HP-UX Workload Manager overview

Introduction 3

What is HP-UX Workload Manager? 4
- Workload management across virtual partitions and nPartitions 5
- Workload management within a single HP-UX instance 6

Why use HP-UX Workload Manager? 6
- Service level objectives 7
- Prioritized SLOs 8

What is the ideal environment for HP-UX Workload Manager? 8

HP-UX Workload Manager solutions 9
- SLOs that ensure a specified amount of CPU resources for workloads 9
  - Reserving CPU resources all of the time 9
  - Reserving CPU resources at specified times 9
  - Reserving CPU resources based on an event or condition 9
  - Reserving CPU resources in an HP Serviceguard failover 9
- SLOs that dynamically allocate resources based on usage goals 10
  - Allocating CPU resources dynamically based on utilization 10
  - Controlling sharing and borrowing of excess CPU resources 10
  - Automatically resizing pSets 10
  - Automatically resizing virtual partitions 11
  - Automatically resizing nPartitions using HP Instant Capacity cores 11
  - Optimizing the use of HP Temporary Instant Capacity and HP Pay per use 11
- SLOs that maintain performance based on metric goals 12
  - Allocating CPU resources per metric 12

HP-UX Workload Manager operational overview 12

Managing application performance: Quick start 15

Managing application performance with metric goals: Advanced HP-UX Workload Manager management 15

HP-UX Workload Manager configuration 17

Testing your configuration file without HP-UX Workload Manager controlling the system 18

Monitoring your service level objectives 19

Monitoring with the command line interface 19
Monitoring with the graphical user interface 21
Forming Secure Resource Partitions 22
Using HP-UX Workload Manager for server consolidation 22
Using HP-UX Workload Manager for auditing and billing 22
Managing nested partitions 23
Using HP-UX Workload Manager on multiple servers 23
Using HP-UX Workload Manager with HP Temporary Instant Capacity and HP Pay per use 24
Using HP-UX Workload Manager with HP Integrity Virtual Machines 24
Using HP-UX Workload Manager with HP Serviceguard 25
Using HP-UX Workload Manager with HP Systems Insight Manager and HP Servicecontrol Manager 26
Using HP-UX Workload Manager with Oracle databases 26
Using HP-UX Workload Manager with Apache 26
Using HP-UX Workload Manager with BEA WebLogic Server 27
Using HP-UX Workload Manager with SAP 27
Using HP-UX Workload Manager with SNMP 27
HP-UX Workload Manager, HP Global Workload Manager, and HP Process Resource Manager 28
Related information 28
For more information 30

Introduction

Today, most servers are highly underutilized. While average utilization varies by customer and operating system, in the HP-UX environment it is often around 30%. There are myriad reasons for this utilization, but one of the primary reasons is that customers often have one application per server, and they size that server for a peak load of typically three to five times the average utilization. This means that millions of dollars in server resources—CPU resources and memory—can lie idle for most of the time.

Resource optimization, which is one of the goals of the HP Adaptive Enterprise strategy, enables you to combine applications while maintaining performance. The Adaptive Enterprise strategy helps customers synchronize business and IT resources to adapt to and capitalize on change. To help you realize the promise of becoming an Adaptive Enterprise, HP provides virtualization technologies that pool and share resources to optimize utilization and meet demands automatically.

HP-UX Workload Manager (WLM) is a virtualization solution that helps you achieve a true Adaptive Enterprise. As a goal-based policy engine in the HP Virtual Server Environment (VSE), WLM integrates virtualization techniques—including partitioning, resource management, utility pricing resources, and clustering—and links them to your service level objectives (SLOs) and business priorities. WLM enables a virtual HP-UX server to grow and shrink automatically based on the demands and SLOs for each application it hosts. WLM helps you receive greater return on your IT investment while ensuring that end-users receive the service and performance they expect.

The HP Virtual Server Environment offers several forms of partitioning that WLM can manage:

- **Hard partitions**—Hard partitions are implemented and isolated through hardware. The first form of hard partitions is a complete server, which can be clustered in an HP Serviceguard high-availability cluster or a Hyperplex configuration. nPartitions, which are portions of a single server,are other forms of hard partitions. Each hard partition runs its own instance of HP-UX. Isolation provided by nPartitions guarantees that an application running in one partition is not affected by an application or hardware failure in another.
- **Virtual partitions**—Virtual partitions are implemented and isolated through software, with each virtual partition running its own instance of the HP-UX operating system. HP virtual partitions offer unique granularity for partitioning servers. You can create a virtual partition consisting of one or more cores, and you can use virtual partitions within hard partitions. (A core is the actual data processing engine within a processor, where a single processor can have multiple cores.) Virtual partitions provide complete software isolation between partitions.
- **Virtual machines**—Virtual machines, much like virtual partitions, are created with software. However, they emulate generic servers and, therefore, can offer subcore and shared I/O capabilities. Each virtual machine runs its own operating system. HP Integrity Virtual Machines can be used within hard partitions.
- **Resource partitions**—Resource partitions are provided by HP Process Resource Manager (PRM) to manage processor sets and Fair Share Scheduler (FSS) groups. These partitions enable you to partition system resources (including memory and disk bandwidth) within a single instance of HP-UX and consolidate multiple workloads within that instance. You can use these partitions within (but not across) hard partitions and virtual partitions.

You can use WLM to manage system resources within resource partitions, in which case WLM creates and manages its own PRM configuration (PRM must be installed on the same system). You can use WLM to manage CPU resources across hard partitions and virtual partitions. WLM automatically moves cores between partitions based on the SLOs in the partitions. (Given the physical nature of hard partitions, the "movement" of cores among partitions is achieved by deactivating a core on one nPartition and then activating a core on another.) You can use WLM to manage resources within a virtual machine. On an Integrity Virtual Machines (Integrity VM) host, you can use WLM to manage

resources across partitions; within an Integrity VM guest, you can use WLM to manage the HP-UX resources but not using Instant Capacity (iCAP, formerly known as HP Instant Capacity on Demand, or iCOD), Pay per use (PPU), or virtual partition integration.

This white paper explains the capabilities and benefits of HP-UX WLM A.03.02 and HP-UX WLM A.03.02.01. WLM A.03.02 is available with the following operating system and hardware combinations:

Operating systems	Hardware
HP-UX 11i v1 (B.11.11)	HP 9000 servers
HP-UX 11i v2 (B.11.23)	HP Integrity servers and HP 9000 servers
HP-UX 11i v1 (B.11.11) and HP-UX 11i v2 (B.11.23)	Servers combining HP 9000 partitions and HP Integrity partitions (in such environments, HP-UX 11i v1 supports HP 9000 partitions only)

WLM A.03.02.01 is available with the following operating system and hardware combinations:

Operating systems	Hardware
HP-UX 11i v3 (B.11.31)	HP 9000 servers, HP Integrity servers, and servers combining HP 9000 partitions and HP Integrity partitions

What is HP-UX Workload Manager?

WLM is a software product that assesses resource usage in real time and then automatically allocates resources and manages application performance based on your SLOs and business priorities. WLM is available as a stand-alone product, but is also included in the HP-UX Mission Critical Operating Environment (MCOE) bundle.

Key uses of HP-UX WLM include:

- Using excess server capacity by consolidating multiple applications on fewer servers while ensuring that mission-critical applications still get the resources they need during peak demand times
- Reallocating system resources automatically in response to changing priorities, conditions that change over time (night/day, month-end processing, and so on), package movement in a cluster, resource demand, and application performance
- Automating the deployment of reserve capacity so that customers pay only for what they need when they need it
- Enabling higher utilization in clusters by enabling you to define, monitor, and enforce SLOs when a failure occurs, causing a workload to fail over a server or partition already running other workloads

You can use WLM within a whole server that can be clustered in an HP Serviceguard high-availability cluster, Extended Campus Cluster, Metrocluster, Continentalcluster, or a Hyperplex configuration. You can also use WLM on an Integrity VM host and within any individual Integrity VM (guest). You can use WLM within nPartitions and virtual partitions as well as across partitions.

WLM is most effective managing applications that are CPU-bound. It adjusts the CPU allocation of a group of processes known as a workload, basing adjustments on the current needs and performance of the applications in that workload.

To have WLM migrate resources among workloads as needed, you must define in the WLM configuration file one or more SLOs for each workload. In defining an SLO, you must specify its relative level of importance (priority). WLM enables you to prioritize the SLOs so that an SLO assigned a high priority has precedence over SLOs with a low priority. Typically, you also specify a usage goal to attain a targeted resource usage. If a performance measure (metric) is available, you can specify a metric goal. As the applications run, WLM compares the application usage or metrics against the goals. To achieve the goals, WLM then automatically adjusts CPU allocations for the workloads.

CPU resources can be allocated in shares (portions or time slices) of multiple cores or, when using WLM partition management or pSet management, in whole cores. WLM supports the logical CPU (Hyper-Threading) feature for pSet-based groups. Hyper-Threading is available on certain processors starting with HP-UX 11i v3 (B.11.31). A logical CPU is an execution thread contained within a core. Each core with Hyper-Threading enabled can contain multiple logical CPUs. WLM automatically sets the Hyper-Threading state for the default pSet o optimize performance. (The default pSet is where FSS groups are created.) When new pSets are created, they inherit the Hyper-Threading state that the system had before WLM was activated (because WLM may change the Hyper-Threading setting of the default pSet to optimize performance). Cores can be moved from one partition to another and will take on the Hyper-Threading state of their destination pSet. You can override the default state for cores assigned to a specific pSet-based group; you can also modify the Hyper-Threading state of the system. (Modifications to the Hyper-Threading state should not be made while WLM is running.) For more information, see the *HP-UX Workload Manager User's Guide* or the wlmconf(4) manpage.

