour period.

The maximum frequency for automated AWR snapshots is every 10 minutes. When investigating a performance issue (especially an issue that does not last long), frequent snapshots can be beneficial. It helps to eliminate some of the “background noise” of normal activity and focuses on the issue.

If you are manually capturing the AWR snapshots for a specific issue, creating two snapshots during the peak allows you to capture a clear image of what is occurring during the peak.

From Oracle 12c onwards, the AWR report contains the output from ADDM. This inclusion can be useful for finding performance issues that were detected by the Oracle Database. Often, a solution is proposed.

If you do not have the license for the Diagnostic and Tuning pack, STATSPACK is still available but provides significantly less information and analysis.

7.4 Performance troubleshooting

In this section, we discuss on the following main areas that must be considered:

- CPU
- Memory
- I/O

7.4.1 CPU

A 100% CPU consumption rate is not necessarily an issue. The true indicator of a system that is overloaded is the run queue. If the run queue is higher than the number of CPU threads, the system is CPU bound. This information can be found in the PROC tab of the `nmon` report.

Running **`nmon`** from the command line allows users to see CPU use in real time. It also helps users to review top processes that use CPU by allowing users to determine whether an Oracle process or some other workload is using the CPU resources.

The AWR report can help to identify individual SQL requests that can be using excessive CPU time.

7.4.2 Memory

Oracle asks for a page space (or swap space) that is twice the size of the memory available at the time of running the installer. On AIX, we do not want to have any paging to paging space at all. If the paging space is being used, a shortage of physical memory exists (or did exist) that forced AIX to page memory pages to paging space.

You can see the paging space-related I/O activity in the PAGE tab of the `nmon` report. Current usage of paging space can be determined by running **`lsps -a`**.

> **Note:** Workloads exist in which not using CIO can be beneficial. The types of memory that are allocated are shown in the MEMNEW tab in the `nmon` report.

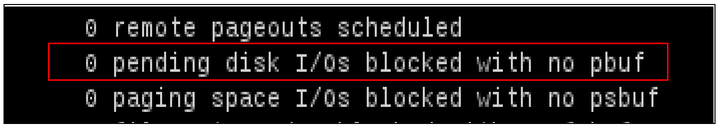
If insufficient memory was reserved for the operating system and the database connections, a high number of connections can result in memory swapping to paging space.

7.4.3 I/O

By using the nmon report, we check the general performance of the I/O (by using the **-d** option).

The following utilities can be used to help to discover issues:

- **vmstat -v** helps to detect a lack of physical buffers in logical volumes, as shown in Figure 7-1. This utility is for jfs2 only, not ASM.



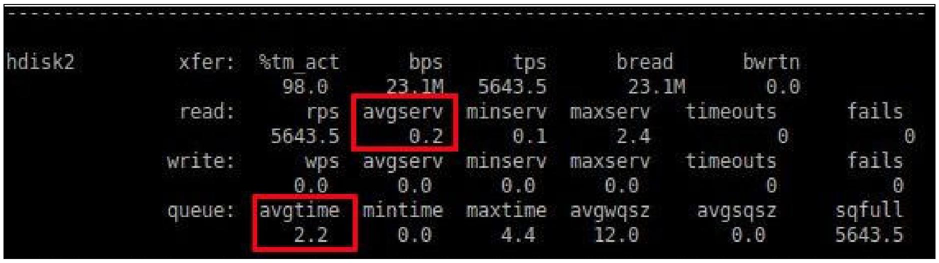
```
        0 remote pageouts scheduled
        0 pending disk I/Os blocked with no pbuf
        0 paging space I/Os blocked with no psbuf
```

Figure 7-1 Output checking lack of physical buffers if any

  You also find this information about the BBBP tab of your nmon report with the output from **vmstat -v** and then again with the output of ending **vmstat -v**.

  By comparing the two, you can determine if the number of blocked pbufs increased during the nmon capture. If so, this issue can be resolved by adding LUNs or by increasing the pv_pbuf_count by using of the **lvmo** command.

- **iostat -D** shows if queuing is occurring at the operating system level, as shown in Figure 7-2.



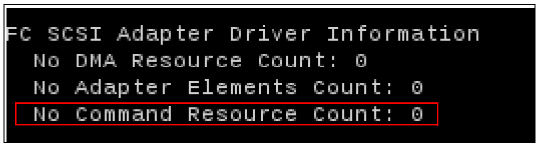
```
--------------------------------------------------------------------------------

hdisk2         xfer:  %tm_act      bps      tps      bread      bwrtn
                         98.0    23.1M   5643.5      23.1M        0.0
               read:      rps  avgserv  minserv  maxserv   timeouts      fails
                       5643.5      0.2      0.1      2.4          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
                          2.2      0.0      4.4     12.0        0.0     5643.5
```

Figure 7-2 iostat -D showing queuing information

  Figure 7-2 shows the avgserv of the I/O read is 0.2 ms, which is good but the avgtime that is spent in the queue of 2.2 ms is an issue. This issue can be resolved by increasing the queue depth on the disk or by adding \ LUNs. Queue avgtime is also in the DISKWAIT tab of nmon (by using the **-d** option).

