Redbooks
ibm.com/redbooks

IBM PowerVP

Introduction and Technical Overview

Liviu Rosca

Stefan Stefanov

Cloud

Power Systems

IBM

Redpaper

International Technical Support Organization

IBM PowerVP: Introduction and Technical Overview

August 2015

REDP-5112-01

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

Second Edition (August 2015)

This edition applies to Version 1, Release 1, Modification 3 of PowerVP Standard Edition (product number 5765-SLE).

This document was created or updated on August 14, 2015.

Contents

Notices

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

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

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

Trademarks

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

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

AIX®
Global Technology Services®
IBM®
Power Systems™
POWER7®
POWER7 Systems™
POWER8™
PowerHA®
PowerVM®
PowerVP™
Redbooks®
Redpaper™
Redbooks (logo) ®
System i®
WebSphere®

The following terms are trademarks of other companies:

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

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

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

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

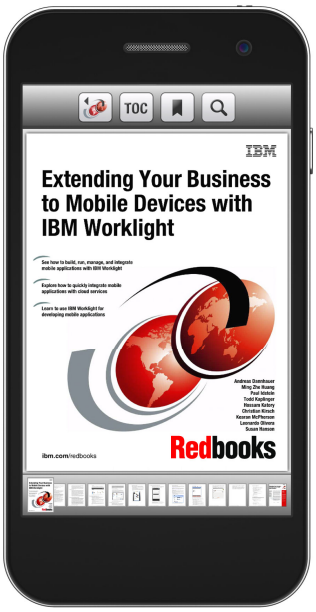
Other company, product, or service names may be trademarks or service marks of others.

Find and read thousands of IBM Redbooks publications

- Search, bookmark, save and organize favorites
- Get up-to-the-minute Redbooks news and announcements
- Link to the latest Redbooks blogs and videos

Get the latest version of the Redbooks Mobile App

iOS **Download Now** Android



Promote your business in an IBM Redbooks publication

Place a Sponsorship Promotion in an IBM® Redbooks® publication, featuring your business or solution with a link to your web site.

Qualified IBM Business Partners may place a full page promotion in the most popular Redbooks publications. Imagine the power of being seen by users who download millions of Redbooks publications each year!

ibm.com/Redbooks
About Redbooks → Business Partner Programs

THIS PAGE INTENTIONALLY LEFT BLANK

Preface

This IBM® Redpaper™ publication is a comprehensive guide that covers IBM Power Virtualization Performance (PowerVP™) for IBM Power Systems™ Version 1.1.3. The objective of this paper is to introduce the features and benefits of PowerVP:

- Real-time monitoring
- Intuitive and customizable GUI
- Background data collection
- Multiple dynamic views
- Recording and replay
- Data stores
- Thresholds and alarms

This publication is for professionals who are involved in performance and capacity planning, support, design, and deployment of IBM PowerVM® on IBM Power Systems. The intended audience includes the following roles:

- Clients
- Capacity planning managers
- Sales and marketing professionals
- IT Architects
- Technical support professionals
- IBM Business Partners independent software vendors

This paper explains the current set of IBM PowerVP features and documents the installation and configuration. It includes real-life examples of use of PowerVM.

It is intended to help you take advantage of the capabilities of IBM Power Systems and optimize the use of system resources.

Authors

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

Liviu Rosca is a Senior Product Support and IT Specialist with IBM Global Technology Services® in Romania. He has 14 years of experience in Power Systems, IBM AIX®, and IBM PowerHA®. His areas of expertise include Power Systems, AIX, PowerHA, networking, security, and telecommunications. He teaches AIX and PowerHA classes. He co-authored other IBM Redbooks® publications.

Stefan Stefanov is a Certified IT Specialist in IBM Global Technology Services at IBM Czech Republic. He has 12 years of broad experience in various UNIX and Linux operation systems, networking, virtualization, performance tuning, and disaster recovery solutions. He has extensive expertise in design, implementations, and support of IBM Power Systems technologies, with emphasize on PowerVM, PowerHA, and AIX.

The project that produced this publication was managed by:

Scott Vetter, PMP

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

Dave Bennin
Richard M. Conway
Ann Lund
IBM International Technical Support Organization, Poughkeepsie Center

Kent L. Bruinsma
IBM i Performance Tools and development, IBM Rochester

Jesse L. Sathre
Software Developer for Workload Estimator, IBM Rochester

Ashley Good
IBM i Sockets, SSL, and VPN Development, IBM Rochester

Marsinah Blazek
IBM i Serviceability Team, IBM Rochester

Pete Heyrman
PowerVM Hypervisor, IBM Rochester

Mathew Accapadie
AIX Performance, IBM Austin

Indra Suryatama
IBM i Specialist, IBM Indonesia

Craig Hunt
Team Computing, Sydney, Australia

Harihara Balakrishnan
Team Computing, IBM Singapore

Thanks to the authors of the previous edition of this paper.

Authors of the first edition, IBM PowerVP: Introduction and Technical Overview, published in September 2014, were:

Ramesh Palakodeti IBM US
Liviu Rosca IBM Romania

Now you can become a published author, too!

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

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

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  https://twitter.com/ibmredbooks

- Look for us on LinkedIn:

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

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

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

- Stay current on recent Redbooks publications with RSS Feeds:

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

1

Introduction

IBM Power Virtualization Performance (PowerVP) for Power Systems is a licensed program that provides a performance view into the PowerVM virtualized environment. PowerVP gathers resource utilization information using new PowerVM Hypervisor interfaces and displays which virtual workloads are using specific physical resources on an IBM Power System.

Clients who are using IBM Power Systems in Cloud computing with large and small enterprises with PowerVM features now can use PowerVP to understand resource distribution among the virtual workloads and plan for business demands.

This chapter includes the following sections:

- PowerVP overview
- PowerVP features and benefits
- PowerVP architecture

1.1 PowerVP overview

PowerVP is the first IBM product to graphically display the utilization of individual processor cores and their mapping to logical partitions on an IBM Power System in *real time*. The dashboard view of PowerVP gives a simple to understand snapshot of resource utilization that can be used by system administrators to understand resource usage by different applications that run on logical partitions.

PowerVP is helpful for further tuning of usual PowerVM virtualization features, especially in environments that comprise many IBM Power Systems and logical partitions.

PowerVP dashboard is intuitive and provides in-depth views of physical, virtual, and logical resource utilization. PowerVP represents a handy tool for troubleshooting performance issues, and regular system performance reviews. PowerVP can record and store performance data for later analysis. This feature is useful for assessment of performance levels during critical sustained activities such as batch processing or regular backups.

The product can be used along with existing operating system performance utilities to identify performance bottlenecks on IBM Power Systems. System administrators can use PowerVP to diagnose resource contention issues quickly, without running multiple, and complex OS-specific or Hardware Management Console (HMC) commands.

1.2 PowerVP features and benefits

PowerVP collects performance data *directly* from the Hypervisor, which offers the most accurate performance information about logical partitions that are running on IBM Power Systems. This performance information is then displayed on a real-time, continuous graphical dashboard. Data collection is available for historical review.

PowerVP has the following major features:

- Real-time monitor

  PowerVP can collect, update, and display the performance information as frequent as one second.

- Easy to use graphical user interface (GUI)

  PowerVP graphical monitor (dashboard) produces an easy-to-read display that shows an overall performance state of the IBM Power System.

- Customizable GUI

  PowerVP uses colorized heat technique to display the utilization metrics. It allows you to define the colors and the performance thresholds to modify the dashboard to match your monitoring requirements.

- Power platform support

  PowerVP supports all Power virtualized workloads such as AIX, VIOS, IBM i, and Linux.

- Performance data collection using the GUI

  PowerVP GUI can record performance data in real-time just like a DVR.

- DVR-like replay

  PowerVP can play back historical performance data. This feature enables you to replay performance data sequences to figure out performance bottlenecks. During the play back, DVR-like functions, fast forward, rewind, pause, and stop are available.

- Data repository

  PowerVP agents can collect and store performance data in a permanent data store. The data is stored in comma-separated values (CSV) format. This feature allows you to read and analyze the data using any tool that can read and interpret CSV files.

- Multiple dynamic views:
  - System level view

    PowerVP system level view depicts overall performance status of virtualized workload of the entire system.

  - Logical partition view

    PowerVP partition drill-down view portrays performance details of individual logical partitions. Along with the CPU and memory utilization, this view shows more details that include disk transfer rate, Ethernet adapter transfer rate, and a detailed view of the Load/Store Unit in load store unit cycles per instruction (LSU CPI).

  - Hardware topology view

    PowerVP illustrates Power System hardware topology along with the resource measurement views.

  PowerVP is designed to complement existing performance monitoring utilities. The operating system performance monitoring tools can still be used to understand resource utilization of the LPAR and applications running on it. PowerVP produces a visual that shows overall utilization of a Power System.

PowerVP provides the following benefits:

- A simplified performance monitoring and analysis of physical and virtual workloads with a graphical user interface.
- An easy way to understand, manage, and optimize the performance of a Power System or group of Power Systems in cloud computing.
- A means to view resource utilization of individual processor cores and their mapping to logical partitions that can be useful in effectively balancing system resources on a Power System.
- An approach to understand the affinity and optimize the performance of logical partitions including the utilization of internal buses.
- A graphical view into real-time workload that highlights possible contentions and overcommitted resources.
- A way to define user preferred performance thresholds, colors, and customize dashboard specific to a Power System.
- Understanding Memory Access Latency and Thread Locality.
- A simple way to validate the performance improvement of Dynamic Platform Optimizer (DPO).

1.3 PowerVP architecture

The architecture of PowerVP includes components that interact with the PowerVM Hypervisor, operating system, and a GUI. PowerVP has the following components:

- System level agent

  PowerVP requires a designated partition on a Power System to be used for the collection of system-wide resource utilization data across a frame or central processor complex. For the designated partition, select the **Allow performance information collection** check box as shown in Figure 1-1. The agent that is installed on this partition is named a *System level agent*.

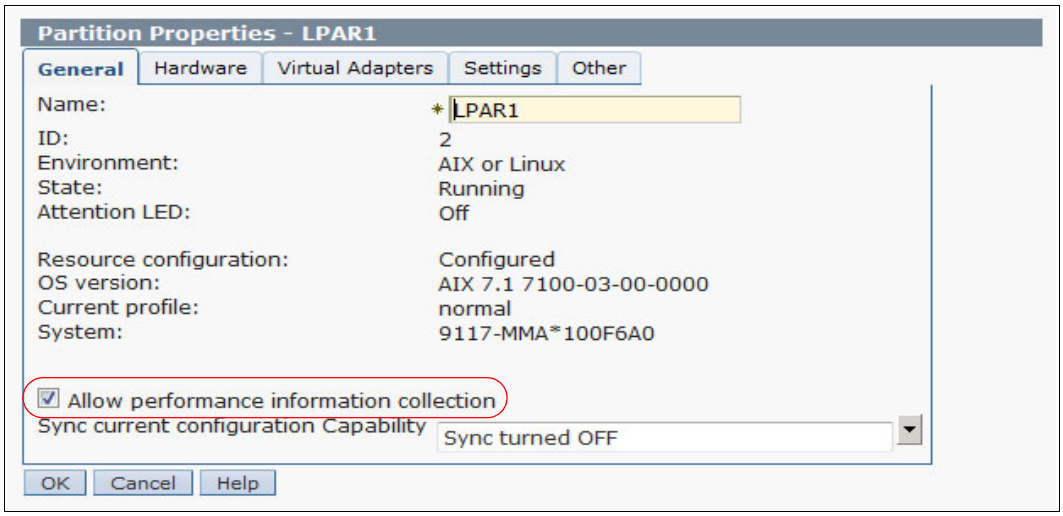


Figure 1-1 Enabling performance information collection

The System level agent uses a kernel extension module available on UNIX systems to make privileged *hcalls* (Hypervisor calls) to the Hypervisor. On IBM i, the application programming interface (API) by the way of Extended Profiling Facility (XPF) and System Licensed Internal Code (SLIC) interfaces with the Hypervisor. After the System level agent makes this connection with the Hypervisor, the agent requests performance counters from the Hypervisor as needed.

There can be multiple System level agents installed on a server. We recommend installing two System level agents for redundancy. Even if not required, we recommend installing the system level agents on the VIOS partitions.

Note: A System level agent is unaware of the existence of other System level agents. You cannot register a System level agent with another System level agent.

- Partition level agent

  To view partition-specific information, PowerVP requires an agent to be installed on the logical partition. This agent is named *Partition level agent.* The Partition level agent registers with the System level agent. Therefore, the System level agent must be up and running *before* installing the Partition level agent.

Note: The Partition level agent keeps trying to reconnect to the System level agent if the connection is lost. When you install the Partition level agent, you must register with the System level agent for the first time. We recommend having at least one System level agent running always.

The Partition level agents use partition APIs to retrieve Ethernet and Disk I/O utilization data. The Partition level agent gets access to Performance Monitoring Unit (PMU) to get hardware thread information. On AIX and the Virtual I/O Server (VIOS), the PMU is accessed by using Performance Monitor API (pmapi). On IBM i, PEX is used to get PMU data. On Linux, the Partition level agent uses **`perf stat`** commands to obtain performance information.

Note:

- The System level agent also acts as partition agent for the partition it is running on. There is no need to install a partition level agent on a partition in which a System level agent is already installed.
- There are performance monitoring tools that also access PMU, and PMU cannot always be shared by multiple tools. In this case, if another tool is accessing PMU, PowerVP might not be able to collect the information needed, and a partition drill-down might not work. Similarly, when PowerVP GUI is used for real-time monitoring or PowerVP is used for recording, the second monitoring tool that accesses PMU might not function.

Figure 1-2 displays a high-level architectural diagram of PowerVP. It displays the data flow between various components of PowerVP and PowerVM.

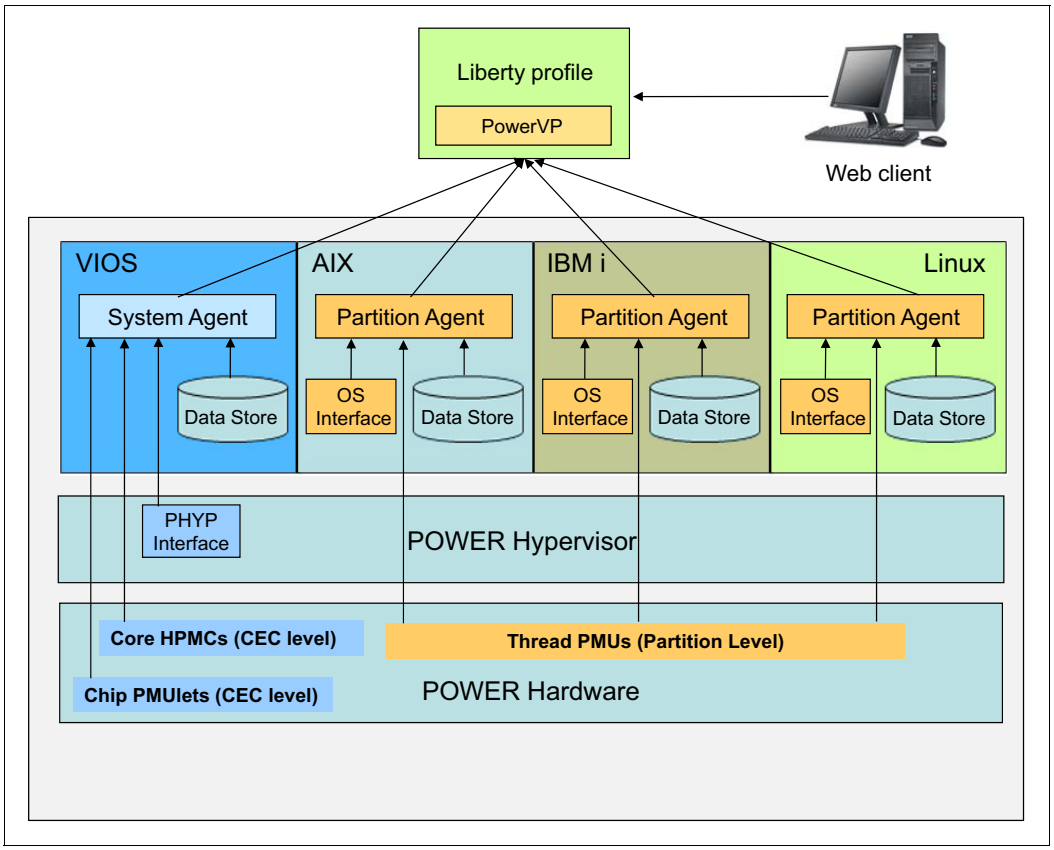


Figure 1-2 PowerVP architectural diagram

- Web-based GUI and application server

  Starting with Version 1.1.3, IBM PowerVP has a web-based GUI. It is packaged in the web application archive (WAR) format and it must be deployed onto an application server.

  By default, PowerVP GUI uses IBM WebSphere® Application Server Liberty Core. Liberty profile is a new server profile of IBM WebSphere Application Server V8.5. Liberty profile provides all features that are required to run the PowerVP GUI, it is lightweight, has a small footprint, and fast start time.

  PowerVP and a pre-configured Liberty profile are packaged into a compressed file. This provides for an easy and efficient distribution method and a simplified installation procedure.

  Because the new PowerVP GUI is web-based, it is now possible that a single instance of this GUI is accessed by multiple users using web browsers. This eliminates the need to install a console for each PowerVP user and avoids the potential overhead generated by additional performance data requests initiated from multiple consoles.

  PowerVP users can connect to the web GUI using web browsers. Users must be able to connect to the ports on which the application server is listening. Default port numbers are 9080 for HTTP traffic and 9443 for HTTPS traffic. Port numbers can be changed during the installation process.

  PowerVP connects to the System level agent. It must be able to connect to the port on which the System level agent is listening. Default port numbers are 13001 for SSL and 13000 for non-SSL. Port numbers can be changed during the installation process.

Note: If there are firewalls between the PowerVP GUI and the agent, ensure that the appropriate firewall rules are in place to allow connections on these ports.

- Data store

  Starting with Version 1.1.3, IBM PowerVP has a new feature named *data store*. By using this feature, PowerVP agents can be configured to collect performance data and save it to a file on the logical partition running the agent. The data is stored in CSV format. PowerVP can be configured to rotate and archive the files automatically.

  When you want to play back historic performance data from the data store, you specify a start date and time in the PowerVP GUI. The GUI instructs the agent to locate the corresponding data in the data store. The agent retrieves the data and sends it back to the GUI. The data store feature eliminates the need to store performance data on the system that runs the GUI.

  Note: Recording of performance data with PowerVP console GUI is still supported. In this case, the recording data is on the application server where the PowerVP GUI is running.

2

Planning and installation

This chapter provides details about planning and installation of IBM Power Virtualization Performance (PowerVP). It describes the architectural components of PowerVP and contains detailed step-by-step procedures to install PowerVP software components on various operating systems.

This chapter includes the following sections:

- Planning for PowerVP
- PowerVP installation
- Starting and stopping PowerVP
- Upgrading PowerVP agents
- PowerVP SSL configuration

2.1 Planning for PowerVP

At the time of writing this publication, IBM PowerVP comes in a single edition: PowerVP Standard Edition.

PowerVP is included in the PowerVM Enterprise Edition. Current and new PowerVM Enterprise Edition clients are entitled for PowerVP. PowerVM Standard Edition clients can purchase PowerVP Standard Edition.

PowerVP has the following prerequisites:

- Hardware

  PowerVP runs on IBM POWER7® processor-based and IBM POWER8™ processor-based systems.

  Note: PowerVP Version 1.1.2 is the minimum version that is supported on POWER8 systems.

- Firmware

  On IBM POWER7 Systems™, the firmware level of 770 and 780 or later is required. The firmware levels before 770 lack the new instrumentation required for PowerVP. Table 2-1 shows the supported Power Systems/Flex System Power compute nodes and firmware combinations at the time of writing.

Table 2-1 List of supported Power Systems and Flex System Power compute nodes

Server family and firmware	List of supported models
Power System models with 770 firmware support	► 710-E1D, 720-E4D, 730-E2D, 740-E6D (also includes Linux D models) ► 750-E8D, 760-RMD ► 770-MMC, 780-MHC ► 71R-L1S, 71R-L1C, 71R-L1D, 71R-L1T, 7R2-L2C, 7R2-L2S, 7R2-L2D, 7R2-L2T
Flex System Power compute nodes	p260-22X, p260-23X, p460-42X, p460-43X, p270-24X, p470-44X, p24L-7FL
POWER7 System models with 780 firmware support	770-MMB, 780-MHB, 795-FHB, 770-MMD, 780-MHD
POWER8 systems	All systems and all firmware levels

- Operating systems

  PowerVP agent runs on all PowerVM supported operating systems. PowerVP can be installed on AIX, VIOS, IBM i, and Linux. Following are the minimum releases levels supported:

  – AIX operating system releases 6.1 and 7.1

  – All VIOS versions are supported

  – IBM i operating system releases 7.1 (TR6 with PTF S152700, MF58478, MF58480) and 7.2 (with PTF S152700, MF58479, MF58481)

  Note: All IBM i program temporary fixes (PTFs), prerequisites, and corequisites must also be applied before installing PowerVP.

- RedHat Enterprise Linux (RHEL) 6.4 and later
- SUSE Linux Enterprise Server 11 SP3 or later

- Application server and Java JRE

  PowerVP relies on features provided by an application server. By default, PowerVP uses IBM WebSphere Application Server Liberty Core. The Liberty profile requires a Java Runtime Environment (JRE) to run in. PowerVP version 1.1.3 requires Java JRE 1.7.0 version.

  Note: Java JRE is not included with PowerVP.

- HMC

  There are no HMC-specific requirements for PowerVP, but for a logical partition that is running a System level agent, the **Allow Performance Information Collection** check box must be selected as shown in Figure 2-1. Figure 2-1 shows the new version of the HMC GUI.

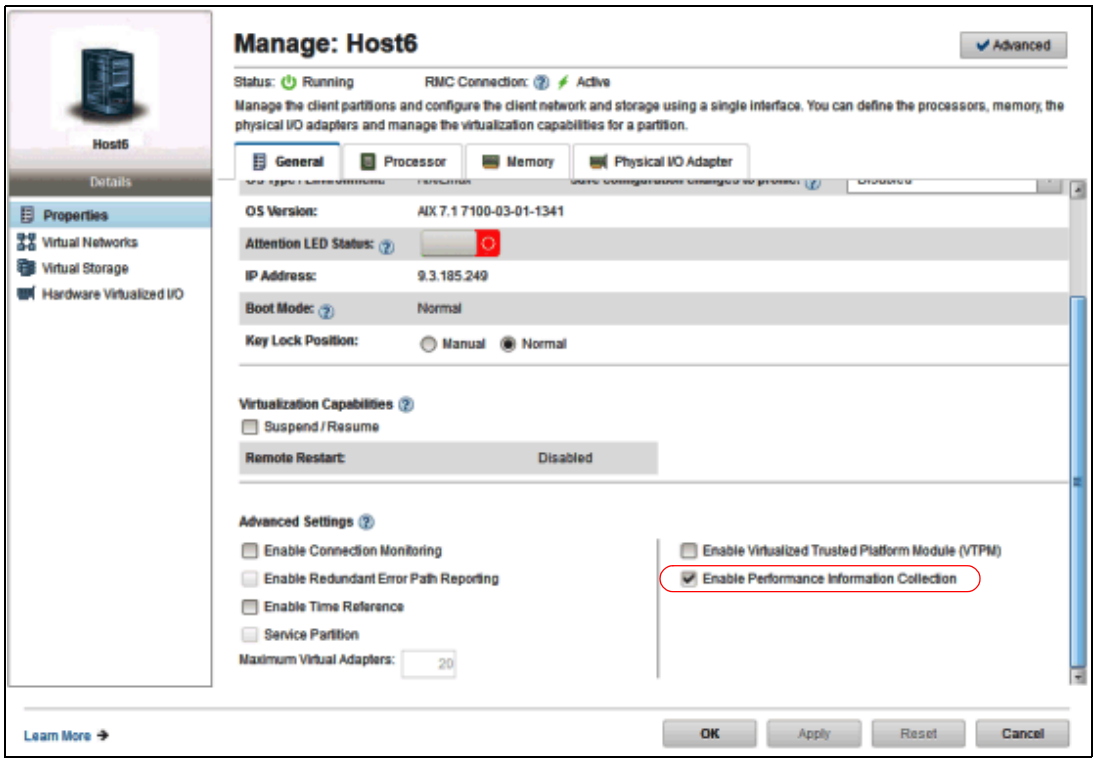


Figure 2-1 Configuring an LPAR to allow collection of performance information

- Network ports

  PowerVP uses the TCP protocol, and by default uses port 13000 for non-SSL and 13001 for SSL traffic. The port numbers can be customized to fit the security needs of the customer. The requirement is for the System level agent and the Partition level agent to use the same port numbers.

  If PowerVP agents are behind a firewall, appropriate firewall rules must be in place for the Partition level agent to be able to communicate with the System level agent and the GUI to be able to communicate with both the Partition level agent and the System level agent.

PowerVP GUI is accessible to users via web browsers. By default, the port numbers used are 9080 for HTTP traffic and 9443 for HTTPS traffic. You can change the port numbers to comply with the requirements of your environment.

- File system requirements

  PowerVP has a small footprint and is installed in `/opt/ibm/powervp` on UNIX systems. On IBM i, it is installed in the QSLE library. On UNIX, configuration files are installed in `/etc/opt/ibm/powervp`, and on IBM i, the configuration files are installed in `/QIBM/UserData/PowerVP`.

  Files used by the permanent data store are located by default in `/opt/ibm/powervp/logs` on UNIX and `/QIBM/UserData/PowerVP`/logs on IBM i.

  You must ensure that the file systems are not full. The amount of disk space required depends on how you use the performance data collection feature of PowerVP.

2.1.1 PowerVP security considerations

The PowerVP GUI connects to the System level agent by using a valid user name and password on the partition. When accessing partition information, or when using the Monitor LPAR option, the GUI is connecting to the Partition level agent to access the data.

There are two options to authenticate to the Partition level agent:

- Using the System level agent to authenticate the connection to the partition. This is the default option. After the client GUI is authenticated by the System level agent, the user is able to drill down into the partition. In this method, there is no need to provide authentication for signing on to the Partition level agent.
- The second method is to provide valid user and password details for the GUI client to connect to the Partition level agent.

  PowerVP provides both secure and non-secure options for connection between the agents and the GUI. Even though user and password information is not transmitted in plain text in non-secure mode, the rest of the data is transmitted in plain text. The secure option uses SSL protocol for communication between PowerVP components. To use this option, each PowerVP agent needs an SSL certificate.

2.2 PowerVP installation

PowerVP is packaged to include the following installable software components:

Java based GUI	This component is included only in PowerVP versions 1.1.0 and 1.1.2. It can run on Windows, AIX, and Linux.
Agents	The agents are packaged in the format that is specific to the platform on which they run.
Web-based GUI	This component was introduced with PowerVP version 1.1.3 and replaces the Java based GUI.

The PowerVP installer has a GUI with an easy to follow menu-driven dialog. The PowerVP installer and the GUI client can run on AIX, Windows, or Linux systems.

To install PowerVP, mount the installation media and change the directory according to the appropriate operating system. The two executable files in the directory are PowerVP and PowerVP_IBMi_Agent. The PowerVP executable file must be used to run the installation. Run the installation from the media itself because there are other directories on the installation media that are required by the installer (for example, the license directory).

2.2.1 PowerVP installation overview

When PowerVP installer is run, the packages specific to IBM i logical partitions are extracted locally and the agent installation is done automatically using Restore Licensed Program (RSTLICPGM) with remote commands using Java toolbox. If the installation fails, you must upload the packages to the IBM i logical partitions in binary mode and perform the installation manually.

As opposed to IBM i, the agents for AIX, Virtual I/O Server, and Linux systems are not installed automatically. Instead, the PowerVP packages required for each platform are extracted locally. You must upload these packages to the logical partitions in binary mode and perform the installation manually.

Section 2.2.4, “Installing PowerVP agent manually on IBM i” on page 30 describes how to perform manual installation on IBM i logical partitions.

Section 2.2.5, “Installing PowerVP agent on AIX and the VIOS” on page 31 describes how to perform manual installation on AIX/VIOS logical partitions.

Section 2.2.6, “Installing PowerVP agent on Linux” on page 34 describes how to perform manual installation on Linux partitions.

2.2.2 PowerVP installer on Microsoft Windows systems

This section shows how to use PowerVP installer on a Microsoft Windows system. PowerVP installer can be run from AIX or Linux as shown in 2.2.3, “Installing PowerVP GUI on AIX or Linux” on page 27.

When PowerVP.exe is run, after language selection and acceptance of license agreement, you are prompted to choose the type of installation to run.

There are three installation options as shown in Figure 2-2 on page 14:

- Typical

  This option installs PowerVP client GUI and the agents.

- PowerVP Client GUI

  This option installs only the client GUI, which is useful for users who have PowerVP agents already installed.

- PowerVP Server Agents

  This option installs the PowerVP agent code on the logical partitions. It is useful for users who have the PowerVP client GUI installed and are adding new LPARs for PowerVP monitoring.

Figure 2-2 PowerVP installation options

For an initial installation, select the **Typical** option to install and click **Next**. The next window prompts you for the destination directory for PowerVP installation as shown in Figure 2-3.

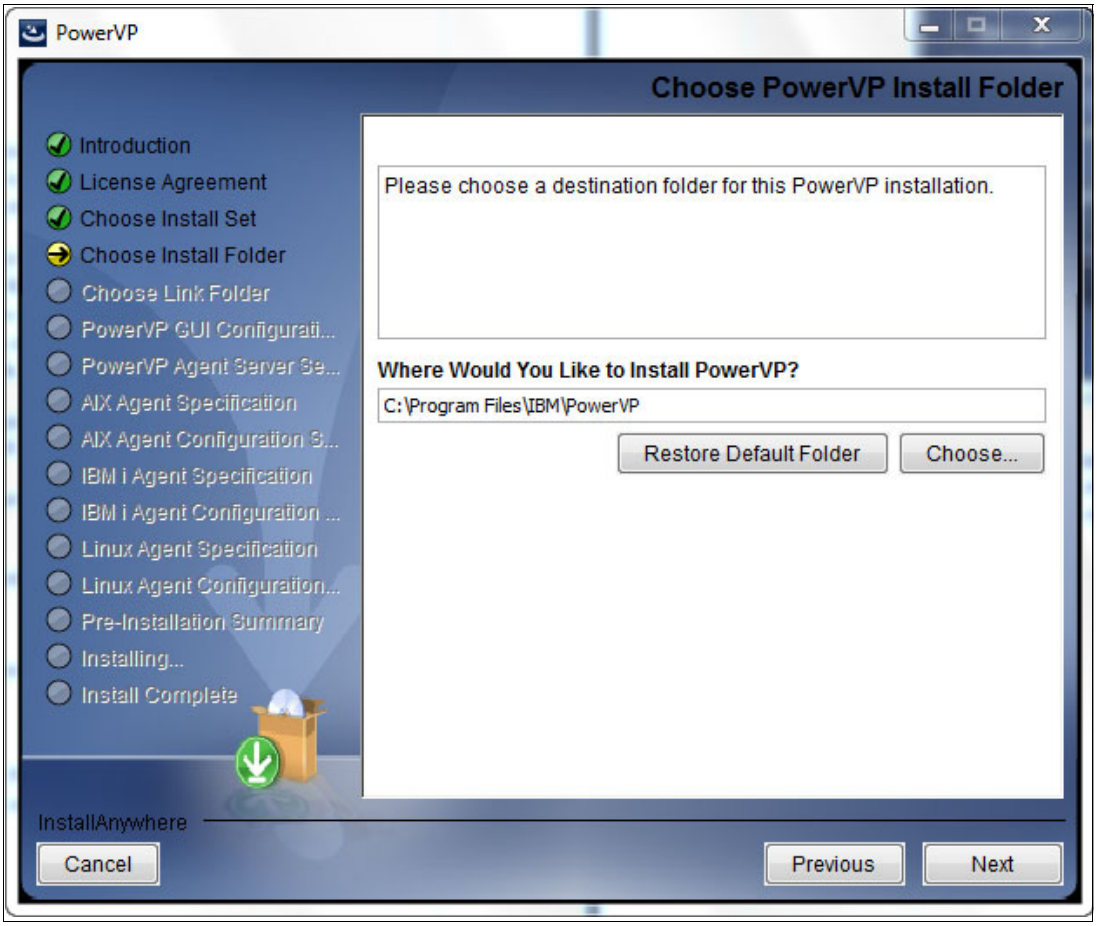


Figure 2-3 Selecting PowerVP installation directory on Microsoft Windows systems

The next window allows you to install and configure the Liberty profile. It prompts you to install IBM WebSphere Application Server Liberty Core and to configure the ports used. By default, the Liberty profile uses port 9080 for non-SSL and 9443 for SSL traffic as shown in Figure 2-4. The port numbers can be changed to fit the requirements of your environment.

Note: If you want to deploy the PowerVP GUI on an existing application server, you do not need to install the Liberty profile. In this case, the installer copies the WAR file and the `server.xml` file to the install directory.