Workload management across virtual partitions and nPartitions

WLM is optimized for moving cores among hosts such as virtual partitions and nPartitions. Using these hosts as workloads, WLM manages workload allocations while maintaining the isolation of their HP-UX instances. WLM automatically moves or "virtually transfers" cores among partitions based on SLOs and priorities that you define for the partitions.

With virtual partitions, WLM can automatically balance resources across the partitions. For example, if a processor is not being utilized within one virtual partition, WLM can deallocate it and reassign it to an alternate virtual partition that currently requires additional resources.

With nPartitions, which represent physical hardware, WLM does not move resources physically across partitions. With HP iCAP present, core movement is simulated by deactivating one or more cores in one nPartition and then activating cores in another nPartition.

The tools WLM uses to manage cores depend on the software enabled on the complex—such as HP iCAP, HP PPU, and virtual partitions.

For each host (nPartition or virtual partition) workload, you define one or more SLOs in the host's WLM configuration file. Once configured, WLM then automatically manages CPU resources to satisfy the SLOs for each workload. On an HP-UX system that has network connectivity to the partitions being managed by WLM, you configure the global arbiter (`wlmpard`). The global arbiter takes input from the WLM instances on the individual partitions and then moves cores between partitions as needed to better achieve the SLOs specified in the WLM configuration file that is active in each partition.

WLM can manage nested workloads, with workloads based on FSS groups and pSets inside virtual partitions inside nPartitions. For more information, see "Managing nested partitions" on page 23. In addition, you can integrate WLM with HP Serviceguard to reallocate resources in a failover situation according to defined priorities (for more information on integrating with HP Serviceguard, see "Using HP-UX Workload Manager with HP Serviceguard" on page 25).

Workload management within a single HP-UX instance

When you use WLM to manage workloads to divide resources within a single HP-UX instance, WLM manages SLOs for workloads that are based on PRM-based pSets or FSS groups. These workloads are usually referred to as "workload groups."

Note
In WLM interfaces (such as the displays provided by the WLM configuration wizard `wlmcw` or the WLM graphical user interface `wlminfo`), the term "workload group" typically refers to a workload; "workload" and "workload group" are often used interchangeably. However, workload groups must be created and referred to as such only when referring to a pSet or FSS group in one HP-UX instance.

You configure WLM by defining workload groups for the system or partition and then assigning specific applications, users, and UNIX groups to each workload group. You can also establish your own criteria for placing application processes in specified workload groups by defining process maps. A process map associates a specific workload group with a script or command and its arguments that gathers specified process IDs at 30-second intervals. At each interval, the identified processes are placed in their associated group. (In conjunction with HP Serviceguard Extension for SAP (SGeSAP), the WLM SAP Toolkit takes advantage of process maps, providing a script that enables you to place specified SAP processes in specific workload groups managed by WLM. For more information, see "Using HP-UX Workload Manager with SAP" on page 27.)

In addition, you can assign secure compartments to workload groups, using the HP-UX Security Containment feature. Secure compartments isolate files and processes. WLM can then automatically allocate resources for these secure compartments.

When you configure WLM, you define one or more SLOs for each workload group and prioritize them. To satisfy the SLOs for the workload groups, WLM automatically manages CPU resources within the HP-UX instance or partition; no allocation is made across partitions. Optionally, you can configure WLM to manage memory and disk bandwidth, although not in response to SLO performance. With real memory, WLM enables you to specify lower and upper limits on the amount of memory a workload group receives. You can statically assign disk bandwidth shares in the configuration file. If multiple users or applications within a workload group are competing for resources, standard HP-UX resource management determines the resource allocation.

Why use HP-UX Workload Manager?

The traditional open systems usage model has been one application running per server. This model has led to surplus capacity per server and to a proliferation of servers—too many to manage effectively. Typically, each server is sized to provide headroom for peak capacity and future growth. As new servers are introduced, the surplus capacity grows without providing any opportunity to share this excess capacity among the applications.

WLM, in collaboration with other HP Virtual Server Environment tools, significantly lessens the need to provide the same degree of headroom, allowing multiple applications on the same system to share the excess capacity. You can configure WLM to ensure that applications run at the performance level required to meet your business goals. By enabling you to consolidate data centers and multiple applications onto fewer servers, WLM significantly reduces your administration and computer resource expenses.

With WLM, you can:

- Run multiple workloads on a single system and maintain performance of each workload
- Prioritize workloads on a single system, adjusting the CPU allocations based on the workloads' goals
- Ensure that critical workloads have sufficient resources to perform at desired levels
- Manage by SLOs within and across virtual partitions or nPartitions
- Adjust resource allocations by automatically enabling or disabling SLOs based on time of day, system events, or application metrics
- Enable SLOs associated with an HP Serviceguard package failover
- Adjust the number of cores in a partition or pSet to meet SLOs
- Grant a workload dedicated CPU and memory resources in the form of a pSet
- Create Secure Resource Partitions (in conjunction with the HP-UX 11i v2 and v3 Security Containment feature) based on pSets or FSS groups, providing file and process isolation and automatic resource allocation
- Grant a workload CPU resources in direct proportion to a metric, such as number of processes in the workload
- Set minimum and maximum amounts of CPU resources and memory resources available to a workload
- Set and manage user expectations for performance
- Monitor resource consumption by applications or users through HP Glanceplus, WLM tools, or PRM tools

Service level objectives

A key reason for using WLM is its ability to manage SLOs. WLM automatically allocates CPU resources to workloads based on whether the application in the workload is underperforming, meeting, or outperforming its SLOs.

SLOs can be shares-based or goal-based:

- **Shares-based SLO**—You specify a fixed number of CPU shares or a shares-per-metric allocation for a workload. WLM tries to grant the workload the specified amount of CPU shares. (A CPU share is 1/100 of a single core or 1/100 of each core on a system, depending on the WLM mode of operation.) The actual amount of CPU shares granted to the workload is subject to the availability of CPU resources after the needs of higher priority SLOs have been met. You can specify maximum and minimum CPU bounds and an explicit shares request.

  A shares-based SLO consists of:
  - A workload
  - A shares allocation
  - A priority
  - Optional conditions (time of day, an event, and so on)
  - Optional CPU lower and/or upper bounds
- **Goal-based SLO**—You specify one of two goal types:
  - **Usage goals**—Goals based on a workload's utilization of its allocated CPU resources. If the processes in a workload are not using a certain amount of the workload's allocation, the allocation is decreased. If the processes are using a high percentage of the workload's allocation, the allocation is increased.
  - **Metric goals**—Goals based on a metric, such as processing at least x transactions per minute or having a response time under y seconds. Metric goals are based on performance data and require understanding of that data. HP recommends using usage goals instead—usage goals can be implemented immediately without prior knowledge of workload performance.

WLM grants CPU resources based on the usage goals or metric goals you specify. WLM dynamically changes CPU allocations for an associated workload of an SLO based on:

- SLO priority
- Limits placed on the SLO, such as minimum and maximum CPU bounds
- Performance or utilization of the workload
- Amount of CPU resources needed to meet the goal, as determined by the controller
- Available resources (including those after higher priority SLOs have been satisfied)

A goal-based SLO consists of:

- A workload
- A goal
- A priority
- Optional conditions (time of day, an event, and so on)
- Optional CPU lower and/or upper bounds

Prioritized SLOs

WLM enables you to prioritize the SLOs. When CPU resources are not sufficient to satisfy all SLOs, WLM grants CPU resources to the highest priority SLOs first. After the demands of the higher priority SLOs are satisfied, WLM grants any remaining resources to the lower priority SLOs. Valid priorities start at 1, with 1 being the highest priority.

A single workload can have multiple SLOs, each with a different priority. One SLO would be the high-priority, "must meet" goal, while the remaining SLOs would be lower priority, "meet if possible" goals (stretch goals). For example, an SLO might have a priority 1 goal of maintaining an allocation of at least two cores for a workload. Another SLO for the same workload could have a priority 2 goal to allocate three or four cores if available when the workload becomes very busy. This lower priority goal is met only after the priority 1 SLO for the workload and all other workloads are met.

SLO priorities do not have to be uniquely assigned—multiple SLOs can be granted the same priority, allowing more than one workload's objective to be top priority. This configuration can be beneficial when more than one workload is equally important. Typically, all the SLOs in a given configuration should not be assigned the same priority; otherwise, under a heavy system load, WLM might not be able to allocate CPU resources effectively when there are not enough CPU resources to satisfy all SLOs.

What is the ideal environment for HP-UX Workload Manager?

You will benefit most from WLM if your environment meets one or more of the following conditions:

- You run more than one workload concurrently on a server. The workloads can all run under one instance of HP-UX or in separate partitions, each with its own instance of HP-UX. These workloads could be multiple database servers, a database server and an applications server, or any other combination of workloads, provided that they are on PA-RISC servers (HP 9000) running HP-UX 11i v1 or later or on PA-RISC or Intel® Itanium®-based servers running HP-UX 11i v2 (B.11.23) or HP-UX 11i v3 (B.11.31), or later.
- You have CPU-intensive workloads that can be prioritized.
- You have an important workload with end-user performance requirements.
- You want consistent performance from applications under varying application and system loads.
- You run Serviceguard and need proper prioritization of workloads after a failover.
- You want more control over resource allocation than PRM provides.