- If queuing is occurring at the Fibre Channel (FC) adapter level, you see the output (see Figure 7-3) by using the **fcstat** command.



```
FC SCSI Adapter Driver Information
  No DMA Resource Count: 0
  No Adapter Elements Count: 0
  No Command Resource Count: 0
```

Figure 7-3 Fibre Channel adapter information

  This setting is similar to the queue_depth but for the adapters. The num_cmd_elements is set by using the **chdev -l fcsX -o num_cmd_elems=YYY** command, where fcsX is the name of the Fibre Channel adapter. The value of YYY is the sum of the queue_depths divided by the number of FC adapters. FC information also is in the nmon BBBF tab (with nmon option -^).

If you use VIO servers, check for any contention at that level. For example, if the VIO servers are starved of CPU, they can become a bottleneck for network or I/O adapters.

Note: The num_cmd_elems in the VIO Servers typically is set to maximum that is supported by SAN. It must not be lower than what is configured in client LPARs.

Oracle provides a tool that is called orion, which can be used to test I/O bandwidth and latency. This tool is used by the I/O calibrate function in the Oracle Database.

The difference between the two tools is that orion does not require a database to be created to work and I/O calibrate updates the statistics in the database for use by the Oracle optimizer.

Note: Both of these tools return a value with which you can compare the I/O capacity of different environments. However, they must be used with caution, especially in an active production environment because they attempt to saturate the I/O resources during their test.

8

Oracle Database and storage

Typically, database data is persisted on external storage subsystems. Performing read/write access to that data is paramount for a successful deployment of an Oracle Database.

This chapter highlights key components that are involved in a database I/O. Then, it discusses a proven approach to use available server and storage resources in support of database I/O.

This chapter includes the following topics:

- 8.1, “AIX I/O stack” on page 112.
- 8.2, “AIX I/O queues” on page 113.
- 8.3, “Data layout for Oracle Database” on page 115.

8.1 AIX I/O stack

Figure 8-1 shows the AIX components that are involved in block I/O, from the database to the physical storage device for the different storage methods.

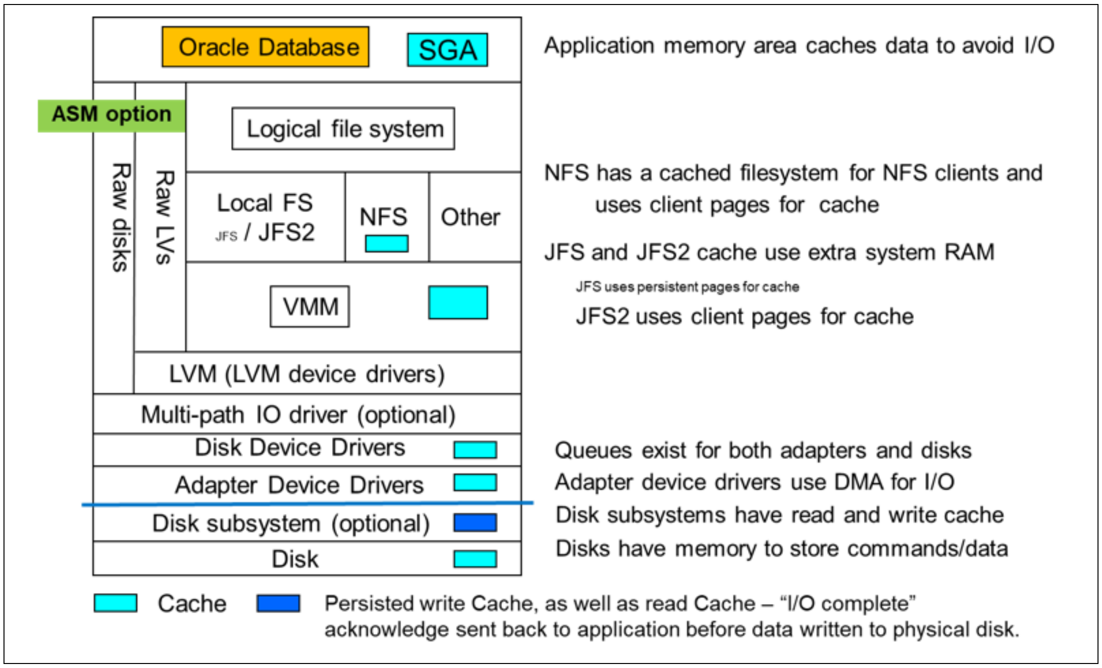

Oracle Database
SGA
Application memory area caches data to avoid I/O
ASM option
Logical file system
Raw disks
Raw LVs
Local FS
JFS / JFS2
NFS
Other
NFS has a cached filesystem for NFS clients and uses client pages for cache
JFS and JFS2 cache use extra system RAM
JFS uses persistent pages for cache
JFS2 uses client pages for cache
VMM
LVM (LVM device drivers)
Multi-path IO driver (optional)
Disk Device Drivers
Adapter Device Drivers
Disk subsystem (optional)
Disk
Queues exist for both adapters and disks
Adapter device drivers use DMA for I/O
Disk subsystems have read and write cache
Disks have memory to store commands/data
Cache
Persisted write Cache, as well as read Cache – “I/O complete” acknowledge sent back to application before data written to physical disk.

