

# HP Process Resource Manager overview



Introduction .....	3
What is HP Process Resource Manager? .....	4
Benefits .....	4
Features .....	4
Comparing HP Process Resource Manager and HP-UX priorities and “nice” features .....	5
Standard HP-UX scheduler .....	5
How HP Process Resource Manager manages resources .....	6
Managing CPU allocation .....	7
Limiting maximum CPU consumption .....	9
Managing private real memory allocation .....	9
Managing shared real memory allocation .....	10
Managing disk I/O bandwidth allocation .....	10
Allocating resources to users .....	11
Allocating resources to UNIX groups .....	11
Managing applications .....	12
Managing multiple instances of an application .....	12
Adapting to changing needs .....	13
Assigning resources hierarchically .....	13
Command-line utilities .....	14
Graphical interface .....	14
Isolating the CPU and memory of an application .....	15
HP Process Resource Manager, HP-UX Workload Manager, and HP Global Workload Manager .....	16
Integrating HP Process Resource Manager with other products .....	16
Integration with HP Integrity Virtual Machines .....	16
Integration with Oracle, Sybase, and Informix .....	16
Integration with HP Serviceguard .....	16
Integration with HP System Management Homepage .....	17

Integration with HP Systems Insight Manager .....	17
Forming Secure Resource Partitions .....	17
Application monitoring with HP Glanceplus .....	17
Integration with HP-UX commands.....	18
Integration with HP-UX accounting.....	18
Example: Checking for patterns and configuration accuracy .....	18
Conclusion .....	19
Related information.....	20
For more information.....	21

# Introduction

Traditional IT environments are often silos in which both technology and human resources are aligned around an application or business function. Capacity is fixed, resources are over-provisioned to meet peak demand, and systems are complex and difficult to change. Costs are based on owning and operating the entire vertical infrastructure—even when it is underutilized.

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

The HP Virtual Server Environment (VSE) for HP-UX, part of the HP Adaptive Enterprise, has the broadest range of server virtualization capabilities, including clustering, both for high availability and high performance, utility pricing technologies such as HP Instant Capacity and HP Pay per use (PPU), partitioning, resource management, and more.

The VSE provides several forms of partitioning:

- Hard partitions (or nPartitions)—These partitions are implemented through hardware. nPartitions, a form of hard partitions, 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.
- Virtual machines—These partitions, much like virtual partitions, are created with software. However, they emulate generic servers and, therefore, can offer sub-core (sub-CPU) 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—HP Process Resource Manager (PRM), managing processor sets (pSets) or Fair Share Scheduler (FSS) groups, provides resource partitions. These partitions can be used within (but not across) hard partitions, virtual partitions, and virtual machines. These partitions run within a single instance of HP-UX.

HP PRM is an important part of the VSE. It provides resource partitions, enabling you to partition system resources within a single HP-UX instance. With these partitions in place, you can consolidate multiple workloads within that HP-UX instance.

This paper provides an overview of HP PRM. It describes benefits and features and provides links for more detailed information. The paper covers PRM C.03.03.01, which is supported on:

- HP 9000 servers running HP-UX 11i v1 (B.11.11), HP-UX 11i v2 (B.11.23), or HP-UX 11i v3 (B.11.31)
- HP Integrity servers running HP-UX 11i v2 or HP-UX 11i v3 (B.11.31)

# What is HP Process Resource Manager?

HP PRM is a resource management tool that enables you to control the allocation of CPU, private and shared real memory, and disk bandwidth resources to multiple workloads or users within an HP-UX instance. This capability enables you to stack multiple applications within a single HP-UX instance and ensure that the critical applications and users get system resources in a timely manner.

PRM is policy-based; it manages resources based on policies you define. These policies govern resource consumption by users, applications, and UNIX groups. You can define multiple policies to manage resources. For example, you might have one policy during the day and another one at night. You can specify resource policies to PRM through a text configuration file or an easy-to-use GUI.

## Benefits

PRM works within a single HP-UX instance to provide many benefits. With PRM, you can:

- Ensure that critical users or applications have sufficient CPU, memory, and disk I/O bandwidth resources, allowing multiple applications to share an HP-UX instance
- Restrict the CPU, memory, and disk I/O bandwidth resources available to users or applications that might otherwise consume more than their fair shares of resources
- Allocate resources based on budgeting
- Ensure that an application package in an HP Serviceguard cluster has sufficient resources on an active standby system in the event of a failover
- Consolidate multiple instances of an application, such as Oracle®, onto a single HP-UX instance, saving the effort of tuning numerous systems in the exact same manner for the application

## Features

The benefits noted in the previous section are the results of various PRM features, with which PRM:

- Enables you to define policies to manage the allocation of the most critical shared resources: CPU, private and shared real memory (including isolation of a pSet's memory), and disk bandwidth
- Supports resource allocation policies for both users and applications, which makes PRM valuable for interactive and application systems
- Supports resource allocation policies for UNIX groups
- Supports multiple mission-critical applications on a single image of HP-UX by ensuring application isolation and policy-based resource allocation
- Requires no modifications for applications to operate under PRM control (Any application can work with PRM.)
- Enables you to modify the PRM configuration dynamically at any time, even while the system is under load (PRM and all applications continue to run on the system when the resource policy is modified—there is no downtime. This functionality allows the resource allocation policy to be changed as needed by business goals that vary based on the time or date. Policy changes can be automated so that no operator or administrator action is required.)
- Is aware of users defined in netgroups (defined in the file `/etc/netgroup`), which saves time during configuration and is invaluable in networked computing environments
- Supports allocation of resources among up to 256 PRM groups (partitions of the HP-UX instance)
- Supports allocation of resources in a hierarchical manner, where a PRM group's resources are divided among its subgroups

- Enables you to create Secure Resource Partitions, which combine HP-UX 11i v2 security capabilities with the resource management capabilities of PRM
- Can scan UNIX® accounting files for data on the specified resource (CPU, real memory, and disk I/O bandwidth) and then order the accounting records by user, UNIX group, command name, or PRM group (This feature helps in better understanding resource usage for capacity planning purposes and for fine-tuning resource allocation configurations.)