HP-UX Workload Manager solutions

The following sections illustrate how WLM provides various business solutions. The SLOs are outlined without including the necessary configuration file syntax (for configuration file syntax, see the wlmconf(4) manpage or the *HP-UX Workload Manager User's Guide*).

SLOs that ensure a specified amount of CPU resources for workloads

The solutions in this section illustrate shares-based SLOs. They grant a workload a specified amount of CPU shares.

Reserving CPU resources all of the time

In this first example, the SLO requests a fixed allocation of CPU shares for the Marketing workload, reserving a portion of the CPU resources available. The 300 CPU shares being reserved equate to three cores (when managing SLOs for partitions, WLM equates each core to 100 shares). This SLO is priority 1 and is in effect at all times.

Workload: Marketing
Priority: 1
CPU shares: 300

Reserving CPU resources at specified times

This SLO also requests a fixed allocation, in this case, reserving 800 CPU shares. However, the associated workload contains a payroll application that runs only twice a month. Consequently, the SLO is enabled only twice a month, on the 15th and 28th.

Workload: Payroll
Priority: 1
CPU shares: 800
Condition: 15th and 28th

Reserving CPU resources based on an event or condition

The following SLO is enabled only part time when a specific condition is met (rather than at a specified date or time). The SLO is enabled only when a system accounting program is running, as indicated by some metric (a condition to be met). When the SLO is active, it works to enable the accounting program to complete quickly by reserving 600 CPU shares for the associated workload. When the program is completed, the SLO is disabled.

Workload: SysAcct
Priority: 1
CPU shares: 600
Condition: System accounting program is running

Reserving CPU resources in an HP Serviceguard failover

The following SLO is also enabled based on a metric. This example illustrates an SLO that is only active if the Serviceguard package pkgA is active on the current server. WLM provides a utility that generates a metric indicating whether a package is active. When the metric has value 1, the package is active, thus enabling the SLO. The SLO then causes WLM to attempt to allocate 300 CPU shares to the associated workload.

Workload: Failover_pkgA
Priority: 1
CPU shares: 300
Condition: pkgA is active on the current server

SLOs that dynamically allocate resources based on usage goals

The solutions in this section illustrate SLOs based on usage goals. They allocate resources dynamically, based on current demand or utilization. When the demand is high enough, more resources are allocated for the workload. When the demand falls below a certain level, unused resources can be made available for other workloads.

Allocating CPU resources dynamically based on utilization

Consider an SLO with a usage goal. Usage goals do not require a metric value. WLM tracks the metric itself. In this example, the workload (named Orders) is the collection of processes running in a virtual partition. WLM adjusts the CPU allocation of the virtual partition based on the CPU utilization of the workload within that partition. If the utilization is low (perhaps caused by fewer applications running), the CPU allocation for the partition is reduced, making more resources available to other partitions in the complex. If utilization is high, the partition receives a larger CPU allocation. Regardless of utilization, this SLO ensures that the partition is allocated at least 200 shares but no more than 800 shares, where 100 shares represents one core (when managing partitions, WLM equates one core to 100 shares).

Workload: Orders
Priority: 1
Goal: Match CPU allocation to consumption
Minimum CPU: 200 shares
Maximum CPU: 800 shares

Controlling sharing and borrowing of excess CPU resources

Consider a workload with multiple SLOs, each SLO having a usage goal. The associated workload, named Development, is owned by a department that funded 30% of the server. Consequently, that department expects to get 30% of the server when needed. In the following SLOs, 100 CPU shares represent the total CPU resources (cores) on the server. Therefore, 30% of the server is 30 CPU shares. When the Development workload is not busy, excess resources are available for sharing, as long as the Development workload is getting at least 15 shares. This condition is represented by the following SLO:

Workload: Development
Priority: 1
Goal: Match CPU allocation to consumption
Minimum CPU: 15 shares
Maximum CPU: 30 shares

When the workload becomes very busy, the Development workload must borrow from other workloads that might have excess resources. Based on usage patterns, the system administrator has agreed to let the workload borrow up to an additional 20 shares, if available, which allows the Development workload to access up to 50 CPU shares. The associated SLO is summarized as:

Workload: Development
Priority: 2
Goal: Match CPU allocation to consumption
Minimum CPU: 25 shares
Maximum CPU: 50 shares

Because this SLO is priority 2, it is met only after all priority 1 SLOs have been met.

Automatically resizing pSets

With multiprocessor systems, you can group processors together to form pSets. By creating pSets, you isolate CPU resources for users and applications.

WLM enables you to define workloads based on pSets. If you then specify SLOs for the workloads, WLM automatically adjusts the number of processors in the pSets based on progress toward the SLOs. In this example, the pSet workload named Batch gets five processors, but only between 10:00 p.m. and 4:00 a.m. At other times, Batch gets a default minimum of one processor. (Here, WLM is using "absolute CPU units," for which 100 shares represent one core.)

Workload: Batch
Priority: 1
Goal: 500 CPU shares (which corresponds to five cores)
Condition: Time is between 10:00 p.m. and 4:00 a.m.

Automatically resizing virtual partitions

HP virtual partitions enable you to partition a server, with each partition consisting of one or more cores. Each virtual partition runs its own instance of the HP-UX operating system. With WLM, you can automate the resizing of partitions.

Consider a system with two virtual partitions. The SLO for the Apps workload in partition 1 has a higher priority than the SLO for the Dev workload in partition 0. When CPU usage for the Apps workload in partition 1 reaches a certain point, WLM automatically migrates a core from partition 0 to partition 1 to satisfy the higher priority SLO of Apps.

Workload: Apps (partition1)
Priority: 1
Goal: Match CPU allocation to consumption

Workload: Dev (partition0)
Priority: 2
Goal: Match CPU allocation to consumption

Automatically resizing nPartitions using HP Instant Capacity cores

If you have HP Instant Capacity (iCAP) software configured on each partition, you can configure WLM to "move" cores to the partitions where they are most needed. Given the hardware isolation, the cores are not physically moved. WLM deactivates cores on one partition and then activates cores on another partition, thereby giving the appearance of moving cores without incurring a charge for an additional core.

Consider a system with two nPartitions, in which nPartition 0 runs the production, customer-accessible version of a shopping website, while nPartition 1 runs the test version of this website. The SLO for the Production workload on partition 0 has a higher priority than the SLO for the Test workload on partition 1. When CPU utilization for the Production workload reaches a certain point, WLM automatically migrates cores from nPartition 1 to nPartition 0 to satisfy the higher priority SLO.

Workload: Production (nPartition 0)
Priority: 1
Goal: Match CPU allocation to consumption

Workload: Test (nPartition 1)
Priority: 2
Goal: Match CPU allocation to consumption

Optimizing the use of HP Temporary Instant Capacity and HP Pay per use

If you have WLM on an HP Temporary Instant Capacity (TiCAP) system, you can configure WLM to minimize the costs of using these resources by optimizing the amount of time the resources are used for meeting the needs of your workloads.

Similarly, HP Pay per use (PPU) provides capacity as needed, basing payment on actual metered or monitored use of that capacity. Using WLM with a system running a supported version of PPU, WLM increases or decreases the capacity automatically, allocating the minimum number of cores needed to satisfy SLOs. By minimizing the number of active cores, WLM reduces your costs.

You can use WLM to manage TiCAP or PPU on stand-alone systems, as well as across partitions. For more information on using TiCAP or PPU with WLM, see "Using HP-UX Workload Manager with HP Temporary Instant Capacity and HP Pay per use" on page 24.

SLOs that maintain performance based on metric goals

HP recommends working with metric goals for advanced WLM users only. Metric goals are based on performance data and require the ability to collect and understand that data. Because you can implement usage goals immediately without requiring prior knowledge of workload performance, HP recommends using them instead of metric goals. The SLO in this section shows a simple example based on a metric goal. For more information on using metric goals and collecting performance data, see "Managing application performance with metric goals" on page 15. For more details, see the *HP-UX Workload Manager User's Guide*, which includes an appendix on advanced WLM usage with performance metrics.

Allocating CPU resources per metric

In the following SLO, the webserver workload gets five CPU shares for each active process. This metric is easily retrieved through the WLM interface to Glanceplus. Thus, five active processes would obtain 25 CPU shares. However, the workload's CPU shares request is not allowed to fall below 10 shares or exceed 90 shares.

Workload: webserver
Priority: 1
Goal: Five CPU shares for each active process in the workload
Minimum CPU: 10 shares
Maximum CPU: 90 shares

HP-UX Workload Manager operational overview

WLM automatically allocates system resources to maintain application performance, even during changing system conditions and fluctuating workload demands. To enable WLM to determine the appropriate resource allocation, you can create SLOs for each workload.

Specify usage goals to allow WLM to give a workload more CPU resources when the system is busy and to take them away when the system is idle. WLM does this by tracking an internally collected CPU utilization metric. Usage goals can be easily established, even when you first use the WLM product, and these goals do not require advanced knowledge of workload performance metrics, unlike metric goals.