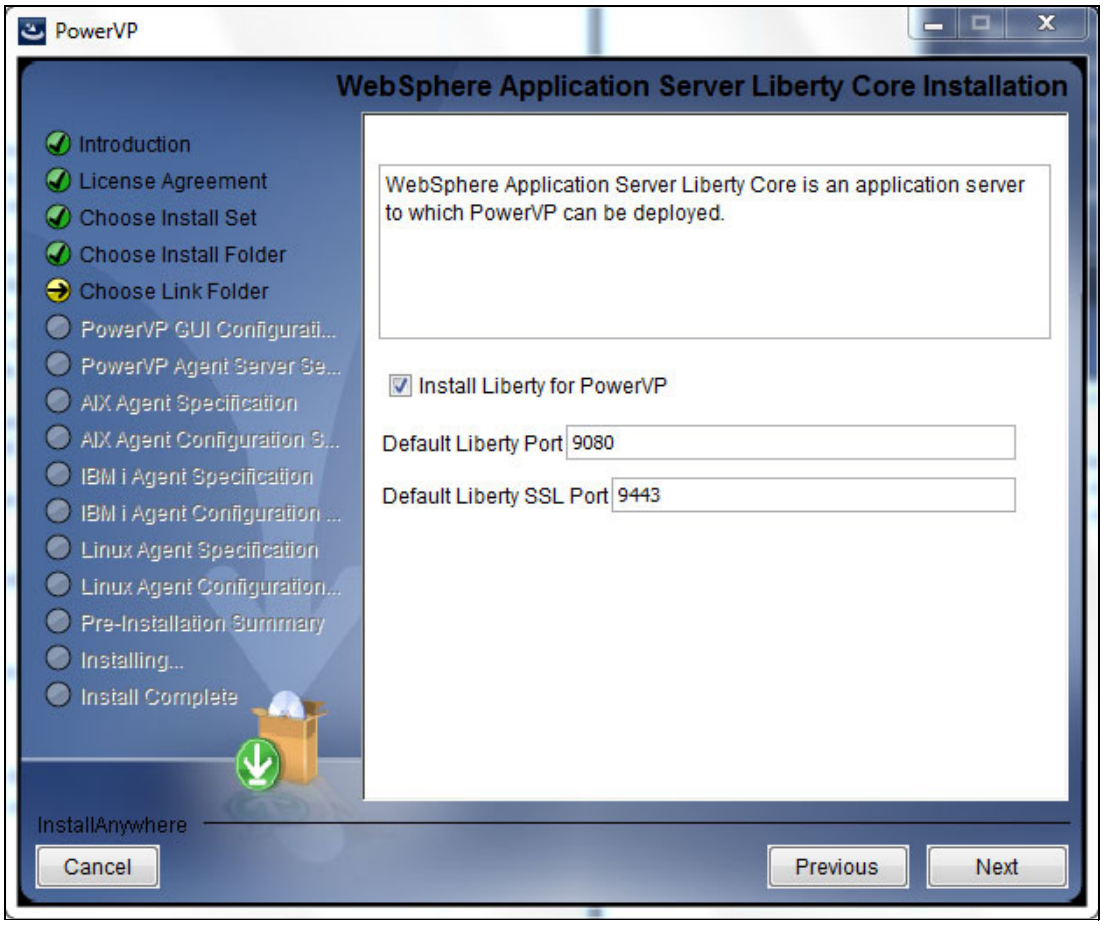


Figure 2-4 Installing Liberty profile and configuring Liberty profile ports

The next window provides you a summary of Liberty profile settings as shown in Figure 2-5 and allows you to review the settings before continuing the installation.

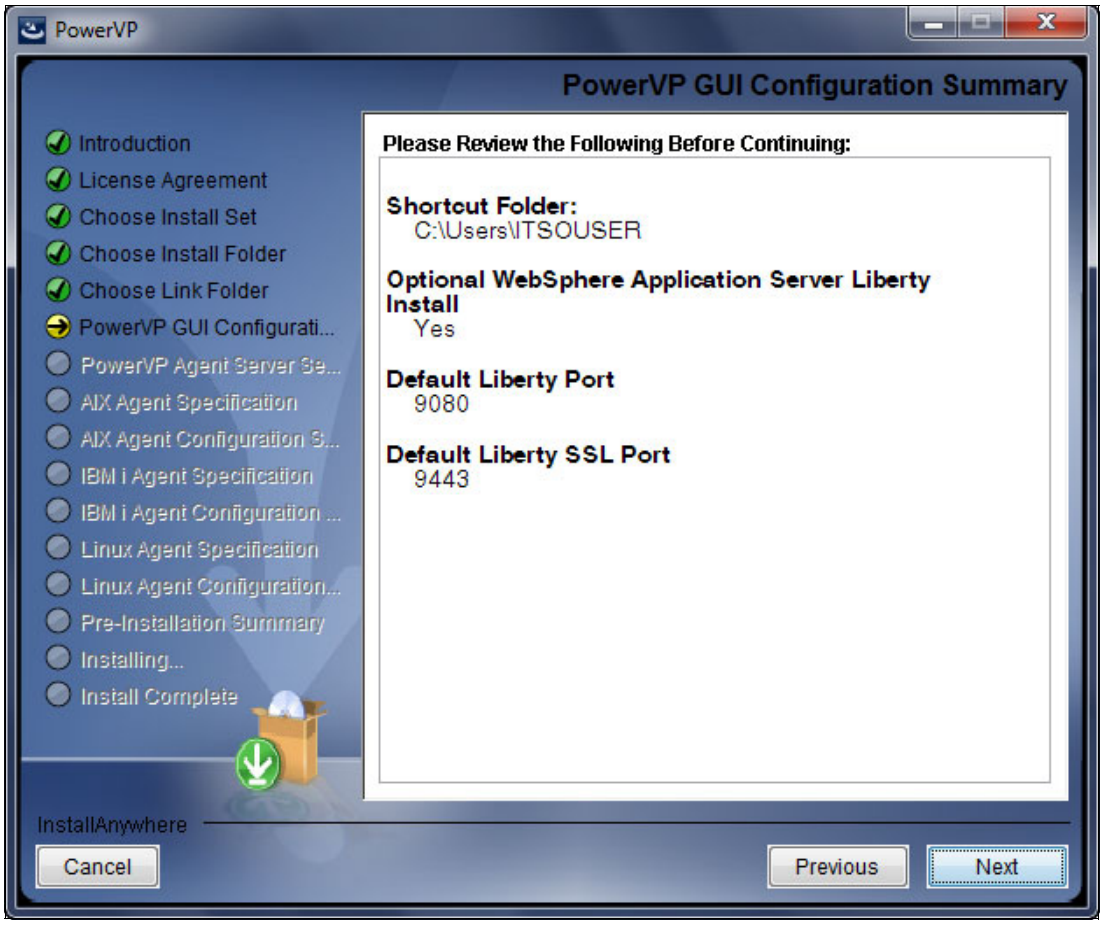


Figure 2-5 Reviewing Liberty profile settings

Following the summary page, the next window of the installer allows you to choose the types of systems for which you want to perform the installation of server agents. You can select to install server agents for **AIX/VIOS**, **IBM i**, and **Linux** systems as shown in Figure 2-6.

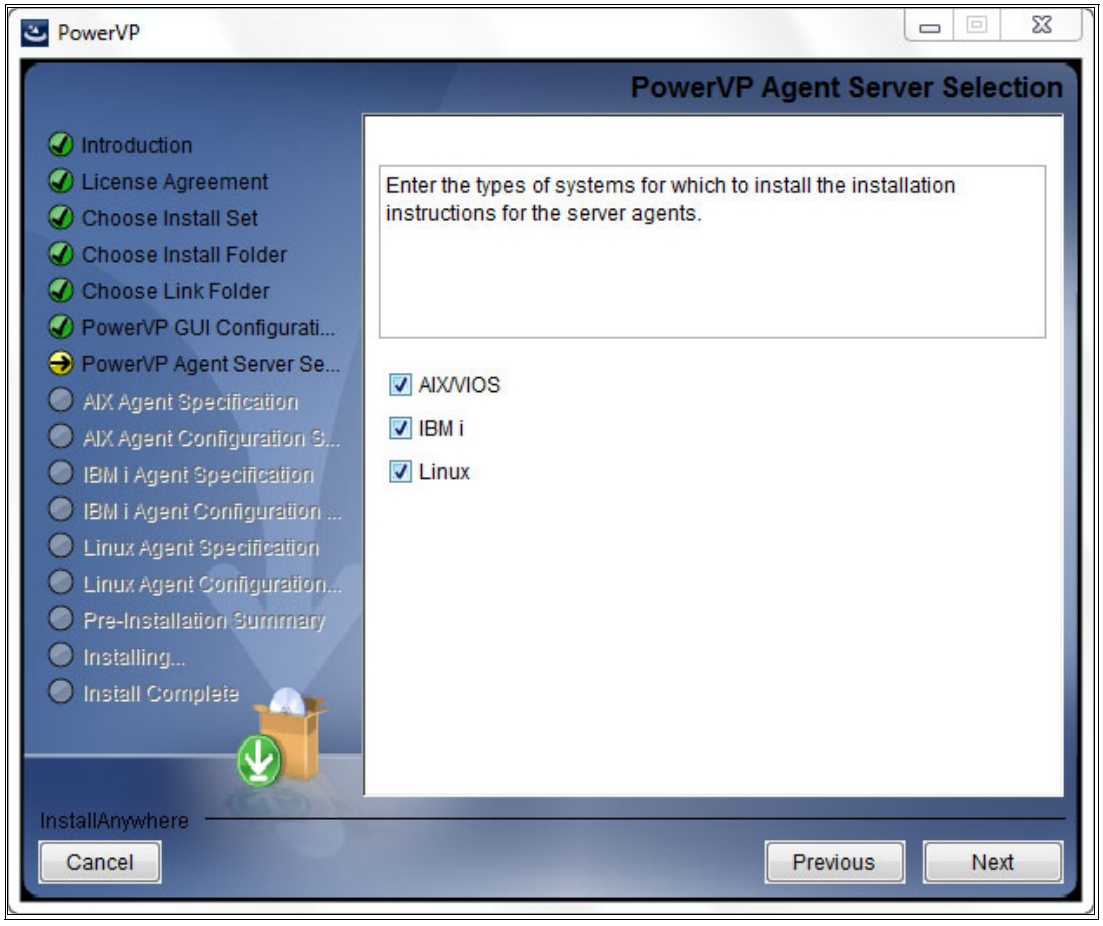


Figure 2-6 Selecting the types of systems for which you install server agents

The next window allows you to specify the configuration for AIX/VIOS agents as shown in Figure 2-7. You must specify the host name or IP address of the System level agent.

If you do not use Secure Sockets Layer (SSL) and you do not want to change the default port number, leave the value of 13000 in the field **Default Agent Port.** If you want to change the port number you have to specify the new value. The **Use non-SSL** check box is selected by default. If you do not want to use the non-SSL option you must clear the **Use non-SSL** check box**.**

If you want to use SSL you must select the **Use SSL** check box**.** By default, this option is cleared and needs to be explicitly selected. If you do not want to change the default port number, leave the value of 13001 in the field **Default SSL Agent Port**. If you want to change the port number you have to specify the new value.

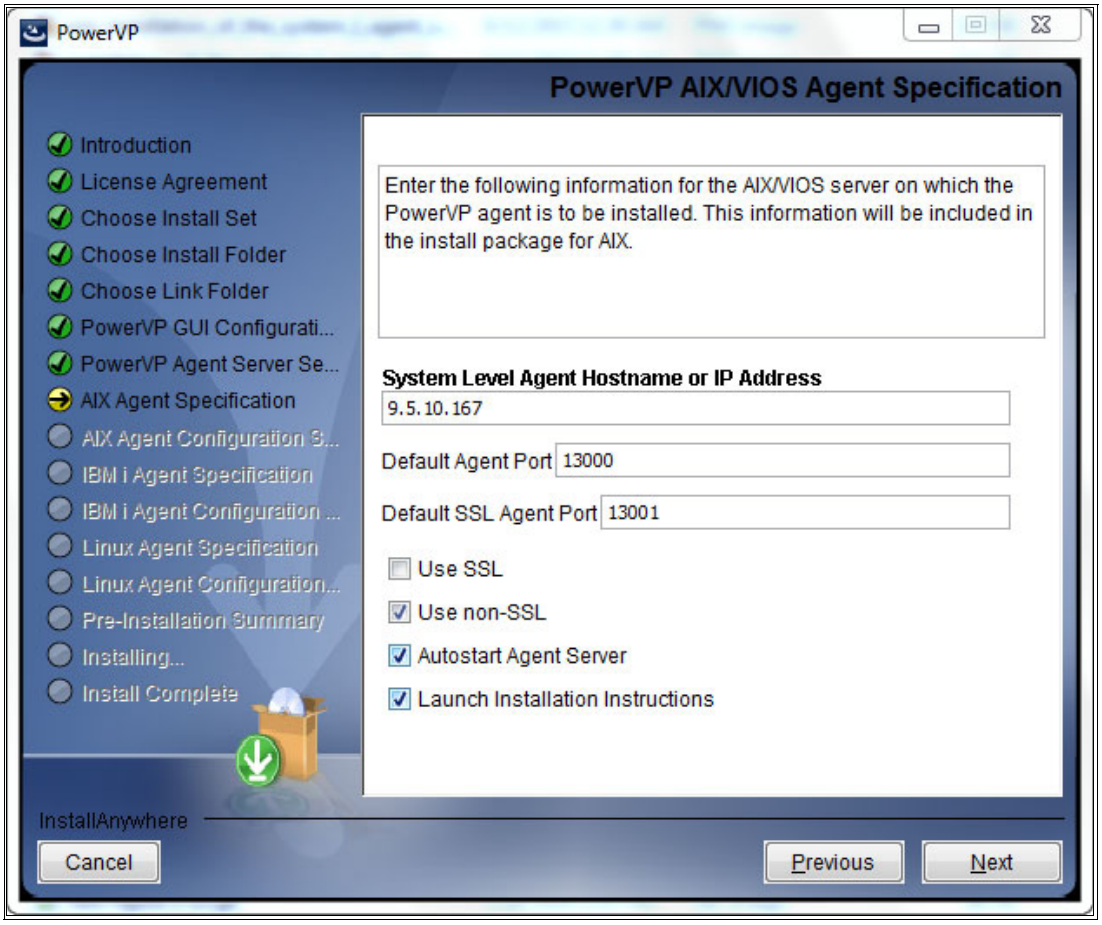


Figure 2-7 Configuring AIX/VIOS agents

The next window allows you to specify the IBM i systems on which you want to install agents as shown in Figure 2-8.

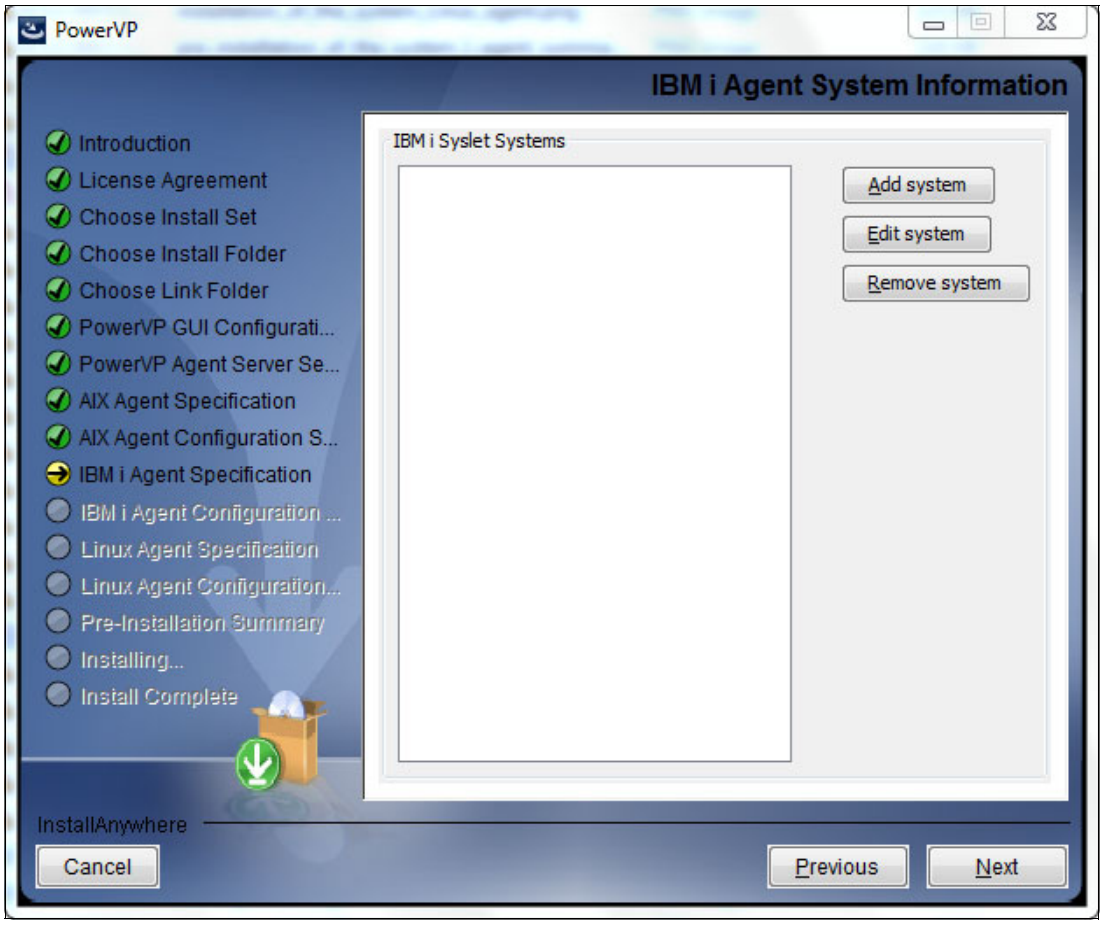


Figure 2-8 Specifying IBM i systems on which to install agents

When you click **Add system**, a separate window is displayed as shown in Figure 2-9 on page 21. This new window allows you to specify the information required to install the agent. In the first field, enter the IBM i LPAR host name or IP address. In the second field, enter the host name or IP address of the System level agent.

You must provide a valid combination of an IBM i user and password to run remote installation.

Note: The user designated for remote installation must have authority to the RSTLICPGM command, or have *SECADM or *ALLOBJ authority.

You can specify other system options. For example, you must select the **Autostart Server Agent** check box if you want the PowerVP agent to auto start as shown in Figure 2-9 on page 21.

If you do not use SSL and you do not want to change the default port number, leave the value of 13000 in the field **Default Agent Port.** If you want to change the port number you have to specify the new value. The **Use non-SSL** check box is selected by default. If you do not want to use the non-SSL option you must clear the **Use non-SSL** check box**.**

If you want to use SSL you have to select the **Use SSL** check box**.** By default, this option is cleared and needs to be explicitly selected. If you do not want to change the default port number, leave the value of 13001 in the field **Default SSL Agent Port**. If you want to change the port number you have to specify the new value.

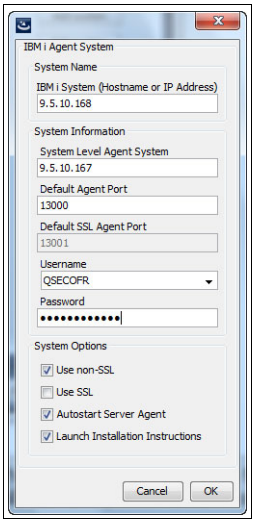


Figure 2-9 Specifying information required to install IBM i agent

Click **OK** to complete the configuration of the IBM i system. The IBM i system is added to the list of systems on which agent installation is done automatically by the PowerVP installer as shown in Figure 2-10. The installer allows you to add or delete other IBM i systems from the list. In case you want to review or change the configuration of an individual system, you must select the system from the list and click **Edit**.

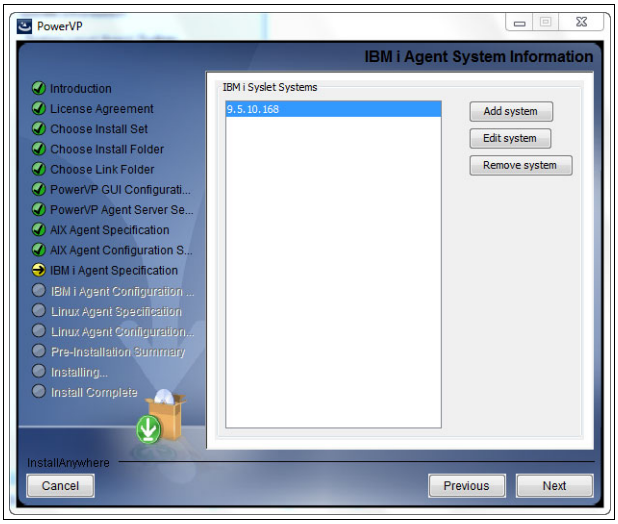


Figure 2-10 List of IBM i systems on which agent installation is done automatically

When all IBM i systems are added to the list, you must click **Next.** PowerVP installer displays a pre-installation summary for IBM i systems. The output is similar to what is shown in Figure 2-11.

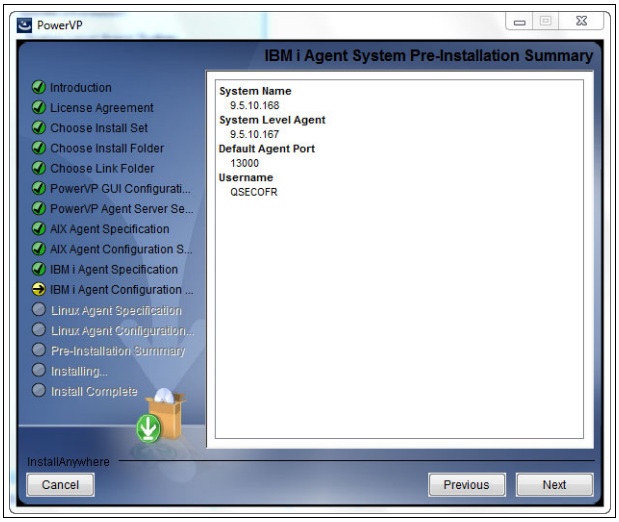


Figure 2-11 Pre-installation summary for IBM i systems

Next, the PowerVP installer displays a window that allows you to specify the configuration for Linux systems, as shown in Figure 2-12.

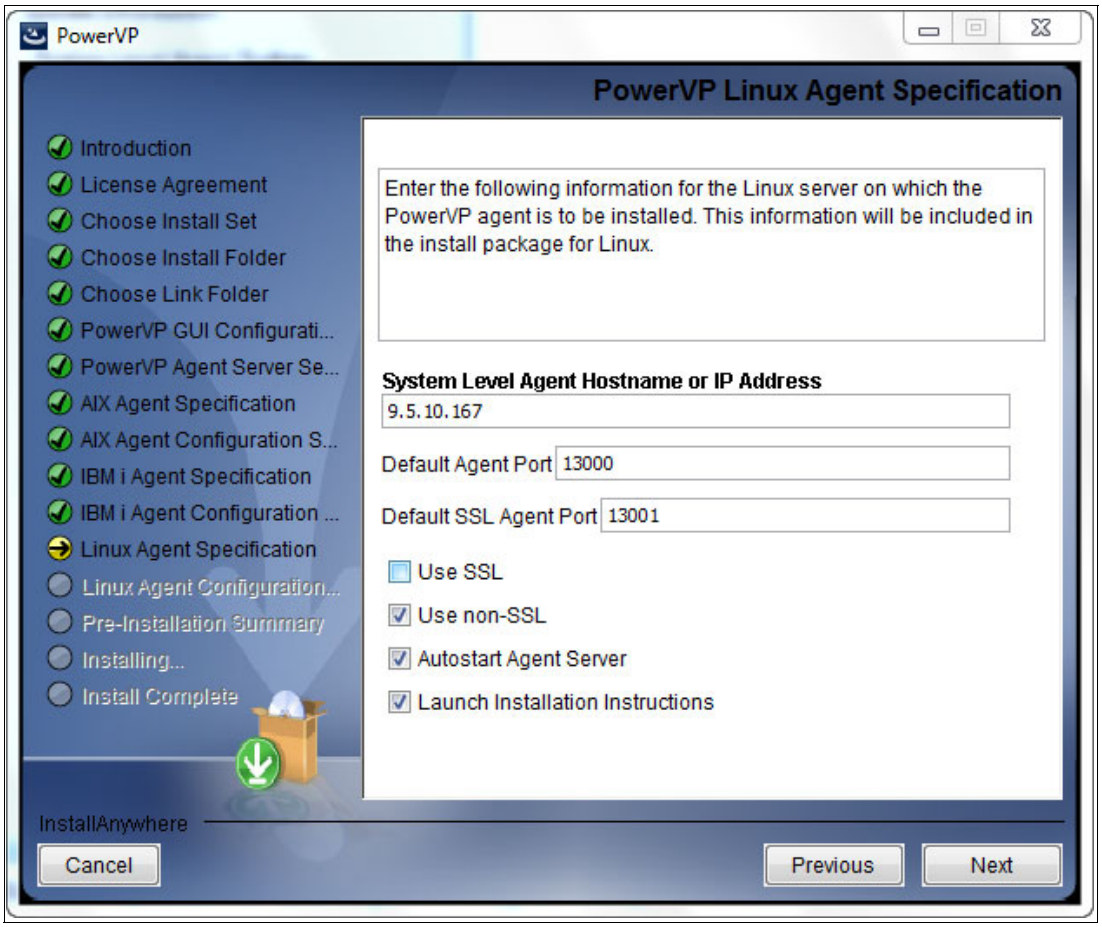


Figure 2-12 Specifying information required to install the Linux agent

You must specify the host name or IP address of the System level agent.

If you do not use SSL and you do not want to change the default port number, leave the value of 13000 in the field **Default Agent Port.** If you want to change the port number you have to specify the new value. The **Use non-SSL** check box is selected by default. If you do not want to use the non-SSL option you must clear the **Use non-SSL** check box**.**

If you want to use SSL you have to select the **Use SSL** check box**.** By default, this option is cleared and needs to be explicitly selected. If you do not want to change the default port number, leave the value of 13001 in the field **Default SSL Agent Port.** If you want to change the port number you have to specify the new value.

When the configuration of the Linux system is completed, the installer displays a pre-installation summary. The output is similar to what is shown in Figure 2-13.

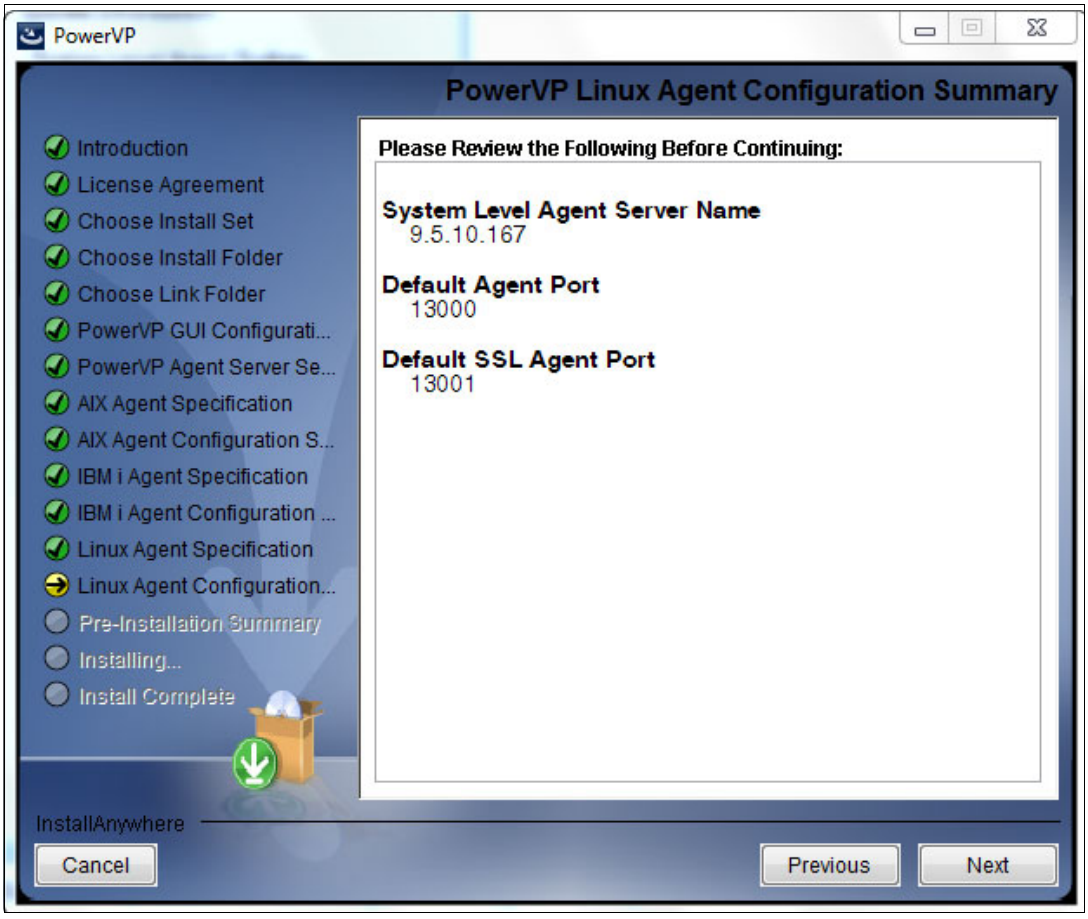


Figure 2-13 Pre-installation summary of Linux agents

When the information required for all types of systems is completed, the installer displays a pre-installation summary of PowerVP as shown in Figure 2-14.

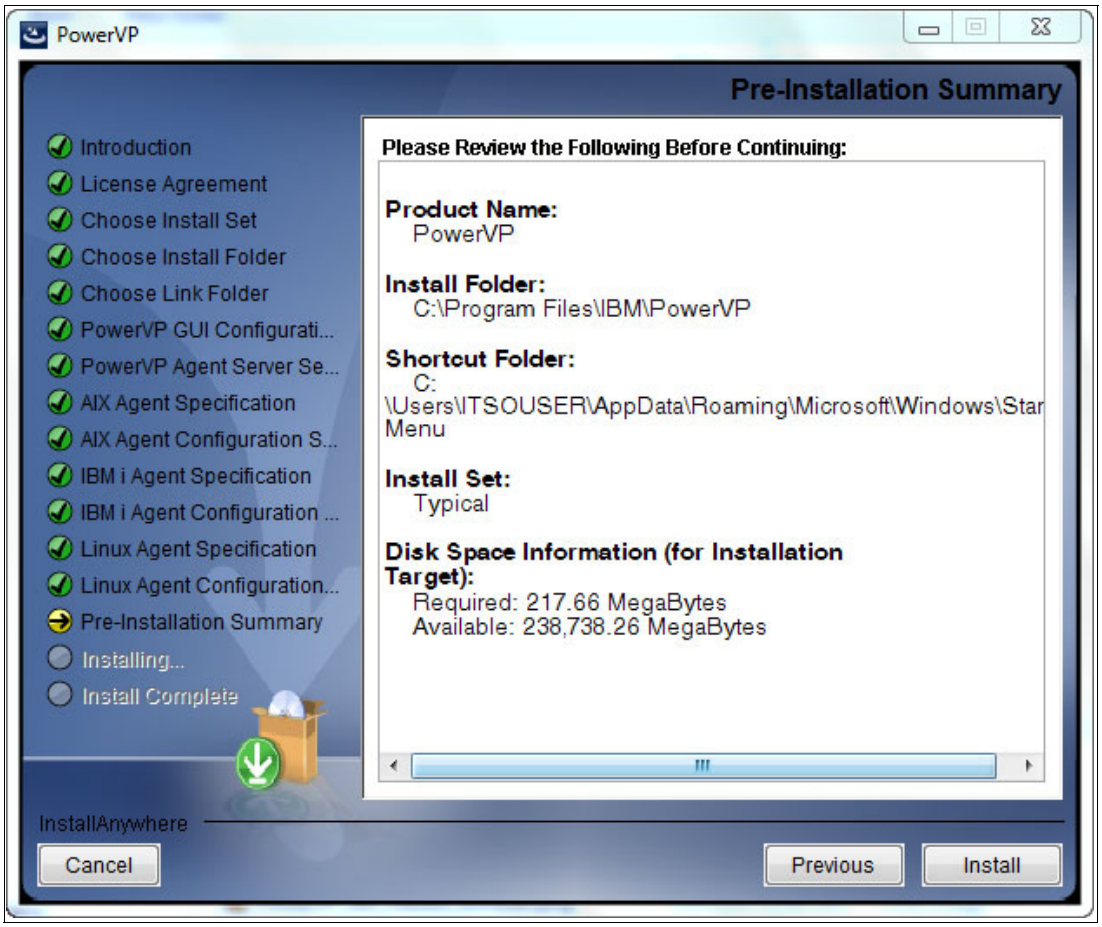


Figure 2-14 PowerVP pre-installation summary

Next, the installer starts the installation process. When the installation is complete, PowerVP installer displays a window similar to what is shown in Figure 2-15.

PowerVP installer saves the data that is related to the installation process in a log file in `C:\Program Files\IBM\PowerVP\PowerVP_Installation\Logs`. We recommend reviewing the data in this file, which includes the following information:

- Date and time of the installation
- Java version
- Installation folder
- Ports used by Liberty profile
- IP addresses or host names and ports used by agents for all platforms and systems
- List of files created

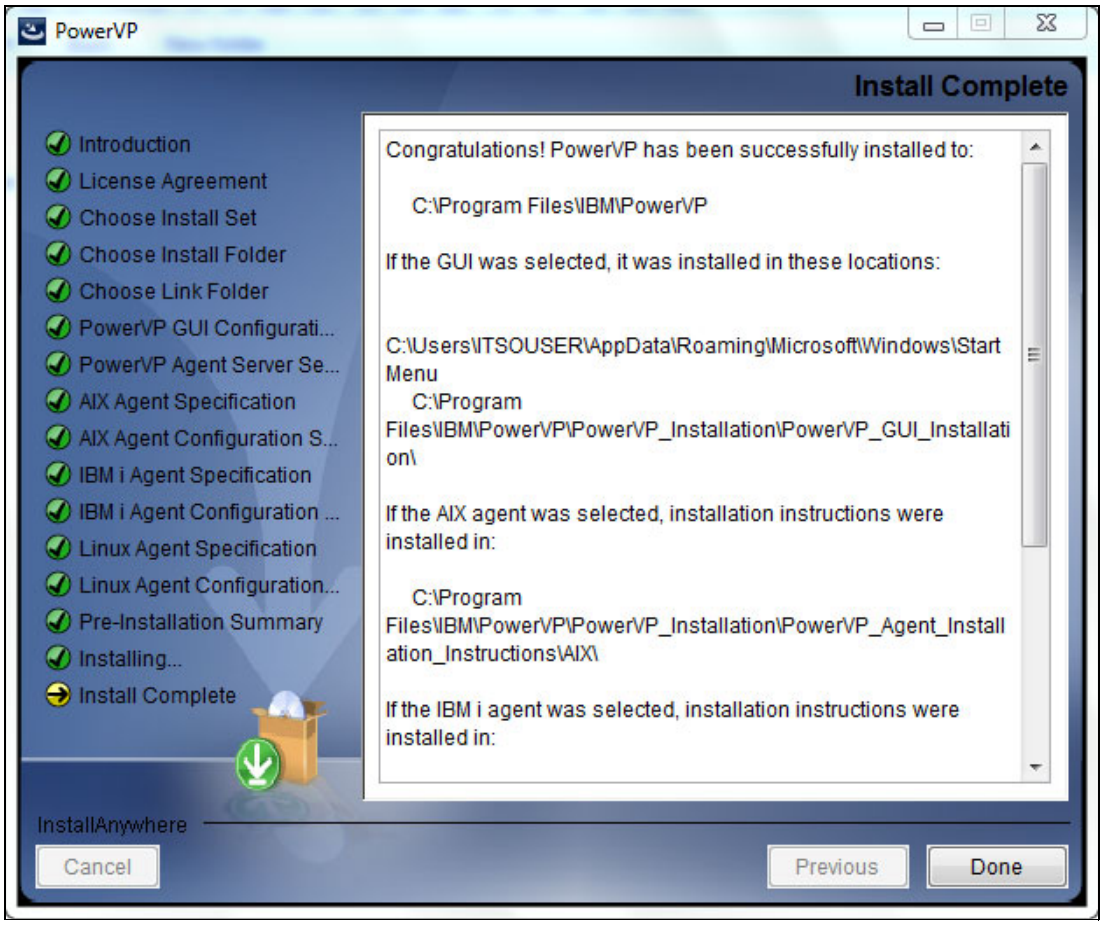


Figure 2-15 PowerVP installation summary

For each type of platform that is specified during the configuration phase, the installer creates a directory that contains binary files specific to each platform along with a `README.html` file. The `README.html` file contains installation instructions, configuration instructions, start instructions, and uninstall instructions that are specific to each platform.