## Comparing HP Process Resource Manager and HP-UX priorities and “nice” features

You could ask the question, “Why do I need PRM when HP-UX has the ability to assign different priorities and different ‘nice’ values to processes?”

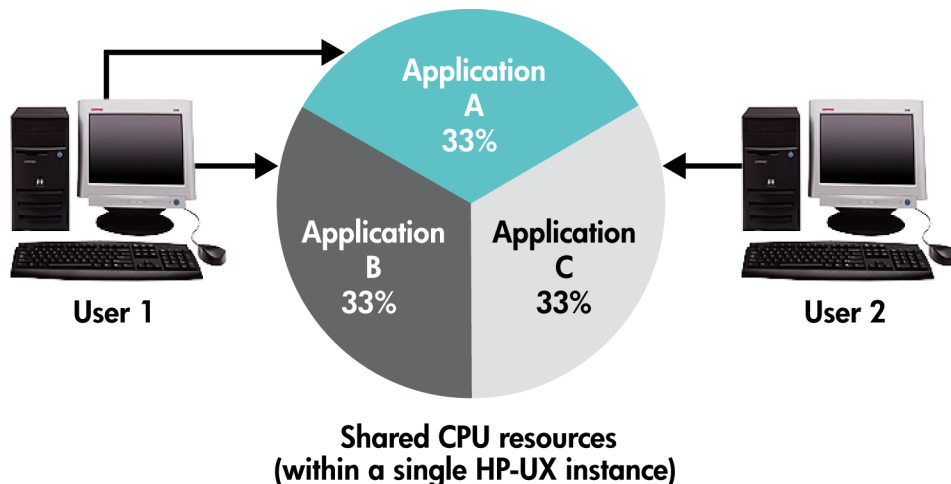
The answer is simple. Both of these HP-UX features help influence an application’s access to CPU resources, but neither of these two HP-UX features enables an application to have a guaranteed percentage of the system (CPU resources). With PRM, an application can be guaranteed a configured CPU allocation and other critical system resources.

### Standard HP-UX scheduler

How does CPU allocation work under the standard UNIX scheduler?

When the standard UNIX CPU scheduler is in effect, all processes compete with each other for CPU time. In this case, low-priority applications can affect the performance of higher priority applications. Because the standard CPU scheduler attempts to share CPU cycles equally among all processes, it is not possible to guarantee that business priorities will be achieved for critical applications. Figure 1 illustrates the situation in which one user is consuming most of the available CPU resource by executing two applications.

Figure 1. Standard UNIX CPU scheduler



Because the CPU time is allocated equally to the executing processes, each application receives 33% of the available CPU resource. However, equality is not always a desired characteristic. Application

C might be much more important than application A or B. However, the standard UNIX scheduler might not give application C the extra CPU resources required to meet its service goals.

## How HP Process Resource Manager manages resources

PRM enables you to manage the most critical resources on a system: CPU, shared and private real memory, and disk bandwidth. PRM manages these resources by controlling the allocation of these resources to PRM groups.

Each PRM group is a partition of the system's resources. Because this partitioning is accomplished in the operating system, it can be changed at any time, even while the system is in use.

When PRM is configured on an HP-UX system, each executing process belongs to one of the defined PRM groups. You define PRM groups by assigning a meaningful name and a CPU allocation. Optionally, you can also allocate memory (private and shared) and disk I/O bandwidth. PRM provides several options for assigning users and applications to the PRM groups.

PRM has two reserved PRM groups: PRM\_SYS and OTHERS. These groups, which are created automatically if you do not explicitly define them, are the default PRM groups for superuser and non-superuser processes, respectively. Any process that is not explicitly assigned to an administrator-defined PRM group is automatically placed in the appropriate default group. Consequently, all user and application processes on the system belong to a PRM group.

You assign resource shares to PRM groups to allocate CPU, private memory, and disk I/O bandwidth resources. A resource share is a guaranteed minimum when the system resource is fully utilized. If the system resource is not fully used, busy groups can use the excess capacity (unless you specifically disallow sharing). When the activity on the system consumes all of the available resource, PRM enforces the shares to ensure that the specified policies are met. For shared memory, you can assign a minimum amount of memory for a group to use. The resources you assign your PRM groups are what remain after PRM assigns resources to the PRM\_SYS group.

Resource shares allow simple additions and removals of groups to a configuration because each group's resource allocation is determined by its number of shares relative to the total number of shares assigned in the configuration for the given resource.

Valid share values are integers from 1 to MAXINT (the maximum integer value, as defined in `/usr/include/sys/param.h`). PRM calculates the sum of the shares and then allocates a percentage of the system resource to each PRM group based on the group's number of shares relative to the total number of shares assigned.

The following table illustrates the concept. The OTHERS group is included automatically with its minimum one share. The CPU shares assigned in the following table are what remain after the PRM\_SYS group takes the resources it needs. Thus, even though the percentages add up to 100, it is not 100% of the system.