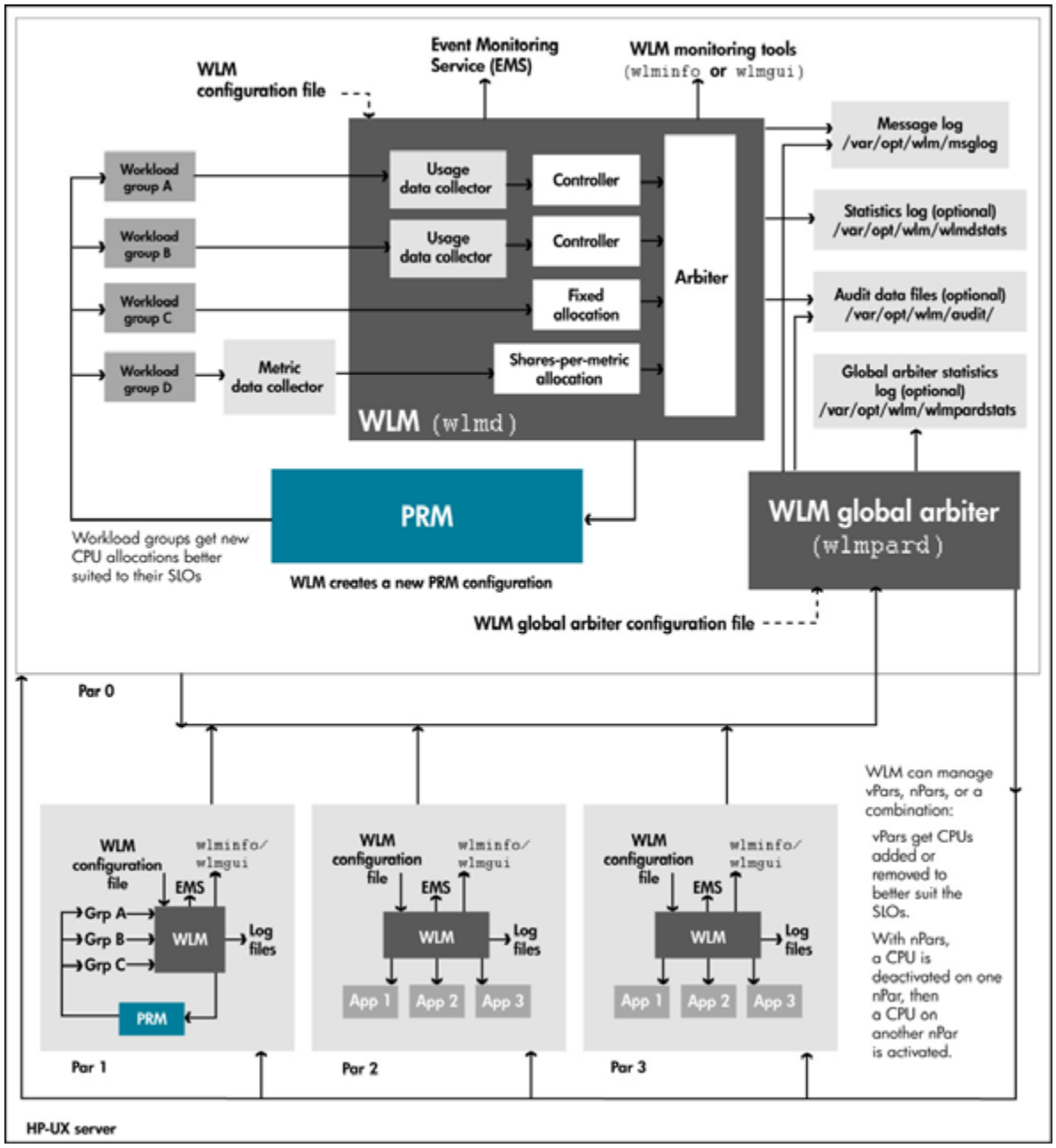
You can specify metric goals if performance data for the workload is available and understood. For each metric goal, you select a performance metric (such as a response-time goal) to measure the extent to which the goal is being met, exceeded, or underachieved. In addition, you choose how to send the data to WLM. Utilities that send data to WLM are called data collectors. For information on sources of data and sending the data, see "Managing application performance with metric goals" on page 15.

As the applications run, WLM compares the goals and metrics for each application to determine the appropriate CPU allocations.

Figure 1 shows the data flow on a system managed by WLM. It shows WLM running in each partition on the system. If you are using WLM on a system without partitions, then consider Par 0 as a stand-

alone system, focusing only on the processes shown in Par 0. If you want to understand how WLM manages resources and SLOs across partitions, consider Par 0 as one of four partitions along with Par 1, Par 2, and Par 3; the WLM global arbiter (defined in the partition Par 0) determines resource allocations for each of the four partitions.

Figure 1. Process flow diagram of WLM



As shown in Figure 1, the main functional process flow for the WLM design is as follows:

1. Workloads (and/or workload groups) and their goal-based or shares-based SLOs are defined in the WLM configuration file. The WLM configuration file also provides the path names for any data collectors. WLM reads the configuration file and starts the data collectors.

2. For each application with a usage goal, WLM creates a controller. The controller is an internal component of WLM. Each controller tracks its application's actual CPU usage (utilization of allocated CPU resources). No user-supplied metrics are required. The controller requests an increase or decrease to the workload's CPU allocation to achieve the usage goal.
3. For the application running with a metric goal, a data collector reports the application's metrics. The measurement, for example, might be transaction response times for an online transaction processing (OLTP) application.
4. For each metric goal, WLM creates a controller. You assign a data collector to track and report a workload's performance. Each controller receives the metric from the respective data collector. The metric is compared to the metric's goal to determine how a workload's application is performing. If the application is performing below expectations, the controller then requests an increase in CPU allocations for the workload; if the application is performing above expectations, the controller requests a decrease in CPU allocations for the workload.
5. For applications without goals, WLM requests CPU resources based on the CPU shares requested in the SLO definitions. These requests could be for fixed allocations or for shares-per-metric allocations, with the metric coming from a data collector.
6. The arbiter, an internal module of WLM (not to be confused with the global arbiter used for managing resources across partitions), collects all the requests for CPU shares. These requests come from controllers or, if allocations are fixed, from the SLO definitions. The arbiter satisfies the requests based on priority. If resources are insufficient for every application to meet its goals, the arbiter satisfies the highest priority requests first. If multiple SLOs at the same priority cannot be satisfied, WLM raises the CPU allocation for each SLO's associated workload to the same level or to the SLO's CPU request—whichever is smaller.
7. Optionally, with PRM resource management available for a single HP-UX instance, WLM determines how much memory to distribute to meet the minimum memory requests and then, if any memory remains, divides it among the workload groups with active SLOs.
8. For managing resources within a single HP-UX instance, WLM then creates a new PRM configuration that applies the new CPU and (optional) memory shares for the various workload groups.
9. For managing CPU resources (cores) across partitions, the WLM process flow described in the preceding steps is duplicated in each partition. The WLM instance in each partition regularly requests from the WLM global arbiter a certain number of cores for its partition. The global arbiter then uses these requests to decide how to allocate cores to the various partitions. Next, it adjusts each partition's number of cores to better meet the SLOs in the partition.

   For partitions, you can bypass creating workloads (workload groups), treating the partition itself (and applications that run on it) as the workload. Par 2 and Par 3 show this scenario.

The monitoring and logging processes shown in Figure 1 include the following:

- The status of the SLOs and information about the performance of WLM are sent to the Event Monitoring Service (EMS). Using an EMS client such as System Administration Manager (SAM) or System Management Homepage (SMH), which is an enhanced version of SAM, you can choose from several notification methods (such as email, SNMP traps, TCP, User Datagram Protocol (UDP), and OPC Messaging) for receiving events of specific interest.
- The WLM monitoring command line utility `wlminfo` or the graphical user interface `wlmgui` enables you to get a variety of types of WLM information.
- WLM keeps you up to date on the operations of its daemon by updating the message log in /var/opt/wlm/msglog.
- WLM adds data to the statistics log in /var/opt/wlm/wlmdstats if enabled through the `wlmd -l` option. The data collectors continue to feed application metrics to WLM, which periodically calculates new resource allocations and performs any needed PRM reconfiguration.

- WLM produces audit data in /var/opt/wlm/audit if the WLM configuration file is activated using the `-t` option to `wlmd`.
- WLM produces audit data in /var/opt/wlm/wlmpardstats/ if the WLM global arbiter configuration file is activated using the `-l` option to `wlmpard`.

Note
WLM's network interfaces are designed to operate correctly to defend against attacks in a moderate to high threat environment, such as a demilitarized zone (DMZ). (A DMZ is a computer or small subnetwork located between a trusted internal network, such as a private corporate LAN, and an untrusted external network, such as the public Internet.) You may use network protections such as firewalls to provide an additional level of defense and to give you additional time to react if a security loophole is found.

Managing application performance: Quick start

You can quickly and easily set up WLM to manage application performance with shares-based SLOs and usage goal-based SLOs. These SLOs do not require user-supplied metrics.

To set up either type of SLO:

1. Set up the application's host as a workload or assign the application to a workload group.
2. Define an SLO for that workload:
   a. Specify the priority of the SLO.
   b. (Optional) Specify the minimum and maximum CPU requests for the SLO.
   c. For a shares-based SLO, specify a shares-per-metric allocation (using a data collector included with WLM) or a fixed-allocation request for the SLO.
      For a usage-goal-based SLO, specify the usage goal (using the default range of percentages or specifying a range. WLM adjusts the workload's allocation, giving it more CPU resources when utilization falls below the minimum percentage and fewer CPU resources when it rises above the maximum).

