AlphaServer ES47/ES80/GS1280

Getting Started With Partitions

Version 4.0
January 2005

This document provides information on the concepts of partitioning hp *AlphaServer* ES47/ES80/GS1280 platforms. Specific information is also given on how to configure, plan, and construct hard and soft partitions on these platforms, using the three tools available: the AlphaServer Management Utility (AMU), the AlphaServer Platform Wizard (APW), and the Server Management Command Line Interface (CLI).

Regulatory Notices for Series ET2003

FCC Notice

Part 15 of the Federal Communications Commission (FCC) Rules and Regulations has established Radio Frequency (RF) emission limits to provide an interference-free radio frequency spectrum. Many electronic devices, including computers, generate RF energy incidental to their intended function and are, therefore, covered by these rules. These rules place computers and related peripheral devices into two classes, A and B, depending upon their intended installation. Class A devices are those that may reasonably be expected to be installed in a business or commercial environment. Class B devices are those that may reasonably be expected to be installed in a residential environment (i.e., personal computers). The FCC requires devices in both classes to bear a label indicating the interference potential of the device as well as additional operating instructions for the user. The rating label on the device shows which class (A or B) the equipment falls into. Class B devices have an FCC ID on the label. Class A devices do not have an FCC ID on the label. Once the class of the device is determined, refer to the following corresponding statement.

Class A Equipment

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user will be required to correct the interference at personal expense.

Class B Equipment

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio or television technician for help.

Modifications

The FCC requires the user to be notified that any changes or modifications made to this device that are not expressly approved by Hewlett-Packard Company may void the user's authority to operate the equipment.

Cables

Connections to this device must be made with shielded cables with metallic RFI/EMI connector hoods in order to maintain compliance with FCC rules and regulations.

Declaration of Conformity for products marked with the FCC logo – United States only

This device complies with Part 15 of the FCC rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

Taiwanese Notice

警告使用者:

這是甲類的資訊產品，在居住的
環境中使用時，可能會造成射頻
干擾，在這種情況下，使用者會
被要求採取某些適當的對策。

Japanese Notice

この装置は，情報処理等電波障害自主規制協議会（VCCI）基準に基づくクラス A
情報装置です。この装置を家庭環境で使用すると電波障害を引き起こすことがあり
ます。この場合には，使用者が適切な対策を講じるよう要求されることがあります。

Canadian Notice (Avis Canadien)

Class A Equipment

This Class A digital apparatus meets all requirements of the Canadian Interference-Causing Equipment Regulations. Cet appareil numérique de la classe A respecte toutes les exigences du Règlement sur le matériel brouilleur du Canada.

Class B Equipment

This Class B digital apparatus meets all requirements of the Canadian Interference-Causing Equipment Regulations.Cet appareil numérique de la classe B respecte toutes les exigences du Règlement sur le matériel brouilleur du Canada.

European Union Notice

Products with the CE marking comply with the EMC Directive (89/336/EEC) and the Low Voltage Directive (73/23/EEC) issued by the Commission of the European Community and if this product has telecommunication functionality, the R&TTE Directive (1999/5/EC).

Compliance with these directives implies conformity to the following European Norms (in parentheses are the equivalent international standards and regulations):

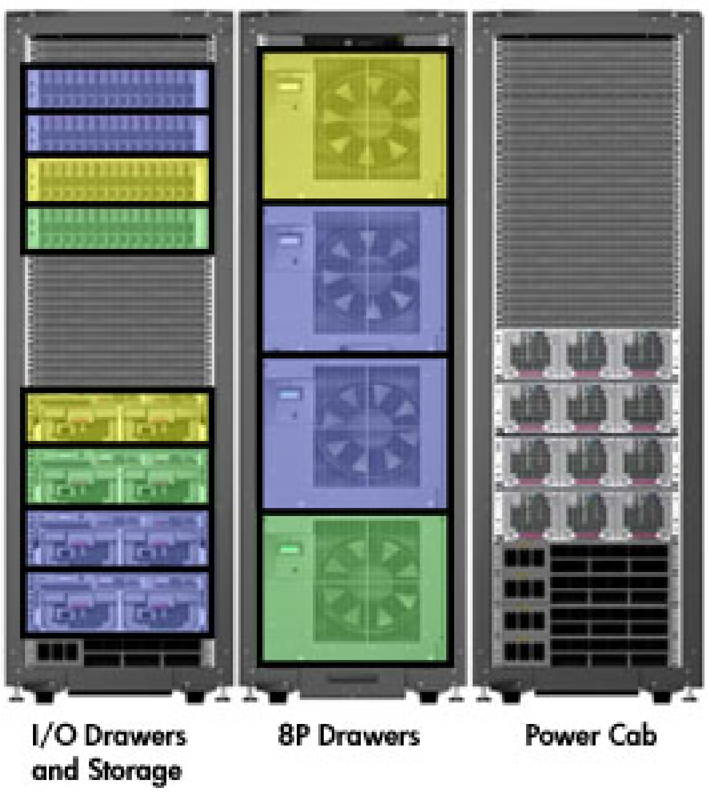
- EN55022 (CISPR 22) – Electromagnetic Interference
- EN55024 (IEC61000-4-2, 3, 4, 5, 6, 8, 11) – Electromagnetic Immunity
- EN61000-3-2 (IEC61000-3-2) – Power Line Harmonics
- EN61000-3-3 (IEC61000-3-3) – Power Line Flicker
- EN60950 (IEC60950) – Product Safety

Table of Contents

Chapter 1 Introduction

1.1 What Are Partitions?

Figure 1-1 Hardware Partitions on a 32P System



An individual hp AlphaServer ES47/ES80/GS1280 platform can be divided into partitions, each capable of running an instance, or copy, of an operating system.

For example, consider the 32P platform shown in Figure 1-1. Each of the 8P drawers contains processors and memory. Each of the I/O drawers contains a PCI card cage with related I/O capabilities.

Each partition must have some CPU resources, some memory, and some I/O capability. In this way, a partition provides the attributes of a separate system, capable of functioning independently.

In the picture, consider all the yellow boxes to comprise one partition; all the green boxes, a second partition; and all the blue boxes, a third partition.

1.2 Why Use Partitions?

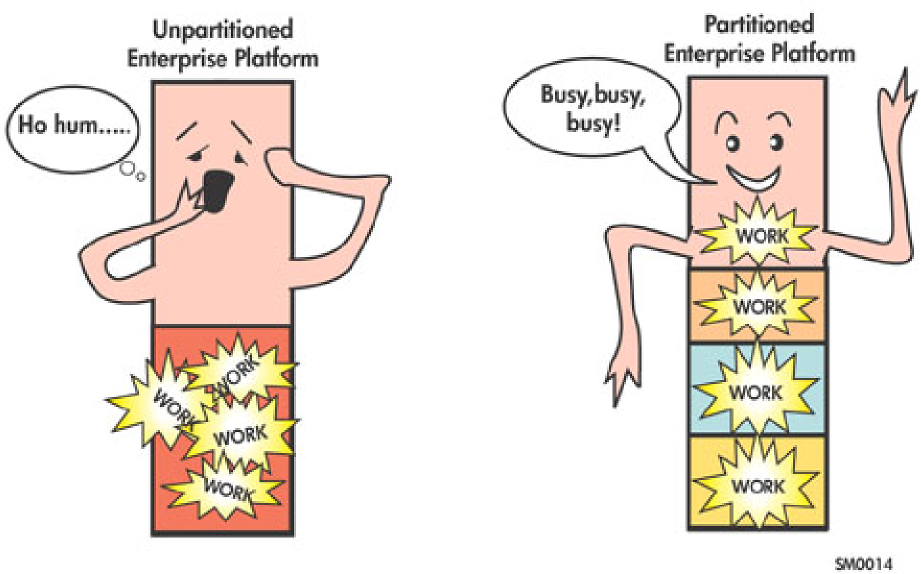
Why use partitions to divide a platform into smaller systems? Why not just buy smaller platforms?

There are some good reasons for employing a partitioned platform:

- Better resource utilization
- Safe environment for testing new software
- Save resources
- Ease of management
- Flexibility

1.2.1 Better Resource Utilization

Figure 1-2 Partitioning Can Affect Resource Utilization



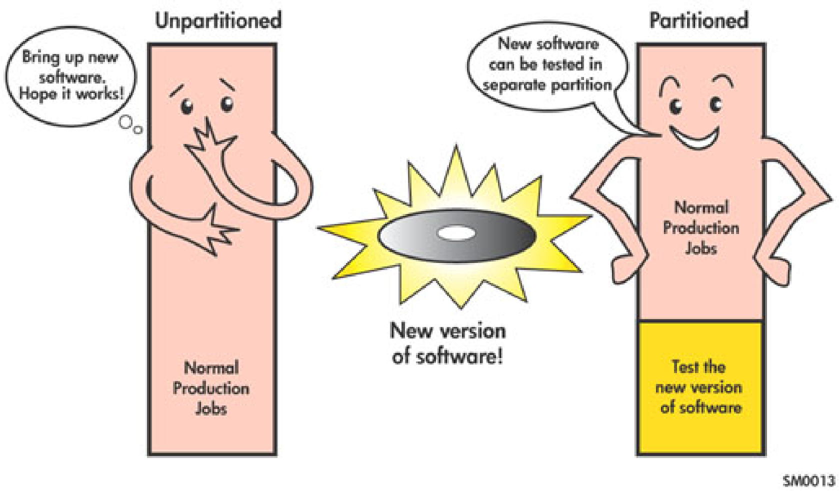
With very large enterprise platforms, full utilization can be hard to achieve. Although operating systems are designed to handle symmetric multiprocessing, throughput does not scale to the same degree as the increased amount of processing power.

Partitions can help bring about full utilization by separating the tasks at hand into chunks that are sized according to the requirements of the individual applications.

The unpartitioned enterprise platform on the left, above, is working at only a little more than half capacity. The same platform divided into, say, four partitions (on the right) use the platform at closer to full capacity.

1.2.2 Safe Environment for Testing New Software

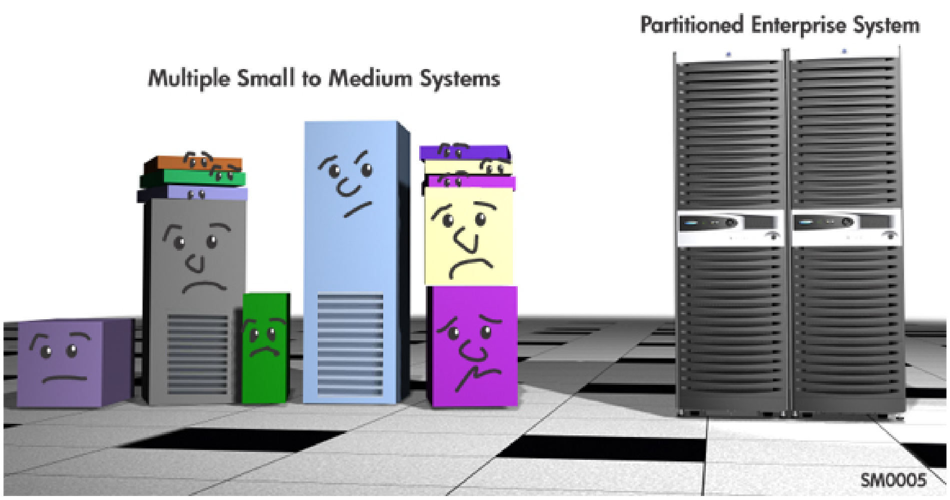
Figure 1-3   Partioning Provides a Safe Environment for Testing New Software



What to do when a new version of software -- operating system or application – comes along? With partitions, you can isolate the new software to a partition and run production jobs under the old software in a new partition until testing is complete.

1.2.3 Save Resources

Figure 1-4   Partitioning Can Be Used to Consolidate Resources



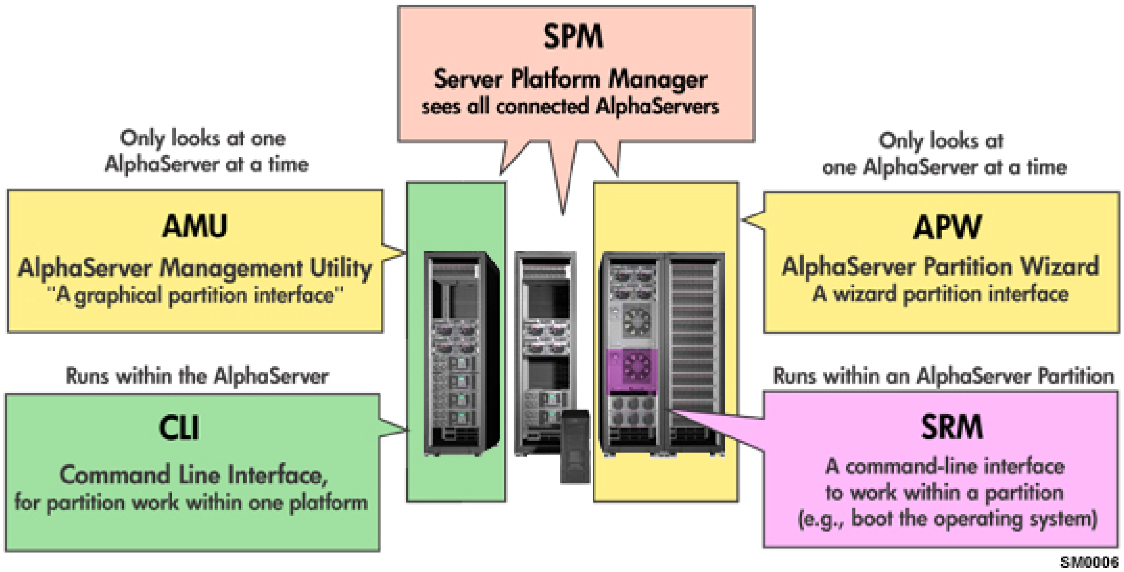
With partitions, multiple small and medium-sized platforms can be consolidated into one enterprise platform. Environmental aspects such as

- Floor space
- Special electrical requirements
- Special cooling systems
- Other physical requirements

can be concentrated in one place, reducing total cost of operation.

1.2.4 Ease of Management

Figure 1-5   AMS/SPM Provides Centralized Management



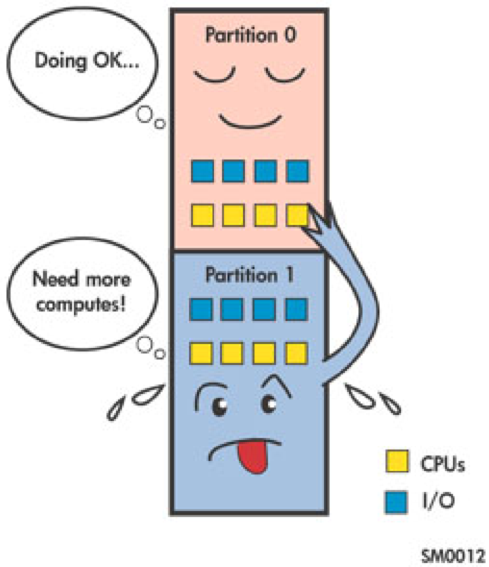
Different platforms, and their partitions, can be managed from a central observation point: the AlphaServer Management Station, or AMS. The Server Platform Manager (SPM) software on the AMS lets you examine more than one platform. The SPM, in turn, allows you to run, or telnet to, various partitioning tools:

- The AlphaServer Management Utility, or AMU (see the AlphaServer Management Station Software Installation and User's Guide.)
- The AlphaServer Partitioning Wizard, or APW (see the AlphaServer Management Station Software Installation and User's Guide.)
- The AlphaServer Management Command Line Interface, or CLI (see the CLI Reference.)
- The SRM command line interface, used for defining the operating system to be used within a partition, booting the operating system, and so forth. (see the SRM Console Reference.)

Note that you can establish a telnet connection to manage or partition a platform using the CLI without using the AlphaServer Management Station (AMS).

1.2.5 Flexibility

Figure 1-6 Partitioning Allows Resource Allocation Changes



The allocation of resources within partitions can be changed. The ease with which resources can be moved from one partition to another depends on the type of partition.

With hard partitions, the partitions must be brought to a quiescent state before resources can be moved. Once the operating system has been brought down in the relevant partitions, the system manager can use the AMU utility, the APW wizard, or the Server Management Command Line Interface (CLI) to redefine the boundaries of the hard partitions.

OpenVMS and Tru64 Unix support hard partitions.

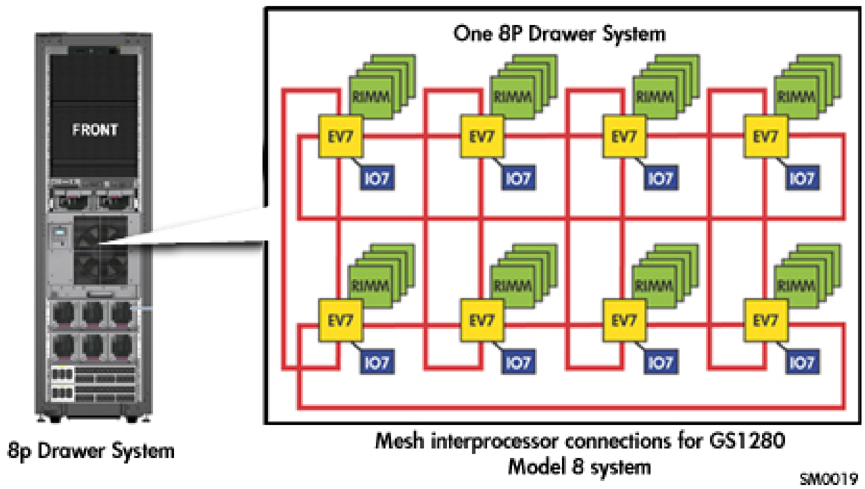
With soft partitions on OpenVMS Galaxy, for example, the Galaxy GCM (Galaxy Configuration Manager) can be used to move CPUs and I/O from one soft partition to another without shutting down the operating system, as illustrated in the figure above. OpenVMS Galaxy supports soft partitions.

1.3 Definitions

This section defines the terms mesh, torus, hard and soft partitions, and partition and subpartition.

1.3.1 The Machine and the Mesh

Figure 1-7 The Machine



The diagram above shows a sketch of a 8P platform, leaving out power, the LANs, and cabling. At this point, we just want to consider processors, memory, and I/O.

The 8P drawer holds processors and memory, and the smaller rectangles represent PCI drawers and storage.

The Mesh

You have probably heard about the hp AlphaServer ES47/GS80/GS1280 "mesh" architecture. We will examine this architecture in a two-dimensional drawing first. As you can see from the picture above, showing only an 8-processor platform, the processor chips are connected (red lines).

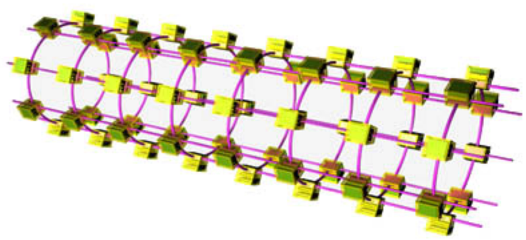
Each chip has an east, west, north, and south connection to other processors in the platform. The drawing shows the north-south connections between the top and bottom rows as a long curved line. A similar technique shows the east-west connections between the left and right columns.

One can think of the mesh as a torus, or donut shape. This is discussed in the next section.

1.3.2 The Torus

In designing valid partitions, it is important to keep in mind what is called "the torus" -- a way of looking at the mesh architecture in three dimensions.

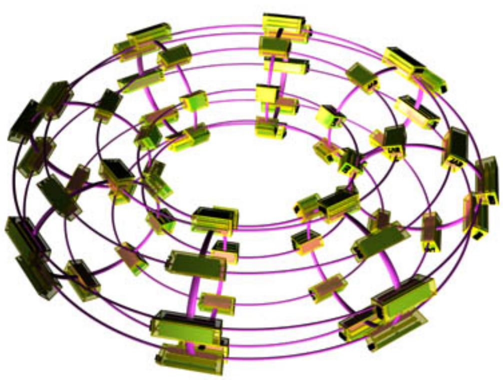
Figure 1-8 Start of the Torus



We have been examining the mesh in two dimensions. The picture above shows a representation of what would happen if this two-dimensional view were "rolled up" so that the east-west connections were all of the same length. The mesh becomes a tube.

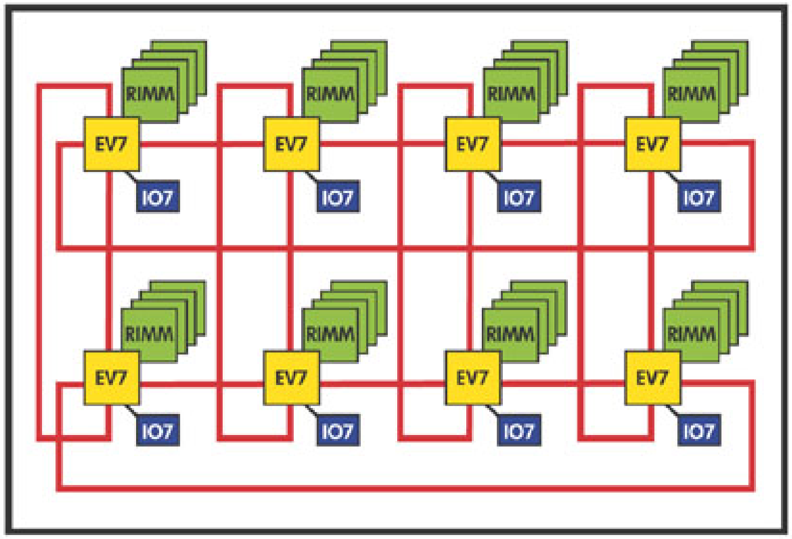
Now consider what would happen if we connected the open north-south connections. The mesh becomes a torus -- a doughnut-shaped structure.

Figure 1-9 The Torus Shape



1.3.3 Hard Partitions and Soft Partitions

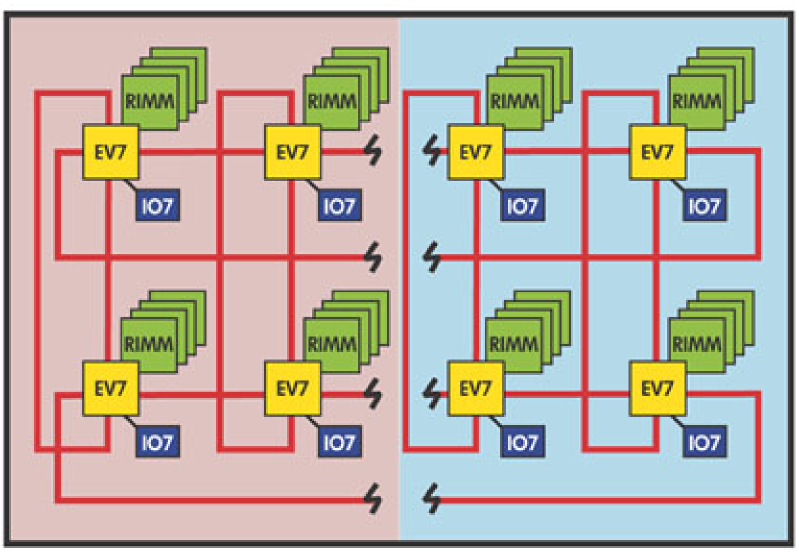
Figure 1-10 InterProcessor Connections on an 8P System



There are two types of partitions that can be defined on hp AlphaServer ES47/ES80/GS1280 platforms: Hard partitions and soft partitions. Consider the diagram of an 8P platform, shown above. This diagram shows the "mesh" -- the connections between processors -- in two dimensions. Note that each processor has attached memory and an I/O chip.

Hard Partitions

Figure 1-11 Hard Partition Interconnects

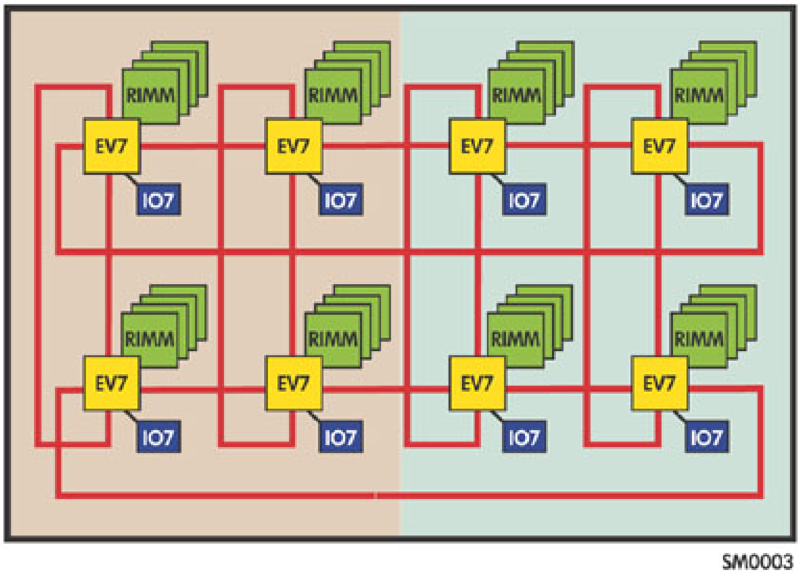


If you divided this platform into two equal 4P hard partitions, the IP interconnects would be turned off as shown in the diagram above. The memory and I/O resources attached to the processors in each hard partition belong to that partition only.

No communication can take place between the two hard partitions. In return, most faults propagated in one hard partition will not affect the operation of other hard partitions on the platform. The exception is when a power failure occurs, removing power from an 8P drawer. In this case, all hard partitions that have components in this drawer will be affected.

Soft Partitions

Figure 1-12 Soft Partition Interconnects



If you divided the platform into two equal 4P soft partitions, the IP interconnects are still in place. The firmware enforces the division in that you cannot assign the same resource as belonging to two different soft partitions. However, when you define soft partitions, you can define a quantity of memory, called community memory, which is to be shared between soft partitions.

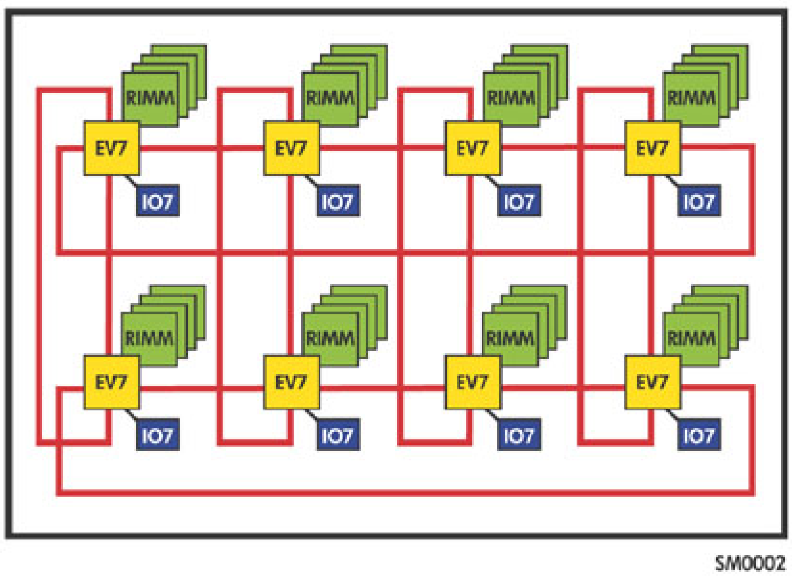
OpenVMS Galaxy software lets you move CPUs between soft partitions, as long as the soft partitions reside within the same hard partition, and as long as the CPUs do not have attached I/O.

And, OpenVMS Galaxy software allows the exchange of information between soft partitions to take place in community memory.

Unlike hard partitions, however, any faults occurring in one soft partition will be propagated across the mesh to other soft partitions within that hard partition.

1.3.4 Partitioning Tool Terminology: Partitions and Subpartitions

Figure 1-13 The "Default_HP" for an 8P System



All of the tools used for partitioning -- the AMU, the APW, and the CLI -- use the terms "partition" and "subpartition." Most simply, a partition is a hard partition. Soft partitions can be created only as subpartitions of hard partitions.

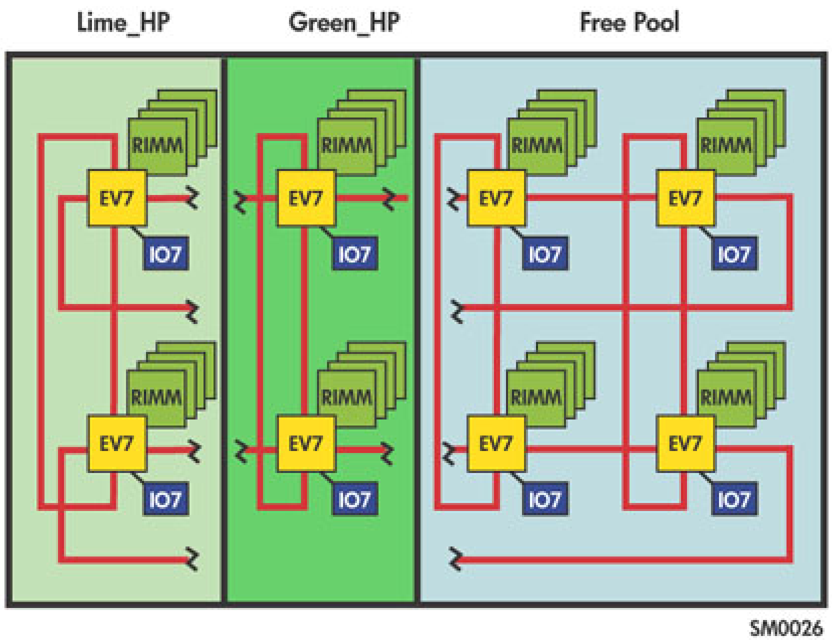
Partitions.

Initially, the resources in the platform are defined as being a hard partition called the default hard partition, named "Default_HP".

You can allocate resources in the Default_HP to two or more hard partitions, if desired.

You define a hard partition by identifying the processors that you want the partition to contain. All memory and I/O resources attached to those processors are also allocated to the hard partition.

Figure 1-14 Two Hard Partitions and the Free Pool



The picture above shows a simple example. Here, two hard partitions named Lime_HP and Green_HP have been created by the user. Each consists of two processors and their associated memory and I/O. The resources in the blue region are currently unassigned (see below). Note also that "Default_HP" has disappeared. It will reappear only if all user-created hard partitions are powered off and deleted.

Subpartitions

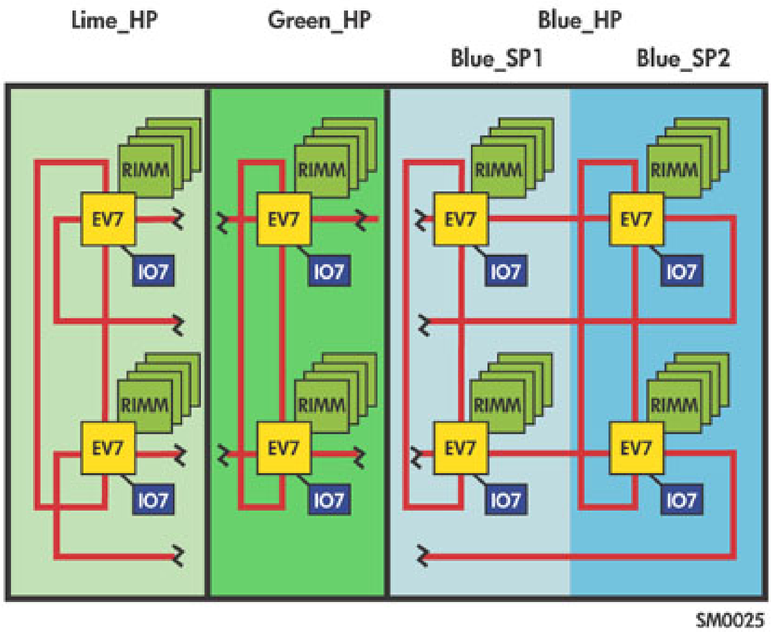
As noted above, the resources in a platform are initially defined as being a hard partition called the default hard partition, named "Default_HP". This hard partition is also contains one subpartition, the "Default_SP".

Within any user-created hard partition, you can create sub-partitions. (You cannot create subpartitions directly from the Default_HP. You must create a user-defined hard partition first.)

Conceptually, there are several types of subpartitions, but the only officially supported subpartition type is the soft partition.

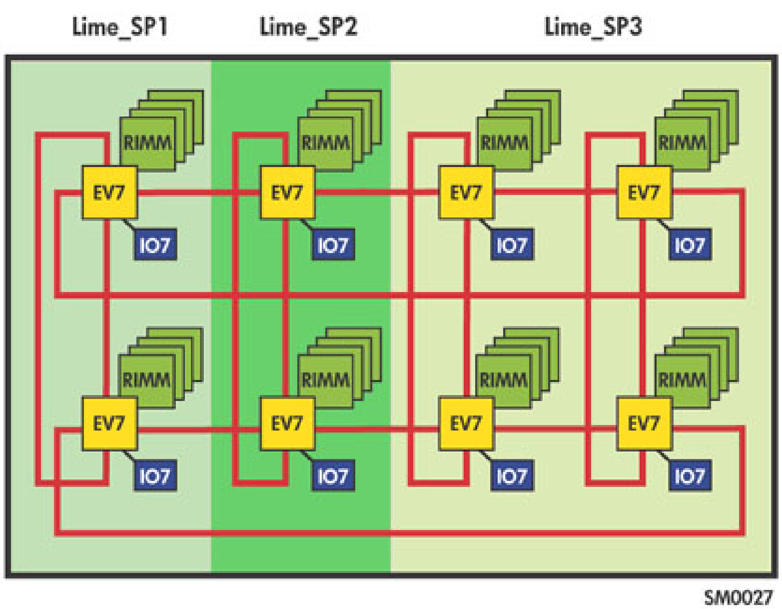
If you define subpartitions within a hard partition, any resources that you do not allocate to subpartitions or community memory are assigned to a special subpartition that is also called "Free_Pool".

Figure 1-15 Three Hard Partitions, One Divided into Two Soft Partitions



The picture above shows a simple example of subpartitions within a hard partition. Here, the 8P platform has been divided into three hard partitions named Lime_HP, Green_HP, and Blue_HP. The hard partition Blue_HP has been divided into two subpartitions called Blue_SP1 and Blue_SP2.

Figure 1-16 Three Soft Partitions within a User-Created Hard Partition



Note also that you can create only soft subpartitions on the platform, if you desire. All subpartitions must reside in a user-created hard partition, so you would first create and name a hard partition consisting of all the platform's resources. You would then define subpartitions within that hard partition. The picture above shows a simple example. Here, a user-defined hard partition named HP1 has been divided into three subpartitions, called Lime_SP1, Lime_SP2, and Lime_SP3.

Unassigned Resources

Resources which are not assigned to any hard partition are considered to be in a global Free Pool. You could use these resources to make a new hard partition or they could be added to an existing hard partition by powering that hard partition off, adding them, and powering the hard partition on.

