

*HP Raises the Bar for
UNIX Workload
Management*

April 2002

*A D.H. Brown Associates, Inc.
White Paper Prepared for
Hewlett-Packard*

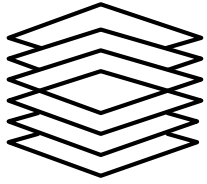
This document is copyrighted © by D.H. Brown Associates, Inc. (DHBA) and is protected by U.S. and international copyright laws and conventions. This document may not be copied, reproduced, stored in a retrieval system, transmitted in any form, posted on a public or private website or bulletin board, or sublicensed to a third party without the written consent of DHBA. No copyright may be obscured or removed from the paper. All trademarks and registered marks of products and companies referred to in this paper are protected.

This document was developed on the basis of information and sources believed to be reliable. This document is to be used "as is." DHBA makes no guarantees or representations regarding, and shall have no liability for the accuracy of, data, subject matter, quality, or timeliness of the content. The data contained in this document are subject to change. DHBA accepts no responsibility to inform the reader of changes in the data. In addition, DHBA may change its view of the products, services, and companies described in this document.

DHBA accepts no responsibility for decisions made on the basis of information contained herein, nor from the reader's attempts to duplicate performance results or other outcomes. Nor can the paper be used to predict future values or performance levels. This document may not be used to create an endorsement for products and services discussed in the paper or for other products and services offered by the vendors discussed.

TABLE OF CONTENTS

EXECUTIVE SUMMARY	1
WORKLOAD MANAGEMENT REQUIREMENTS	3
WORKLOAD MANAGEMENT TOOLS	5
<i>Entitlement-Based Resource Managers</i>	5
Figure 1: Entitlement-Based Resource Manager Operation	6
<i>Goal-Based Workload Managers</i>	6
Figure 2: Goal-Based Resource Manager Operation	7
<i>Tradeoffs Between Entitlement-Based and Goal-Based Approaches</i>	7
PARTITION INTEGRATION	9
HIGH AVAILABILITY CLUSTER INTEGRATION	10
UTILITY PRICING INTEGRATION	11
APPLICATION INTEGRATION	11
PRODUCT POSITIONING	12
HP PROCESS RESOURCE MANAGER (PRM)	12
HP-UX WORKLOAD MANAGER (HP-UX WLM)	12
AIX WORKLOAD MANAGER (AIX WLM)	14
SOLARIS RESOURCE MANAGER (SRM)	15
TRU64 UNIX	16



HP Raises the Bar for UNIX Workload Management

EXECUTIVE SUMMARY

As UNIX servers are increasingly deployed in mainframe-like roles, workload management functions have risen to the top for UNIX operating system differentiation. In addition to supporting the basic resource management capabilities needed to simultaneously run multiple dominant applications on a single server, workload management tools must now address other critical needs for deploying UNIX in datacenter and enterprise environments. These include capabilities for:

- Specifying performance requirements using high-level metrics based on application behavior, rather than simply relying on low-level metrics such as processor usage.
- Integrating with partitioning functions in order to work in tandem with HA (high availability) cluster packages and make automatic adjustments in failover situations.
- Activating Capacity-on-Demand (CoD) resources automatically, as well as taking the increased processing power into consideration when these resources become active.
- Coordinating native workload management functions with application-specific resource management functions in dominant applications such as databases.

Workload management tools have existed on UNIX systems for some time (usually referred to as “resource management” tools), and the leading UNIX systems now all effectively address the most basic requirements. HP offers a comprehensive set of workload management tools for its HP-UX operating system that address all relevant functional requirements faced by today’s IT environments. HP’s Process Resource Manager (PRM) add-on has offered resource management functions in HP-UX for several years. HP PRM allows system administrators to set policies for how the system will allocate processor, real memory, and I/O resources to users, groups of users, and applications. HP also offers a goal-based workload management tool called HP-UX Workload Manager (HP-UX WLM) as an extension to PRM. HP-UX WLM remains the only UNIX-based product to allow specifying workloads using Service Level Objectives (SLOs).

These capabilities all give HP-UX users a clear advantage for meeting several important business goals, including the ability to:

- Maximize the administrative and cost benefit from their server-consolidation efforts.