Managing application performance with metric goals: Advanced HP-UX Workload Manager management

Using metric goals, you can fine-tune how workloads are managed. For each SLO with a metric goal, WLM collects metrics from either built-in or user-supplied data collectors. Use metric goals only if performance data is available and understood. Otherwise, use usage goals.

With metric-goal-based SLOs, you must:

1. Set up the application's host as a workload or assign the application to a workload group.
2. Define an SLO for that workload:
   a. Specify the priority of the SLO.
   b. (Optional) Specify the minimum and maximum CPU requests for the SLO.
   c. Specify the metric goal.
3. Specify the data collector that will provide the metric.

Methods for collecting data include:

- Glanceplus data collectors (included with WLM)
- WLM Oracle® data collector and WLM data collectors for other third-party applications
- Scripted data collectors launching the `wlmsend` command
- Scripted or binary data collectors providing data on `stdout`
- Binary data collectors providing data through the native WLM data collection API

The data is available from many sources:

- **Glanceplus metrics**—You can use these metrics by specifying one of the Glanceplus data collectors in your configuration file. These data collectors are included with WLM.
- **Oracle database metrics**—You can retrieve these metrics using the WLM Oracle data collector, which is part of the WLM Toolkits (WLMTK) that come free with WLM. For more information about WLMTK, see the white paper, "Using HP-UX Workload Manager most effectively with your critical applications" at http://h20338.www2.hp.com/hpux11i/wlm.toolkits.pdf.
- **Various third-party applications**—You can obtain metrics for a number of third-party applications through the WLMTK. In addition to the toolkit for Oracle databases, WLMTK provides toolkits for Apache, BEA WebLogic Server, SAP, and SNMP.
- **Existing metrics**—If you already maintain application- or system-specific metrics, you can use that data.
- **Dummy transactions**—You can use these transactions to give a rough feel for the application performance. These transactions require the same types of processing as the real transactions of interest. The process generating the dummy transactions would then report the performance to WLM.
- **ARM-instrumented applications**—You can use the `wlmrcvdc` utility with the `glance_tt` command to send WLM any data collected by an ARM agent. ARM* is the Application Response Measurement standard for application instrumentation.

After the data is collected, it can be sent to WLM in one of two ways:

- WLM `wlmrcvdc` and `wlmsend` utilities (which provide command line, shell script, and PERL program interfaces)
- WLM C language application programming interface (API)

* Application Response Measurement (ARM) is the industry standard for application instrumentation. For information on ARM, see http://www.opengroup.org/management/arm.htm.

HP-UX Workload Manager configuration

When you configure WLM, you define the workloads, their goals, and the means to measure their progress toward goals.

You can configure WLM in several ways:

- Use a text editor to create a configuration or modify an example configuration.
- Use the configuration wizard.
- Use the WLM graphical user interface.

NOTE
Usage of the WLM wizard and graphical user interface requires Java™ Runtime Environment version 1.4.2 or later. For PRM-based configurations, PRM C.03.00 or later is required (to take advantage of the latest updates to WLM, use the latest version of PRM available).

Figure 2 shows the WLM configuration wizard.

Figure 2. The WLM configuration wizard

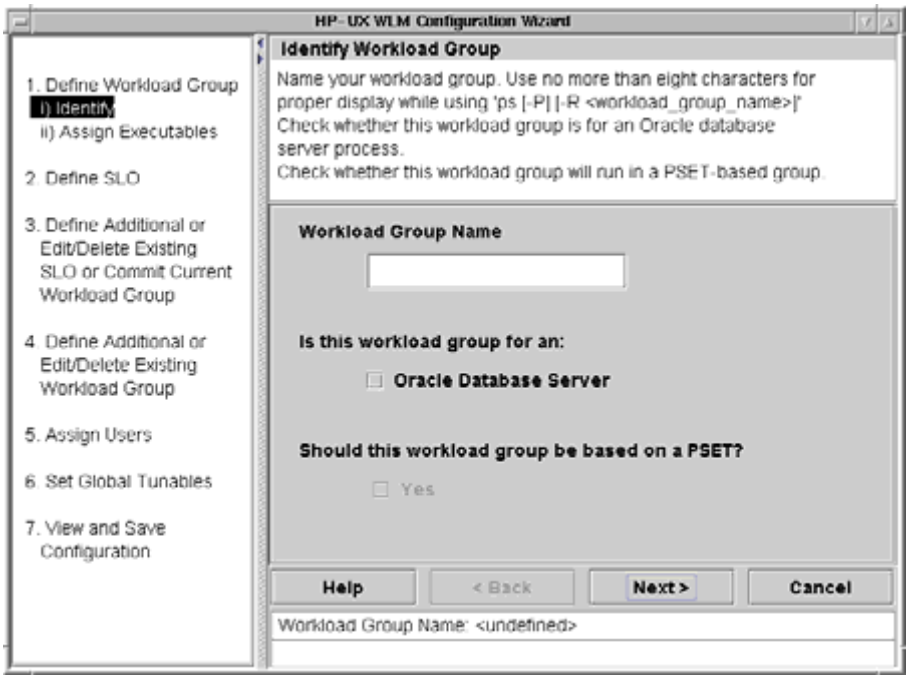
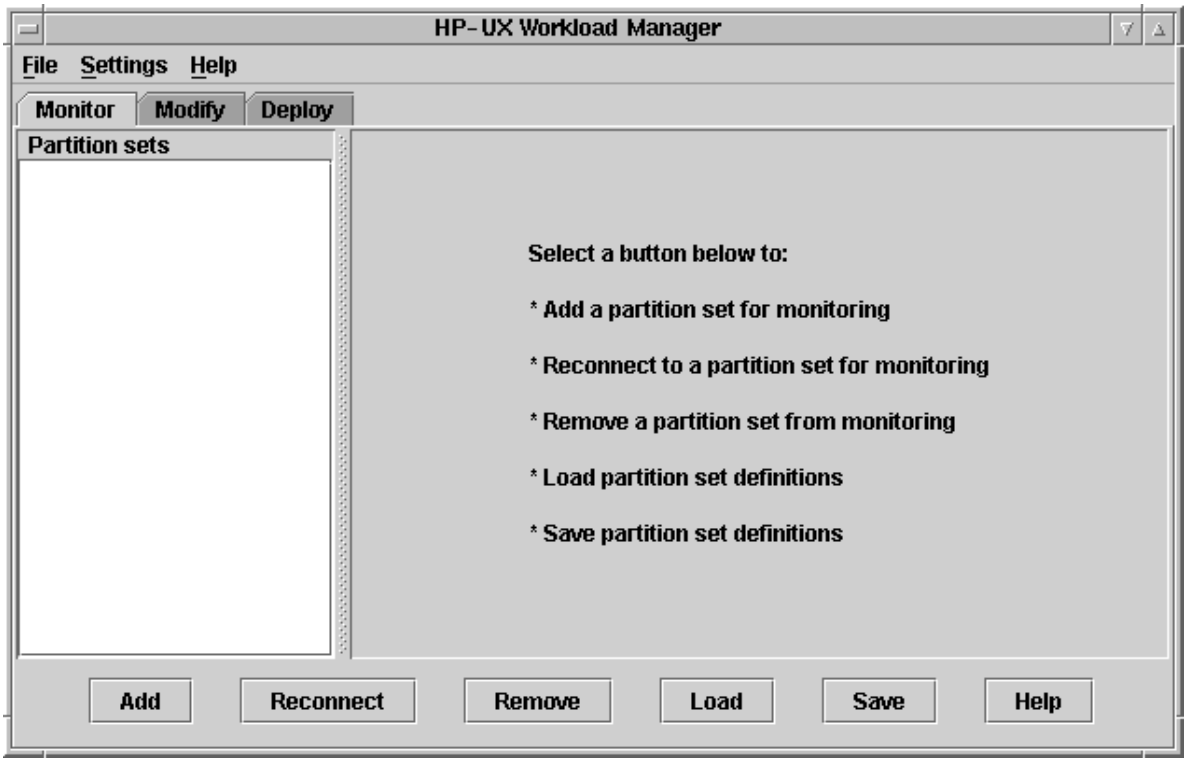


Figure 3 shows the WLM graphical user interface.

Figure 3. The WLM graphical user interface



Testing your configuration file without HP-UX Workload Manager controlling the system

WLM provides a passive mode that enables you to see approximately how WLM will respond to a given configuration, without putting WLM in charge of your system resources. Using this mode, you can analyze the behavior of your configuration, with minimal effect on the system. Besides being useful in understanding and experimenting with WLM, passive mode can be helpful in capacity-planning activities.

Passive mode is available with both the WLM daemon and the WLM global arbiter daemon through the `-p` option.

Monitoring your service level objectives

WLM enables you to monitor SLO compliance and other information through its `wlminfo` command line interface and the `wlmgui` graphical user interface.

Monitoring with the command line interface

A few examples of the command line interface are shown in the following discussion.

