HP-UX Workload Manager overview

invent

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, but one of the primary reasons is that customers often have one application per server, and they size that server for peak load of typically three to five times the average utilization.

Resource optimization is one of the goals of HP's Adaptive Enterprise strategy—a strategy for helping customers to synchronize business and IT to capitalize on change. Virtualization is a cornerstone of HP's approach to helping customers realize the promise of becoming an adaptive enterprise. It is an approach to IT that pools and shares resources so utilization is optimized and supply automatically meets demand.

Part of the virtualization model is the Virtual Server Environment. It offers the broadest virtualization capabilities in the industry today and the only goal-based policy engine, HP-UX Workload Manager. HP-UX Workload Manager, or WLM, ties together virtualization techniques including partitioning, resource management, on-demand resources, and clustering—linking them back to business goals and service-level objectives. It allows companies to consolidate multiple applications onto a single server, which minimizes server administration expenses and makes more efficient use of available resources.

Key uses of HP-UX WLM include:

- Using excess capacity on servers by consolidating multiple applications on fewer servers while ensuring the mission-critical applications still get the resources they need in times of peak demand
- Automatically reallocating system resources in response to changing priorities, conditions that change over time (night/day, month-end processing, and so forth), package movement in a cluster, resource demand, and application performance

HP-UX WLM provides this functionality by automating many features of Hewlett-Packard’s Process Resource Manager (PRM), HP-UX Processor Sets (PSETs), HP-UX Virtual Partitions (vPars), and hard partitions (nPars) that use iCOD.

When using WLM, the system administrator:

1. Defines workload groups for each workload
2. Places the applications or users that define the workloads into their own workload groups (the workload’s processes share the resources WLM makes available to the workload group)
3. Creates one or more SLOs for each workload group

In defining an SLO, the system administrator specifies a priority. The administrator can also specify a goal to attain some specific measure (metric) or a targeted resource usage. Such goals can be thought of as metric goals or usage goals. As the applications run, WLM compares the application metrics or usage against the goals. WLM then automatically adjusts CPU allocations for the workload groups to achieve the goals.

This document explains the capabilities and benefits of HP-UX WLM Version A.02.02, which is available for PA-RISC servers running HP-UX 11.0 or HP-UX 11i V1.0 (B.11.11) and for Itanium®-based servers running HP-UX 11i V2.0 (B.11.23).

What is HP-UX Workload Manager?

HP-UX Workload Manager is a software product that provides automatic resource allocation and application performance management through the use of prioritized SLOs.

WLM manages workload groups as defined in its configuration file. The WLM administrator assigns applications and users to workload groups and sets up one or more SLOs for each group. WLM then manages each workload group's CPU resources according to the SLOs. WLM can also manage real memory, although not in response to SLO performance. Disk bandwidth can be statically allocated 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.

WLM is integrated with various products to provide better interoperability. These products include Oracle, Apache, HP-UX Virtual Partitions, HP-UX Processor Sets, GlancePlus, iCOD (Instant Capacity on Demand), Pay Per Use, Servicecontrol Manager, and Serviceguard.

Why use HP-UX Workload Manager?

The traditional open systems usage model has been one application running per server. This has led to a proliferation of servers—too many servers to manage effectively. Generally, each server is sized to provide "headroom" for peak capacity and future growth. With each new server, the surplus capacity grows—and no ability exists to share this excess capacity among the applications. Therefore, many companies want to reduce administration expenses and use computing resources more efficiently by consolidating their data centers onto fewer systems and consolidating multiple applications onto a single server.

With the tools in the HP Partitioning Continuum, there is less need to provide the same degree of headroom, allowing the excess capacity to be shared among the different applications. As one of the partitioning tools, WLM provides the ability to:

- Prioritize workloads on a single system, adjusting the associated workload groups' CPU resources based on their goals
- Manage by service-level objectives within and across virtual partitions (vPars) or node partitions (nPars)
- Adjust resource allocations by automatically enabling or disabling SLOs based on time of day, system events, or application metrics
- Enable SLOs associated with a Serviceguard package failover
- Ensure critical workload groups have sufficient resources to perform at desired levels
- Adjust the number of CPUs in a processor set or virtual partition to meet SLOs
- Grant a workload group CPU in direct proportion to a metric, such as number of processes in the workload group
- Grant a workload group dedicated CPU and memory resources in the form of a processor set
- Set and manage user expectations for performance
- Run multiple workloads on a single system and maintain performance of each workload
- Monitor resource consumption by applications or users through HP GlancePlus, WLM tools, or PRM tools
- Set minimum and maximum amounts of CPU and memory available to a workload group

Service-level objectives (SLOs)

A key reason for using WLM is its ability to manage service-level objectives.

After defining a workload group, the system administrator specifies one or more SLOs for each workload group. WLM allocates CPU to workload groups based on whether the group's application is underperforming, meeting, or overperforming its SLOs.

SLOs can be shares-based or goal-based. With a shares-based SLO, WLM simply tries to grant the associated workload group a certain amount of the CPU by allocating it CPU shares. (A CPU share is 1/100 of a single CPU or 1/100 of each CPU on a system, depending on WLM's mode of operation.) With goal-based SLOs, WLM actively changes the associated workload group's CPU allocation to best meet the SLO. These SLOs are based on one of two goal types:

- Metric goals—goals based on a metric, such as processing at least *x* transactions per minute or having a response time under *y* seconds
- Usage goals—goals based on a workload group's utilization of its allocated CPU: If the processes in a workload group are not using a certain amount of the group's allocation, the allocation is decreased; similarly, if the processes are using a high percentage of the group's allocation, the allocation is increased

A goal-based SLO consists of:

- A workload group
- A goal
- A priority
- Constraints (minimum/maximum CPU)
- Conditions (time of day, an event, ...) if desired

A shares-based SLO contains most of the above elements, but does not include a goal.

Prioritized SLOs

Another great reason for using WLM is that it allows the system administrator to prioritize the SLOs. Valid priorities start at 1, with 1 being the highest priority.

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 can be beneficial when multiple workloads are equally important. Typically, all the SLOs in a given configuration should not be assigned the same priority; otherwise, under heavy system load, WLM may not be able to allocate CPU optimally when there is not enough to meet all SLOs.

A single workload can have multiple SLOs, each with a different priority. One SLO would be the high priority, "must meet" goal and the remaining SLOs would be lower priority, "meet if possible" goals (stretch goals). For example, a priority 1 goal might be for the workload to keep response time under three seconds. The priority 2 goal might then be to keep response time under one second. This lower priority goal might be met, but only after the priority 1 SLOs of its associated workload group and all other workload groups are met.