PRM group	CPU shares	CPU %
GroupA	1	1/4 = 25.00%
GroupB	2	2/4 = 50.00%
OTHERS	1	1/4 = 25.00%

You allocate:

- CPU resources in shares or processor sets
- Private memory and disk I/O bandwidth in shares
- Shared memory in megabytes

The following sections describe how these resources are managed by PRM.

## Managing CPU allocation

PRM enables you to manage CPU resources at multiple granularities:

- Sub-core—FSS PRM groups are named for the PRM CPU scheduler FSS. An FSS PRM group can be allocated resources of less than a single core or resources of multiple cores. The available CPU resources are shared by all FSS PRM groups, with each group taking a portion of each core. (A core is the actual data-processing engine within a processor. A single processor might have multiple cores. Starting with HP-UX 11i v3, a core can support multiple execution threads. This feature is known as Hyper-Threading. With Hyper-Threading disabled or unavailable, each core is seen as a CPU. With Hyper-Threading enabled, each core can be seen as multiple, logical CPUs.)
- Whole-core—pSet PRM groups are based on processor sets (a collection of cores). The cores within a pSet PRM group are dedicated to that PRM group. No processes outside the PRM group can access those cores.

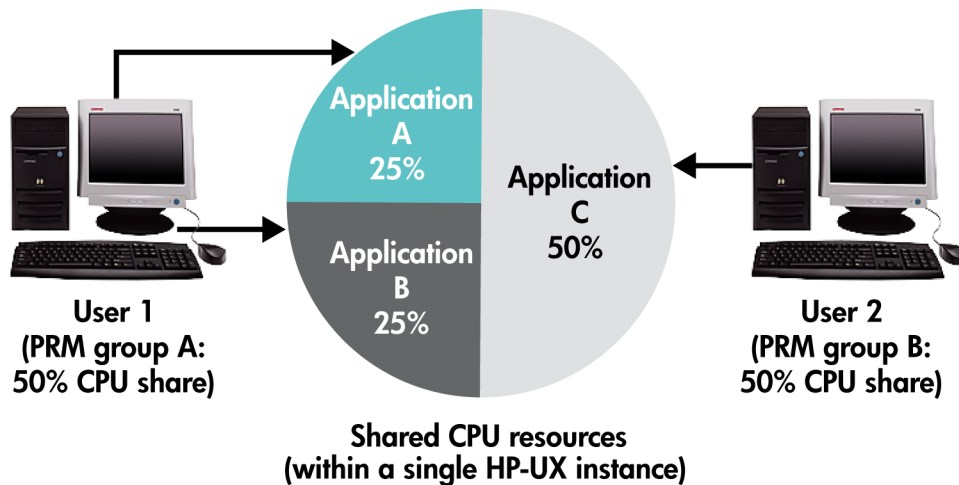
The PRM CPU scheduler takes precedence over the normal HP-UX CPU scheduler to ensure that FSS PRM groups are allocated their specified share of CPU cycles. CPU time is divided among the PRM groups according to the configured CPU shares. With shares, a group's CPU allocation is determined by dividing the group's number of shares by the total number of shares assigned. For example, if a group has five shares and 50 shares were assigned to all the groups in the policy, the group has 5/50 or 10% of the available CPU resources.

With a pSet PRM group, you assign the desired number of cores to the PRM group. These cores are then available only to the applications and users assigned to the group, effectively creating a partition on your system.

Within either type of PRM group, the standard HP-UX CPU scheduler is in effect. That is, all processes within a particular group compete with each other for CPU time based on their standard scheduling priorities.

Figure 2 illustrates a system using FSS PRM groups to ensure that applications receive a certain amount of CPU resources.

Figure 2. PRM (FSS) CPU scheduler



By creating two FSS PRM groups based on the users and assigning 50% of the system CPU resources to each, the shared CPU resource is partitioned according to the established policy.

Free CPU cycles are available for use by processes in any FSS PRM group. Additionally, if the CPU resources are 100% busy but one or more FSS PRM groups are not using their CPU shares, the busy FSS PRM groups get the idle groups' CPU cycles in proportion to the busy groups' shares. For example, suppose there are three users, each with a dedicated PRM group, defined as follows:

- "CEO" has 30% of the CPU resources
- "Joe\_Mktg" has 20% of the CPU resources
- "Lisa\_Sales" has 50% of the CPU resources

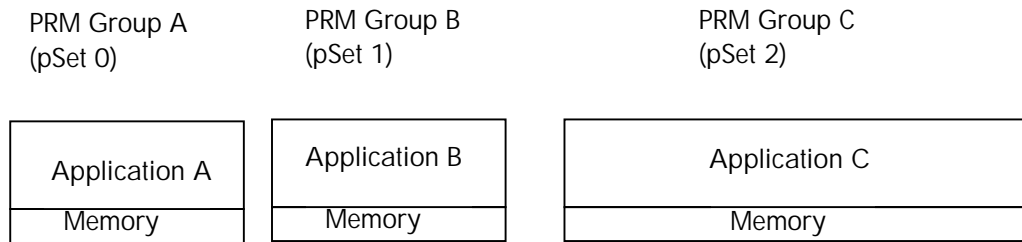
If processes in Joe\_Mktg are idle but processes in CEO and Lisa\_Sales are using all of the available CPU resources, then the CEO processes receive 37.5% of the CPU resources and processes in Lisa\_Sales receive 62.5% of the CPU resources. That is, the busy groups get the spare CPU cycles in a 3:5 ratio (the ratio of their configured CPU allocations).