Resources which are assigned to a hard partition, but not to any soft partition within that hard partition are in that hard partition's free pool. OpenVMS Galaxy can migrate resources out of a soft partition and into the hard partition's free pool. It can not migrate resources in or out of the global free pool (that is, in and out of the hard partition).

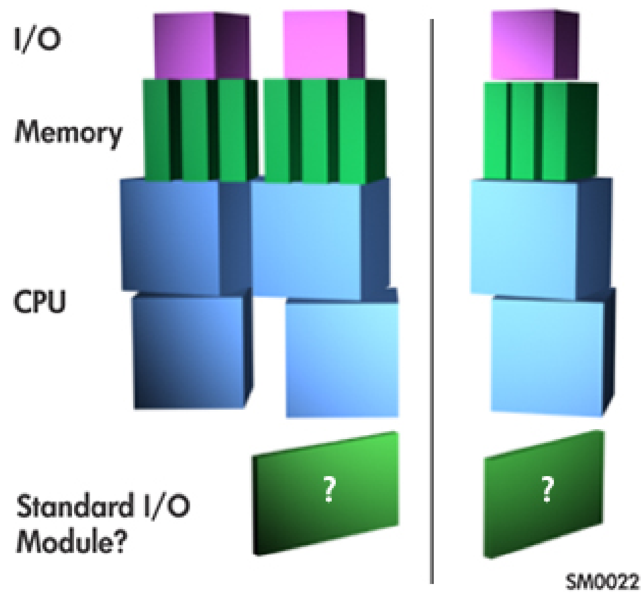
Both the global free pool and the free pool local to each hard partition exist even if they are empty.

1.4 Configuration Rules and Guidelines

This section defines the configuration rules and guidelines for creating hard and soft partitions on hp *AlphaServer* ES47/ES80/GS1280 platforms

1.4.1 Hardware Requirements

Figure 1-17 Hardware Requirements for Partitions



Partition Construction. Each partition, hard or soft, must have the following resources: CPU, memory, and at least one attached I/O subsystem,

Furthermore, the CPUs in a given partition must run at the same speed. As of this writing, there are two possible speeds for CPUs (see Appendix D) in a given platform. You cannot mix CPUs running at different speeds in the same partition.

Standard I/O Module. While not strictly a requirement for creating partitions and subpartitions, you may find that doing without a standard I/O module (and the hardware it allows) in each partition and subpartition creates difficulties. Specifically:

- It is impossible to update the firmware in an individual partition using the Loadable Firmware Update (LFU) utility without a standard I/O module with an attached CD-ROM drive included in the partition.
  Although you can boot LFU from the network (see the hp AlphaServer ES47/ES80/GS1280 Service document) there is no way to use the firmware update CD without a CD-ROM drive attached to that particular partition.
- It requires some forethought to install the operating system to a boot device that is accessible to the partition, before partitions are defined on the platform. This can be done from the system CD-ROM device; however, your flexibility in dealing with the operating system (upgrading, applying patches, and so forth is limited to situations in which the system CD-ROM device is accessible to all other storage devices in the system.
- You may want to attach a monitor, keyboard, and mouse directly to the partition. The monitor can be driven from an AGP or PCI graphics adapter. However, the keyboard and mouse must be connected to the two USB ports on the front of the standard I/O module.

1.4.2 Physical Adjacency Requirements

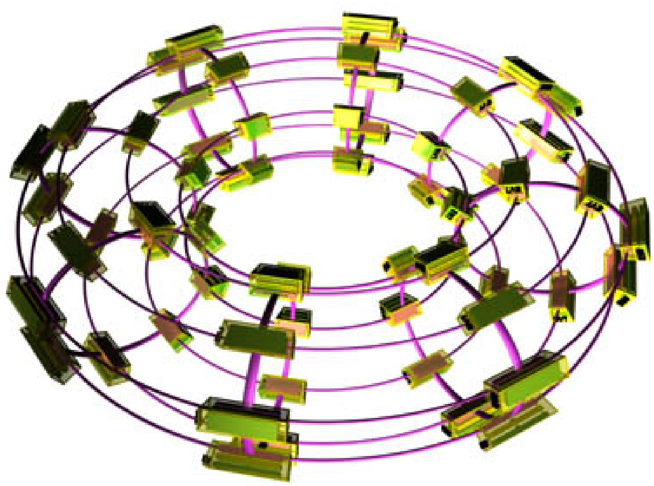
The requirements for physically adjacency are different for hard partitions and soft partitions.

Hard Partitions.

Because of the mesh architecture and its IP interconnects, all of the CPUs that you wish to allocate to a hard partition must be adjacent. In the two-dimensional way we have been looking at partitions, this would translate to the concept that "hard partitions should be rectangular in shape."

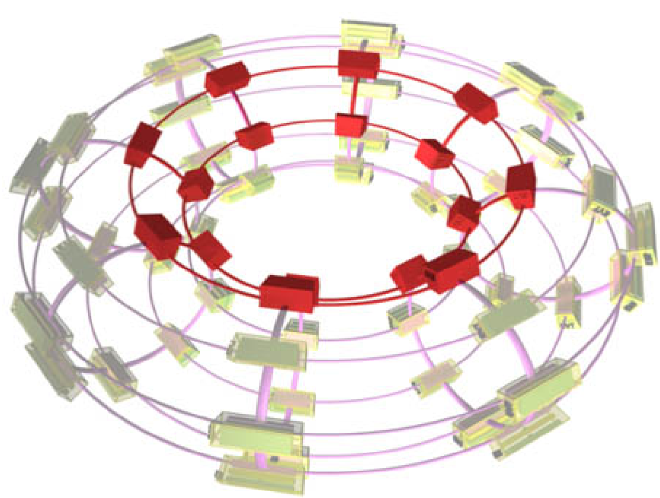
However, remember that we are really dealing with a three-dimensional construct: the torus, as shown below.

Figure 1-18 Physical Adjacency in Terms of the Torus



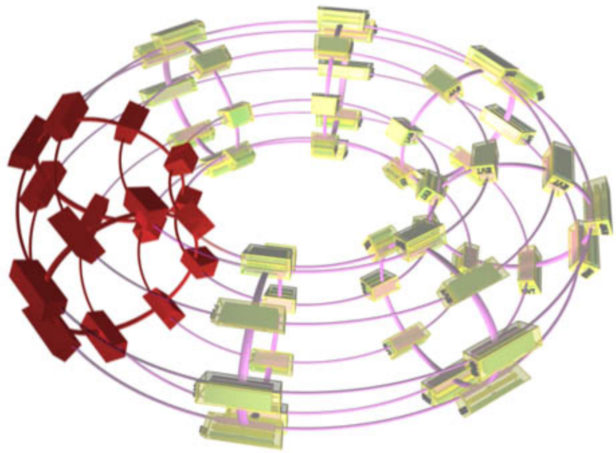
The picture below shows an example of physical adjacency that takes advantage of all possible east-west connections in a 16-processor partition of a 64-processor platform.

Figure 1-19 East-West Connections within the Torus



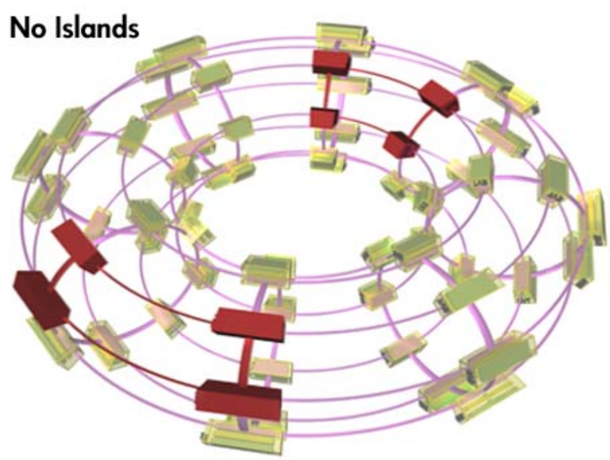
The picture below shows an example that takes advantage of all possible north-south connections in a 16-processor partition of a 64-processor platform.

Figure 1-20 North-South Connections within the Torus



Note that you do not have to "slice the donut" horizontally or vertically, as shown in the two previous pictures. This could be used to maximize the number of IP interconnects in a two-dimensional "rectangle". But, you cannot create "islands" and have them be part of the same hard partition. The picture below shows an example of two islands as separate chunks of the donut that are not connected by any IP connections.

Figure 1-21 Hard Partitions Cannot Be Constructed of “Islands” within the Torus



In general, non-rectangular shapes should be avoided for hard partitions. There are some exceptions to the "rectangular rule," but they are most useful in understanding what happens when a CPU slot is empty or contains a filler module. In this case, north/south/east/west IP connections are compromised, and can affect the routing of requests between modules. This is discussed further in the section in Appendix B called "Routing."

1.4.3 Memory Assignment

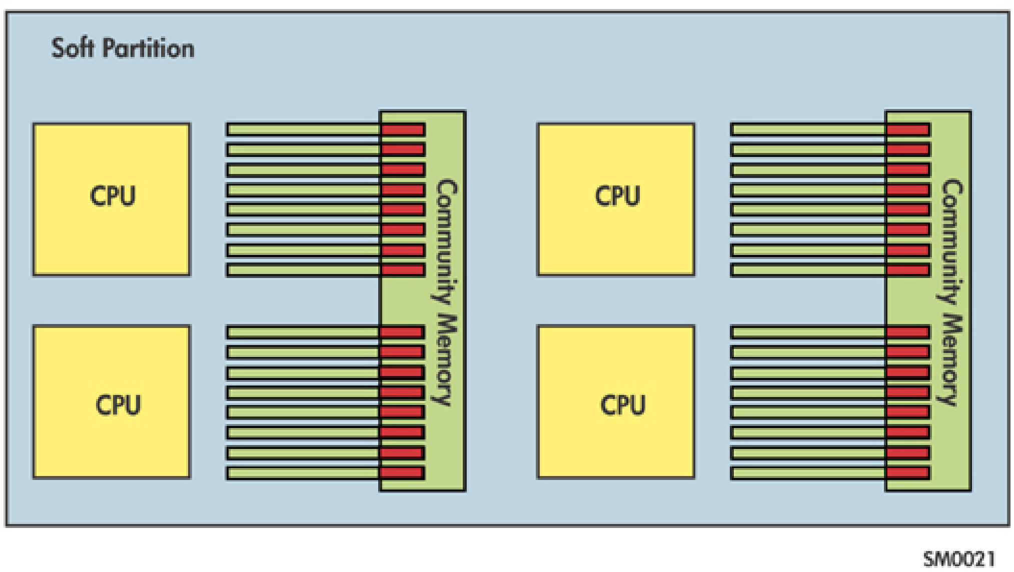
Hard Partitions. When you define a hard partition using any of the partitioning tools available, the memory residing on the dual-CPU module that is assigned to each CPU in the hard partition is automatically assigned to the hard partition.

Soft Partitions. The specifics of memory assignment to soft partitions, or subpartitions, depends upon the tool you use to define partitions.

With the AlphaServer Partitioning Wizard (APW), for example, you assign "duos" (dual-CPU modules) to subpartitions one by one. For each duo, the APW displays the amount of memory available on that duo, and then lets you specify an amount of memory to be used for the subpartition, and an amount to be used for community memory.

The Server Management CLI provides the assign memory command, and the AMU also lets you assign memory to soft partitions, but it is recommended that you let the firmware make memory allocation to subpartitions by default. It is best to only assign memory to define an amount of community memory. Community memory is memory shared between the subpartitions defined within a hard partition.

Figure 1-22 Concept of Community Memory in Soft Partitions



By default, the firmware assigns the memory associated with a given CPU on the dual CPU module to the subpartition in which the CPU resides. A CPU's memory cannot be split between subpartitions.

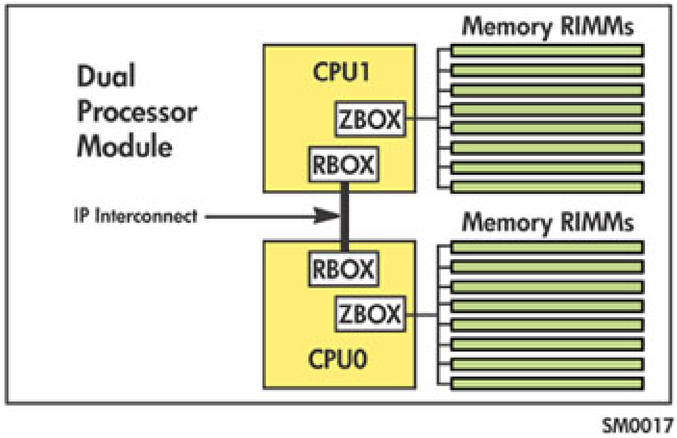
Community memory, if requested, is allocated by taking alternating 64-Mbyte chunks from the CPUs in each subpartition, starting at the highest memory locations.

If the total amount of memory requested for the subpartitions within a hard partition is less than the actual amount of memory in the hard partition, the left-over memory is assigned to the hard partition's "free pool". It is inaccessible. This is another reason for not assigning memory for anything other than community memory.

The firmware itself uses some memory on each CPU module; 16 Mbytes of memory is allocated from the lowest addresses of each CPU that has memory for use by PALcode and the SRM console.

1.4.4 Memory Striping

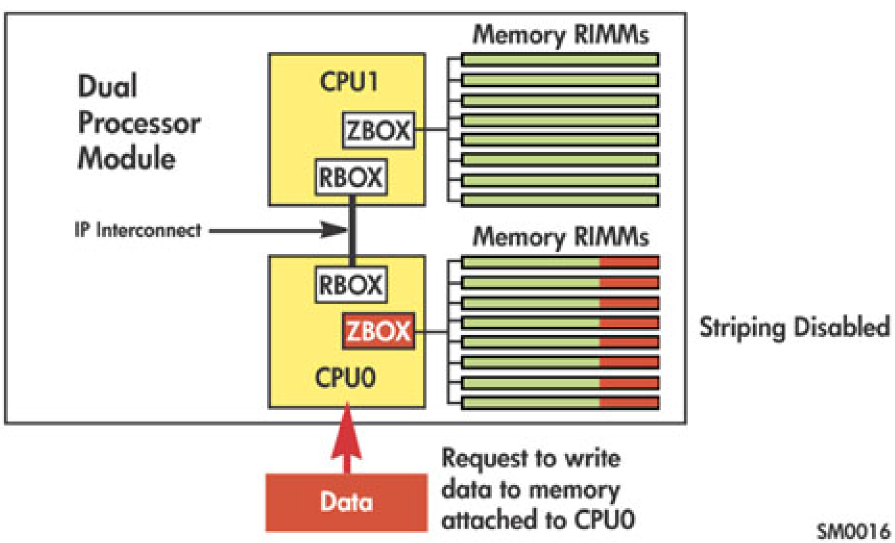
Figure 1-23 Memory in a Dual-Processor Module



Memory striping is a partition attribute that can be requested from any of the three user interfaces that let you work with partitions. Consider the picture above, which shows a very simple dual CPU module with two CPUs and attached memory. Note: to request striping for a partition, each module in a dual-processor module within the partition must have the same amount of memory assigned to it.

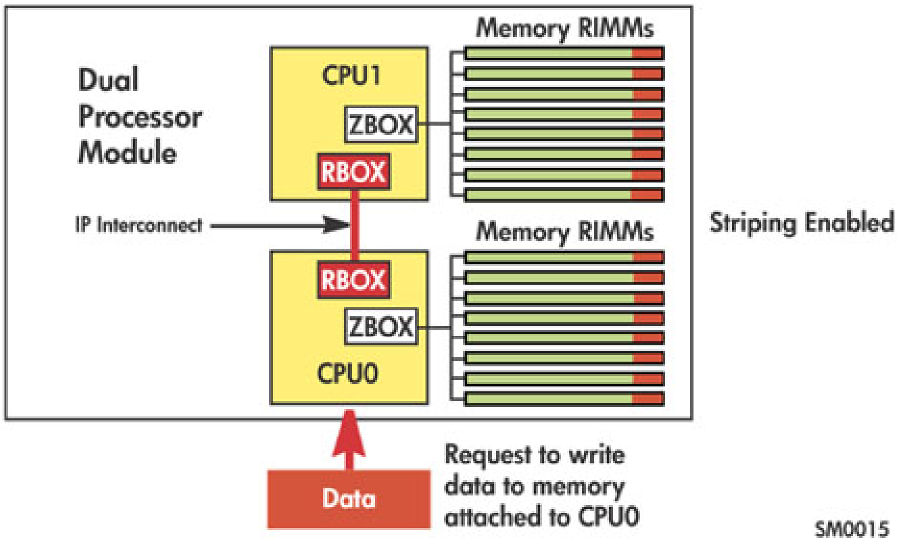
When memory striping is not enabled (the default), a request to write to memory on a CPU causes data to be written to the memory attached to that CPU. The picture below shows an example.

Figure 1-24 Memory Striping Disabled



When you request memory striping, memory is interleaved across the two CPUs in a dual-processor module, using alternating cache lines. The following picture shows an example. A memory write to CPU0 distributes the data to be written across memory attached to both CPU0 and CPU1.

Figure 1-25 Memory Striping Enabled



In most applications, memory striping degrades throughput. Memory striping may enhance throughput in some rare applications that have a lot of requests for memory in a certain area -- called "hot spots".

1.4.5 Fault Resiliency

The AlphaServer GS1280 platform was designed to minimize the conditions under which a failure in one hard partition can cause another hard partition to fail.

There are the two configuration guidelines that you can use to minimize this possibility, listed below. The reasons behind the guidelines are described in following paragraphs.

Create hard partitions on system building block (SBB) boundaries. A system building block on AlphaServer ES47 and ES80 platforms is a 2P drawer. A system building block on an AlphaServer GS1280 platform is an 8P drawer.

Do not split CPUs on a dual-processor module between hard partitions.

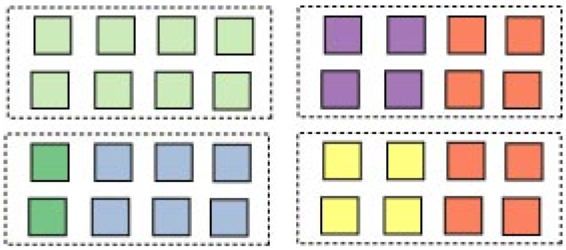
Splitting CPUs on a dual-processor module is not supported. The architecture will allow it, but the operating system will not work properly. Note that this is only true for hard partitions; soft partitions will work properly because the IP links are not turned off.

The reasons underlying these two guidelines are:

- It is unlikely, but it is possible that two of the AC power supplies in a power subrack can fail, or a rare power fault can occur, removing power from an 8P drawer. Any hard partition that contains resources within or attached to this drawer will fail.
- Splitting the two processors in a dual-processor module between two hard partitions entails the risk of a failure in the module, bringing down both hard partitions. The CPUs on a dual module share some hardware -- for example, the clock oscillator. If this hardware fails, it will cause both processors to fail, bringing down both hard partitions.

1.4.6 Fault Resiliency Example

Figure 1-26 Fault Resiliency Example



The picture above shows a 32P platform divided into six hard partitions, each shown as a different color. The dashed boxes represent 8P drawers, which are system building blocks (SBBs) on an AlphaServer GS1280 platform. (An SBB on an AlphaServer ES47 or ES80 platform is a 2P drawer.)

Assume that CPU modules form 2P columns within an 8P drawer. (This display is similar to what you would see with the AlphaServer Management Utility, or AMU.)

The lime green partition follows the "Partition Within SBB Boundaries" guideline. If a power failure occurs on the power subrack that supplies this 8P drawer, the fault is not propagated in the other hard partitions.

The darker green partition follows the "Don't Split a Duo" guideline. If a non-power related fault on this dual-CPU module occurs, the fault is not propagated in the other hard partitions.
The orange partition does not follow the "Partition Within SBB Boundaries" guideline. If a power subrack failure occurs that removes power from either of the two 8P drawers in which the orange partition resides, the entire orange hard partition will fail.

Specifically, if the power subrack to the 8P drawer containing the purple hard partition and part of the orange hard partition fails, both the purple and orange hard partitions will fail. (The yellow hard partition will continue to run.)

Likewise, if the power subrack to the 8P drawer containing the yellow hard partition and part of the orange hard partition fails, both the yellow and orange hard partitions will fail. (The purple partition will continue to function.)

Thus, the risk of a power subrack failure bringing down the orange hard partition is greater than the risk for either the yellow or the purple hard partitions.

1.5 Planning Your Platform's Partitions

Now that you understand the concepts of partitioning -- the torus, hard and soft partitions and their attributes, and configuration rules, you can begin planning for partitions.

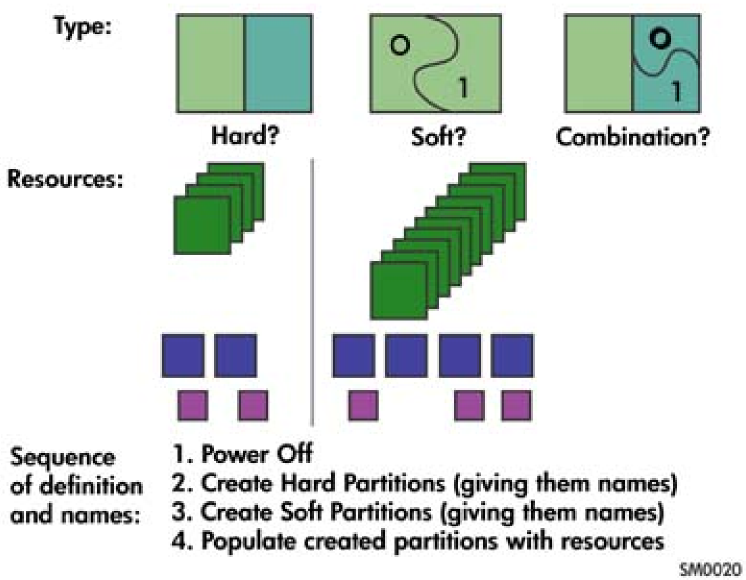
Ideally, partition planning will be done before the platforms are configured and purchased. Even so, needs change and hardware upgrades are available:

- 2P and 8P processor drawers as well as individual dual-processor modules
- Memory: adding RIMMs to dual-processor modules, or upgrading RIMM sizes
- I/O drawers, adapters, and storage

This section gives you some ideas and some forms you can print out and use to plan partitions on a platform.

1.5.1 Planning Considerations

Figure 1-27 Partition Planning Considerations



Here are some things you should think about and define before you start using any of the tools to create partitions.

Types of Partitions. You may want to use only hard partitions on your platform, only soft partitions, or a combination of hard and soft partitions. Remember that the factors are:

- Hard partitions restrict fault propagation across partitions, while soft partitions do not. OpenVMS, Tru64 UNIX, and LINUX support hard partitions.
- Soft partitions allow community memory and provide the capability to move CPU and I/O resources between soft partitions. OpenVMS Galaxy supports soft partitions.

Resources. You define the resources of a partition by selecting the processors in each partition and their location on the mesh.

Since each processor has attached memory and the possibility of an attached I/O, you should include these factors in configuring each hard partition. Soft (sub) partitions also allow the definition of community memory to be shared by all the soft partitions in the hard partition.

Sequence of Definition. When working with the partitioning tools, you must define a hard partition before defining its component soft (sub) partitions, if any. You may find it simpler to define all the hard partitions to be used on the platform first, and then define their component soft (sub) partitions (if any), one by one within each hard partition.

Even if you choose to use only soft partitions, you must first define and name a hard partition consisting of the entire platform's resources before you can define subpartitions (soft partitions) and community memory within the platform.

Names. Choose a consistent naming scheme for partitions within a platform. You may want to name them for the function of the primary application to be run in that partition. For example, "Payroll_HP" and "Records_HP".

Or, if partitions reflect the organizational structure of the user group, you can use names such as "Math_HP," "Biology_HP", "Psychology_HP" and so forth. Such hard partitions might be divided into subpartitions for a given subgroup. For example, "Math_HP" might consist of the subpartitions "Algebra_101", "Calculus 203", and so forth.

In short, try to choose partition and subpartition names that reflect whatever organizing principle you are using in creating the partitions.

1.5.2 Planning Forms

Figure 1-28 CPU and Memory Planning Form

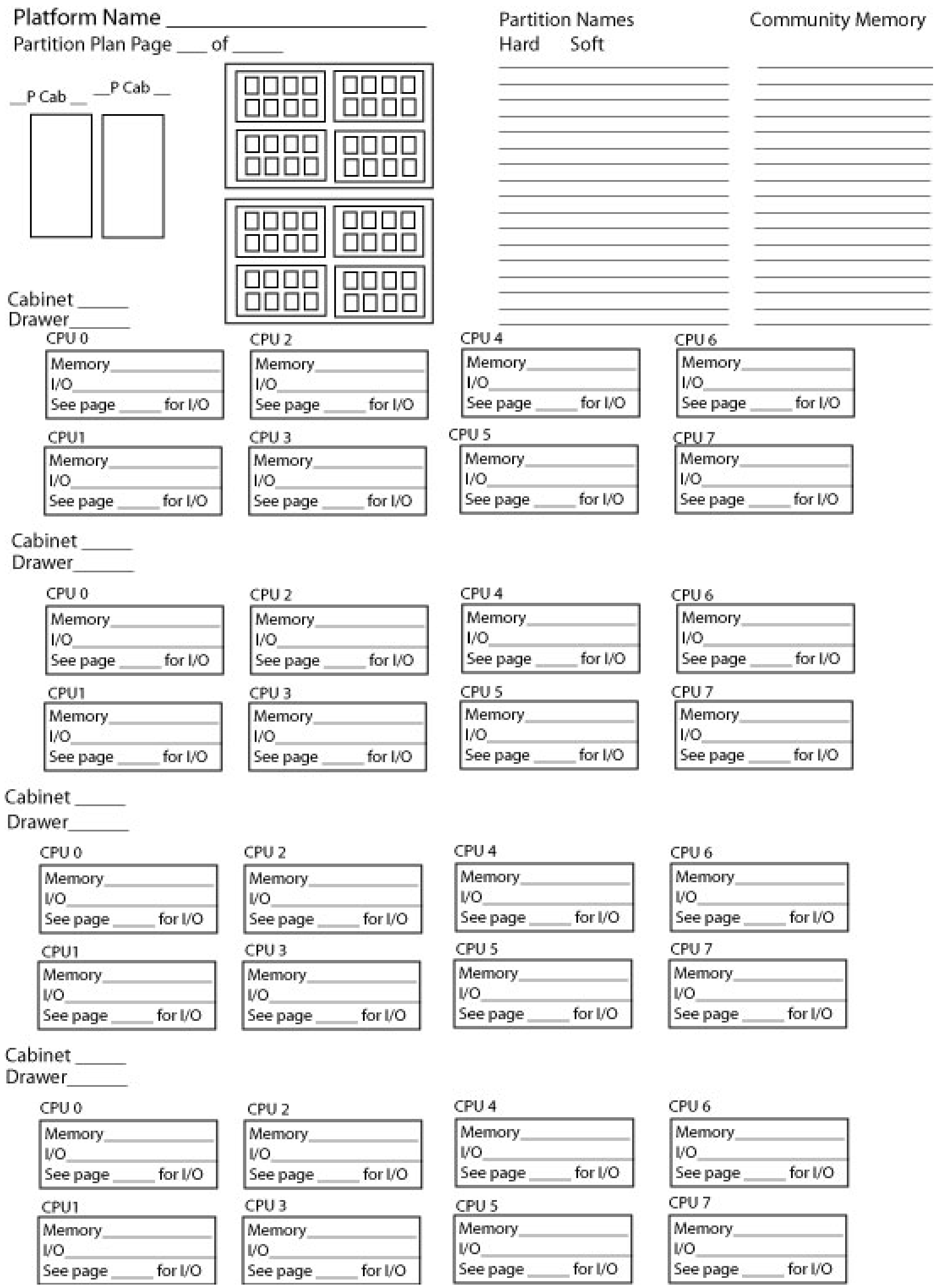


Figure 1-29 I/O Planning Form

Platform Name ____________________

Partition Plan Page ___ of _____ (I/O Drawer)

Cab __

Drawer No. ______

Master I/O Drawer with Disks ______

Internal Disks

____

____

Port S0

Standard I/O module: ☐ Yes ☐ No

Slot 0 ____________

Slot 1 ____________

Port S3

Accelerated Graphics Module (AGP): ☐ Yes ☐ No

Port S1 (133 MHz)

Slot 0 ____________ Attached Storage

Slot 1 ____________ Attached Storage

Port S2 (66 MHz)

Slot 0 ____________ Attached Storage

Port S2 (66 MHz) -- Continued

Slot 1 ____________ Attached Storage

Slot 2 ______ Attached Storage

Slot 3 ____________ Attached Storage

Slot 4 ____________ Attached Storage

Slot 5 ____________ Attached Storage

1.5.3 Planning Form Examples

Figure 1-30 CPU and Memory Planning Form Example

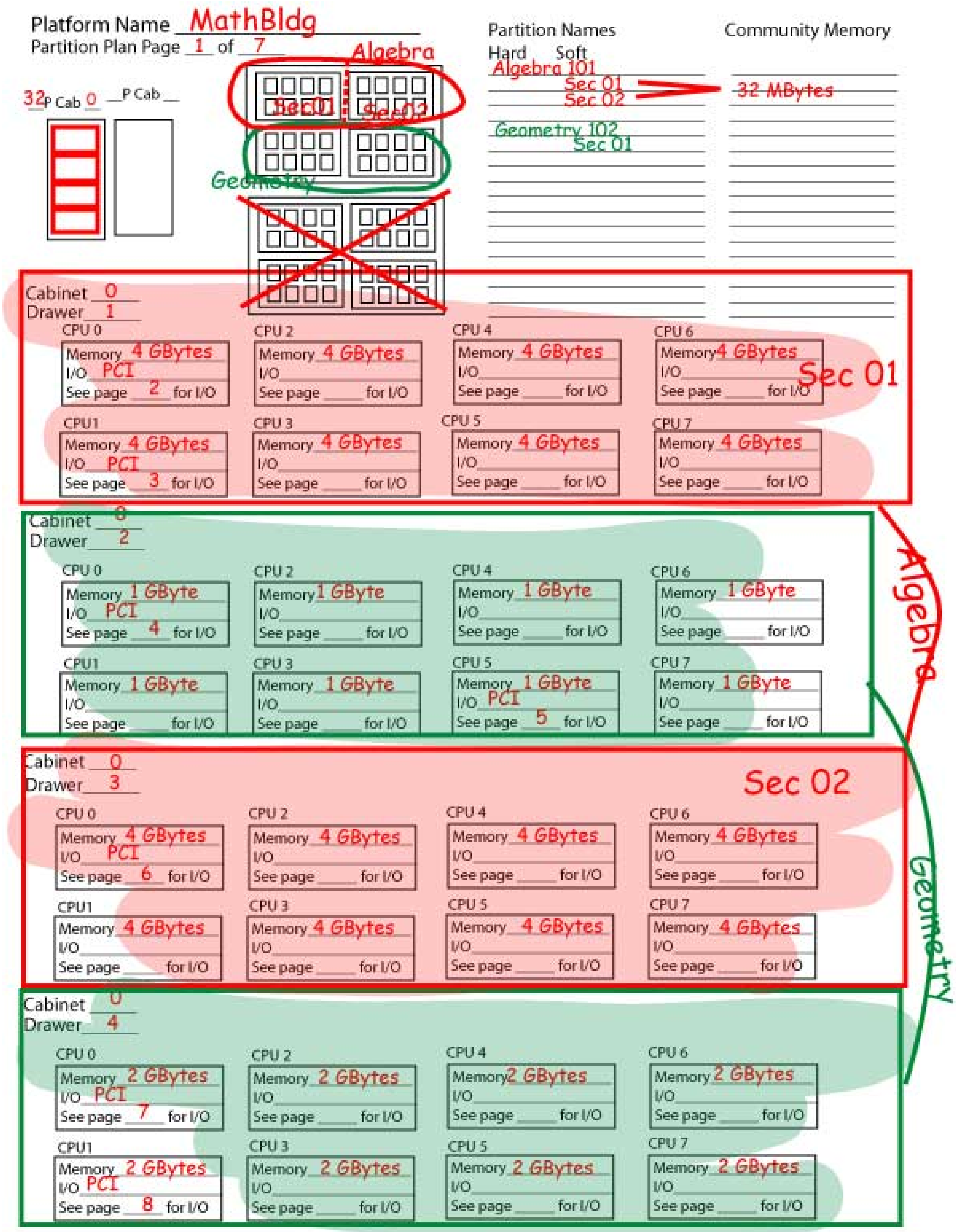


Figure 1-31 I/O Planning Form Example

Platform Name MathBldg
Partition Plan Page 2 of 8 (I/O Drawer)

Cab 1

Drawer No. 1
Master I/O Drawer with Disks Yes
Internal Disks
36.4 G,15K RPM 3R-A3849-AA

Port S0
Standard I/O module: ☒ Yes ☐ No
Slot 0
Slot 1

Port S3
Accelerated Graphics Module (AGP): ☐ Yes ☒ No

Port S1 (133 MHz)
Slot 0 — Attached Storage

Slot 1 — Attached Storage

Port S2 (66 MHz)
Slot 0 — Attached Storage

Port S2 (66 MHz) -- Continued
Slot 1 Ethernet 3X-DE602-BB — Attached Storage

Slot 2 SCSI 3X-KZPEA-DB — Attached Storage: DS-SL13R-BA Shelf with 14 72.8 GB disks (3R-A3839-AA)

Slot 3 — Attached Storage

Slot 4 SCSI 3X-KZPEA-DB — Attached Storage: DS-SL13R-BA Shelf with 14 72.8 GB disks (3R-A3839-AA)

Slot 5 — Attached Storage

1.5.4 What's Available?

If you are ordering a new platform, use the QuickSpecs to determine processor speed, memory RIMM sizes, I/O options. These specifications give detailed information for the ES47/ES80/GS1280 platforms and can be located from the main corporate site at:

http://www.hp.com

Using the Search on that page for "AlphaServer QuickSpecs", look for the entry: "HP AlphaSystems - what's new"." in the search results.

1.5.5 What's There Already?

If you are working with an existing platform, and don't know the details of the platform's CPU, memory, and I/O configuration, the various tools will display them for you. The key word for each tool is:

- AMU -- Select "Properties" from menus at various levels.
- APW -- Select the "Resources" button on the first wizard screen.
- CLI -- Use the show commands. For example, **show partition** shows the current partition setup, memory and I/O.