In the first example, the focus is on SLOs. By entering `wlminfo slo`, you can see the number of shares requested in the Req (request) column. The Shares column indicates the number of shares actually allocated. From the State column, you can see that the two priority 1 SLOs are passing, while the priority 2 SLO is failing. The Concern column helps to highlight information that could possibly indicate configuration issues or other issues that require attention. In this example, the `Priority` entry indicates that the shares granted are less than what was requested because WLM is satisfying requests from other SLOs at the same or higher priority. Note that you can use the `–v` option with the `wlminfo slo` command to display the goals of the SLOs (if any) and the metrics showing how the workloads are performing relative to the goals.

```
% /opt/wlm/bin/wlminfo slo

SLO Name             Group        Pri    Req Shares  State Concern
finance_deposit      Finance        1     52     53  PASS
sales_analysis       Sales          1     20     20  PASS
marketing_analysis   Marketing      2     30     26  FAIL  Priority
```

In the next example, the focus is on workloads. By entering `wlminfo group`, you can see by the CPU Util column ("Util" represents utilization) that all the user-defined workloads are consuming CPU resources, with the OTHERS group having the highest consumption. The command also displays (as of WLM A.03.02) a Mem Util column indicating the memory utilization of each group for which memory management is in effect.

```
% /opt/wlm/bin/wlminfo group

 Workload Group PRMID CPU Shares CPU Util Mem Shares Mem Util State

 OTHERS             1     432.00   171.34      40.00    30.21 ON
 g_nice             2      84.00    49.22      15.00     4.89 ON
 g_nightly          3       0.00     0.00       0.00     0.00 OFF
 g_team             4       6.00     0.00      15.00     0.00 ON
 g_apache           5      72.00     0.00      29.00     0.00 ON
 _IDLE_             7       6.00     0.00       1.00     0.00 ON
```

Note that beginning with WLM A.03.02 you can use the `–v` option with the `wlminfo group` command to see the minimum and maximum percentage of CPU resources (`gmincpu` and `gmaxcpu`) and the minimum and maximum of memory (`gminmem` and `gmaxmem`) assigned for each group, if applicable, in the active configuration. (If a minimum or maximum is not assigned, `wlminfo` displays "-" in place of a numeric value; if memory management is not being used for a group, `wlminfo` displays a "-" in the Mem Shares column.) The width of the display exceeds 80 characters, so each row might wrap to the next line.

```
% /opt/wlm/bin/wlminfo group –v

Workload Group PRMID CPU Shares CPU Min CPU Max CPU Util Mem Shares Mem Min Mem Max Mem Util State

OTHERS             1  432.00         -      -    171.34      40.00       -       -   30.21     ON
g_nice             2   84.00        10     80     49.22      15.00      10      20    4.89     ON
g_nightly          3    0.00         -      -      0.00          -      10      20       -     ON
g_team             4    6.00         2      4      0.00      15.00       2      10    0.00     ON
g_apache           3   72.00        10     55      0.00      29.00       4      15    0.00     ON
_IDLE_             7    6.00         -      -      0.00       1.00       1       -    0.00     ON
```

In the following example, the focus is on partitions (hosts). The `wlminfo par` command displays information about each nPartition or virtual partition that WLM manages. The Intended Cores column shows the number of cores that WLM wants to allocate to the partition, while the Cores column shows the current number of active cores. The number of intended and active cores is usually the same, except when WLM is in the process of modifying a partition or is operating in passive mode. (In passive mode, the intended core allocation is not made; the partition retains the current number of active cores.) The Cores Used column shows the CPU (core) utilization of the partition. The Interval column shows the WLM interval, which is the frequency at which WLM checks for new performance data for the workload and then adjusts core allocations.

```
% /opt/wlm/bin/wlminfo par

Hostname       Intended Cores  Cores  Cores Used  Interval
north                       2      2         1.3         6
south                       3      3         2.1         6
east                        1      1         0.4         6
west                        2      2         1.7         6
northwest                   3      3         2.3         6
northeast                   2      2         1.4         6
```

The `wlminfo host` command displays information pertaining to the local host (by default) or a specified host, including the number of cores on the host, the amount being used, and the WLM interval. In the following example, the local host has two cores, with 1.7 of the cores being used:

```
% /opt/wlm/bin/wlminfo host

Hostname       Cores  Cores Used  Interval
localhost          2         1.7         6
```

The `wlminfo metric` command displays information about the metrics used in the current WLM configuration. It displays the PID of the data collector providing the metric to WLM, the status of the metric value, and its current value. In the following example, the WLM daemon, `wlmd`, with PID 27570 is providing the metric `_CPU_Sales` for the usage goal. The other metric is updated by a process outside WLM. The State column indicates whether the metric value was updated in the interval (`NEW`), no value has been received for the metric since the WLM daemon started (`INIT`), or the metric's value was not updated in the interval (`OLD`). The last column shows the value received for the metric. The value for the `_CPU_Sales` metric (22.695808) represents the percentage of its CPU allocation that it is using. The metric value is well under the goal of using at least 60% of the workload's allocation. Consequently, WLM might soon decrease the workload's allocation, depending on the current allocation and needs of other SLOs.

```
% /opt/wlm/bin/wlminfo metric

Metric Name                        PID State Value
fin_app.deposit.trans_se         27571 NEW   1.8
_CPU_Sales                       27570 NEW   22.695808
```

The `wlminfo` command enables you to specify multiple SLOs, groups, metrics, partitions, or processes. For example, to display data about groups Finance and Marketing, you can enter the following `wlminfo group` command:

```
% /opt/wlm/bin/wlminfo group -g Finance -g Marketing
```

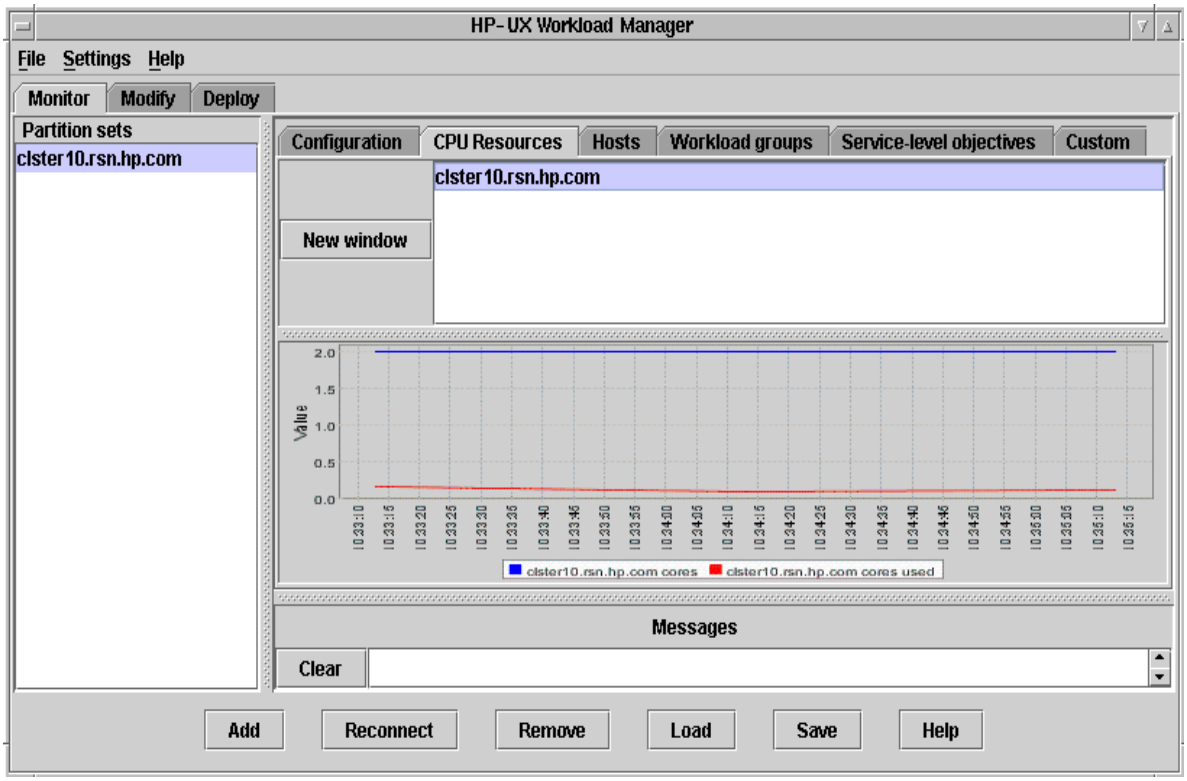
Monitoring with the graphical user interface

To use the WLM graphical user interface, enter the `wlmgui` command:

```
% wlmgui
```

The WLM graphical user interface provides monitoring in addition to configuration features. The Monitor tab of the graphical user interface provides graphs of allocation and usage for the workloads (you can also graph a workload's minimum CPU and maximum CPU values (`gmincpu` and `gmaxcpu`). For example, the CPU Resources tab displays how many CPU resources (cores) have been active for a particular partition over time. In Figure 4, active CPU resources remained constant for the given partition.

Figure 4. Monitoring WLM



Forming Secure Resource Partitions

The HP-UX 11i v2 (and later) Security Containment feature provides process and file isolation inside secure compartments. The combination of WLM and Security Containment, which creates Secure Resource Partitions, is powerful. WLM manages resource allocations for each application instance. However, each application instance can still interact with other processes, access files, and even system processes. When Security Containment is combined with WLM, applications can be consolidated on a single operating system, while ensuring that any application, file, and system interaction is carefully controlled. Using Security Containment, you can ensure that application instances cannot access processes or files from other applications or the system unless a rule is created to specifically allow the interaction. This functionality ensures that multiple application instances run securely in a consolidated environment, providing the benefits of consolidation while preserving the security of a scale-out environment (where each application runs by itself in an operating system instance on its own server).