Figure 8-1 AIX components and block I/O

Figure 8-1 shows in light blue where data cache and buffers are implemented. For Oracle Databases, the most critical cache is the buffer cache in the Oracle SGA and is available regardless of what storage technology that is used on the lower layers.

Although the caches in the disk device driver and adapter device driver layers are comparatively small, they must be sized correctly to support the maximum concurrent number of active I/O the database is driving against the storage subsystem at any time.

The AIX JFS2 file system is the simplest and most optimally performing option to deploy a single database instance on a cooked file system. The depicted cache for the JFS2 file system is in most cases not used because it is for most Oracle workload types that are recommended to use the JFS2 Concurrent I/O (CIO) feature of AIX to minimize constraints on concurrent write access to data files. CIO and correct file system layout can provide I/O performance, such as raw devices or Oracle Automatic Storage Management (ASM), while still providing the convenience of a cooked file system.

For Oracle Real Application Cluster deployments, ASM or IBM Spectrum Scale (see Chapter 6, “Oracle RAC for AIX with IBM Spectrum Scale” on page 69) typically are chosen to provide the shared concurrent access to data from multiple servers.

Note: Starting with Oracle 12c, raw devices (disks or raw Logical Volumes [LV]) are supported as devices for ASM only. For more information, see Oracle Doc ID 578455.1.

8.2 AIX I/O queues

Queues and buffers are inserted at several points in the AIX I/O stack. In this section, we review those queues and buffers.

Figure 8-2 shows a typical deployment that is based on Fibre Channel (FC) attached external storage where the Fibre Channel Host Bus Adapters, physical or virtual (NPIV) are physically allocated to the AIX LPAR. This configuration typically is used only for workloads with high I/O requirements and sharing of FC adapters with other Logical Partitions (LPAR) in the physical server is not feasible.

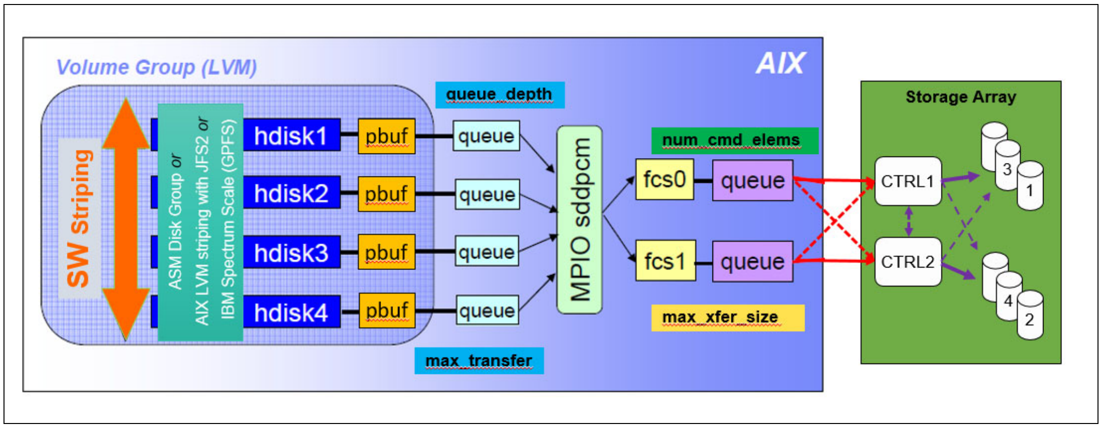


Figure 8-2 Fibre Channel deployment

Workload consolidation that results in improved resource usage and reduced TCO typically drives the sharing of FC adapters between multiple LPARs. Figure 8-3 shows a typical deployment with dual Virtual I/O Servers (VIOS) where the physical FC adapters are presented as N_Port ID virtualization (NPIV) adapters in the AIX LPAR.

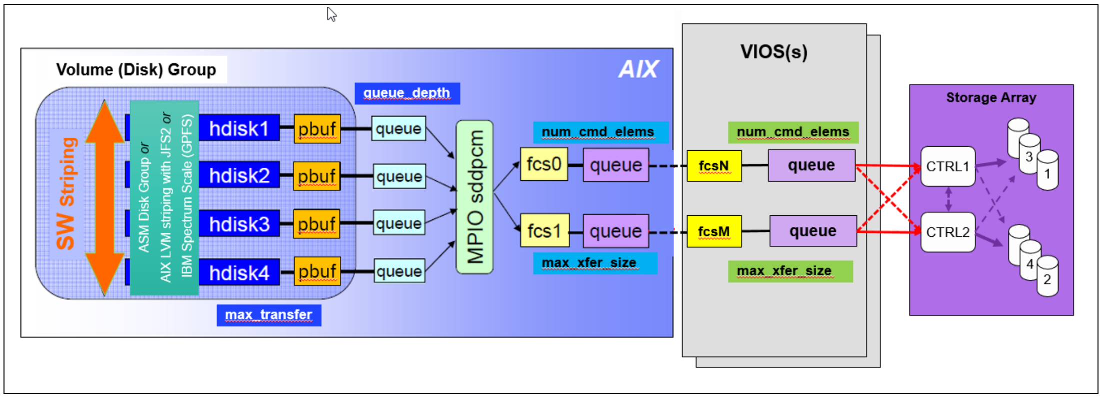


Figure 8-3 Deployment with dual Virtual I/O Servers (VIOS)

The use of virtual SCSI (vscsi) devices to map storage space from VIOS to client LPAR for use by an Oracle Database is discouraged because it introduces increased I/O latency and CPU usage in the VIOS as compared to the NPIV technology.