- Closely align IT's management practices to business performance objectives.
- Deliver more consistent service levels to their enterprise customers, external customers, suppliers, and partners at lower costs.
- Plan for increases in system resources to maximize application performance.

WORKLOAD MANAGEMENT REQUIREMENTS

As administrators react to the rampant server proliferation that has resulted from client-server computing by consolidating multiple workloads on larger systems, centralization has once again become fashionable. Many IT organizations want to better employ computing resources by consolidating their datacenters onto fewer systems. They also prefer to reduce administrative expenses by cutting the number of servers they must maintain. Server consolidation has thus become an attractive option to businesses that seek to control the increasing cost of managing distributed application servers by improving resource usage. But when administrators try to run multiple “dominant” applications on a single server, ensuring consistent responsiveness becomes a significant challenge. Since critical applications are often designed to dominate the resources of a dedicated server, administrators must use specialized management functions to deploy many such applications simultaneously on SMP servers.

Web-based applications also introduce a need for flexible and guaranteed application service levels. Web applications make back-end corporate computing infrastructures accessible to customers, partners, suppliers, and investors via e-business front-ends. Web applications also tend to involve wildly unpredictable workloads, with a high rate of overall growth in workload size. Therefore, providing consistent and high levels of performance for different web applications can directly affect a company’s online brand image and revenues.

UNIX SMP servers have become particularly attractive options for consolidating servers and hosting web applications. Because they support 32 or more processors in SMP configurations, UNIX servers can be used to consolidate multiple smaller (four- and eight-way) servers now used predominantly for managing departmental and branch functions. UNIX servers have also become prime targets for deploying web applications due to their strong scalability and reliability, and their historic affinity with Internet protocols. Consolidating multiple dominant applications onto a single UNIX server requires advanced workload management tools to ensure that critical applications receive enough resources to complete their workloads. Running business-critical web applications on UNIX systems requires these tools to optimize resource usage in a network environment, and maintain performance for workloads running in specific environments, i.e., web application servers.

Workload management tools allow large numbers of resource-intensive applications to run simultaneously on a single server through flexible scheduling policies, and are thus key enablers for a variety of server-consolidation and web application tactics. These tools work within a single operating system instance to effectively manage constantly changing workloads. As a result, multiple dominant applications can coexist in a single environment. Workload management tools allow administrators to gain a thorough understanding of application behavior under load, the ability to anticipate what expected loads will be, and the ability to develop policies governing user and department entitlements. They also allow

administrators to change resource allocation very rapidly and with maximum precision, using scripts, traditional system management tools, and other IT infrastructure components.

Workload management tools have existed on UNIX systems for some time (usually referred to as “resource management” tools), and they have effectively addressed many of these requirements. However, as UNIX servers continue to gain new functions, and are increasingly deployed in mainframe-like roles, the bar has been raised, introducing new requirements. In addition to supporting basic resource management capabilities, workload management tools must now also address the following issues:

- *High-Level Metrics:* Workload management tools should allow administrators to specify performance requirements using high-level metrics based on application behavior, rather than simply relying on low-level metrics such as processor utilization.
- *Integration with Partitioning Functions:* On some servers, partitions can be used to run multiple instances of an operating system simultaneously on a single server. Workload management tools need to work properly when partition boundaries are changed dynamically, and should allow applications needing additional resources to draw on resources in neighboring partitions, if these partitions have excess resources available.
- *Integration with High Availability (HA) Clusters:* In HA clusters, a failover event increases the workload of backup servers, because they must absorb at least part of the workload from the failed primary server. Workload management tools should be configurable to make automatic adjustments in failover situations, so that applications already running on the backup server continue to deliver acceptable performance.
- *Integration with Capacity-On-Demand (COD) Offerings:* UNIX vendors have begun to offer customers the ability to increase the processing power of systems without disrupting operations, usually by pre-installing processors that can be activated as needed, for a fee. Workload management tools need to have some way of activating COD resources automatically, as well as tapping into the increased processing power when these resources become active.
- *Integration with Key Applications:* Some particularly dominant applications, such as database systems, effectively perform their own resource management. The native workload management functions need to coordinate their activities with these application-specific resource management functions to ensure consistent performance for the application.

The following sections describe each of these issues in greater detail.