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:

- Run more than one workload concurrently on a server
  The workloads may 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 running HP-UX 11.0 or later or on Itanium®-based servers running HP-UX 11i V2.0 (B.11.23)
- Have workloads that can be prioritized
- Have an important workload with end-user performance requirements
- Desire consistent performance from applications under varying application and system loads
- Run Serviceguard and need to ensure proper prioritization of workloads after a failover
- Want more control over resource allocation than PRM provides

Solutions HP-UX Workload Manager provides

The following sections illustrate how WLM provides various business solutions. For information on some of the configurations behind the SLOs below, see "SLO configuration examples" on page 17.

Reserving a portion of the server all of the time

In this first example, the SLO requests a fixed CPU allocation for the workload defined by the group Marketing, reserving a portion of the server's CPU. This SLO is priority 1 and is in effect at all times.

Group: Marketing
Priority: 1
CPU shares: 30

Reserving a portion of the server at specified times

The next SLO also requests a fixed allocation, reserving 80 CPU shares in this case. However, the associated workload contains a payroll application that only runs twice a month. Consequently, the SLO is enabled only twice a month, on the 6th and 21st.

Group: Payroll
Priority: 1
CPU shares: 80
Condition: 6th or 21st

Reserving a portion of the server based on an event

The following example shows another SLO that is enabled only some of the time. Rather than a date or time though, the condition is a metric. The SLO is enabled only when a system accounting program is running, as indicated by some metric. Once the SLO is active, it works to ensure the accounting program completes quickly by reserving 60 CPU shares for the associated workload group. When the program is completed, the SLO is disabled.

Group: SysAcct
Priority: 1
CPU shares: 60
Condition: System accounting program is running

Ensuring resources in a Serviceguard failover

Similarly, the next SLO is 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 25 CPU shares to the associated workload group.

Group: Failover_pkgA
Priority: 1
CPU shares: 25
Condition: pkgA is active on the current server

Allocating CPU as used

Now consider an SLO with a usage goal. Usage goals do not require a metric value; WLM tracks the metric itself. In this example, WLM adjusts the group's CPU allocation based on the CPU utilization of applications executing in the Orders group. If the utilization is low (due perhaps to fewer of the group's applications running), then the CPU allocation for the Orders group is reduced, making more resources available to other groups. If utilization is high, the group receives a larger CPU allocation. Regardless, the allocation should be at least 20 shares but no more than 80 shares.

Group: Orders
Priority: 1
Goal: Match CPU allocation to consumption
Min CPU: 20 shares
Max CPU: 80 shares

Controlling how workloads share and borrow excess cycles

Here, the workload has multiple SLOs, where each SLO has a usage goal. The associated workload, defined by the group Development, funded 30% of the server. Consequently, the group expects to get 30% of the server when needed. In WLM's default mode of operation, 100 CPU shares represent the total CPU on the server; so, 30% of the server is 30 CPU shares. When the workload is not busy, some of the excess resources are available for sharing, as long as the Development group always gets at least 15 shares. This is represented by the following SLO:

Group: Development
Priority: 1
Goal: Match CPU allocation to consumption
Min CPU: 15 shares
Max CPU: 30 shares

When the workload becomes very busy, the Development group would like to borrow from other groups that may have excess resources. Based on usage patterns, the system administrator has agreed to let the group borrow up to an additional 30 shares if available. This allows the Development group to access up to 60 CPU shares. The associated SLO is shown below:

Group: Development
Priority: 2
Goal: Match CPU allocation to consumption
Min CPU: 30 shares
Max CPU: 60 shares

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

Allocating CPU per metric

WLM allows you to grant a workload group a certain amount of CPU per unit of a metric. In our example below, the workload group gets 5 CPU shares for each active process in the workload group. (This metric is easily retrieved through WLM's interface to GlancePlus.) Thus, 5 active processes would result in 25 CPU shares. However, the group's CPU shares request is not allowed to fall below 10 shares or exceed 90 shares.

Group: webserver
Priority: 1
Goal: 5 CPU shares for each active process in the workload group
Min CPU: 10 shares
Max CPU: 90 shares

Meeting peak demand

In this next example, the SLO has a metric goal. The SLO becomes active, based on the metric number_of_active_jobs, once the server is sufficiently busy. It attempts to keep the average job completion time under ten seconds.

Group: Jobs
Priority: 1
Goal: Average completion time < 10 seconds
Min CPU: 40 shares
Max CPU: 90 shares
Condition: number_of_active_jobs > 100

Automatically resizing processor sets (PSETs)

With multi-processor systems, you can group processors together to form processor sets, also known as PSETs. By creating PSETs, you isolate CPU resources for users and applications.

WLM allows you to define workload groups based on PSETs. If you then specify SLOs for these workload groups, WLM automatically adjusts the number of processors in the PSETs based on progress toward the SLOs. Here is a PSET workload group Batch that gets five processors, but only between 10pm and 4am. At other times, Batch gets a default minimum of one processor. (In this example, WLM is using "absolute CPU units," where 100 shares represent one CPU.)

Group: Batch
Priority: 1
Goal: 500 CPU shares (which corresponds to five CPUs)
Condition: Time is between 10pm and 4am

Automatically resizing virtual partitions (vPars)

HP’s virtual partitions allow you to partition a server, with each partition consisting of one or more CPUs. Each virtual partition runs its own instance of the HP-UX operating system. You can adjust the partition size by the dynamic addition and deletion of CPUs. With WLM, you can automate the resizing of partitions.

Consider a system with two vPars. The SLO for the Apps workload group on vPar 1 has a higher priority than the SLO for vPar 0's workload group.

When CPU usage for the Apps group reaches a certain point, WLM automatically migrates a CPU from vPar 0 to vPar 1 to satisfy the higher priority SLO.

Group: Apps
Priority: 1
Goal: Match CPU allocation to consumption

Automatically resizing node partitions (nPars) via iCOD CPUs

HP's node partitions allow you to partition a server. Each partition provides hardware isolation as well as operating-system isolation. If you have the iCOD (instant Capacity on Demand) software configured on each partition, you can configure WLM to “move” CPUs to the partitions where they are most needed. (Given the hardware isolation, the CPUs are not literally moved: WLM deactivates a CPU on one partition, then activates a CPU on another partition, thereby giving the appearance of moving CPUs without incurring a charge for an additional CPU.)

Consider a system with two nPars. nPar 0 is running the production, customer-accessible version of a shopping web site. nPar 1 runs the test version of this web site. The SLO for the Production workload group on nPar 0 has a higher priority than the Test SLO for nPar 1's workload group.