If you selected the check box **Launch Installation Instructions** during the PowerVP configuration phase for an individual platform, the corresponding `README.html` file is displayed.

Figure 2-16 shows the files that are required to install the PowerVP agent on AIX/VIOS systems. You must upload these files to the logical partitions in binary mode and perform the installation manually. Section “Configuring PowerVP agent on AIX and VIOS” on page 32 describes how to perform manual installation on AIX/VIOS logical partitions.

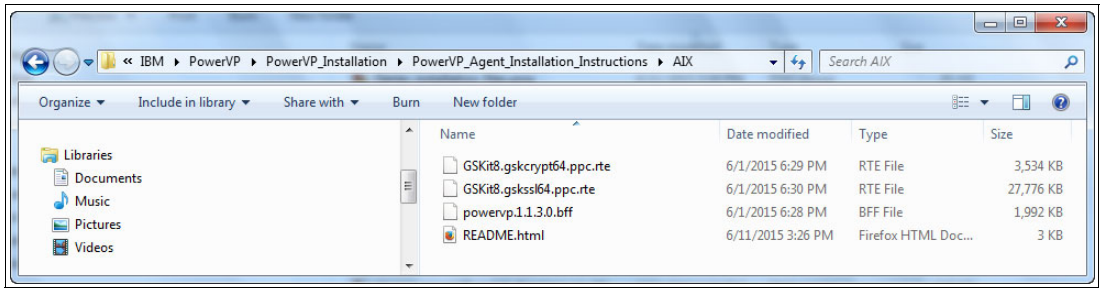


Figure 2-16 Files required to install PowerVP agent on AIX/VIOS systems

PowerVP installer extracts the packages specific to IBM i logical partitions and performs the agent installation automatically. If the installation fails for various reasons, the packages are available for manual installation as shown in Figure 2-17. You must upload these packages to the IBM i logical partitions on which the installation failed and perform the installation of the PowerVP agent manually. Section “Manual installation of PowerVP agent on IBM i” on page 30 describes how to perform manual installation on IBM i logical partitions.

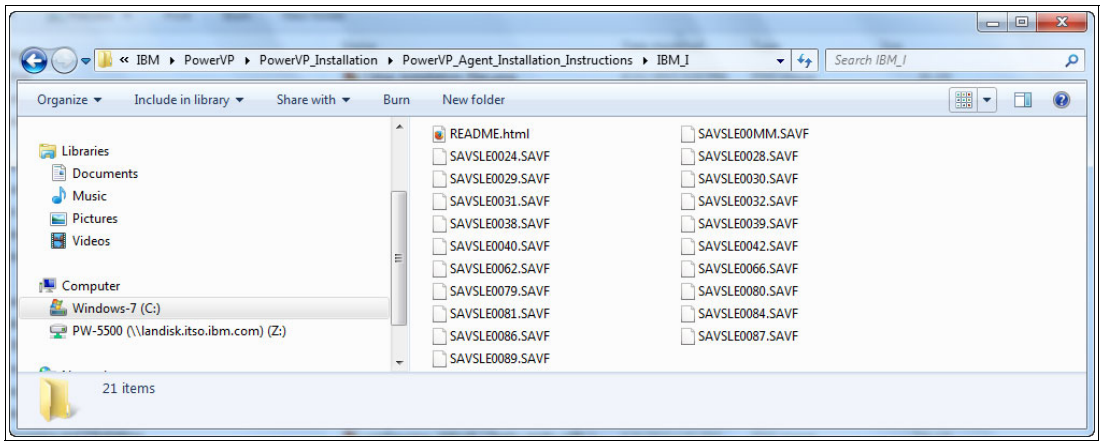


Figure 2-17 Files required to install PowerVP agent on IBM i systems

Figure 2-18 shows the files required to install the PowerVP agent for Linux systems. You must upload these files to the Linux partitions in binary mode and perform the installation of the PowerVP agent manually. Section “Configuring PowerVP agent on Linux” on page 38 describes how to perform manual installation on Linux logical partitions.

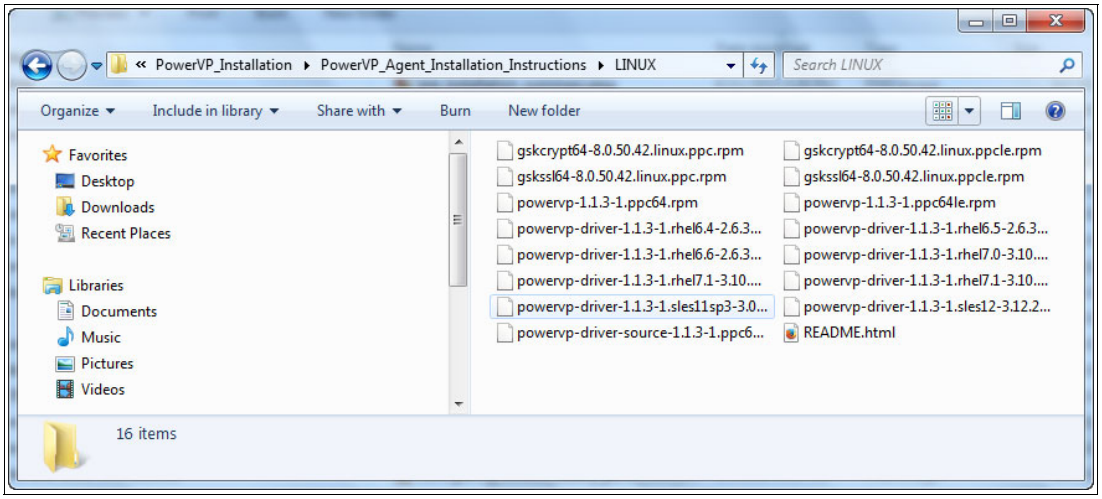


Figure 2-18 Files required to install PowerVP agents on Linux systems

2.2.3 Installing PowerVP GUI on AIX or Linux

This section shows how to use PowerVP installer on Linux and AIX systems. PowerVP installer can be run from Microsoft Windows as shown in 2.2.2, “PowerVP installer on Microsoft Windows systems” on page 13.

One of the major changes in PowerVP Version 1.1.3 is that PowerVP GUI interface is an application running on IBM WebSphere Application Server Liberty Core. The new PowerVP GUI provides an increased flexibility. You can install it on a remote server and connect to it by using a web browser. Liberty profile is included in the PowerVP installation package.

PowerVP installation wizard requires a functional graphical environment. When you deploy PowerVP GUI on a remote Linux or AIX partition, you must use a graphical remote access application.

In our scenario, we use VNC software. If you install PowerVP on a local computer the installation wizard is just the same.

Note: You can run `PowerVP.bin` file in a terminal with no graphic environment. When running in a terminal, PowerVP installer uses the default options and extracts files to the default location. This installation method does not install Liberty profile.

Before installation, you must copy `PowerVP.bin` and the license folder to the target server. To start remote installation, you must connect to the target computer using `vncviewer`. Ensure that you have the location of Java binary files specified in the `PATH` environment variable. You navigate to the installation folder and run the `PowerVP.bin` file. Graphical installation wizard starts as shown in Figure 2-19.

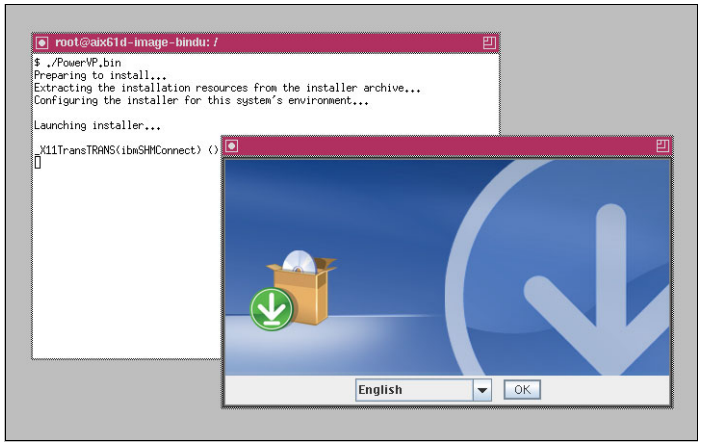


Figure 2-19 PowerVP GUI installation via remote VNC connection

After language selection you need to read and accept the license agreement. To install the PowerVP GUI, you must select either “Typical” or “PowerVP Client GUI”. On the next windows, you can specify the location where PowerVP GUI is installed and whether you want to create applications links.

You can choose whether to install Liberty profile and configure which network ports to be used for connections as shown in Figure 2-20. If you choose not to install the Liberty profile, the wizard only extracts the `PowerVP.war` and `server.xml` files.

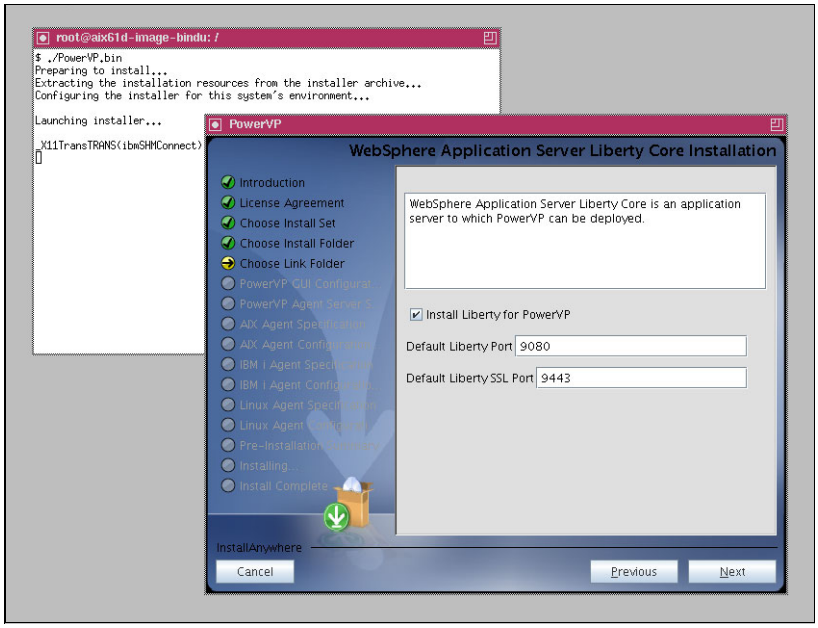


Figure 2-20 Using PowerVP installer GUI to install the Liberty profile

You continue through the next windows of the wizard and complete installation. PowerVP GUI requires Java version 7. In case of having multiple Java versions installed, you can set the `JAVA_HOME` environment variable to specify which version is used by default. You must ensure that you have the location of Java 7 binary files that are specified in the `PATH` variable.

To start the PowerVP GUI, you can either use the `PowerVP` link in the path that is specified during the installation or you can run `the launch-powervp` script in `[`*`path_to_PowerVP`*`]/PowerVP_Installation/PowerVP_GUI_Installation/`.

Launch script starts the Liberty profile, PowerVP GUI application, and a web browser pointing to the following URL:

`http://localhost:9080/powerVPWeb/PowerVP.html`

Note: After the installation is complete, you can change Liberty profile network ports and IP address by editing the `PowerVP_GUI_Installation/wlp/usr/servers/powervp/server.xml` file. By default, the application is bound to localhost. If you want to connect to the PowerVP GUI from a different computer you need to change the IP address.

Figure 2-21 shows a web browser that is connected to the PowerVP GUI installed on a remote computer.

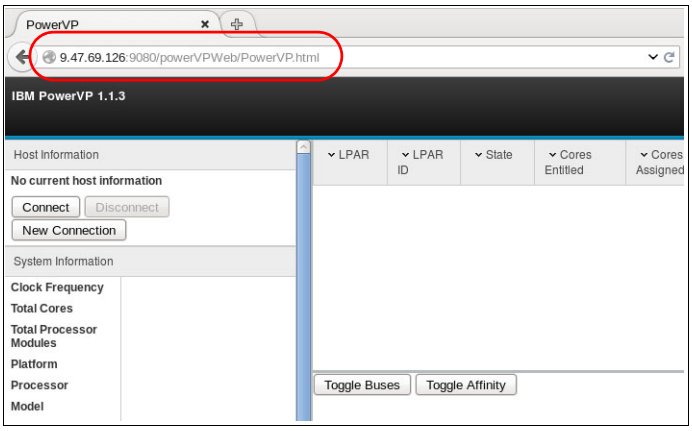


Figure 2-21 PowerVP GUI installed on a remote computer

2.2.4 Installing PowerVP agent manually on IBM i

This section describes how to install PowerVP manually on IBM i in case the automatic installation fails.

Manual installation of PowerVP agent on IBM i

PowerVP agent for IBM i can be installed manually following the instructions that are shown in Example 2-1.

Example 2-1 Instructions to manually install PowerVP on IBM i

```
Create a new library using:
CRTLIB POWERVP
Create a new savf on the IBM i system using:
CRTSAVF POWERVP/SAVSLE00MM
Create a new savf on IBM i system for language:
CRTSAVF POWERVP/SAVSLE0XX (Where XX represent the language code)
FTP the savsle00mm.savf and savsle00XX files in binary mode to the savf created on
IBM i system using: put *.savf[POWERVP]/[filename]
Run:
RSTLICPGM LICPGM(5765SLE) RSTOBJ(*PGM) DEV(*SAVF) SAVF(POWERVP/SAVSLE00MM)
and
RSTLICPGM LICPGM(5765SLE) RSTOBJ(*LNG) DEV(SAVF) SAVF(POWERVP/SAVSLExxxx)
Where xxxx is the language code. For US english it is 2924
```

Configuring PowerVP agent on IBM i

PowerVP agent for IBM i can be configured following the instructions shown in Example 2-2.

Example 2-2 Commands to configure PowerVP agent on IBM i

```
CALL QSLE/QPFICONFIG PARM('config-keyword=config-value'
'config-keyword=config-value')
CALL QLSE/QPFICONFIG PARM('Listen=*13000' SystemAgent=<IP Address of System level
agent server>)
```

Note: You can use the IP address or host name of the System level agent server if the host name can be resolved.

Verifying the installation of PowerVP agent on IBM i

To check whether PowerVP agent for IBM i agent is installed successfully, see Example 2-3.

Example 2-3 Instructions to verify PowerVP installation on IBM i

```
GO LICPGM
Select option 10
Check the list for product 5765SLE
```

2.2.5 Installing PowerVP agent on AIX and the VIOS

The PowerVP GUI installer only extracts the installp file sets locally. It does not automatically install the PowerVP agent on AIX/VIOS. The file sets are copied to the installation directory from where the installation GUI was run.

If you choose the default installation directory on a Microsoft Windows system, the files are found in:

```
C:\Program Files
\IBM\PowerVP\PowerVP_Installation\PowerVP_Agent_Installation_Instructions/AIX
```

There are three files that are contained in the installation directory as shown in Example 2-4.

Example 2-4 AIX and VIOS installation files

```
GSKit8.gskcrypt64.ppc.rte
GSKit8.gskssl64.ppc.rte
powervp.x.x.x.x.bff where x.x.x.x is 1.1.3.0 for the current version
```

To install the PowerVP agent on AIX or VIOS, complete the following steps:

1. Upload the three file sets into a directory on an AIX or VIOS where the agent needs to be installed.

   Note: You must upload the files in binary mode, or the installation will fail.

2. Log in to the server as root, change into the directory that contains the file sets, and run the following command:

   ```
   installp -agXd. powervp.rte
   ```

   The **`installp`** command also installs two extra GSKit file sets along with the `powervp` fileset.

   Note: You do not have to log in as root, but this process requires root privileges to install.

Configuring PowerVP agent on AIX and VIOS

The installed PowerVP agent can be configured using the **iconfig** program. The **iconfig** program automatically updates the PowerVP configuration file. The **iconfig** options depend on the type of the agent.

- To configure System level agent as the root user, run the following command:

  /opt/ibm/powervp/iconfig Listen="* *[Listen_port_of_System level agent]***" SystemLevelAgent=**

  Example 2-5 shows an example of this command.

  Example 2-5 Command to configure System level agent

```
/opt/ibm/powervp/iconfig Listen="* 13000" SystemLevelAgent=
```

- To configure Partition level agent as the root user, run the following command:

  /opt/ibm/powervp/iconfig Listen="* *[Listen_port_of_Partition level agent]***" SystemLevelAgent=***[IP_address_of_server_running_SystemLevelAgent]*

  Note: You can use the IP address or the host name of the System level agent server if the host name can be resolved.

  Example 2-6 shows an example of this command.

  Example 2-6 Command to configure Partition level agent

```
/opt/ibm/powervp/iconfig Listen="* 13000" SystemLevelAgent=9.3.185.56
```

Starting and stopping PowerVP agent on AIX and VIOS

To start the PowerVP agent on AIX or VIOS, run the start script as root user:

/etc/rc.d/rc2.d/SPowerVP

To stop PowerVP agent on AIX or VIOS, find the process number of **syslet** process and end it:

ps -ef |grep syslet

Note: It is important to terminate only the syslet process, and not its parent, which is `/opt/ibm/powervp/PowerVP.sh`. In case the syslet is terminated, the script unloads powervp kernel module and exits. If you kill the script, the module remains loaded. Proper termination is shown in Example 2-7.

Example 2-7 Terminating PowerVP agent on AIX or VIOS

```
# ps -ef | grep syslet
    root  7798970 13500668   0 11:24:54      -  0:00 /opt/ibm/powervp/syslet
# proctree 7798970
13500668    /bin/ksh /opt/ibm/powervp/PowerVP.sh
   7798970    /opt/ibm/powervp/syslet
# genkex | grep powervp
f1000000c030e000     4000 /opt/ibm/powervp/kex
# kill -9 7798970
# genkex | grep powervp
# ps -ef | grep powervp
#
```

Verifying that PowerVP agent started on AIX and VIOS

You can check whether the PowerVP agent is up and running by using the command that is shown in Example 2-8.

Example 2-8 Verifying whether the PowerVP agent is up and running

ps -ef |grep syslet

- Must display /opt/ibm/powervp/syslet process running

netstat -an |grep <PowerVP agent listening port>

- Must show a TCP port listening.

When PowerVP Partition level agents are running, you also see established connections between them and the System level agent as shown in Example 2-9:

- PowerVP System level agent is running.
- A new entry is added to the PowerVP log file when the System level agent is up and running.
- PowerVP is listening on port 13000.
- There are established connections between the System level agent running on 9.5.10.167 and the Partition level agents running on 9.5.80.109, 9.5.41.92, 9.5.41.22, 9.5.10.168, and 9.5.41.57.
- A new entry is added to the PowerVP log file on the partition when the Partition level agent is up and running and has successfully registered with the System level agent.

Example 2-9 Commands to verify PowerVP agent

```
# ps -ef|grep -i syslet
    root 10289194  9764910   1 16:06:59  pts/2  0:00 grep -i syslet
    root 11272216  7798944   0   Jun 10      -  3:47 /opt/ibm/powervp/syslet
# cat /var/log/powervp.log
2015-06-16 11:12:02.606 MSG0080: IP9-114-251-83.POK.STGLABS.IBM.COM - Running as
the system level agent.
2015-06-16 11:12:03.722 MSG0082: Agent running, waiting for clients to request
data.
# netstat -an|grep 1300
tcp4       0      0  9.5.10.167.13000       9.5.80.109.57028       ESTABLISHED
tcp4       0      0  9.5.10.167.13000       9.5.41.92.49115        ESTABLISHED
tcp4       0      0  9.5.10.167.13000       9.5.41.22.50330        ESTABLISHED
tcp        0      0  *.13000                *.*                    LISTEN
tcp4       0      0  9.5.10.167.13000       9.5.10.168.15916       ESTABLISHED
tcp4       0      0  9.5.10.167.13000       9.5.41.57.35110        ESTABLISHED
f1000e0000521300
[root@jay118 log]# cat /var/log/powervp.log

2015-06-16 12:36:56.942 JAY118.POK.STGLABS.IBM.COM - running in partition server
mode.
2015-06-16 12:36:58.049 Server running, waiting for clients to request data.
2015-06-16 12:36:58.178 Host JAY118.POK.STGLABS.IBM.COM successfully registered
with syslet server 9.114.251.83.
```

2.2.6 Installing PowerVP agent on Linux

The PowerVP GUI installation does not automatically install PowerVP agents on Linux. However, PowerVP installer extracts the required packages onto the local work station. The Linux installation media contains RPM files for RedHat and SUSE Linux distributions.

If first extracting the RPMs on Windows 7, the RPM files are in:

```
C:\Program
Files\IBM\PowerVP\PowerVP_Installation\PowerVP_Agent_Installation_Instructions\LIN
UX
```

Note: Starting with version 1.1.3, PowerVP introduces support for RHEL 7.1 and little endian. The corresponding kernel modules are:

- powervp-driver-1.1.3-1.rhel7.1-3.10.0-223.ael7ble.ppc64le.rpm for RHEL7
- powervp-driver-1.1.3-1.sles12-3.12.28-4.6.ppc64le.rpm for SLES

The directory has the installation files for RedHat and SUSE Linux distributions and the source for PowerVP driver as shown in Example 2-10.

Example 2-10 PowerVP Linux RPMs

```
gskcrypt64-8.0.50.42.linux.ppc.rpm
gskcrypt64-8.0.50.42.linux.ppcle.rpm
gskssl64-8.0.50.42.linux.ppc.rpm
gskssl64-8.0.50.42.linux.ppcle.rpm
powervp-1.1.3-1.ppc64.rpm
powervp-1.1.3-1.ppc64le.rpm
powervp-driver-1.1.3-1.rhel6.4-2.6.32-358.el6.ppc64.rpm
powervp-driver-1.1.3-1.rhel6.5-2.6.32-431.el6.ppc64.rpm
powervp-driver-1.1.3-1.rhel6.6-2.6.32-504.el6.ppc64.rpm
powervp-driver-1.1.3-1.rhel7.0-3.10.0-123.el7.ppc64.rpm
powervp-driver-1.1.3-1.rhel7.1-3.10.0-223.ael7ble.ppc64le.rpm
powervp-driver-1.1.3-1.rhel7.1-3.10.0-223.el7.ppc64.rpm
powervp-driver-1.1.3-1.sles11sp3-3.0.101-0.42.1.ppc64.rpm
powervp-driver-1.1.3-1.sles12-3.12.28-4.6.ppc64le.rpm
powervp-driver-source-1.1.3-1.ppc64.rpm
```

To install the PowerVP agent on Linux complete these steps:

1. The PowerVP agent has dependency on the following Linux packages:

   `sysstat, procps, perf, net-tools, ethtool, coreutils, ksh`

   Check and install all missing dependencies before installing PowerVP.

   Note: The dependent packages for PowerVP can have their own dependencies too. All dependent packages must be installed before installing PowerVP.

   Use**: rpm -qpR <rpm file>** to get the list of dependencies

   Example: **rpm -qpR perf-2.6.32-431.el6.ppc64.rpm**

2. Upload the necessary RPMs to the Linux partition. The RPM files that are required for installing the agent depend on the Linux distribution of the partition.
   - For RedHat Linux (RHEL 6.4), upload the RPM files as shown in Example 2-11.

     Example 2-11 PowerVP RHEL 6.4 RPMs

     ```
     powervp-x.x.x-1.ppc64.rpm
     gskcrypt64-8.0.50.42.linux.ppc.rpm
     gskssl64-8.0.50.42.linux.ppc.rpm
     powervp-driver-x.x.x-1.rhel6.4-2.6.32-358.el6.ppc64.rpm
     where x.x.x is 1.1.3 for current version of PowerVP
     ```

   - For RedHat Linux (RHEL 6.5), upload the RPM files that are shown in Example 2-12.

     Example 2-12 PowerVP RHEL 6.5 RPMs

     ```
     powervp-x.x.x-1.ppc64.rpm
     gskcrypt64-8.0.50.42.linux.ppc.rpm
     gskssl64-8.0.50.42.linux.ppc.rpm
     powervp-driver-1.1.3-1.rhel6.5-2.6.32-431.el6.ppc64.rpm
     where x.x.x is 1.1.3 for current version of PowerVP
     ```

   - For RedHat Linux (RHEL 6.6), upload the RPM files that are shown in Example 2-13.

     Example 2-13 PowerVP RHEL 6.6 RPMs

     ```
     powervp-x.x.x-1.ppc64.rpm
     gskcrypt64-8.0.50.42.linux.ppc.rpm
     gskssl64-8.0.50.42.linux.ppc.rpm
     powervp-driver-1.1.3-1.rhel6.6-2.6.32-504.el6.ppc64.rpm
     where x.x.x is 1.1.3 for current version of PowerVP
     ```

   - For RedHat Linux (RHEL 7.0), upload the RPM files that are shown in Example 2-14.

     Example 2-14 PowerVP RHEL 7.0 RPMs

     ```
     powervp-x.x.x-1.ppc64.rpm
     gskcrypt64-8.0.50.42.linux.ppc.rpm
     gskssl64-8.0.50.42.linux.ppc.rpm
     powervp-driver-1.1.3-1.rhel7.0-3.10.0-123.el7.ppc64.rpm
     where x.x.x is 1.1.3 for current version of PowerVP
     ```

   - For RedHat Linux Little Endian (RHEL 7.1), upload the RPM files that are shown in Example 2-15.

     Example 2-15 PowerVP RHEL 7.1 RPMs

     ```
     powervp-x.x.x-1.ppc64.rpm
     gskcrypt64-8.0.50.42.linux.ppc.rpm
     gskssl64-8.0.50.42.linux.ppc.rpm
     powervp-driver-1.1.3-1.rhel7.1-3.10.0-223.ael7ble.ppc64le.rpm
     where x.x.x is 1.1.3 for current version of PowerVP
     ```

– For RedHat Linux (RHEL 7.1), upload the RPM files that are shown in Example 2-16.

Example 2-16 PowerVP RHEL 7.1 RPMs

```
powervp-x.x.x-1.ppc64.rpm
gskcrypt64-8.0.50.42.linux.ppc.rpm
gskssl64-8.0.50.42.linux.ppc.rpm
powervp-driver-1.1.3-1.rhel7.1-3.10.0-223.el7.ppc64.rpm
where x.x.x is 1.1.3 for current version of PowerVP
```

– For SUSE Linux (SLES 11 SP3), upload the RPM files that are shown in Example 2-17.

Example 2-17 PowerVP SLES 11 SP3 RPMs

```
powervp-x.x.x-1.ppc64.rpm
gskcrypt64-8.0.50.42.linux.ppc.rpm
gskssl64-8.0.50.42.linux.ppc.rpm
powervp-driver-1.1.3-1.sles11sp3-3.0.101-0.42.1.ppc64.rpm
where x.x.x is 1.1.3 for current version of PowerVP
```

– For SUSE Linux (SLES 12), upload the RPM files that are shown in Example 2-18.

Example 2-18 PowerVP SLES 12 RPMs

```
powervp-x.x.x-1.ppc64.rpm
gskcrypt64-8.0.50.42.linux.ppc.rpm
gskssl64-8.0.50.42.linux.ppc.rpm
powervp-driver-1.1.3-1.sles12-3.12.46.ppc64le.rpm
where x.x.x is 1.1.3 for current version of PowerVP
```

– For any other Linux distribution, upload the RPM files as shown in the Example 2-19. For big endian distributions, use the files with `ppc.rpm` and `ppc64.rpm` at the end of the file name. For little endian distributions, use the files with the `ppcle.rpm` and `ppc64le.rpm` at the end of the file name.

Example 2-19 PowerVP generic Linux RPMs

```
powervp-x.x.x-1.ppc64.rpm
gskcrypt64-8.0.50.42.linux.ppc.rpm
gskssl64-8.0.50.42.linux.ppc.rpm
powervp-driver-source-x.x.x-1.ppc64.rpm
where x.x.x is 1.1.3 for current version of PowerVP
```

3. Following are the commands to install the PowerVP agent on different Linux distributions:
   – On RHEL 6.4, run the command that is shown in Example 2-20.

Example 2-20 PowerVP agent installation command for RHEL 6.4

```
rpm -i powervp-x.x.x-1.ppc64.rpm gskcrypt64-8.0.50.11.linux.ppc.rpm
gskssl64-8.0.50.11.linux.ppc.rpm
powervp-driver-1.1.3-1.rhel6.4-2.6.32-358.el6.ppc64.rpm

where x.x.x is 1.1.3 for current version of PowerVP
```

– On RHEL 6.5, run the command that is shown in Example 2-21.

Example 2-21 PowerVP agent installation command for RHEL 6.5

```
rpm -i powervp-x.x.x-1.ppc64.rpm gskcrypt64-8.0.50.11.linux.ppc.rpm
gskssl64-8.0.50.11.linux.ppc.rpm
powervp-driver-1.1.3-1.rhel6.5-2.6.32-431.el6.ppc64.rpm

where x.x.x is 1.1.3 for current version of PowerVP
```

– On SLES 11 SP3, run the command that is shown in Example 2-22.

Example 2-22 PowerVP agent installation command for SLES 11 SP3

```
rpm -i powervp-x.x.x-1.ppc64.rpm gskcrypt64-8.0.50.11.linux.ppc.rpm
gskssl64-8.0.50.11.linux.ppc.rpm
powervp-driver-1.1.3-1.sles11sp3-3.0.76-0.11.1.ppc64.rpm

where x.x.x is 1.1.3 for current version of PowerVP
```

– On other Linux distributions, installing PowerVP agent requires building `powervp-driver` for that particular Linux distribution. To build the driver, complete these steps:

  i. Install the `powervp-driver-source-x.x.x-1.ppc64.rpm` by using this command:

     `rpm -i powervp-driver-source-x.x.x-1.ppc64.rpm`

     where x.x.x is 1.1.3, the current version of PowerVP

     The prerequisites for this are kernel-level package, gcc, and rpm-build.

  ii. Change directory to:

     `/opt/ibm/powervp/driver-source`

  iii. Run **`make`**.

> **Note:** There can be prerequisite packages that are needed to successfully **`make`** the kernel module.

4. After the `powervp-driver` is built, install the PowerVP as shown in Example 2-23.

Example 2-23 PowerVP agent installation on other Linux distributions

```
rpm -i powervp-x.x.x-1.ppc64.rpm gskcrypt64-8.0.50.42.linux.ppc.rpm
gskssl64-8.0.50.42.linux.ppc.rpm powervp-driver-1.1.3-1.yyyyy.ppc64.rpm

where x.x.x is 1.1.3 for current version of PowerVP
where yyyyy is linux kernel version of the linux distribution
```

Full details about installation instructions are provided in the `README.html` file that is provided along with the binary files.

Configuring PowerVP agent on Linux

The PowerVP is installed in the `/opt/ibm/powervp` directory. The PowerVP agent can be configured to run as a System level agent or as a Partition level agent. The commands are the same for all Linux distributions:

- To configure PowerVP as a System level agent on Linux, run the following command:

  `/opt/ibm/powervp/iconfig Listen`=*"* [Listen_port_of_System level agent]"* **`SystemLevelAgent`**=

  Example 2-24 shows an example of this command.

  Example 2-24 Example of the /opt/ibm/powervp/iconfig command

  ```
  /opt/ibm/powervp/iconfig Listen="* 13000" SystemLevelAgent=
  ```

- To configure PowerVP as a Partition level agent on Linux, run this command:

  `/opt/ibm/powervp/iconfig Listen`="* *[Listen_port_of_Partition level agent]*" **`SystemLevelAgent`**=*[IP_address_of_server_running_SystemLevelAgent*]

  Note: The IP address or host name can be used for the System level agent server.

  Example 2-25 shows an example of this command

  Example 2-25 Example of the /opt/ibm/powervp/iconfig command

  ```
  /opt/ibm/powervp/iconfig Listen="* 13000" SystemLevelAgent=9.3.185.56
  ```

Start and stop of PowerVP agent on Linux

The start and stop commands on all Linux distributions are the same. PowerVP installs a start script in `/etc/init.d/ibm_powervp`.

- To start PowerVP agent, run the following command as root user:

  ```
  service ibm_powervp start
  ```

- To stop PowerVP agent, run the following command as root user:

  ```
  service ibm_powervp stop
  ```

Verifying that PowerVP agent started on Linux

There are two ways to check whether a PowerVP agent is successfully started on a Linux partition, as shown in Example 2-26.

Example 2-26 Instructions to check PowerVP agent started on Linux

`ps -ef |grep syslet`

- Must show the `/opt/ibm/powervp/syslet` command running

`netstat -an |grep <PowerVP agent listening port>`

- Must show TCP port listening
- If the PowerVP agent is configured as Partition level agent, you will also an established connection to System level agent

2.3 Starting and stopping PowerVP

PowerVP is an individual web application that uses features that are provided by the Liberty profile. There might be cases when you want to start and stop PowerVP from command line. On Microsoft Windows systems, you can use commands similar to what is shown in Example 2-27.