WORKLOAD MANAGEMENT TOOLS

Workload management tools efficiently allocate system resources such as CPU, memory, and I/O to different applications with flexible scheduling policies. These functions effectively override the operations of the default UNIX scheduler, taking customized policies into consideration instead. Currently two classes of workload management tools are available for UNIX systems: *entitlement*-based resource managers and *goal*-based workload managers.

ENTITLEMENT-BASED RESOURCE MANAGERS

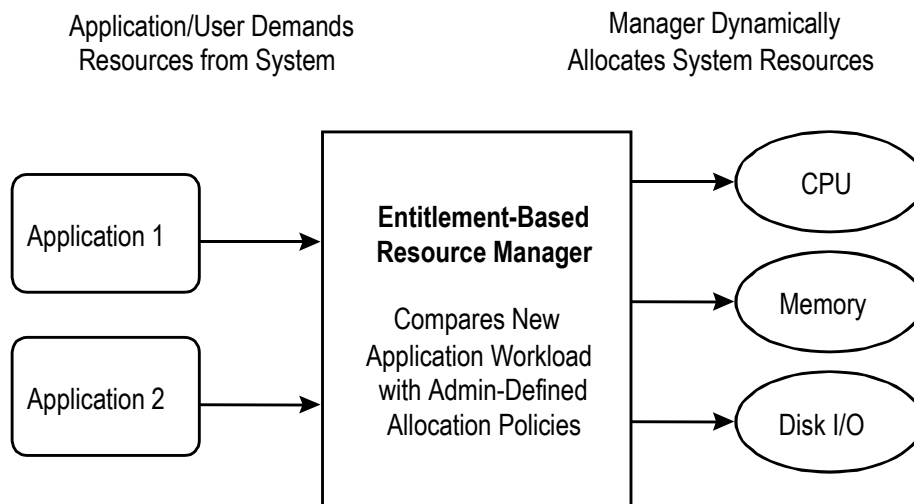
Entitlement-based resource managers take a bottom-up approach. Different business applications have separate business priorities – for example, an enterprise requirements planning (ERP) application affects a business more than an internal human resources web server does. If both these applications are consolidated onto the same server, the ERP application should receive a larger portion of the available system resources than the web server applications. Entitlement-based resource management solutions ensure that applications receive resources commensurate with their business importance.

With entitlement-based resource managers, system administrators implement the resource controls that indicate the importance of a particular application to the business. Administrators write allocation policies that specify resource percentages, allocate resource shares, or define resource limits for different users, groups, or applications. As noted above, each workload's significance to the business controls the amount of resources it gets. The following rules might apply to our example:

- The ERP application's memory use should not drop below 100 MB.
- Batch processes from accounting should receive 10% of processor time.
- Database queries from executives should be granted 60% of I/O bandwidth.

An entitlement-based resource manager would analyze both the existing and the incoming workloads according to these policies (see Figure 1). It would then allocate system resources among the workloads to match these policies. These resource managers perform such dynamic reallocation on an ongoing basis, thereby allowing mission-critical applications better access to more system resources while ensuring less critical workloads still get some resources. Resource managers also can keep a single user or a process with a memory leak from unfairly monopolizing memory and starving other processes and users.

FIGURE 1:
*Entitlement-Based
Resource Manager
Operation*



GOAL-BASED WORKLOAD MANAGERS

Goal-based workload managers take a top-down approach, which starts with the business goals and priorities of multiple application workloads. Particular workloads can have very different business objectives that translate into specific application response times or batch turnaround times. A traditional example is payroll processes, which must be completed in a certain amount of time for employees to be paid on time. e-Retailers present a different possible scenario, in which catalog database transactions must occur in accordance with certain response times to ensure a satisfactory level of service.