To simplify the example that is shown in Figure 8-3, only two FC adapters or ports are shown per VIOS. Those two ports connect to different SAN switches for highest reliability. The number of connections and active paths into the storage subsystem are configured to support the aggregated peak throughput requirements that are driven through the server FC adapters to that storage.

The physical adapters in the VIO Servers have a command queue, which restricts the number of concurrent I/O that can be driven by that adapter or port at any time. The size of that queue is specified by way of the `num_cmd_elems` adapter parameter (`fcsX`). The maximum value for this setting depends on the adapter type.

Older adapters typically support a value up to 2048, but more current 16 Gbit adapters support a value of 3200, or even 4096. Before setting or changing this value, verify with the storage vendor whether the SAN can support the number of FC adapter ports multiplied by `num_cmd_elems` concurrent pending I/O.

The `max_xfer_size` setting for a physical FC adapter has a dual meaning. On one side, it specifies the maximum size of an I/O that is driven against the SAN. On the other side, it influences how much DMA physical memory is allocated to the adapter. This parameter value typically is never reduced and increased only if guided so by IBM Support. Effective maximum I/O sizes typically are configured on the AIX `<hdisk>` layer and must be lower or equal to the value of `max_xfer_size`.

The NPIV adapter (virtual HBA) in the client LPAR also features a parameter `num_cmd_elems`, which constrains the number of concurrent I/O that is driven by the LPAR by way of the respective NPIV adapter.

Note: The `num_cmd_elems` in the client LPAR is limited to a value of 2048.

Storage capacity is made available to the client LPAR in the form of one or more storage volumes with assigned Logical Unit Numbers (LUNs), which can be accessed by way of a defined set of NPIV HBAs in the client LPAR. The AIX multi-path device driver, or storage vendor-specific alternative, automates the distribution of physical I/O to the available volumes over all available paths.

Each volume is represented as a single hdisk, or vendor-specific device name, such as `hdiskpower`, in AIX. Each hdisk includes a SCSI command queue and its depth is controlled by way of the `queue_depth` parameter. The maximum `queue_depth` value is 256 and it specifies how many I/O can be concurrently active to the underlying volume over all defined paths. If the hdisk queue is full, more I/O is placed into a separate wait queue, which is used in a first-come, first-served fashion.

The size restriction of the `hdisk queue_depth` drives the need to define and map more than one volume for an Oracle Database to use for data or redo. A good starting point for storing Oracle data, index, and temp files is eight volumes. For redo data, a different set of volumes at minimum four, but often eight volumes work well.

For Oracle use, those volumes are less than 2 TB and typically less than 1 TB. If more than 8 TB of space are required, more volumes are mapped (again, in multiples of eight). Eight was chosen based on typical characteristics of today’s SAN-attached storage solutions where the number of controllers in the SAN-attached storage are multiples of two for redundancy and I/O to a specific volume is typically routed by way of an “owning” controller. Only if that controller becomes unavailable is I/O rerouted to an alternative path.

Even if all LUNs are spread over all physical storage space in the SAN-attached storage, you still want to configure several volumes to not be I/O constrained by the queue depth of a single or small number of hdisks.

Recent AIX releases and FC adapters added the support for multiple I/O queues to enable significantly higher I/O rates. For more information, see this IBM Developer web page.

8.3 Data layout for Oracle Database

The approach to data and storage layout changed significantly over the years. That change was driven by what did and did not work over the longer term.

The initial approach was based on the idea to isolate database files based on function and usage; for example, define different pools of storage for data files and index files. this approach included the following key observations:

- Manually intensive planning and manual file-by-file data placement is time-consuming, resource intensive, and iterative.
- Best performance is achievable, but only with continues maintenance.
- It can lead to I/O hot spots over time that affect throughput capacity and performance.

The current approach, which also is the idea under ASM, is based on stripe and mirror everything (SAME). The “stripe” pertains to performance and load balancing and is relevant for this discussion. The “mirror” provides redundancy and availability. It is in a SAN-attached based environment that is implemented in the storage subsystem and not in AIX (it is not discussed further here). Oracle Databases with ASM on SAN-attached storage typically use EXTERNAL as the redundancy setting.