Example 2-27 Starting and stopping PowerVP from command line on Microsoft Windows systems

```
C:\Program Files\IBM\PowerVP\PowerVP_Installation\PowerVP_GUI_Installation>launc
h-powervp.bat

C:\Program Files\IBM\PowerVP\PowerVP_Installation\PowerVP_GUI_Installation>cd "C
:\Program Files\IBM\PowerVP\PowerVP_Installation\PowerVP_GUI_Installation\wlp\bi
n"

C:\Program Files\IBM\PowerVP\PowerVP_Installation\PowerVP_GUI_Installation\wlp\b
in>call server start powervp
Starting server powervp.
Server powervp started.

C:\Program Files\IBM\PowerVP\PowerVP_Installation\PowerVP_GUI_Installation>stop-
powervp.bat

C:\Program Files\IBM\PowerVP\PowerVP_Installation\PowerVP_GUI_Installation>cd "C
:\Program Files\IBM\PowerVP\PowerVP_Installation\PowerVP_GUI_Installation\wlp\bi
n"

C:\Program Files\IBM\PowerVP\PowerVP_Installation\PowerVP_GUI_Installation\wlp\b
in>call server stop powervp
Stopping server powervp.
Server powervp stopped.
```

PowerVP settings are in a configuration file that is named `server.xml`, which is shown in Example 2-28.

Example 2-28 PowerVP xml configuration file

```
<server description="powervp server">

    <!-- Enable features -->
    <featureManager>
        <feature>jsp-2.2</feature>
        <feature>localConnector-1.0</feature>
        <feature>servlet-3.1</feature>
    <feature>websocket-1.0</feature>
       <feature>jsonp-1.0</feature>
        <feature>jaxrs-1.1</feature>
    </featureManager>

    <!-- To access this server from a remote client add a host attribute to the
following element, e.g. host="*" -->
    <httpEndpoint host="localhost" httpPort="9080" httpsPort="9443"
id="defaultHttpEndpoint"/>
```

```
    <applicationMonitor updateTrigger="mbean"/>

    <webApplication id="powerVPWeb" location="powerVPWeb.war" name="powerVPWeb"/>
</server>
```

The configuration file includes the port numbers that are used by the server as specified during the configuration phase. By default, the server uses TCP/IP port 9080 for HTTP traffic, and port 9443 for HTTPS traffic.

2.4 Upgrading PowerVP agents

There is no direct upgrade path from previous versions of PowerVP agents to Version 1.1.3. To upgrade to Version1.1.3, you must install the newer PowerVP agent. You cannot register previous versions of PowerVP Partition level agents to a PowerVP Version 1.1.3 System level agent.

When upgrading to PowerVP Version 1.1.3, upgrade the PowerVP GUI first, followed by the System level agent and then Partition level agents.

To upgrade a PowerVP agent, follow these steps:

1. Stop the PowerVP agent.

   Note: See Example 2-7 on page 32 about how to properly stop the PowerVP agent on AIX and VIOS. For Linux, ensure that the agent is stopped by stop script. It unloads the kernel module.

2. Uninstall PowerVP packages for previous versions.
3. Install PowerVP 1.1.3 agent.
4. Configure the newly installed agent and start.
5. Verify that the agent is up and running and you can connect to it using the PowerVP GUI.

   Note: PowerVP keeps an old copy of the `powervp.conf` file for reference.

2.5 PowerVP SSL configuration

PowerVP offers two types of connections when connecting to the agents: non-secure connection and a secure connection. The non-secure type does not secure the data that is transferred between the System level agent, the Partition level agent, and PowerVP.

PowerVP supports the use of SSL application protocol to secure the data transferred between the System level agent, the Partition level agent, and PowerVP. The SSL protocol is used to verify the identity of data senders and receivers and to encrypt the data exchanged between them.

To secure the connectivity, you must meet the following conditions:

- SSL certificates must be used on all partitions.

- The same SSL port must be used for all agents, both System level agents and Partition level agents.
- You must use a self-signed certificate, a certificate signed by certificate authority (CA), or an SSL certificate that is issued by a trusted certificate authority.

2.5.1 Using OpenSSL to secure PowerVP connectivity

This section shows how to quickly generate certificates signed by a local CA. We use OpenSSL installed on the VIOS partition, which runs a System level agent. The procedure of generating the certificates includes the following activities:

- Setting up the CA
- Generating the CA certificate
- Generating a Certificate Signing Request (CSR)
- Signing a CSR to generate a signed certificate

To generate a certificate for use with PowerVP, complete the following steps:

1. Set up the CA on the VIOS partition. The CA uses a pre-defined hierarchy of files and directories, which references a *home directory*. For our scenario, we create a home directory for OpenSSL named `/cert_repo`. We also create other subdirectories and files that are required by OpenSSL as shown in Example 2-29.

Example 2-29 Creating the CA infrastructure

```
# cat  /var/ssl/openssl.cnf|grep -ip repo
dir             = /cert_repo            # Where everything is kept
certs           = $dir/certs            # Where the issued certs are kept
crl_dir         = $dir/crl              # Where the issued crl are kept
database        = $dir/index.txt        # database index file.
#unique_subject = no                    # Set to 'no' to allow creation of
                                        # several certificates with same subject.
new_certs_dir   = $dir/newcerts         # default place for new certs.
# mkdir /cert_repo
# chmod 700 /cert_repo
# cd /cert_repo
# mkdir certs private newcerts
# echo 1024 >> serial
# touch index.txt
```

For more information about setting up a CA with OpenSSL, see the following site:

http://www.openssl.org

2. The CA must have its own key and certificate. Generate the key and the CA certificate as shown in Example 2-30. Use the Fully Qualified Domain Name (FQDN) for the common name.

Example 2-30 Creating CA key and certificate

```
# openssl req -new -x509 -days 365 -extensions v3_ca -keyout private/cakey.pem
-out cacert.pem -config  /var/ssl/openssl.cnf                                  <
Generating a 1024 bit RSA private key
...................++++++
................++++++
writing new private key to 'private/cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
```

```
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:NY
Locality Name (eg, city) []:Poughkeepsie
Organization Name (eg, company) [Internet Widgits Pty Ltd]:IBM
Organizational Unit Name (eg, section) []:IT
Common Name (e.g. server FQDN or YOUR name) []:ip9-114-251-83.pok.stglabs.ibm.com
Email Address []:administrator at p9_vios1 dot ibm dot com
```

3. Use the **openssl** command to review the details of the CA certificate as shown Example 2-31.

Example 2-31 Reviewing details of the CA certificate

```
# openssl x509 -in cacert.pem -noout -text
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            85:03:34:71:54:b9:69:4a
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=US, ST=NY, L=Poughkeepsie, O=IBM, OU=IT,
CN=ip9-114-251-83.pok.stglabs.ibm.com/emailAddress=administrator at p9_vios1 dot
ibm dot com
        Validity
            Not Before: Jun 25 13:41:41 2015 GMT
            Not After : Jun 24 13:41:41 2016 GMT
        Subject: C=US, ST=NY, L=Poughkeepsie, O=IBM, OU=IT,
CN=ip9-114-251-83.pok.stglabs.ibm.com/emailAddress=administrator at p9_vios1 dot
ibm dot com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (1024 bit)
                Modulus:
                    00:a9:26:17:42:9a:56:96:bf:08:3e:75:ba:e9:4e:
                    f5:d5:69:0d:4f:9e:c3:72:22:44:d8:62:b5:69:a2:
                    73:b7:72:42:da:7d:b0:56:4b:54:76:7c:48:8c:eb:
                    19:08:a1:4a:87:7d:ee:4d:5a:fc:23:ec:55:6c:ad:
                    e4:1b:5a:2f:eb:70:7a:e7:e0:d6:fd:cc:c8:d6:66:
                    3e:33:f8:7d:26:02:04:af:f4:d6:56:98:84:7c:18:
                    78:aa:97:2a:b6:29:dd:66:15:27:ab:01:8a:3f:57:
                    c1:23:f0:5c:4e:8c:a1:3d:2d:8d:78:4b:9d:7a:70:
                    48:2f:e0:81:c0:c0:cd:d8:b9
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                D2:93:A6:39:3E:65:80:6C:F6:A1:E8:48:40:5B:2C:A0:1B:AD:2F:DA
            X509v3 Authority Key Identifier:
```

```
            keyid:D2:93:A6:39:3E:65:80:6C:F6:A1:E8:48:40:5B:2C:A0:1B:AD:2F:DA

        X509v3 Basic Constraints:
            CA:TRUE
 Signature Algorithm: sha1WithRSAEncryption
      60:5f:c1:94:b5:f4:84:0f:56:67:fe:44:f0:13:dd:1e:ae:76:
      74:6b:36:d9:c8:95:5e:c2:8d:a6:74:0d:58:64:fb:76:78:c3:
      77:d4:83:55:38:d8:7f:28:40:0e:72:3a:fe:72:d5:4f:7d:28:
      39:92:7f:79:e1:0b:ba:bf:1c:5a:04:40:54:59:86:aa:31:9d:
      57:68:54:27:b7:53:44:5b:cb:5c:c3:02:d8:31:84:19:b4:67:
      c0:4d:cc:8f:fd:2b:cb:07:b0:c7:1e:56:b7:9d:c7:8c:fa:2e:
      73:92:dc:24:de:a3:28:c0:26:16:33:1f:94:b8:cf:30:42:23:
      bf:2f
```

4. Generate the certificates used by web browsers to connect to the PowerVP GUI. Generate a CSR for a new certificate as shown in Example 2-32.

Example 2-32 Creating a request to sign a certificate

```
# openssl req -newkey rsa:1024 -keyout tempkey.pem -keyform PEM -out
tempreq.pem -outform PEM
Generating a 1024 bit RSA private key
..++++++
.................++++++
writing new private key to 'tempkey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:NY
Locality Name (eg, city) []:Poughkeepsie
Organization Name (eg, company) [Internet Widgits Pty Ltd]:IBM
Organizational Unit Name (eg, section) []:IT
Common Name (e.g. server FQDN or YOUR name)
[]:ip9-114-251-83.pok.stglabs.ibm.com
Email Address []:administrator at p9_vios1 dot ibm dot com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

5. Use the CA to sign the CSR as shown in Example 2-33.

Example 2-33 Signing the certificate request

```
# openssl ca -in tempreq.pem -out client.pem
Using configuration from /var/ssl/openssl.cnf
Enter pass phrase for /cert_repo/private/cakey.pem:
Check that the request matches the signature
```

```
Signature ok
Certificate Details:
        Serial Number: 4132 (0x1024)
        Validity
            Not Before: Jun 25 13:51:02 2015 GMT
            Not After : Jun 24 13:51:02 2016 GMT
        Subject:
            countryName               = US
            stateOrProvinceName       = NY
            organizationName          = IBM
            organizationalUnitName    = IT
            commonName                = ip9-114-251-83.pok.stglabs.ibm.com
            emailAddress              = administrator at p9_vios1 dot ibm dot
com
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                88:14:53:C6:BD:BA:DE:0E:12:9D:13:57:28:BB:8B:91:2A:63:1B:28
            X509v3 Authority Key Identifier:

keyid:D2:93:A6:39:3E:65:80:6C:F6:A1:E8:48:40:5B:2C:A0:1B:AD:2F:DA

Certificate is to be certified until Jun 24 13:51:02 2016 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

6. Generate the client certificate as shown in Example 2-34. The certificate is included in the `client.pem` file.

Example 2-34 Generating client certificate

```
# openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout client.pem -out
client.pem

Generating a 1024 bit RSA private key
...........++++++
...........++++++
writing new private key to 'client.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:NY
Locality Name (eg, city) []:Poughkeepsie
```

```
Organization Name (eg, company) [Internet Widgits Pty Ltd]:IBM
Organizational Unit Name (eg, section) []:IT
Common Name (e.g. server FQDN or YOUR name)
[]:ip9-114-251-83.pok.stglabs.ibm.com
Email Address []:administrator at p9_vios1 dot ibm dot com
```

7. Convert the client certificate to one of the formats supported by PowerVP. Use the Public-Key Cryptography Standard 12 (PKCS12) format and label the certificate `PowervPGUI` as shown in Example 2-35. The file named `client.p12` contains the certificate in the new format. Copy the new certificate file to the `/etc/opt/ibm/powervp/certs` directory, which is the PowerVP default location for storing certificates.

Example 2-35 Exporting the certificate in PKCS12 format

```
# openssl pkcs12 -export -out client.p12 -in client.pem -name PowervPGUI
Enter Export Password:
Verifying - Enter Export Password:
# cp client.p12 /etc/opt/ibm/powervp/certs/client.p12
```

You can review the label of existing certificates by using a command similar to what is shown in Example 2-36. The certificate can be flagged as either * (default) or - (personal). The label of the certificate is displayed next to the flag.

Example 2-36 Displaying a certificate label

```
#/usr/opt/ibm/gsk8_64/bin/gsk8capicmd_64 -cert -list all -db
/etc/opt/ibm/powervp/certs/client.p12                              <
Source database password :
Certificates found
* default, - personal, ! trusted, # secret key
-       PowervPGUI
```

2.5.2 Configuring PowerVP to use certificates

Follow these steps to configure PowerVP to use certificates:

1. After the certificate is created, PowerVP requires a password to use the certificate. You can provide the password in either of the following two ways:
   - Edit the `powervp.conf` file and provide a clear text password.
   - Encrypt the password into a separate stash file.

> **Note:** We recommend encrypting the password by using a stash file. If you use a plain text password in the `powervp.conf` file, we recommend setting restricted permissions on the `/etc/opt/ibm/powervp/powervp.conf` file.

You can stash the password by using a command similar to what is shown in Example 2-37. A *stash file* is created in the `/etc/opt/ibm/powervp/certs` directory.

Example 2-37 Stashing the password

```
# /usr/opt/ibm/gsk8_64/bin/gsk8capicmd_64 -keydb -stashpw -db
/etc/opt/ibm/powervp/certs/client.p12                              <
Source database password :
# ls -l /etc/opt/ibm/powervp/certs
```

```
total 48
-rw-r--r--    1 root     staff          1914 Jun 25 09:06 client.p12
-rw-------    1 root     staff           129 Jun 25 09:08 client.sth
```

2. The PowerVP agent configuration file must be updated with all data required to use the certificate. You can use the **iconfig** command or manually edit the /etc/opt/ibm/powervp/powervp.conf file. The relevant lines of the file are listed in Example 2-38.

Example 2-38 PowerVP configuration file for certificate usage

```
# cat /etc/opt/ibm/powervp/powervp.conf|grep -v "#"|egrep
"KeyringFile|StashFile|CertificateLabel"
KeyringFile client.p12
StashFile client.sth
CertificateLabel PowervPGUI
```

Note: If the certificate label contains spaces you must enclose the label in quotation marks, for example: “Power VP”.

3. The PowerVP agent configuration file must be updated to use the appropriate port to allow for SSL connectivity. By default, PowerVP uses port 13000 and port 13001 for SSL connections. You can use the **iconfig** command or manually edit the /etc/opt/ibm/powervp/powervp.conf file. The commands that are used to update the powervp.conf to listen to both default non-SSL and SSL ports are shown in Example 2-39.

Example 2-39 Commands to configure PowerVP agent to listen on 13000 and 13001 ports

```
For System level agent
/opt/ibm/powervp/iconfig Listen=”*13000”,”* 13001 ssl”
For Partition level agent
/opt/ibm/powervp/iconfig Listen =”*13000”, “*13001 ssl”
SystemLevelAgent=[ip_address_of_System level agent]
```

To configure the PowerVP agent to listen *only* to 13001 SSL port, you can run the commands that are shown in Example 2-40.

Example 2-40 Command to configure PowerVP agent to listen on port 13001

```
For System level agent
/opt/ibm/powervp/iconfig Listen=”* 13001 ssl”
For Partition level agent
/opt/ibm/powervp/iconfig Listen=”* 13001 ssl”
SystemLevelAgent=[ip_address_of_System level agent]
```

PowerVP provides support for protocols TLS1.0, TLS 1.1, and TLS 1.2. The protocols can be enabled by editing the PowerVP configuration file.

4. After PowerVP is configured, restart the PowerVP agent and verify if the agent is listening on the specified SSL port by using the **netstat** command. Example 2-41 shows PowerVP agent listening on port 13000 for non-SSL connections and port 13001 for SSL connections.

Example 2-41 Verifying that PowerVP agent is listening to the specified ports

```
# netstat -an|grep 1300
tcp        0      0  *.13000                *.*                    LISTEN
```

```
tcp        0      0  *.13001                *.*                    LISTEN
```

5. Select **Secure** and ensure that you specify the appropriate SSL port in PowerVP GUI as shown in Figure 2-22.

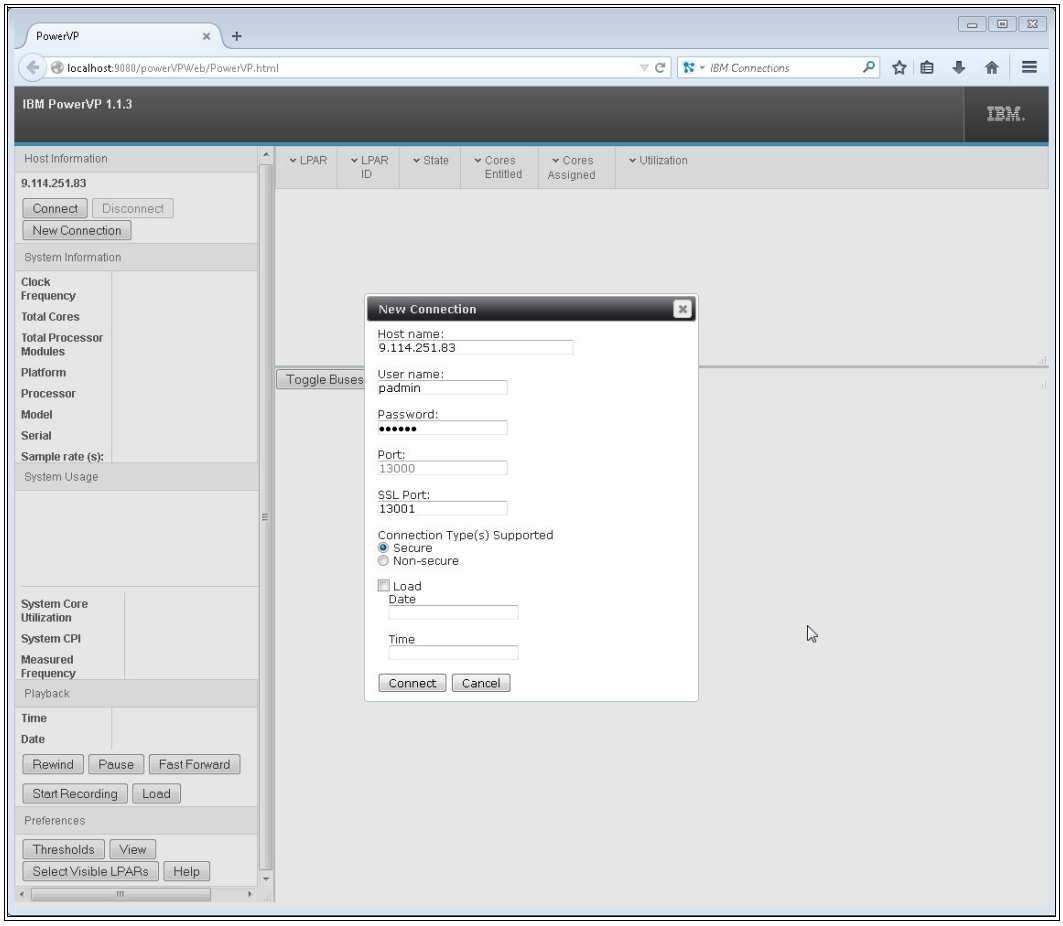


Figure 2-22 Selecting SSL connectivity for PowerVP

Note:

- Appropriate firewall rules must be in place to allow PowerVP GUI to connect to the agents.
- When changing from a non-secure to a secure connection, ensure that you select **Secure** and use the correct port number for SSL that is specified.

3

The PowerVP GUI

This chapter explains in detail how IBM Power Virtualization Performance (PowerVP) graphical user interface (GUI) can be used in environments that comprise the IBM Power system.

This chapter includes the following sections:

- PowerVP client GUI
- PowerVP recording and playback
- PowerVP permanent data store
- Customizing the PowerVP GUI

3.1 PowerVP client GUI

The PowerVP client GUI is typically installed as part of the PowerVP agent installation. It can be installed by itself or by using PowerVP installer. PowerVP GUI must first connect to the System level agent before it can be used for monitoring performance on a Power system. To connect, click **Connect** or **New Connection** on the Host Information window pane. A new connection window is displayed as shown in Figure 3-1.

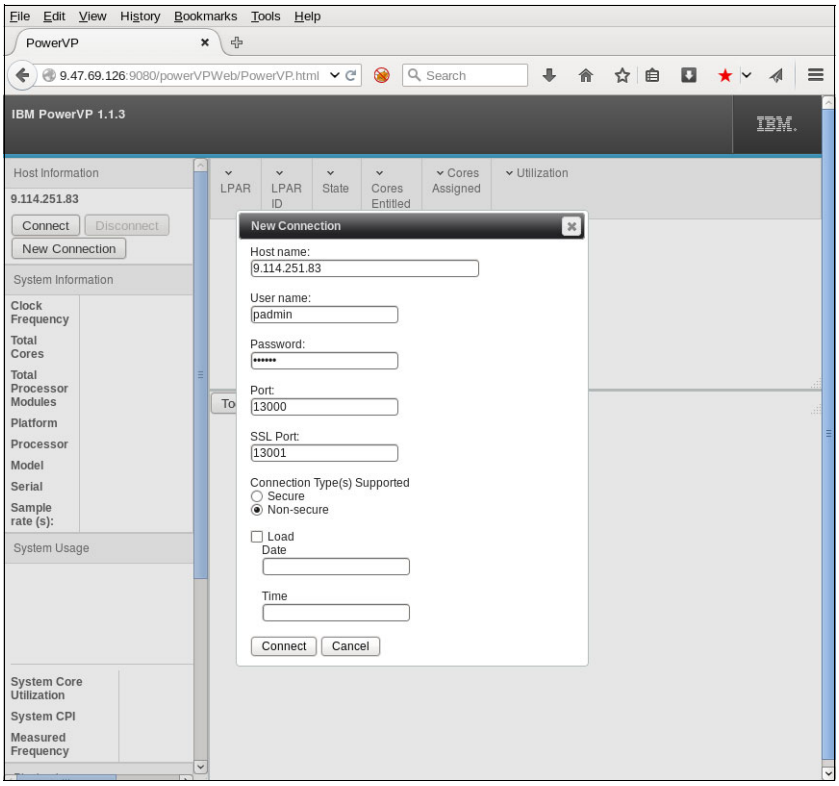


Figure 3-1 PowerVP new connection window

In the **Host name** field, enter the IP address or host name of the partition where the System level agent is installed. The **Port** and **SSL Port** fields are pre-configured with the default port numbers 13000 for non-SSL and 13001 for SSL. If you want to use different ports, change them accordingly. For **Connection Type(s) Supported**, select **Secure** or **Non-secure** based on your System level agent configuration. In the login credentials section, enter a valid user name and password, and click **Connect**.

3.1.1 PowerVP GUI main window

The main window of PowerVP GUI is displayed when the connection to the System level agent is successful. The main window has six different focus areas as shown in Figure 3-2.

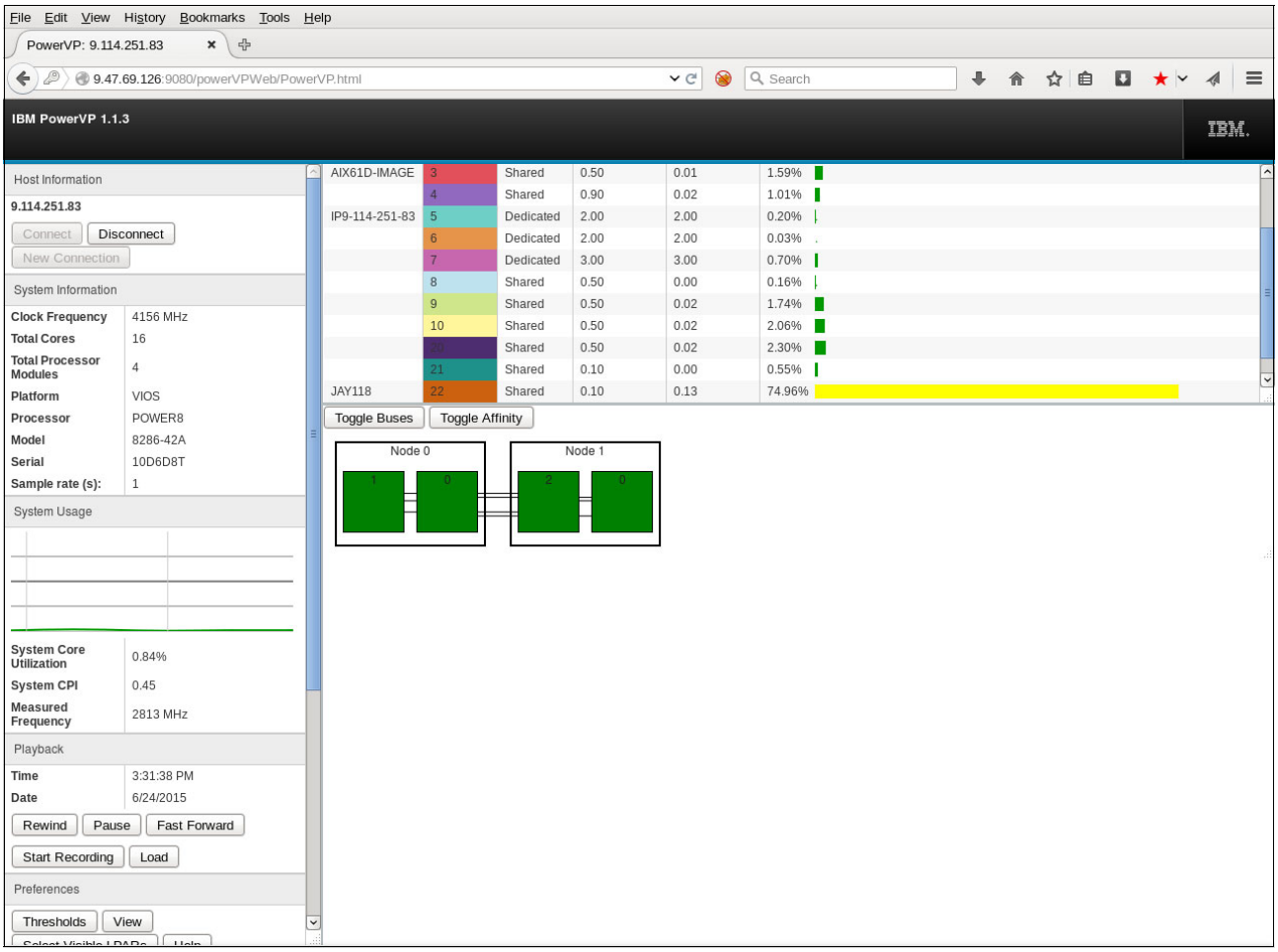


Figure 3-2 PowerVP main window

The following six areas are included in the main window:

- Host Information

  This section provides PowerVP GUI connectivity control. If you are connected, it displays the host name of the destination PowerVP system agent. If you are not connected, you can connect to previously accessed PowerVP system agent or set up a new connection.

- System Information

  The system information area displays a summary of the Power System that is being monitored. It displays information related to POWER7 or POWER8 processor-based systems, total cores on the system, total processor modules, platform of the partition, server model, serial number, and sample rate interval.

- System Usage

  The system usage section provides the overall usage of the system that includes cycles per instruction (CPI) and a running graph of system overall core utilization. The CPI gives an overall efficiency of the Power System. Typically, a lower CPI is better than a higher CPI.

It also provides measured frequency of the core from the Hypervisor.

- Playback

  This section gives you control over record and playback performance data with PowerVP GUI interface. PowerVP version 1.1.3 provides two different methods to record and play back historical data. For more information, see 3.2, “PowerVP recording and playback” on page 60.

- Preferences

  The preferences section provides possibility to customize PowerVP GUI interface. The **Thresholds** button allows you to customize the color of utilization bars according to the percentage of usage. It is configurable for memory controllers, buses, and CPU cores. The **Select Visible LPARs** button allows you to select which partitions are visible on the right side. By default, all partitions are visible. The **View** button allows you to customize PowerVP GUI window panes.

- VIOS Performance Advisor

  A new feature introduced in PowerVP 1.1.3 is integration with VIOS Performance Advisor. The control section is available in PowerVP GUI as shown on Figure 3-3.

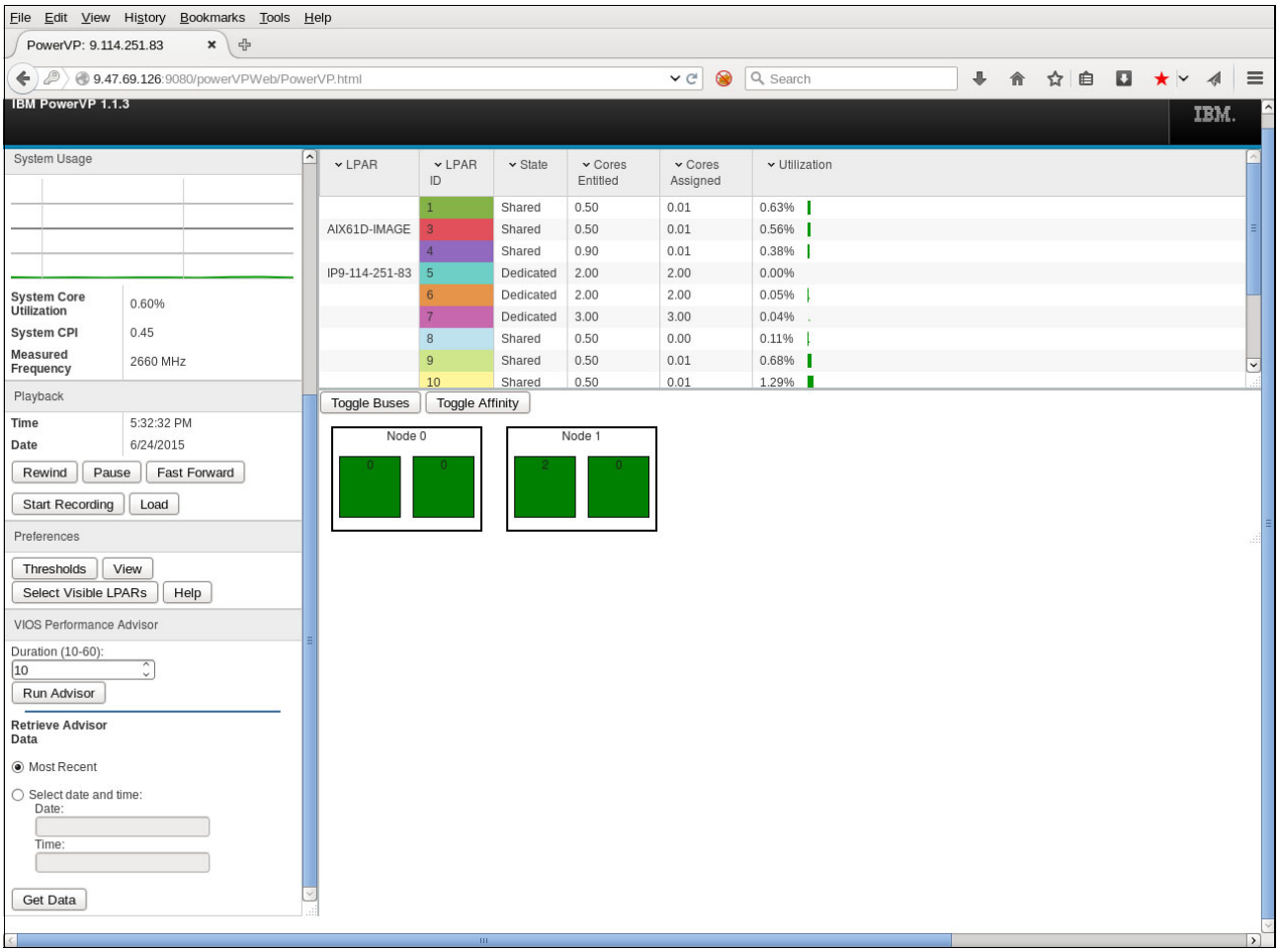


Figure 3-3 PowerVP GUI VIOS Performance Advisor

More information about VIOS Performance Advisor is available in section 5.6, “Integration with VIOS performance advisor” on page 99.

- Partition list

  The right side of the main window is split into two sections. The top pane contains a list of all partitions on the Power System as shown in Figure 3-4.

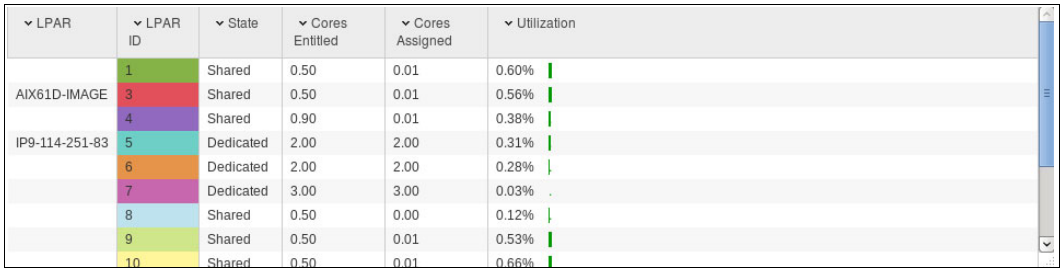


Figure 3-4 Partition list

There are six columns. The table has a sortable header, so you can click an individual column heading to sort the table:

  - The first column is the **LPAR** column. This column can be used to identify what partitions have agents installed. A blank indicates that no partition agent is installed.
  - The second column is **LPAR ID**. It shows the LPAR identification number of the partition and matches the HMC profile data.
  - The third column is **State**. It shows if the partition is shared or dedicated.
  - The fourth column is **Cores Entitled**. It displays the configured entitlement of the partition.
  - The fifth column is **Cores Assigned**. It displays cores that are currently used by the partition.
  - The last column, **Utilization**, shows a moving bar that indicates the percent of CPU utilization of the partition. This column can be used to easily identify the partitions that are overcommitted.

- System Display

  The large section is a graphical representation of the Power System in the system display. Depending on the Power System type and model, there will be single or multiple nodes represented by boxes.

  Inside these node boxes are the boxes that account for the processor modules on the node. If you select **Toggle Buses**, PowerVP GUI shows lines between the processor module boxes and processor nodes, which represent buses. The color and the numbers on processor module boxes indicate the usage as shown in Figure 3-5. On the screen capture, you can see a POWER8 System with two nodes (Node0 and Node1). **Toggle Affinity** is intended to show affinity where every partition has a different color.