As noted above, goal-based workload managers adopt a performance approach to managing workload tasks. IT managers specify business-oriented priorities and performance targets for each application workload that runs on the server. Returning to our original examples, a goal-based manager might operate using one or more of the following principles:

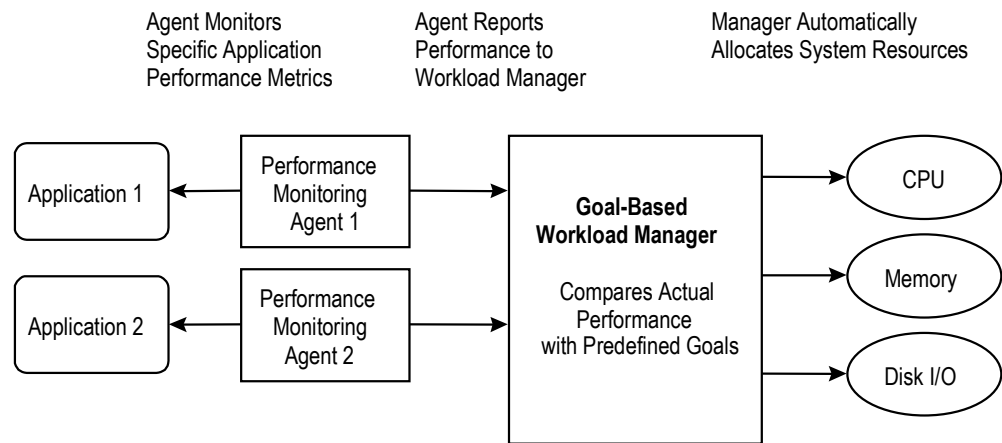
- Give workloads from the ERP application top priority.
- Maintain specified runtimes for batch processes.
- Maintain specified response times for transaction- or query-based applications.

Depending on the principles specified, the goal-based workload manager then oversees workload priorities and system resources to achieve the business objectives for all active workloads. Goal-based managers dynamically perform this manipulation, automatically making adjustments on an ongoing basis. Thus, administrators do not have to specify and constantly update priorities and resource allocations for each application as the workload mix changes.

Application and batch process performance monitoring capabilities are required for goal-based workload managers to operate effectively (see Figure 2). For example, applications can be instrumented for detailed performance monitoring or dummy transactions can be used to estimate response times. The monitoring agent collects the actual performance value, which the manager uses to determine

whether the specified goal is being achieved. If application performance is not meeting business standards, the workload manager automatically allocates more resources to the application.

FIGURE 2:
*Goal-Based Resource
Manager Operation*



TRADEOFFS BETWEEN ENTITLEMENT-BASED AND GOAL-BASED APPROACHES

Both entitlement-based resource management and goal-based workload management deliver more administrative control over system usage and reduce the wasting of computing resources by consolidating multiple applications on a single system. Differences between the two types of solutions show up, however, when application performance is measured.

Entitlement-based systems can produce wide performance variances depending on their mode of operation and how many applications use the server at one time. For example, a system administrator may allocate a percentage-based portion of resources to each application so the total entitlements sum to 100%. This approach is known as the non-capping mode. If a design application received 100 shares of CPU time in non-capping mode, the resource manager would let it have 100% of the CPU when it was the only application running. But when an ERP application with 200 shares of CPU started, the resource manager would reallocate the resources, so the design application only got 33% of the CPU. The user of the design application would see a drastic drop in performance.

The wild changes in application performance can be mitigated if an entitlement-based system runs in capping mode. In this approach, the administrator creates more sophisticated allocation rules that include resource maximums and minimums. This scenario would keep the design application from getting 100% of the system even if it were the only application running, thereby smoothing out the performance seen by the end user.

The capping-mode workaround does not address the fundamental issue, however, which is that business applications should be managed according to their performance. With multiple applications from potentially different departments now residing on the same server, the IT administrators can

guarantee similar or better application performance from the new consolidated server. The more important an application is to the department's mission, the more critical its performance consistency becomes.

Web applications add another element to this equation by exposing internal corporate applications and databases to external customers and investors. The performance of web applications now forms the basis of the customer shopping experience, which directly affects a corporation's image and branding. For these reasons, systems administrators must focus on managing systems and resources in terms of measurable performance metrics. Entitlement-based resource managers have no way to automatically fold application performance information into the feedback loop for dynamic readjustment of allocation policies. Instead, administrators have to monitor application performance separately and then manually readjust the allocation policies. Most enterprise IT departments are already overburdened, leaving little time to perform this task effectively on an ongoing basis.

The extra administrative effort required to manage application performance undermines the management benefits of moving to a consolidated server environment. The prime benefit of server consolidation is reducing the effort and expense required for managing the corporate infrastructure. If managing the performance of a consolidated server requires a lot of extra effort, the benefit is lessened.