Figure 1-32 AMU Properties Display

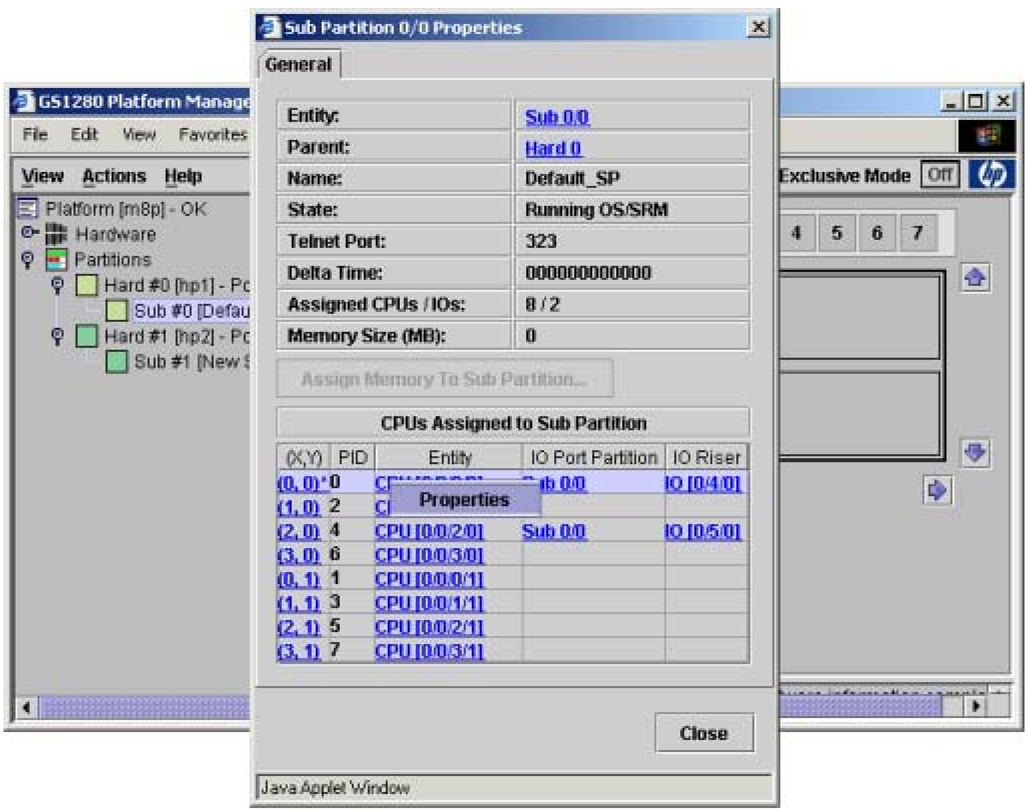
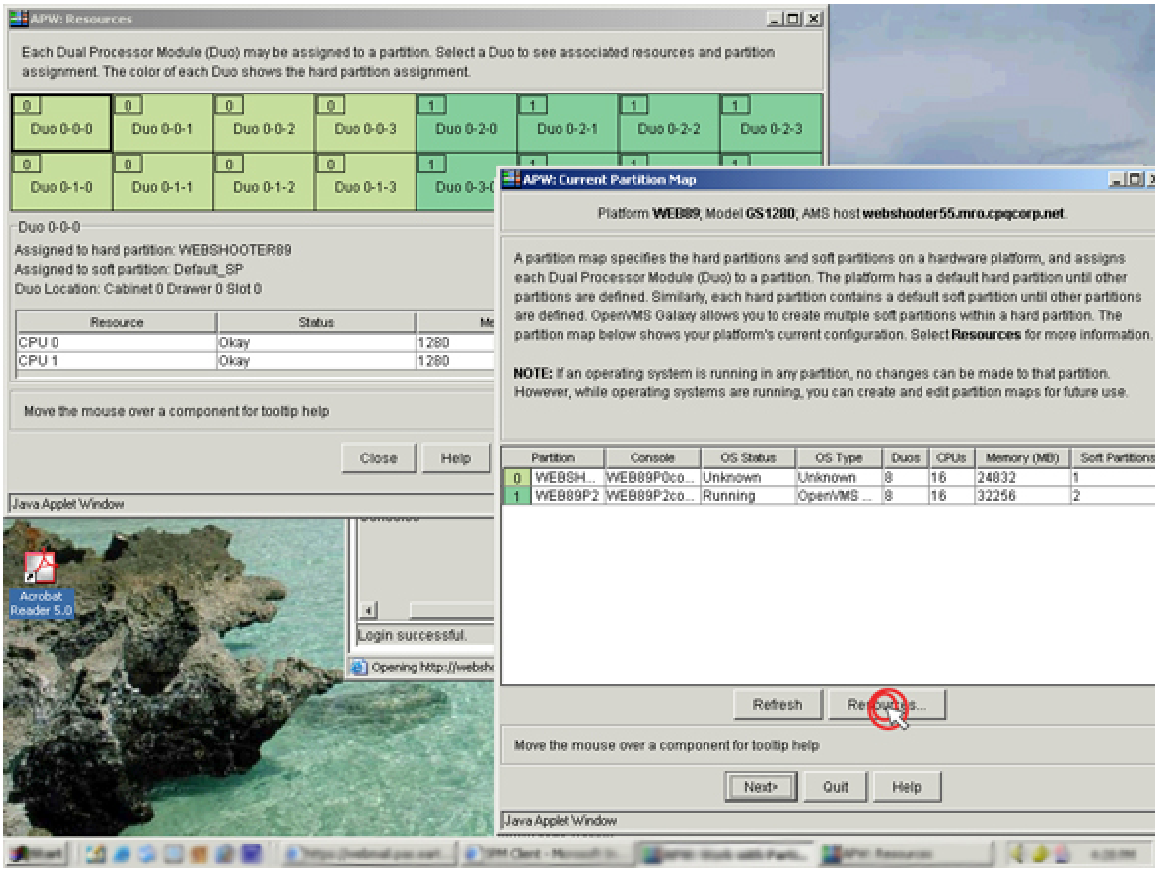


Figure 1-33 APW Resources Display



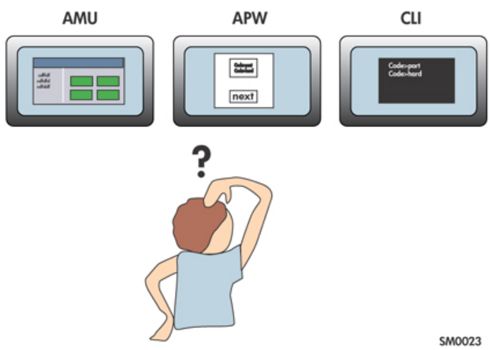
Example 1-1 CLI Show Partition Example

```
MBM> sho part
Hard Partition : HP Name = Default_HP, HP No.= 0, SP count = 2
Attributes     : max CPUs = 255, SP type = soft, Non-stripe
Physical Memory: 12288MB (12.000GB)
        Community Memory: 0MB (0.000GB)
        Sub Partition: HP Name = Default_HP, HP No.= 0
                       SP Name = Free_Pool, SP No.= 255
                       State = Not Running
                Free Memory: 0MB (0.000GB)
                CPUs: None
                IOPs: None
        Sub Partition: HP Name = Default_HP, HP No.= 0
                       SP Name = Default_SP, SP No.= 0
                       State = Not Running, Telnet port = 323
                Assigned Memory: unspecified
                CPUs:
                    Cab  Drw  CPU   (NS,EW)   PID    Type
                     0    0    0    ( 0,0 )    0     Primary
                     0    0    2    ( 0,1 )    2     Non-primary
                     0    0    4    ( 0,2 )    4     Filler
                     0    0    6    ( 0,3 )    6     Filler
                     0    0    1    ( 1,0 )    1     Non-primary
                     0    0    3    ( 1,1 )    3     Non-primary
                     0    0    5    ( 1,2 )    5     Filler
```

```
     0    0    7    ( 1,3 )     7    Filler
IOPs:
           SBB                             PCI Drawer
    Cab  Drw  IOP   (NS,EW)           Cab  Drw  IOR
     0    0    0    ( 0,0 ) ---------  0    1    0
     0    0    2    ( 0,1 )
     0    0    4    ( 0,2 )
     0    0    6    ( 0,3 )
     0    0    1    ( 1,0 )
     0    0    3    ( 1,1 )
     0    0    5    ( 1,2 )
     0    0    7    ( 1,3 )
```

Chapter 2
Overview of Partitioning Tools

2.1 What's the Difference Between the Tools?



We have stated that the AMU provides a graphic interface for partitioning; the APW, a wizard interface; and the Server Management CLI, a command line interface.

The topics in this chapter should clarify these concepts. Bear in mind these concepts as you study the tools:

AMU -- AlphaServer Management Utility

- **Pros:** Provides the most detailed view of the torus, showing of each processor and associated I/O, with displays of memory available under "Properties".
- **Cons:** You are working with the "live" platform as you partition.
- **Other Factors:** The AMU is not just for partitioning. It displays detailed hardware status, and lets you power down parts of the platform for maintenance.

APW -- AlphaServer Partitioning Wizard

- **Pros:** You can construct partition maps without affecting the "live" platform. These maps can be saved, letting you create multiple ways of partitioning the platform that can be implemented at any time.
- **Cons:** The APW is only for partitioning. You can view resources on the platform, but cannot power the platform or an individual partition on or off, as you can with the other interfaces.
- **Other Factors:** The APW insists that you follow the partitioning configuration rules. For example, it will not let you define a partition without I/O resources.

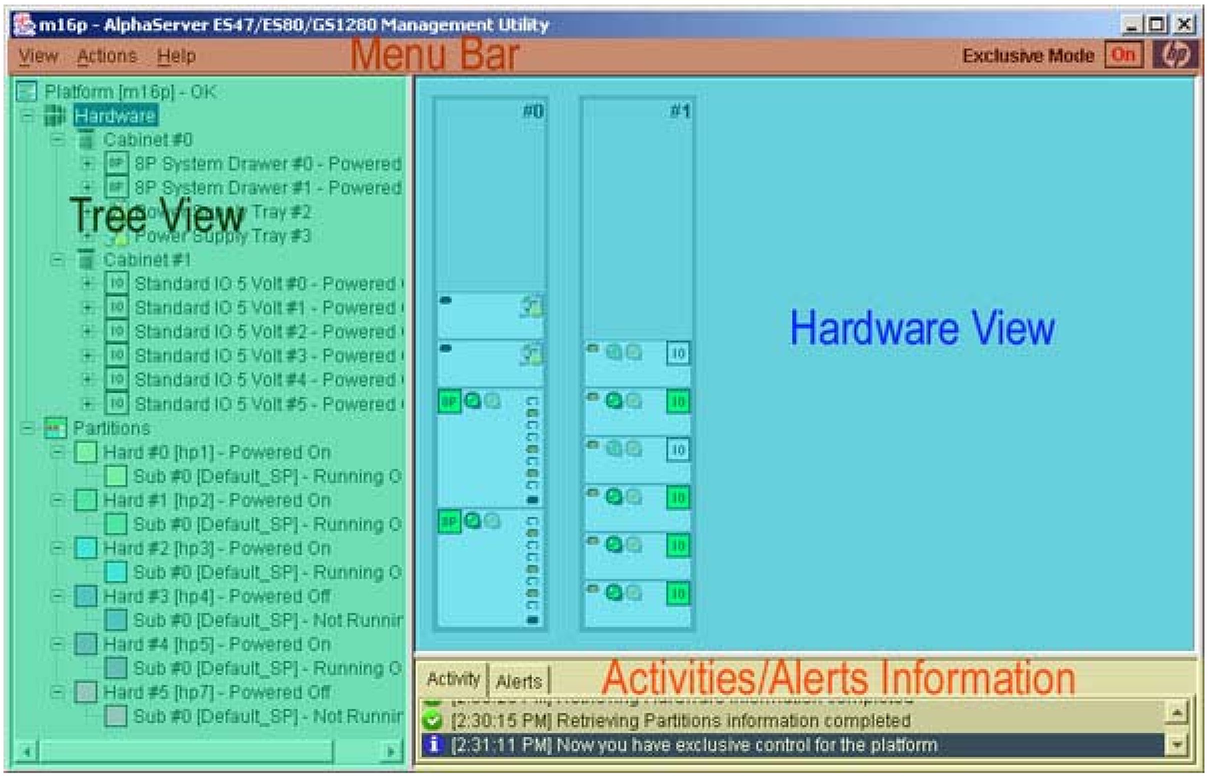
Server Management Command Line Interface (CLI)

- **Pros:** No graphic interface.
- **Cons:** No graphic interface.
- **Other Factors:** Partitioning is just one of the tasks you can perform with the CLI. If you tend to work with the CLI to do other platform management tasks, you may be more comfortable using the CLI to do partitioning, too. All of the CLI commands are described in the *CLI Reference Manual*.

2.2 Overview of the AMU

2.2.1 AMU Startup Screen

Figure 2-1 AMU Startup Screen Regions



As shown in the picture on the facing page, the AMU display consists of a screen with four regions: a menu bar and three panes displaying information.

You will probably work the most with the tree view (on the left) and the logical view of partitions. The logical view appears in the pane showing the hardware view when the AMU first starts. Continue to the next section to see how to manipulate the tree view to obtain the partition view that you will work with in creating partitions.

The two tabs at the bottom show information: the Activities tab shows the actions performed during the current session of the AMU. The Alerts tab shows any alert conditions that have occurred during the current session.

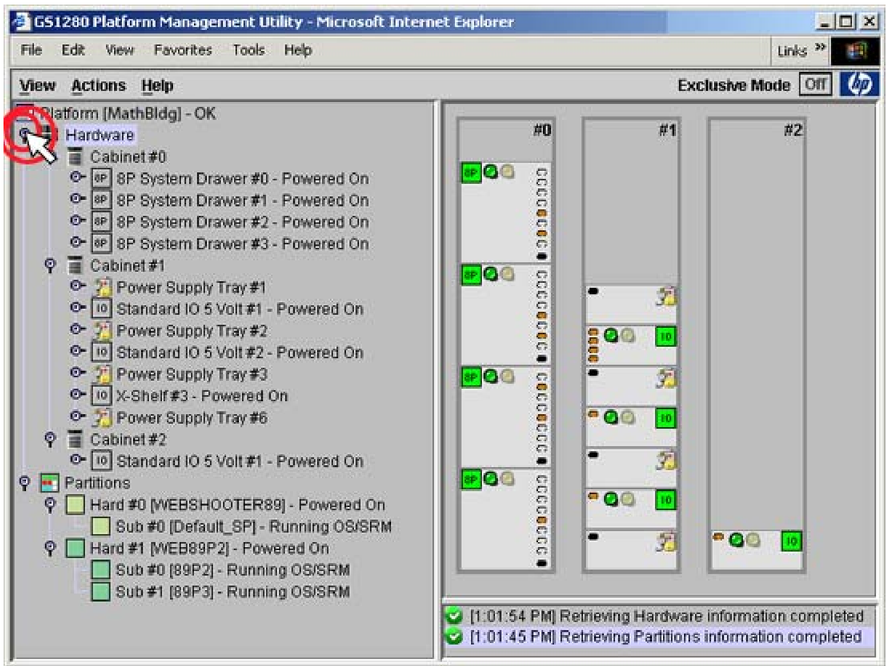
(Note: The alert information is provided in a recent release of the AMU. See the Server Management Tutorial for a more complete discussion of alert conditions. This tab is not shown in further screen samples in this document.)

2.2.2 Getting Started with the Screen

In our descriptions, we have collapsed the hardware tree to leave more room for partition information.

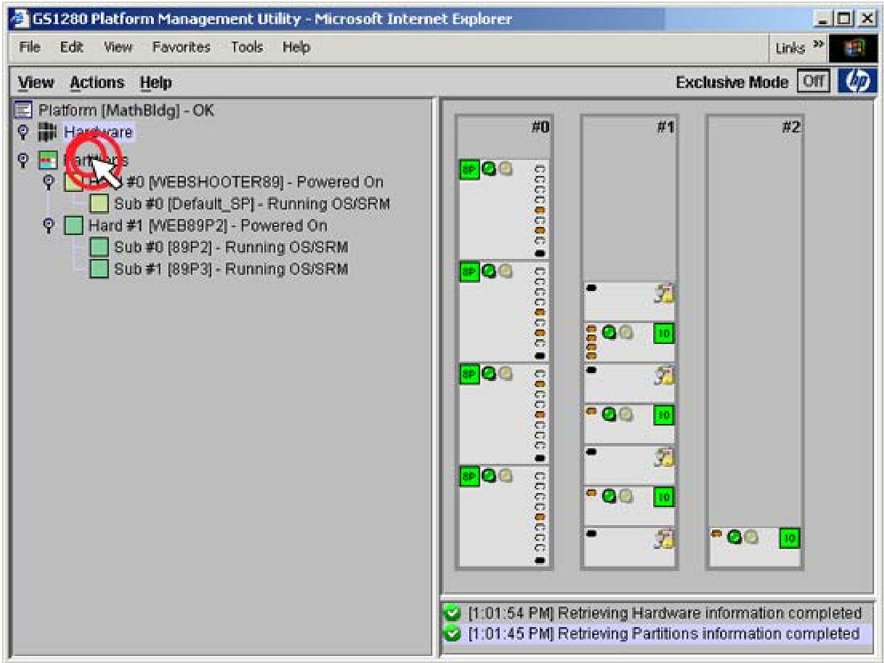
To do this, left-click on the "key" symbol in the Hardware line in the tree view to collapse the hardware tree display.

Figure 2-2 Collapsing the Hardware Tree



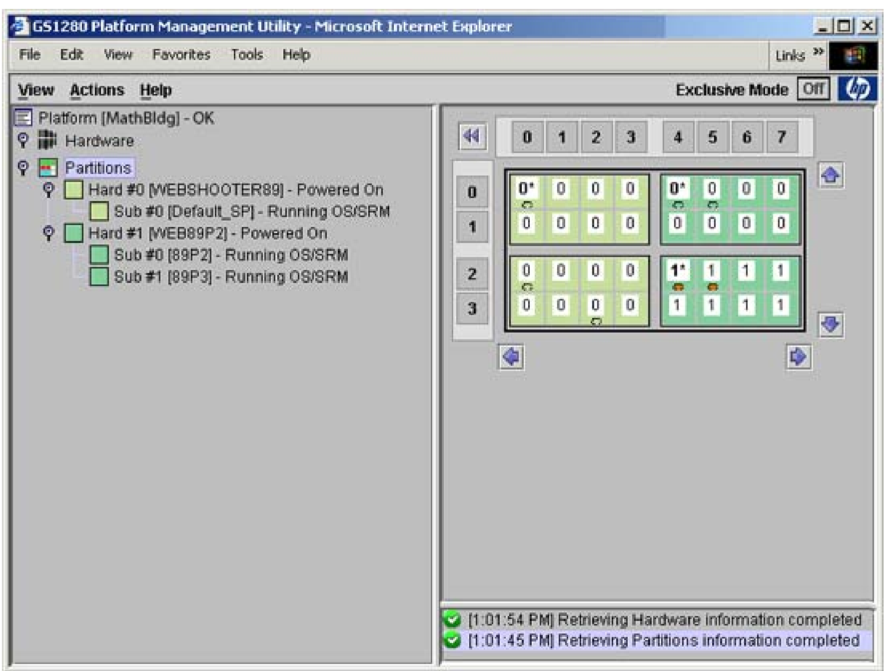
The result is shown below. Now, to see the logical, or partition view, left-click on the "Partitions" line in the tree view.

Figure 2-3 Requesting the Logical (Partition) View



The result is shown below, with the partition tree view in the left pane, and the partition, or logical view, in the right pane.

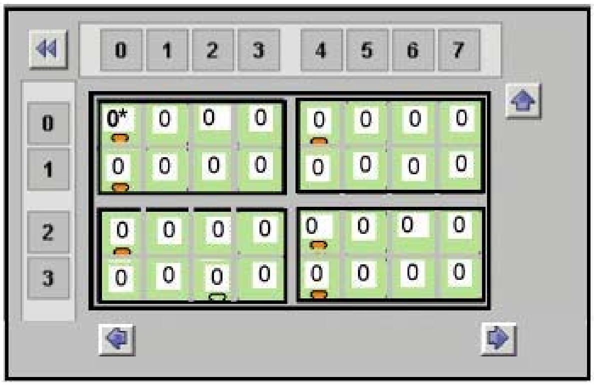
Figure 2-4 The Logical View



2.2.3 Viewing the Torus in the AMU

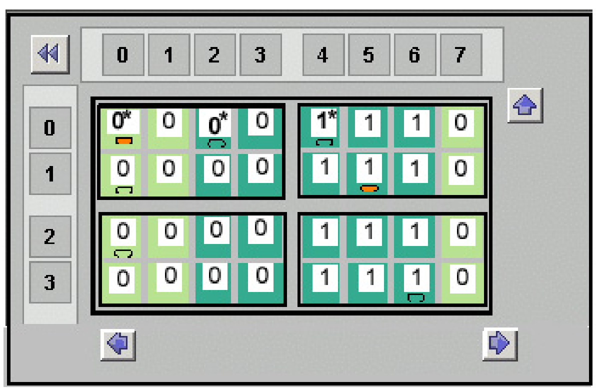
The AMU provides a logical view of the building blocks of the system that conveys the torus construct of the system in two dimensions. The diagram below shows the logical view of an unpartitioned 32-processor system.

Figure 2-5 Logical View of an Unpartitioned 32P System



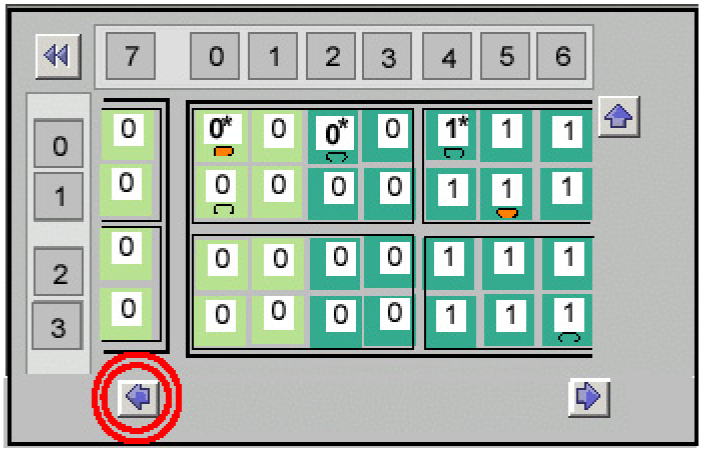
In the following picture, this 32-processor system has been divided into two hard partitions, shown by yellow-green boxes and green boxes. (Hard partitions are shown by the different colors of the boxes in the logical view of the AMU.)

Figure 2-6 Logical View of a Partitioned 32P System



The yellow-green partition is all one hard partition, as indicated by all the boxes being numbered 0. The green partition has been divided into two subpartitions, shown by the different numbers within the boxes, 0 and 1.It might seem at first that the yellow-green hard partition violates the "rectangular rule" of hard partitions: contiguous north-south and east-west connections. But, by using the left arrow of the logical view (Figure 2-7), we see that the east-west connections of the yellow-green partition have been maintained. The arrows rotate the "torus" so that you see the connections coming in the direction indicated.

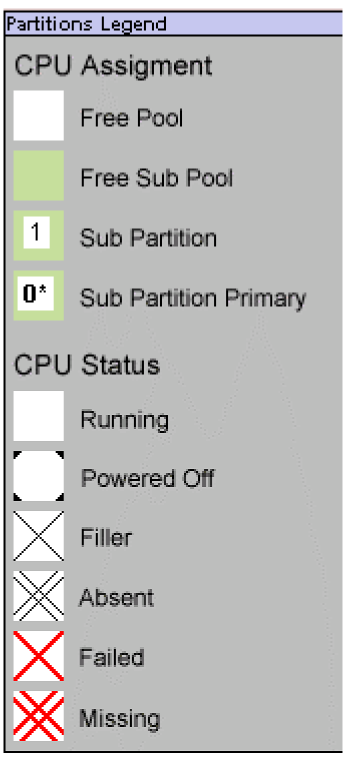
Figure 2-7 Rotating the Partition View



2.2.4 Legend for the Logical View

Each small square in the logical view represents a CPU. In the AMU display shown in the previous topic, each CPU in each slot in each drawer in the platform was occupied and running. The legend in Figure 2-8 shows the manner in which this and other states are shown.

Figure 2-8 Legend for the Logical View



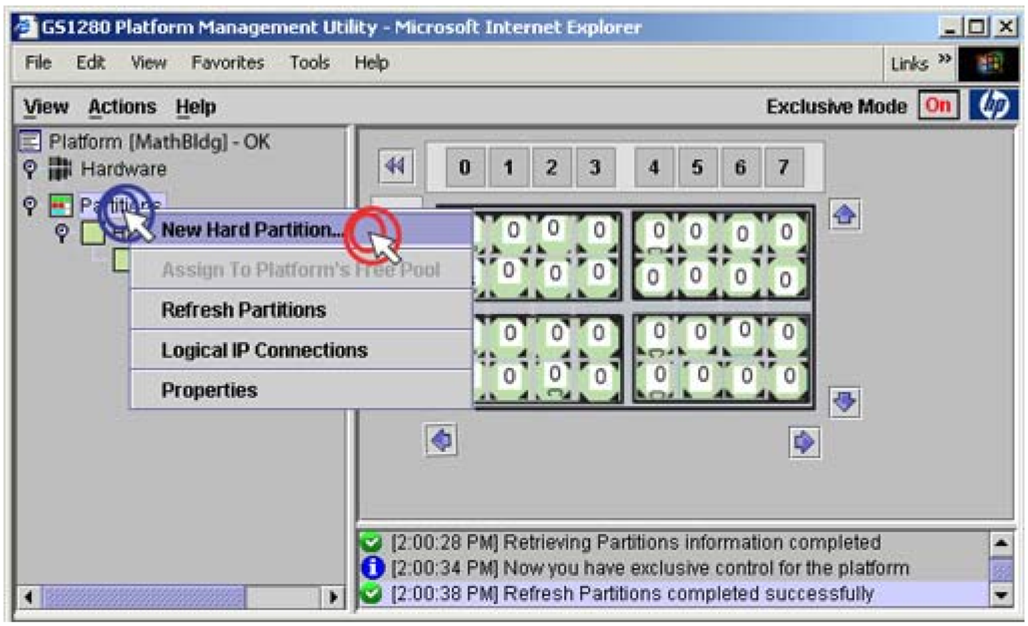
If a CPU has an attached I/O module, it is shown as follows on the square:

2.2.5 The General Process

You can work with the AMU in a variety of ways, but the most straightforward is to use your plan, and work back and forth between the tree and logical views. The general steps are these:

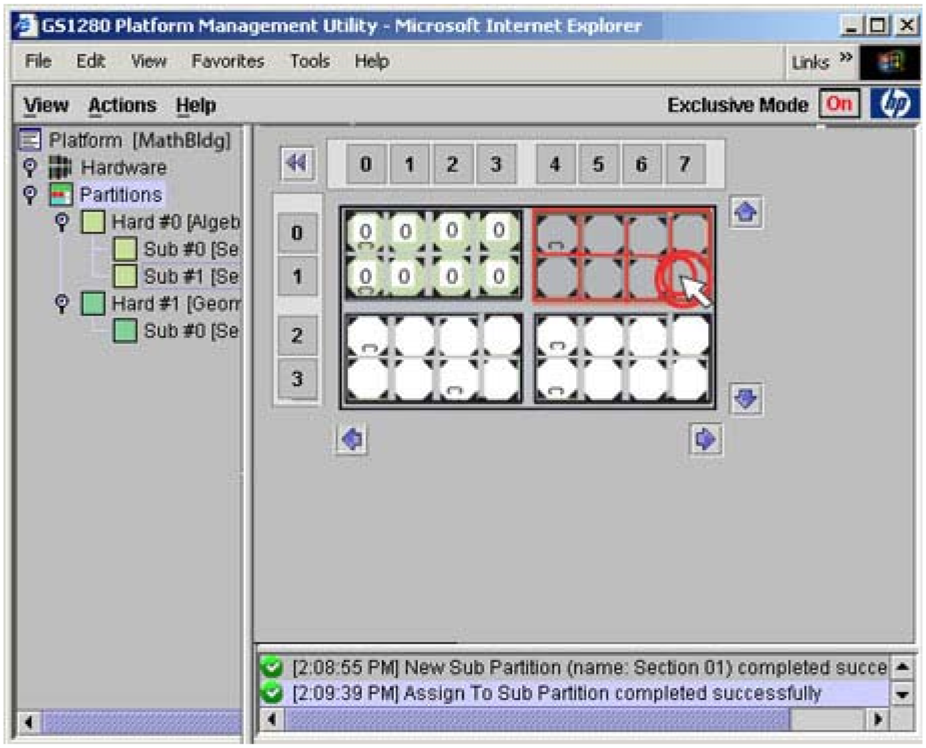
1. Work in the tree view to define the names of hard and soft partitions. You use the mouse to select the level under which you wish to create partitions, use a pull down menu to select the "Create" option, and type to fill out the names of the partitions from popup forms.
2. Choose the resources for a given partition by using the mouse to select them from the logical view.
3. Again at the tree view, use the mouse and pulldown menus to assign the resources you have selected to a given partition.

1 **Figure 2-9 Pull Down to Create New Hard Partition**



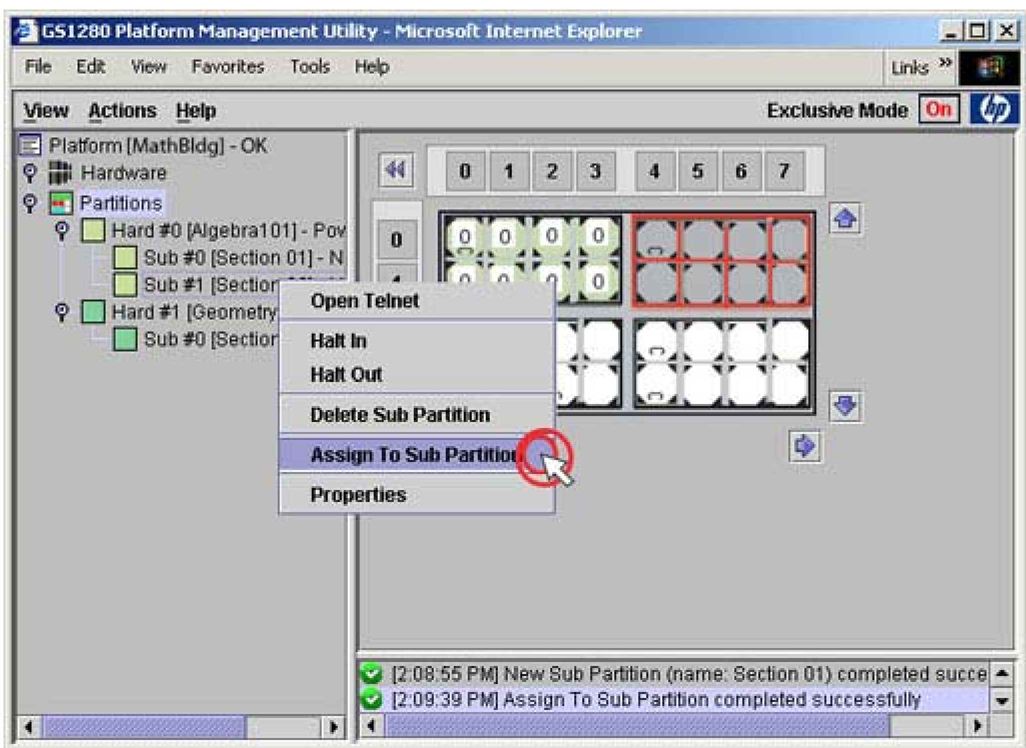
2

Figure 2-10 Select Resources



3

Figure 2-11 Assign Resources



2.3 Overview of the APW

2.3.1 APW Startup Screen

As shown in Figure 2-12, the first APW screen shows general information about APW maps at the top of the screen. One item of information is of special interest:

You can create and save partition "maps" that you can apply now or later.

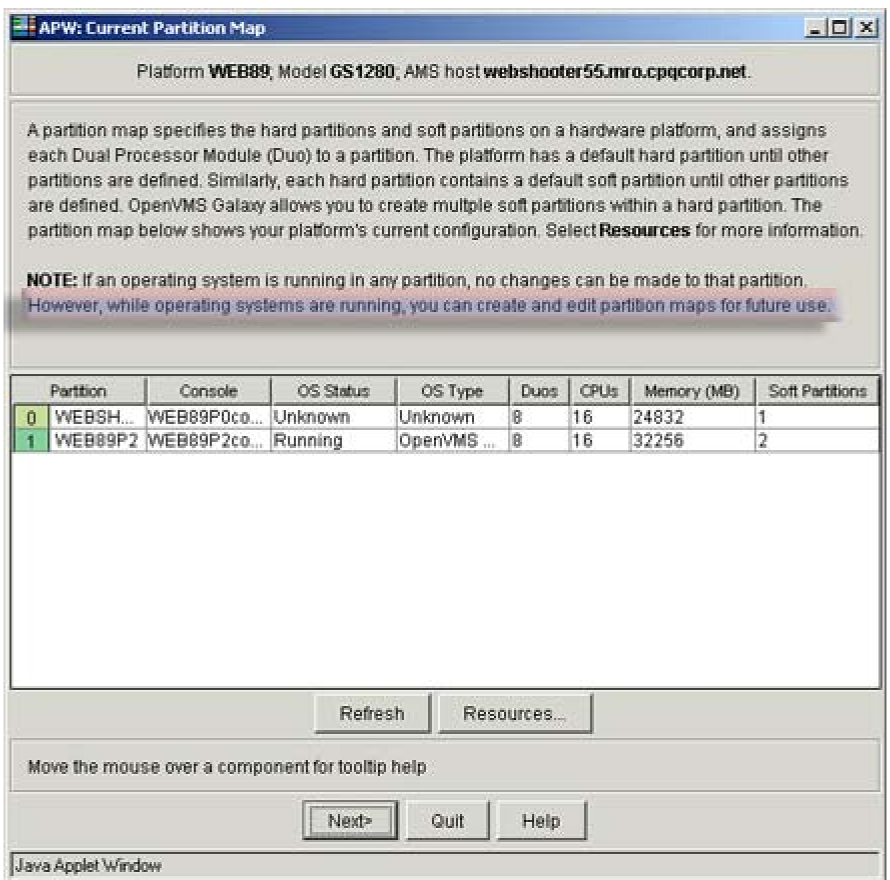
This feature can be useful in several ways:

If your application requires different partitioning at different times, you can create different maps in advance that can be applied with little platform downtime.

You can create a map that can then be applied to several different platforms, without performing the process manually on each platform.

Also, this feature allows you to learn how the APW works without interrupting the operation of the platform. Just don't "Commit" the map (an option presented in a later screen) until you and the platform are ready.

Figure 2-12 APW Startup Screen



2.3.2 APW Resources Window

If you click "Resources" in the first APW screen, the APW displays a graphic view of the current map, as shown in Figure 2-13.