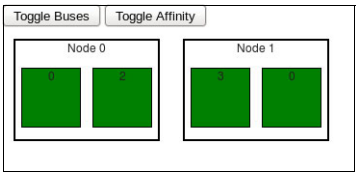


Figure 3-5 PowerVP GUI system display

3.1.2 PowerVP hardware drill-down display

The PowerVP hardware drill-down display appears when you click the node boxes in the system display pane of the main PowerVP window. The hardware drill-down shows the detailed hardware structure of the node. The processor modules are shown in larger boxes, and within the boxes are columns that represent the cores in each module. The core utilization is depicted on the cores by bars, and the colors of the columns change based on core utilization thresholds that are configured.

The lines between processor modules represent the buses between the modules. If the lines extend beyond the display, they are the lines depicting the buses between the nodes.

If the processor modules are connected to boxes above and below, these boxes represent I/O controllers. These I/O controllers are known as GX controllers on POWER7 and PHB controllers on POWER8. The letters GX# and PHB# are displayed in the I/O controller boxes, where # is the number of the controller. Similarly, the boxes to the side of processor modules are memory controllers (MC) as shown in Figure 3-6.

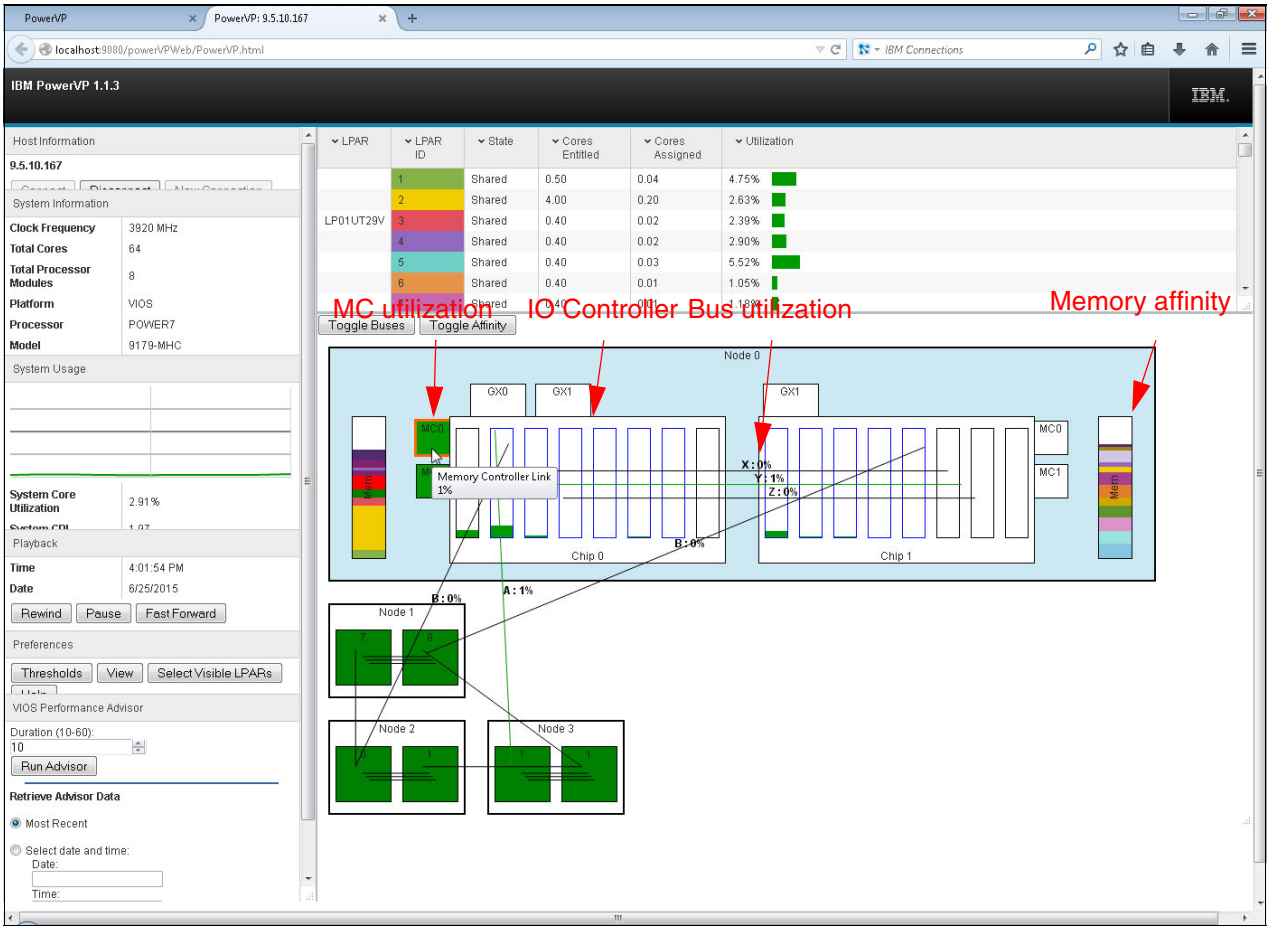


Figure 3-6 Hardware drill-down of a POWER7 node

Processor core modules and memory controllers are connected by a bus. When you hover the mouse over the memory controller, PowerVP displays the percentage of utilization of the Memory Controller Link as shown in Figure 3-6.

The boxes next to the memory controller are *memory affinity boxes*. These boxes are colored to show the percentage of memory partitions that are assigned on that memory controller.

When you hover the mouse over this box, PowerVP displays the LPAR ID and the percentage of memory assigned to the LPAR. This information is helpful when assessing the memory affinity as shown in Figure 3-7.

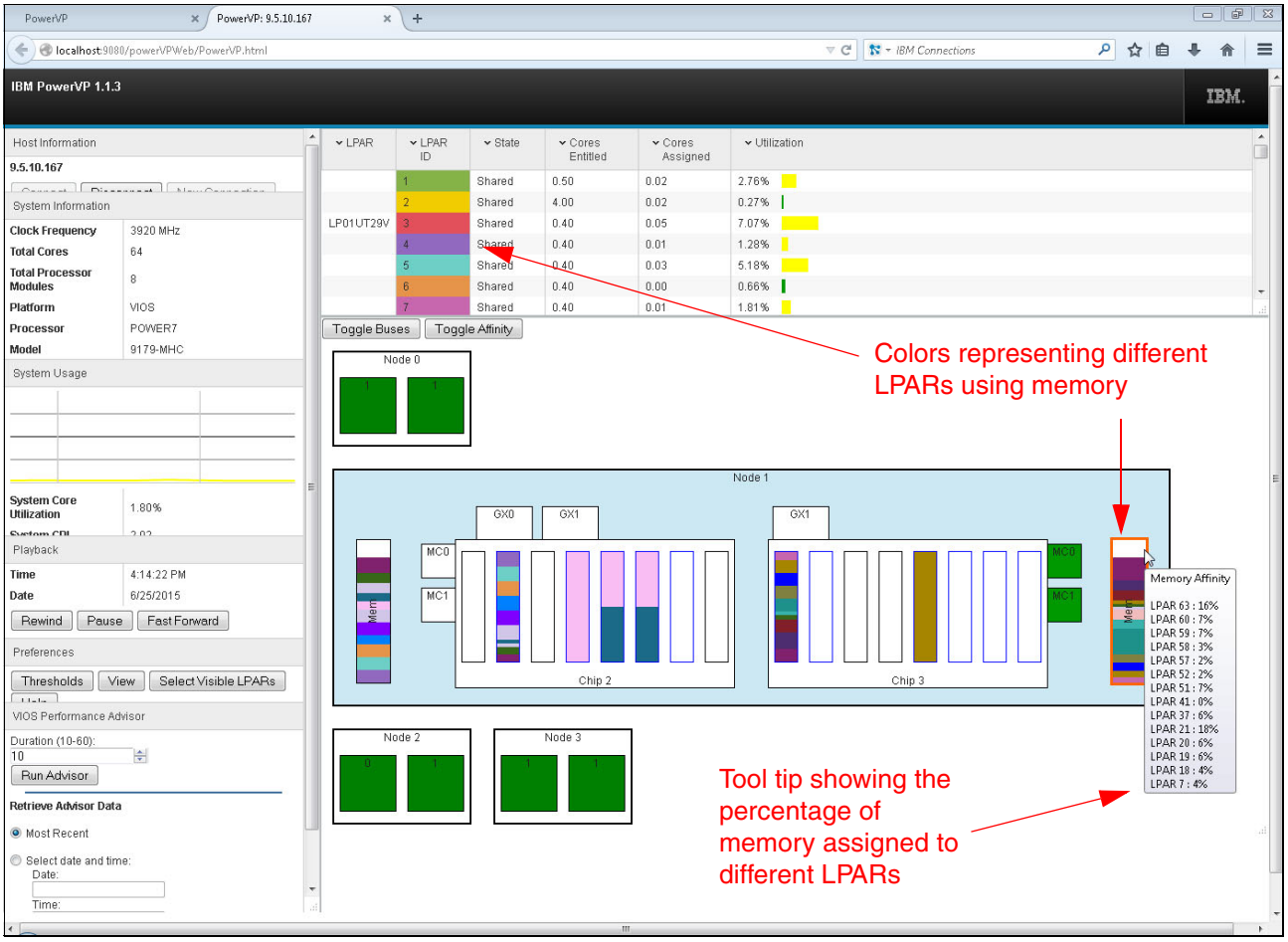


Figure 3-7 Displaying the percentage of memory allocated to partitions

PowerVP GUI can display CPU affinity information. The CPU affinity feature is enabled by selecting **Toggle Affinity**. When the CPU affinity feature is enabled, the core columns utilization information is replaced with partition affinity. The columns take the colors of the LPARs that have virtual processors dispatched on the cores. When you hover the mouse over a column, a tooltip shows the LPAR IDs and the number of virtual CPUs that are assigned to the core.

This is an easy to understand analysis of affinity of a Power System. The affinity showed by PowerVP can be confirmed by running `lsmemopt` on the HMC.

Figure 3-8 displays a system with multiple shared partitions. You can note a single processor core being the home core for 35 virtual processors. If there are processors cores not used by any partition, the cores are not highlighted in any color. This view is useful to assess the opportunity and efficiency of Dynamic Platform Optimizer.

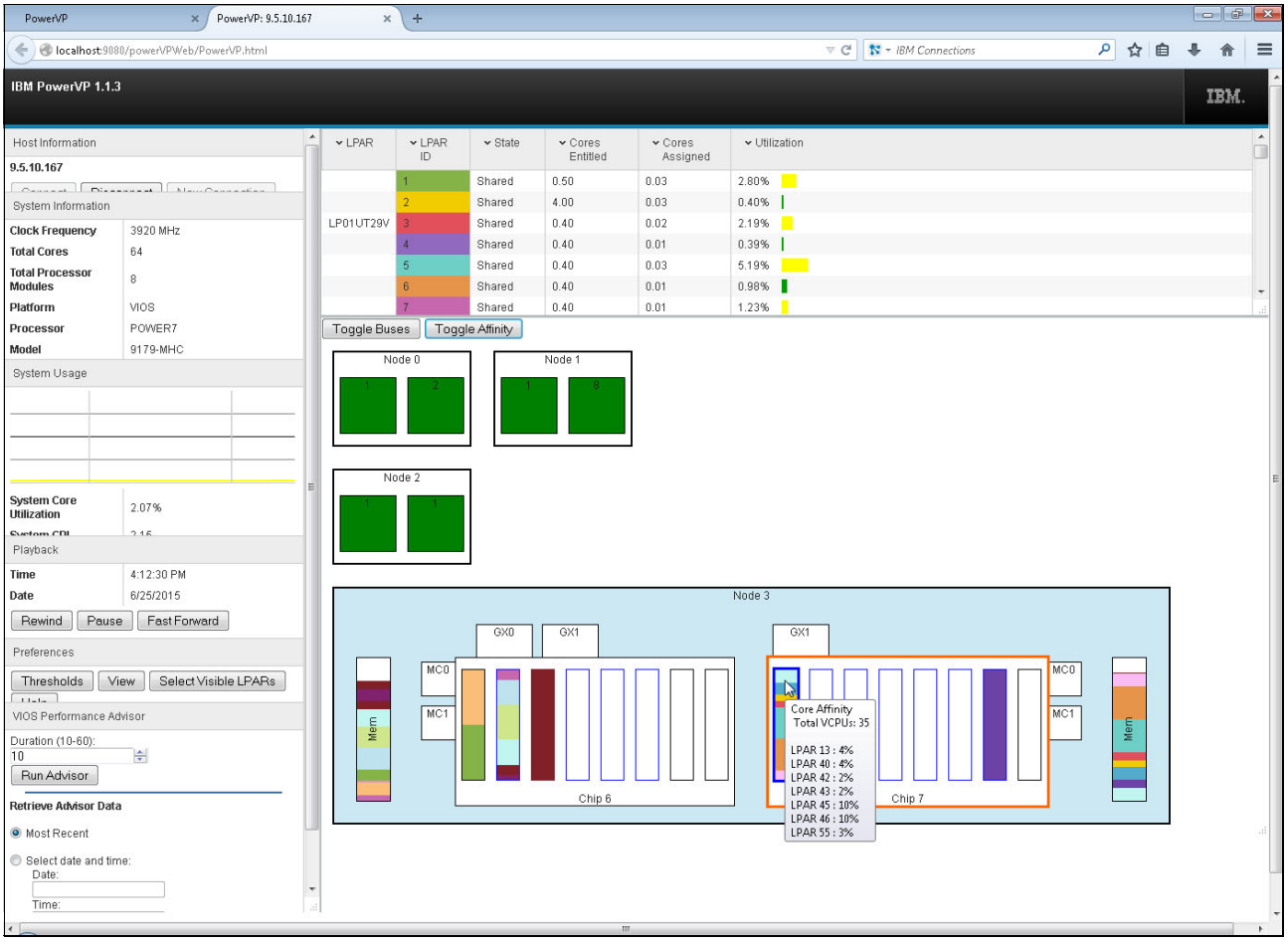


Figure 3-8 PowerVP displaying processor core affinity

Note: An easy way to know what colors are assigned to each LPAR is by clicking **Select Visible LPARs**.

Occasionally, you might notice available cores being utilized by the Hypervisor or by uncapped partitions that need extra CPU cycles. In Figure 3-9, we show how an uncapped shared partition uses computing cycles from a processor core that is not home for any virtual processor. The partition uses almost four times more CPU cycles than its entitled capacity (EC).

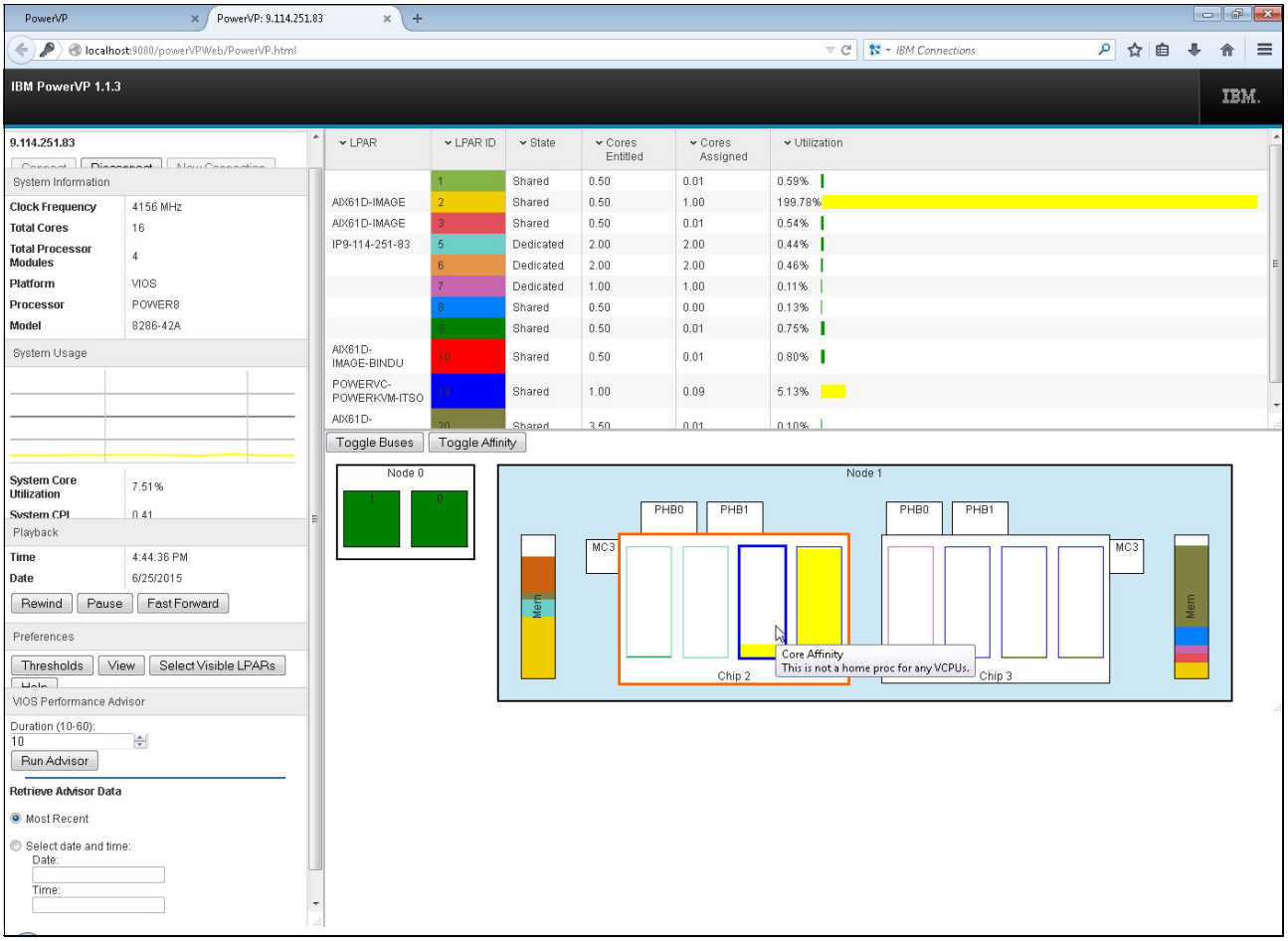


Figure 3-9 PowerVP showing a used processor core, which is not home for any vCPU

3.1.3 PowerVP partition drill-down

To get details for a partition, a Partition level agent must be installed on the partition and it must be running. After PowerVP GUI connects to the System level agent, it builds the list of partitions that are actively running the Partition level agent. This information is shown in the LPAR list in the main window.

If the **LPAR** column in the LPAR list has either the host name or IP address of the partition, PowerVP GUI can connect and provide a partition view.

The partition drill-down view can be opened by clicking the host name or IP address in partition list. The partition view opens in a new web browser window.

Figure 3-10 shows the partition view of a POWER8 LPAR. The partition view window has summary LPAR information. It gives static information of the LPAR such as host name, processor type and mode, platform, number of disk drives, and Ethernet adapters. Along with static data, it provides configured and real-time used cores, CPI, and the million instructions per second (MIPs) definition.

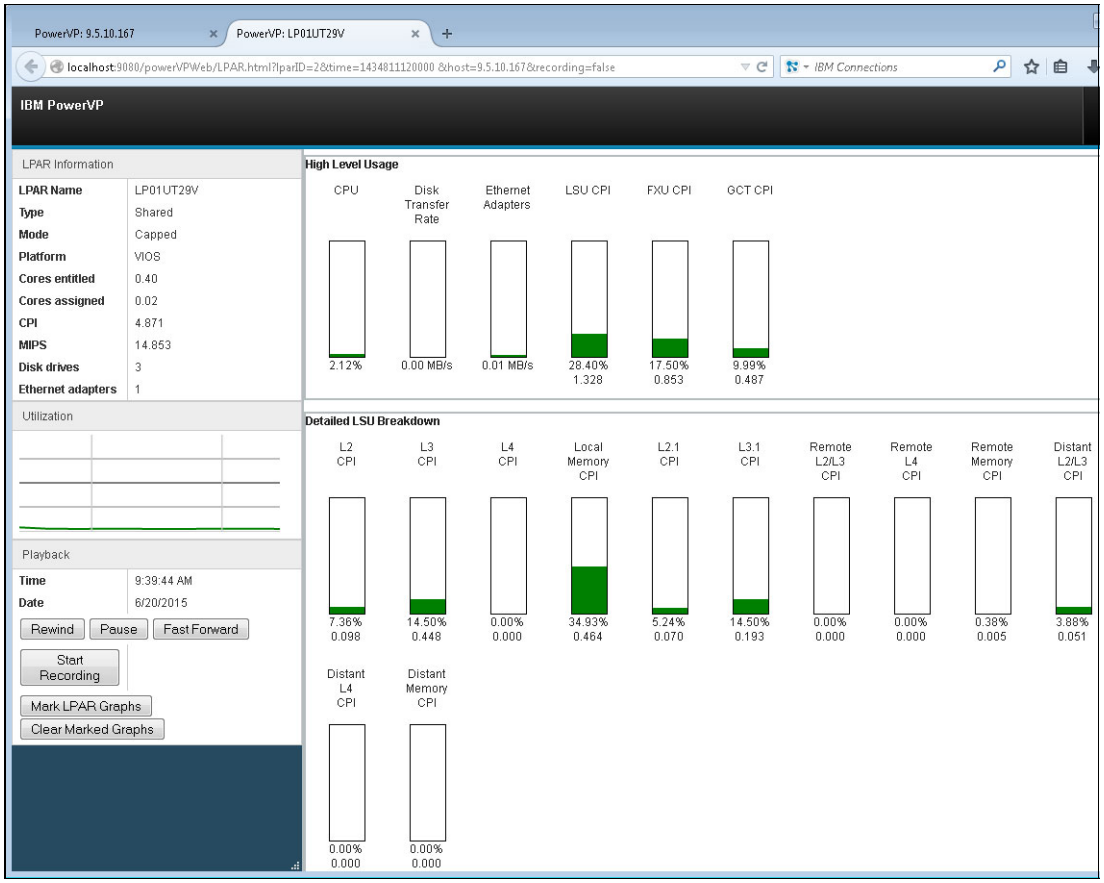


Figure 3-10 Partition View of POWER8 LPAR

In the right pane of LPAR Information are columns that are labeled CPU, Disk Transfer Rate, Ethernet Adapters, LSU CPI, FXU CPI, and GCT CPI:

- CPU

  The CPU column shows the CPU utilization of the partition. It is displayed as a percentage of entitled processors.

- Disk transfer rate

  The disk transfer rate shows the total bytes read and written to all disks on the LPAR. This is useful when working with VIO and VIO clients using a virtual Small Computer System Interface (SCSI) disk. Currently, there is no support for Fibre Channel or Virtual Fibre Channel adapters and storage area network (SAN) disks.

The Disk Breakdown pane shown in Figure 3-11, provides transfer rate information for each individual disk.

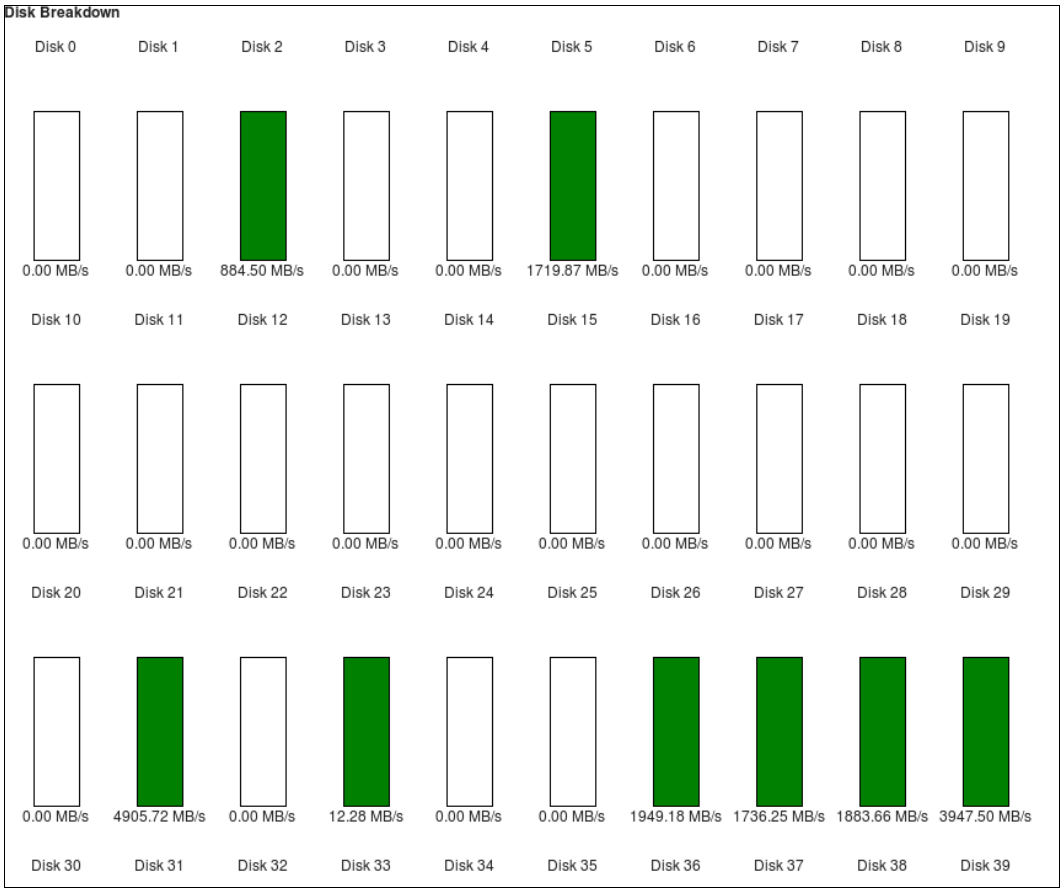


Figure 3-11 Disk transfer rate details

The disk transfer rate feature can be used to easily identify the hot disks on a partition and can be used to check the VIOS details to identify the real disk.

- Ethernet Adapters

  The Ethernet Adapters column represents that total transfer rate of the all the Ethernet adapters on the server that have an IP address configured. The column provides individual adapter transfer rates based on bytes sent and received. The data can be used to see Ethernet adapter utilization of LPARs from one central location.

- LSU CPI

  The LSU CPI column represents the total cycles spent on the Load/Store unit resources of an LPAR. More information about LSU CPI is displayed in Detailed LSU Breakdown pane as shown in Figure 3-12 on page 60.

  The LSU CPI reflects the cycles consumed for accessing data from L1 cache, L2 cache, L3 cache, and memory. In the Detailed LSU Breakdown pane, there are columns that represent the multiple memory/cache access cycle times. The L2 CPI and L3 CPI columns show the percent of utilization and the cycles that it took to access L2 and L3 cache access. The Local Memory CPI gives the percent of usage and the cycles taken to access local memory controller of the processor module. The L2.1 and L3.1 CPI columns represent cache access across the cores within the same chip. The column Remote L2/L3 CPI displays the usage and cycles spent for accessing L2/L3 cache across the chip boundaries within the same node.

The Remote Memory CPI column depicts the usage of memory across the chip boundaries within a node. The Distant L2/L3 CPI column represents the usage and cycles spend accessing the L2 and L3 cache across the node boundaries. The Distant Memory CPI column illustrates the memory usage across the node boundary and cycles spent accessing it.

Note: A POWER8 processor has an L4 cache, so the LSU detail has an extra column.

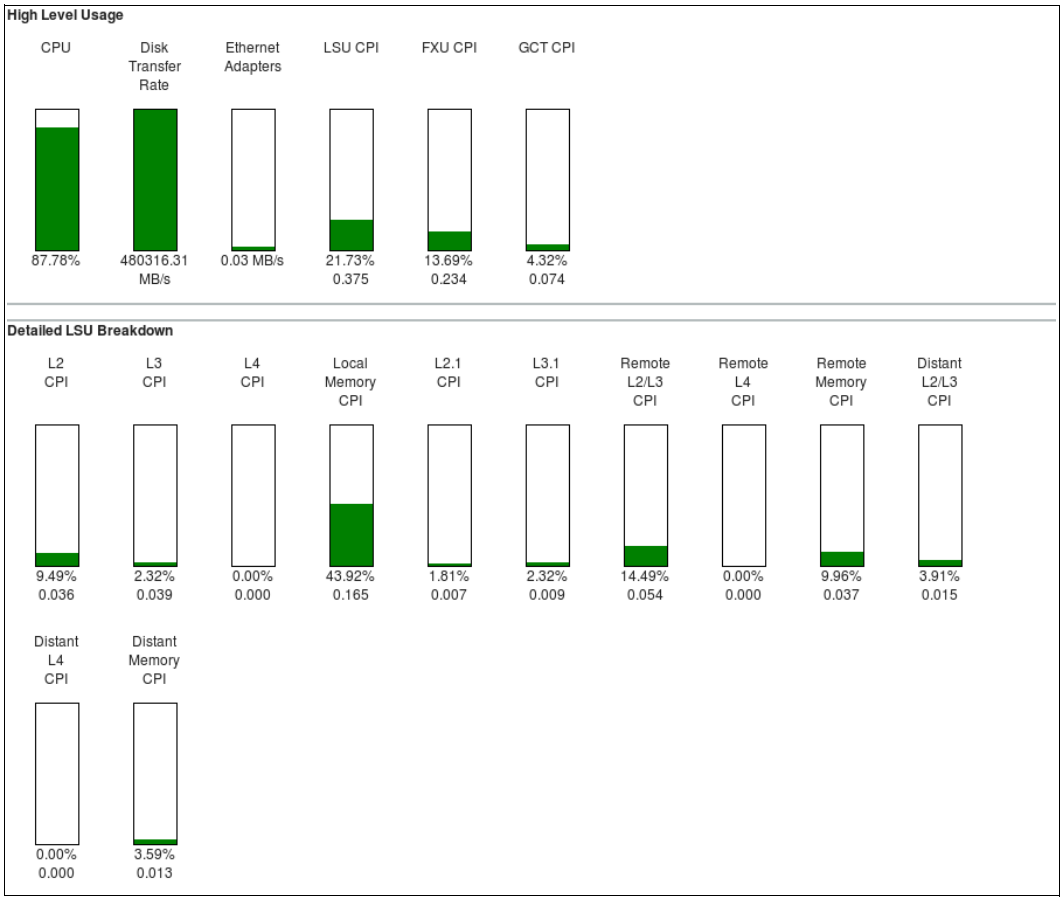


Figure 3-12 LSU CPI detail view of a POWER7 partition

- FXU CPI

  The FXU CPI represents the cycles spent on floating point execution in a partition.

- GCT CPI

  The GCT CPI represents the cycles spent waiting Global Completion Table (GCT) for the partition.

3.2 PowerVP recording and playback

PowerVP provides two different methods to record and play back performance data of a Power System. This chapter considers GUI interface. For performance data recorded by PowerVP agent, see 3.3, “PowerVP permanent data store” on page 63.

The recording can be started and stopped by a simple click of a button in the GUI. Before recording the performance metrics, you must connect from the GUI to the appropriate PowerVP system agent. This method is client side and continues when the GUI interface is running or recording is manually stopped.

When you start recording, the data is written to a file on the application server where the PowerVP GUI is running. Location depends on the initial settings and cannot be changed. Full path of recording data is:

`[path_to_PowerVP]/PowerVP_Installation/PowerVP_GUI_Installation/wlp/usr/servers/powervp/recordings/<server_name>`

PowerVP creates a folder for every LPAR and writes data in a new file for each recording session. Name convention is YYYY_MM_DD_HH_MM_SS.slt for system and YYYY_MM_DD_HH_MM_SS.prt for partition agent. Name refers to record starting date and time.

Recording of performance data is useful in debugging performance-related problems. It is useful to compare performance impact of a change when adding processors or memory to the partitions as an example. Performance improvements due to Dynamic Platform Optimizer (DPO) or impact of dynamic LPAR (DLPAR) on affinity of the system, and so on. Recording performance metrics before the change and comparing it with performance metrics after the change is an easy way to understand the impact of a change.

The PowerVP can record large amounts of data when recording is enabled. Ensure that sufficient space is available to store recorded data. We recommend increasing the sample rate from the default of 1 second. This setting can be changed by editing the `/etc/opt/ibm/powervp/powervp.conf` file and changing *SampleInterval* directive. The sample interval must be changed for the System level agent only. The Partition level agents pick up the sample interval from the System level agent. To be effective, the PowerVP agent needs to be restarted after the sample rate is changed.

PowerVP provides a feature that can record and play back the performance data of a Power System and an individual partition on the workstation where the GUI is running.

3.2.1 Recording with GUI

The first step for recording is to enable playback by clicking **Preferences** → **View** → **Playback** if playback is not already displayed in the main window. Before recording, ensure that the GUI is connected to the PowerVP system agent. If you want to record drill-down performance data for certain LPAR, you need a Partition level agent installed and connected to a System level agent. When this requirement is met, you can select a wanted LPAR name in the entitlements window pane. GUI opens a new tab, where the same playback control section is available as in the main window.

The buttons that are provided in the playback menu allow you to start, stop, pause, fast forward, and rewind, much like a DVR. In order to start recording, press **Record to Server**. Label of the button changes to **Stop recording**.

Note: If you click the Stop recording button, it stops recording and the GUI continues to show real-time data. This allows you to take several recordings without having to stop and start the GUI.

Figure 3-13 Shows the Playback section of PowerVP GUI.

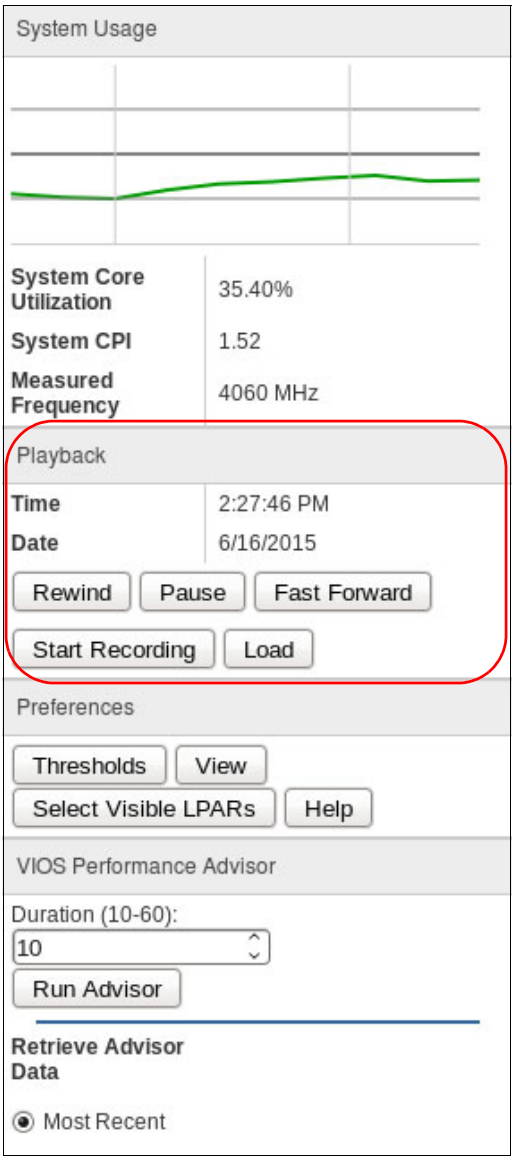


Figure 3-13 PowerVP recording using GUI

3.2.2 Playback GUI

To view a record, a new instance of PowerVP GUI must be started. If you are connected to a live system, the PowerVP GUI does not load the recorded file.