The solution – automating the performance-feedback and allocation-policy adjustment processes – allows administrators to ensure the performance delivered to users of an application aligns with the business goals of the enterprise. Goal-based workload managers deliver this capability, enabling administrators to focus on managing performance objectives and business priorities while the software translates these principles into resource allocations.

It should be noted that goal-based workload managers do not preclude entitlement-based management systems. Indeed, entitlement-based resource-management capabilities are a necessary prerequisite for goal-based management. Goal-based workload managers must be able to allocate resources in a more sophisticated manner than the simple process prioritization that was standard on UNIX systems not long ago. As discussed above, however, goal-based workload managers are superior to entitlement-based systems in terms of aligning IT to business goals. Ultimately, in the web application environments, enterprises will prosper or languish by the performance of their applications. IT organizations can significantly contribute to business success by organizing their system-management capabilities around application performance metrics.

PARTITION INTEGRATION

Partitions allow administrators to run multiple instances of an operating system within a single server. Each instance behaves as if it were running on a standalone machine. Barriers between the different environments maintain overall system robustness, so that even the most extreme application failure or operating system crash in one partition leaves the others unaffected. Each partition receives a subset of the processors, memory, and I/O. Partitions remain isolated from each other, so, if a failure occurs in one, the remaining partitions continue to function. The entire environment, i.e., all partitions, can be managed from a single point.

Several partitioning mechanisms have emerged for UNIX systems, based on hardware, software, or intermediate “firmware” approaches:

- *Hardware-based partitions* typically provide the greatest-level of protection between environments, working at the electrical level to maintain “bullet-proof” isolation. However, reconfiguring hardware-based partitions can be unwieldy, requiring substantial operator intervention. Also, hardware-based partitions usually work only on the granularity of processor sets, which potentially wastes resources in workloads that do not exactly match the count of processors in the set.
- *Software-based partitions* provide greater flexibility in terms of granularity, supporting environments that run on a single processor. Also, it is usually easier for the boundaries of software-based partitions to be changed online (i.e., without rebooting any of the partitions involved).
- *Firmware-based partitions* use a hybrid of both approaches in an attempt to merge the guaranteed protection of hardware-based partitions with the flexibility of software-based partitions.

Workload management tools must be compatible with partitioning because of the potential ability for partitions to be resized dynamically, which fundamentally affects the workload behavior within the affected partitions. Partition resizing is often invoked to accommodate business cycles. For example, a single server might host multiple department systems by day, and then temporarily reconfigure the resources for all partitions into the central database server’s partition at night to run large batch jobs, restoring them in the morning. In this case, the workload management tools would have to support different modes of operation, depending on the partition configuration. The correct configuration needs to be deployed automatically as partition boundaries are changed.

Workload management tools can also be integrated with the partition management mechanisms so that if applications need more resources than are available in a given partition, they can draw on resources in neighboring partitions, assuming they have excess resources to spare.

HIGH AVAILABILITY CLUSTER INTEGRATION

High availability (HA) cluster techniques maintain the availability of applications by failing over to a backup system in the event of system outage due to any failure – hardware, software, or otherwise. HA clusters allow one or more servers to take over for a server that has crashed or stopped processing normally, allowing processing to continue. By isolating faults on the failed node, the remaining nodes can continue functioning, keeping the overall clustered system in operation, albeit at reduced capacity.

In HA clusters, special software monitors the health of systems and applications by running agents that continuously probe for certain conditions. Vendors usually provide agents for monitoring hardware, the operating system, and key applications such as databases and messaging systems. They typically also provide an API that developers can use to configure monitoring of their own applications. When agents detect a failure, they can trigger a variety of actions, depending on the configurability of the clustering package. First, the system must decide whether to attempt a local recovery or initiate a failover, in which case the workload is moved to a backup server. Sophisticated HA packages support more than two nodes, which enables cascading and multidirectional failover. Cascading failover provides higher levels of reliability by allowing the workload to continue migrating to yet another backup node if the primary backup node fails. Multidirectional failover allows a failed node's workload to be split and failed over to multiple backup nodes. Both capabilities deliver a fundamental impact on the workloads of backup nodes, and require special functions in workload management software to coexist properly.