When CPU usage, or utilization, for the Production group reaches a certain point, WLM automatically migrates a CPU from nPar 1 to nPar 0 to satisfy the higher priority SLO.

Group: Production
Priority:1
Goal: Match CPU allocation to consumption

Managing duration

Numerous applications are short-lived. If these applications are resource-intensive, they can greatly hinder the progress of other applications on the system. However, rather than purchasing a larger server to handle these spikes in resource usage, you can use WLM with its Duration Management Toolkit (DMTK). This toolkit is beneficial in batch environments, SAS® environments, and many other environments. Essentially, you specify a desired duration for a workload. DMTK then:

- Increases the workload’s CPU resources if the workload may not complete in time
- Decreases the workload’s CPU resources if the workload is completing too quickly

When using DMTK in a SAS environment, you can fine-tune the duration management using a SAS macro available in the WLM Toolkit for SAS Software (SASTK).

Both these toolkits come with the WLM Toolkits product.

HP-UX Workload Manager operational overview

The WLM product automatically allocates system resources to maintain application performance, even during changing system conditions and fluctuations in workload demand. For WLM to determine the appropriate resource allocation, the system administrator places applications in workload groups then creates one or more SLOs for each workload group. Among other items, an SLO includes a priority and an optional metric or usage goal.

Along with each metric goal, the WLM system administrator selects metrics to measure the extent to which the goal is being met, exceeded, or underachieved. In addition, the system administrator chooses 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 14.

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

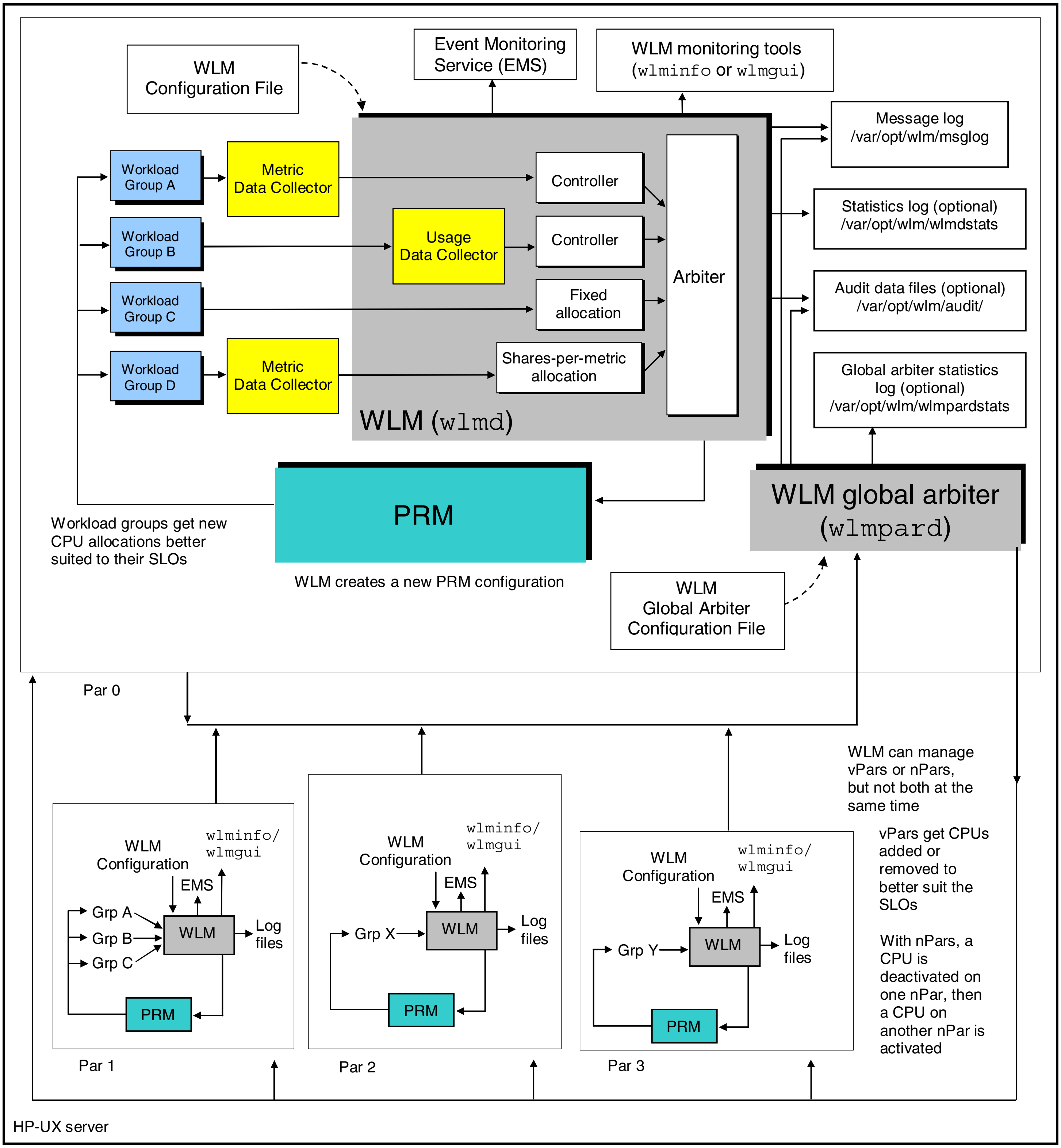
In addition to metric goals, the system administrator can specify usage goals. With usage goals, WLM gives a workload group more CPU when it is busy and takes it away when it is idle. WLM does this by tracking an internally collected CPU utilization metric.

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, focus on Par 0 to understand how WLM controls resources on your system to ensure SLOs are met.

You can use WLM within and across:

- vPars
- nPars that use iCOD software

Figure 1. Process flow diagram of WLM


WLM Configuration File
Event Monitoring Service (EMS)
WLM monitoring tools (wlminfo or wlmgui)
Message log /var/opt/wlm/msglog
Workload Group A
Metric Data Collector
Controller
Workload Group B
Usage Data Collector
Controller
Statistics log (optional) /var/opt/wlm/wlmdstats
Workload Group C
Fixed allocation
Arbiter
Audit data files (optional) /var/opt/wlm/audit/
Workload Group D
Metric Data Collector
Shares-per-metric allocation
Global arbiter statistics log (optional) /var/opt/wlm/wlmpardstats
WLM (wlmd)
PRM
WLM global arbiter (wlmpard)
Workload groups get new CPU allocations better suited to their SLOs
WLM creates a new PRM configuration
WLM Global Arbiter Configuration File
Par 0
WLM can manage vPars or nPars, but not both at the same time
vPars get CPUs added or removed to better suit the SLOs
With nPars, a CPU is deactivated on one nPar, then a CPU on another nPar is activated
WLM Configuration
wlminfo/ wlmgui
EMS
Grp A
Grp B
Grp C
WLM
Log files
PRM
Par 1
WLM Configuration
wlminfo/ wlmgui
EMS
Grp X
WLM
Log files
PRM
Par 2
WLM Configuration
wlminfo/ wlmgui
EMS
Grp Y
WLM
Log files
PRM
Par 3
HP-UX server