Using HP-UX Workload Manager for server consolidation

When consolidating servers, one of the challenges is resource allocation. How do you ensure that one application will not steal needed resources from another?

WLM enables you to specify fixed allocations to ensure each application gets a defined resource amount. Additionally, you can define and prioritize one or more SLOs for each application's associated workload. The workloads are then given the resources they need to achieve the SLOs. This capability is significant for several reasons:

- **Higher system utilization**—By giving each application's workload only what it needs when it needs it, the excess capacity is shared more efficiently.
- **SLOs are met even during peak demand**—Resources are applied when needed so that even during times of peak demand, SLOs are met.
- **Prioritization of SLOs**—A workload with a higher priority SLO is given what it needs to achieve its goal first and then any remaining resources are allocated to lower priority SLOs. Each workload has at least one SLO so that it can be prioritized against other workloads. Furthermore, a workload can have multiple SLOs of various priorities so that it meets a minimal goal when resources are tight but can achieve greater goals when more resources are available.

Using the defined SLOs, WLM determines the amount of CPU resources needed to achieve each SLO. It then allocates CPU resources to each SLO's workload based on the SLO's priority. This is important for:

- Server consolidation
- Higher resource utilization
- Ease of server management
- Sharing of excess capacity
- Application/workload prioritization

Using HP-UX Workload Manager for auditing and billing

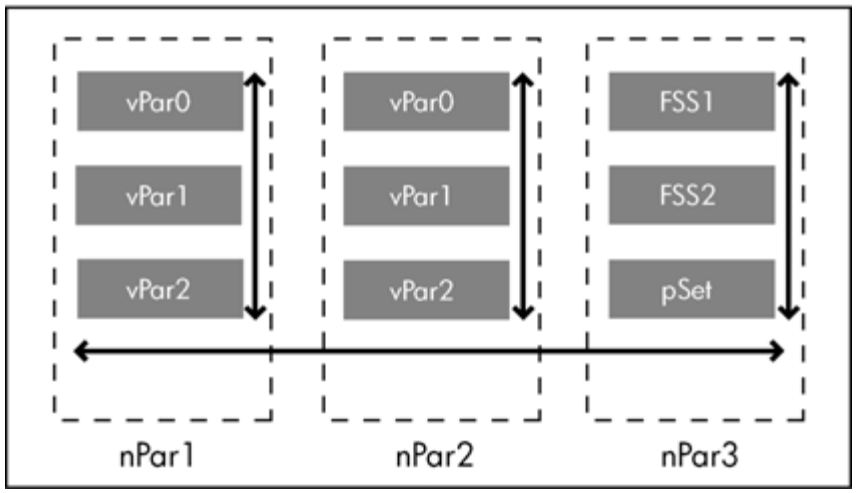
WLM produces audit information when you activate a configuration using the `-t` option with either the WLM daemon `wlmd` or the WLM global arbiter daemon `wlmpard`.

After you have activated a configuration using `-t`, use the `wlmaudit` command to display the audit data. The `wlmaudit` command enables you to specify a date range for the data to display. By default, the output is plain text. However, you can display output in formatted HTML as well.

Managing nested partitions

You can manage any combination of FSS and pSet workload groups inside virtual partitions inside nPartitions. (Instant Capacity must be installed on the nPartitions to simulate movement of cores among the nPartitions.) For example, consider the system shown in Figure 5, consisting of three nPartitions. Two nPartitions each have three virtual partitions, while the third partition has two FSS workload groups and a pSet-based workload group. Each virtual partition or workload group consists of a workload with an SLO. WLM enables you to focus on the workloads and their goals. With the WLM management of nested partitions, the resources needed to satisfy those goals can come from other workloads in the nPartition, as well as from other nPartitions. The arrows in the figure indicate the possible sharing.

Figure 5. Managing nested partitions



Using HP-UX Workload Manager on multiple servers

WLM manages workloads on individual servers. To manage workloads on multiple servers, install and configure WLM on each server.

WLM can be integrated with HP Serviceguard by storing the WLM configuration file in a file system shared by all nodes in the cluster and then activating the configuration on each node independently.

Alternatively, you can use the HP Integrity Essentials Global Workload Manager (gWLM) to manage workloads across multiple servers. gWLM enables you to define resource-sharing policies that can be used across servers. For more information on gWLM, go to http://www.hp.com/go/gwlm.

Using HP-UX Workload Manager with HP Temporary Instant Capacity and HP Pay per use

HP offers the Temporary Instant Capacity (TiCAP) and Pay per use (PPU) features. You can use WLM to manage the use of TiCAP or PPU resources to ensure your workloads use only the amount of CPU resources needed to meet their SLOs and that you pay only for the resources you actually use. WLM can manage these resources in stand-alone (nonpartitioned) systems as well as across partitions.

TiCAP activates CPU capacity in a temporary "calling-card fashion" such as in 30-day increments (where one day equals 24 hours for one core). You purchase a codeword to obtain rights to use certain Instant Capacity cores for a preset number of days. This codeword is applied to a system so that you can turn on and off any number of these cores as long as your prepaid amount of temporary capacity days has not expired. WLM supports version 6 or later of TiCAP.

Using WLM on a system with the PPU software, CPU capacity is increased or decreased automatically to support peak anticipated demand, basing payment for the HP server on actual metered or monitored usage of that capacity. With PPU version 4, capacity can be increased or decreased by whole cores as needed, with billing determined by the number of active cores. Beginning with PPU version 5, all cores on a PPU system are active, and billing is based on your percentage of usage for those cores. Beginning with PPU version 7, which includes version 5 capabilities, billing can also be based on the number of active cores on the system, with WLM activating only those cores that are needed. WLM integrates with PPU versions 4 and 7, or later.

To take advantage of this optimization provided by TiCAP or PPU, set your SLOs in the WLM configuration file and then set up a WLM global arbiter configuration that specifies the `utilitypri` keyword. (For information on using the `utilitypri` keyword, see the wlmparconf(4) and wlmpard(1M) manpages.) WLM and the TiCAP or PPU software automatically adjusts the number of active cores to the smallest number of cores needed to satisfy the SLOs. Using the `utilitypri` keyword also ensures that WLM maintains compliance with your TiCAP usage rights: when your prepaid amount of temporary capacity expires, WLM no longer attempts to use temporary resources. (By default, when 15 or fewer days of temporary capacity are available, WLM stops using TiCAP. In this case, you must purchase extra capacity. As of WLM A.03.02, you can change this 15-day default by setting the WLM global arbiter `utility_reserve_threshold` keyword. For more information, see the wlmparconf(4) and wlmpard(1M) manpages.)

Using HP-UX Workload Manager with HP Integrity Virtual Machines

HP Integrity Virtual Machines is a robust soft partitioning and virtualization technology that provides operating systems isolation, shared CPU resources (with subcore granularity), shared I/O, and automatic dynamic resource allocation. It is available for HP-UX 11i v2 and later, running on HP Integrity servers.

Given a system with Integrity VM installed, you can run WLM both on the Integrity VM host and in an Integrity VM (guest), but each WLM runs as an independent instance. Note the following guidelines:

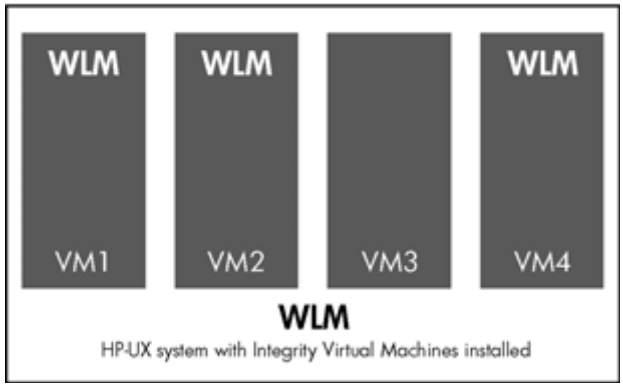
- On the system itself (on the Integrity VM host)—Use a strictly host-based WLM configuration (a configuration designed exclusively for moving cores across partitions or for activating TiCAP or PPU cores). WLM will not run with FSS groups or pSet-based groups on Integrity VM hosts where guests are running.
- Inside any Integrity VM (guest)—Do not use Instant Capacity, PPU, or HP-UX Virtual Partitions (vPar) integration. However, guests can take advantage of CPU resources added to the Integrity VM host

by Instant Capacity, TiCAP, or PPU. The WLM interval should be set to a value greater than 60 seconds to ensure a fair allocation of CPU resources for FSS groups.

In either case, the minimum number of cores allocated to a WLM host must be greater than or equal to the maximum number of virtual CPUs (vCPU count) assigned to each VM guest. For more information, see the *HP-UX Workload Manager User's Guide*.

Figure 6 illustrates how WLM can be used with HP Integrity Virtual Machines. In this example, WLM runs on the Integrity VM host and in three of the four guests.

Figure 6. WLM and Integrity Virtual Machines



For more information on HP Integrity VM, go to http://www.hp.com/go/vse and click the Solutions components tab.