To play a recorded file, select **Load** from the Playback window pane. PowerVP opens the select menu with a list of servers and corresponding playback files. Names refer to record start date and time.

Mark the wanted file and press **Select**. If multiple partitions are recorded simultaneously, you can load all of them together and PowerVP synchronizes the files and aligns the timelines with the system level data and partition level data. The playback controls, such as fast forward, rewind, pause, and stop can be used while replaying the metrics.

Figure 3-14 shows the Playback files select menu.

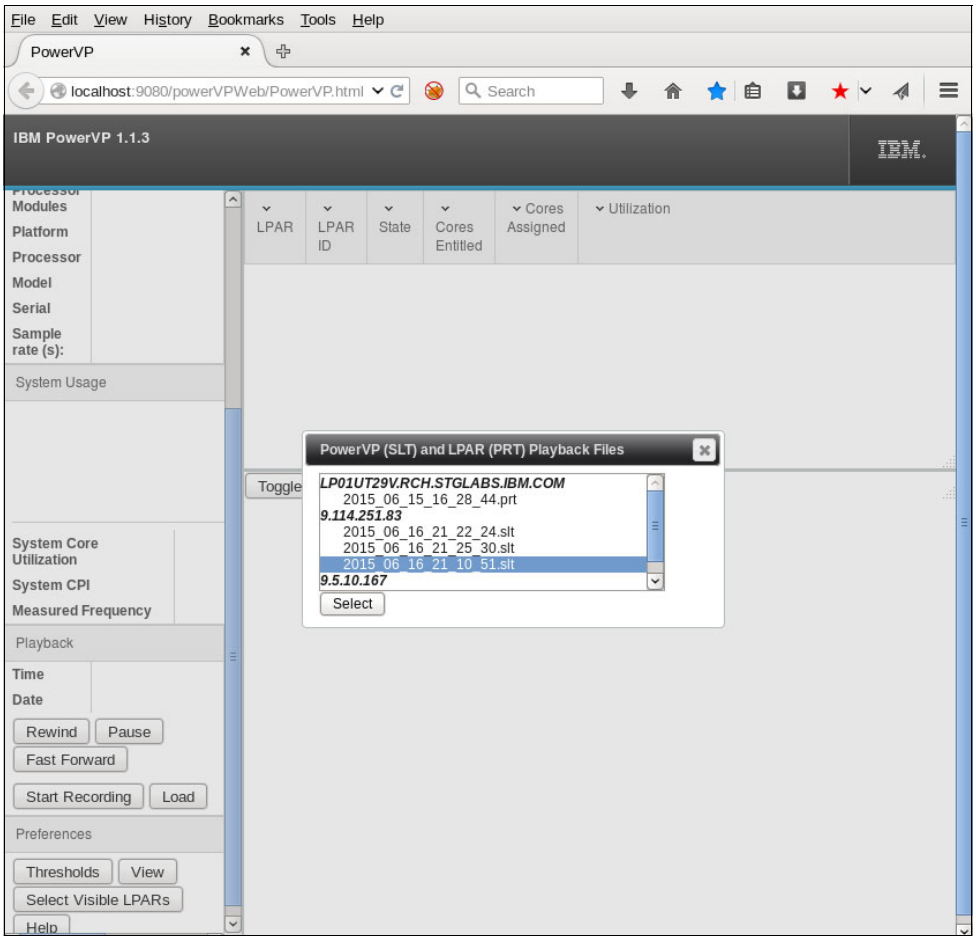


Figure 3-14 PowerVP playback files

3.3 PowerVP permanent data store

In the following sections, the data format used for recording (data store) is described.

3.3.1 PowerVP permanent data store configuration

The PowerVP version 1.1.3 provides a new enhanced method to record performance data. Command line recording is no longer supported. The PowerVP agent can be configured to write the PowerVP performance data to a file, which we refer to as the “data store”. To configure the data store, you need to change the configuration file for your PowerVP agent. When you are “recording” PowerVP performance data, PowerVP is using space on your Power system disk. The amount of space depends on the size of your Power system, the number of partitions configured on your Power system, and the sample interval configured for the PowerVP agent. Monitor the size of the PowerVP recording file to ensure that there is room for the file. You can use the LogFileRotation and LogFileArchive directives to control the PowerVP data store. You can also use the SampleInterval directive to decrease the amount of performance data recorded in the data store. The PowerVP configuration file is `/etc/opt/ibm/powervp/powervp.conf` on AIX/VIOS and Linux, and

`/QIBM/UserData/powervp/powervp.conf` on IBM i. Here are the configuration file directives associated with the PowerVP data store:

`LogData` This directs the PowerVP agent whether it is to create a log file to contain the performance data that it collects. If this directive is not provided, the default is No. If this directive is set to System, system level information only is logged to the log file. If this directive is set to Partition, partition level information only is logged to the log file. If this directive is set to Yes, both system level and partition level information is logged to the log file. If you set this to Yes or Partition, PowerVP is using the Performance Monitor Unit (PMU) to obtain information, which may interfere with other users of the PMU. If you set this to a value other than No, ensure that you also set the *LogFilePath*, *LogFileRotation*, and *LogFileArchive* directives.

Note: If LogData is enabled on partition agent, regardless of the value that is set, only partition data is logged.

`LogFilePath` The LogFilePath configuration file directive is used to specify where PowerVP is to store the data store file. This allows you complete control of where you want the file. The default location is `/opt/ibm/powervp/logs` on AIX/VIOS and Linux, and `/QIBM/UserData/powervp/logs` on IBM i. It is recommended that no other files exist in this folder. The PowerVP agent can be configured to archive old data store files. The archival efficiency is better if there are only PowerVP data store files in the folder. The file name is controlled by PowerVP and has the following format:

`PVPmmddyyyyhhmmss.csv`

where mmddyyyy is the month/day/year and hhmmss is the hour/minute/second of the first performance data in the file. The extension is CSV to reflect that the information is comma-separated, which allows it to easily be imported into spreadsheet applications.

`LogFileRotation` The *LogFileRotation* configuration file directive lets you control when PowerVP closes the current data store file and starts a new one. This allows you control over the size of the data store files. There are two ways you can do this. One is by time: the number of hours that PowerVP writes to a file before it closes it and starts a new one. The other is by the size of the data store file. To specify a time, you provide a number 1 - 24 with the letter H after it, for example, 12H would be 12 hours. To specify a file size, you provide a number followed by either the letter M (for megabytes) or G (for gigabytes). If you use a file size, PowerVP does the file rotation when the file size is close to this value. Since each line of the file is a different size, the result file sizes do not match this value. The minimum value for size is 100M. PowerVP always rotates the log file at midnight, 00:00:00 time for either rotation method.

`LogFileArchive` The *LogFileArchive* configuration file directive lets you control how long the data store files are retained on your Power server. This directive takes a single numeric value, which is the number of days to keep the data store files. At midnight, the PowerVP agent looks to see if any files need to be archived and performs the archival then. The default value is 7 days. If one of the data store files is “in use” by another application at the time that PowerVP attempts to archive it, the file is marked for deletion and is deleted when that application closes the file.

3.3.2 PowerVP Playback permanent data store records

To play this historic information from the PowerVP data store, refer to the section on starting the PowerVP GUI as shown in Figure 3-15. You provide a date and time as the start time, which instructs the agent to locate the data for that date and time and start sending it to the GUI.

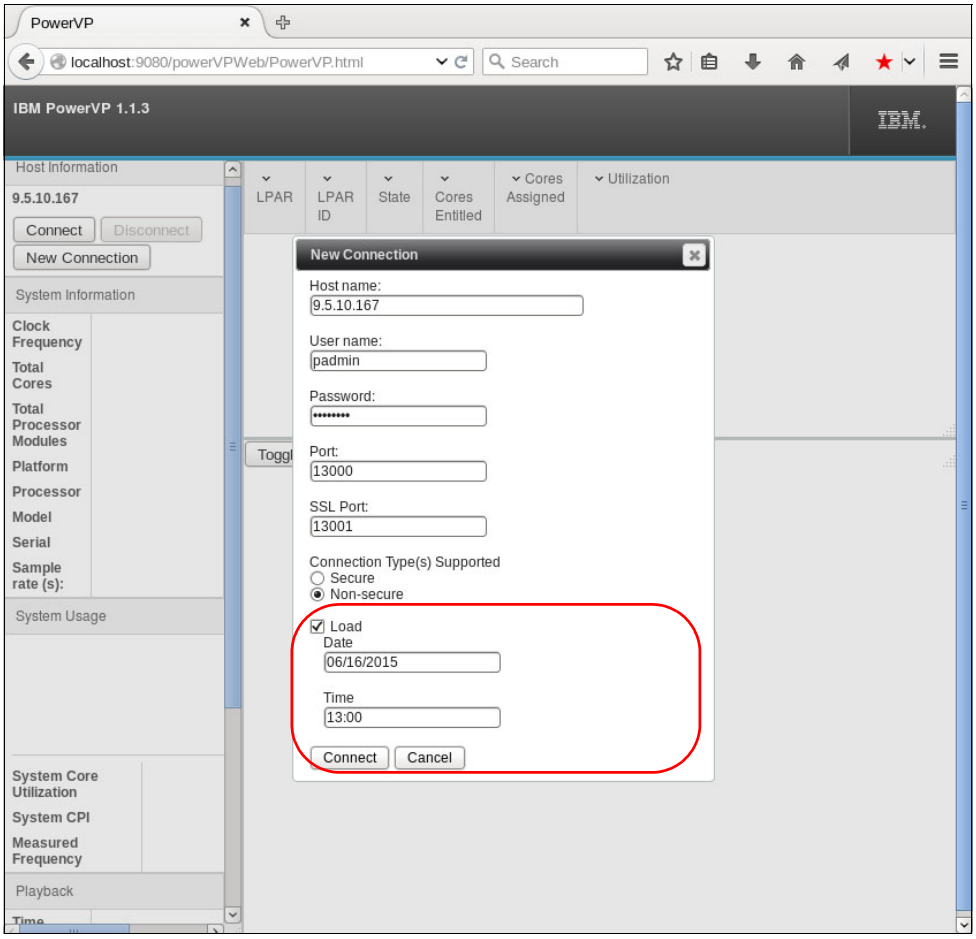


Figure 3-15 PowerVP loading data store record

Note: PowerVP GUI can be connected only to system agent. If LogData is enabled on partition agent, it can be played by GUI interface only if system agent data is available for the same time period. Otherwise, collected partition agent data can be imported into a spreadsheet and analyzed.

3.3.3 PowerVP data store format

The data store is written with unique records for each type of performance data with comma-separated values (csv). In order for the permanent data store to be consumed by other tools, it has been structured so that there are “information” entries for each different record containing performance data. These information entries are written at the start of every file. These records provide column headings for the row type. The first column of each row is the “key” for the data in that row. All rows that match in the first column (except for the

AAA rows) are the same data, but is either for a different time or different hardware processor module, hardware processor core, partition, disk, Ethernet line, or event.

There are two special row types, AAA, and ZZZZ. The AAA rows contain general information about the Power system and about PowerVP. These include the PowerVP version, operating system type, and release level for the partition the agent is running on, the sample rate, and other information about the Power system. The ZZZZ rows contain the time stamp information. The time stamp information has a time stamp identifier in the second column of format Txxxxx where x is numeric, and T00000 is used for the information row. It also has a column for the military time, date, numeric time stamp (seconds since 00:00 hours, Jan. 1, 1970, UTC), and the numeric time base register from the Power system. The time stamp identifier is used in all of the other entries to refer to the time for the data in the row.

The second column of all rows (except the time stamp rows and the information rows) is the occurrence count, an occurrence count of zero means that the rest of the row is column labels, while the rest of the rows are data snapshots with the occurrence count incremented for unique information for that time. The third column of every row (except the time stamp rows and the information rows) is the time column and is used to correlate the row to the correct time stamp row. We now describe each unique row of the file.

Time stamp entries

Time stamp definition record also has a definition row with the second column set to T00000 indicating it is the header row:

> ZZZZ, T00000, Military time in HH:MM:SS, Date in DD-MM-YYYY, time stamp in seconds elapsed since 00:00 hours on Jan 1 1970 UTC, time base register

Time stamp rows look like this. Since PowerVP rolls over the data store file at midnight every day, the largest time identifier is T86400:

- ZZZZ, T00001, 13:50:00, 25/01/2014, 1390679400
- ZZZZ, T00002, 13:50:01, 25/01/2014, 1390679401

PowerVP information entries

The following entries are included as the first entries in every file. They only exist when in the file:

- The PowerVP program row just contains PowerVP
- AAA, progname, PowerVP

The PowerVP version row contains the VRM of the PowerVP product. The first release to support the data store is V1R1M3. For every fix pack or version, we update the version so one can tell what version/mod/fixpack was used to produce the data. For example:

- AAA, Version, V1R1M3
- AAA, Version, V1R1M3 SP1

The operating system row contains the operating system for the agent. We produce the file in ASCII for all OSs including IBM i. The operating system row is one of these 4:

- AAA, operating system, AIX
- AAA, operating system, IBM i
- AAA, operating system, Linux
- AAA, operating system, VIOS

The operation system version row contains the version of the operating system that the agent was running on.

For IBM i, it looks like this:

AAA, operating system version, V7R2M0

For AIX, it looks like this:

AAA, operating system version, 7.1.0.0

For Linux, it looks like one of these:

- AAA, operating system version, RHEL 7.0
- AAA, operating system version, SLES 11 SP3
- AAA, operating system version, VIOS 2.2.3.4

The agent type row indicates if it is a System level agent or a Partition level agent. System level agents write both system and partition data to the file. Partition level agents write only partition data to the file:

- AAA, agent type, system
- AAA, agent type, partition

The TCP/IP host name row contains the host name of the TCP/IP agent. It may or may not be fully qualified depending on the OS TCP/IP configuration on the agent:

- AAA, agent host name, mysystem
- AAA, agent host name, mycompany.com

The processor version row contains the processor version of the Power system. For now, it is one of the following:

- AAA, processor version, POWER7
- AAA, processor version, POWER8

The agent sample rate row contains the sample rate in seconds:

AAA, Sample rate in seconds, 1

The Power system clock frequency row contains the clock frequency in megahertz:

AAA, Clock Frequency in MHz, 3920

The Power system timebase frequency row contains the timebase frequency in megahertz:

AAA, Timebase Frequency in MHz, 512

The Power system serial number row contains the serial number of the Power system:

AAA, System Serial Number, 10CD93T

The partition time zone information. This is the offset from GMT:

- AAA, Partition Timezone offset, 6
- AAA, Partition Timezone offset, -2

Topology

The topology entries contain the physical topology information for the Power system. There can be multiple topology entries in the file. If nodes or chips or cores are activated/deactivated, subsequent topology records are in the file containing the “time” when the change occurred.

System topology

The system topology SYSTOP record contains integers for the number of nodes, chips, cores, and virtual (logical) processors on the Power system. Each file contains information for only one Power system, so there will only be zero and one row for SYSTOP. If the hardware configuration changes while PowerVP is running, a new SYSTOP record is written to indicate the new configuration:

SYSTOP, 0, T00000, number of nodes, number of chips, number of cores, number of virtual processors

And the other rows look like this:

- SYSTOP, 1, T00001, 3, 12, 96, 384
- SYSTOP, 1, T00090, 4, 16, 128, 512

Chip topology

The chip topology CHIPTOP rows contain the information about the processor modules (chips) on the system. Each processor module has its own row with the second column starting with one and incrementing for each processor module. If the topology changes, the CHIPTOP rows are written with the time column set to indicate the time of the change. The rows contain integers for the ID columns and width columns. For the bus columns, the link-attached column contains either “Yes” or “No” if the bus has a connection. If it is “Yes”, the corresponding link endpoint node or chip ID column has the node or chip ID of the endpoint. If the bus is not connected, “No”, the endpoint ID is 0, but that does not mean that it is connected to chip or node 0. For the GX, PHB, and MC busses, the attached column is “Yes”, or “No”:

- CHIPTOP, 0, T00000, physical chip ID, hardware chip ID, fabric chip ID, hardware node ID, A (ABC) bus width, X (WXYZ) bus width, GX bus width, MC bus width, PHB bus width, A0 (A) link attached, A0 (A) link endpoint node ID, A1 (B) link attached, A1 (B) link endpoint node ID, A2 (C) link attached, A2 (C) link endpoint node ID, X0 (W) link attached, X0 (W) link endpoint chip ID, X1 (X) link attached, X1 (X) link endpoint chip ID, X2 (Y) link attached, X2 (Y) link endpoint chip ID, X3 (Z) link attached, X3 (Z) link endpoint chip ID, GX0/PHB0 bus attached, GX1/PHB1 bus attached, PHB2 bus attached, PHB3 bus attached, MC0 bus attached, MC1 bus attached, MC2 bus attached, MC3 bus attached
- CHIPTOP, 1, T00001, 0, 0, 0, 0, 2, 2, 2, 2, 2, Yes, 2, Yes, 2, No, 0, Yes, 1, Yes, 2, No, 0, No, 0, Yes, No, No, No, No, Yes, No, No
- CHIPTOP, 002, T00001, 1, 1, 1, 0, 2, 2, 2, 2, 2, Yes, 2, Yes, 0, No, 0, Yes, 1, Yes, 2, No, 0, No, 0, No, Yes, No, No, Yes, No, No, No

Core topology

The core topology CORETOP records contain the information about the processor cores on the system. A second column of 0 indicates a column heading record. Each core has its own record with the second column starting with 001 and incrementing for each core. If the topology changes, the CORETOP records are rewritten after the new CHIPTOP records with the time column set to indicate the time of the change. The records contain integers for everything except the core state, which is character and is one of the following values: “Not installed”, “Guarded off”, “Unlicensed”, “Shared”, “Borrowed”, or “Dedicated”. The assigned partition is integer 65535 if the core is not assigned to a dedicated partition or if it is in the shared group:

- CORETOP, 0, T00000, core ID, chip ID, module ID, node ID, core state, Logical Processor ID, Processor identification register, assigned partition ID, nominal frequency in MHz, current frequency in MHz, primary affinity domain, secondary affinity domain
- CORETOP, 1, T00001, 0, 0, 0, 0, Dedicated, 1, 0, 1, 4000, 4004, 0, 1

- CORETOP, 2, T00001, 1, 0, 0, 0, Shared, 4, 4, 65535, 4000, 4004, 0, 1
- CORETOP, 3, T00001, 2, 0, 0, 0, Borrowed, 8, 8, 65535, 4000, 40004, 0, 1

Registered partitions

The REGLPARS registered partition records define the partitions that have registered as Partition level agents with the System level agent. A second column of 0 indicates a column heading record. Each registered partition entry has its own record with the second column starting with 001 and incrementing for each registered partition. If a partition registers or removes itself from the partition list, the REGLPARS records are rewritten with the time column set to indicate the time of the change:

- REGLPARS, 0, T00000, partition ID, version, operating system, agent type, processor version, authentication type, host name, IP address
- REGLPARS, 1, T00001, 0, 3, AIX, partition, POWER8, system, myaix.domain.com, 9.5.11.11
- REGLPARS, 2, T00001, 1, 3, IBM i, partition, POWER8, system, myibmi.domain.com, 9.5.11.12
- REGLPARS, 3, T00001, 2, 3, Linux, partition, POWER8, system, myrhel.domain.com, 9.5.11.13
- REGLPARS, 4, T00001, 3, 3, VIOS, system, POWER8, system, myvios.domain.com, 9.5.11.14

Affinity information by domain

The AFFDTOP affinity domain topology records define the affinity domains on the system. A second column of 0 indicates a column heading record. Each affinity domain has its own record with the second column starting with 001 and incrementing for each affinity domain. If the topology changes, the AFFDTOP records are rewritten with the time column set to indicate the time of the change:

- AFFDTOP, 0, T00000, primary domain, secondary domain, total processor units, free dedicated processor units, free shared processor units, total memory, free memory, number of partitions in domain
- AFFDTOP, 1, T00001, 1, 0, 600, 100, 100, 256, 10, 13

Affinity information by partition

The AFFPTOP affinity partition topology records define the affinity by partition. A second column of 000 indicates a column heading record. Each partition has its own record with the second column starting with 001 and incrementing for each partition. If the topology changes, the AFFPTOP records are rewritten with the time column set to indicate the time of the change. The placement spread field is character and contains one of the following: “unknown”, “contained in primary domain”, “contained in secondary domain”, “spread across secondary domain”, “wherever fits”, or “scramble”. The other fields are all numeric. Each partition record also contains a number of affinity element records. The last field in the AFFPTOP record tells you how many affinity element records follow it and the partition ID field in the elements matches the partition ID in the partition record it goes with.

- AFFPTOP, 0, T00000, partition ID, assignment order, placement spread, affinity score (0-100), number of affinity domain elements
- AFFPTOP, 1, T00001, 1, 1027, 1, 90, 2

The affinity elements look like this:

- AFFPELE, 0, T00000, partition ID, primary affinity domain, secondary domain index, dedicated processor units allocated, dedicated memory allocated default, dedicated memory allocated reserved 1, dedicated memory allocated reserved 2, dedicated memory allocated 16 GB pages
- AFFPELE, 1, T00001, 1, 2, 2, 2, 600, 124, 0, 0
- AFFPELE, 1, T00001, 1, 2, 3, 2, 600, 244, 0, 0

Affinity domain information by virtual processor

The AFFVPROC affinity domain information by virtual processor records defines the domain affinity by virtual processor. A second column of 000 indicates a column heading record. Each virtual processor has its own record with the second column starting with 001 and incrementing for each virtual processor. If the topology changes, the AFFVPROC records are rewritten with the time column set to indicate the time of the change. The fields are all numeric:

- AFFVPROC, 0, T00000, partition ID, virtual processor index, physical processor index, primary affinity domain index, secondary affinity domain index
- AFFVPROC, 1, T00001, 1, 3, 4, 2, 4

System agent data

The System level agent information is retrieved from the Hypervisor for all partitions and cores on the system.

CPU utilization by core

The CPU utilization data is retrieved from the Power system Hypervisor by core ID. Each core has its own row with the second column starting with 1 and incrementing for each core:

- SCPUBC, 0, T00000, core ID, user plus kernel PURR delta, unfiltered PURR delta, run instructions delta, total run cycles delta, timebase delta for this sample, current core frequency in MHz
- SCPUBC, 1, T00001, 8, 105582, 253924, 15066481, 98522555, 180168156, 3255

Timebase cycles by core

The time base cycle utilization is retrieved by core. Each core has its own row with the second column starting with 1 and incrementing for each core:

- SCYCBC, 0, T00000, core ID, time base cycles delta, timebase delta
- SCYCBC, 1, T00001, 8, 239116358, 527434092

Bus utilization by chip

The Power bus utilization is retrieved by chip. Each chip has its own row with the second column starting with 1 and incrementing for each chip. Power 8 systems do not currently provide this information, so there are not any records in the file for Power 8:

- SBUSBCH, 0, T00000, chip ID, A0 (A) bus utilization percent, A1 (B) bus utilization percent, A2 (C) bus utilization percent, X0 (W) bus utilization percent, X1 (X) bus utilization percent, X2 (Y) bus utilization percent, X3 (Z) bus utilization percent, inbound PHB0 or GX0 bus utilization percent, inbound PHB0 or GX0 bus rate, inbound PHB1 or GX1 bus utilization percent, inbound PHB1or GX1 bus rate, inbound PHB2 bus utilization percent, inbound PHB2 bus rate, inbound PHB3 bus utilization percent, inbound PHB3 bus rate, outbound PHB0 or GX0 bus utilization percent, outbound PHB0 or GX0 bus rate, outbound PHB1 or GX1 bus utilization percent, outbound PHB1 or GX1 bus rate,

outbound PHB2 bus utilization percent, outbound PHB2 bus rate, outbound PHB3 bus utilization percent, outbound PHB3 bus rate, MC0 bus utilization percent, MC1 bus utilization percent, MC2 bus utilization percent, MC3 bus utilization percent

- SBUSBCH, 1, T00001, 5, 1, 0, 0, 4, 3, 0, 1, 65, 0, 0, 0, 0, 0, 0, 5, 389, 0, 0, 0, 0, 0, 0, 9, 8, 0, 0

CPU utilization by partition

The CPU utilization by partition consists of the entitled, capped, uncapped, donated, and idle cycles by partition. Each partition has its own row with the second column starting with 1 and incrementing for each partition:

- SCPUBP, 0, T00000, partition ID, entitled capacity, capped cycles, uncapped cycles, donated cycles, idle cycles
- SCPUBP, 1, T00001, 6, 100.00, 51.79, 41.79, 0.0, 2.71

Partition metrics

The partition metrics record consists of individual partition metrics. Each partition has its own row with the second column starting with 1 and incrementing for each partition:

- SMETRICBP, 0, T00000, partition ID, performance data version, delta timebase cycles waiting for entitlement, delta number of times waited on entitlement, delta timebase cycles waiting for a physical processor, delta number of times LPAR dispatched to run, delta home processor dispatches, delta primary affinity domain dispatches, delta secondary affinity domain dispatches, delta remote dispatches, delta dedicated donate processor dispatches, instruction count delta, timebase cycles delta
- SMETRICBP, 1, T00001, 6, 8, 300, 2, 200, 20, 18, 1, 1, 0, 0, 23245, 34323895

Partition agent data

The following records are Partition level agent records and thus are collected by all agents since the System level agents collect partition level information about themselves.

Partition status

The partition status row contains partition status information. To retrieve this information about AIX, PowerVP uses the perfstat() apis; on Linux, it uses information from the /proc/ppc64/lparcfg file; and on IBM i, it uses system APIs to get the information:

- PSTAT, 0, T00000, partition ID, dedicated or shared, capped or uncapped, donate enabled, entitled capacity, active processors in shared pool, partition name
- PSTAT, 1, T00001, 4, Shared, Uncapped, No, 1, 7, PARTITION4

Partition cycles per instruction (CPI)

The partition CPI row contains the cycles per instruction information obtained from the PMU. PowerVP collects event counts for the following groups on POWER8: pm_utilization, pm_cpi_stack2, pm_cpi_stack4, pm_cpi_stack15, pm_cpi_stack18, pm_dsource1, pm_dsource4, pm_dsource5, pm_dsource6, pm_dsource 7, and pm_dsource8; and for the following groups on POWER7: pm_dlatencies3, pm_cpi_stack1, pm_cpi_stack2, pm_cpi_stack7, pm_dsource1, pm_dsource2, pm_dsource3, pm_dsource4, pm_dsource5, pm_dsource6, pm_psource10, pm_dsource12, and pm_prefetch2:

- PCPI, 0, T00001, group name, event 1 count, event 2 count, event 3 count, event 4 count, event 5 count, event 6 count
- PCPI, 1, T00001, pm_utilization, 9790840, 8165099, 3656739, 2536237, 2536237, 8165035

Partition Ethernet throughput

The partition Ethernet throughput row contains the Ethernet send/receive counts per Ethernet interface on the partition:

- PENET, 0, T00000, interface name, kilobytes sent, kilobytes received
- PENET, 1, T00001, en0, 6256, 3425

Partition disk I/O

The partition disk I/O row contains the disk I/O per disk for the partition:

- PDISK, 000, T00000, name, kilobytes read, kilobytes written
- PDISK, 001, T00001, hdisk0, 2.57, 4.75

Partition CPU cycles

The partition CPU cycles row contains the cycles used by the partition:

- PCPU, 0, T00000, entitled cycles, capped cycles, uncapped cycles, donated cycles, idle cycles
- PCPU, 1, T00001, 100.00, 51.47, 43.27, 0.0, 2.71

3.4 Customizing the PowerVP GUI

The PowerVP GUI provides an easy way to customize the thresholds to suit individual requirements. It allows users to customize the GUI to show a selected subset of data. There are two types of threshold customizations:

- Core thresholds

  The core threshold customization allows a user to define the thresholds and the colors. A user can choose the colors for each threshold. By default, the PowerVP GUI is configured for four different thresholds: 100, 95, 70, and 0.1. PowerVP GUI provides a way to define thresholds greater than 100. This feature is useful when dealing with uncapped micro partitions. To edit the thresholds, go to **Preferences** pane and click **Thresholds**, then select **Core Thresholds** in the Select Threshold Type menu.

- Bus thresholds

  This setting allows the user to set thresholds on the buses. The PowerVP GUI can customize each bus threshold separately. Four different types of buses can be customized by going to select the type of bus in the **Select Threshold Type** menu:

AB or **A**	Represents the buses between the nodes
WXY or **X**	Represents the buses between processor modules
GX or **PHB**	Represents the I/O controller buses
MC	Represents the memory controller buses

Figure 3-16 shows the window to define the core thresholds and colors. The bus threshold customization window is similar to the core threshold window. To add a threshold, click **Add** and a new row gets added to the window at the top. The threshold is defined in the left box. Click the right box to choose the threshold color. To delete the threshold, click **X**. Click **Default** to reset the thresholds to the installation defaults.

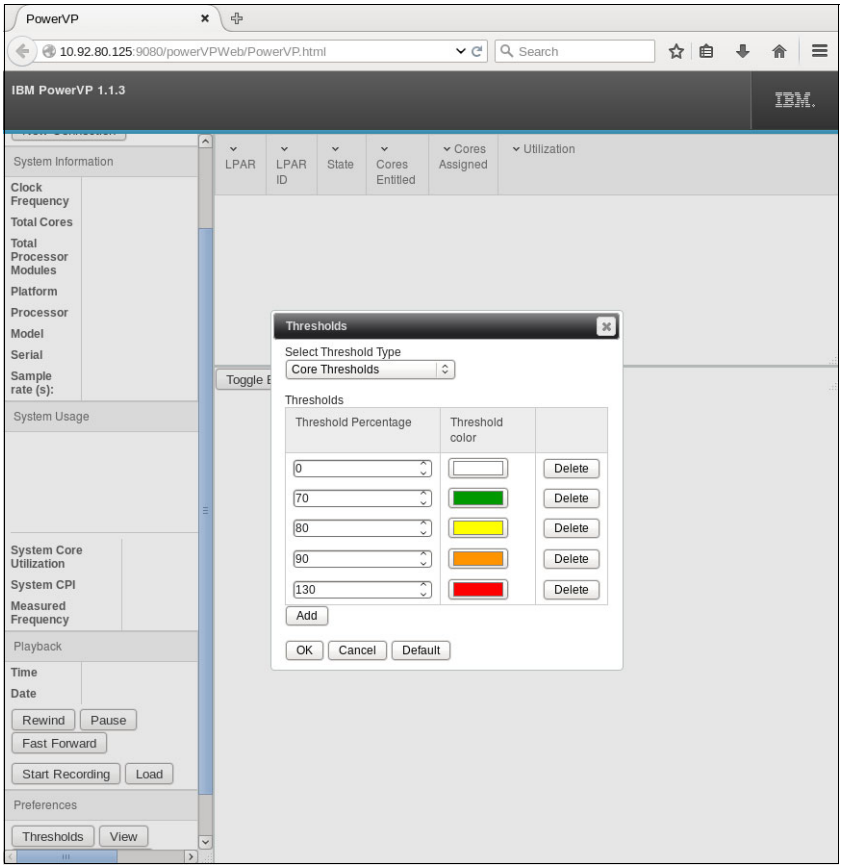


Figure 3-16 PowerVP GUI core threshold window

- Visible partitions

  The PowerVP GUI gives an option of selecting the visible partitions in the partition list. Go to the **Preferences** window pane and click **Select Visible LPARs** to select the partitions that you want to see. Use the Ctrl key to select multiple partitions. This feature is useful when there are many LPARs on a system.

Using the View menu, the main PowerVP GUI can be configured to display a customized view of the main window. The **View** menu acts as a simple toggle to turn on and off individual features of the main window.

4

PowerVP and PowerVM features

IBM PowerVM introduces new features that can be used in the deployment of IBM Power Systems. This chapter describes how IBM Power Virtualization Performance (PowerVP) can be used in typical IBM Power Systems environments for planning, implementing, tuning, and investigating performance issues.

PowerVP has an intuitive graphical interface that is easy to use and understand. It allows fast access to system hardware details and important partition data, including the number of virtual processors or entitled capacity.

PowerVP is ideal to quickly identify performance problems, both at system and partition level. The benefit of using PowerVP GUI is that it enables faster troubleshooting of performance issues with PowerVP than traditional command-line and HMC session techniques.

Although every operating system has its own performance tools that can provide performance details at the partition level, PowerVP is able to offer a system-wide performance view. PowerVP is intended to complement rather than replace OS-specific performance tools.

After an issue is identified with PowerVP, more investigations can be done using other performance-specific tools and commands.

This chapter includes the following sections:

- General hardware layout of IBM Power Systems
- Use cases of PowerVP

4.1 General hardware layout of IBM Power Systems

The architecture of a Power Systems server is composed of processor cores, processor chips, memory controllers, memory modules, and various levels of cache units. All these components are connected by buses. There are different types of buses, each type of which has a different width and frequency. The number of connections among all processor cores and chips grows quickly as the number of processor cores increases.

A single processor chip contains multiple cores. Depending on the configuration, a POWER7 processor chip can contain four, six, or eight processor cores with on-chip L1, L2, and L3 caches. POWER8 processor chips can contain 6 or 12 processor cores.

Each processor chip has at least one memory controller attached that allows direct access to a specific number of memory dual inline memory modules (DIMMs). Every core on every processor chip has access to *all* memory in the system, even if the memory is directly attached to a different processor chip.

Each processor chip has multiple L2 and L3 (or L4) cache units that provide internal cores fast access to data. Every processor core on every processor chip has access to *all* cache units, even if they are on different processor chips.