Benefits of this newer approach:

- Simplified planning
- Minimal manual intervention
- Evenly balanced I/O across all available physical components
- Good average I/O response time and object throughput capacity with no hot spots

SAME in an AIX environment features the following typical implementation options:

- Oracle ASM
- IBM Spectrum Scale; mostly in the context of Oracle RAC
- AIX logical volume manager; mostly in the context of single instance (non-RAC) deployments

In ASM devices, raw hdisks often are grouped by way of an ASM disk group. ASM automatically stripes all data within a disk group over all underlying devices. Typically, disk groups are used for:

- DATA: Data/index/temp
- FRA: Redo
- OCR: OCR/vote

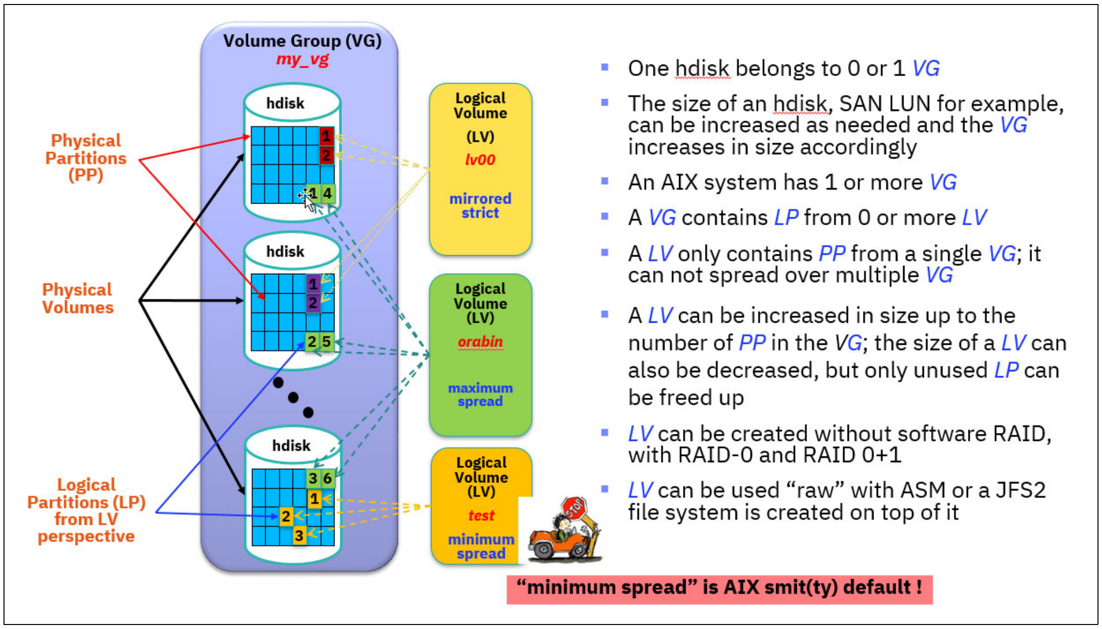
In the context of a production database deployment in AIX, the DATA disk group has eight (or a multiple of eight) devices. The FRA disk group has at minimum four (more typical is eight devices).

It is recommended to separate OCR/Vote (and management DB, if configured) into an independent disk group so the ASM disk groups for a database can be taken offline without shutting down Oracle cluster services.

For more information about Oracle ASM, see this web page.

The remainder of this section discusses how data striping can be efficiently implemented with the AIX logical volume manager for a single instance database that persists its data in AIX JFS2 file systems.

Figure 8-4 shows the key concepts of the AIX logical volume manager.



- One hdisk belongs to 0 or 1 *VG*
- The size of an hdisk, SAN LUN for example, can be increased as needed and the *VG* increases in size accordingly
- An AIX system has 1 or more *VG*
- A *VG* contains *LP* from 0 or more *LV*
- A *LV* only contains *PP* from a single *VG*; it can not spread over multiple *VG*
- A *LV* can be increased in size up to the number of *PP* in the *VG*; the size of a *LV* can also be decreased, but only unused *LP* can be freed up
- *LV* can be created without software RAID, with RAID-0 and RAID 0+1
- *LV* can be used “raw” with ASM or a JFS2 file system is created on top of it

“minimum spread” is AIX smit(ty) default !

Figure 8-4 AIX logical volume manager concepts

As with ASM, AIX groups hdisks into volume groups. A minimum of three AIX volume groups is recommended for the deployment of a production Oracle Database:

- ORAVG: Oracle binaries: Can be a single volume or two volumes.
- DATAVG: Oracle data/index/temp: Contains eight volumes or multiples of eight.
- REDOVG: Oracle redo; flash recovery: A minimum four volumes, (eight is better).

To stripe data over all hdisk devices in a volume group, AIX supports the following approaches, which are discussed in this section:

- Striping that is based on logical volume striping
- Striping by way of PP striping or PP spreading; for example, see the `orabin` LV in Figure 8-4.

Data striping by way of by way of logical volume striping

The following process is used to create an AIX JFS2 file system for an Oracle Database based on logical volume striping:

> **Note:** The specified block size (`agblksize`) must be adjusted to `512` for the file system that contains redo log files.