Figure 2-13 APW Display Resources

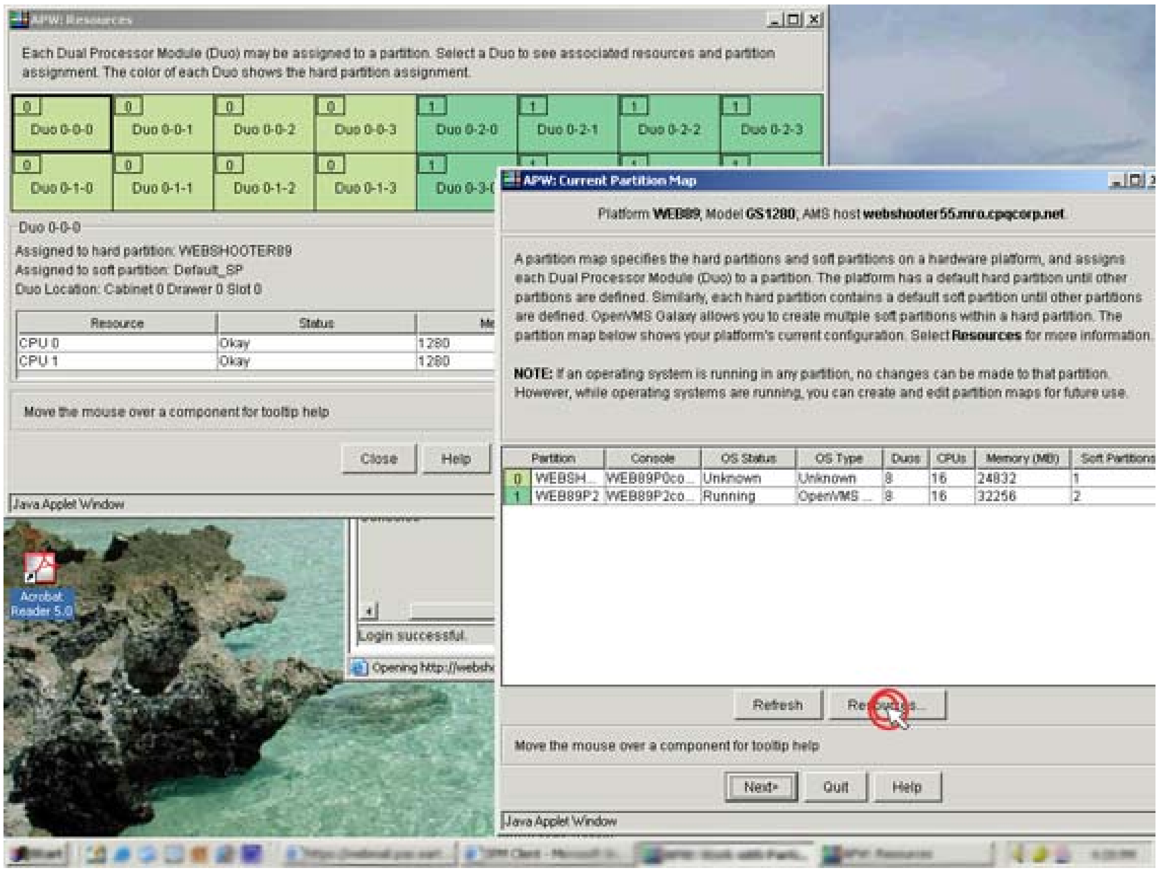


Figure 2-14 APW Resources Window

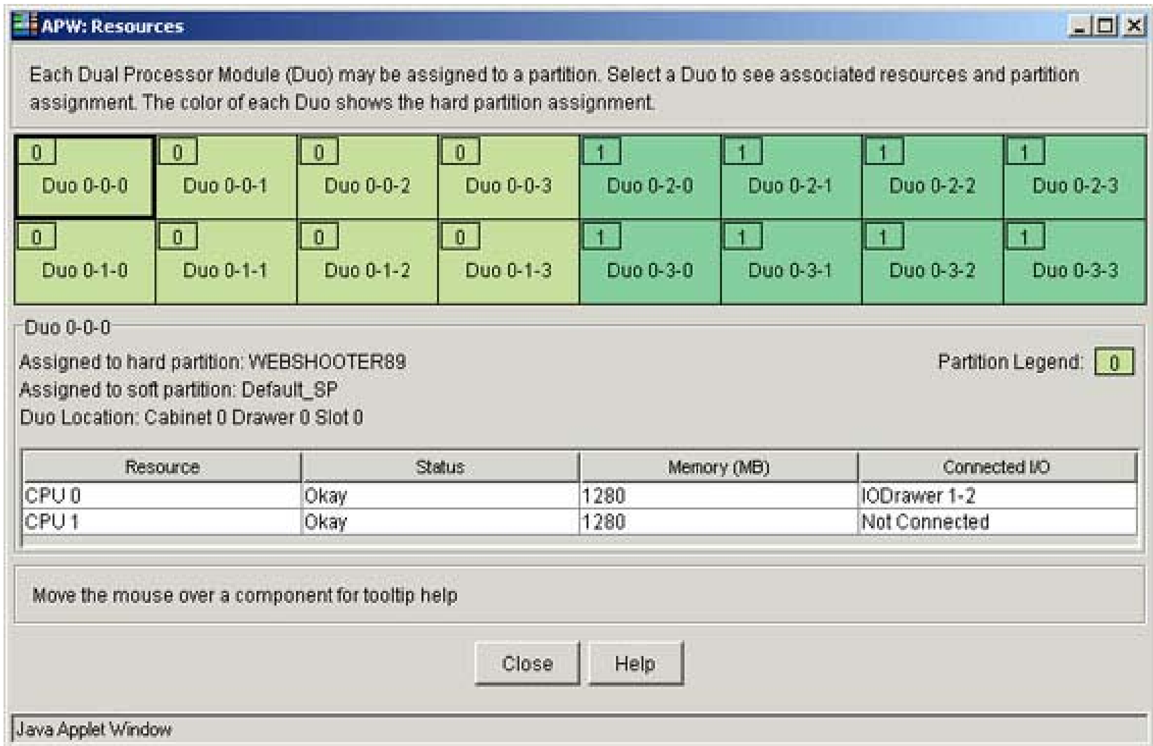


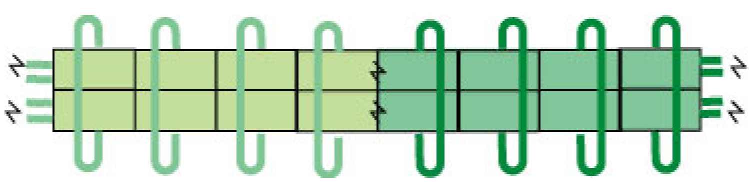
Figure 2-14 shows the Resources screen in more detail.

The Resources screen shows a map of dual-processor modules, or duos. This map is quite similar to the logical display of the AMU. You can display the resources of an individual duo by clicking on it. In the display above, duo 0-0-0 has been selected. The numbering (x-y-z) shows (cabinet - drawer - slot) where slot ranges from 0 - 3, identifying the dual-processor module slots in the drawer.

Current assignments as to hard and soft partition are shown below the colored blocks. Information on the resources associated with this duo (memory, I/O), as well as processor status, is displayed in the table below the colored blocks.

2.3.3 APW Resource Display: the Torus

Figure 2-15 APW Torus

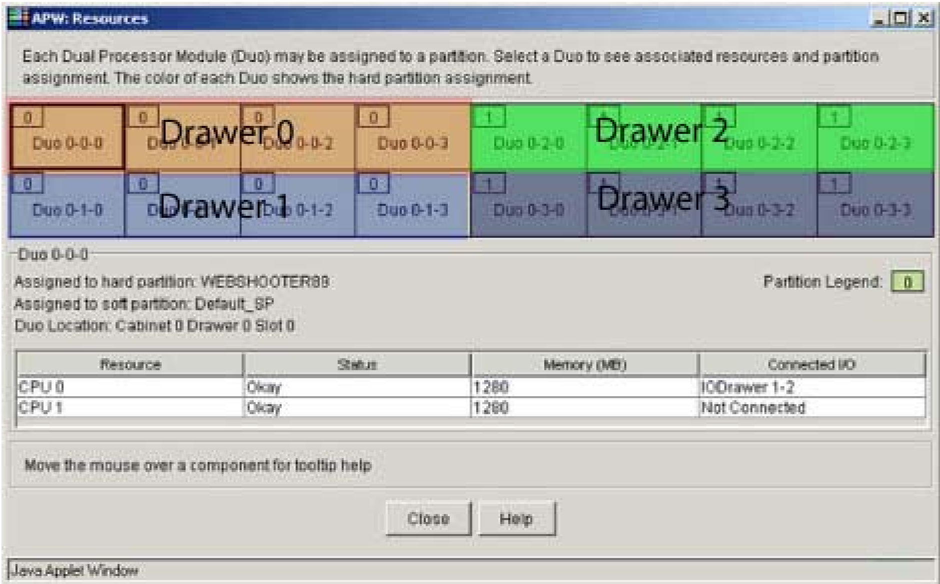


The torus construction is shown in two dimensions by the colored blocks. You simply have to imagine the north-south and east-west "wraps". For example, the two hard partitions shown in Figure 2-15 have top-to-bottom north-south connection in the example above. They do not have left-to-right east-west connections.

Also note that the colors used to distinguish hard partitions are the same as those used for the AMU. And, the hard partitions are numbered as they are in the tree view of the AMU.

The drawer designations are as shown in Figure 2-16.

Figure 2-16 APW Drawer Designation

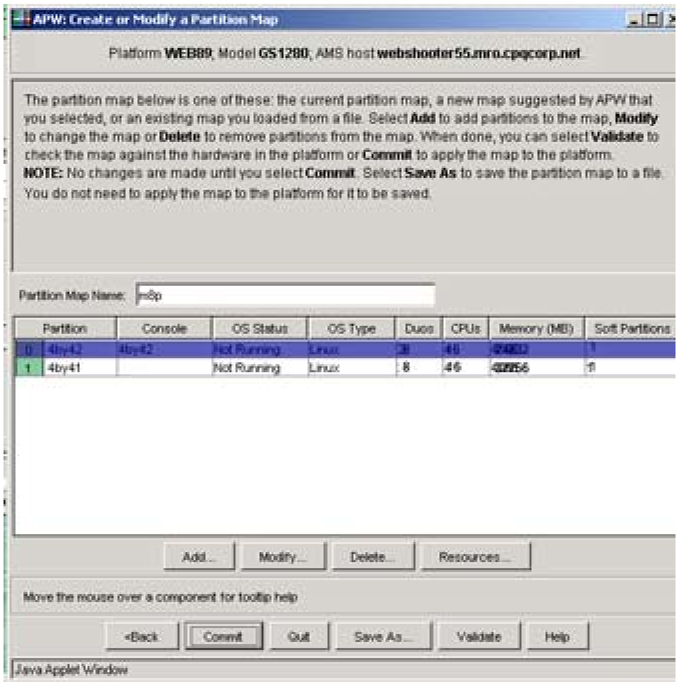


2.3.4 The General Process

The APW provides a wizard-like interface to do partitioning. In the early screens, you make a selection, and click "Next" to go to the next step, as in a normal wizard (see Figure 2-17).

Figure 2-17 APW Wizard Screen

Figure 2-18 APW "Create or Modify a Partition Map" Screen



Once you get to the screen shown in Figure 2-18, the selection you make can display a window on top of the window you are working with. Complete the window on top and close it before continuing. Clicking back to a window "underneath" can be confusing.

2.4 Overview of the CLI

You will see the "MBM>" prompt on a CLI startup screen. At this point, you can type any CLI commands.

The different ways you can access the CLI are discussed in Section x.

2.4.1 CLI Show System Command

The Server Management CLI command **show system** displays the system resources available for partitioning.

Example 2–1 Show System Command

```
MBM> sh sys
SBBs (System Building Block):
  Cab Drw Duo MBM CMM CPUs  CPU0 RIMMs  CPU1 RIMMs      IOP0           IOP1
                       01   0123456789  0123456789 (Cab,Drw).ior  (Cab,Drw).ior
   0   0   .   P
   0   0   0   .   P   PP   PPPP.PPPP.  PPPP.PPPP.  PCI( 1,1 ).0   ...........
   0   0   1   .   P   PP   PPPP.PPPP.  PPPP.PPPP.  ...........    ...........
   0   0   2   .  Filler    ..........  ..........  ...........    ...........
   0   0   3   .  Filler    ..........  ..........  ...........    ...........
   0   1   .   P
   0   1   0   .   P   PP   PPPP.PPPP.  PPPPPPPPPP  PCI( 1,3 ).0   ...........
   0   1   1   .   P   PP   PPPP.PPPP.  PPPP.PPPP.  ...........    ...........
   0   1   2   .  Filler    ..........  ..........  ...........    ...........
   0   1   3   .  Filler    ..........  ..........  ...........    ...........
   0   2   .   P
   0   2   0   .   P   PP   PPPP.PPPP.  PPPP.PPPP.  ...........    ...........
   0   2   1   .   P   PP   PPPP.PPPP.  PPPP.PPPP.  PCI( 1,4 ).0   ...........
   0   2   2   .  Filler    ..........  ..........  ...........    ...........
   0   2   3   .  Filler    ..........  ..........  ...........    ...........
   0   3   .   P
   0   3   0   .   P   PP   PPPP.PPPP.  PPPP.PPPP.  PCI( 1,2 ).0   ...........
   0   3   1   .   P   PP   PPPP.PPPP.  PPPP.PPPP.  ...........    ...........
   0   3   2   .  Filler    ..........  ..........  ...........    ...........
   0   3   3   .  Filler    ..........  ..........  ...........    ...........
PCI Drawers:
  Cab Drw PBM IORs       IOR0           IOR1           IOR2           IOR3
              0123  (Cab,Drw).iop  (Cab,Drw).iop  (Cab,Drw).iop  (Cab,Drw).iop
   1   1   P  P      SBB( 0,0 ).0
   1   2   P  P      SBB( 0,3 ).0
   1   3   P  P      SBB( 0,1 ).0
   1   4   P  P      SBB( 0,2 ).2
SBB Power Sub-systems:
  Cab Drw  Type  WPI/SDI  Power Supply  Powered SBB
                  0123    0123456789AB  SBB(cab,drw)
   1   0    WPI    P...   PPP.........   SBB( 0,0 )
   1   1    WPI    P...   PPP.........   SBB( 0,1 )
   1   2    WPI    P...   PPP.........   SBB( 0,2 )
   1   3    WPI    P...   PPP.........   SBB( 0,3 )
MBM> p off -all
MBM> sh sys
SBBs (System Building Block):
  Cab Drw Duo MBM CMM CPUs  CPU0 RIMMs  CPU1 RIMMs      IOP0           IOP1
                       01   0123456789  0123456789 (Cab,Drw).ior  (Cab,Drw).ior
   0   0   .   P
   0   0   0   .   p   pp   pppp.pppp.  pppp.pppp.  PCI( 1,1 ).0   ...........
   0   0   1   .   p   pp   pppp.pppp.  pppp.pppp.  ...........    ...........
   0   0   2   .  Filler    ..........  ..........  ...........    ...........
   0   0   3   .  Filler    ..........  ..........  ...........    ...........
   0   1   .   P
   0   1   0   .   p   pp   pppp.pppp.  pppppppppp  PCI( 1,3 ).0   ...........
```

```
   0   1   1   .   p   pp   pppp.pppp.  pppp.pppp.  ..........    ..........
   0   1   2   .  Filler    ..........  ..........  ..........    ..........
   0   1   3   .  Filler    ..........  ..........  ..........    ..........
   0   2   .   P
   0   2   0   .   p   pp   pppp.pppp.  pppp.pppp.  ..........    ..........
   0   2   1   .   p   pp   pppp.pppp.  pppp.pppp.  PCI( 1,4 ).0  ..........
   0   2   2   .  Filler    ..........  ..........  ..........    ..........
   0   2   3   .  Filler    ..........  ..........  ..........    ..........
   0   3   .   P
   0   3   0   .   p   pp   pppp.pppp.  pppp.pppp.  PCI( 1,2 ).0  ..........
   0   3   1   .   p   pp   pppp.pppp.  pppp.pppp.  ..........    ..........
   0   3   2   .  Filler    ..........  ..........  ..........    ..........
   0   3   3   .  Filler    ..........  ..........  ..........    ..........
PCI Drawers:
  Cab Drw PBM IORs       IOR0           IOR1           IOR2           IOR3
              0123  (Cab,Drw).iop  (Cab,Drw).iop  (Cab,Drw).iop  (Cab,Drw).iop
   1   1   P  p      SBB( 0,0 ).0
   1   2   P  p      SBB( 0,3 ).0
   1   3   P  p      SBB( 0,1 ).0
   1   4   P  p      SBB( 0,2 ).2
SBB Power Sub-systems:
  Cab Drw  Type  WPI/SDI  Power Supply  Powered SBB
                  0123    0123456789AB  SBB(cab,drw)
   1   0    WPI    p...   ppp.........   SBB( 0,0 )
   1   1    WPI    p...   ppp.........   SBB( 0,1 )
   1   2    WPI    p...   ppp.........   SBB( 0,2 )
   1   3    WPI    p...   ppp.........   SBB( 0,3 )
MBM>
```

2.4.2 CLI Show Commands: the Torus

The Server Management CLI **show grid** command displays a two-dimensional view of the torus of the entire platform. The **show partition** command shows the current partition divisions.

Example 2–2 Show Grid Command

```
MBM> sh grid
Powerup grid for partition Default_HP
     0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
     ................................................
0   .P--P--P--F.....................................
    .|..|..|..|.....................................  P  Processor
    .|..|..|..|.....................................  F  Filler
1   .P--P--P--F.....................................  |  connection
    ................................................  -  connection
    ................................................
2   ................................................   W <-----------> E
    ................................................ N ^ (0,0)
    ................................................   |  .
3   ................................................   |    .
    ................................................   |       .
    ................................................ S v      (ns,ew)
4   ................................................
    ................................................
    ................................................
5   ................................................
    ................................................
    ................................................
6   ................................................
    ................................................
    ................................................
7   ................................................
```

```
      . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
       0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
MBM>
```

2.4.3 The General Process

In general, you use the Server Management Command create partition to create hard partitions and subpartitions, and the assign component command to assign resources to partitions. The process is described in more detail in Chapter 5.

Chapter 3
Using the AMU to Create Partitions

The AlphaServer Management Utility (AMU) provides a graphical interface for partitioning.

3.1 Running the AMU

3.1.1 Running the AMU Standalone from a PC or Workstation

You can install the AMU on a Windows 2000 PC or on a workstation and, after adding a platform as accessible to the AMU in standalone mode, plug the PC or workstation into the NAT box on the platform and run the AMU from there.

To plug the PC or workstation into a platform, use a network cable (RJ45) to connect the laptop/PC/workstation to a spare port in the LAN side of the hub, as shown highlighted in red in Figure 3-1.

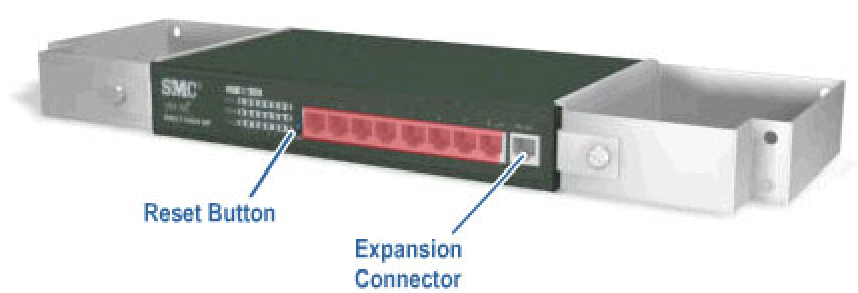
Directions for installing the AMU on a Windows 2000 PC or on a workstation are given at:

ftp://ftp.digital.com/pub/Digital/Alpha/firmware/interim/ams/index.html

Directions for adding a platform as accessible to the AMU in standalone mode are given in the ES47/ES80/GS1280 AlphaServer Management Station User's Guide at:

ftp://ftp.digital.com/pub/Digital/Alpha/firmware/interim/ams/releases/Version_2_1_0/ams-user-guide.pdf

Figure 3-1 LAN Connections



3.1.2 Login from the SPM

Figure 3-2 Step 1: Select "Login" from File Menu

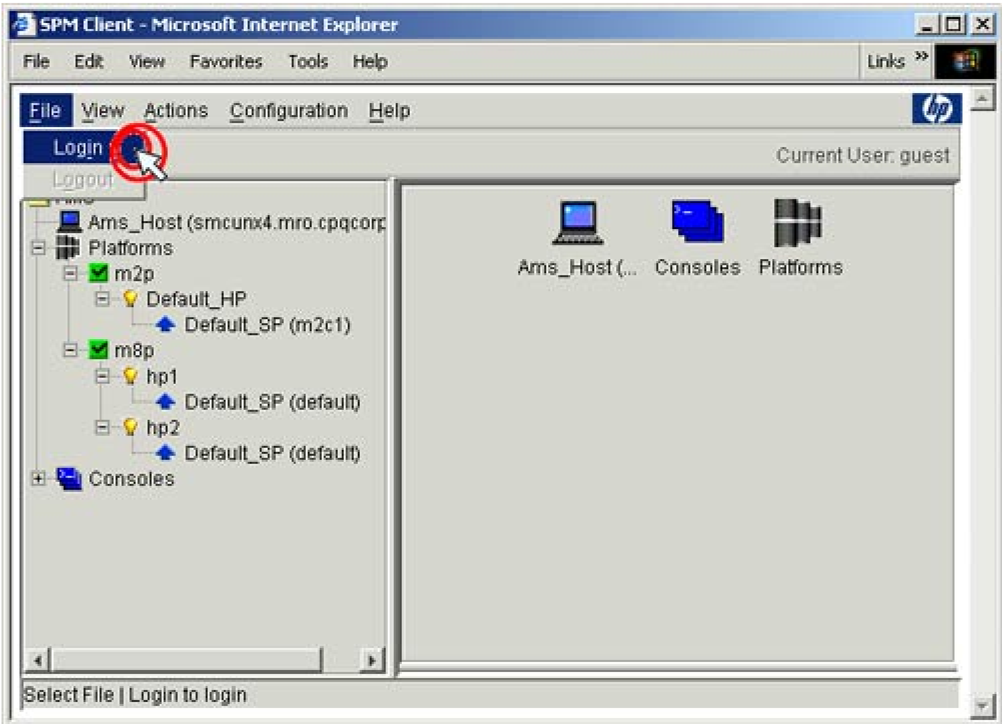


Figure 3-3 Type your username and password in the login screen and left click "OK".

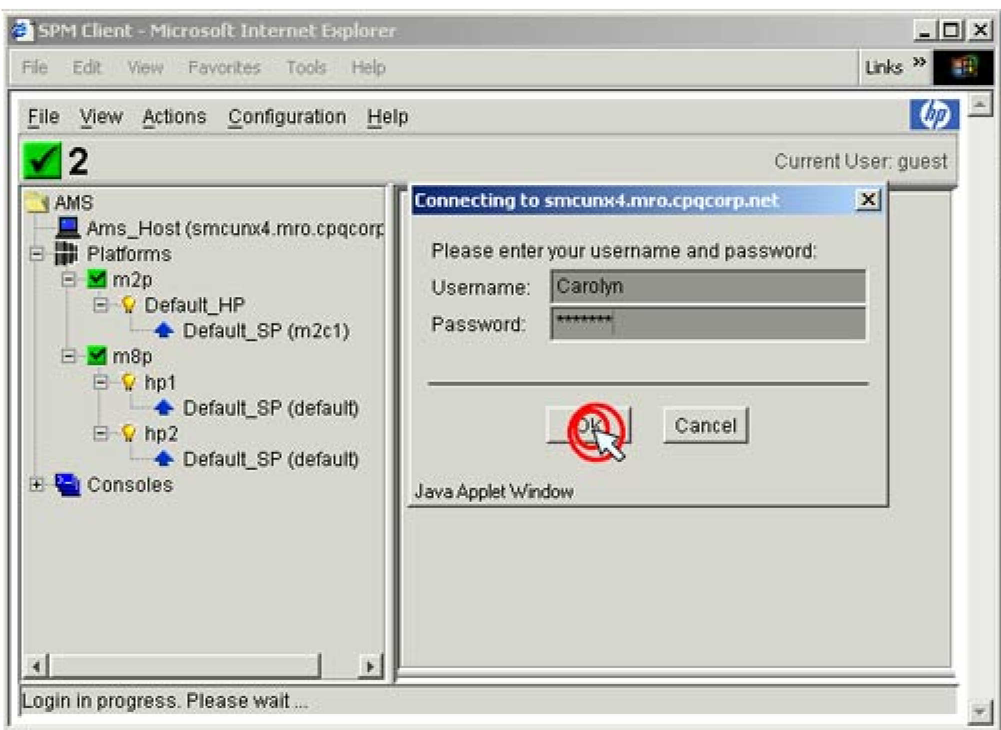


Figure 3-4 Right-Click on Platform and Select “AMU”

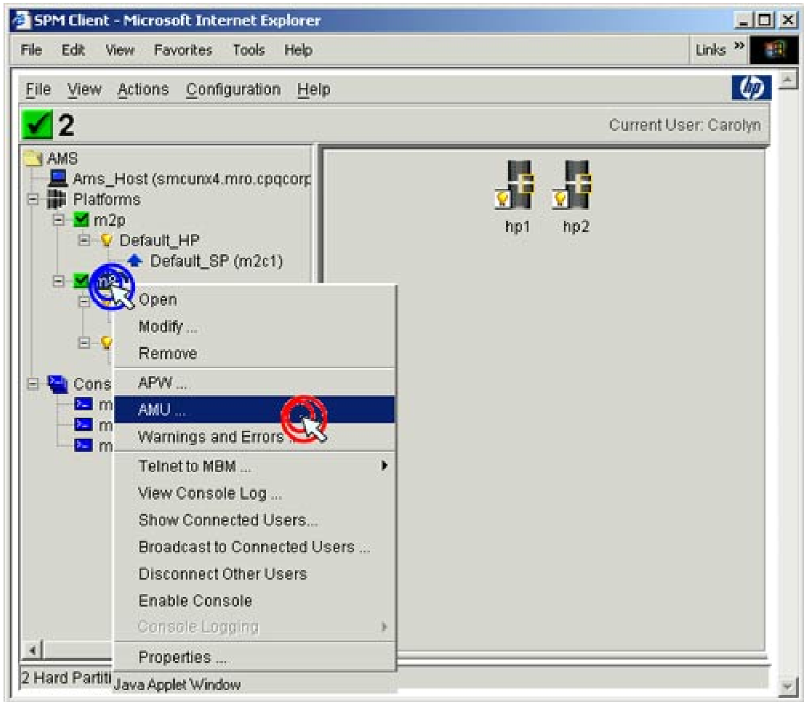
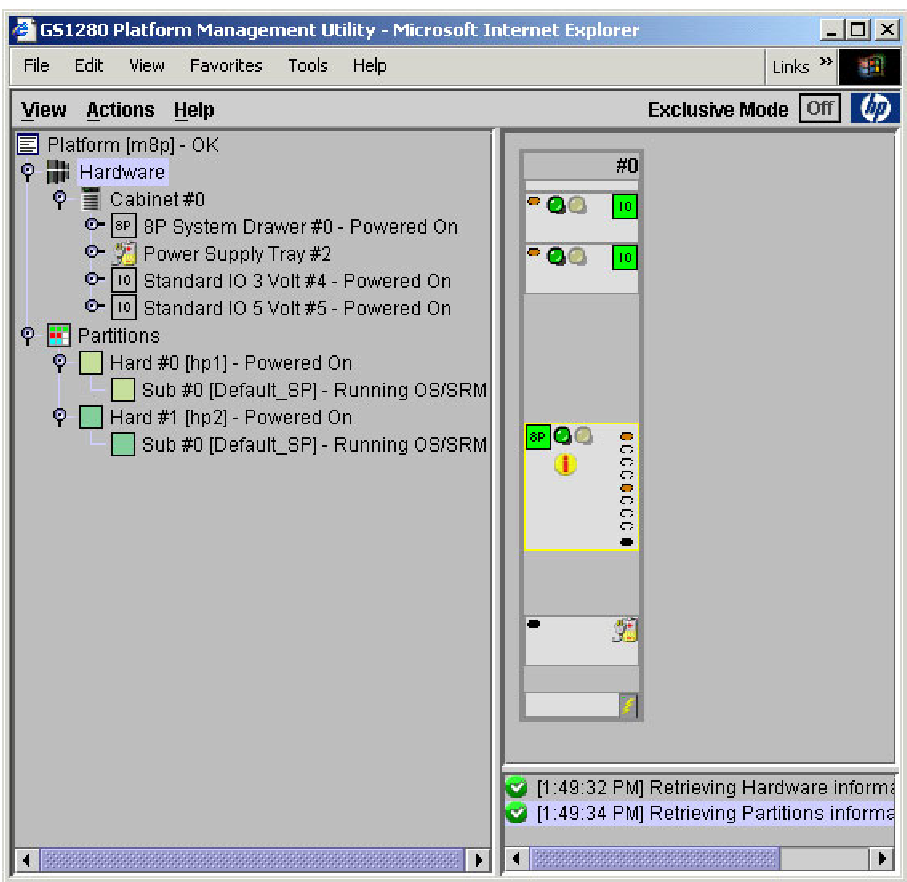


Figure 3-5   AMU Hardware View Displays

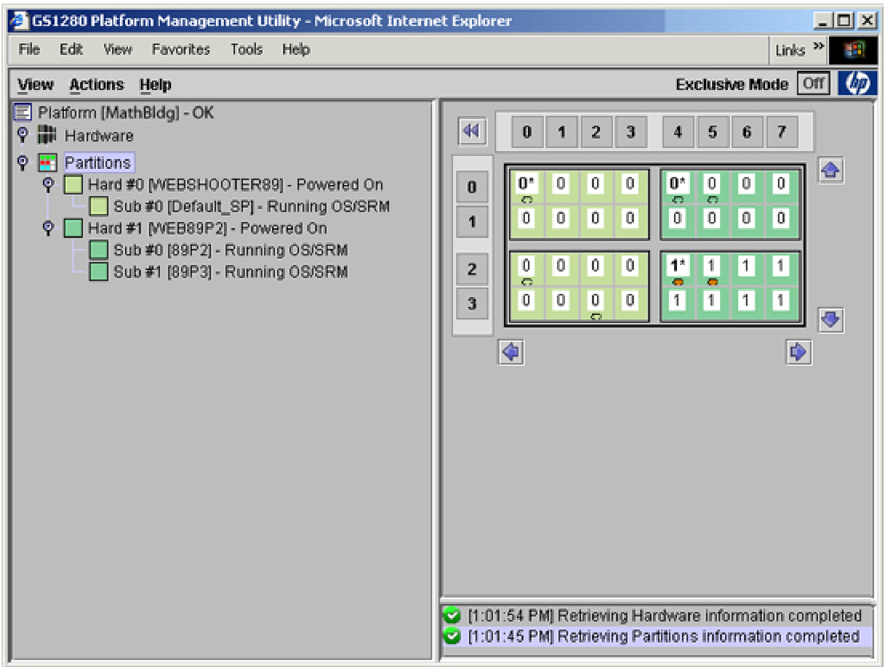


3.2 Displaying the Partition View

When the AMU first comes up, it displays the "hardware view" in the right top pane of the window, as shown in Figure 3-5.

To see the "logical view" in this pane that is used for partitioning, you click on the "Partitions" icon in the tree view on the left. A sample logical view is shown in Figure 3-6.

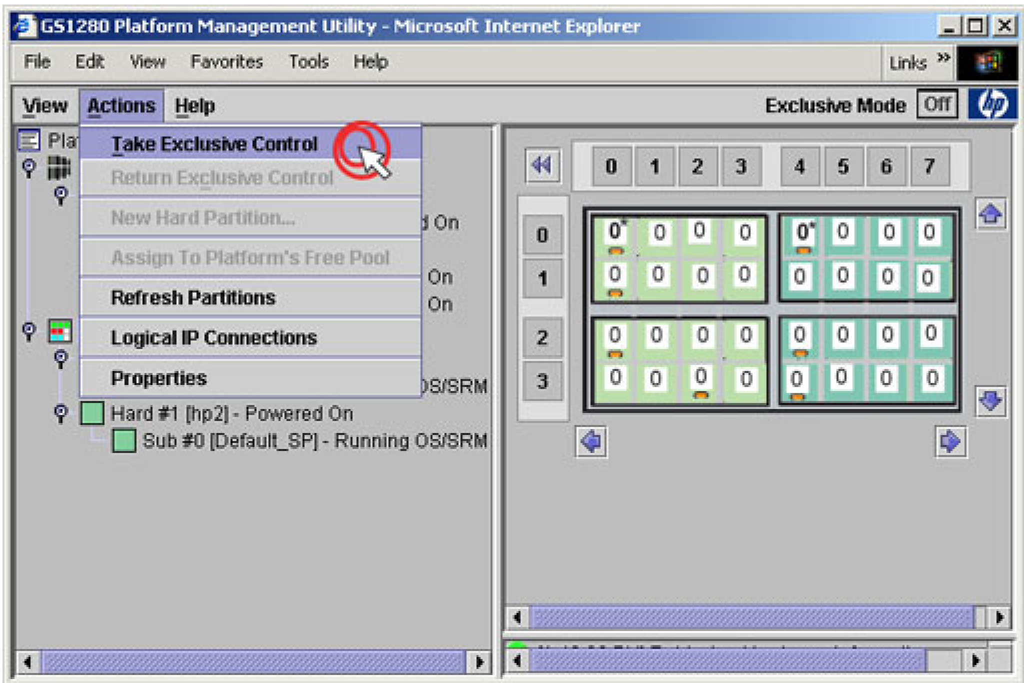
Figure 3-6 AMU Logical View



3.3 Take Exclusive Control

Take exclusive control of the platform, to ensure that your partitioning work cannot be inadvertently changed by others working with the AMU through the same AMU server. If someone else has exclusive control of this platform, you will not be able to take control until they have relinquished it.