Referring to Figure 1, the functional process flow for the WLM design is:

- The WLM configuration file specifies the goal-based or shares-based SLOs for each workload group. This file also provides the pathnames for the data collectors. WLM reads the configuration file and starts the data collectors.
- With the WLM configuration file activated using the `-t` option to `wlmd`, WLM produces audit data in /var/opt/wlm/audit/.
- For each application with a metric goal, as the application runs, a data collector for that application reports the application's metrics. The measurement, for example, might be transaction response times for an online transaction processing (OLTP) application.
- For each metric goal, WLM creates a controller. The controllers are an internal component of WLM. Each controller receives the metric from the respective data collector. The metric is compared to the goal for that metric to determine if the application is overperforming or underperforming. The controllers then request increases to the CPU allocations of workload groups with underperforming applications or decreases to the CPU allocations of workload groups with overperforming applications.
- For each application with a usage goal, WLM creates a controller. Each controller tracks its application's actual CPU usage (utilization of allocated CPU); no user-supplied metrics are required. The controller requests an increase or decrease to the workload group's number of CPU allocation to bring the utilization percentage within a configurable range.
- For applications without goals, WLM requests CPU 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.
- The arbiter, an internal module of WLM, collects all the requests for CPU shares. These requests come from controllers or from the SLO definitions. The arbiter satisfies the requests based on priority. If there are not enough resources 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 group to the same level, or to the SLO's CPU request—whichever is smaller.
- Optionally, WLM determines how much memory to distribute to meet the minimum memory requests and then, if any memory remains, divides it among the groups with active SLOs.
- WLM then creates a new PRM configuration applying the new CPU and (optional) memory shares. Also, WLM adds data to the statistics log /var/opt/wlm/wlmdstats if enabled through the `wlmd -l` option. The data collectors continue to feed application metrics to WLM, which at intervals calculates new resource allocations and performs any needed PRM reconfiguration.
- If configured for partition management, the WLM instance in each partition regularly requests from the WLM global arbiter a certain number of CPUs for its partition.

  The global arbiter adds data to the statistics log /var/opt/wlm/wlmpardstats if enabled through the `wlmpard -l` option.

  With the WLM global arbiter configuration file activated using the `-t` option to `wlmpard`, WLM produces audit data in /var/opt/wlm/audit/.
- The WLM monitoring command-line utility `wlminfo` or the graphical user interface allow you to get a variety of types of WLM information.
- 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), a system administrator can choose from a number of notification methods (such as email, SNMP traps, TCP, UDP, and OPC Messaging) for receiving events of specific interest.
- WLM keeps you up-to-date on the operations of its daemon by updating the message log /var/opt/wlm/msglog.

What types of service-level objectives are supported?

WLM provides automatic PRM, PSET, vPar, and nPar settings based on the defined SLOs. The supported types of SLOs are:

- Shares-based SLO—This SLO type allows an administrator to specify a fixed number of shares or a shares-per-metric allocation for a workload group without specifying a goal. The actual amount of CPU granted to the workload group is subject to the availability of CPU resources after the needs of higher priority SLOs have been met.
- Goal-based SLO—Goal-based SLOs cause WLM to grant more CPU or take away CPU based on metrics or usage. These SLOs have either metric goals or usage goals. Metric goal-based SLOs are suitable for applications that can generate metrics. For example, OLTP applications are good candidates for metric goal-based SLOs.

  Usage-goal-based SLOs attempt to keep a workload group's utilization of its CPU allocation within a given percentage range. With a usage goal, a workload group's CPU allocation is reduced if its processes are not consuming enough of the group's currently allotted CPU, allowing other workload groups to consume more CPU if needed. Similarly, if the workload group's processes are using a high percentage of the group's allocation, more CPU is requested for it.

  WLM automatically changes CPU allocations for goal-based SLOs to better achieve their stated goals. The actual CPU allocation granted is based on the amount of CPU needed to meet the goal as determined by the controller, the limits placed on the SLO, and the availability of CPU resources after the needs of all higher priority SLOs have been met.

When a workload group has to accommodate a "must meet" goal and optional, lower-priority stretch goals, you can assign multiple SLOs, as long each SLO is a fixed-allocation or goal-based SLO.

For examples of how to specify SLOs in the WLM configuration file, see "SLO configuration examples" on page 17.

Managing application performance: quick start

The system administrator 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 data collectors.

To set up a shares-based SLO, the system administrator:

1. Assigns the application to a workload group
2. Defines an SLO for that workload group
3. Specifies the priority of the SLO
4. Specifies a shares-per-metric allocation (using a data collector included with WLM) or a fixed-allocation request for the SLO

With usage-goal-based SLOs, the system administrator:

1. Assigns the application to a workload group
2. Defines an SLO for that workload group
3. Specifies the priority of the SLO
4. Specifies minimum and maximum CPU requests for the SLO
5. Specifies the usage goal

This type of SLO attempts to keep the workload group's utilization of its CPU allocation within a given percentage range, giving the group more CPU when required and taking CPU away when not needed.

Managing application performance with metric goals

Using metric goals, the system administrator 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.

With metric goal-based SLOs, the system administrator:

1. Assigns the application to a workload group
2. Defines an SLO for that workload group
3. Specifies the priority of the SLO
4. Specifies minimum and maximum CPU requests for the SLO
5. Specifies the metric goal
6. Specifies the data collector that will provide the metrics

There are several methods for collecting data:

- GlancePlus data collectors (included with WLM)
- WLM Oracle data collector and WLM's data collectors for other third-party applications
- Scripted data collectors invoking 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 a number of sources:

- GlancePlus metrics
  You can easily 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 product that comes free with WLM.
- Various third-party applications
  Metrics can be easily obtained for a number of third-party applications through the WLM Toolkits (WLMTK). In addition to the toolkit for Oracle databases, WLMTK provides toolkits for Apache, BEA WebLogic Server, and SAS.
- Existing metrics
  If you already maintain application-specific or system-specific metrics, you can use that data.
- Dummy transactions
  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
  ARM[*] is the Application Response Measurement standard for application instrumentation.