1. Allocate volumes from storage (typically in multiples of 8) and map to LPAR.
2. Discover disks by using the **`cfgmgr`** command.
3. Configure the FC adapter and hdisk parameters.
4. Export `disklist="hdiskX hdiskY …"`.
5. Create the DATAVG volume group and specify all the corresponding hdisks. Consider the specified PP size of 256 MB and the implications for future growth as shown in the following example:

```
mkvg -S -s 256 -y DATAVG $disklist
```

6. Create the striped logical volume with a 1 MB strip size over all hdisks in DATAVG:

   ```
   mklv -y oradatalv -S 1M -a e  -t jfs2 -x <max # PP> DATAVG <size of LV in PP>
   $disklist
   ```

7. Create JFS2 file system with inline log on the created LV:

   ```
   crfs -v jfs2 -d'oradatalv' -m /oradata -A'yes' -p'rw' -a agblksize='4096' -a
   logname='INLINE'
   ```

8. Mount the file system:

   ```
   mount /oradata
   ```

If striped logical volumes are used, consider and plan for the following points:

- All allocated volumes and hdisks in a VG must be of the same size.
- The LV is stripped over some number of hdisks (typically all hdisks in a VG) from one VG.
- The LV space allocation can grow only in multiples of N times the PP size. For example, N=8, PP size=1 GB ' 8G, 16G, 24G, …
- A file system (FS) on top of a stripped LV always grows with the same increments (N * PP) so that no space is wasted. You must create the stripped LV first and then, create the JFS2 file system on top of the striped LV.
- If any of the N hdisks runs out of available PP, attempts to grow the LV fail. The following options are available to resolve this issue:
  - Grow the underlying volumes dynamically by way of SAN methods and discover the new size dynamically by using the **`chvg -g <VG name>`** command. Minimum size increase for each volume is PP size and increases are in full multiples of PP size.

    This option is preferred because it requires no changes in the SAN configuration and host mapping. It must be verified with the storage vendor if any dynamic resize limitations exist.
  - Add hdisks (volumes) to the VG and expand the LV to those new volumes. This technique is called a *stripe column*. The system administrator manually adds the hdisks to the VG and then expands the LV.

    This option requires significant SAN and storage subsystem changes and more work by the AIX administrator. Adding volumes also affects FlashCopy configurations, for example.

Data striping by way of PP spreading

The following process is used to create an AIX JFS2 file system for an Oracle Database based on PP-Spreading:

> **Note:** The specified block size (`agblksize`) must be adjusted to `512` for the file system containing redo log files.

1. Allocate volumes from storage (typically in multiples of 8) and map to LPAR.
2. Discover disks by using the **`cfgmgr`** command.
3. Configure FC adapter and hdisk parameters.
4. Export `disklist="hdiskX hdiskY …"`.

5. Create the DATAVG volume group and specify all the corresponding hdisks. This PP size is the strip size that is used for striping the data over all hdisks in the VG and is selected as small as possible while still providing sufficient space in the VG for database data with less than about 60,000 PPs. Alternative larger PP sizes are 16 MB, 32 MB, and potentially 64 MB. However, at that size, a good likelihood exists to see hot spots that rotate over the disks:

   ```
   mkvg -S -s 8 -y DATAVG $disklist
   ```

6. Create the logical volume with maximum spread over all hdisks in DATAVG. The value for <size of LV in PP> is evenly devisable by the number of disks in the VG:

   ```
   mklv -y oradatalv -e x -a e  -t jfs2 -x <max # PP> DATAVG <size of LV in PP>
   $disklist
   ```

7. Create JFS2 file system with inline log on created LV:

   ```
   crfs -v jfs2 -d'oradatalv' -m /oradata -A'yes' -p'rw' -a agblksize='4096' -a
   logname='INLINE'
   ```

8. Mount the file system:

   ```
   mount /oradata
   ```

If PP Spreading is used to stripe data over all disks in a VG, consider and plan for the following points:

- All allocated volumes and hdisks in a VG are of the same size.
- PP size must be planned carefully as management of VG with more than 60k PP becomes slower.
- The LV is PP spread over some number of hdisks (typically, all hdisks in VG and a multiple of 8) from one VG.
- The LV space allocation can grow in multiples of one PP size, but grows in multiples of M PP sizes for balanced I/O distribution.
- A file system (FS) on top of a PP spread LV always grows with the same increments (1+ or M * PP) so no space is wasted. You must create the LV first and then, create the JFS2 file system on top of the PP-spread LV.
- If any of the M hdisks in the VG run out of available PP, AIX skips that hdisk in the round-robin allocation from eligible hdisks in the VG. If no eligible hdisk has an available PP, further attempts to grow the LV fail.
- The following options are available to resolve the `<out-of-space>` condition:
  - (Preferred option) Grow the underlying volumes dynamically by way of SAN methods and discover the new size dynamically by way of **`chvg -g <VG name>`**. Minimum size increase per volume is PP size.
  - Add another K hdisks (volumes) to the VG and then run **`reorgvg`** ***`<VG name>`*** to redistribute the allocated space evenly over all hdisks in the VG (`K >= 1`).

    This option requires significant SAN changes and more work by the AIX administrator. Adding volumes also affects IBM FlashCopy® configurations, for example.

The `reorgvg` operation is I/O intensive. No method is available to specify which PP is migrated. You can end up with some original data files still being backed only by the initial M hdisks, which results in some disks being more accessed in a VG than others.

9

Networking options for LPARs running Oracle DB

This chapter provides an overview of the available networking options for Oracle Database deployments in POWER9-based servers. It also discusses the arguments for and against the use of a specific technology in that context.

This chapter includes the following topics:

- 9.1, “Introduction” on page 120.
- 9.2, “Dedicated physical network adapter in logical partition” on page 120
- 9.3, “Shared Ethernet Adapter in Virtual I/O Server” on page 121.
- 9.4, “Multiple RAC clusters sharing servers with SEA-based interconnect” on page 122.