In an HA clustering configuration, backup nodes must always run at a fraction of their potential usage because they must be prepared to absorb the workloads of failed primary nodes. For example, in a two-node cluster, neither node could run at more than 50% usage because each must potentially take over the other's workload (at which point the backup node would be running at 100% usage). Multidirectional failover improves usage in HA clusters because the workload of a failed node can be spread out across more nodes. For example, with multidirectional failover in a four-node cluster, if node 1 runs three applications, A, B, and C, upon failover application A would move to node 2, application B to node 3, and application C to node 4. Before failover, each backup node could run at 66% average use (rather than 50% in a two-node cluster), since only one-third of the failed primary node's workload (equaling 33% use of a node) would be failed over to each of the remaining nodes.

Since HA failover clearly has a dramatic impact on the workloads of clustered servers, workload management tools become deeply involved in the failover process. To work properly in an HA cluster environment, workload management tools need to adjust their configuration appropriately to handle the new workload mix when a failover occurs. Then, when the primary node recovers and its applications return, the workload management tools running on each node need to resume their original configuration.

UTILITY PRICING INTEGRATION

COD options allow users to increase the processing power of systems without disrupting operations. Typically, COD programs involve the purchase of fewer processors than are actually installed in the system, introducing a distinction between the physical installation and the purchaser's license to use. Extra processors remain idle until more capacity is needed, at which time users license these processes and the operating system activates them. COD options have long been available to users of traditional high-end commercial systems such as mainframes, for which users could "lease" capacity by paying a regular fee. On UNIX systems, COD programs have proven particularly attractive for addressing the wildly fluctuating workloads of web applications related to e-commerce.

Workload management tools can also be integrated with the COD mechanisms that enforce processor licensing by triggering a processor addition when certain load thresholds are reached, or a removal (if the COD contract allows it) when the workload drops below the threshold.

APPLICATION INTEGRATION

Certain types of dominant applications, such as databases, make the assumption that they can take control of all processor resources managed by the operating system. Then they typically impose their own resource management mechanisms to allocate the response given to applications that depend on the database. For these applications to be properly managed, they need to be integrated directly with the native workload management tools. This calls for the workload management system to provide APIs that allow applications to register their requirements, and respond to reconfiguration requests from the system.

Another way for workload management tools to manage dominant applications is be able to manage operating system processor sets. Some operating systems support a function called processor sets that can restrict applications to a specific group of processors within the system. If the dominant application can be restricted to a processor set, the native workload management software can use the remaining processors to deliver resources to other applications.

PRODUCT POSITIONING

As UNIX servers are increasingly deployed in mainframe-like roles, workload management functions have become an important area of solution differentiation for UNIX operating systems. In response, UNIX systems developers have started to compete heatedly for technical leadership in the workload management area. Below are some of the native workload management solutions for the leading UNIX systems today.

HP PROCESS RESOURCE MANAGER (PRM)

HP has offered resource management functions in HP-UX for several years with its Process Resource Manager (PRM) add-on. HP PRM allows system administrators to set policies for how the system will allocate processor, real memory, and I/O resources to users, groups of users, and applications.

In HP PRM, the system administrator identifies resource groups called PRM groups. Each PRM group gets a percentage of the total processor resources and can also obtain a percentage of disk I/O bandwidth and system physical memory. PRM's resource allocation policies are written in a hierarchical manner, so that control can be as granular as the administrator requires. In addition, administrators can specify resource allocation minimums and resource allocation caps to better control application performance. The system administrator then attaches applications to PRM groups and provides a set of rules that will classify users into PRM groups. When the system is at 100% usage, each PRM group will receive its allocated percentage of processor resources. On a busy server, the PRM scheduler will inform the HP-UX scheduler regarding the PRM group from which it should schedule the next process. Within each PRM group, HP-UX uses its normal scheduling process.

HP-UX WORKLOAD MANAGER (HP-UX WLM)

HP also offers a goal-based workload management tool called HP-UX Workload Manager (HP-UX WLM) as an extension to PRM. HP-UX WLM allows dynamic resource allocation to be performed based on the performance of an application. Administrators specify Service Level Objectives (SLOs) for each application and conditions under which this performance goal applies, such as date and time ranges. Each SLO must have a priority and minimum and maximum entitlements. This allows the manager to determine resource allocations while preventing resource-hogging by an application.