Figure 3-7 Take Exclusive Control

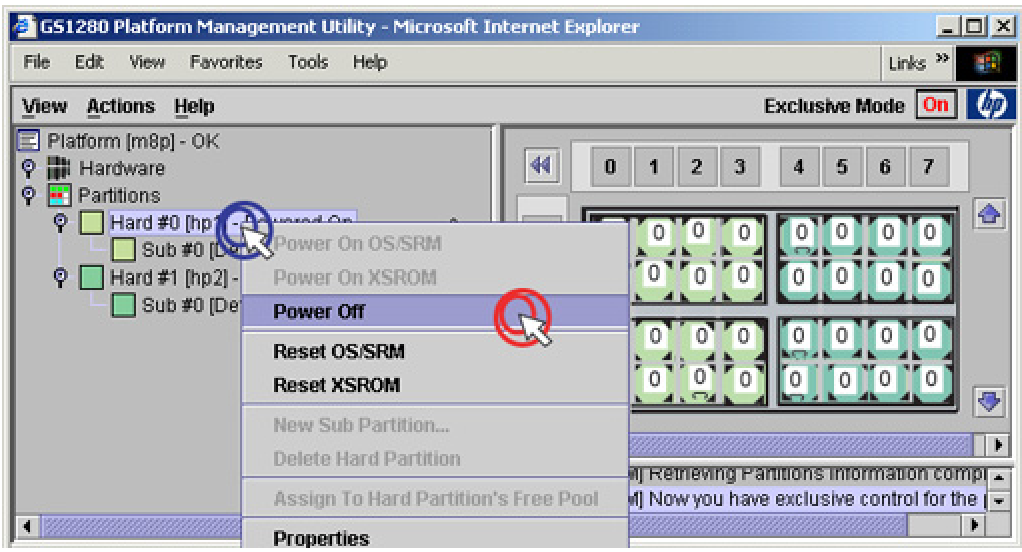


3.4 Power Off Existing Hard Partitions

To create new partitions on a platform, power off all existing hard partitions.

For each hard partition in the tree view, right click on the hard partition name, select "Power Off" from the drop-down menu, and left-click.

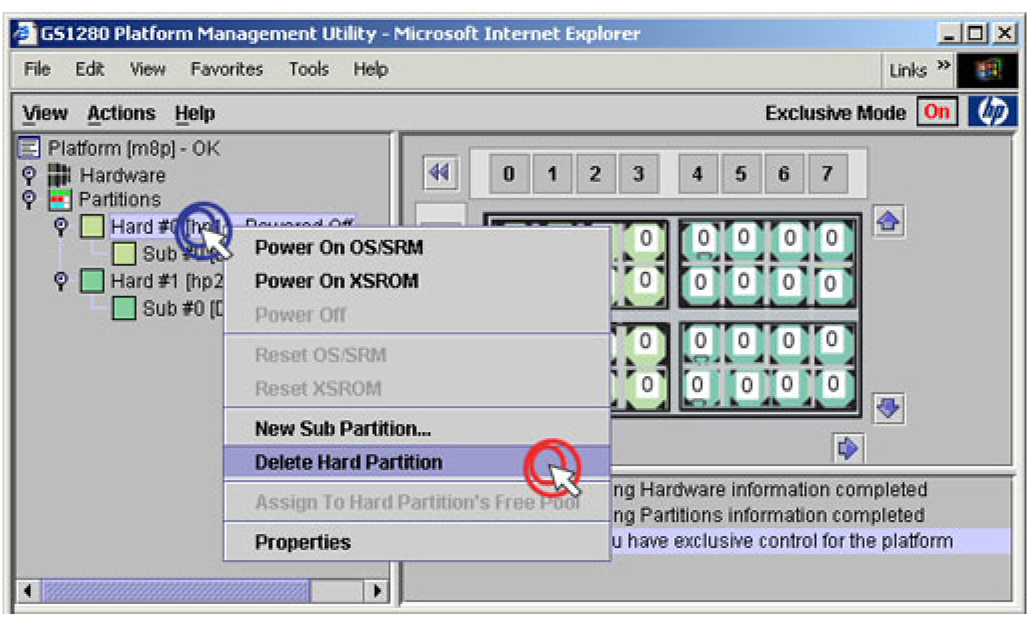
Figure 3-8 Power Off Existing Hard Partitions



3.5 Delete Existing Partitions

Once all partitions have been powered off, delete them by right-clicking on each hard partition name in the tree view and selecting "Delete Hard Partition" from the drop-down menu.

Figure 3-9 Delete Existing Partitions

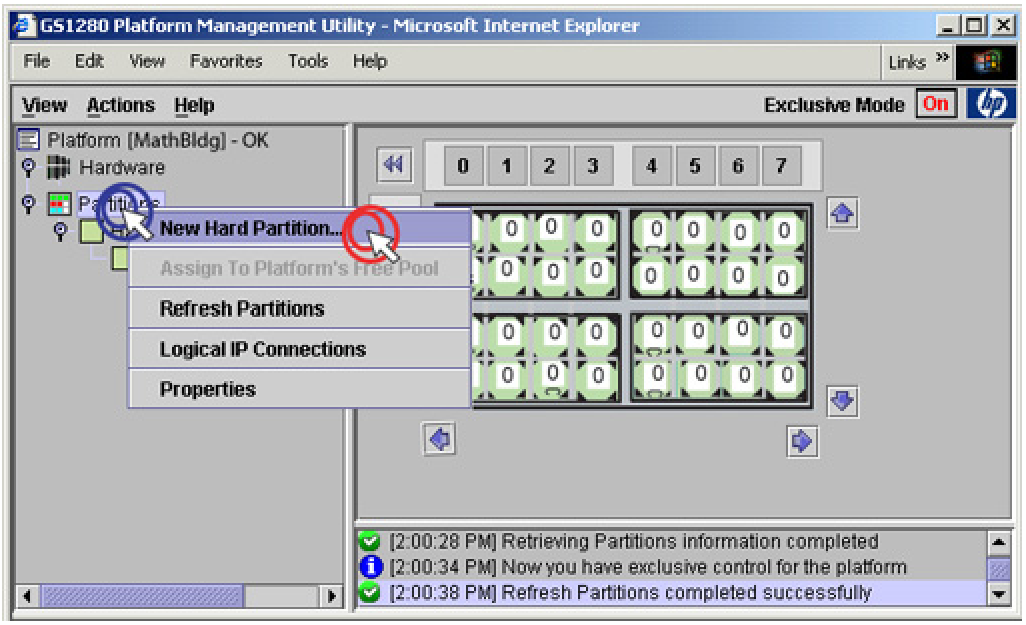


3.6 Create New Hard Partitions in Tree View

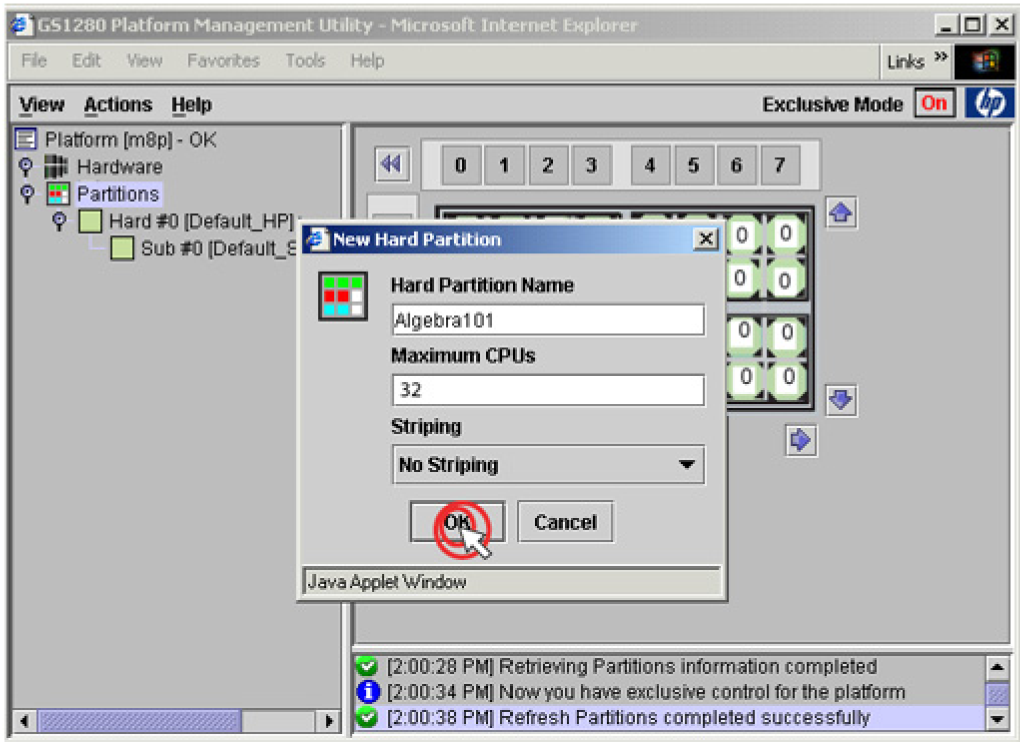
Using the plan you devised, create the partitions for the platform in the tree view. First create hard partitions (if any), and then subpartitions (if any). To create hard partitions:

1. Right click on the platform and select "New Hard Partition" from the pulldown menu.

Figure 3-10 Select "New Hard Partition"

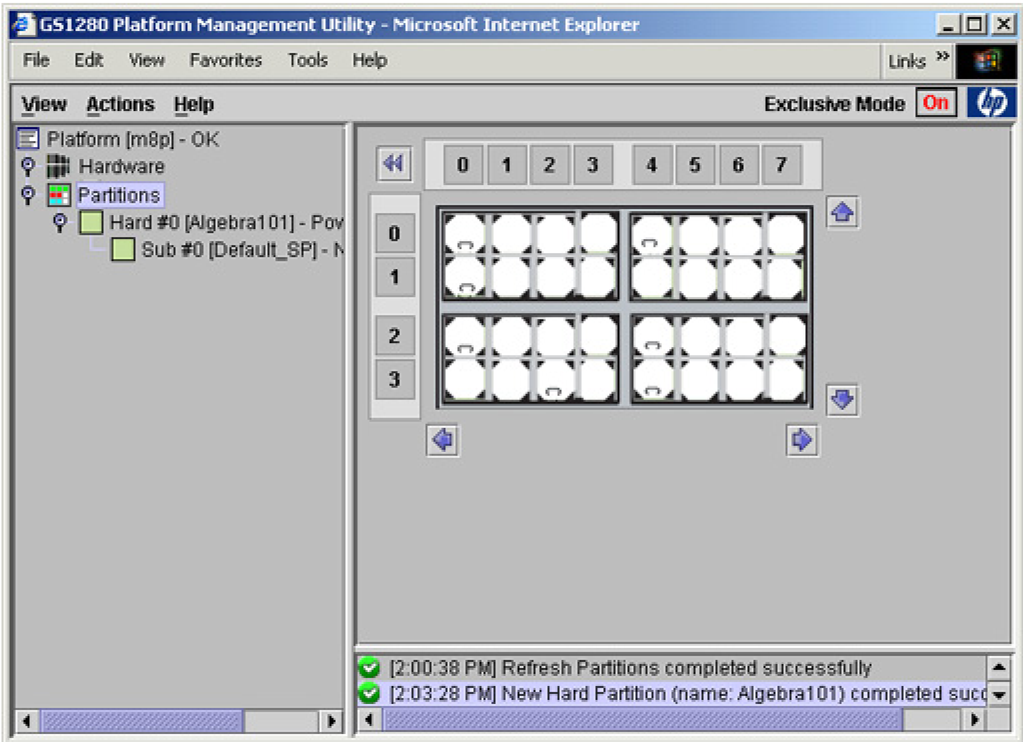


2 Fill out the form with the hard partition name. The default maximum number of CPUs will always be the total number of CPUs available. It is fine to leave it at that. Click OK.



3 A new hard partition has been created, but it is as yet unpopulated. The logical view shows white powered off resources. Repeat steps 1 and 2 until you have created all the hard partitions you want.

Figure 3-11 Powered Off, Unassigned Resources

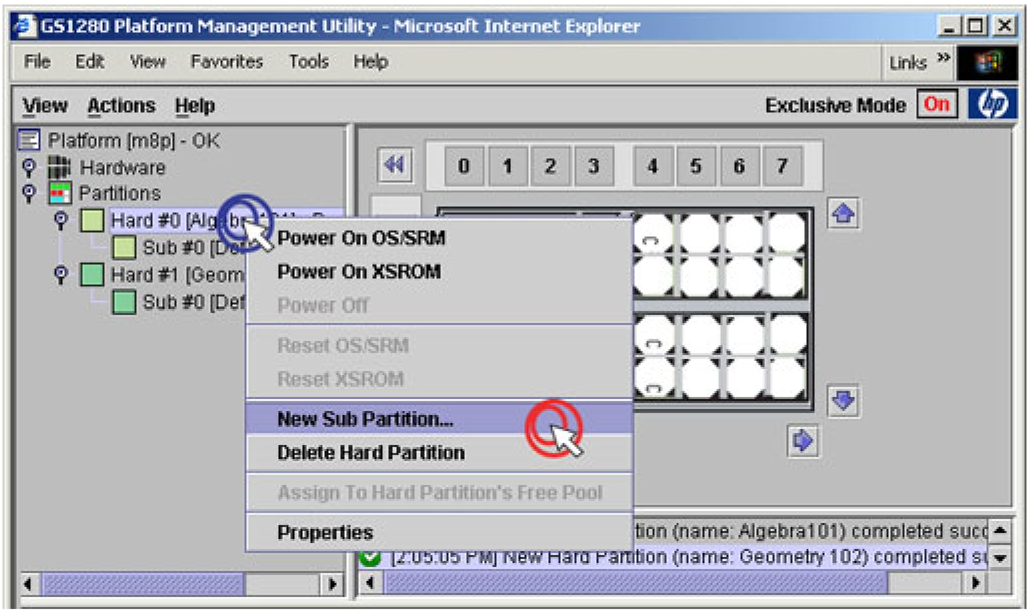


3.7 Create New Subpartitions in the Tree View

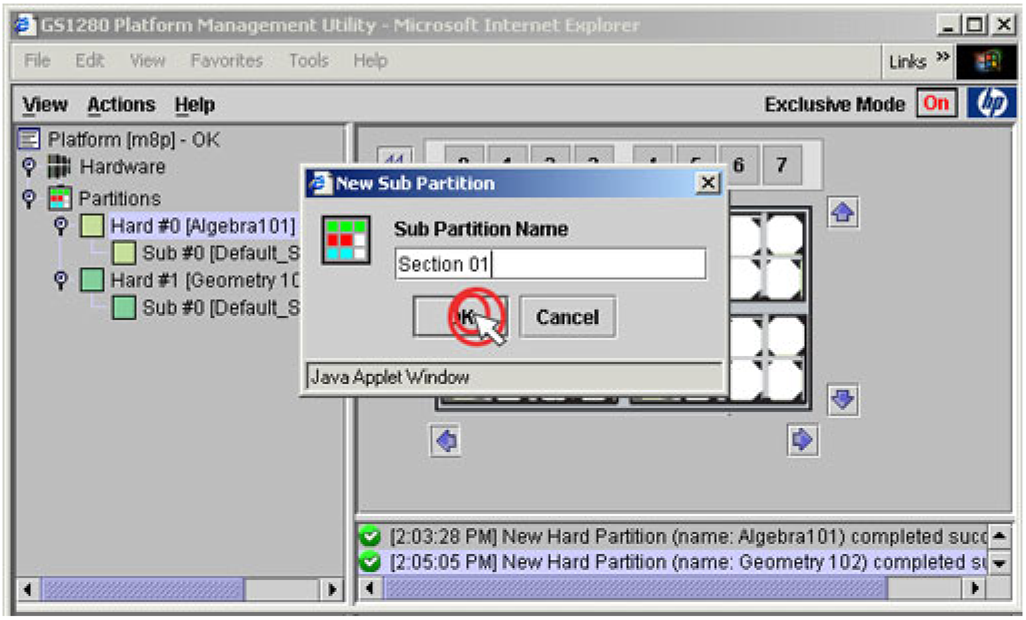
You can create subpartitions in any hard partition you have created. This example shows the creation of subpartitions after two hard partitions have been created.

1 Select the hard partition under which you wish to create subpartitions by right-clicking on it. From the dropdown menu, select "New Sub Partition..."

Figure 3-12 Create New Subpartitions



2 Fill in the form with the subpartition name and click "OK". Repeat steps 1 and 2 for as many subpartitions as you wish.

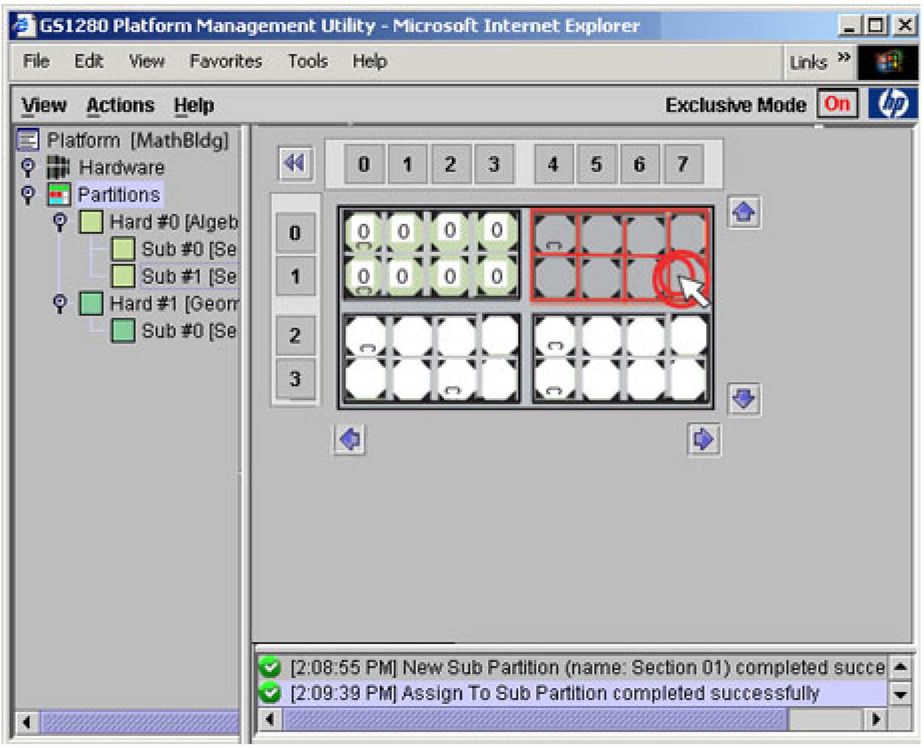


3.8 Assign Resources to Partitions

Once you have created partitions and subpartitions, you assign resources to them.

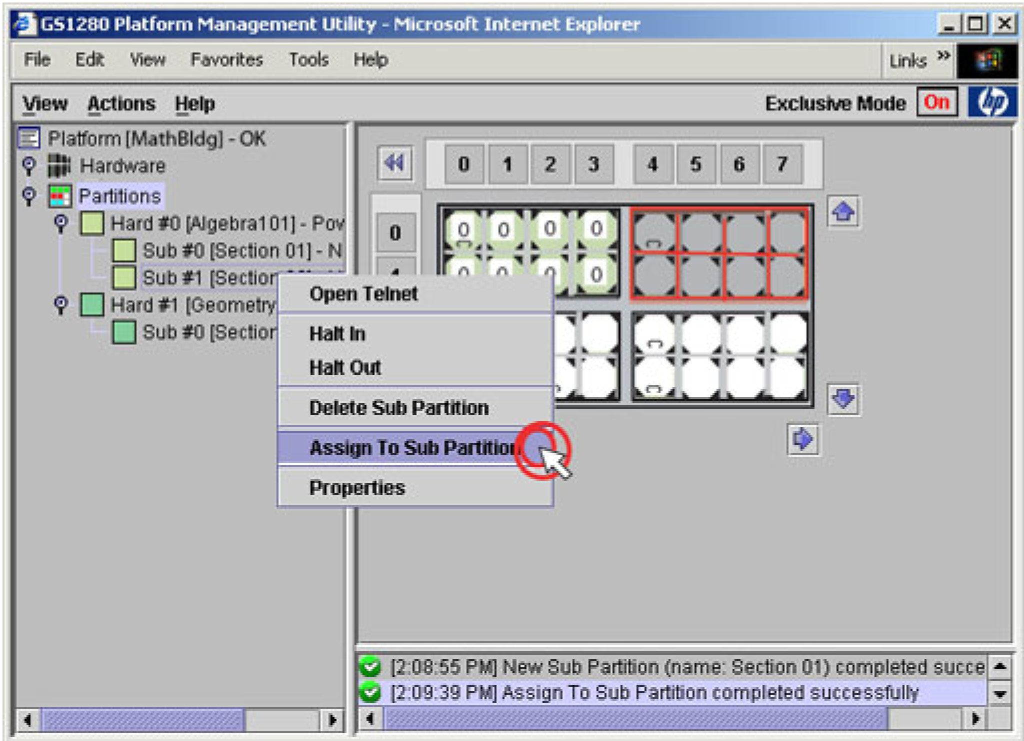
1 Select resources in the logical view by holding down the left mouse button and passing it over the resources you want to assign to a particular partition. As you do, the resources are grayed out and surrounded by a red box. Mistakes are easily corrected: simply release the mouse button and click outside the logical resource area and start over.

Figure 3-13 Select Resources



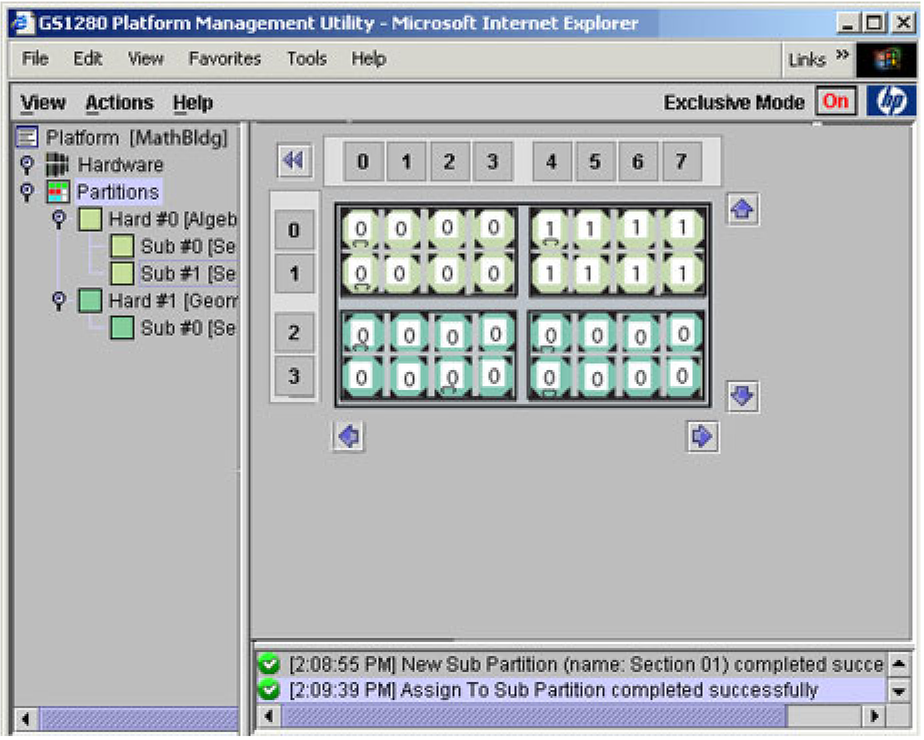
2 Once you have grayed out the resources you want to add to a partition, right click on the partition name in the tree view and select "Assign to Sub Partition".

Figure 3-14 Assign to Subpartition



3 This picture shows two hard partitions, one with two subpartitions. All are still powered off.

Figure 3-15 Powered Off Partitions and Subpartitions

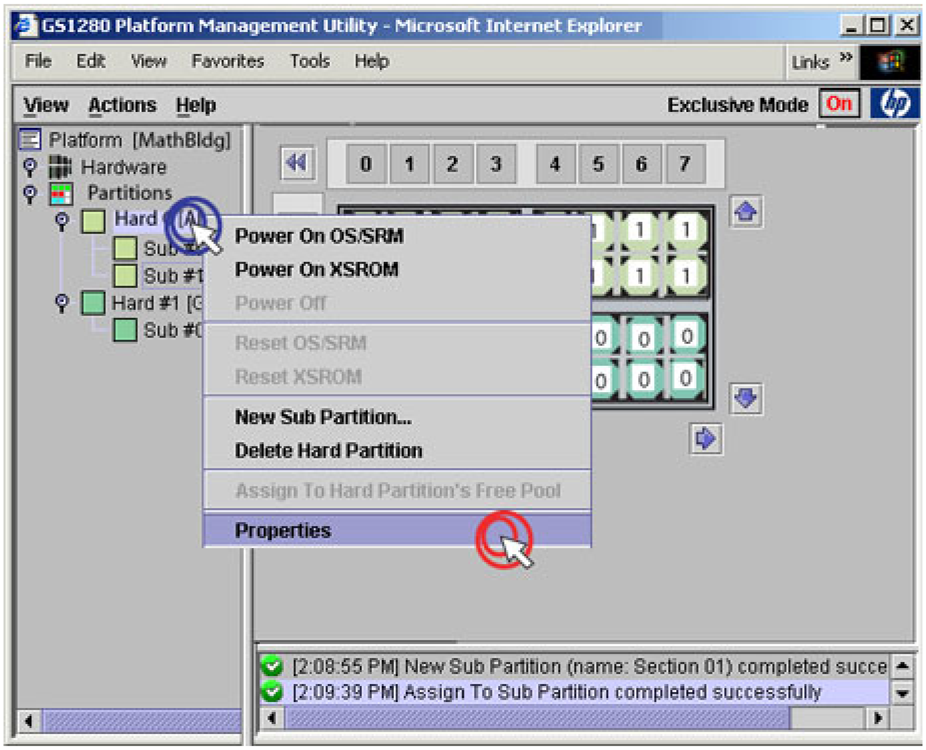


3.9 Assign Community Memory to Subpartitions

To assign community memory to subpartitions, follow these steps.

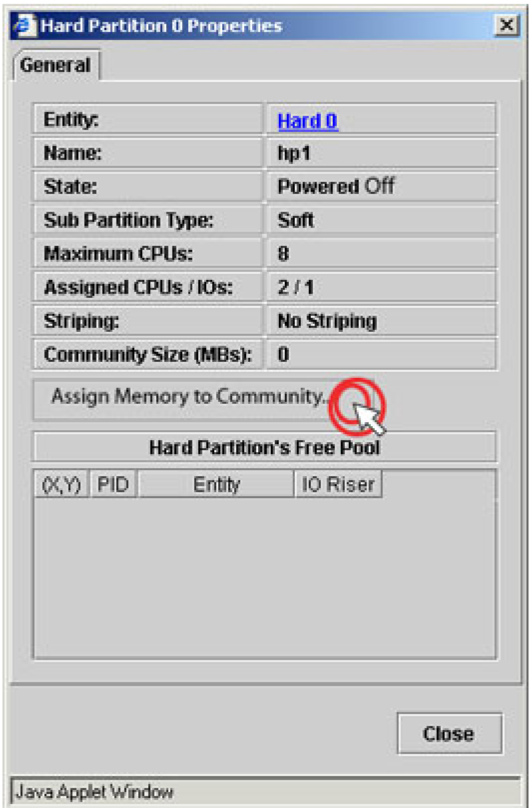
1 Right-click on the hard partition containing the soft partitions to which you wish to assign community memory. Select "Properties" from the drop-down menu.

Figure 3-16 Right-Click Hard Partition, and Select “Properties” from Menu



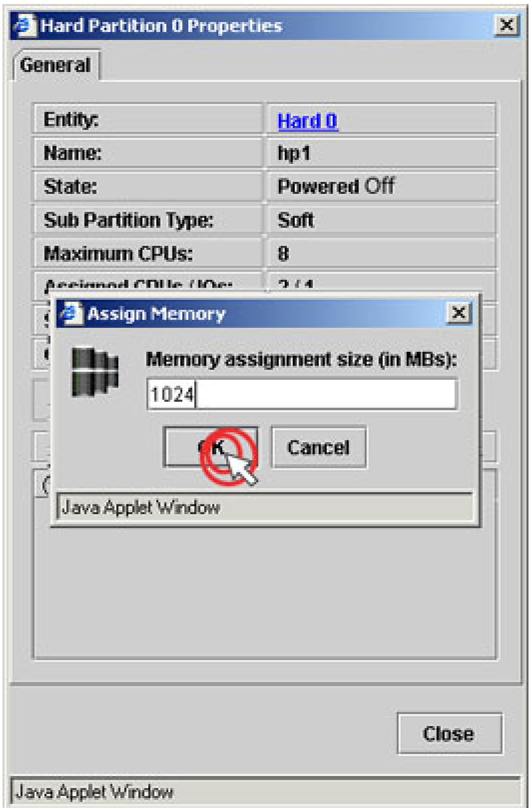
2 The "Properties" window displays. Left-click on "Assign Memory to Community..." (If assigning community memory is not appropriate, this selection will be grayed out. This can happen if you select a hard partition with only one subpartition, select a subpartition, or if any of the subpartitions are powered on.)

Figure 3-17 Select "Assign Memory to Community"



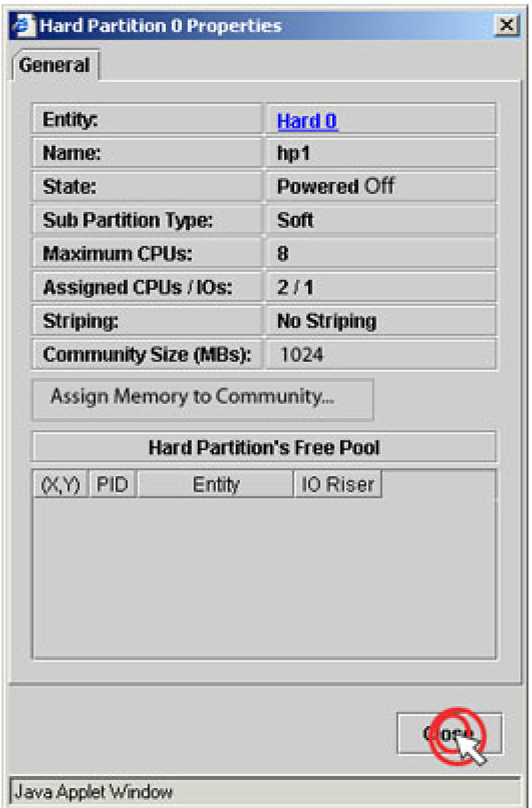
3 Type in the amount of memory you wish to assign to the community and click "OK".

Figure 3-18 Type in Amount of Community Memory



4 The "Properties" window displays the new assignment. Click on "Close" (at the bottom of the window) to close the Properties window.

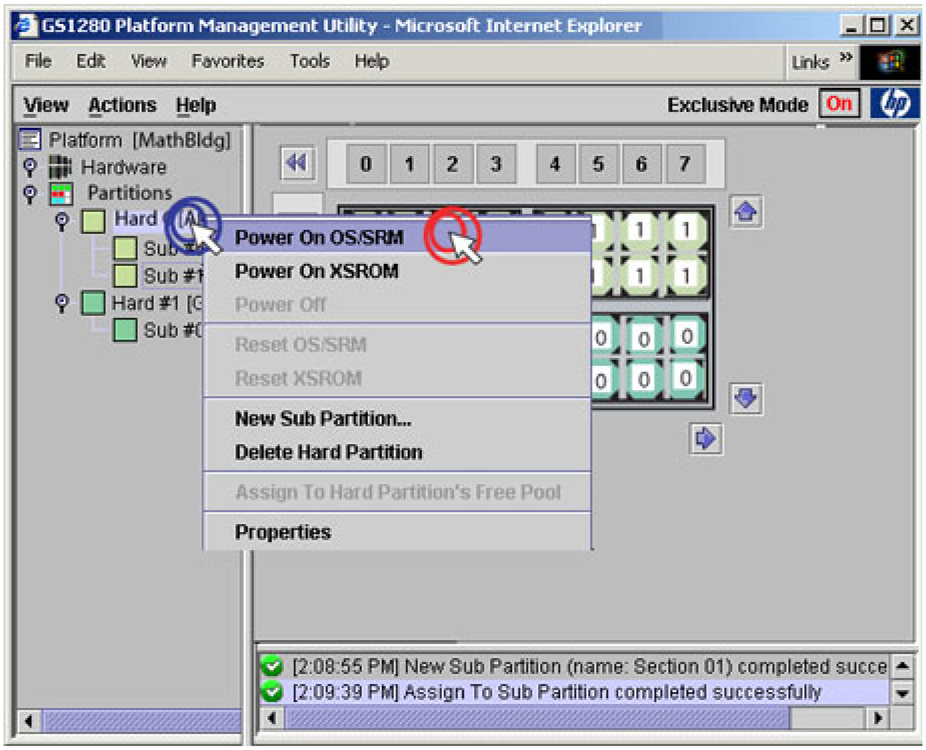
Figure 3-19 Close the Properties Window



3.10 Power On Hard Partitions

Once you have defined and populated all the hard partitions and subpartitions for a platform, power on each hard partition, as shown below.

Figure 3-20 Power On Each Hard Partition



Chapter 4
Using the APW to Create Partitions

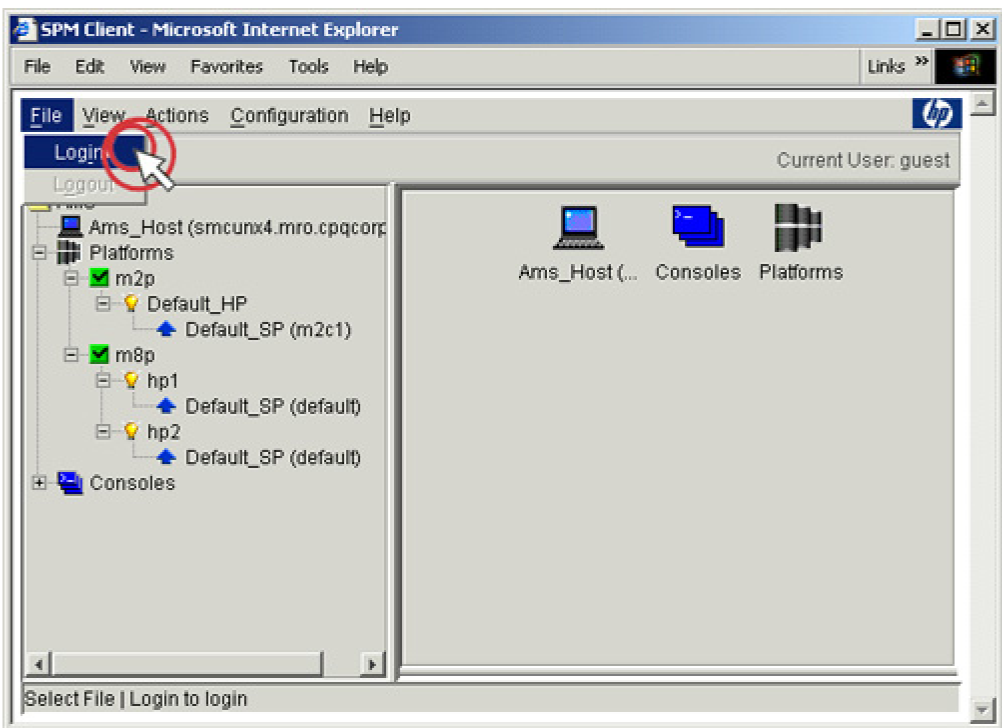
The AlphaServer Partitioning Wizard (APW) provides a wizard interface for partitioning

4.1 Running the APW

You can run the APW from the SPM. Follow the steps below.