[*] Application Response Measurement (ARM) is the industry standard for application instrumentation. For information on ARM, visit www.cmg.org/regions/cmgarmw.

HP-UX WLM configuration

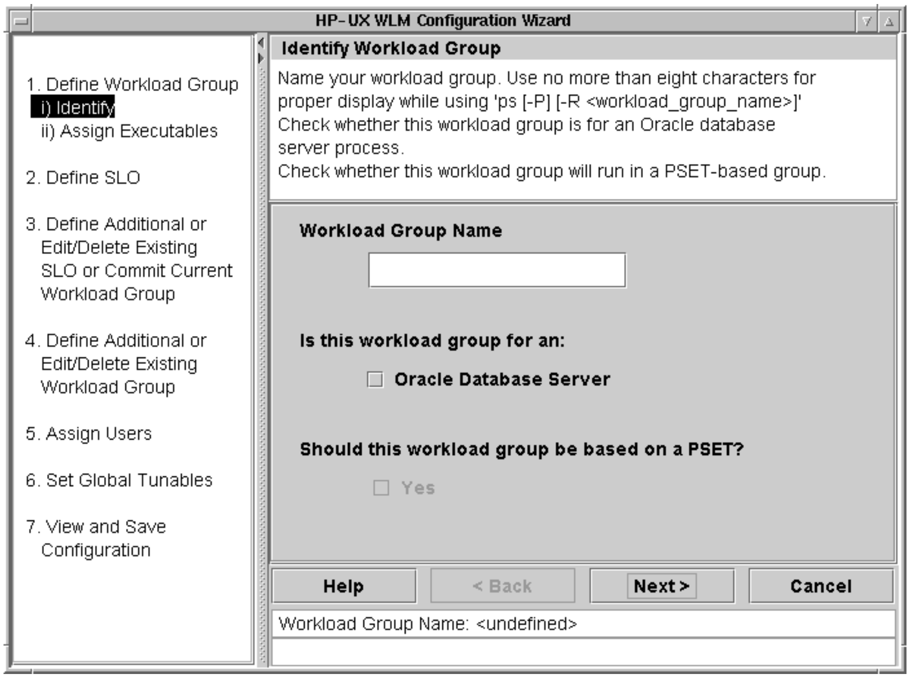
When you configure WLM, you define the workload groups, their goals, and the means to measure their progress toward goals. You can also tune WLM's behavior.

You can configure WLM in a number of ways:

- Use a text editor to create a configuration or modify an example configuration
- Use the configuration wizard
- Use the WLM GUI

The following figure shows the wizard.

Figure 2. The WLM configuration wizard



HP-UX WLM Configuration Wizard

1. Define Workload Group
   i) Identify
   ii) Assign Executables
2. Define SLO
3. Define Additional or Edit/Delete Existing SLO or Commit Current Workload Group
4. Define Additional or Edit/Delete Existing Workload Group
5. Assign Users
6. Set Global Tunables
7. View and Save Configuration

Identify Workload Group

Name your workload group. Use no more than eight characters for proper display while using 'ps [-P] [-R <workload_group_name>]'
Check whether this workload group is for an Oracle database server process.
Check whether this workload group will run in a PSET-based group.

Workload Group Name

Is this workload group for an:

□ Oracle Database Server

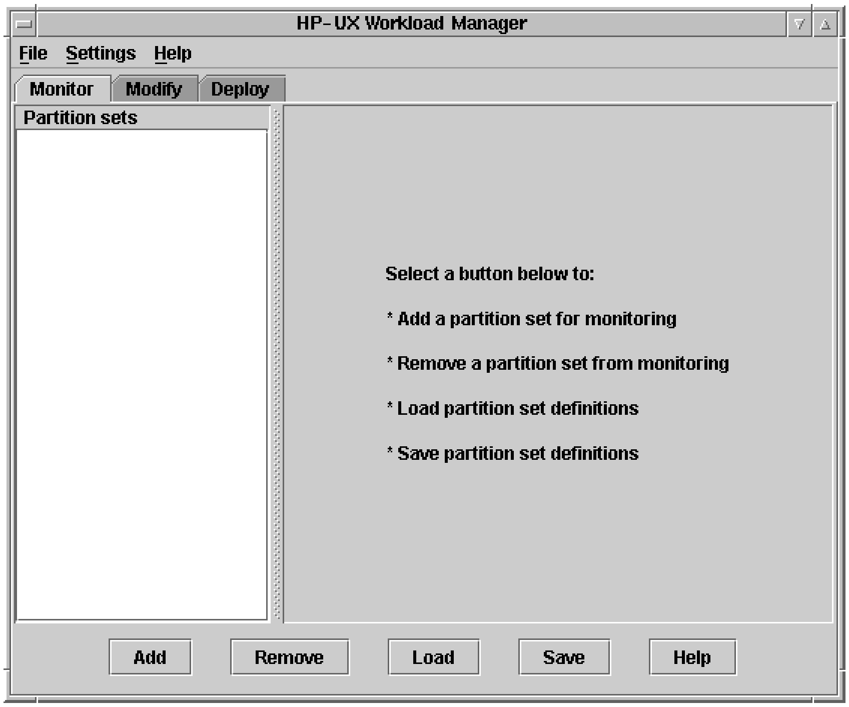
Should this workload group be based on a PSET?

□ Yes

Workload Group Name: <undefined>

The next figure shows the GUI.

Figure 3. The WLM GUI



Select a button below to:

* Add a partition set for monitoring

* Remove a partition set from monitoring

* Load partition set definitions

* Save partition set definitions

The following sections examine WLM configuration by exploring the WLM configuration file and SLO definitions.

Defining SLOs

The WLM configuration file is used to specify the SLOs for each workload group. Each SLO definition begins with the keyword `slo`, followed by a name. The SLO is expressed as a collection of keyword/value pairs, delimited within brackets { }. The syntax for expressing an SLO is:

`slo` *slo_*name `{` *keyword* `=` *value* `;` *keyword* `=` *value*`;` `...` `}`

This expression is called an `slo` structure. An example of an `slo` structure is:

```
slo product_query_goal {
      pri = 1;
      entity = PRM group e_commerce;
      mincpu = 30;
      maxcpu = 70;
      goal = metric e_biz.query_time < 10.0;
      condition = (Mon - Fri) && (09:00 - 17:00);
}
```

This SLO's goal is to keep the metric e_biz.query_time under 10 seconds Monday through Friday between 9am and 5pm, using between 30 and 70 CPU shares. This SLO is priority (`pri`) 1 and applies to the workload group (PRM group) e_commerce.