Administrators must also supply monitoring agents that report real-time performance metrics to HP-UX WLM. HP promotes the use of the Application Response Measurement (ARM) management standard for instrumenting applications to provide detailed transaction performance information. For existing and off-the-shelf applications with no instrumentation, administrators

can use dummy transactions to estimate application performance. For batch processes, the agent should track CPU consumption and report an estimate of the total time taken to complete the job to HP-UX WLM.

HP-UX WLM then compares these actual performance reports to the specified goals. If application performance does not meet the business objectives or goals, HP-UX WLM automatically updates PRM's configuration file and allows allocation of more resources to the application, up to the predefined maximum. These resources get borrowed from low-priority workloads so that mission critical applications are not affected. If the SLOs are still not being met, then the system notifies the Event Monitoring Service (EMS). EMS sends out an alarm to the system administrator, who in turn addresses the problem.

HP leapfrogged the other UNIX vendors to deliver a goal-based workload management solution by automating the performance-feedback and allocation-policy adjustment processes in HP-UX WLM. Since then, HP has delivered several extensions to PRM and HP-UX WLM to address emerging requirements related to partitions, HA clusters, COD, and key application integration.

HP-UX supports two types of partitioning:

- *nPartitions* are hard partitions that are available on certain HP 9000 servers. These partitions allow multiple instances of HP-UX to be deployed on a single server in groups of four or more processors. Each partition is electrically protected from neighboring partitions, so they are completely unaware of each other.
- *vPartitions* are soft partitions that can be deployed on a single server, or within individual *nPartitions*. *vPartitions* are managed by software, which allows their boundaries to be adjusted dynamically.

Both PRM and HP-UX WLM can be deployed within *nPartitions* and *vPartitions* to manage resources in a partition as if they were single systems. Additionally, HP-UX WLM is integrated with the mechanism that manages *vPartition* boundaries, so that it can automatically balance the resources between *vPartitions* in order to satisfy the needs of the separate workloads in different partitions. That is, if a *vPartition* requires additional resources that happen to be available in a neighboring *vPartition*, HP-UX WLM will automatically move the resources from the less-used partition to the one that needs them.

HP offers comprehensive HA clustering functions for HP-UX as part of its MC/ServiceGuard option. HP-UX WLM cooperates with MC/ServiceGuard by allowing administrators to define an SLO for backup servers that only becomes active if a workload fails over to that server. This makes sure that HP-UX WLM allocates the necessary resources to handle the additional workload at the time a failover occurs, reverting to the original SLO when the added workload fails back to the primary server.

HP introduced its “instant Capacity On Demand” (iCOD) program in 1999, covering both midrange and high-end systems from the start. HP’s approach involves paying for fewer processors than are actually installed in the system at the time of system purchase. The extra processors remain idle until more capacity is needed, at which time users have the option to license and activate them to meet the extra demand. This allows users to increase the processing power of their systems without disrupting operations. HP-UX WLM is integrated with iCOD to automatically turn on/off the iCOD CPUs so that SLOs can be met.

HP offers two approaches to make sure that the resources for key applications are managed correctly. First, HP offers API toolkits, along with sample configuration files, that allow several key third-party applications to be configured for management by HP-UX WLM, including the following:

- Oracle Database Toolkit (ODBTK) for Oracle databases.
- ApacheTK for the Apache web server.
- SASTK for the SAS decision-making software package.
- Duration Management Toolkit (DMTK), which grants jobs only the amount of CPU needed to complete their work within a certain time frame. This package is primarily used in conjunction with SASTK.

These toolkits generally allow an instance of the particular application to be placed in its own workload group. HP-UX WLM can then manage the performance of each instance through prioritized SLOs. Another approach for managing dominant applications such as Oracle databases is to use PRM in conjunction with HP-UX processor sets to make sure that Oracle’s internal Database Resource Manager coexists properly with other applications on the system. This approach involves limiting Oracle to a group of processors, while PRM manages the remaining processors, which prevents unpredictable behavior in either Oracle or the other applications.