You can limit a group's borrowing of unused resources with per-group capping, available starting with HP-UX 11i v3. For more information, see the "Limiting maximum CPU consumption" section.

With pSet PRM groups, a group's cores are available only for use by the processes in that group. The following figure shows an HP-UX instance divided into three processor sets. Processes in PRM group A have no access to the cores in pSet 1 or pSet 2. Using PRM private memory management, you can also isolate the memory available to the PRM groups.



Figure 3. PRM groups based on pSets



## Limiting maximum CPU consumption

For pSet PRM groups, CPU use is automatically restricted to (capped at) the number of cores in the group.

For FSS PRM groups, there are two forms of capping:

- For all FSS PRM groups in the configuration—Each FSS PRM group can use only up to its entitlement, regardless of whether additional CPU cycles are available. (The entitlement, or guaranteed minimum, becomes a guaranteed maximum.)
- On a per-group basis (Available for HP-UX 11i v3 and later)—You set a cap for only those FSS PRM groups for which you want to cap CPU consumption. You cap these groups at or above their entitlement values.

Capping is useful for setting and controlling users' or applications' expectations of system performance. For example, a shared system could be configured so users always see the same response time, regardless of the load on the system. The CPU capping feature is particularly useful during the process of consolidating multiple applications onto a single server. Initially, only one application may be running on that server, and users experience exceptionally fast response times. Later, when more applications are added to the server, the users of the first application could experience a decrease in performance, resulting in problem reports. By capping resource allocation to applications from the start, you prevent such performance problem perceptions.

CPU capping is also useful to "contain" resource-intensive applications (such as a CPU-bound program), applications that might have gone into a bad state (for example, infinite loop) or applications that are poorly written. CPU capping ensures that these applications cannot starve critical applications on the system.

## Managing private real memory allocation

Applications need access to private real memory to perform their jobs. Contention for private memory can affect application performance. IT managers who are considering application consolidation want to be assured that their applications will not experience performance degradation caused by memory contention with other applications on the system.

Without PRM, critical applications could have their memory paged out in favor of lower priority applications or applications that might already be consuming too much memory.

The PRM memory manager complements the standard HP-UX memory manager by prioritizing processes. This prioritization is performed according to the memory shares assigned to the PRM

groups. You can be assured that critical applications will get the real memory they need to meet their service level objectives.

The PRM memory manager enables you to assign shares (a guaranteed minimum allocation) of private real memory. PRM uses a kernel-space method for enforcing minimum memory allocations. PRM pages out the memory of applications in PRM groups that are consuming more than their configured memory allocations. When excess memory is available, sharing of the excess is the default behavior. However, you can restrict which groups can borrow or lend their memory resources, potentially isolating memory for your groups. This isolation can be particularly beneficial when paging out memory is time-intensive. Memory capping is also available. When a PRM group's memory is being isolated or capped, if the group attempts to exceed its minimum memory allocation, PRM frees more memory by paging out the memory used by that PRM group.

Memory shares can be changed dynamically while the system is in operation to reflect changes in application priorities.

Combining memory management with pSet PRM groups, you can completely isolate a group's CPU and memory resources, which prevents other PRM groups from accessing that group's CPU or memory resources.

## Managing shared real memory allocation

By default, all shared memory is allocated in the PRM\_SYS group.

Starting with HP-UX 11i v2 Update 2 and PRM C.03.01, PRM can control shared memory allocations on a PRM group basis. You only control shared memory for the groups that need it—you can omit control for groups where shared memory control would not be helpful.

You set a minimum size in megabytes for a group's shared memory allocation. (This size is usually available from the configuration settings for the consuming application, as is the case with the Oracle SGA size.)

Capping is not available for shared memory.

## Managing disk I/O bandwidth allocation

Applications can underperform if they cannot complete their disk reads and writes in a timely manner. Hence, it is important to treat disk I/O bandwidth as a critical resource and manage it to ensure critical applications do not underperform because of contention for disk bandwidth.

PRM provides a disk I/O bandwidth control feature that enables you to define the priorities of disk read and write operations for different PRM groups. The PRM disk I/O control works with the HP Logical Volume Manager (LVM) and the VERITAS Volume Manager (VxVM). The I/O control operates at the volume group layer (with LVM) or disk group layer (with VxVM) in the operating system kernel. PRM controls disk bandwidth usage by reordering the I/O requests of a volume group (disk group). When the volume group (disk group) is busy, requests from low-priority processes are delayed and requests from high-priority processes are accelerated. Operating at this layer provides the capability to control the widest array of disk technologies in a single, easy-to-configure feature.

The disk I/O bandwidth shares can be modified while the volume group (disk group) is in use. Unused disk I/O bandwidth is shared among active PRM groups in proportion to their PRM disk I/O shares.

PRM cannot manage disk I/O bandwidth for pSet PRM groups.

## Allocating resources to users

With HP Process Resource Manager, you can easily manage resource allocations to users, as well as applications, by assigning users to PRM groups.

After defining PRM groups and setting resource allocations for CPU, real memory, and disk I/O bandwidth resources, you can specify how users are assigned to the PRM groups. This feature is useful for systems in which users are logging on and accessing interactive applications. PRM enables you to specify, for each valid user name, which PRM group the user is assigned to when logging on to the system. Then, when users access the system, their UNIX shell processes are automatically placed in their assigned PRM groups. Because a child process inherits the parent process's PRM group affiliation, all applications and jobs run by a user also execute in the assigned PRM group. All users and applications in a PRM group share the resources assigned to that PRM group. If a user does not have an assigned PRM group, the user's shell process is placed in the default PRM group OTHERS at login and that user is not allowed to access other PRM groups.