SLO configuration examples

The following examples demonstrate various types of SLOs that are available. They start by showing simple functionality, then progress into the more advanced capabilities of WLM.

Note that many of the SLOs below are also discussed in the section “Solutions HP-UX Workload Manager provides” on page 6. Previously, we focused on the business solutions the SLOs provided. Here, we see the actual SLO configurations, as they would appear in a WLM configuration file.

In this first example, the SLO requests a fixed allocation of 30 CPU shares for its workload group, which is named Marketing. This SLO is priority 1 and is in effect at all times.

```
slo marketing_analysis {
     pri = 1;
     cpushares = 30 total;
     entity = PRM group Marketing;
}
```

The next SLO also requests a fixed allocation, in this case 80 shares. However, the associated application is a payroll application that only runs twice a month. Consequently, the SLO is enabled only twice a month, on the 6th and 21st, using the `condition` statement.

```
slo payroll_processing {
     pri = 1;
     cpushares = 80 total;
     entity = PRM group Payroll;
     condition = */06/* || */21/*;
}
```

The following example shows another SLO that is enabled only some of the time. It is enabled only when a system accounting program is running, as indicated by a nonzero value for the metric sysacct_running in the `condition` statement. This metric is collected using a `tune` structure, which is not shown below. Once the SLO is active, it works to ensure the accounting program completes quickly by reserving 60 CPU shares for the associated workload group. Once the program completes, the SLO is disabled.

```
slo sysacct {
     pri = 1;
     cpushares = 60 total;
     entity = PRM group SysAcct;
     condition = metric sysacct_running;
}
```

Similarly, the next SLO is 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. The pkgA_active metric is provided by a utility that is included with WLM. When the metric has value 1, the SLO is active, causing WLM to attempt to allocate 25 CPU shares to the associated workload group.

```
slo pkgA_slo {
     pri = 1;
     cpushares = 25 total;
     entity = PRM group Failover_pkgA;
     condition = metric pkgA_active;
}
```

Now consider an SLO with a usage goal. Usage goals do not require a metric value. In this example, WLM adjusts the group's CPU allocation (to between 20 and 80 shares) based on the CPU utilization of applications executing in the Orders group. If the utilization is low (due perhaps to fewer applications running in the group), then the allocation for the Orders group is reduced, making more resources available for other groups. If utilization is high, the group receives a larger CPU allocation.

```
slo order_processing {
     pri = 1;
     mincpu = 20;
     maxcpu = 80;
     entity = PRM group Orders;
     goal = usage _CPU;
}
```

WLM allows you to grant a workload group a certain amount of CPU per unit of a metric. In our example below, the workload group gets 5 CPU shares for each active process in the workload group, as given by the metric apache.active_procs. (The tune structure required to get the apache.active_procs metric is not shown.) However, its CPU request is not allowed to fall below 10 shares or exceed 90 shares. 5 active processes would result in 25 CPU shares with the following SLO:

```
slo servers_proportional {
     pri = 1;
     mincpu = 10;
     maxcpu = 90;
     entity = PRM group webserver;
     cpushares = 5 more per metric apache.active_procs ;
}
```

This next SLO has a metric goal. It becomes active, based on the metric number_of_active_jobs, once the server is sufficiently busy. It attempts to keep the average job completion time under ten seconds. (The methods for collecting the metrics are shown after this example.)

```
slo job_processing {
     pri = 1;
     mincpu = 40;
     maxcpu = 90;
     entity = PRM group Jobs;
     goal = metric avg_completion_time < 10;
     condition = metric number_of_active_jobs > 100;
}
```

This SLO has two metrics that need to be collected: avg_completion_time and number_of_active_jobs. The first metric is generated outside WLM and then fed into WLM as defined in a `tune` structure. The second metric is retrieved using WLM's `glance_prm` data collector, which pulls information from GlancePlus. These structures are placed in the configuration file and are shown below:

```
tune avg_completion_time {
      coll_argv = wlmrcvdc;
}


tune number_of_active_jobs {
      coll_argv = wlmrcvdc glance_prm APP_ALIVE_PROC Jobs;
}
```

SLO combinations

WLM allows the various SLO types to be combined in a single configuration. Consider the following example, which combines three SLOs. The finance_deposit SLO is active only Monday through Friday. The other two SLOs are active at all times.

The goal of the finance_deposit SLO is to keep the metric fin_app.deposit.trans_sec under 2.0 seconds. The SLO specifies that its controller will never request fewer than 50 CPU shares, nor will it request more than 60 shares. This goal is priority 1.

The sales_analysis SLO is also priority 1. Instead of a metric goal, it has a usage goal. This goal attempts to keep the workload using between 60% and 80% of the allocation it is given through adjustments in that allocation.

If both the priority 1 SLOs make their maximum requests for CPU of 50 shares and 60 shares, there clearly won’t be enough CPU to meet the demand. In this case, WLM satisfies the SLOs to the same extent. That is, both would get 50 CPU shares. However, the required workload group OTHERS, as well as the Marketing group, must each receive at least 1% of the CPU. Removing 1% from both the Finance and Sales workload groups leaves those groups both with 49%.

The marketing_analysis SLO is a shares-based SLO; it has no goal. Also, it is priority 2; the other SLOs will be met before WLM attempts to meet this SLO.

```
slo finance_deposit {
     pri = 1;
     mincpu = 50;
     maxcpu = 60;
     entity = PRM group Finance;
     goal = metric fin_app.deposit.trans_sec < 2.0;
     condition = Mon - Fri;
}
slo sales_analysis {
     pri = 1;
     mincpu = 20;
     maxcpu = 50;
     entity = PRM group Sales;
     goal = usage _CPU 60 80;
}
slo marketing_analysis {
     pri = 2;
     cpushares = 30 total;
     entity = PRM group Marketing;
}
```

The various SLOs shown above give a hint of the flexibility and power that WLM SLOs provide. The SLOs can always be active, or they can be active based on a metric or the time of day. Also, SLOs can strive to meet certain shares allocations, resource usage goals, or any type of metric goals. Attaching priorities to the SLOs, the system administrator can easily emphasize the more important workload groups or give an individual workload group multiple goals at various priority levels.

Trying your configuration file without WLM controlling the system

WLM provides a passive mode that allows you to see how WLM will approximately respond to a given configuration, without putting WLM in charge of your system’s resources. Using this mode, you can safely analyze your configuration’s behavior, 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 SLOs

WLM allows you to monitor SLO compliance and a wealth of other information through its `wlminfo` utility. This utility has both a command-line interface and a graphical interface.

A few examples of the command-line interface are shown below.