AIX WORKLOAD MANAGER (AIX WLM)

IBM includes its workload manager, AIX WLM, in the AIX base operating system. AIX WLM takes a unique, job-oriented approach to resource management. The concept is based on the traditional entitlement approach of most resource management tools, rather than HP-UX WLM’s goal-based approach. AIX WLM can allocate CPU cycles, disk I/O, and memory by user or program, based on different priorities. Like HP’s PRM, it can also allocate real (i.e., physical) memory, because AIX WLM is implemented in the AIX kernel. Administrators define entities called job classes and specify rules by which jobs are assigned to these classes. Applications, users, or groups can be specifically assigned to a class or even excluded from a class. Each job class receives processor and real memory shares that are used to determine resource usage. Incoming workloads are automatically assigned to a job class based on the administrator-defined rules. The system calculates the total shares represented by these active job classes, calculates the percentage of each class’s shares of this total, and then assigns a resource percentage to each class based on the share

percentage. Administrators can change the entire AIX WLM configuration while it is running.

In addition to specifying job classes and rules for resource usage, administrators can assign upper and lower limits to each class. These limits are expressed as a percentage and represent the absolute maximum and minimum amount of a resource that class should consume. Since limits take precedence over targets, they can be used to guarantee certain levels of resource consumption for key classes, regardless of the cumulative effects of fine-tuning shares in large numbers of registered classes. Administrators can further assign classes into tiers representing different service levels. Low-priority tiers get only those resources that are idle in higher-priority tiers. AIX WLM can also support before-and-after AIX WLM comparisons. The system gathers “before” usage statistics by allowing AIX WLM to classify incoming workloads, but not allowing it to regulate resource allocation.

IBM recently began to support firmware-based partitions on its high-end pSeries servers, which allow multiple copies of AIX to be run on a single server. However, the boundaries between these partitions cannot be adjusted dynamically, so they have little interaction with AIX WLM at present.

SOLARIS RESOURCE MANAGER (SRM)

Sun’s Solaris Resource Manager (SRM) is an unbundled option for Solaris that uses an entitlement approach for managing resources. SRM is based on technology that Sun acquired from Aurema’s ShareII. SRM provides the ability to control and allocate CPU time, processes, virtual memory, connect time, and logins, thereby enforcing administrator-defined usage policies. SRM uses a fair-share scheduler to control CPU consumption. Each user, group, or application gets different numbers of CPU shares. As the user, group, or application processes consume CPU services, SRM tracks CPU usage and adjusts the priorities of all processes. This procedure forces the relative ratios of CPU usage to converge on the relative ratios of CPU shares assigned. The scheduler considers only active users and applications.

SRM also imposes limits on an application’s virtual memory and other resource consumption. It tracks overall virtual memory used and per-process virtual memory used. Whenever a process attempts to increase its memory size, for example, it is subject to memory limits (total and per-process).

As with PRM, resource allocation policies are written in a hierarchical manner in SRM, so that control can be as granular as the administrator requires. For example, resources can be allotted to a department, then subdivided into workgroups, and divided again among individuals. SRM also collects resource-usage information on any level of the allocation hierarchy (department, application, user, etc.) for accounting and billing purposes with the “accrue” attribute.

Sun offers hard partitions on some of its SPARC servers called Dynamic Domains. The boundaries between Dynamic Domains can be adjusted dynamically, and SRM collaborates with them by allocating and controlling resources within a defined domain to encourage maximum usage. SRM can also work with processor sets by allocating resources on those CPUs that are not part of the processor sets.

TRU64 UNIX

Tru64 UNIX recently added resource management tools from a third party, Aurema's Active Resource Management Technology (ARMTech), into Tru64 UNIX. ARMTech also uses an entitlement approach for managing resources, allowing administrators to designate fixed or dynamic shares of CPU and memory, and establish limits for applications, processes, suites or users. This allows the administrators to create custom resource consumption policies that dynamically manage distribution of server resources.

ARMTech works by defining resource consumer classes with rule-based memberships. Resource consumers can include users, groups, and applications. For every resource consumer, ARMTech records current and historical usage. ARMTech uses this history and an administrator-defined policy to calculate entitlements for active processes on the server.

Compaq recently began to support hard partitions on its high-end AlphaServers, which allow multiple copies of Tru64 UNIX to run on a single server. However, the boundaries between these partitions cannot be fully adjusted dynamically (only expanded, at present), so they have little interaction with ARMTech.