PRM configuration allows netgroups in user records to specify initial and alternate PRM groups for all members of a netgroup\*. Instead of adding a user record for each user on the system, you could create only one user record. This record would be for a netgroup you define, for example, `finance_dept`. The netgroup would include these users. Using a netgroup also simplifies updates when the staff changes.

Similarly, PRM enables you to place processes based on their effective UNIX group IDs, as described in the "Allocating resources to UNIX groups" section.

The PRM user assignment feature is integrated with standard HP-UX commands such as `cron` and `at`. This integration ensures that commands—or jobs that are executed on behalf of users at specified times—run in the appropriate PRM groups as well.

Assigning users to PRM groups ensures user processes receive the assigned allocations for CPU, real memory, and disk I/O bandwidth resources when the system is under heavy load. When using pSet PRM groups or when using FSS PRM groups with CPU capping enabled, users are prevented from using more than their CPU allocation, which ensures they always see a consistent level of system responsiveness. The objective with this PRM feature is not to restrict users' connect time as might be the case with a campus student server—there are already tools to deal with that problem. The objective is to ensure a consistent level of service to users of servers in an enterprise.

## Allocating resources to UNIX groups

UNIX groups defined in `/etc/group` enable you to easily give a collection of users the same level of access to various files on a system. You can configure PRM to provide a UNIX group with a certain level of access to system resources as well, placing processes in PRM groups based on their effective UNIX group IDs.

---

\* The file `/etc/netgroup` defines network-wide groups and is used for permission checking when executing remote mounts, remote logins, and remote shells. For remote mounts, the information in `netgroup` classifies machines. For remote logins and remote shells, it classifies users. See the `netgroup(4)` man page for additional details.

# Managing applications

The amount of resources available to an application is determined by the resource allocation for the PRM group to which it is assigned. You assign an application to a PRM group in the PRM configuration based on the application's executable name.

You can assign an application to only one PRM group at a time. When an application is started, it runs in the initial PRM group of the user that invoked it. If the application is assigned to a PRM group in the configuration file, the application manager soon moves the application to its assigned group. Child processes inherit their parent's PRM group. Therefore, all the application's child processes run in the same PRM group as the parent application by default.

You can explicitly place an application in a PRM group of your choosing with two commands. Use the `prmmove` command to move an existing application to another group. Use the `prmrn` command to start an application in a specified group. These rules might not apply to processes that bypass login.

The PRM application manager verifies that applications are running in the correct PRM groups on a regular basis (30 seconds by default).

## Managing multiple instances of an application

UNIX servers have evolved in the data center such that many application servers are running the same application or many database servers are running the same database software. The trend with system consolidation has been to reduce the number of servers needed to house the multiple copies or instances of application and database servers. This consolidation is accomplished by combining multiple copies of the application or database onto a single server. When combining multiple instances of an application onto a single server or partition, system administrators want to control how resources are allocated to each instance of the application. They want to control resource allocation for business and performance reasons.

As an example, consider a case in which three different business units share a single database server. This condition results in three copies of the database executing on the server. The system administrator might want to ensure that the resources allocated to each database instance are proportional to that business unit's funding level for that server, which might not be an equal funding level. Because the name of the executable is the same for all three instances, specifying the name of the executable file alone in the PRM group configuration does not enable the administrator to allocate separate resources for the three different instances of the same database executable. Fortunately, many database applications such as Oracle have a process naming mechanism that enables you to distinguish between multiple instances of the database.

With Oracle, when you open an instance, the `$ORACLE_HOME/bin/oracle` executable renames itself using the UNIX environment variable `ORACLE_SID` for a given database. This variable is used in generating Oracle process names: `ora_process_name_$ORACLE_SID`. Thus, if you have three instances of the Oracle databases, you can assign different process names to each instance by changing `$ORACLE_SID`—for example "Inv" for the inventory, "Fin" for finance, and "Mkt" for marketing. The database instance processes might appear in the output of the `ps (1)` command as:

```
ora_pmon_Inv, ora_dbwr_Inv, and ora_smon_Inv  
ora_pmon_Fin, ora_dbwr_Fin, and ora_smon_Fin  
ora_pmon_Mkt, ora_dbwr_Mkt, and ora_smon_Mkt
```

You can specify the application names to PRM using regular expressions, such as "ora\*Inv," "ora\*Fin," and "ora\*Mkt," and assign them to different PRM groups with different resource shares. In this case, PRM would correctly recognize the executing database instance processes and move them

to the assigned PRM groups. The database processes are then allocated resources based on their configured allocations. You can also use Extended Regular Expressions to match entire command lines.

You can assign each instance of the same application to separate PRM groups and consequently separate system resources. This feature is applicable to any process that renames its processes, which facilitates consolidating multiple instances or copies of an application onto a single server.

The PRM commands `prmrn` and `prmmove` can also be used to ensure each instance of an application belongs to the desired PRM group.

## Adapting to changing needs

PRM was designed with the operations of the data center in mind. HP realizes that a system, be it a stand-alone server, an nPartition, a virtual partition, or a virtual machine (provided by HP Integrity Virtual Machines), is not doing the same tasks 24 hours per day. As an example, many systems run batch workloads during the day and often have a backup window at night. You might want to have different resource policies in effect at these times. For example, during the day, backup processes should not get any CPU time. At night, however, backup processes can consume up to 50% of the CPU resources.

The HP PRM dynamic configuration change features enable you to create multiple configurations that represent resource policies for different times. Using `cron` jobs, PRM can be instructed to change the resource policy for each period.