In the first example, we focus on SLOs. Entering `wlminfo slo`, we see the number of shares requested in the ‘Req’ (request) column. The next column, ‘Shares’, indicates the number of shares actually allocated. From the ‘State’ column, we see that the two priority 1 SLOs are passing, while the priority 2 SLO is failing. The ‘Concern’ column helps highlight information. In this example, the `Priority` entries indicate that the shares granted are less than what was requested due to WLM’s satisfying requests from other SLOs at the same priority or at a higher priority.

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

SLO Name             Group        Pri    Req Shares  State Concern
finance_deposit      Finance        1     50      49  PASS  Priority
sales_analysis       Sales          1     49      49  PASS
marketing_analysis   Marketing      2     30       1  FAIL  Priority
```

Now we focus on workload groups. Entering `wlminfo group`, we see by the ‘CPU Util’ column (Util represents utilization) that all the user-defined groups are consuming CPU, although the Marketing group’s consumption is minimal.

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

Workload Group   PRMID  CPU Shares  CPU Util  Mem Shares  State
OTHERS               1        1.00      0.00        0.00  ON
Finance              2       49.00     43.08        0.00  ON
Sales                3       49.00     28.64        0.00  ON
Marketing            4        1.00      0.88        0.00  ON
```

Last, we look at the metrics used in the current WLM configuration. The 'PID' column shows the PID of the data collector providing the metric to WLM. In this example, the WLM daemon, `wlmd`, with PID 9410 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. For the _CPU_Sales metric, its value represents the amount of its CPU allocation it is using. The metric value is just under the goal of using at least 60% of the group's allocation. Consequently, WLM will soon reduce the group's allocation.

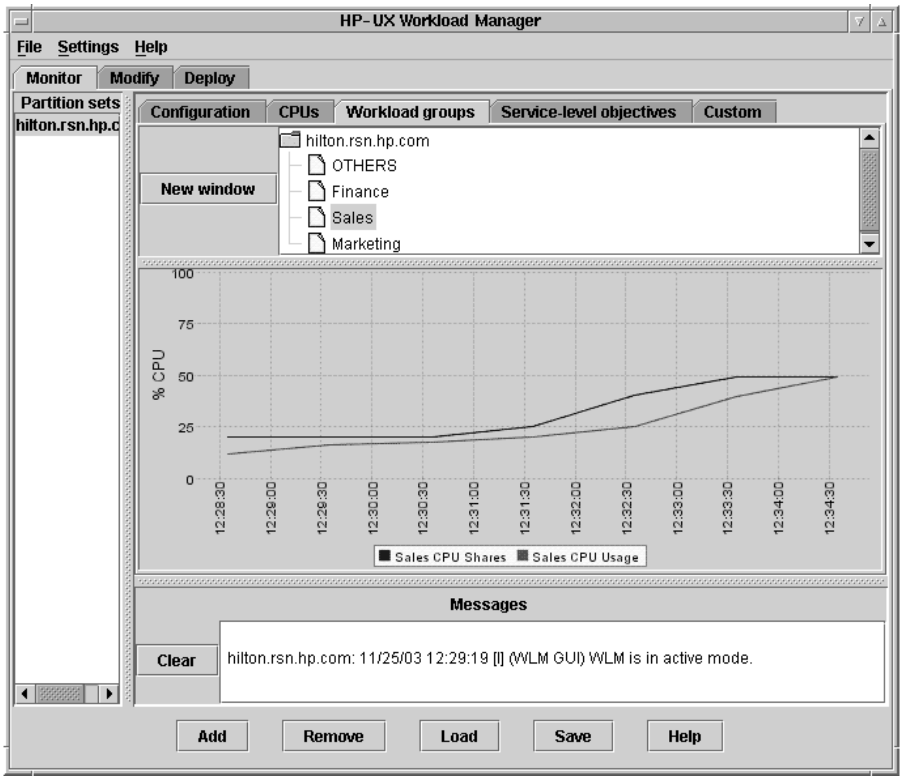
```
% /opt/wlm/bin/wlminfo metric

Metric Name                PID State Value
fin_app.deposit.trans_se  9411 NEW   1.900000
_CPU_Sales                9410 NEW   59.815505
```

The `wlminfo` utility also has a graphical interface. To use this interface, use the `-i` option:

```
% wlminfo -i
```

This option invokes the WLM GUI (`wlmgui)`, which provides monitoring in addition to configuration features. The Monitor tab of the GUI provides graphs of allocation and usage for the workload groups. The following figure shows how much CPU is allocated to the `Sales` group (top line), as well as how much CPU the group is actually using. A usage goal is deciding the group's CPU allocation. So, WLM adjusts its allocation so that the group's utilization of that allocation falls within a given percentage range.

Figure 4. Monitoring WLM



hilton.rsn.hp.com: 11/25/03 12:29:19 [I] (WLM GUI) WLM is in active mode.

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 allows you to specify fixed allocations to make sure each application gets a defined resource amount. Additionally, the system administrator can define and prioritize one or more SLOs for each application's associated workload group. The workloads groups are then given the resources they need to achieve the SLOs. This 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.
- Service-level objectives 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—then lower priority SLOs are met. Each workload has at least one SLO so 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 resources are available.

Using the defined SLOs, WLM determines the amount of CPU needed to achieve each SLO. It then allocates CPU to each SLO's workload group 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 with Serviceguard

Using WLM with Serviceguard, you can:

- Minimize the performance impact of a failover caused by software being unavailable in active-active configurations
- Maximize utilization of a system that receives failover packages
- Simplify the configuration of failover scripts
- Simplify routine/scheduled maintenance

WLM provides a number of features that support integration with Serviceguard. For more information, as well as step-by-step procedures on 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" page on http://www.hp.com/go/wlm.

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`.

Once you've activated a configuration using `-t`, use the `wlmaudit` command to display the audit data. The `wlmaudit` command allows 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.

Using HP-UX Workload Manager with Oracle® databases

HP-UX WLM Oracle Database Toolkit (ODBTK) simplifies getting metrics on Oracle database instances into WLM. This allows 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 when a particular user connects to the instance
- Increase an instance's available CPU when more than *n* users are connected
- Increase an instance's available CPU 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 processor sets

HP-UX 11i V1.0 (B.11.11) and later offer a feature known as processor sets, or PSETs. A processor set represents one or more processors grouped together for exclusive access by applications assigned to that processor set.

WLM allows you to define your workload groups based on processor sets. These groups have dedicated processors. Furthermore, WLM allows you to dedicate physical memory to a workload group that is based on a processor set. This isolates the workload's memory from other workloads running on the system. Rather than simply isolating the group though, you can configure WLM to adjust the number of CPUs in your PSET-based workload groups, giving and taking away CPUs as indicated by your SLOs.