9.1 Introduction

Supported networking technologies for Oracle Database on AIX are Ethernet and InfiniBand. Several communication adapters with different speeds and feature sets are supported for IBM Power Systems servers. For more information about IBM Power Systems servers and supported communication adapters, IBM Documentation.

For more information about the latest supported network technologies for Oracle RAC, see this Oracle web page.

In this chapter, after the options overview, a sample configuration is presented that shows how Shared Ethernet Adapter functions can be used to deploy two independent RAC clusters onto the same two physical Power Systems servers.

9.2 Dedicated physical network adapter in logical partition

In contrast to many other virtualization solutions, IBM PowerVM provides the option to configure a mix of virtualized and physical communication adapters to a logical partition (LPAR).

For environments with stringent requirements on latency and or bandwidth or packet rates, dedicated physical network adapters can be the best choice. The use of dedicated physical communication adapters is more expensive than the use of virtualized communication adapters that provide sharing across LPARs.

In addition to cost, another limitation is that a physical server has only a relatively small number of adapter slots. This limitation limits how many LPARs with physical adapters can be configured on a specific server.

The use of dedicated network adapters can be the choice for the Oracle Real Application Cluster Interconnect in a large environment.

9.3 Shared Ethernet Adapter in Virtual I/O Server

Figure 9-1 shows the well-established Shared Ethernet Adapter (SEA) configuration, which also is referred to as *virtual Ethernet*. This technology is available with IBM PowerVM.

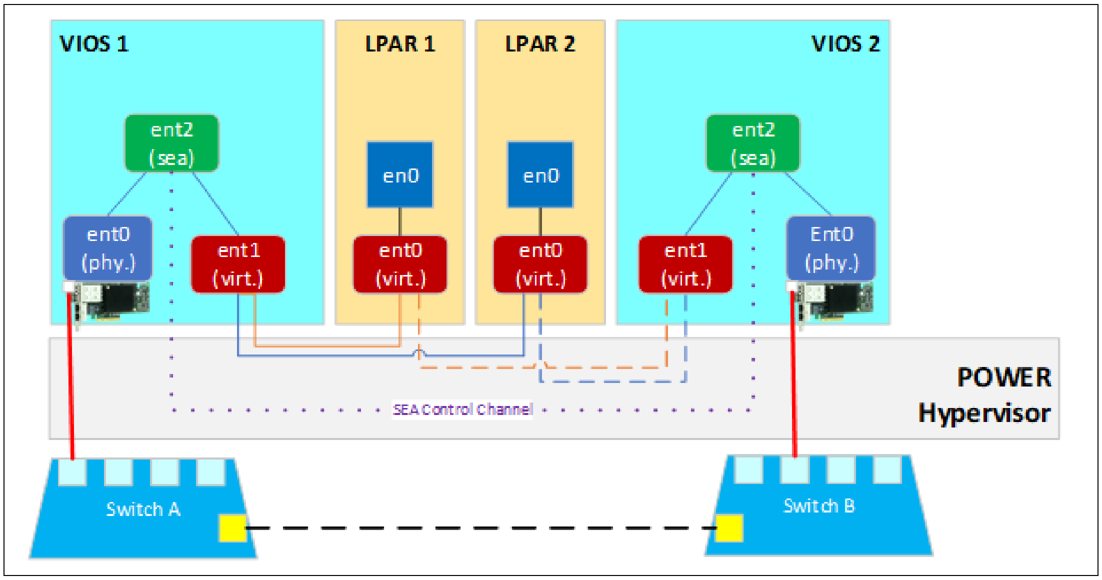


Figure 9-1 Shared Ethernet Adapter (SEA) configuration

SEA requires at minimum one (typically two) VIO Servers. This configuration allows the sharing of the physical Ethernet adapter between multiple client LPARs. With dual VIO Servers configuration and Shared Ethernet Adapter Failover, redundant access to the external network is provided.

In addition to sharing of physical network adapters between LPARs, SEA technology enables Live Partition Mobility (LPM). LPM enables the migration of an active LPAR between two separate physical Power servers without having to stop the applications or the operating system. LPM is supported for LPARs that are running single instance Oracle Databases or Oracle RAC.

For more information about supported configurations, see this Oracle web page.

LPM is fully supported for client access network and the RAC interconnect. Figure 9-1 also shows only a single network. For Oracle RAC, a minimum of two independent networks is required.

9.4 Multiple RAC clusters sharing servers with SEA-based interconnect

This section takes a closer look at how to use Shared Ethernet Adapters in the context of deploying two, two-node RAC clusters (four LPARs) onto two IBM Power Systems servers. To simplify the discussion, we concentrate on the required RAC interconnect for the two clusters.

Client access and backup and management networks also must be implemented. The client access and backup and management networks can be implemented with any of the available network technologies, with the most likely choice being SEA. Typically, it is recommended to separate RAC interconnect traffic onto separate physical adapters and network ports.

For an aggregation of clusters with high packet rates on the RAC interconnect, it is better to use dedicated physical adapters for the highest users as high packet rates drive significant CPU usage in the respective VIO Servers.