## Assigning resources hierarchically

You can nest PRM groups inside one another, forming a hierarchy of groups similar to a directory structure, which enables you to divide groups into subgroups. A group's resources are divided among its subgroups. If a group has no subgroups, it uses the allocation itself.

For example, assume the Sales group has five CPU shares and the Finance group has five CPU shares. Consequently, both groups have 50% of the system's CPU resources. The Sales group is then subdivided into an Analysis group, with three shares, and a Forecasting group, with two shares. These subgroups share the 50% that the Sales group is allocated. The Analysis group gets 3/5 of the 50%, or 30% of the total CPU resources. The Forecasting group gets 2/5 of the 50%, or 20%.

## Command-line utilities

PRM is a comprehensive product. In addition to an easy-to-use GUI tool for configuration, PRM has several command-line utilities for administrators and users to control and monitor its operation. The PRM command-line utilities provide the same control functionality as the PRM GUI, plus additional monitoring features.

- `prmconfig`—Enables an administrator or an automated script (with appropriate authorization) to change the PRM configuration by loading a new PRM configuration definition, enable or disable individual resource controls, and modify operational parameters for individual resource controls. `prmconfig` also displays the current PRM configuration information and operational parameters.
- `prmlist`—Displays the current PRM configuration information and operational parameters.
- `prmmonitor`—Displays the resource consumption of the PRM groups.
- `prmrun`—Invokes the specified application in a specified PRM group. Non-administrative (non-root) users must have authorization to access the specified PRM group.
- `prmmove`—Moves the specified process, process group, or user's processes to the specified PRM group. Non-administrative (non-root) users must have authorization to access the specified PRM group.
- `prmanalyze`—Shows patterns in resource usage to help you fine-tune your PRM configurations. In addition, it can be used even when you are not using PRM to perform resource use analysis and capacity planning.
- `prmavail`—Displays available system CPU, memory, and LVM volume group/VxVM disk group resources.

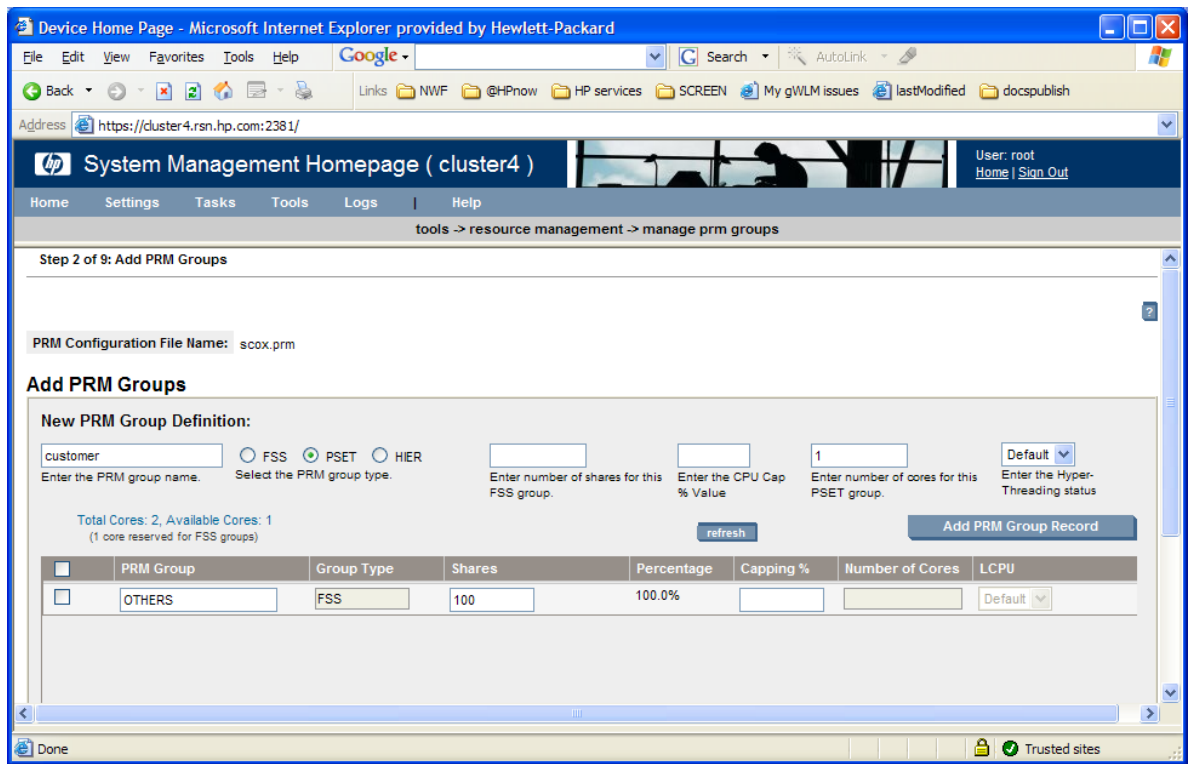
The PRM command-line utilities can be executed from properly authorized scripts or applications to modify PRM operation as needed to adjust to changing workloads and priorities.

## Graphical interface

PRM provides a graphical user interface, or GUI, that simplifies the monitoring and configuration of HP PRM. The PRM GUI, which is integrated with both HP System Management Homepage and HP Systems Insight Manager, enables you to create and modify configuration information through easy-to-use dialog screens. The GUI also enables you to control PRM operation by loading configurations and enabling or disabling individual controls for CPU, memory, and disk I/O bandwidth resources.