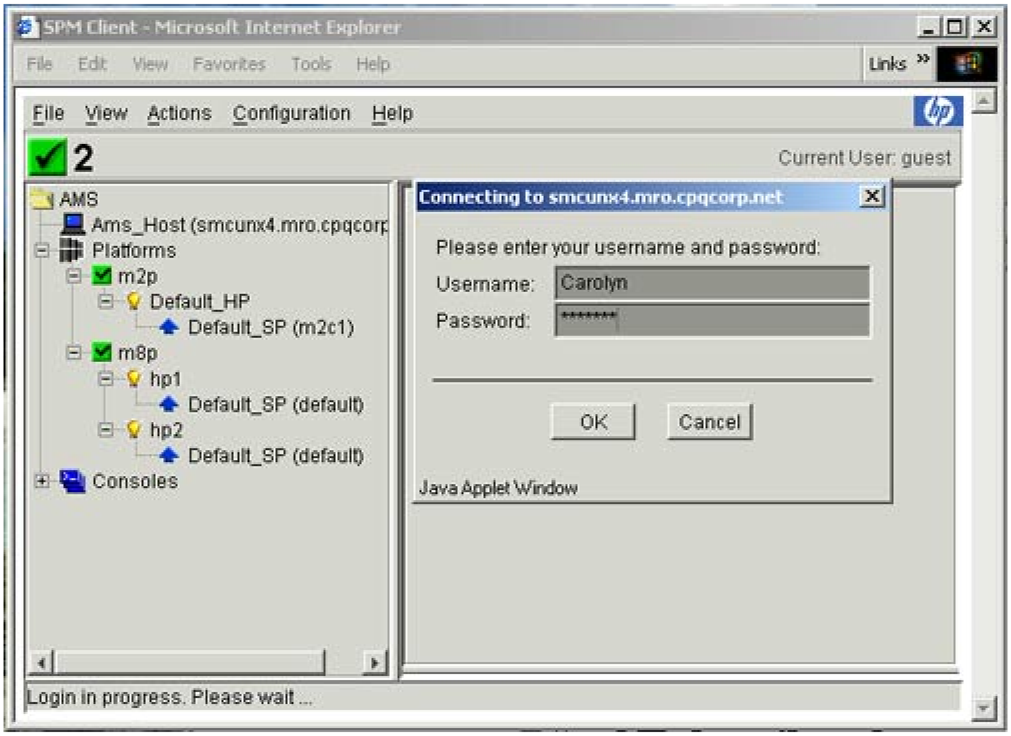
1 From the "File" menu, pull down and left-click "Login."

Figure 4-1 Select "Login" from the SPM "File" Menu



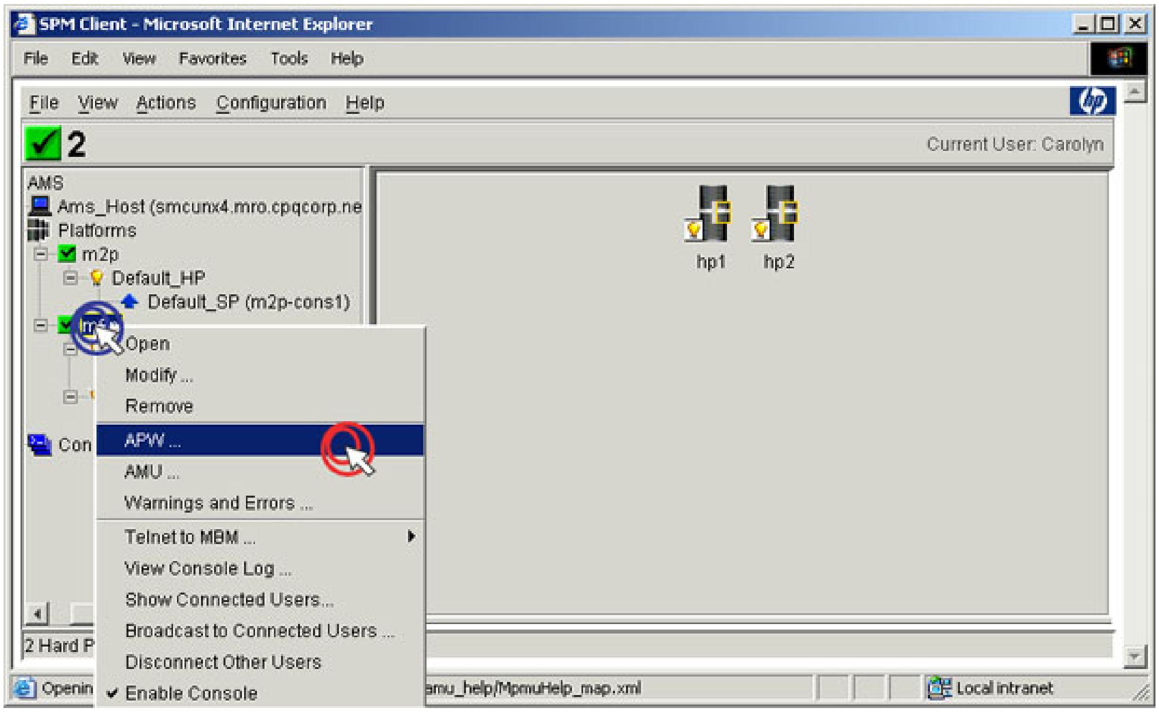
2 Type your username and password in the login screen and left click "OK".

Figure 4-2 Type Username and Password



3 Now you can start the APW. Right-click on the platform that you want to partition, and slide the mouse down the submenu to the APW and left-click.

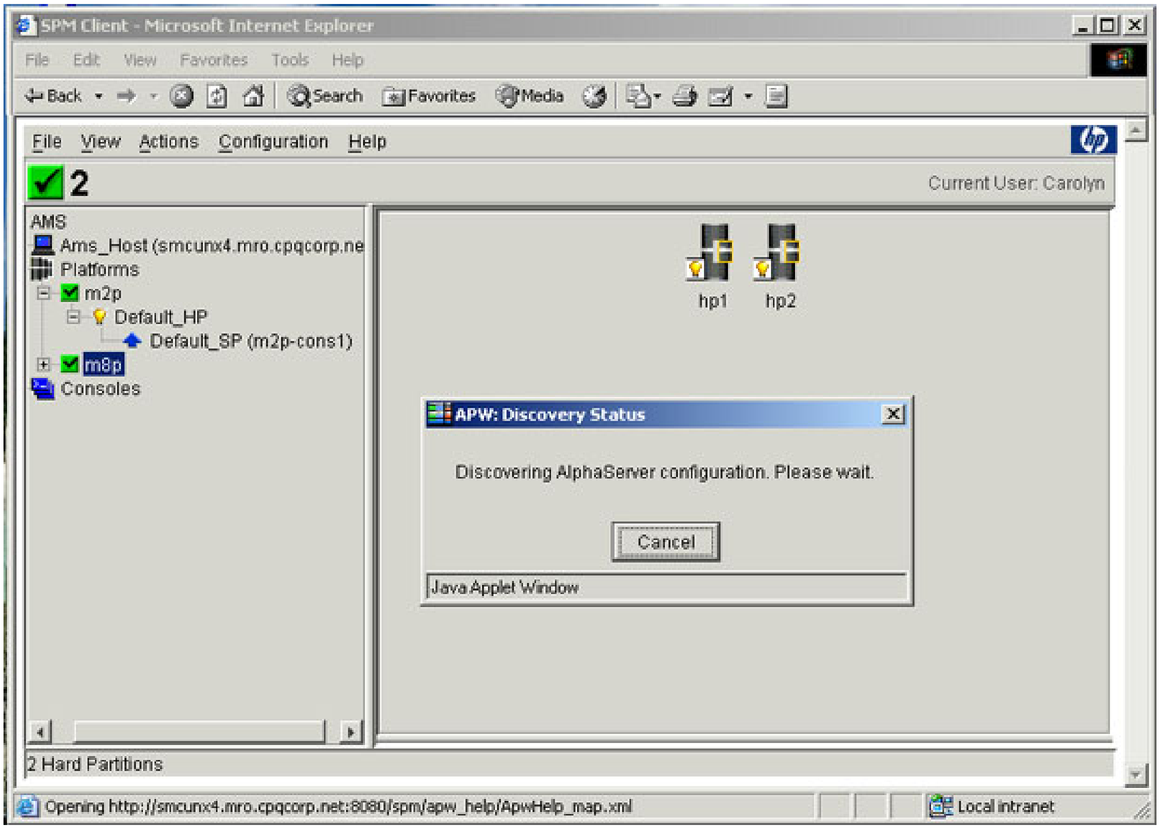
Figure 4-3 Start the APW



4.2 Discovery Phase

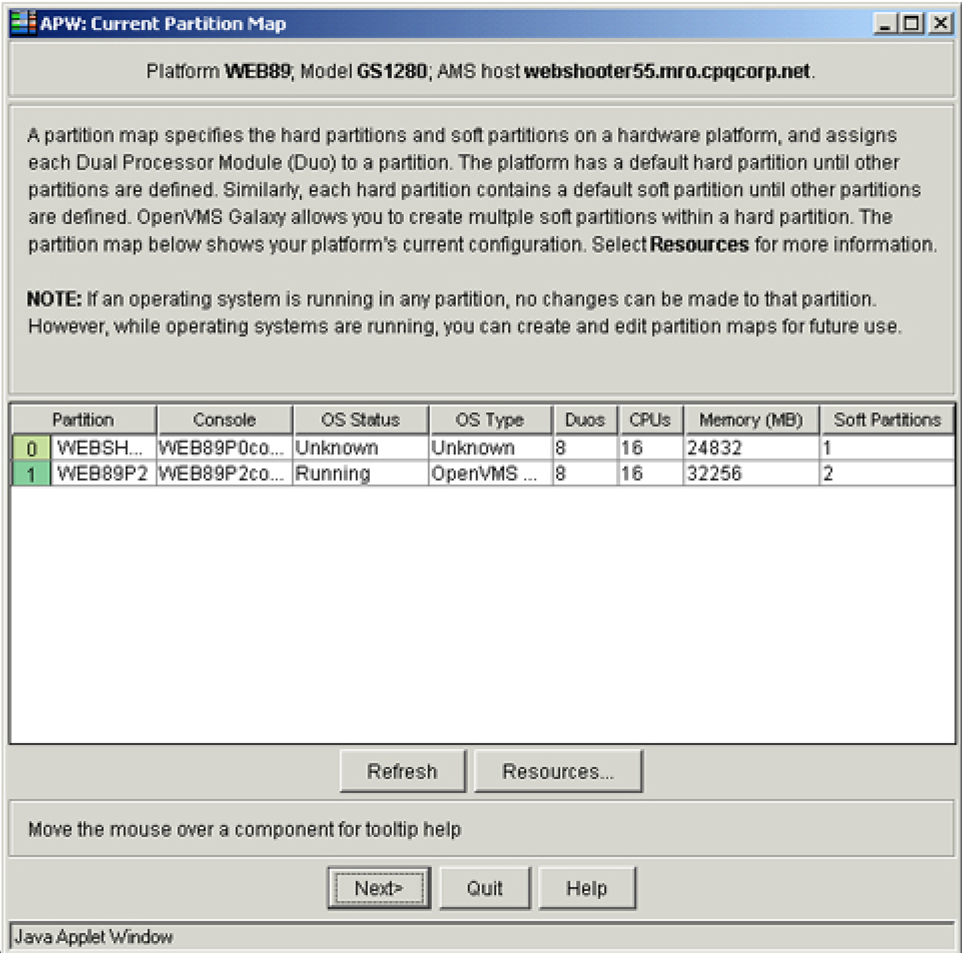
1 The APW "Discovery" window displays as a small screen on top of the SPM screen. The discovery phase may last as long as 5 minutes on a large machine.

Figure 4-4 The APW "Discovery" Window Displays



2 When the discovery phase complete, the APW displays the current partition configuration. The example in Figure 4-5 shows a 32P system divided into two 16P hard partitions.

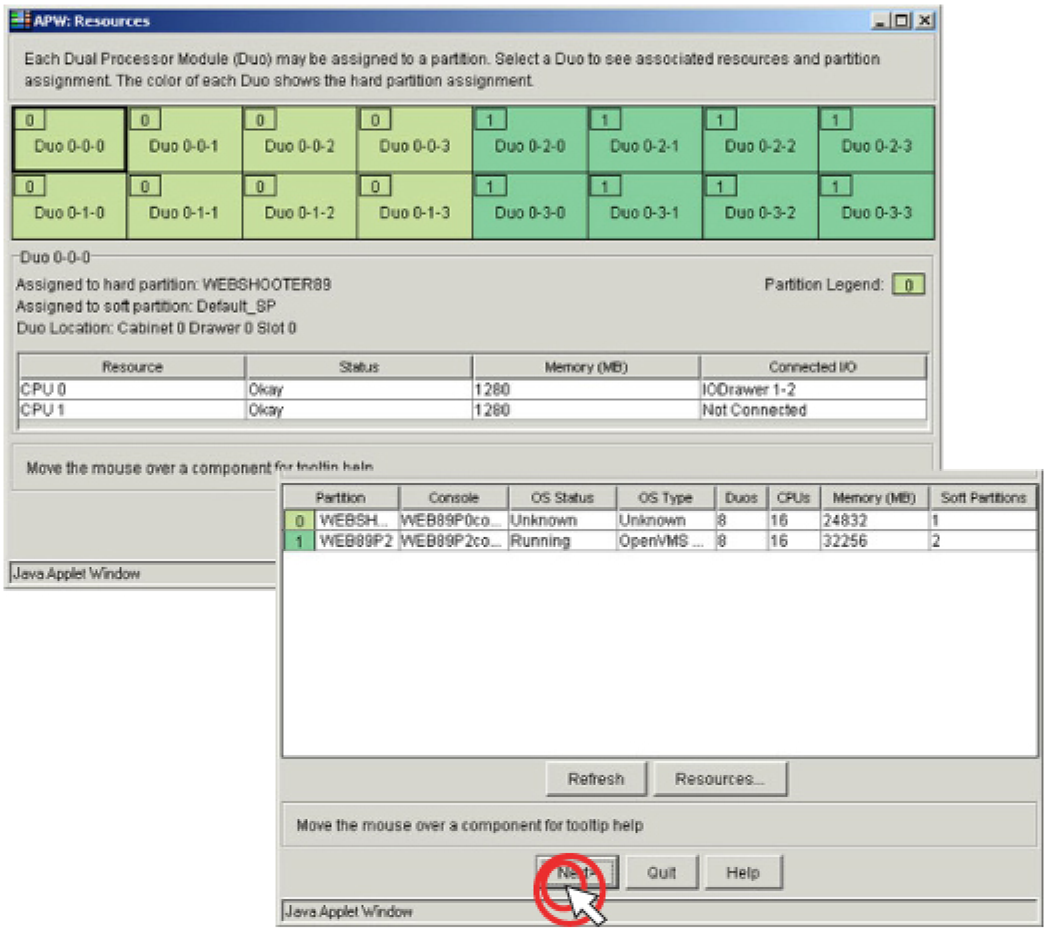
Figure 4-5 APW Current Partition Map Display



4.3 Display Resources, Click "Next"

Near the bottom of the APW wizard startup screen shown above is a button called "Resources..." This screen displays more detail on the partition status of the platform you are working with, as shown in Figure 4-6.

Figure 4-6 APW Resources Screen



Keep this screen on your monitor, outside of the Wizard screens as you are working. It will display the partition map you are creating.

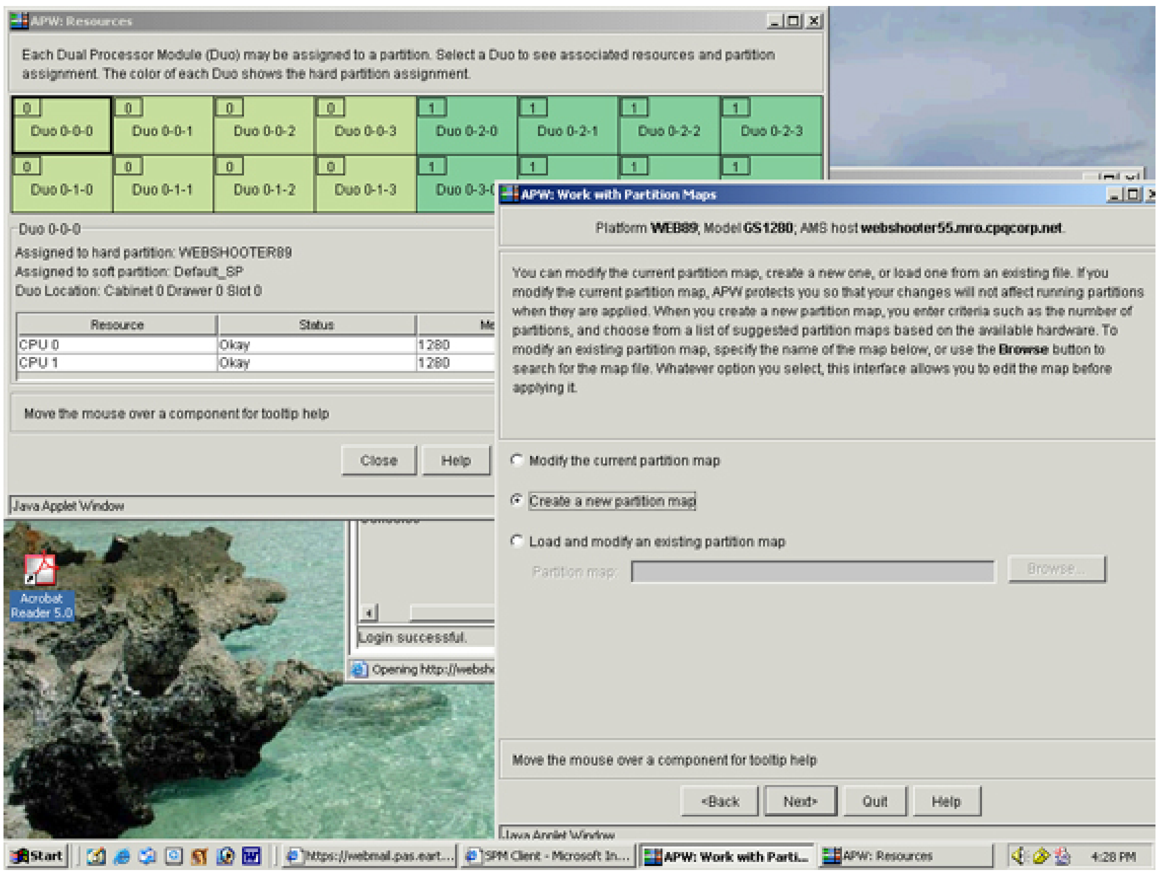
Click "Next" to proceed to the next wizard screen, described in the next section. (You can always select "Quit" to exit the APW without saving or committing to a map, and "Help" displays helpful information.)

4.4 Select Partition Map Option

When you have the Resources screen where you want it, click one of the three choices for creating image maps (see Figure 4-7), and then click "Next" to go to the next wizard screen. The three choices are:

- **Modify the current partition map.** You will work with the existing partition map shown in the resources window.
- **Create a new partition map.** You want to create a new partition map.
- **Load and modify an existing partition map.** Load a previously stored partition map and work with it.

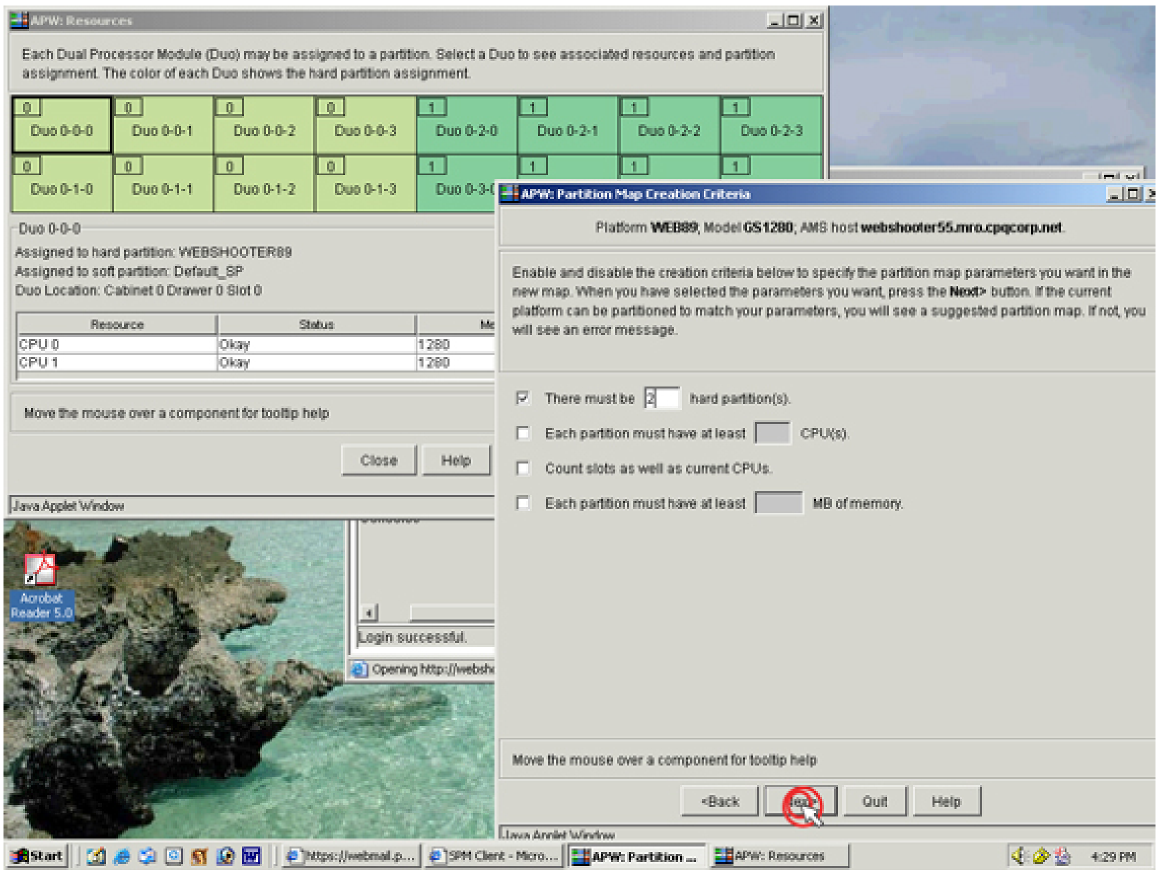
Figure 4-7 Partition Map Creation Options



4.5 Enter New Map Parameters (If Necessary)

If you selected "Create a new partition map", the APW will next display the screen shown in . Enter as many parameters as you wish, and click “Next”. The APW then creates a map, which you can accept or modify using the next screen.

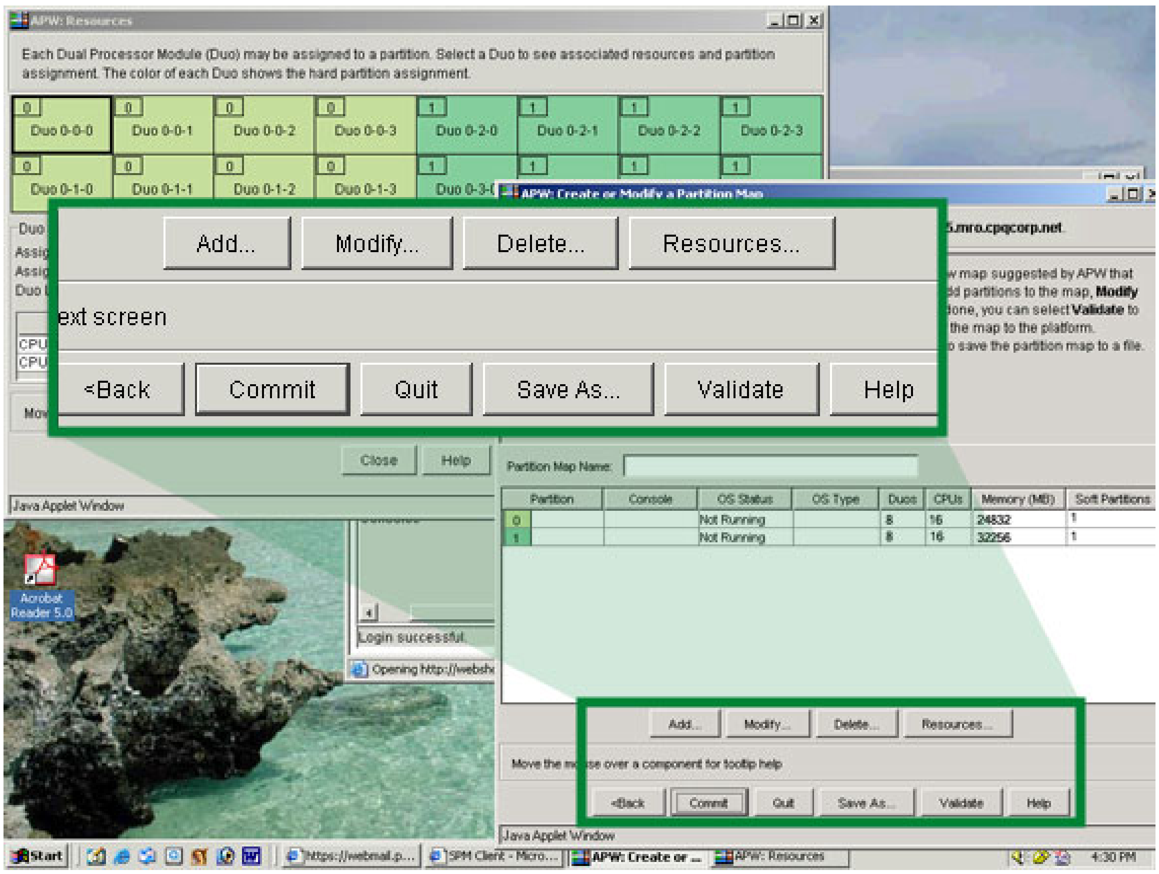
Figure 4-8 New Partition Map Criteria



4.5 Create or Modify a Partition Map

Whichever choice you made in Section 4.4, you are now presented with a screen showing a map, with a series of buttons at the bottom. (The "Resources" screen will be updated to show the appropriate map.) Following sections describe the options available through the “Add…”, “Modify…”, and “Delete…” buttons.

Figure 4-9 Create or Modify Partition Map Screen

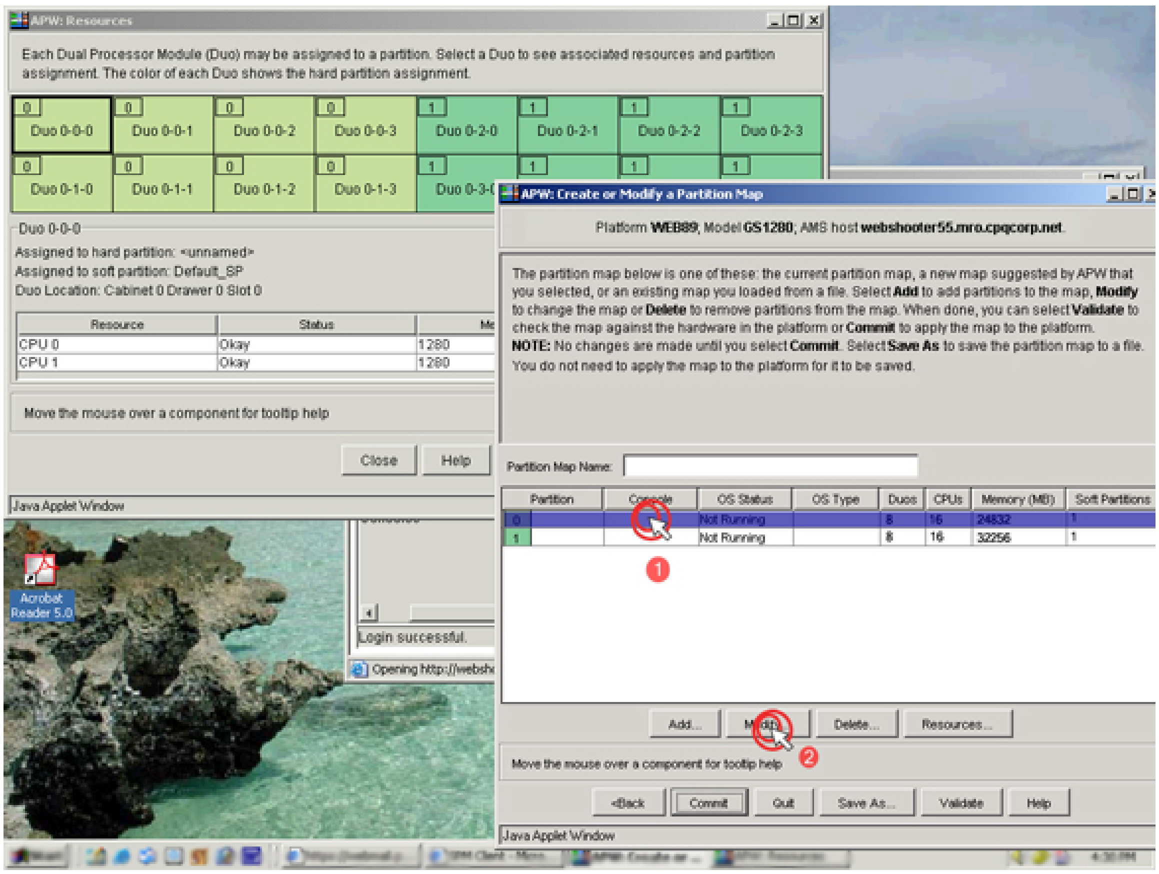


4.5.1 Modify the Map

Select the "Modify" button if you want to change the map shown (either a new map created by the APW, or the map defined for the platform at the moment, or a saved map you have selected.)
The steps are:

1. Select the hard partition you wish to change by left-clicking on the partition, and then clicking the "Modify" button.

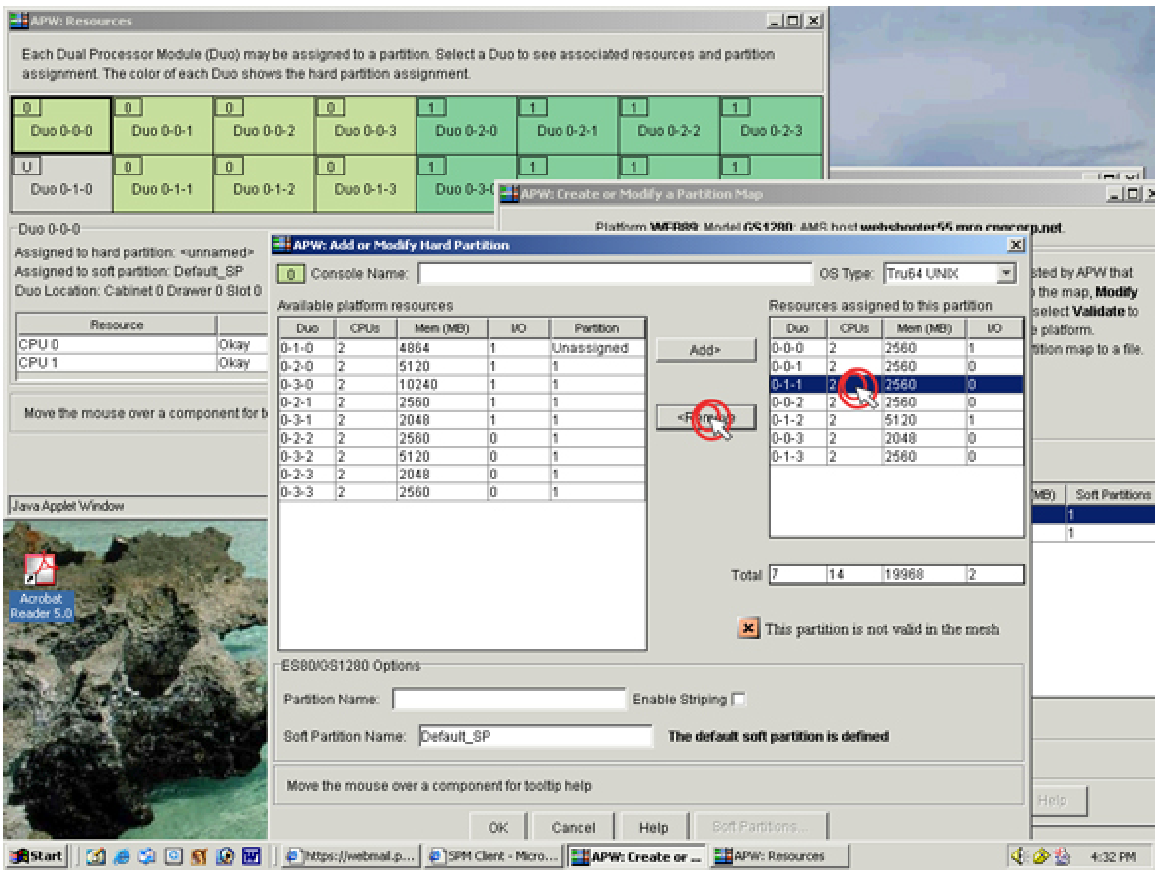
Figure 4-10 Select Hard Partition, click "Modify"



2 You are now presented with a table. The right column of the table lists the duos in the partition that you selected to modify. The left column shows all the other duos in the platform. You can add duos from the available resources to the partition you are modifying, or remove duos from the partition you are modifying to put them in the available resources side of the table.

The screen shown in Figure 4-11 shows that duo 0-1-0 has already been removed from the partition. Note in the "Resources" screen that this duo is gray and has "U" for Unassigned as an identifier. The "clicks" show the process of removing duo 0-1-1 from the partition.