Using HP-UX Workload Manager with HP Serviceguard

Using WLM with Serviceguard, you can:

- Decrease the performance impact of a failover caused by software being unavailable in active/active configurations (where both the primary and standby servers are active)
- Increase utilization of a system that receives failover packages
- Simplify the configuration of failover scripts
- Simplify routine or scheduled maintenance

Incorporate your WLM configuration in the HP Serviceguard configuration so that if an application is failed over to an alternate nPartition or virtual partition, WLM automatically recognizes this and reconfigures all applications within the partition, according to predefined rules and SLOs. You can incorporate your WLM configuration in the HP Serviceguard configuration by storing the WLM configuration file in a file system shared by all nodes in the cluster. Activate the WLM configuration on each node independently.

WLM provides many features that support integration with Serviceguard. For more information, as well as step-by-step procedures for integrating the products, see the white paper, "More efficient high availability through manageability (integrating Serviceguard and HP-UX Workload Manager)," available from the Information Library at http://www.hp.com/go/wlm.

Using HP-UX Workload Manager with HP Systems Insight Manager and HP Servicecontrol Manager

HP Systems Insight Manager and one of its predecessors, HP Servicecontrol Manager, provide a single point of administration for multiple HP-UX systems. The WLM integration with HP Systems Insight Manager and HP Servicecontrol Manager enables you to perform the following activities from the Central Management Server (CMS) on nodes in the cluster that have WLM installed:

- Enable HP-UX WLM
- Disable HP-UX WLM
- Start HP-UX WLM
- Stop HP-UX WLM
- Reconfigure HP-UX WLM
- Distribute HP-UX WLM configuration files to the selected nodes
- Retrieve currently active HP-UX WLM configuration files from the nodes
- Verify the syntax of HP-UX WLM configuration files on either the CMS or the selected nodes
- View, rotate, and truncate HP-UX WLM log files

Using HP-UX Workload Manager with Oracle databases

HP-UX WLM Oracle Database Toolkit (ODBTK) simplifies getting metrics on Oracle database instances into WLM, which enables you to better manage Oracle instances. ODBTK is part of the WLM Toolkits product, which is included with WLM. Benefits of ODBTK include the ability to:

- Keep response times for your transactions below a given level by setting response-time SLOs
- Increase an instance's available CPU resources when a particular user connects to the instance
- Increase an instance's available CPU resources when more than *n* users are connected
- Increase an instance's available CPU resources when a particular job is active
- Give an instance *n* CPU shares for each process in the instance
- Give an instance *n* CPU shares for each user connection to the instance

Using HP-UX Workload Manager with Apache

WLM can help you manage and prioritize Apache-based workloads through the use of the HP-UX WLM Apache Toolkit (ApacheTK). ApacheTK, a component of the WLM Toolkits product that comes with WLM, enables you to:

- Separate Apache from Oracle database instances
- Separate Apache from batch work
- Isolate a resource-intensive common gateway interface (CGI) workload
- Separate all Apache Tomcat workloads from other Apache workloads
- Separate two departments' applications using two Apache instances
- Separate module-based workloads with two Apache instances
- Manage Apache CPU allocations by performance goal

For more information, see the white paper, "Using HP-UX Workload Manager with Apache-based Applications," which offers various use cases. This paper is installed on systems with WLM at

/opt/wlm/toolkits/apache/doc/apache_wlm_howto.html. It is also available at http://h20338.www2.hp.com/hpux11i/cache/325340-0-0-0-121.html.

Using HP-UX Workload Manager with BEA WebLogic Server

HP-UX WLM BEA WebLogic Server Toolkit (WebLogicTK) provides a data collector called `wlmwlsdc` that tracks metrics indicating how busy WebLogic Server instances are. The toolkit is a component of the WLM Toolkits product, which is included with WLM.

WebLogicTK offers the ability to:

- Manually provide a single instance with an increasing amount of CPU resources in the form of a dynamic pSet for benchmarking
- Separate an instance from other workloads, as well as from other instances, while automatically maintaining performance using a dynamic pSet based on:
  - Group CPU usage
  - Server instance queue metrics

For more information, see the wlmwlsdc(1M) manpage and the white paper, "Using HP-UX Workload Manager with BEA WebLogic Server," which offers various use cases. This paper is installed on systems with WLM at /opt/wlm/toolkits/weblogic/doc/weblogic_wlm_howto.html. It is also available at http://h20338.www2.hp.com/hpux11i/cache/325462-0-0-0-121.html.

Using HP-UX Workload Manager with SAP

WLM and its SAP Toolkit (SAPTK), in conjunction with the HP Serviceguard Extension for SAP (SGeSAP) product, take advantage of the WLM process map feature to enable you to identify different SAP processes and place them into separate workloads. The SAPTK process ID identifier, called `wlmsapmap`, enables you to identify and isolate entire SAP instances (or just subsets of an instance's processes) as a separate workload. For example, you can use `wlmsapmap` to collect SAP batch and dialog (interactive) processes and place them in separate workload groups. WLM can prioritize and assign specific SAP processes to workload groups. You can:

- Isolate an entire SAP system as a workload
- Isolate an SAP instance as a workload
- Separate SAP processes from an instance and place them in a workload

For more information, see the white paper, "Using HP-UX Workload Manager with SAP," available at /opt/wlm/toolkits/sap/doc/sap_wlm_howto.html and also at http://www.hp.com/go/wlm.

Using HP-UX Workload Manager with SNMP

WLM and its SNMP Toolkit (SNMPTK) provide integration with the HP-UX SNMP agent. SNMPTK includes a data collector, called `snmpdc`, which fetches values from an SNMP agent so that you can use them as metrics in your WLM configuration. SNMPTK provides easy access to SNMP agent metrics that you can use in your WLM configuration to:

- Drive SLO goals
- Set up shares-per-metric allocations
- Enable and disable SLOs

For more information, see the example configuration files that come with SNMPTK, available in /opt/wlm/toolkits/snmp/config/. In addition, see the wlmtk(5) and snmpdc(1M) manpages.

HP-UX Workload Manager, HP Global Workload Manager, and HP Process Resource Manager

The HP products HP-UX WLM, HP Global Workload Manager (gWLM), and HP Process Resource Manager (PRM) should not be used to manage the same system at the same time. Using these products to control resources at the same time can cause inconsistent behavior and undesirable performance. However, you can use WLM and PRM at the same time if the PRM configuration uses FSS groups only (no pSet-based groups) and the WLM configuration is strictly host-based. (A strictly host-based configuration does not include a `prm` structure; it is designed exclusively for moving cores across partitions or for activating TiCAP or PPU cores.) You might want to use both products on the same system to take advantage of certain features of PRM that are not included with the latest release of WLM, such as PRM's `CPUCAPOFF` mode, enabled with the `prmconfig -M CPUCAPOFF` command. (In this mode, a PRM group's upper bound for CPU resource consumption is determined by the `CAP` value, available on HP-UX 11i v3 (B.11.31) or later. For more information, see the *HP Process Resource Manager User's Guide* or prmconfig(1M).)

Note
As of WLM A.03.01, PRM is no longer included with the WLM bundle. If PRM C.03.00 or later is already on the machine on which you must install or upgrade WLM, you can continue to manage FSS and pSet-based workload groups (just as if PRM had been installed with WLM). If you are installing WLM for the first time on a machine, you can use a strictly host-based configuration (no FSS or pSet workload groups). However, to manage FSS and pSet-based workload groups, you must install PRM (C.03.00 or later) separately.

Related information

The following references provide useful background information on related products and topics.

- HP-UX Workload Manager (HP-UX WLM)—http://www.hp.com/go/wlm
- *HP-UX Workload Manager User's Guide*—http://www.docs.hp.com/hpux/netsys/
  (Click the **HP-UX Workload Manager** link on this page or scroll down to the HP-UX Workload Manager section.)
- WLM white papers—http://www.hp.com/go/wlm (Click the **Information library** link on the right.)
  - "Using WLM: A quick reference"
  - "Resizing partitions automatically with HP-UX Workload Manager"
  - "More efficient high availability through manageability (integrating Serviceguard and HP-UX Workload Manager)"
- HP-UX Workload Manager Toolkits white paper—http://h20338.www2.hp.com/hpux11i/wlm.toolkits.pdf
  The Toolkits consist of:
  - HP-UX WLM Oracle Database Toolkit (ODBTK)
  - HP-UX WLM Toolkit for Apache (ApacheTK)
  - HP-UX WLM BEA WebLogic Server Toolkit (WebLogicTK)
  - HP-UX WLM Toolkit for SAP Software (SAPTK)

  - HP-UX WLM SNMP Toolkit (SNMPTK)
- *HP-UX Workload Manager Toolkits User's Guide*—http://www.docs.hp.com/hpux/netsys/ (Click the **HP-UX Workload Manager** link on this page or scroll down to the HP-UX Workload Manager section.)
- HP Process Resource Manager—http://www.hp.com/go/prm
- HP Virtual Server Environment—http://www.hp.com/go/vse
- HP OpenView Performance Agent—http://www.openview.hp.com/
- ARM and ARM API—http://www.opengroup.org/management/arm.htm
- HP Systems Insight Manager—http://www.hp.com/go/hpsim
- HP Instant Capacity and HP Pay per use—http://www.hp.com/go/utility

For more information

For more information on HP-UX Workload Manager, contact any HP worldwide sales offices or see the HP website at http://www.hp.com/go/wlm.

To learn more about the Adaptive Enterprise and virtualization, see http://www.hp.com/go/virtualization.

5982-4354EN, Rev. 3 September 2006