The following figure shows the GUI and its CPU controls. A new PRM group named “customer” based on a pSet is about to be added.

Figure 4. PRM GUI



## Isolating the CPU and memory of an application

With HP-UX 11i v1 (B.11.11) and later, you can provide an application with total resource isolation, meaning your application runs in a PRM group that has access to a fixed pool of cores and memory. The PRM group cannot loan these resources to other PRM groups, nor can it borrow additional resources from other PRM groups. If cores are not used, they go idle. If private memory is not used, it is idle. However, both cores and memory remain dedicated to the application.

The application must be assigned to a pSet PRM group to provide isolation of specific CPU resources. (FSS PRM groups guarantee CPU resources, but not specific CPU resources. As a result, FSS PRM groups do not provide isolation.) The private memory isolation capability of the Memory Resource Group (MRG) feature must be enabled to provide isolation of the memory resource.

Fixed and unchanging CPU and private memory resource entitlements allow for more deterministic behavior because surplus CPU and memory resources are not available and any loaned-out resources do not have to be reclaimed when needed.

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

HP PRM, HP-UX Workload Manager (WLM), and HP Global Workload Manager (gWLM) all control system resources. Consequently, you should not use more than one of these products at a time on a single host.

## Integrating HP Process Resource Manager with other products

HP PRM can be easily integrated with many applications to ensure more efficient use of your servers. Also, it is already integrated with HP-UX on several levels. The various types of integration are discussed in the following sections.

### Integration with HP Integrity Virtual Machines

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

Given a system with Integrity VM installed, you can run PRM inside any of the virtual machines; however, you cannot run PRM on the VM Host because the `vm_fssagt` already controls FSS groups on behalf of Integrity VM.

### Integration with Oracle, Sybase, and Informix

By using the PRM application manager described earlier, PRM provides the ability to control resource shares for multiple database instances executing on a single HP-UX server. This feature is useful for ensuring that each database instance can provide appropriate response times.

For example, suppose an HP-UX server contains two database instances, one for an inventory database and one for a personnel database. Suppose also that the inventory database is large and heavily used. By creating separate PRM groups for each database instance (inventory and personnel), PRM ensures that requests to the personnel database receive adequate resources so that they are not affected by the load on the inventory database.

The PRM application manager enables you to assign the database processes that service a particular instance to the same PRM group. This feature allows all processes that service a single database instance to receive the same resource priorities, while allowing each separate database instance to have its own resource priority.

### Integration with HP Serviceguard

HP Serviceguard provides high-availability features in a multisystem environment. It ensures that critical applications are always available, regardless of system hardware, system software, and network or application failures. Serviceguard will automatically fail over applications to active or idle standby systems as needed.

In the event of an application failover, PRM can be used to control how system resources are allocated, given a new load on the failover system, by having the Serviceguard startup scripts for the failed-over packages direct PRM to use a different PRM configuration file. This new PRM configuration file takes into account that another workload is on the system. This change to the PRM configuration is



dynamic and requires no downtime for PRM or any of the applications managed by PRM. The failed-over application and all other existing applications on the standby system will get resources based on shares configured in the new PRM configuration.

## Integration with HP System Management Homepage

The HP System Management Homepage (HP SMH) is a web-based interface that consolidates and simplifies single system management for HP servers on HP-UX, Linux, and Microsoft® Windows® operating systems. By aggregating the data from HP web-based agents and management utilities, HP SMH provides a common, easy-to-use interface for displaying hardware fault and status monitoring, performance data, system thresholds, diagnostics, and software version control for an individual server.

PRM, which runs only on HP-UX, provides an SMH interface so you can configure and monitor PRM on the local system.

## Integration with HP Systems Insight Manager

HP Systems Insight Manager (HP SIM) provides a single point of administration for multiple HP-UX systems. The following PRM functionality is available through HP SIM on nodes in the cluster that have PRM installed:

- Monitor PRM groups
- Configure PRM groups
- Display resource usage
- List resource availability

## Forming Secure Resource Partitions

The HP-UX 11i v2 Security Containment feature provides process and file isolation inside secure compartments. The combination of PRM and Security Containment, which creates Secure Resource Partitions, is powerful. PRM by itself dedicates resources to each application instance. However, those application instances can still interact with other processes, access files, and even interact with system processes. When Security Containment is combined with PRM, 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, which 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 (in which each application runs by itself in an operating system instance on its own server).

## Application monitoring with HP Glanceplus

Previously, with the one-application-per-server model, resource consumption monitoring on each system was sufficient. Resource monitoring on a per-application basis becomes much more important as multiple applications are consolidated onto a single system.

HP Glanceplus provides a graphical display of system resource usage. Glanceplus is integrated with PRM to provide usage information for each PRM group. Glanceplus can also be configured to generate alarm indications if usage of a particular resource in a PRM group reaches a specified threshold.

The alarm feature of HP Glanceplus, which is activated when a resource reaches a threshold for a PRM group, is an excellent tool to discover and fix resource problems with applications before they

occur. The Glanceplus application resource alarm can send an e-mail or page to the system administrator.

## Integration with HP-UX commands

HP developed PRM. PRM is implemented with tight integration with the HP-UX kernel. In addition, various HP-UX commands have been enhanced to support PRM. Some commands, such as `id` and `ps`, display PRM group information. Other commands, such as `cron` and `login`, ensure jobs or processes are assigned to the correct PRM groups. Also, system calls such as `exec` and `fork` ensure processes start in the correct PRM groups.

## Integration with HP-UX accounting

HP-UX has excellent accounting features. For instance, you can log the resources consumed by every process that ran on the system. As an example, the total CPU resources consumed by every process can be logged. These accounting logs are particularly useful as input for chargeback applications.