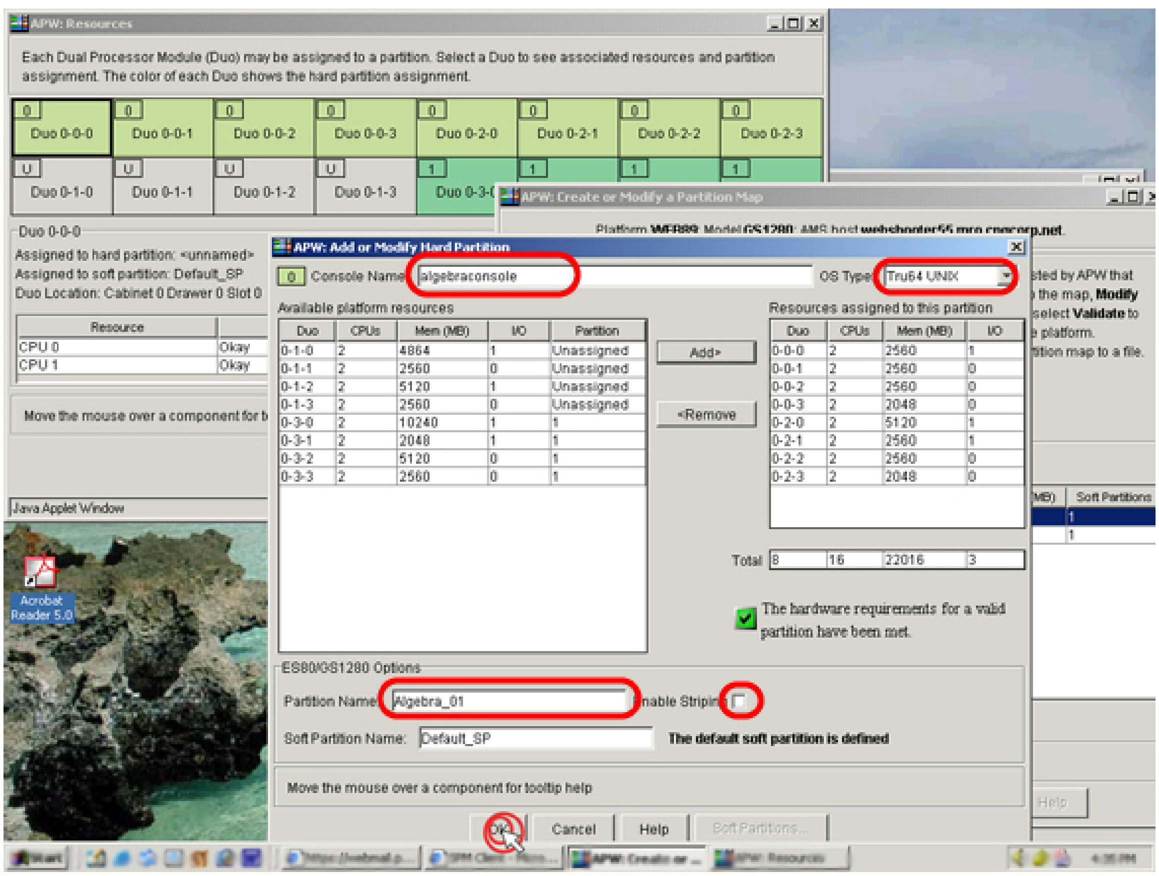
Figure 4-11 Moving Resources to and From a Hard Partition



3 This screen shows the user finishing with the modification of the partition. Note that you are asked for a "console name", the type of the operating system, the partition name, and whether or not to use striping.

- The *console name* must be supplied; it appears on the left pane of the SPM window and allows you to connect directly to the Server Management CLI for the hard partition you are creating. Click "OK" to indicate you are through editing this partition and close the window.
- The *OS Type* will show in APW displays as the operating system associated with this hard partition. It is **not** passed on to the SRM console for the partition. Installing the operating system sets environment variables such as os_type, bootdef_dev, and so forth. The OS Type affects further processing only if you select OpenVMS Galaxy from the choices listed; if you do, you can then create soft partitions within the hard partition.

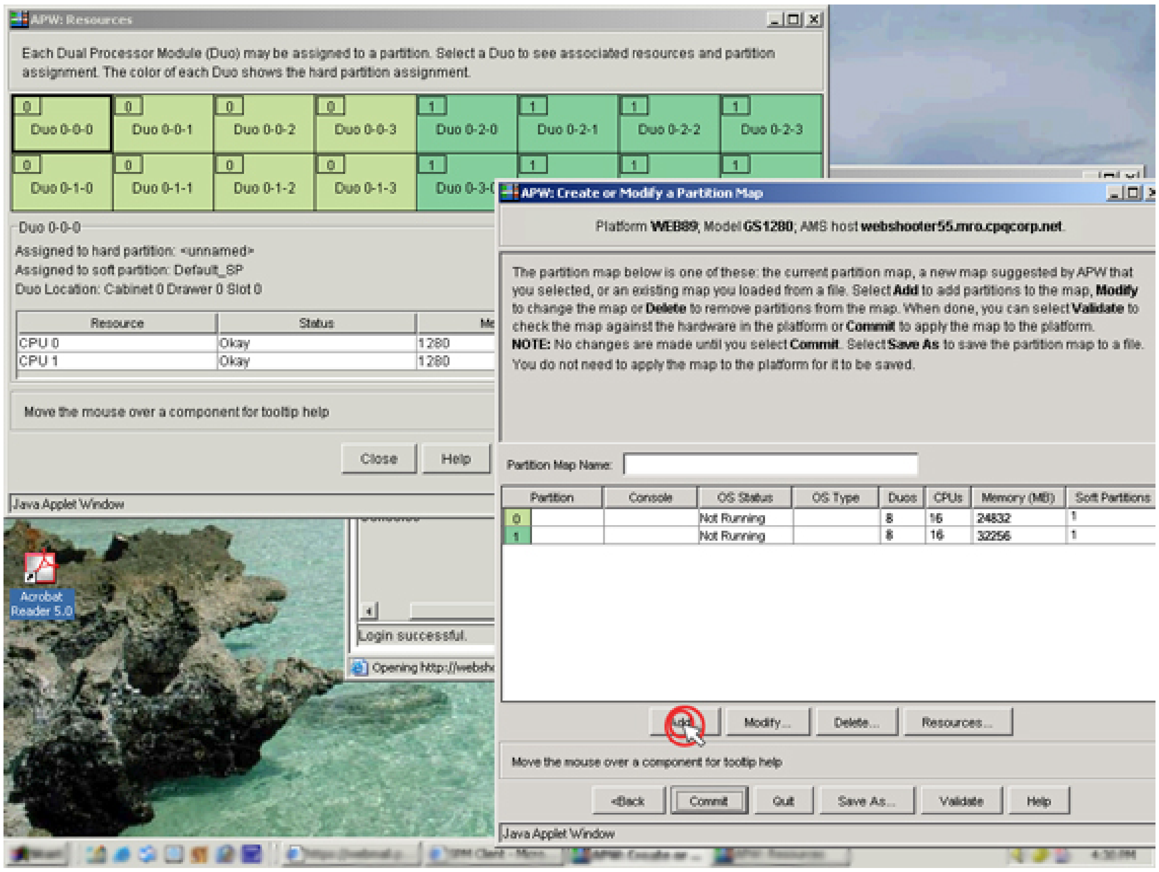
Figure 4-12 Finish the Hard Partition Definition



4.5.2 Add a Hard Partition

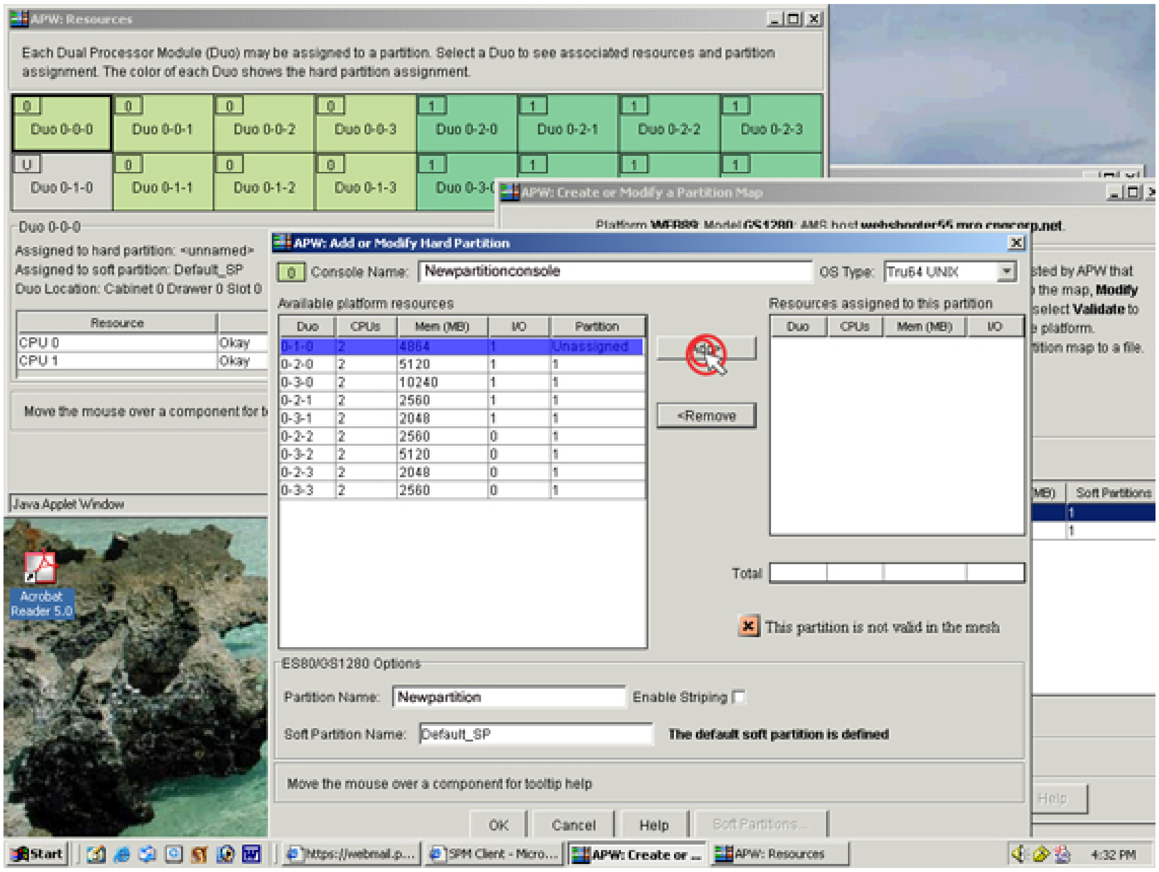
1 Select the "Add" button from the "Create or Modify a Partition" menu.

Figure 4-13 Select “Add” to Add a Hard Partition



2 Type in the new partition's console name, select the operating system, and type in the new partition name. Use the "Add" and "Remove" buttons to add resources to the new hard partition. Click “OK” when finished.

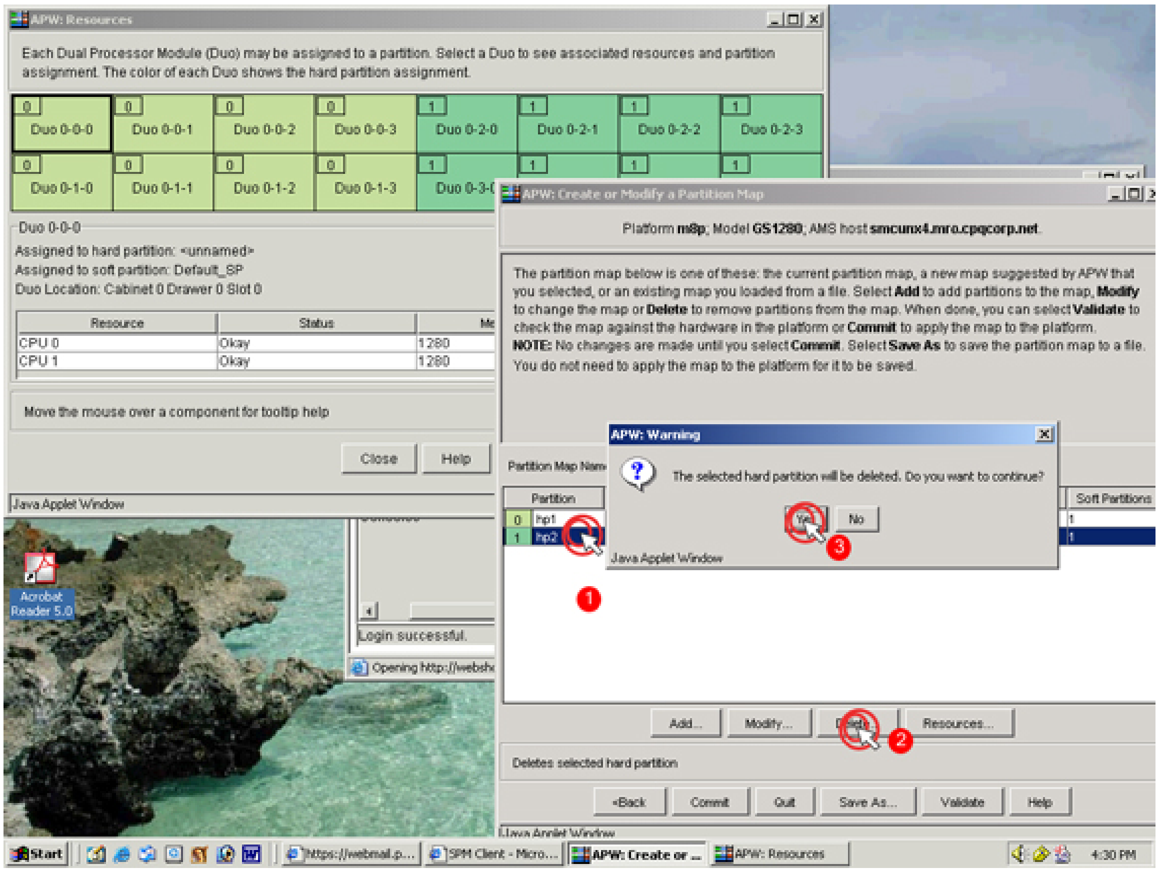
Figure 4-14 Name the New Partition and Its OS and Move Resources to Define



4.5.3 Delete a Hard Partition

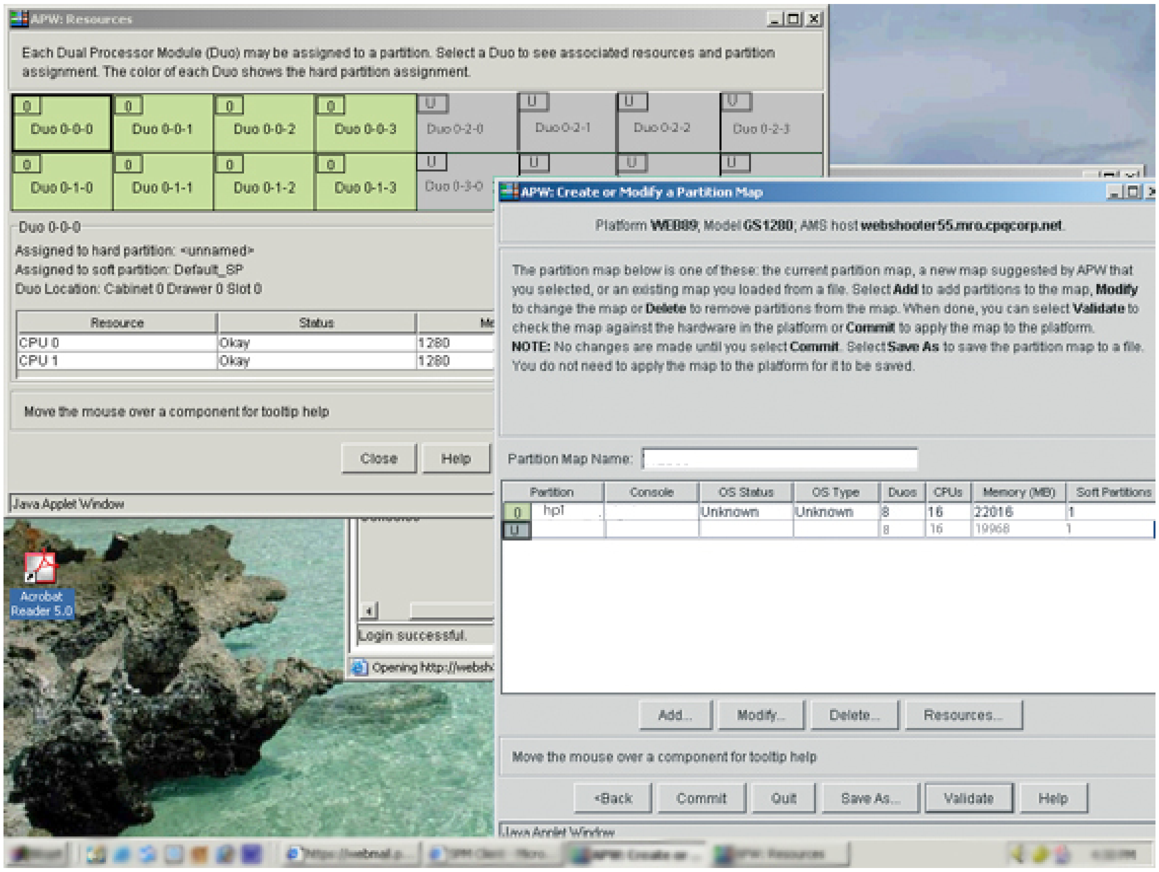
1 Select the hard partition to be deleted, and click "Delete". The APW will present a screen asking if you really want to delete this partition from the map. Click as appropriate.

Figure 4-15 Delete a Hard Partition



2 The APW will show the partition deleted in both the "Resources" screen and the partition map.

Figure 4-16 Hard Partition Delete Completed



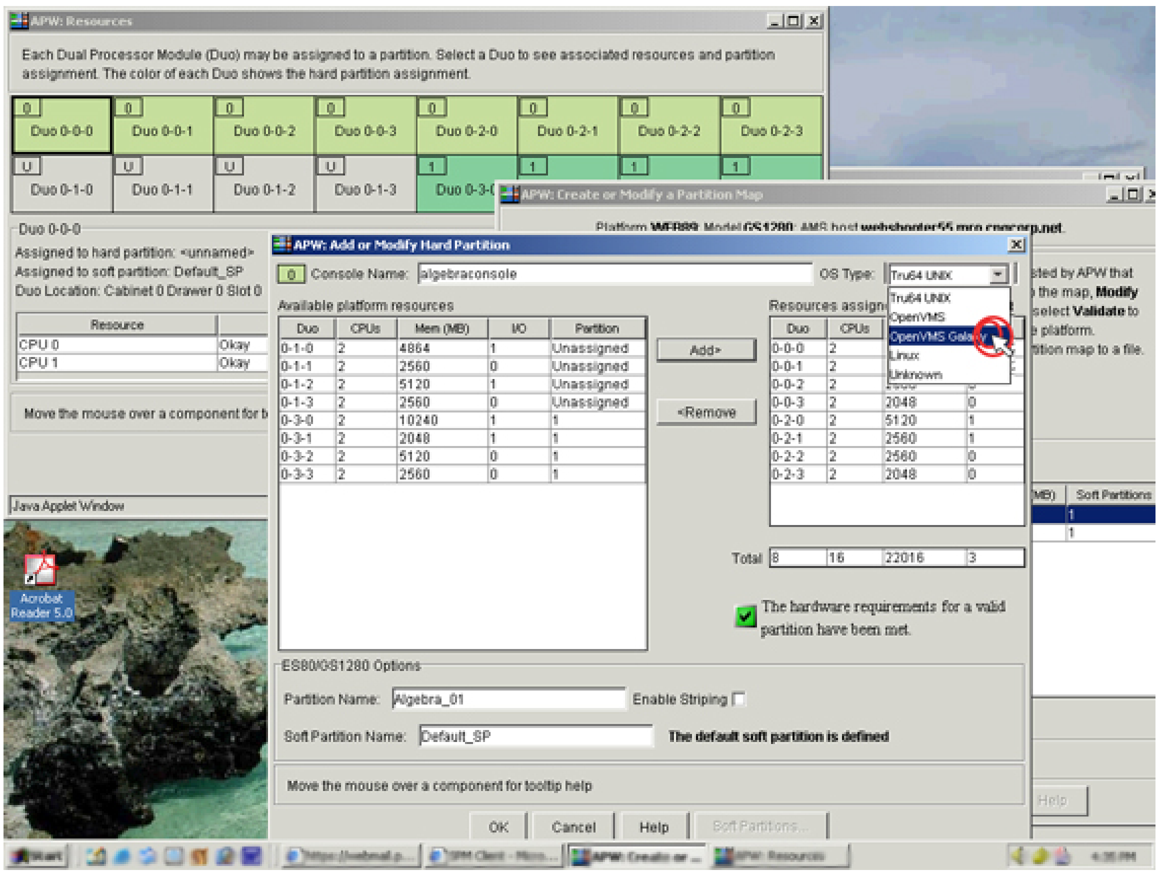
4.5.4 Create Soft Partitions

With the APW, you can create soft partitions (subpartitions) within a hard partition only when you select "OpenVMS Galaxy" as the operating system from within the "Add or Modify Hard Partition" screen. Then, the "Soft Partitions" Button" becomes active.

You can create soft partitions as you are working with a hard partition, or after you have created all hard partitions. (You can get to the "Add or Modify Hard Partition" screen after a hard partition has been created from the "Create or Modify a Partition Map" screen.) Here, we go back to the stage of working with the hard partition "Algebra 101", and decide to create two subpartitions in it, as we did for the AMU. The steps are:

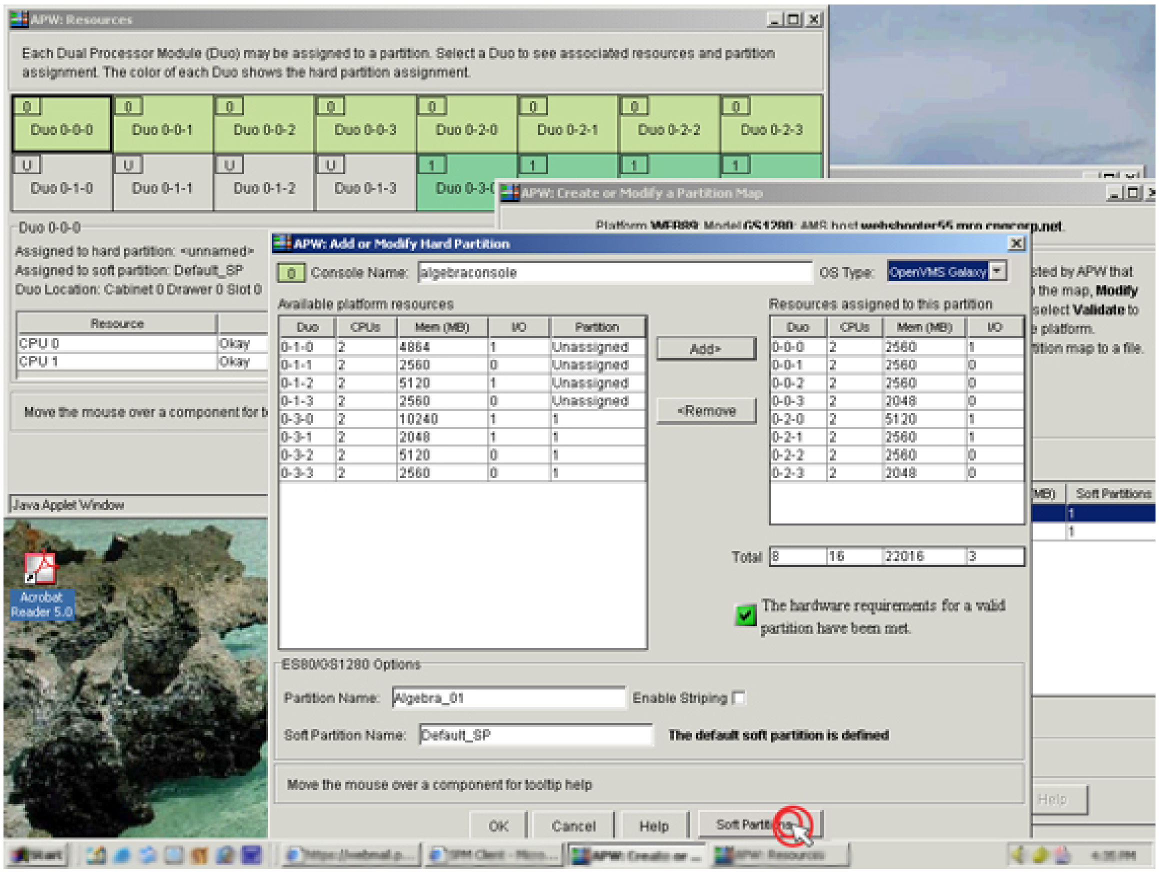
1 Select "OpenVMS Galaxy" as the Operating System.

Figure 4-17 Select OpenVMS Galaxy as the Operating System



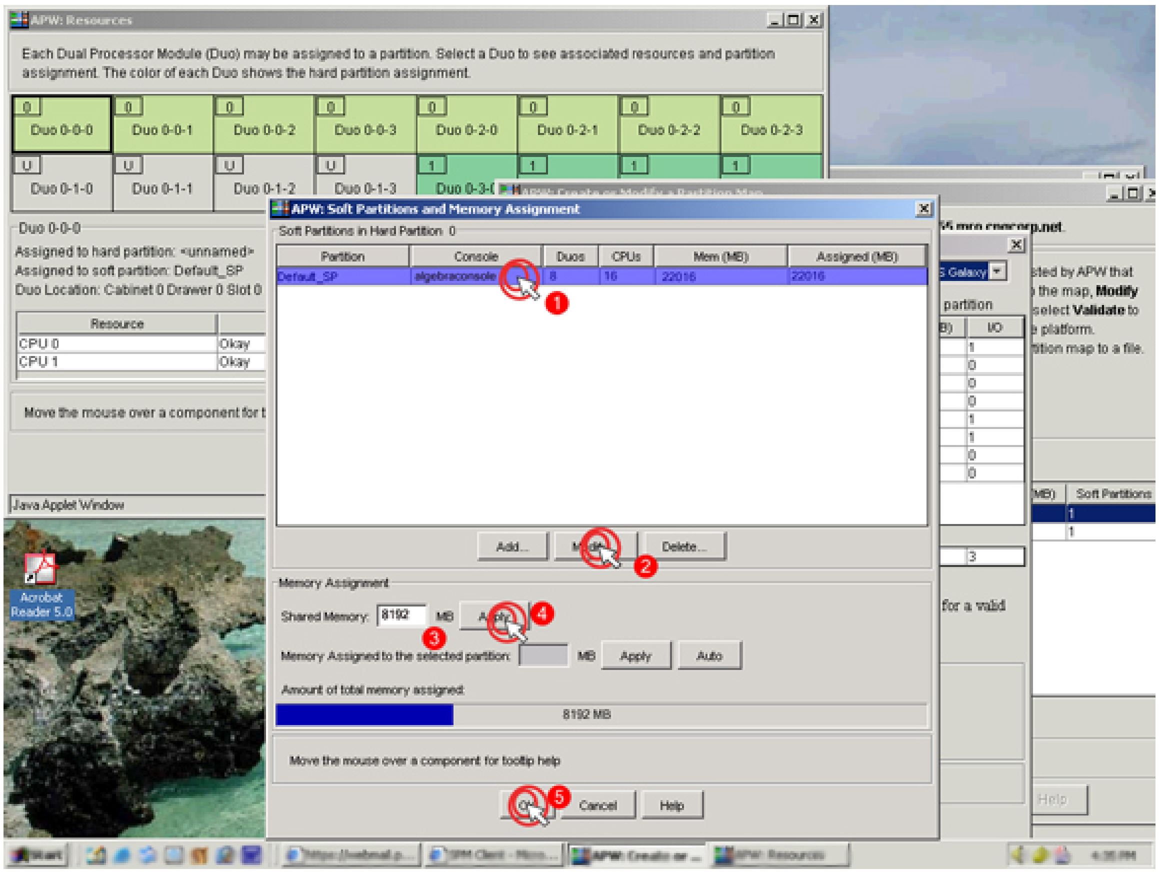
2 The "Soft Partitions..." button is no longer grayed out. Click on it to display the "Soft Partitions and Memory Assignment" screen.

Figure 4-18 Click "Soft Partitions…" Button



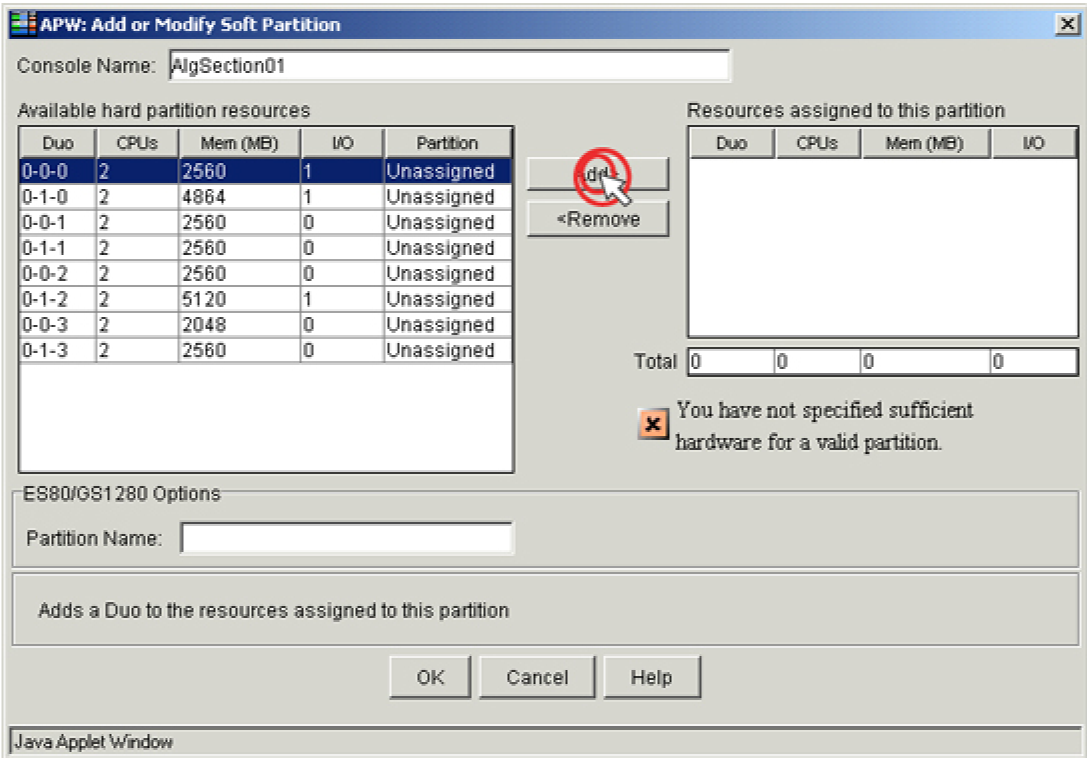
3	Click on the Default_SP to select it and select "Modify". Enter the amount of community memory to be shared by the soft partitions you want to create, and click "Apply". The bar shades to show the amount of memory assigned. You can also specify a particular amount of memory to assign to the soft partition, but it is better just to leave it at "Auto", the default. Click "OK" to display the "Add or Modify Soft Partitions" screen.

Figure 4-19 Soft Partitions and Memory Assignment Screen



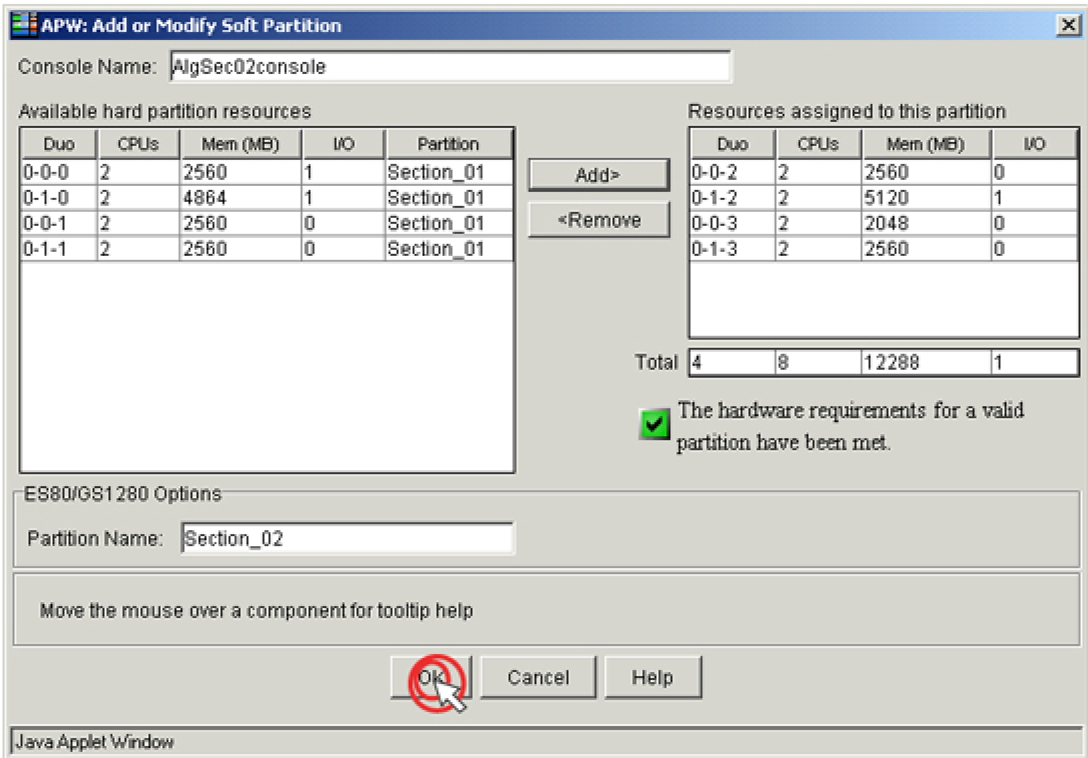
4 This screen is similar to that for hard partitions, except you are only given the resources in the hard partition you selected to work with. Again, move resources using "Add" and "Remove". Click “OK” when finished and select “Soft Partitions” again to define additional soft partitions.

Figure 4-20 Add or Modify Soft Partition Screen



5 This screen shows that a soft partition called "Section 01" has already been created, and the remaining resources moved for assignment to a soft partition named Section_02, with a console name of AlgSec02console. Click "OK" and this screen closes. Click "OK" on the "Add or Modify Hard Partition" screen, and you will be back to the "Create or Modify Partition Map" screen.