Important: Access to memory and cache units is subject to access rights imposed by the Hypervisor, and restrictions and limitations imposed by the operating system.

The following list summarizes a typical hierarchy of these components:

- Multiple CPUs or cores are contained in a processor chip.
- Multiple chips are contained in a module. Depending on the system model, modules can be Single Chip Module (SCM) or Dual Chip Module (DCM).
- Multiple modules are contained in a processor module.
- Depending on the system model, multiple process modules can be included in a Central Electronics Complex (CEC) or a Processor Unit Book (PU Book).
- Large systems can have multiple CECs or PU Books.

Full hardware details for all IBM Power Systems are available at the following site:

`http://www.ibm.com/systems/power/index.html`

4.1.1 Simultaneous multithreading

Multithreading represents the ability of a physical processor core to dispatch *simultaneous* instructions from more than one hardware thread. The existence of multiple registers in the physical processor core allows fetching data, loading, running, and unloading multiple instructions in the same time. Each hardware thread is essentially equivalent to an individual processor.

Processing multiple hardware threads in the same time directly benefits the overall throughput. Threads are represented at operating system level as logical processors.

POWER7 processor cores support running in single-threaded mode (SMT1) with one hardware thread, SMT2 mode with two hardware threads, or SMT4 mode with four hardware threads.

POWER8 processor cores support SMT8, which allows simultaneous processing of eight hardware threads.

With AIX, SMT is enabled by default. It can be disabled and enabled dynamically by using the **smtctl** command. Additionally, the number of simultaneous threads can be changed dynamically.

4.1.2 Memory affinity

Each processor chip has its own memory controller that allows direct access to a specific number of memory DIMMS. Depending on the system model, IBM Power Systems can have multiple levels and types of memory. The design of IBM Power Systems is based on nonuniform memory access (NUMA).

All processor cores have access to the memory that is directly attached to the chip *and* to the memory attached to *all other processor chips* on the system.

When a thread is dispatched on an individual processor core, the core is regarded as the *home* core for that thread. The Hypervisor tries to allocate all data required by the thread to the memory DIMMs directly attached to the home core. A thread that accesses data on DIMMs that are directly attached to the home core is said to have *memory affinity*.

During its lifetime, a thread can be dispatched to different cores and accesses data in memory DIMMs directly attached to the home core chip, another chip in the same central processor complex, or another chip on a different central processor complex. However, it takes longer for a process dispatched to a core to access the memory attached to a distant processor chip rather than the memory attached to the same processor chip.

Based on the distance between the home core and the location of the memory access, memory can be placed into these categories:

Local Memory is on the same chip as the home core. The data exchanged between the home core and memory uses a bus that is local to the chip. This is the optimum case.

Near Memory is on a different chip than the home core, but in the same CEC (or PU Book). The data exchanged between the home core and memory uses a bus that connects *different chips on the same CEC*. This type of access has a higher latency than local access.

Far Memory is on a *different CEC or PU Book* than the home core. The data exchanged between the home core and memory uses multiple buses that connect different chips on different CECs. This type of access has a higher latency than near access.

Because any processor core on any chip can access the memory of the entire system, when the Hypervisor dispatches threads it tries to achieve the highest possible degree of memory affinity.

4.1.3 Cache affinity

Execution time to run an individual workload depends on several factors, which include the following:

Number of instructions

This is the total number of instructions that must be executed to complete the workload.

Number of cycles per instruction

This is the number of clock cycles required to complete an individual instruction.

Clock frequency This is the time taken by each clock cycle.

The number of clock cycles required to run an individual instruction on a processor core is lower if the data used by the instruction is already available in the proximity of the core. Accessing data that is in remote cache units or memory takes more CPU cycles that accessing data in cache units or memory that is directly attached to the processor core.

The optimum number of cycles per instruction implies to have both instruction code and data available in the cache and to write all output data to the cache. In such cases, the workload runs in the lowest possible number of cycles.

Each processor core has its own cache units that are shared among all hardware threads that are dispatched to the same processor core. This cache stores data referenced by multiple threads and provides fast access to it.

Similarly, each processor chip has its own cache units that are *shared* among all processor cores that are contained in the same processor chip. This cache stores data that can be referenced by multiple threads running on any of the processor cores in the processor chip.

Depending on the system model, IBM Power Systems can have multiple levels and types of cache. The design of IBM Power Systems is based on Non-Uniform Cache Access (NUCA). Different characteristics such as capacity, associativity, bandwidth, and interconnection define the *cache geometry*.

With POWER7 processor-based systems, each processor chip contains eight processor cores and each processor core has its own Level 1 i-cache, L1 d-cache, L2 cache, and L3 cache region. L2 cache has a higher latency than L1, and L3 cache has a higher latency than L2. POWER8 processor-based systems have an extra off-chip L4 cache.

Any processor core can read and write the content of memory, L2, and L3 cache associated to any other processor core, even if on a different chip. However, it takes longer for an application thread to access data in a cache unit attached to a remote chip than to access data in the local cache.

For any processor core, the access to directly attached memory is *typically faster* than L2 or L3 cache of any other processor core in the same chip. For example, when a processor core requires access to L3 cache of another processor core, the data is copied from the distant L3 cache to the local L3 cache and then to local L2 cache.

4.1.4 Partition affinity

A partition with the ideal placement of all of its resources, including memory and processor cores, is said to have an optimal *partition affinity*.

Based on how much the resources allocated to a partition are spread across the entire system, licensed programs such as Dynamic Platform Optimizer can compute a partition *affinity score*. The affinity score is a numerical value that ranges 0 - 100, which depicts the current degree of affinity for a partition.

4.1.5 Thread affinity

Thread migration affinity concerns migration of a thread by its L2/L3 cache. When the thread is migrated to the same core, cache content is not moved at all. This is an ideal situation.

However, during its lifetime an individual thread can be dispatched on a processor core different from the home core. This process is referred to as *thread migration*. The thread can be dispatched on a processor core that may reside in a different processor chip or even in a different CEC. In this case, the thread is now distant from its L2/L3 cache. Access to the L2/L3 cache has now a higher latency. To avoid this latency and still benefit from cache affinity the content of the L2/L3 cache attached to the home core is transferred to the L2/L3 cache attached to the new processor core.

Depending on the partition configuration thread migration can also determine access to near and even far memory.

Copy operations can cause an increase of LSU CPI (load store unit cycle per instructions). Detailed information regarding LSU CPI can be observed in PowerVP GUI partition deep-drill mode view.

Note: When you monitor detailed LSU Breakdown with PowerVP, even on a partition with a good affinity score, spikes of usage of not local cache and memory can be observed. These can be caused by thread migrations.

In terms of affinity, a thread can be migrated to four different affinity domains: To the same core (S0rd/S1rd), to a core on the same chip (S3rd), to core on a different chip within the same CEC (S4rd), or to a core on a chip that is on a different CEC (S5rd).

Thread migration can be outside or within the operating system. An example of outside thread migration is when a virtual processor is not mapped to its home core, but to a different core. This is often the case with uncapped partitions when CPU utilization exceeds the entitled capacity.

AIX has the **mpstat** command, by which thread migrations can be monitored as shown in Example 4-1. This is a statistic for a 1-second interval. CPU column stands for logical CPU (one logical CPU per hardware thread unless simultaneous multithreading mode is disabled) numbers.

Example 4-1 Thread migration monitoring within AIX using mpstat command

```
# mpstat -d 1 1

System configuration: lcpu=24 ent=1.5 mode=Uncapped

cpu   .. S0rd  S1rd  S2rd  S3rd  S4rd  S5rd   ilcs   vlcs S3hrd S4hrd S5hrd  %nsp
  0   .. 98.1   0.5   0.0   1.4   0.0   0.0     11    115 100.0   0.0   0.0   101
  1   .. 82.2  17.8   0.0   0.0   0.0   0.0      4     62 100.0   0.0   0.0   101
  2   ..    -     -     -     -     -     -      1     62     -     -     -   101
  3   ..    -     -     -     -     -     -      0     62     -     -     -   101
  4   .. 73.7   4.6   0.0  21.7   0.0   0.0     12     11 100.0   0.0   0.0   101
  5   .. 82.8  17.2   0.0   0.0   0.0   0.0      7     11 100.0   0.0   0.0   101
  6   ..    -     -     -     -     -     -      1     11     -     -     -   101
  7   ..    -     -     -     -     -     -      0     11     -     -     -   101
  8   .. 75.5   6.8   0.0  17.7   0.0   0.0      5     15 100.0   0.0   0.0   101
  9   .. 85.5  14.5   0.0   0.0   0.0   0.0      3     12 100.0   0.0   0.0   101
 10   ..    -     -     -     -     -     -      0     11     -     -     -   101
```

```
 11   ..    -     -     -     -     -     -      1     10     -     -     -   101
 20   ..    -     -     -     -     -     -      1     64     -     -     -   101
ALL   .. 93.6   2.9   0.0   3.5   0.0   0.0     46    457 100.0   0.0   0.0   101
#
```

AIX has the following technologies to maximize thread affinity:

- Enhanced affinity services are used by the AIX dispatcher to keep threads within the same POWER7 multi core-chip. This technique optimizes the work of chip and memory controller.
- ASO and DSO are additional technologies that optimize partitions dynamically. They can improve core, cache, and memory efficiency by moving threads and their memory pages as well as memory pre-fetch algorithms.

4.1.6 Virtual processors

With traditional systems, the operating systems have direct access to the physical processors. Virtual processors represent a PowerVM virtualization feature that introduces an abstract layer of virtualization between the operating systems running in partitions and physical cores. Physical cores are not exposed directly to partitions. Instead, the Hypervisor virtualizes the physical cores and presents processing resources available on physical cores as virtual processors (VPs) to guest operating systems.

The operating system perceives only the virtual processors and assumes that it has entire physical cores available. Processes that are run by the operating system are assigned to a virtual processor. Virtual processors are dispatched by the PowerVM Hypervisor to the actual physical cores.

For dedicated partitions, the Hypervisor dispatches a virtual processor to a whole physical core.

For micro-partitions, the Hypervisor usually dispatches a virtual processor as a time slice of a physical core. If a partition has multiple virtual processors, they might or might *not* be dispatched simultaneously to the physical cores.

Every virtual processor has a *home core*. The PowerVM Hypervisor attempts to dispatch the virtual processor on the home core to take advantage of partition affinity. If a virtual processor is dispatched on the same processor core, it is more likely to benefit from the cache content. However, there may be cases in which a virtual processor is dispatched on a different processor core. In such cases, the content of the L2 and L3 cache home core has to be transferred to the L2 and L3 cache of the new processor core. Dispatching a virtual processor on a processor core different from the home core is more likely to happen for uncapped LPARs that use more than entitled capacity.

Note: We suggest configuring partitions so that every processor core should be home for as few virtual processors as possible.

4.1.7 Virtual processor folding

When using shared partitions on IBM POWER7 Systems, it is important to understand how workload is assigned and distributed to VPs.

When a process is started, it is initially assigned to the first thread of the first VP. As the workload increases, an extra VP is unfolded and workload is assigned again to the first

thread. This process continues until all VPs assigned to the partition are unfolded and the maximum number of VPs is reached. This approach ensures the fastest possible response time while maintaining the overall LPAR throughput.

If the workload continues to increase even more than the allocated virtual processors, the Hypervisor dynamically switches the processor cores to SMT2 and the workload is assigned to the second thread on each VP. If the workload continues to increase, SMT4 is dynamically activated and the process continues. When all threads are activated and the workload continues to increase, it is then distributed on a time-sharing basis.

When not being used, virtual processors are *not* removed from the partition. They are just disabled, and reactivated when more workload is available.

With AIX, the default folding and unfolding settings can be changed by using the **schedo** command.

Note: We suggest configuring the number of virtual processors that are assigned to partitions to avoid processor folding as much as possible.

4.1.8 Entitled capacity

Entitled capacity (EC) is the amount of time a partition is *guaranteed* to be dispatched on physical cores. Partition virtual processors are dispatched on physical cores so the partition receives the processor cycles determined by partition entitlement. A partition cannot be preempted while it is using its entitled cycles. When a partition does not have any workload to run, it might not use all its EC and *voluntarily* cede CPU cycles.

Note: We suggest configuring the number of VPs and EC for partitions to avoid ceding CPU cycles as much as possible.

4.1.9 Logical context switches

A *logical context switch* occurs when the Power Hypervisor undispatches a virtual processor from a processor core and dispatches another virtual processor on the processor core. Virtual processors might or might *not* belong to the same partition.

An *involuntary* logical context switch occurs when a virtual processor dispatched on a processor core is running a workload but is interrupted by the Power Hypervisor and preempted from the processor core.

The number of involuntary logical context switches is typically lower when the partition uses less than its EC.

The number of involuntary logical context switches is typically higher when multiple partitions attempt to use more than their EC and compete for extra CPU cycles.

A *voluntary* logical context switch occurs when a virtual processor dispatched on a processor core does not have any workload to run and voluntarily decides to cede the processor core. It notifies the Power Hypervisor, which undispatches the virtual processor from the processor core and dispatches another virtual processor.

Both cases of logical context switch affect cache efficiency. Cache is updated every time that a partition is dispatched, and it needs the partition to run longer to achieve a better efficiency.

Unnecessary loading and unloading of a partition context multiple times during a Hypervisor cycle reduces cache efficiency.

4.1.10 Resource allocation and distribution by the Hypervisor

With PowerVM, resource allocation for both dedicated and shared partitions is done by the Hypervisor. The goal of the Hypervisor is to ensure that all partitions receive the resources they are entitled to, and to achieve an optimum degree of affinity.

The Hypervisor controls and maintains the allocation of all system resources. When a partition is started, the Hypervisor considers the spreading of all available resources, and allocates the partition the requested number of cores and memory so that it achieves the highest possible affinity. The Hypervisor tries to maintain and improve affinity during all operations that can affect partitions resources during their lifetime, such as a dynamic LPAR or Live Partition Mobility.

However, it is normal that some partitions will have not only local memory, but also near or even far memory. This is more common on large systems that have multiple MCMs and multiple partitions.

Moreover, memory and processor allocation can become suboptimal over time as a consequence of the following operations:

- Dynamic LPAR (DLPAR) add and remove of processors or memory
- Live Partition Mobility (LPM)
- Hibernation suspend or resume
- Dynamic creation or deletion of partitions
- Configuration changes of processor or memory in the partition profile
- CEC Hot Add & Repair Maintenance (CHARM) node repair or add
- Order in which LPARs are powered on and off

Important: The Hypervisor assigns resources to partitions and attempts to achieve the best possible partition affinity. The order in which you start partitions affects partition affinity. We suggest starting the most important partitions first.

4.2 Use cases of PowerVP

This section shows how PowerVP can be used to investigate configuration and performance issues on IBM Power Systems.

4.2.1 Using PowerVP to assess dedicated LPARs

When using dedicated partitions, it is important to understand how memory and processor cores are allocated to the partition.

For partitions that require an amount of CPU and memory smaller than available on a processor chip, the Hypervisor tries to allocate cores from a single processor chip and memory that is local to the chip. These partitions have a good cache and memory affinity.

However, there are environments with multiple dedicated partitions in which the number of dedicated LPARs exceeds the number of processor chips. On these systems, it is not possible to allocate each partition cores from the same processor chip.

Depending on the amount of memory required by a partition, there might not be enough local memory available. The partition then receives near or even far memory. This is a typical case for dedicated LPARs having more memory than available in a single CEC or PU Book.

For larger partitions that require an amount of CPU higher than what is available on a single processor chip, the Hypervisor must allocate cores spread over more than one processor chip. The processor chips might or might not be in the same CEC or PU Book. In this case, the partition has a mixture of local, near, and potentially far memory. This is a typical case on large systems that have multiple MCMs and LPARs.

4.2.2 Using PowerVP to assess shared LPARs

When using shared LPARs, partition parameters must be configured properly. Partition parameters such as EC, number of VPs, and processor weight must be configured according to the expected workload.

Often, the system administrator defines a shared processor pool to group multiple partitions that are not business critical. The shared processor pool accommodates many shared LPARs, each having a small entitlement but several VPs.

Partition entitlement is the amount of CPU that a partition is *guaranteed* to receive so that each LPAR receives its share of CPU. The partition is also expected to handle occasional peak workloads and be able to receive extra CPU cycles.

To set up partition entitlement, complete these steps:

- Mark the partition as uncapped.
- Define a processor weight.
- Assign a number of VPs that ensure the partition gets all CPU cycles that are required to cope with the worst case scenario.

If EC and number of VPs are not planned carefully, you might encounter one of the following undesirable scenarios:

- Some partitions are predominantly idle and use less than the EC. The Hypervisor will unnecessarily dispatch and undispatch them, and all corresponding CPU cycles are wasted.
- Some partitions constantly use more than the EC. In the shared processor pool there might be unused capacity as a result of the following circumstances:
  - Unassigned cores
  - Dedicated donating partitions
  - Unused cycles from entitled partitions
  - Capacity on demanded cores
  - All partitions in the pool have used their entitlement

  Because there is spare capacity, all partitions that are uncapped compete for the extra CPU cycles. The Hypervisor dispatches the competing partitions according to their weights. The context of an individual partition might or might not be dispatched on the core used for the entitlement. Data that is requested by the partition might not be available in the cache, and therefore requires near or even far memory access, depending on the cores assigned to the shared processor pool.

- Some partitions might have an EC such as 1.01. The benefit of the additional 1% of EC must be carefully considered. The partition is dispatched on multiple cores and does not use L2 cache, with the result that the partition might not benefit from the extra 1% of EC.
- Partitions can be dispatched several times during the Hypervisor cycle to receive the guaranteed entitlement. If a partition context is dispatched on a different core, it no longer benefits from cache and memory affinity. The same thing can happen even if the partition is dispatched on the same core because other partitions have been dispatched before and updated the data in cache.

To avoid all these effects, perform these tasks:

- Monitor the performance of each partition in attempt to find the adequate EC.

  If a partition constantly uses more than its entitlement, extra CPU cycles are unnecessarily wasted just to dispatch and undispatch the partition. The partition might not be dispatched on the same core and not benefit from memory affinity.

  Cache is updated every time that a partition is dispatched, and needs the partition to run longer to achieve a better efficiency. Unnecessary loading and unloading of a partition context multiple times during a Hypervisor cycle reduces cache efficiency.

  The number of wasted CPU cycle increases with the number of uncapped partitions that compete for extra CPU cycles.

  If a partition constantly uses less than its entitlement, CPU cycles are wasted when the partition is dispatched to get its entitlement.
- Assign a reasonable number of VPs.

  The number of VPs must not exceed the number of processor cores that are available on the system or maximum value specified for the shared processor pool if the partition runs in a virtual shared processor pool.

  Having too many virtual processors does *not* provide any performance improvement and might cause more context switches that actually reduce the performance.

  The ratio between VPs and processor cores must be as low as possible. The higher this ratio is, the less CPU spare cycles are available for uncapped partitions.
- If possible, avoid using large shared processor pools that might spread across multiple chips.

4.2.3 Quick use case of using PowerVP

Figure 4-1 on page 85 shows how to use PowerVP to assess resource distribution and partition affinity for both dedicated and shared partitions.

Based on the data provided by the PowerVP GUI, you can make the following determinations:

- Both LPARs `21` and `22` are assigned to the same processor core on `Chip2`. Both partitions are shared. They are dispatched and undispatched unnecessarily on the same processor core even if there are other processor cores available. This is a suboptimal case of resource distribution.
- LPARs 5, 6, and 7 are dedicated partitions. The number of entitled cores equals the number of assigned cores even if they have a small utilization rate. You can conclude these partitions are not donating.

- LPAR 22 is a shared partition and has an entitlement of 0.1 cores. Utilization rate is 638.41%, which is abnormally high. The partition uses 1.07 cores. Because the partition uses 1.07 cores, it must have at least two VPs. PowerVP GUI shows that LPAR 2 has two VPs dispatched on core 3 on Chip2.

  Current partition entitlement and peak usage for this partition are not balanced. Current partition workload is more than six times higher that the anticipated workload. The partition entitlement must be increased.

  Current partition entitlement and the number of VPs are not balanced. The number of VPs must be reduced.

- Both LPARs 21 and 22 have a cache affinity for the same processor core on `Chip2`.
- PowerVP GUI shows LPAR `2` still has memory allocated on `Chip2` even if the partition is no longer active. You can release the memory assigned to partition that has been deactivated by running the following command from the HMC:

  ```
  chhwres -r mem -m <system_name> -o r -q <number_of_Mbytes> --id <LPAR_id>
  ```

- PowerVP GUI shows LPAR `1` still has memory allocated from `Node1`. There are no processor cores allocated to LPAR `1` from `Node1`. However, the partition is still active. This is a suboptimal case of resource distribution.

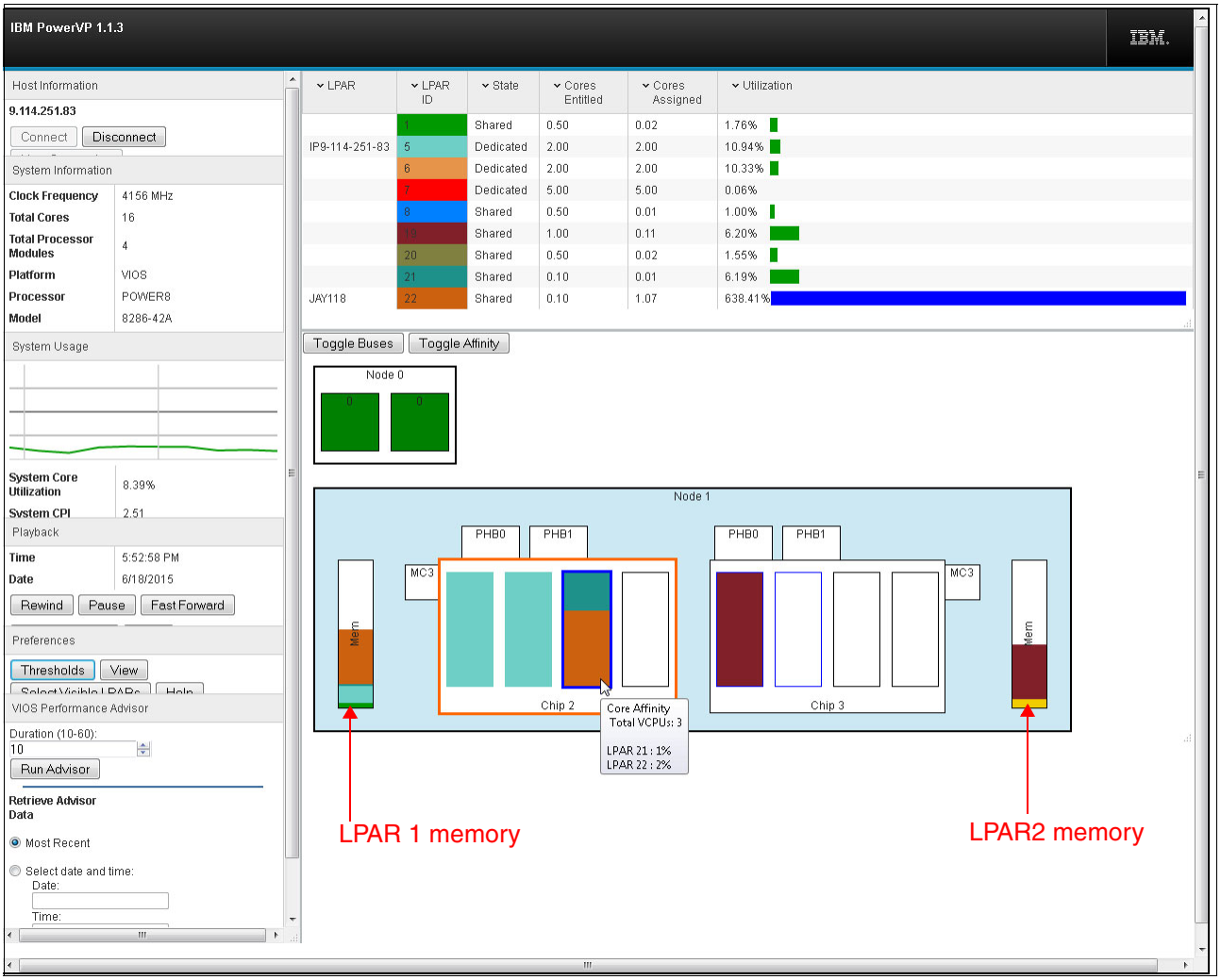


Figure 4-1 Assessing resource distribution and partition affinity using PowerVP

Figure 4-2 shows how to use PowerVP to assess resource distribution by analyzing LSU CPI data.

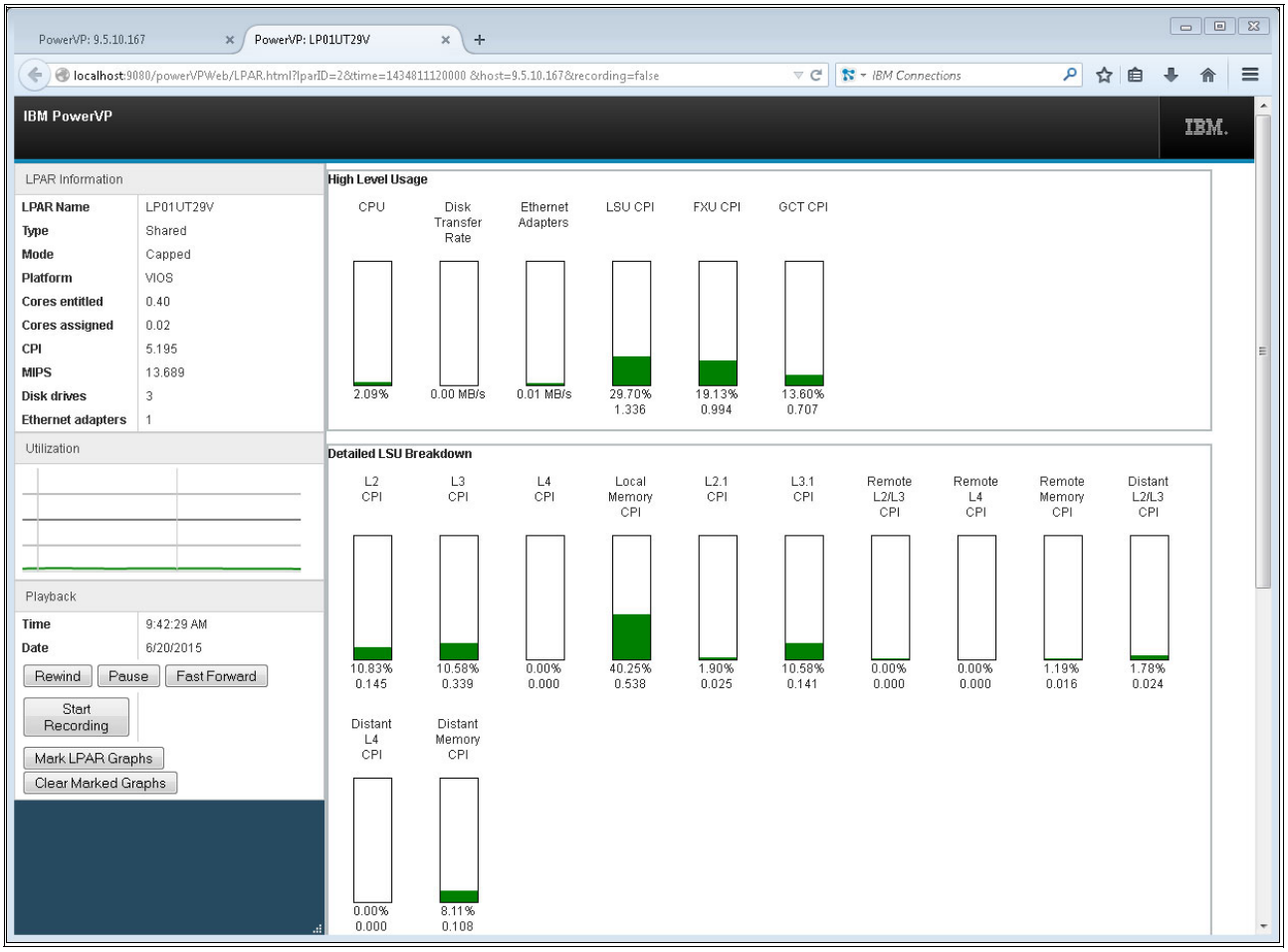


Figure 4-2 Assessing LSU CPI data for an individual partition using PowerVP

The LSU CPI indicator displays the percentage of processor core activity and the number of CPU cycles spent by the Load/Store Unit while handling data. The detailed breakdown of LSU CPI displays statistics collected for all existing levels of cache and memory.

PowerVP GUI displays in the Detailed LSU Breakdown pane the percentage of utilization and the number of CPU cycles used to access existing cache and memory. Various levels of cache and memory are sorted from left to right according to their latencies.

Note: Access to local memory is faster than the access to L2/L3 cache of any other processor core that is in the same processor chip.

Remote L2/L3 cache and memory have a higher latency than local L2/L3 cache and memory.

Distant L2/L3 cache and memory have a higher latency than Remote L2/L3 cache and memory.

To achieve an optimal performance, PowerVP GUI should display nonzero values only for the local counters.

Based on the data provided in Figure 4-2 on page 86 by the PowerVP GUI, you can make the following determinations:

- The LSU CPI shows that 29.70% of the processor core activity is used for loading and storing data. This activity requires 1336 CPU cycles.
- The L2 CPI columns show a 10.83% percentage of utilization and 145 CPU cycles to access L2 cache data.
- The L3 CPI columns show a 10.58% percentage of utilization and 339 CPU cycles to access L3 cache data.
- The Local Memory CPI column shows a 40.25% percent of utilization and 538 CPU cycles to access local memory.
- The L2.1 columns show a 1.90% percent of utilization and 23 CPU cycles to access data in the L2 cache of a processor core located within the same chip.
- The L3.1 columns show a 10.58% percent of utilization and 141 CPU cycles to access data that is in the L3 cache of a processor core located within the same chip.
- The Remote L2/L3 CPI column shows the percent of utilization and CPU cycles to access data that is in L2/L3 cache of a processor core that is in a different chip, but within the same node.
- The Remote Memory CPI column shows a 1.19% percent of utilization and 16 CPU cycles to access data that is in memory attached to a different processor chip, but within the same node.
- The Distant L2/L3 CPI column shows a 1.78% percent of utilization and 24 CPU cycles to access data that is in L2/L3 cache of a processor core that is in a different chip and different node.
- The Distant Memory CPI column shows a 8.11% percent of utilization and 108 CPU cycles to access data that is in memory attached to a different processor chip, but within the same node.

PowerVP GUI shows *remote* and even *distant* cache and memory access. This is a suboptimal case of resource distribution.

5

Experiences with PowerVP and PowerVM features

This chapter provides information about how to use IBM Power Virtualization Performance (PowerVP) features in real virtualized IBM Power Systems environments. The information and practices included in this chapter can be easily extended and reused in similar environments.

This chapter includes the following sections:

- Introduction to Dynamic Platform Console
- Introduction to Dynamic LPAR
- Operating system support
- Description of the environment
- Practical scenario for using PowerVP with DPO and DLPAR
- Integration with VIOS performance advisor
- Configuring PowerVP thresholds and alerts

5.1 Introduction to Dynamic Platform Console

Processor and memory allocation can become suboptimal in time as a consequence of reassigning or moving resources. This chapter shows how to use PowerVP with PowerVM virtualization features to improve the affinity of memory and processors on Power server environments managed by the Hardware Management Console (HMC).

The example in this chapter uses Dynamic Platform Optimizer (DPO) and dynamic LPAR (DLPAR) features in a typical scenario to improve resource distribution and affinity for partitions running various operating systems.

DPO can determine the optimal assignment of resources at both system and partition level. All operations that are required to achieve the optimum level are dynamically performed while the partitions are running and DPO actions are initiated from the HMC. On larger systems, resources relocation can improve the performance considerably depending on the system configuration.

DPO computes a system-wide affinity score based on the partition configuration and the hardware topology. DPO features include calculating the *current* affinity score and the *potential* affinity score that can be achieved following resource optimization. DPO can start and run all memory and processor relocations that are required to achieve the optimum level. When the optimizer completes an optimization, it notifies the operating systems in the partitions that their physical memory and processors configurations have been changed.

5.2 Introduction to Dynamic LPAR

Dynamic LPAR (DLPAR) is a PowerVM feature that allows you to dynamically move resources between partitions without restarting the operating system that runs in the partition. The following resources can be moved:

- Physical processor cores (for dedicated LPARs)
- Virtual processors or entitled capacity (for shared partitions)
- Memory
- Physical I/O adapters
- Virtual I/O adapters

For more information about DPO and DLPAR, see these IBM Redbooks publications:

- *IBM PowerVM Virtualization Managing and Monitoring*, SG24-7590
- *IBM PowerVM Virtualization Introduction and Configuration*, SG24-7940

5.3 Operating system support

The latest versions of operating systems automatically become aware of any change to existing hardware topology of the partitions they are running in. Older versions of operating systems that do not have this capability must be restarted.

The following list includes the operating system levels that can determine automatically any change of existing hardware resources:

- AIX 6.1 TL8 or later
- AIX 7.1 TL2 or later
- IBM i 7.1 PTF MF56058

- VIOS 2.2.2.0 or later
- Red Hat Enterprise Linux 7
- SUSE Linux Enterprise Server 12

5.4 Description of the environment

The example scenario uses an environment with the following components:

- IBM PowerVP web-based GUI running on a Microsoft Windows 7 workstation
- IBM POWER8 System 8286-42A comprising two nodes
- HMC Version 8 Release 8.3.0
- PowerVP System level agent installed on a VIOS partition
- Multiple partitions running VIOS, AIX, IBM i, RHEL, and SUSE

5.5 Practical scenario for using PowerVP with DPO and DLPAR

This section shows a practical example of improving resource distribution by using PowerVP, DLPAR, and DPO. We complete the following steps:

1. We use PowerVP web-based GUI to assess resource distribution. The current layout of resources on Node 1 is shown in Figure 5-1 on page 92. The current layout of resources on Node 0 is shown in Figure 5-2 on page 93.