With application consolidation, you might have situations in which multiple business units are sharing a common server. Accounting and chargeback applications become even more important with application consolidation. Tracking usage and charging back resources consumed by a PRM group is usually much more efficient and meaningful than charging on a per-process or per-user basis.

PRM has a utility called `prmanalyze` for integrating accounting information and resource use. This utility scans accounting files for information on the desired resource type (disk, memory, or CPU) and orders the accounting records by the requested sort key (user, UNIX group, command name, or PRMID). You can use `prmanalyze` to find patterns in resource usage and then change your PRM configurations accordingly.

In addition, you can use `prmanalyze`—even when you are not using PRM—to perform resource use analysis and capacity planning.

With `prmanalyze`, you can generate three classes of reports:

- **Summary**—This report shows who consumes the resources and what the averages are on a high level. It can help you identify what users or applications must be restrained or guaranteed more resources. This report can be used when creating a new PRM configuration.
- **Time-based (hourly, daily, weekly, monthly)**—These reports provide data on resource use over a given time for all the available accounting data. These reports can help you determine what part of the day (or hour, week, or month) each resource is most used. They also identify the users and applications involved in the resource consumption. These reports can be used when enhancing an initial configuration to give special attention to users or applications. Also, they can be used when creating multiple configurations to implement at different times over a given interval.
- **Conflict**—This report provides the most detail, highlighting only the instances when resources are scarce and users are in conflict. This report can be used when fine-tuning a configuration. This report catches items that are missed by the time-based reports. After identifying conflicts, you can determine how much resource each PRM group needed during each conflict and then determine what share of the resource the PRM group actually received. With this data, you can locate users and applications that are not getting as much of the resource as they should. You can also locate the parties involved most often and least often in the conflicts.

Example: Checking for patterns and configuration accuracy

This example shows output from the `prmanalyze` command. The example assumes a single-core system.

Regularly examine daily reports for patterns and configuration accuracy. For reports on recent data, add the `-p` flag to catch jobs that never exit or that run for several days.

```

# prmanalyze -s prmid -r cpu -p -t daily -x 0 accounting_file
daily CPU report by PRM id begins at Thu Jul 8 10:11:00
ave CPUs threshold 0.01

      unique id      ave CPUs      peak CPUs      total secs      % total
Jul 8
      1              0.02         0.55           1195.84          11.59
      2              0.09         0.88           7439.40          43.08
      3              0.05         0.56           4116.09          23.82
      4              0.01         0.14           1226.88           7.11
      5              0.03         0.17           2479.65          14.36
Jul 9
      1              0.02         0.60           2208.72          11.62
      2              0.09         0.87           7890.23          41.51
      3              0.06         0.60           4833.73          25.43
      4              0.01         0.15           1699.32           8.94
      5              0.02         0.14           2442.53          12.85
Jul 10
      1              0.00         0.10           193.63           2.42
      2              0.09         0.88           7348.53          91.89
      3              0.00         0.08           180.96           2.26
      4              0.00         0.04           198.73           2.48
      5              0.00         0.01           74.50            4.15

```

This daily report indicates that the CPU resource is idle most of the time for this period. This result is normal for a business that only uses its computers from 9:00 a.m. to 5:00 p.m. During the week, the CPU usage does not vary by more than about 10%, which is a good indication that the current configuration is working. However, the report for Saturday, July 10 has what appears to be an anomaly. Group 2 is taking up almost all the machine. On closer examination though, the administrator finds that the total seconds used is about the same as every other day, but all the other groups went virtually idle on the weekend. This application might be able to do its job even faster if the administrator took off the memory cap for group 2 only on the weekends. Because there is no contention, a second configuration file could be created to repeal all memory records and change the CPU allocations for the weekend.

Another item to note in the report is that group 1, OTHERS, has bursts of high activity relative to its normal levels. It might be worthwhile to do a CPU conflict report, excluding known applications, to discover who the offender is.

## Conclusion

All aspects of business today are under increasing pressure to do more with fewer resources. This is especially true in the IT environment. IT departments must meet service level agreements, provide superior service to disparate departments, and provision and maintain the information engine that helps enable company growth. It is clear that reducing system management costs and improving computer system resource utilization is an important part of the solution.

HP Process Resource Manager is a key technology for providing an adaptive approach to IT that pools and shares resources so that utilization is optimized and supply automatically meets demand.

## Related information

The following references provide useful information on related products and topics:

- HP Process Resource Manager (PRM)  
<http://www.hp.com/go/prm>
- HP Process Resource Manager User's Guide  
<http://www.docs.hp.com/hpux/ha/>
- HP Virtual Server Environment  
<http://www.hp.com/go/vse>
- HP-UX Workload Manager  
<http://www.hp.com/go/wlm>
- HP Serviceguard  
<http://www.docs.hp.com/hpux/ha/>
- HP-UX 11i v2 Security Containment  
<http://www.hp.com/go/securitycontainment>
- "Using PRM with Oracle Databases"  
<http://www.docs.hp.com/hpux/ha/>

## For more information

For more information on HP Process Resource Manager, contact any of the HP worldwide sales offices or see the HP website at <http://www.hp.com/go/prm>.

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

© 2004–2007 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Microsoft and Windows are U.S. registered trademarks of Microsoft Corporation. Oracle is a registered U.S. trademark of Oracle Corporation, Redwood City, California. UNIX is a registered trademark of The Open Group.

5982-4359EN, Rev. 4 June 2007