> **Note:** Single Root IO Virtualization (SR-IOV) is a new technology that is available with latest AIX and IBM Power Systems servers. However, at the time of this writing, this is not yet supported by Oracle.
>
> When Oracle support for SR-IOV based technology becomes available, it is highly recommended to use this new technology to share network adapters more efficiently between multiple client LPARs.

The configuration that shown in Figure 9-2 is the minimum recommended configuration for an Oracle RAC cluster, which also provides protection against networking single points of failure by using SEA failover.

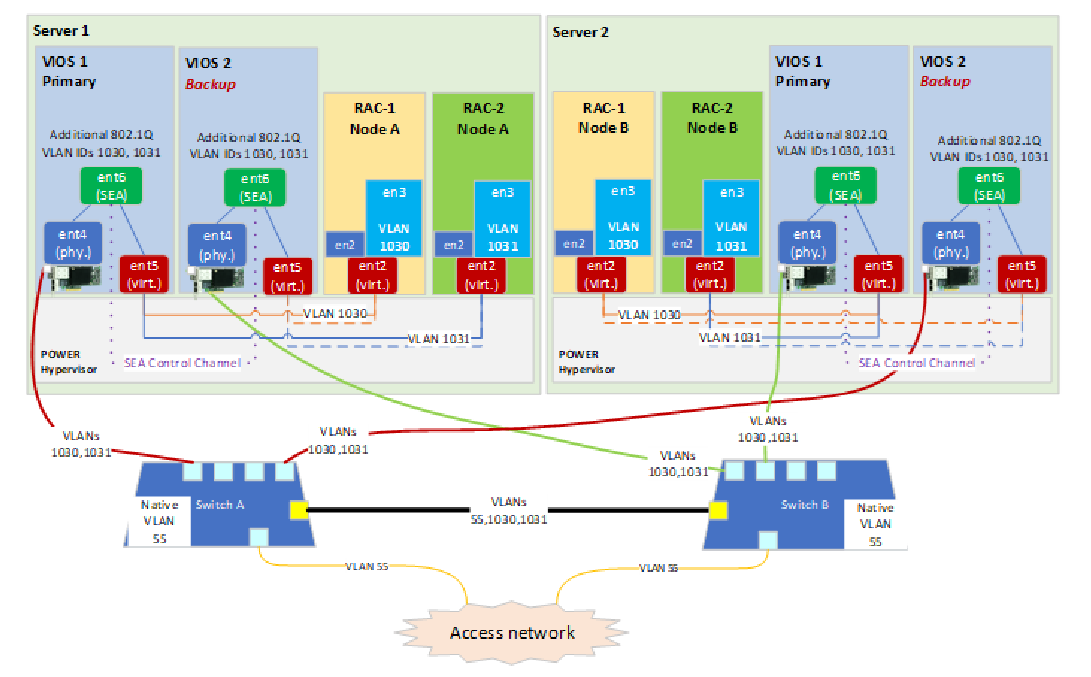


Figure 9-2 Oracle RAC cluster minimal configuration

If higher bandwidth is needed, physical Ethernet link aggregation over two or more physical network ports in the VIO Servers can be used for SEA backing adapters. Another option is to configure multiple SEA adapters in VIO Servers and two or more corresponding Virtual I/O Ethernet Adapters in the client LPAR.

Oracle High Availability IP (HAIP) can then aggregate those virtual network ports for the RAC interconnect providing availability and high bandwidth by distributing network traffic over all specified interfaces.

A

Additional material

This book refers to additional material that can be downloaded from GitHub as described in the following sections.

Locating the GitHub material

This section provides the GitHub repository for this publication. This repository is used for storing the configuration files and the sample code that was developed for this IBM Redbooks publication.

The code is provided on an as-is basis and is available at this web page.

Related publications

The publications that are listed in this section are considered particularly suitable for a more detailed discussion of the topics that are covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide more information about the topic in this document. Note that some publications that are referenced in this list might be available in softcopy only:

- *IBM Power Systems Private Cloud with Shared Utility Capacity: Featuring Power Enterprise Pools 2.0,* SG24-8478
- *IBM PowerVC Version 2.0 Introduction and Configuration*, SG24-8477
- *Red Hat OpenShift V4.X and IBM Cloud Pak on IBM Power Systems Volume 2*, SG24-8486

You can search for, view, download, or order these documents and other Redbooks, Redpapers, Web Docs, draft, and additional materials at the following website:

ibm.com/redbooks

Online resources

The following websites also are relevant as further information sources:

- Licensing Oracle Software in the Cloud Computing Environment:

  https://www.oracle.com/assets/cloud-licensing-070579.pdf
- On-premises: IBM Private Cloud with Dynamic Capacity:

  https://www-01.ibm.com/common/ssi/ShowDoc.wss?docURL=/common/ssi/rep_ca/1/897/ENUS120-041/index.html&lang=en&request_locale=en
- Off-premises: IBM Power Systems Virtual Server:

  https://www.ibm.com/cloud/power-virtual-server

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Redbooks

Oracle on IBM Power Systems

(0.2”spine)
0.17”<->0.473”
90<->249 pages

SG24-8485-00

ISBN 0738460125

Printed in U.S.A.