Using HP-UX Workload Manager with iCOD

iCOD, or Instant Capacity on Demand, offers the ability to activate reserve CPU capacity that is already on your system. You do not pay for the iCOD CPUs until they are activated. You would activate the iCOD CPUs either to increase the active CPU capacity on the system or to replace the CPU capacity of a failing processor on that system. In either case, action can be taken—with no waiting for new CPUs to arrive—because the iCOD CPUs are already on the system.

If you have HP-UX WLM on a system with the iCOD software, you can configure HP-UX WLM to let you know that SLOs are failing and the reserves are needed. By itself, WLM only notifies you of the situation; you must manually activate the reserves. However, if you use WLM with the Pay Per Use Toolkit (PPUTK, part of the WLM Toolkits product) on an iCOD system, you can have CPUs automatically activated when SLOs are failing.

Using HP-UX Workload Manager with Pay Per Use

HP offers a Pay Per Use (PPU) feature that provides the capacity to support peak anticipated demand, but with payment for the HP server based on actual metered or monitored usage of that capacity. This capacity can be increased or decreased, as needed, on a per-CPU basis.

Using WLM on a system with the PPU software (up to and including version B.04.01), this capacity is increased or decreased automatically. After you set your SLOs in the WLM configuration file, WLM and its Pay Per Use Toolkit (PPUTK) automatically optimize the number of active CPUs to the minimum number of CPUs needed to satisfy the SLOs. With PPU, you pay only for the amount of CPU you use; thus, minimizing the number of active CPUs minimizes your costs.

PPUTK is available as part of the WLM Toolkits product, which is included with WLM.

Using HP-UX Workload Manager with Servicecontrol Manager

Servicecontrol Manager provides a single point of administration for multiple HP-UX systems. The WLM integration with Servicecontrol Manager allows system administrators at the Central Management Server (CMS) to perform the following activities 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
- Check 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 Apache

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

- Separate Apache from Oracle database instances
- Separate Apache from batch work
- Isolate a resource-intensive 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 on the web at http://h30081.www3.hp.com/products/wlm/docs/apache_wlm_howto.html.

Using HP-UX Workload Manager with SAS®

WLM and its Duration Management Toolkit (DMTK) allow you to manage the duration of your applications. Because many SAS jobs run from a few minutes to a few hours, DMTK can help manage your SAS environments, minimizing the impact of these jobs and preventing the need for purchasing excess hardware. Given the usefulness of DMTK in these environments, HP also produced the WLM Toolkit for SAS Software (SASTK). This toolkit provides a SAS macro that you place in your SAS jobs to fine-tune duration management. (Both these toolkits come with the WLM Toolkits product, which is included with WLM.)

Using HP-UX Workload Manager with BEA WebLogic Server™

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 an increasing amount of CPU resources in the form of a dynamic PSET (processor set) 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 wlmwlsdc(1M) 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 on the web at http://h30081.www3.hp.com/products/wlm/docs/weblogic_wlm_howto.html.

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 of the servers.

WLM can be integrated with Hewlett-Packard's Serviceguard product by storing the WLM configuration file in a filesystem shared by all nodes in the cluster, then activating the configuration on each node independently.

Using HP-UX Workload Manager with vPars and other partitions

The HP Partitioning Continuum offers three layers of partitioning. These layers are:

- Hard partitions—These partitions are implemented through hardware. The first form of hard partitions is a complete server, which can be clustered in a HyperPlex configuration. Another form of hard partitions is known as nPartitions, which are portions of a single server. Each hard partition runs its own instance of HP-UX.
- Virtual partitions—These partitions are created with software, with each virtual partition running its own instance of the HP-UX operating system. Virtual partitions can be used within hard partitions.
- Resource partitions—PRM combined with processor sets provides resource partitions. These run within, but not across, hard partitions and virtual partitions. These partitions run within a single instance of HP-UX.

WLM can be used within both hard partitions and virtual partitions. Furthermore, you can use WLM across partitions, automatically moving CPUs between partitions based on SLOs in the partitions. (Given the physical nature of hard partitions, the "movement" of CPUs between partitions is achieved using iCOD software: a CPU is deactivated on one nPar, then a CPU on another nPar is activated.)

For more information on partitions, visit www.hp.com/go/servicecontrol.

For information on WLM's management of virtual partitions, see the HP-UX Workload Manager User's Guide.

HP-UX WLM and PRM

The HP products HP-UX WLM and PRM both use the same HP-UX resource controls. Consequently, you should not use more than one of these products at a time on a single host.

Related information

The following list of references provides useful background information on related products and topics:

- HP-UX Workload Manager (HP-UX WLM)
  www.hp.com/go/wlm
  HP-UX Workload Manager User's Guide www.docs.hp.com/hpux/netsys/
  White papers (www.hp.com/go/wlm):
  – Using WLM: A quick reference
  – More efficient high availability through manageability (integrating Serviceguard and HP-UX Workload Manager)
- HP-UX Workload Manager Toolkits, which consists of:
  – HP-UX WLM Oracle Database Toolkit (ODBTK)
  – HP-UX WLM Pay Per Use Toolkit (PPUTK)
  – HP-UX WLM Duration Management Toolkit (DMTK)
  – HP-UX WLM Toolkit for SAS Software (SASTK)
  – HP-UX WLM Toolkit for Apache (ApacheTK)
  – HP-UX WLM SNMP Toolkit (SNMPTK)
  – HP-UX WLM BEA WebLogic Server Toolkit (WebLogicTK)

  www.hp.com/go/wlm
  HP-UX Workload Manager Toolkits User's Guide www.docs.hp.com/hpux/netsys/
- HP Process Resource Manager (PRM)
  www.hp.com/go/prm
  Process Resource Manager User's Guide www.docs.hp.com/hpux/ha/
- HP Partitioning Continuum
  www.hp.com/go/servicecontrol
- OpenView Performance Agent (OVPA)
  www.openview.hp.com/
- Application Response Measurement (ARM)
  Application Response Measurement (ARM) API
  www.cmg.org/regions/cmgarmw/
- Event Monitoring Service (EMS)
  www.unix.hp.com/highavailability
- Servicecontrol Manager
  http://www.hp.com/go/servicecontrol
- Instant Capacity on Demand (iCOD)
  http://www.hp.com/go/icod

For more information

For more information on HP-UX Workload Manager, contact any of our worldwide sales offices or visit our web site at: http://www.hp.com/go/wlm.

To learn more about the adaptive enterprise and virtualization, please visit http://www.hp.com/go/virtualization.

01/2004