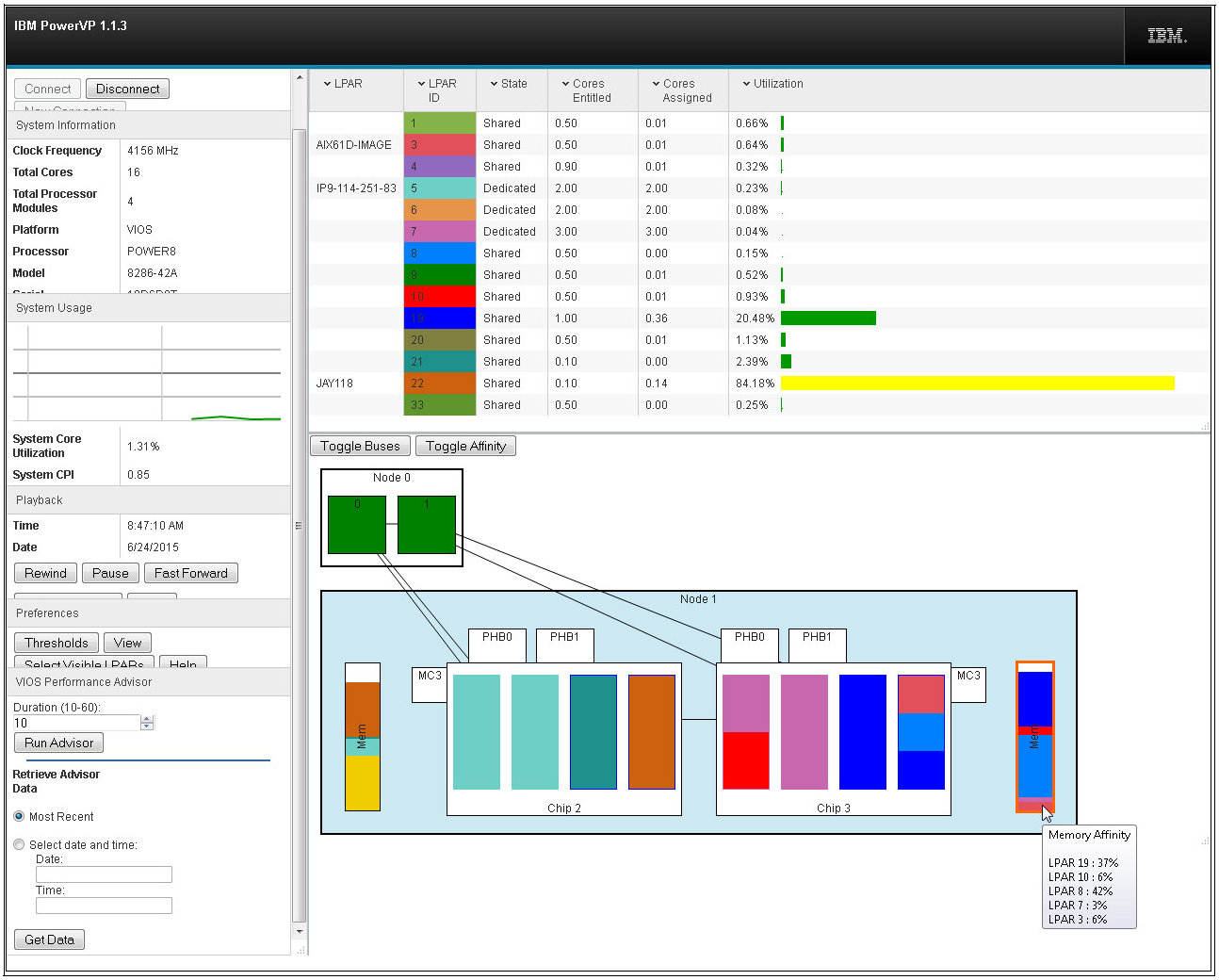


Figure 5-1 Assessing resource distribution on Node 1 before DPO

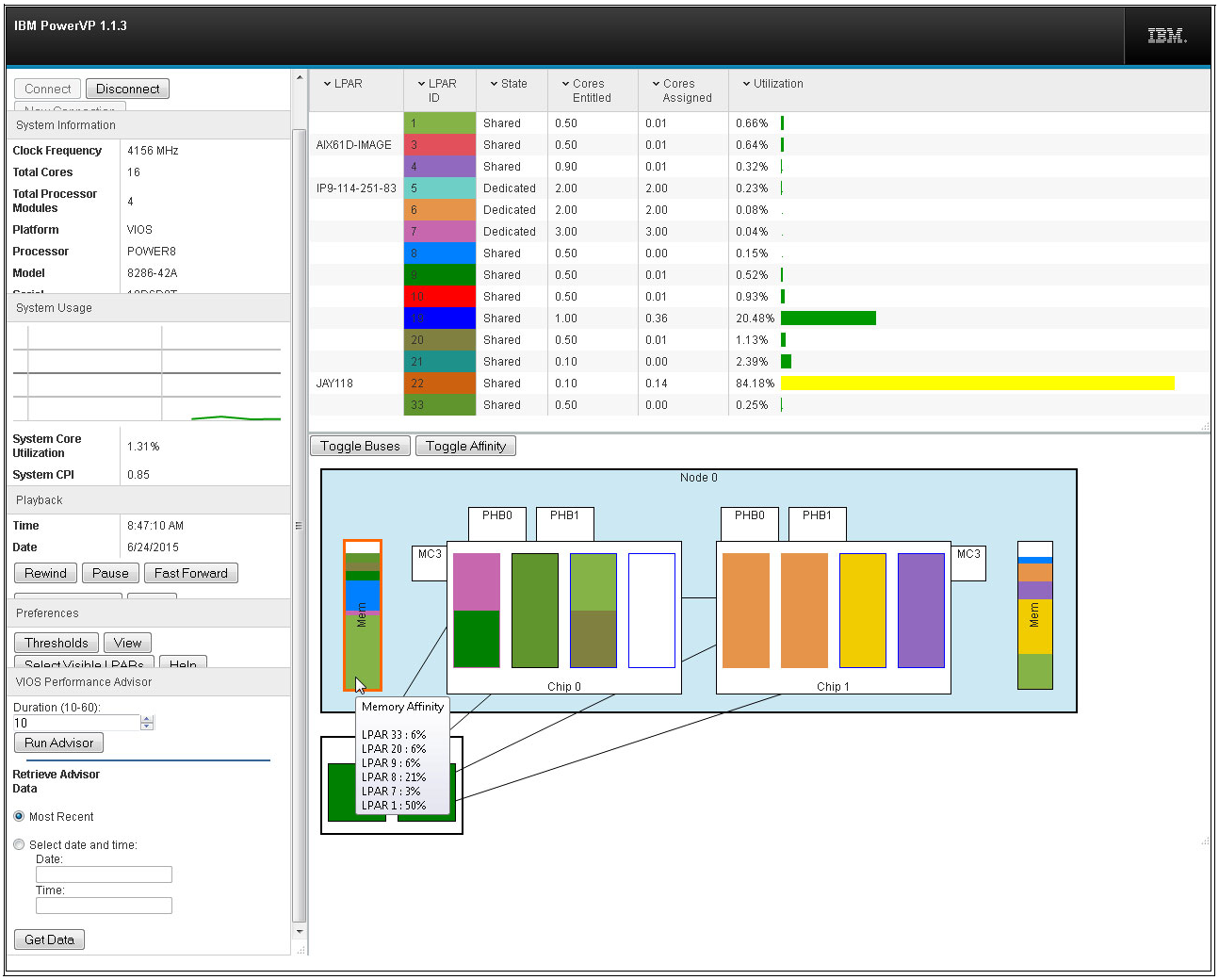


Figure 5-2 Assessing resource distribution on Node 0 before DPO

2. We use the **lssyscfg** command to verify whether the IBM Power System has the DPO feature enabled. We use the HMC **lsmemopt** command to assess the current and potential affinity score for all LPARs and the entire system as shown in Example 5-1.

Example 5-1 Using HMC lsmemopt command to assess affinity score for all LPARs and the whole system

```
hscroot@ip9-114-251-165:~> lssyscfg -m P8_10 -r sys -F
dynamic_platform_optimization_capable
1

hscroot@ip9-114-251-165:~> lsmemopt -m P8_10 -r sys -o calcscore
curr_sys_score=65,predicted_sys_score=81,requested_lpar_ids=none,protected_lpar
_ids=none

hscroot@ip9-114-251-165:~> lsmemopt -m P8_10 -r lpar -o calcscore
lpar_name=divya_root-bb3dda65-00000007,lpar_id=1,curr_lpar_score=32,predicted_l
par_score=100
lpar_name=New_IBM_i_ima-f9e98c74-0000002a,lpar_id=2,curr_lpar_score=50,predict
ed_lpar_score=100
```

```
lpar_name=AIX61d_image-61e1254f-0000005b,lpar_id=3,curr_lpar_score=100,predicte
d_lpar_score=0
lpar_name=aix7133_image-0abaf297-00000062,lpar_id=4,curr_lpar_score=66,predicte
d_lpar_score=25
lpar_name=P8_10_vios1,lpar_id=5,curr_lpar_score=100,predicted_lpar_score=100
lpar_name=P8_10_vios2,lpar_id=6,curr_lpar_score=100,predicted_lpar_score=100
lpar_name=AIX61d_Image-bcfc03ec-00000008,lpar_id=7,curr_lpar_score=50,predicted
_lpar_score=100
lpar_name=SUSE11SP3_Ima-d2096500-00000034,lpar_id=8,curr_lpar_score=28,predicte
d_lpar_score=100
lpar_name=aix7133_image-a90b6beb-00000065,lpar_id=9,curr_lpar_score=100,predict
ed_lpar_score=0
lpar_name=aix7133_2pv_0-f40137ab-00000066,lpar_id=10,curr_lpar_score=100,predic
ted_lpar_score=0
lpar_name=powervc123_powerkvm,lpar_id=19,curr_lpar_score=100,predicted_lpar_sco
re=66
lpar_name=AIX61d_Image_-d89666bb-00000001,lpar_id=20,curr_lpar_score=100,predic
ted_lpar_score=37
lpar_name=SUSE11SP3_Ima-ecb0e0f7-00000008,lpar_id=21,curr_lpar_score=100,predic
ted_lpar_score=0
lpar_name=RHEL7.1LE-ae7e44d8-00000016,lpar_id=22,curr_lpar_score=55,predicted_l
par_score=100
lpar_name=TestScriptDep-291f34ea-00000004,lpar_id=33,curr_lpar_score=100,predic
ted_lpar_score=0
```

Note these details in Example 5-1 on page 93:

- Current system affinity score is 65 and can be improved to 81.
- Multiple LPARs have affinity scores that can be improved up to 100.
- Multiple LPARs have an affinity of 100.
- LPAR 2 has an affinity score displayed, even if the LPAR is not powered on. This means that the LPAR still has resources that are allocated to it.

Important: There are cases when running DPO increases the overall system affinity score while individual LPAR affinity scores may remain unchanged or even decrease.

3. We start the DPO operation by using the **optmem** command as shown in Example 5-2. However, the command fails to start because the system does not have enough free memory.

Example 5-2 DPO cannot run because it does not have enough free memory

```
hscroot@ip9-114-251-165:~> optmem -m P8_10 -t affinity -o start
hscroot@ip9-114-251-165:~> lsmemopt -m P8_10
in_progress=0,status=Error - insufficient free
memory,type=affinity,opt_id=23,progress=0,requested_lpar_ids=none,protected_lpar
_ids=none,impacted_lpar_ids=none
```

4. We release 4 GB of memory from the amount of memory that is still allocated to LPAR 2 using the **chhwres** command as shown in Example 5-3. When the memory is released, the DPO starts successfully as shown by the **lsmemopt** command.

Example 5-3 Releasing partition memory and running DPO

```
hscroot@ip9-114-251-165:~> chhwres -r mem -m P8_10 -o r -q 4096 --id 2
hscroot@ip9-114-251-165:~> optmem -m P8_10 -t affinity -o start
hscroot@ip9-114-251-165:~> lsmemopt -m P8_10
in_progress=1,status=In
progress,type=affinity,opt_id=24,progress=55,requested_lpar_ids=none,protected_
lpar_ids=none,impacted_lpar_ids=none
```

5. We use the HMC **lsmemopt** command to assess the new affinity score for all partitions and the entire system when DPO completes as shown in Example 5-4.

Example 5-4 Verifying new system and partition affinity scores

```
hscroot@ip9-114-251-165:~> lsmemopt -m P8_10
in_progress=0,status=Finished,type=affinity,opt_id=25,progress=100,requested_lp
ar_ids=none,protected_lpar_ids=none,"impacted_lpar_names=New_IBM_i_ima-f9e98c74
-0000002a,AIX61d_image-61e1254f-0000005b,aix7133_image-0abaf297-00000062,aix713
3_image-a90b6beb-00000065,powervc123_powerkvm,SUSE11SP3_Ima-ecb0e0f7-00000008",
"impacted_lpar_ids=2-4,9,19,21"

hscroot@ip9-114-251-165:~> lsmemopt -m P8_10 -r sys -o calcscore
curr_sys_score=82,predicted_sys_score=83,requested_lpar_ids=none,protected_lpar_
ids=none

hscroot@ip9-114-251-165:~> lsmemopt -m P8_10 -r lpar -o calcscore
lpar_name=divya_root-bb3dda65-00000007,lpar_id=1,curr_lpar_score=100,predicted_
lpar_score=100
lpar_name=New_IBM_i_ima-f9e98c74-0000002a,lpar_id=2,curr_lpar_score=100,predict
ed_lpar_score=100
lpar_name=AIX61d_image-61e1254f-0000005b,lpar_id=3,curr_lpar_score=56,predicted
_lpar_score=56
lpar_name=aix7133_image-0abaf297-00000062,lpar_id=4,curr_lpar_score=90,predicte
d_lpar_score=90
lpar_name=P8_10_vios1,lpar_id=5,curr_lpar_score=100,predicted_lpar_score=100
lpar_name=P8_10_vios2,lpar_id=6,curr_lpar_score=100,predicted_lpar_score=100
lpar_name=AIX61d_Image-bcfc03ec-00000008,lpar_id=7,curr_lpar_score=100,predicte
d_lpar_score=100
lpar_name=SUSE11SP3_Ima-d2096500-00000034,lpar_id=8,curr_lpar_score=90,predicte
d_lpar_score=100
lpar_name=aix7133_image-a90b6beb-00000065,lpar_id=9,curr_lpar_score=0,predicted
_lpar_score=0
lpar_name=aix7133_2pv_0-f40137ab-00000066,lpar_id=10,curr_lpar_score=0,predicte
d_lpar_score=0
lpar_name=powervc123_powerkvm,lpar_id=19,curr_lpar_score=50,predicted_lpar_scor
e=50
lpar_name=AIX61d_Image_-d89666bb-00000001,lpar_id=20,curr_lpar_score=0,predicte
d_lpar_score=0
lpar_name=SUSE11SP3_Ima-ecb0e0f7-00000008,lpar_id=21,curr_lpar_score=0,predicte
d_lpar_score=0
lpar_name=RHEL7.1LE-ae7e44d8-00000016,lpar_id=22,curr_lpar_score=100,predicted_
lpar_score=100
```

```
lpar_name=TestScriptDep-291f34ea-00000004,lpar_id=33,curr_lpar_score=0,predicte
d_lpar_score=0
```

Note these details in Example 5-4 on page 95:

- DPO operation affected LPARs 2-4, 9, 19, and 21.
- System affinity score is now 82 and can still be improved up to 83.
- Releasing unused memory allowed the system to exceed the predicted affinity score of 81.

Important:

- There are cases when the affinity score obtained after a DPO can be different than predicted.
- There are cases when the predicted affinity score obtained before a DPO can be further increased by more DPO operations.

6. We release the memory that is allocated to LPAR 2 in several steps, 4G of memory each time until the whole amount of memory is released as shown in Example 5-5. We compute the new potential system affinity score. After the whole memory is released, current system affinity score is 80 and can be increased up to 98.

Example 5-5 Releasing all memory allocated to LPAR 2 and computing a new system affinity score

```
hscroot@ip9-114-251-165:~> chhwres -r mem -m P8_10 -o r -q 4096 --id 2
hscroot@ip9-114-251-165:~> chhwres -r mem -m P8_10 -o r -q 4096 --id 2
hscroot@ip9-114-251-165:~> chhwres -r mem -m P8_10 -o r -q 4096 --id 2
hscroot@ip9-114-251-165:~> chhwres -r mem -m P8_10 -o r -q 4096 --id 2
hscroot@ip9-114-251-165:~> chhwres -r mem -m P8_10 -o r -q 4096 --id 2
hscroot@ip9-114-251-165:~> chhwres -r mem -m P8_10 -o r -q 4096 --id 2
HSCLA44A You cannot remove more memory than your partition currently has
assigned, which is 0 MB.

hscroot@ip9-114-251-165:~> lsmemopt -m P8_10 -r sys -o calcscore
curr_sys_score=80,predicted_sys_score=98,requested_lpar_ids=none,protected_lpar_
ids=none
```

7. Because the affinity can still be improved, we start a second DPO operation, the results of which are shown in Example 5-6.

Example 5-6 Running a second DPO operation and displaying the results

```
hscroot@ip9-114-251-165:~> optmem -m P8_10 -t affinity -o start

hscroot@ip9-114-251-165:~> lsmemopt -m P8_10 -r sys -o calcscore
curr_sys_score=97,predicted_sys_score=98,requested_lpar_ids=none,protected_lpar_
ids=none

hscroot@ip9-114-251-165:~> lsmemopt -m P8_10 -r lpar -o calcscore
lpar_name=divya_root-bb3dda65-00000007,lpar_id=1,curr_lpar_score=100,predicted_
lpar_score=100
lpar_name=New_IBM_i_ima-f9e98c74-0000002a,lpar_id=2,curr_lpar_score=none,predic
ted_lpar_score=none
lpar_name=AIX61d_image-61e1254f-0000005b,lpar_id=3,curr_lpar_score=100,predicte
d_lpar_score=100
```

```
lpar_name=aix7133_image-0abaf297-00000062,lpar_id=4,curr_lpar_score=90,predicte
d_lpar_score=90
lpar_name=P8_10_vios1,lpar_id=5,curr_lpar_score=100,predicted_lpar_score=100
lpar_name=P8_10_vios2,lpar_id=6,curr_lpar_score=100,predicted_lpar_score=100
lpar_name=AIX61d_Image-bcfc03ec-00000008,lpar_id=7,curr_lpar_score=100,predicte
d_lpar_score=100
lpar_name=SUSE11SP3_Ima-d2096500-00000034,lpar_id=8,curr_lpar_score=90,predicte
d_lpar_score=100
lpar_name=aix7133_image-a90b6beb-00000065,lpar_id=9,curr_lpar_score=100,predict
ed_lpar_score=100
lpar_name=aix7133_2pv_0-f40137ab-00000066,lpar_id=10,curr_lpar_score=90,predict
ed_lpar_score=90
lpar_name=powervc123_powerkvm,lpar_id=19,curr_lpar_score=90,predicted_lpar_scor
e=90
lpar_name=AIX61d_Image_-d89666bb-00000001,lpar_id=20,curr_lpar_score=100,predic
ted_lpar_score=100
lpar_name=SUSE11SP3_Ima-ecb0e0f7-00000008,lpar_id=21,curr_lpar_score=90,predict
ed_lpar_score=90
lpar_name=RHEL7.1LE-ae7e44d8-00000016,lpar_id=22,curr_lpar_score=100,predicted_
lpar_score=100
lpar_name=TestScriptDep-291f34ea-00000004,lpar_id=33,curr_lpar_score=100,predic
ted_lpar_score=100
```

From Example 5-6 on page 96, note the following details:

- LPAR 2 is no longer considered by DPO.
- The system has reached the affinity score of 97, close to the predicted 98.

8. Use PowerVP web-based GUI to assess the new resource distribution. The current layout of resources on `Node 1` is shown in Figure 5-4 on page 99. The current layout of resources on `Node 0` is shown in Figure 5-3.

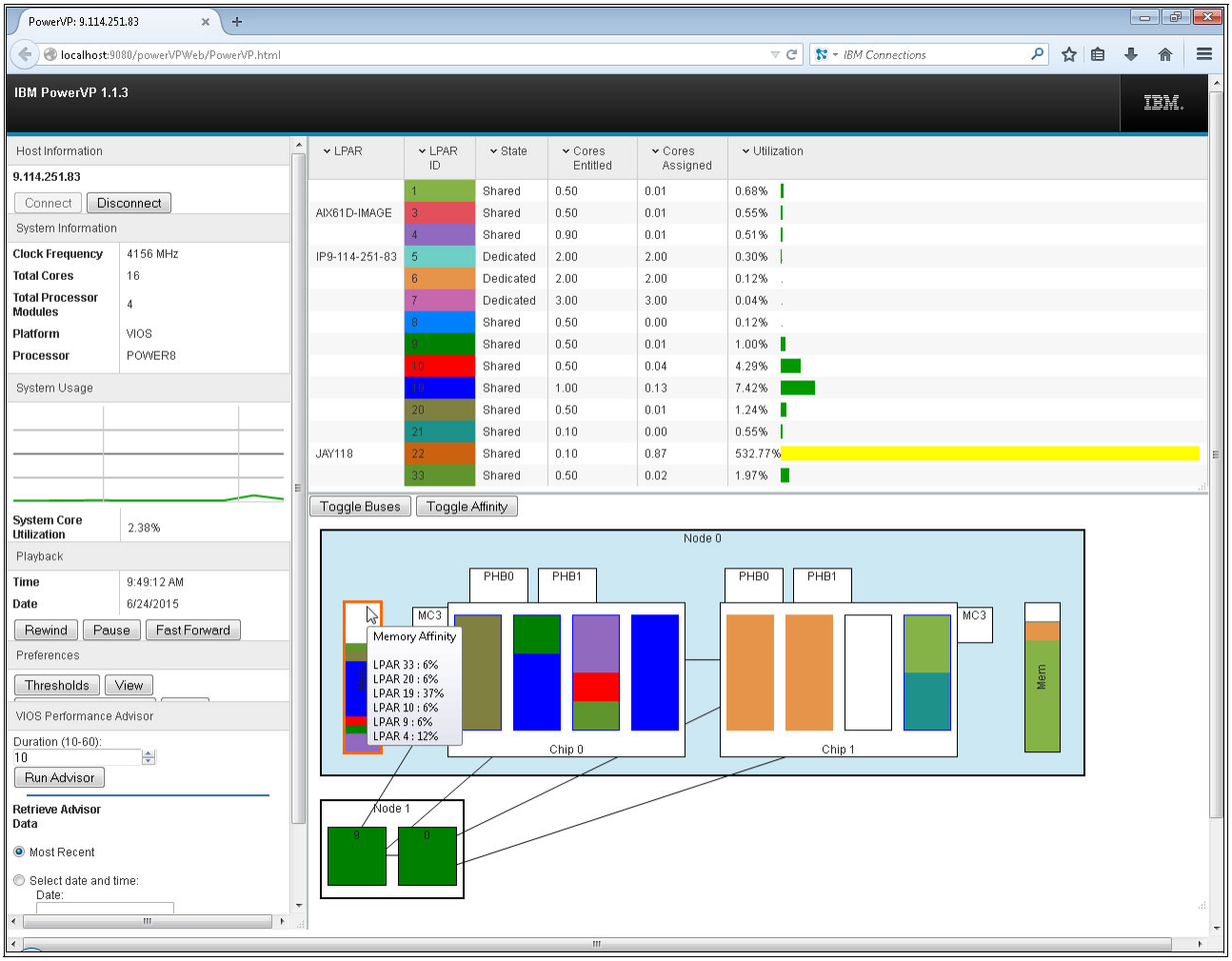


Figure 5-3 Assessing resource distribution after DPO on Node 0

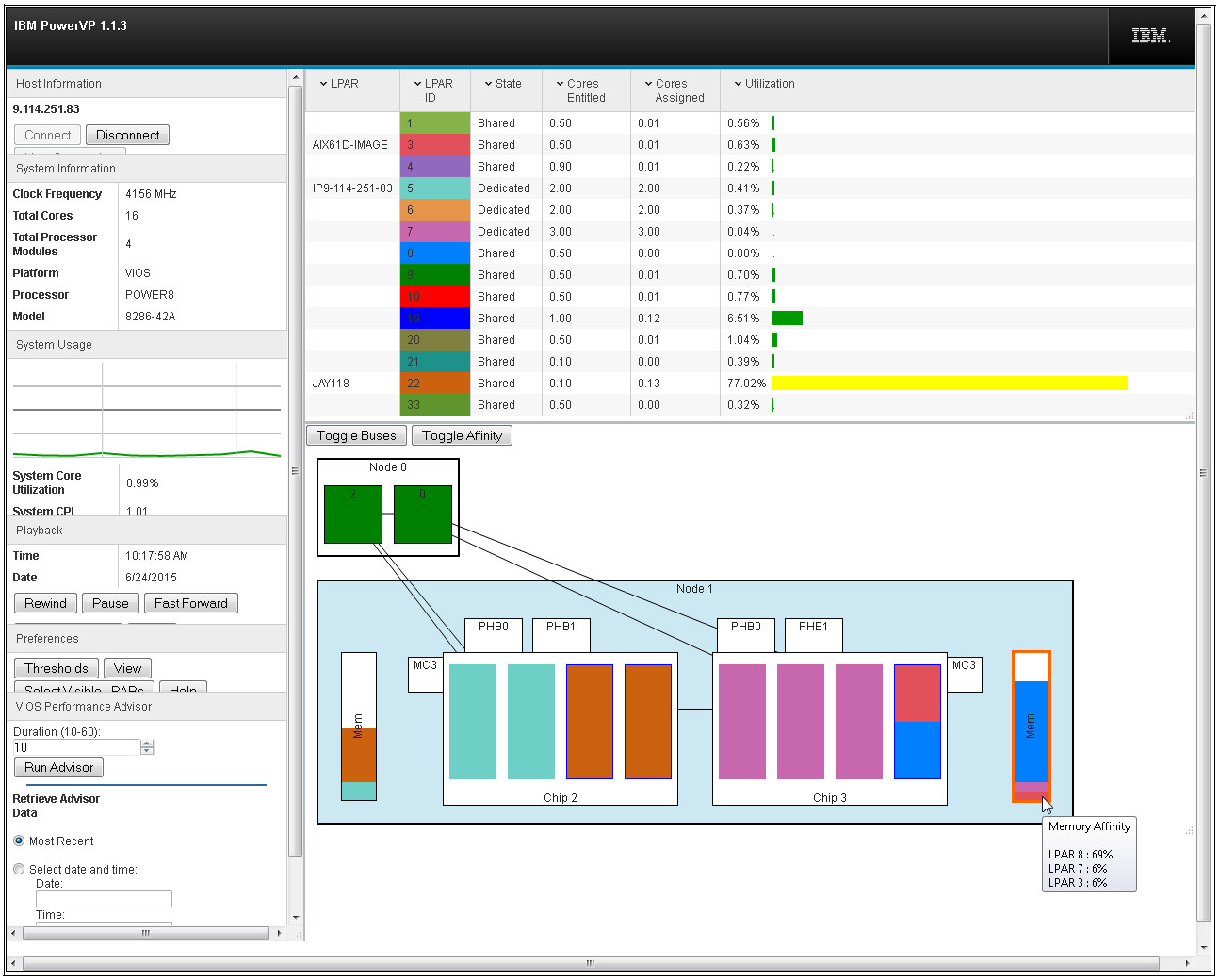


Figure 5-4 Assessing resource distribution after DPO one Node 1

Important: Location of resources can change following DPO operations and the affinity score may remain unchanged. There can be multiple distributions of system resources that generate the same affinity score.

5.6 Integration with VIOS performance advisor

The VIOS Performance Advisor tool provides advisory reports that are based on the key performance metrics on various partition resources collected from the VIOS environment. The VIOS Performance Advisor is shipped with VIOS starting with version 2.2.2.0. You can configure the PowerVP agent running on VIOS to run the VIOS Performance Advisor on a schedule. This is done by using the `VIOSAdvisor` directive set in PowerVP configuration file `-/etc/opt/ibm/powervp/powervp.conf`.

```
VIOSAdvisor runTime timeToRun
```

where `runTime` is the duration of time that you want VIOS Performance Advisor to collect information before generating the report. This value must be 10 - 60 and is in minutes. The

`timeToRun` is the hour of the day that you want PowerVP to start the VIOS Performance Advisor. This value must be 0 - 23 and is in hours, corresponding to the military time of day. If you want it to run multiple times a day, you can specify multiple VIOSAdvisor directives. Each report that is generated by VIOS Performance Advisor is a .tar file archive in the `/opt/ibm/powervp/advisor/` directory. File name format is `hostname_YYMMDD_HH_MM_SS.tar`.

Important: The size of the report files depends on runTime duration. Ensure that sufficient free disk space is available.

To display the VIOS Performance Advisor reports in the GUI, you first need to connect the GUI to the VIOS system agent. All control options are available on an intended pane that is shown in Figure 5-5.

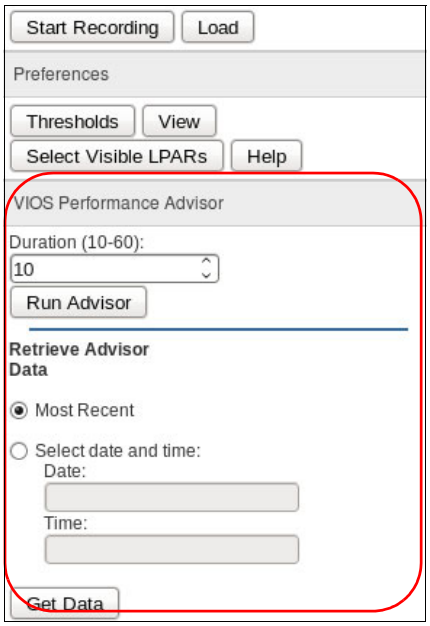


Figure 5-5 VIOS Performance Advisor

The VIOS Performance Advisor section is presented for you to either view the latest report (a report from a specified date and time) or to run the VIOS Performance Advisor at that time and provide a duration to be used. When PowerVP needs to run the VIOS Performance Advisor, it starts it and then waits for the output to be generated and sends it to the GUI when it is available. When you select to view the latest report, PowerVP finds it and sends it to the GUI. When you provide a date and time, PowerVP sends the report that has the time stamp equal to or the first one greater than the date and time provided.

For more information about VIOS Performance Advisor, see the documentation in the IBM Knowledge Center:

http://www-01.ibm.com/support/knowledgecenter/POWER7/p7hb1/iphb1_vios_perf_adv.htm?cp=POWER7%2F1-8-3-5-8-8&lang=en

5.7 Configuring PowerVP thresholds and alerts

PowerVP allows you to configure System level agents or Partition level agents to monitor specific resources, such as CPU and system buses. You can configure specific thresholds for each resource.

When the usage rate exceeds a predefined threshold for a specific amount of time, the PowerVP agent logs a message to the PowerVP log and sends an alert. For AIX, Linux, and VIOS an alert represents a message to the syslog. For IBM System i®, an alert represents a message to the QSYSOPR message queue.

You can use operating system features to monitor these messages or you can use other applications to send notifications such as emails, console messages, or text messages.

To activate thresholds and alerts, you must update the agent configuration file accordingly and restart the agent. You can update the agent configuration file by specifying one or more of the following directives:

UtilizationAlertPartitionCPU	CPU usage of an individual partition based on partition EC.
UtilizationAlertSystemCPU	CPU usage of the whole IBM Power system.
UtilizationAlertAbus	Bus usage for an inter-Node bus.
UtilizationAlertXbus	Bus usage for an intra-Node bus.
UtilizationAlertMCbus	Bus usage for memory controller bus.
UtilizationAlertInputIObus	Bus usage for incoming IO activity.
UtilizationAlertOutputIObus	Bus usage for outgoing IO activity.

The configuration lines in the PowerVP agent configuration file must have the following syntax:

```
Directive percentage duration realert level
```

where:

Directive	One of the predefined directives.
Percentage	Usage threshold at which alert monitoring is started.
Duration	Amount of time in seconds for which the usage rate must exceed the preconfigured threshold. After this period an alert message is sent.
Realert	If the usage rate *continues* to exceed the preconfigured threshold, this is the amount of time in seconds *after* which another alert message is sent.
Level	The severity of the alert message sent. This parameter is *not* used for IBM System i. This parameter is used only for AIX, Linux, and VIOS and can take any of the values that are valid for syslog severity levels. The syslog facility used is *daemon*.

Example 5-7 on page 102 shows how to use directive `UtilizationAlertPartitionCPU` to configure the PowerVP partition agent to send a warning alert when the partition CPU usage rate exceeds 80% for 10 seconds. The realert value is set to 10 seconds.

In the PowerVP log, warning alerts showing partition 22 exceeding 80% CPU usage are displayed every 10 seconds starting at `13:41:04`.

Example 5-7 Partition alert when CPU usage rate exceeds 80%

```
# cat /etc/opt/ibm/powervp/powervp.conf|grep -i warn
#   levels. Values include Emergency, Alert, Critical, Error, Warning, Notice,
UtilizationAlertPartitionCPU 80 10 10 warning

# cat powervp.log|grep " partition 22"
2015-06-22 13:41:04.538 MSG0106: CPU utilization for partition 22 at 88 percent.
2015-06-22 13:41:15.539 MSG0106: CPU utilization for partition 22 at 82 percent.
2015-06-22 13:41:26.540 MSG0106: CPU utilization for partition 22 at 123 percent.
2015-06-22 13:41:37.540 MSG0106: CPU utilization for partition 22 at 93 percent.
2015-06-22 13:41:48.548 MSG0106: CPU utilization for partition 22 at 84 percent.
```

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this paper.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.

- *IBM PowerVM Best Practices*, SG24-8062
- *IBM PowerVM Virtualization Introduction and Configuration*, SG24-7940
- *IBM PowerVM Virtualization Managing and Monitoring*, SG24-7590

You can search for, view, download or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

http://www.ibm.com/redbooks

Other publications

These publications are also relevant as further information sources:

For more information about performance on IBM Power Systems, see the following website and select the Doc-type **Word document** *P7 Virtualization Best Practice*:

https://www.ibm.com/developerworks/wikis/display/WikiPtype/Performance+Monitoring+Documentation

Online resources

This website is also relevant as a further information source:

PowerVP 1.1.0 announcement letter

http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?infotype=AN&subtype=CA&htmlfid=649/ENUSA13-0798&appname=USN#h2-abstrx

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Redbooks

IBM PowerVP: Introduction and Technical Overview

(0.2”spine)
0.17”<->0.473”
90<->249 pages

REDP-5112-01

ISBN 0738454362

Printed in U.S.A.