Figure 4-21 Creating Soft Partitions

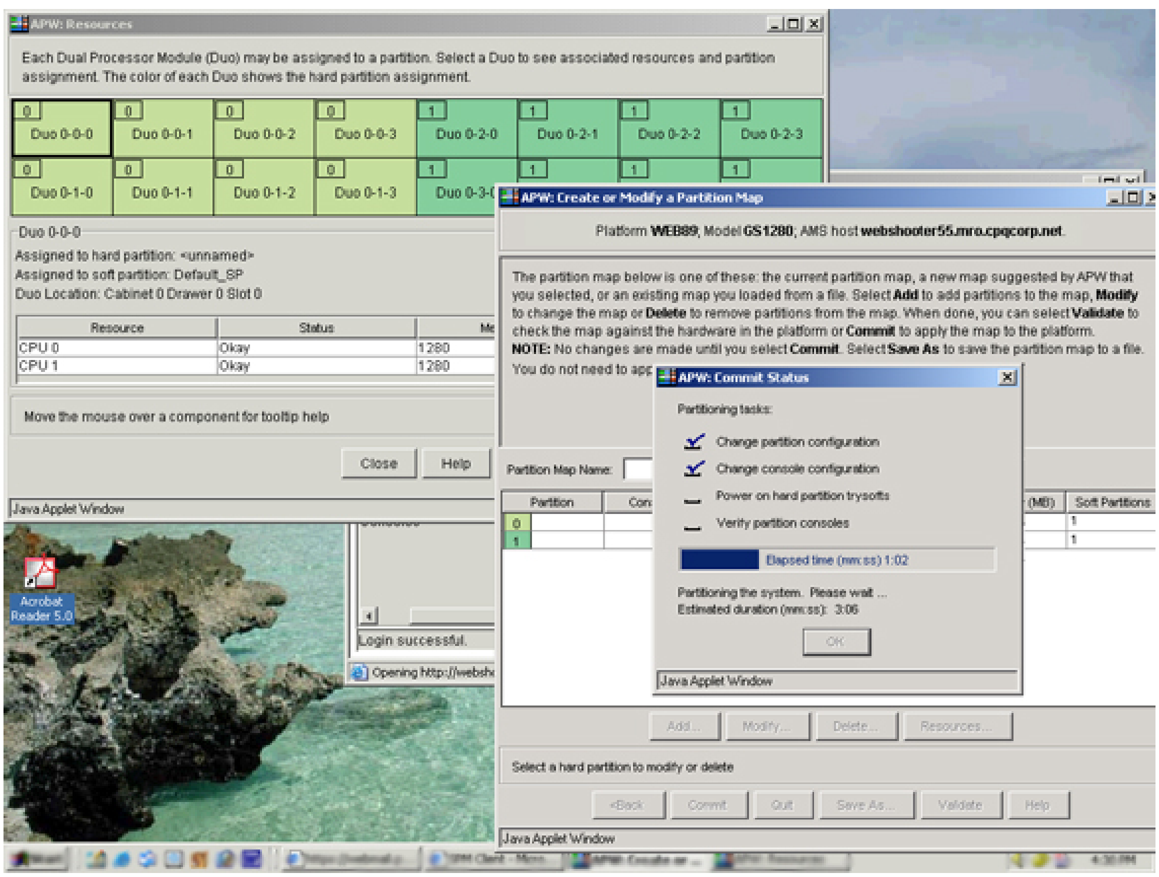


4.6 Finishing the APW Process

Once you have finished creating the partition map you want, you have several options. The bottom row of buttons on the "Create or Modify a Partition Map" screen are:

- **Back.** Go back to the "Work with Partition Maps" screen. You will lose any settings you have made up to this point if you do this. However, if you want to start over, the APW allows you to do this.
- **Commit.** This button resets the entire platform and applies the current map, saving the parameters to the MBM. Of course, the operating systems should be shut down in each of the current partitions of the platform. The screen in Figure 4-22 shows this process, which has a mini-window that displays several steps that are checked off as they are completed, as well as a "completion" bar at the bottom, showing the progress of the commit process.
- **Quit.** Destroys the current map and exits the APW utility.
- **Save As...** Saves the current map as a file which can later be recalled and applied, modified, etc.
- **Validate.** Checks the current map to make sure that it has valid partition definitions.
- **Help.** Displays the online help for the utility.

Figure 4-22 APW Commit Screen



Chapter 5
Using the CLI to Create Partitions

The sequence of Server Management Command Line Interface (CLI) commands used to create and work with partitions is perhaps best shown by example.

5.1 Accessing the CLI

You can access the Server Management CLI in the following ways:

- From the SPM
- From the AMU
- From a remote telnet session
- From the MBM Backplane

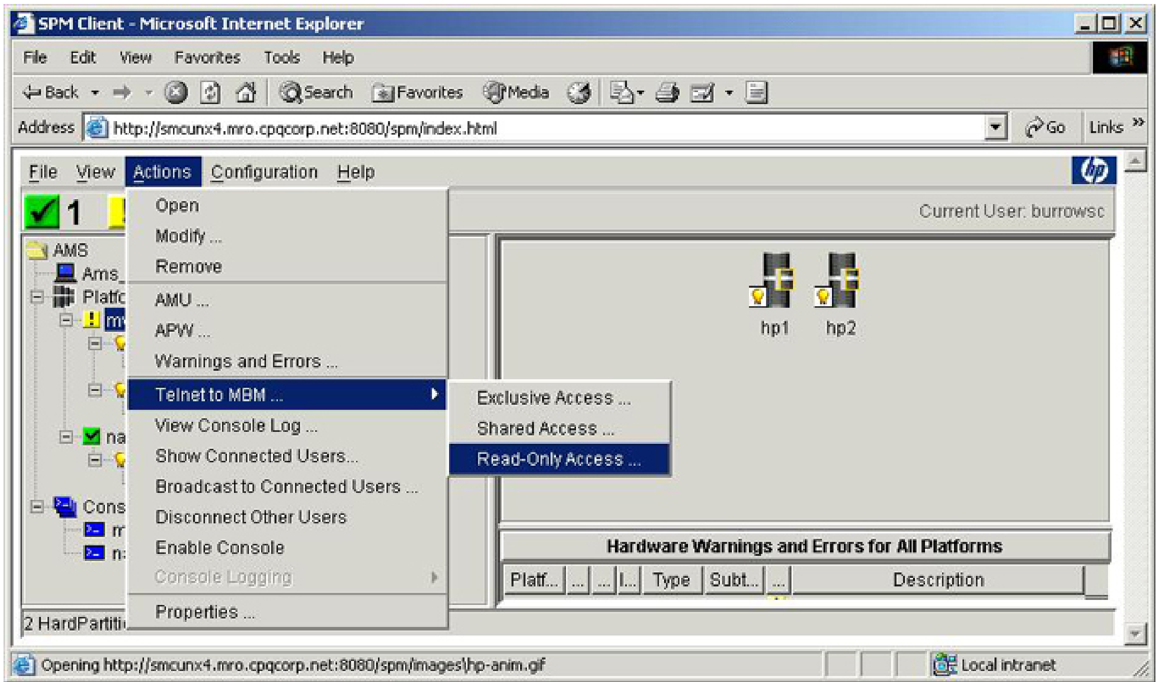
5.1.1 Access the CLI from the SPM

If your site has an AlphaServer Management Station (AMS) running the Server Platform Management (SPM) software, and you have administrator or operator access to it, you can use the SPM window to telnet into the Server Management CLI of a particular platform. You can do this in three different ways.

From the SPM "Actions" Menu

Once you have logged in to the SPM as a valid user, you can left-click on a platform in either the tree view or the right-hand view. Once you select a platform, you can left-click on the "Actions" menu item to telnet to the Server Management CLI of the platform you selected.

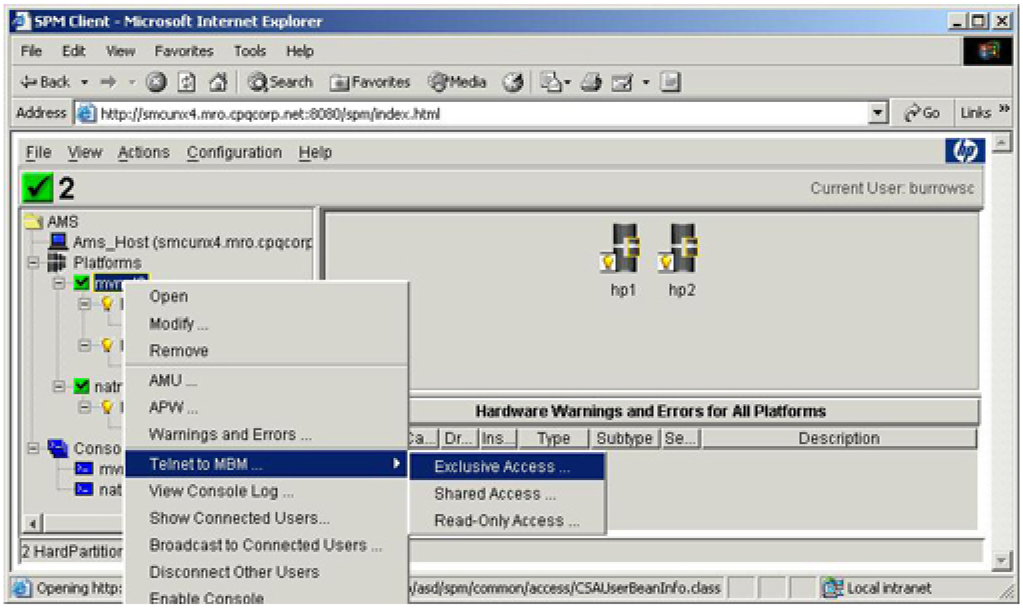
Figure 5-1 From the SPM “Actions” Menu



From the SPM Tree View

Once you have logged in as a valid SPM user, you can directly right-click on a platform in the tree view to display a pulldown menu from which you can telnet to the Server Management CLI for that system.

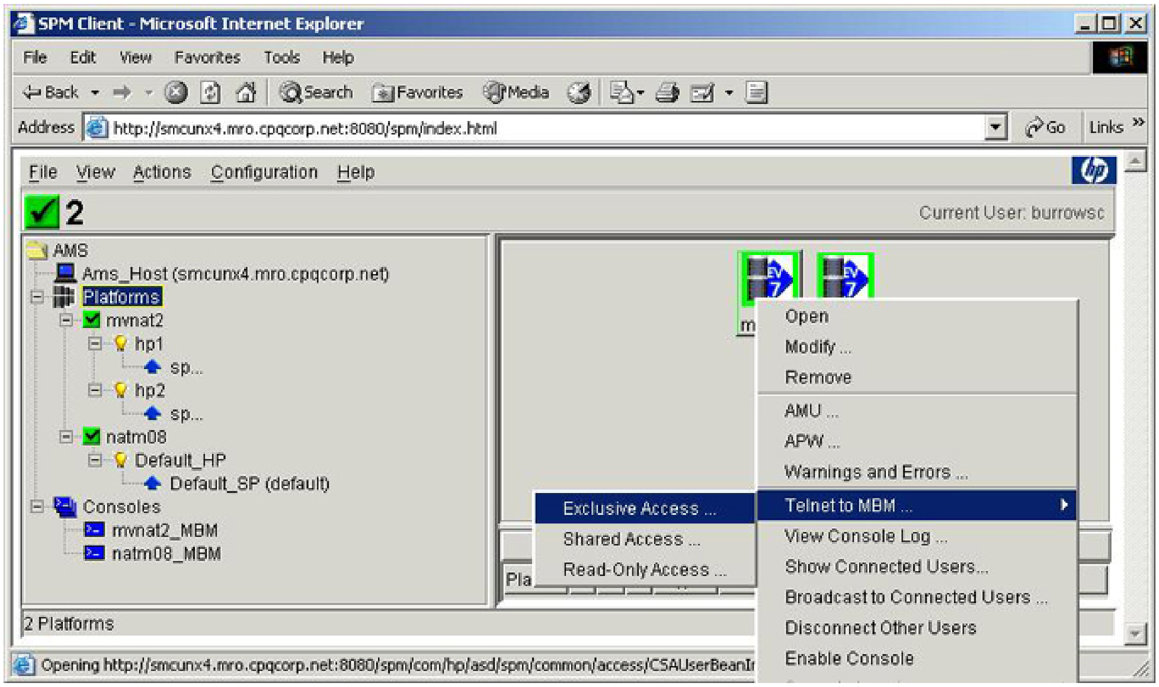
Figure 5-2 From the SPM Tree View



From the SPM Platform View

From the right pane of the SPM window, you can right-click on a platform to display a pulldown menu that allows you to telnet to the Server Management CLI of that particular platform.

Figure 5-3 From the SPM Platform View

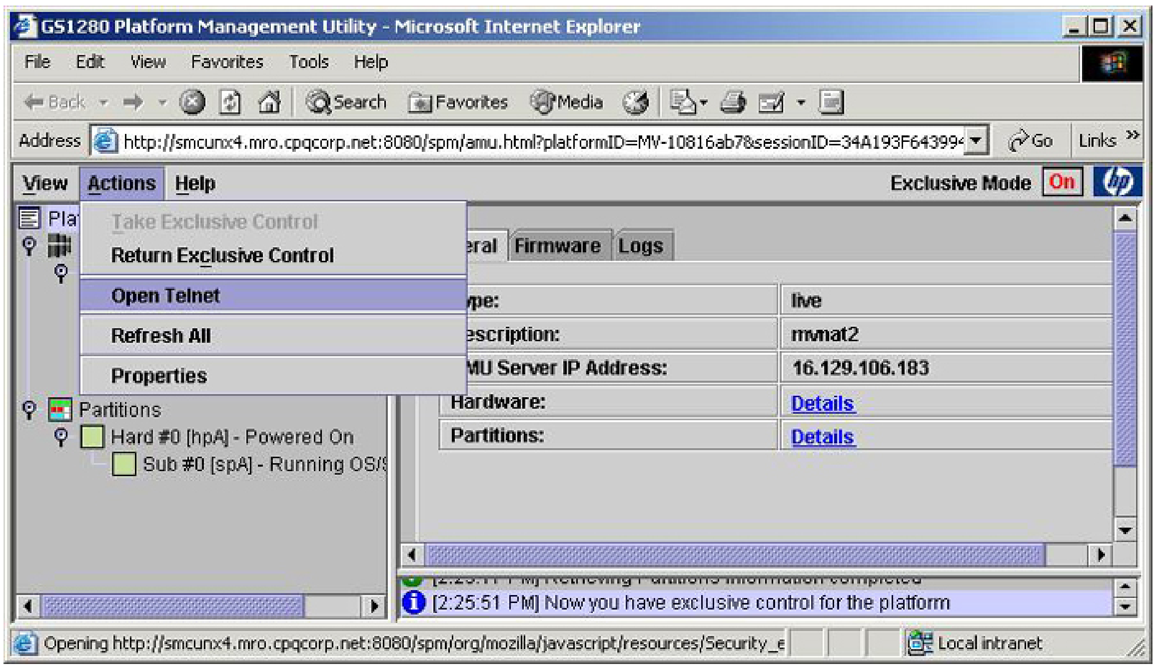


5.1.2 Access the CLI from the AMU

You can access the Server Management CLI interface from the AlphaServer Management Utility interface as follows:

1. Click on the "Platform" icon in the tree view. (Remember that the AMU is only attached to a single platform at a time.)
2. Pull down the "Actions" menu and select "Open Telnet" from the options available, as shown below.

Figure 5-4 Accessing the CLI from the AMU



5.1.3 Access the CLI from a PC Telnet session

You can access the Server Management CLI from a PC attached to the Corporate LAN or the Multi-Server LAN by installing KEAterm or similar software on the PC.

Instructions for installing this software on your PC is given in the *ES47/ES80/GS1280 Service Manual.*

5.1.4 Console Access

The telnet address to access the platform's Command Line Interface (CLI) is 10.253.0.1.

For example, you can telnet to a platform from the SPM interface. Right-click on the platform icon in the tree view and select telnet from the menu. When the console screen displays, type

    telnet 10.253.0.1

and you will receive the MBM> prompt and can issue CLI commands.

When partitions have been created, you can telnet to a partition within the platform from the SPM, using the same address plus a port number. For example:

telnet 10.253.0.1 323

The first partition's default port is 323, the second partition's port is 324, the third, 325, and so forth.

Once partitions have been created, the port number can be displayed by the CLI **show partition** command.

If you are using the SPM, you can telnet to a subpartition by right-clicking on the subpartition in the right-hand display and selecting Telnet. If you are using the AMU, you can telnet to a partition or subpartition by selecting the partition in the AMU tree view, right clicking on it, and selecting telnet from the drop-down menu.

5.2 General Command Sequence

The detailed format of CLI commands that are used to create and work with partitions are given in Appendix A.

The following list shows a typical sequence of steps in creating partitions:

1. Do a **show partition** command before making any other requests, to see if any current partition definitions exist on the platform. The example following lists a show partition command on an unpartitioned 8P platform.
2. Execute a **power off -all** command.
3. Delete the current partition table, if any, with a **del part -all** command.
4. Create a hard partition (or subpartition). If you are creating a 16P hard partition that will eventually contain soft subpartitions, an example command would be:

   create part -hp lime 255 soft

   where **lime** is the name of the hard partition, **16** is the maximum number of processors to be contained in the hard partition, and **soft** declares that, if any subpartitions are created in this hard partition, they will be soft subpartitions.
5. Assign components to the partition or subpartition. An example command would be:

   assign comp CPU0 -hp lime

   In this case, an individual CPU is being assigned to the hard partition named **lime**.

   The components that you can assign using the **assign component** command are: CPUs, IOPs (I/O drawers), and system building blocks (SBBs) which are 8P drawers or IOP drawers.

   Repeat this step as necessary until all the components comprising the partition have been allocated.
6. Repeat steps 4 and 5 until all partitions and subpartitions have been defined.
7. Power on all partitions.

   power on -all

Note: The **show grid** command can be used to display partition construction in a grid using the torus coordinates used by other software and firmware in partitioning. See the show grid command description for an example.

5.3 Example 1: Show Partition Command

The example below lists a show partition command on a platform.

Equation 5-1 Show Partition Command

```
MBM> sho part
Hard Partition : HP Name = Default_HP, HP No.= 0, SP count = 2
Attributes     : max CPUs = 255, SP type = soft, Non-stripe
Physical Memory: 12288MB (12.000GB)
        Community Memory: 0MB (0.000GB)
        Sub Partition: HP Name = Default_HP, HP No.= 0
                       SP Name = Free_Pool, SP No.= 255
                       State = Not Running
                Free Memory: 0MB (0.000GB)
                CPUs: None
                IOPs: None
        Sub Partition: HP Name = Default_HP, HP No.= 0
                       SP Name = Default_SP, SP No.= 0
                       State = Not Running, Telnet port = 323
                Assigned Memory: unspecified
                CPUs:
                    Cab  Drw  CPU   (NS,EW)   PID    Type
                     0    0    0    ( 0,0 )    0     Primary
                     0    0    2    ( 0,1 )    2     Non-primary
                     0    0    4    ( 0,2 )    4     Filler
                     0    0    6    ( 0,3 )    6     Filler
                     0    0    1    ( 1,0 )    1     Non-primary
                     0    0    3    ( 1,1 )    3     Non-primary
                     0    0    5    ( 1,2 )    5     Filler
                     0    0    7    ( 1,3 )    7     Filler
                IOPs:
                           SBB                            PCI Drawer
                    Cab  Drw  IOP   (NS,EW)             Cab  Drw  IOR
                     0    0    0    ( 0,0 ) ---------  0    1    0
                     0    0    2    ( 0,1 )
                     0    0    4    ( 0,2 )
                     0    0    6    ( 0,3 )
                     0    0    1    ( 1,0 )
                     0    0    3    ( 1,1 )
                     0    0    5    ( 1,2 )
                     0    0    7    ( 1,3 )
```

5.4 Example 2: Two Hard Partitions on a 16P Platform

The following sequence defines two hard partitions, each containing a single soft partition. Each partition consists of an 8P drawer and its attached memory and I/O boxes.

Example 5-2 Creating Two Hard Partitions on a 16P Platform

`MBM> p off –all`	*Power off everything*
`MBM> del part –all`	*Clear the current partition table.*
`MBM> create part –hp A_lime_VMS 255 soft` `MBM> create part –hp B_green_UNIX 255 soft`	*Create two hard partitions. The number is the maximum number of CPUs.*
`MBM> assign comp SBB –hp A_lime_VMS`	*Put the first box of 8 CPUs (the default drawer 0) with all its attached PCI drawers in the VMS partition.*
`MBM> assign comp –d 1 SBB –hp B_green_UNIX`	*Put the other 8 CPU box (the drawer 1) with all its I/O in the UNIX partition.*
`MBM> sho part –all`	*Show the result.*
`MBM> p on –all`	*Power on all partitions.*

`telnet <natbox address> 323`
`P00>>> sho boot*`
`P00>>> set bootdef_dev …`
`P00>>> set boot_osflags 0,0`
`P00>>> set os_type VMS`
`P00>>> b`
	Once the poweron tests are finished, connect to the first partition and define the boot device, boot flags, and operating system type. Then boot the partition.

`telnet <natbox address> 324`
`P00>>> sho boot*`
`P00>>> set bootdef_dev …`
`P00>>> set boot_osflags a`
`P00>>> set os_type UNIX`
`P00>>> b`
	Connect to the second partition, and perform the same operations as for the first.

5.5 Example 3: Three Partitions on an 8P Platform

In general, the **power off** command turns off the power to a platform, component, or hard partition. This command powers off a non-partitioned platform if no arguments are given, or it powers off a specific hard partition or component.

The **power on** command turns on the power to a platform, component, or hard partition. If no arguments are given, this command powers on a platform with one hard partition.

Example 5-3 Creating Three Partitions on an 8P Platform

`MBM> p off –all`	*Power off all partitions.*
`MBM> del part –all`	*Clear the current partition table*

`MBM> create part –hp A_lime_VMS 255 soft`
`MBM> create part –hp B_green_UNIX 255 soft`
`MBM> create part –hp C_blue_LINUX 255 soft`
	Create three hard partitions. The number is the maximum number of CPUs.

`MBM> assign comp CPU0 –hp A_lime_VMS`
`MBM> assign comp CPU1 –hp A_lime_VMS`
`MBM> assign comp CPU2 –hp A_lime_VMS`
`MBM> assign comp CPU3 –hp A_lime_VMS`
	Put four CPUs with all their attached PCI drawers into the VMS partition.

`MBM> assign comp CPU4 –hp B_green_UNIX` `MBM> assign comp CPU5 –hp B_green_UNIX`	*Put two CPUs with all their I/O into the UNIX partition.*
`MBM> assign comp CPU6 –hp C_blue_LINUX` `MBM> assign comp CPU7 –hp C_blue_LINUX`	*Put two CPUs with all their I/O into the LINUX partition.*
`MBM> sho part –all`	*Show the result.*
`MBM> p on –all`	*Power on all partitions.*

`telnet  <natbox address> 323`
`P00>>> sho boot*`
`P00>>> set bootdef_dev …`
`P00>>> set boot_osflags 0,0`
`P00>>> set os_type VMS`
`P00>>> b`
	Once the poweron tests are finished, connect to the first partition and define the boot device, boot flags, and operating system type. Then boot the partition.

`telnet  <natbox address> 324`
`P00>>> sho boot*`
`P00>>> set bootdef_dev …`
`P00>>> set boot_osflags a`
	Connect to the second partition, and perform the same operations as for the first.

```
P00>>> set os_type UNIX
P00>>> b
telnet  <natbox address> 325
P00>>> sho boot*
P00>>> set bootdef_dev …
P00>>> set os_type UNIX
P00>>> b
```

Connect to the third partition, and perform the same operations as for the first.

Chapter 6
Final Steps

After you have created partitions on a platform, using whichever tool, you need to install the operating system of choice in each partition.

Instructions for installing an operating system are given in the Installation Guides for the particular OS. This documentation is available online and can be found by using the search facility at the www.hp.com website. It is generally easiest to search for, say, "OpenVMS documentation" and then locate the appropriate installation material from there.

The installation process involves defining a boot device and of course, the operating system type is known, so the firmware environment variables for this information are already defined once the installation is complete.

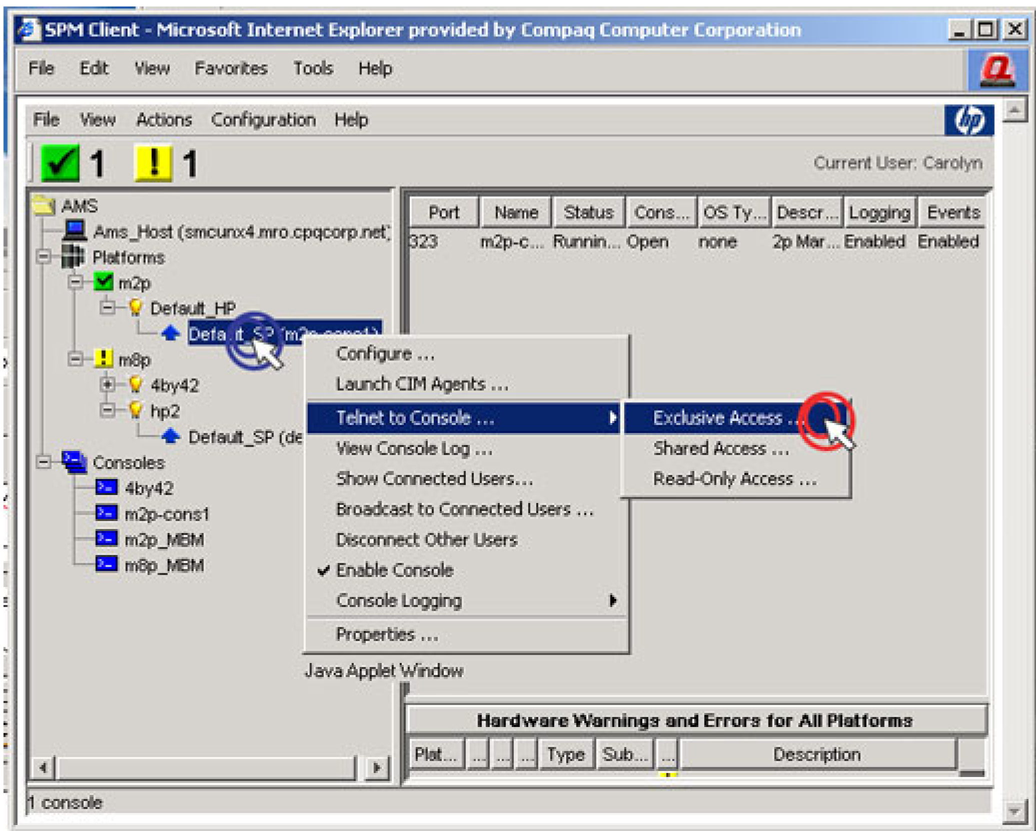
6.1 Accessing an SRM Console from the SPM

There are several ways you can access an SRM console from the SPM.

6.1.1 Using Soft Partition Icons:

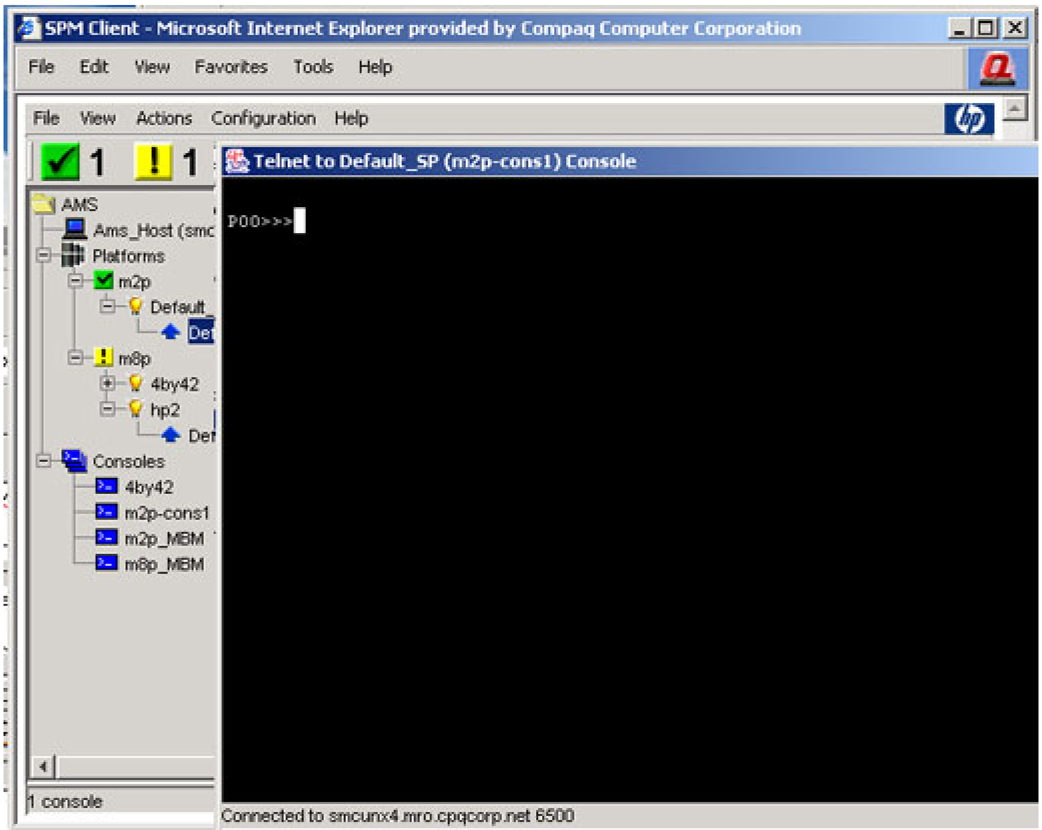
1 Right-click on the **soft** partition you want to access, in either the tree view or the right pane. Left-click on "Telnet to Console" from the drop-down menu, with whichever type of access you want.

Figure 6-1 Accessing the SRM Console from the SPM Using Soft Partition Icons



2 When the black telnet screen appears, press "Return" and the boot processor will display the SRM console prompt.

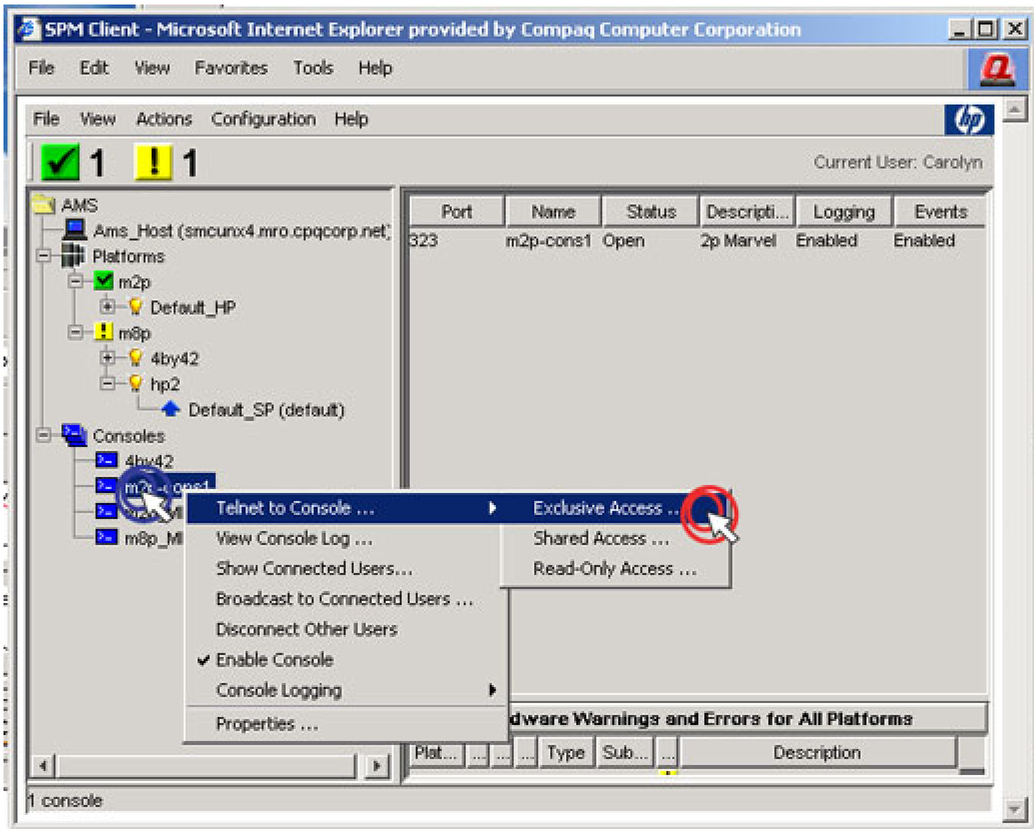
Figure 6-2 Press “Return” at the Black Telnet Screen



6.1.2 Using Console Icons Defined by the APW

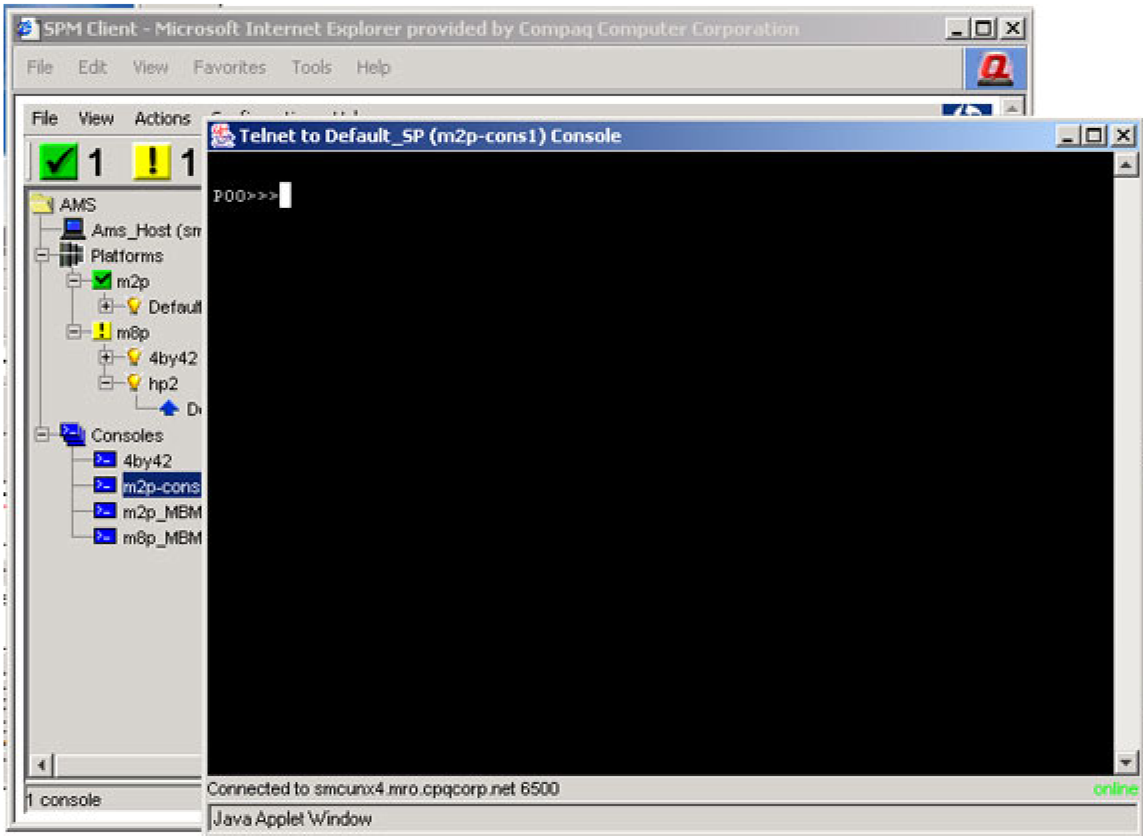
1. Right-click on the console icon defined in the APW for the desired soft partition. Left-click on "Telnet to Console" from the drop-down menu, with whichever type of access you want.

Figure 6-3 Accessing the SRM Console from the SPM Using Console Icons



2 When the black telnet screen appears, press "Return" and the boot processor will display the SRM console prompt.

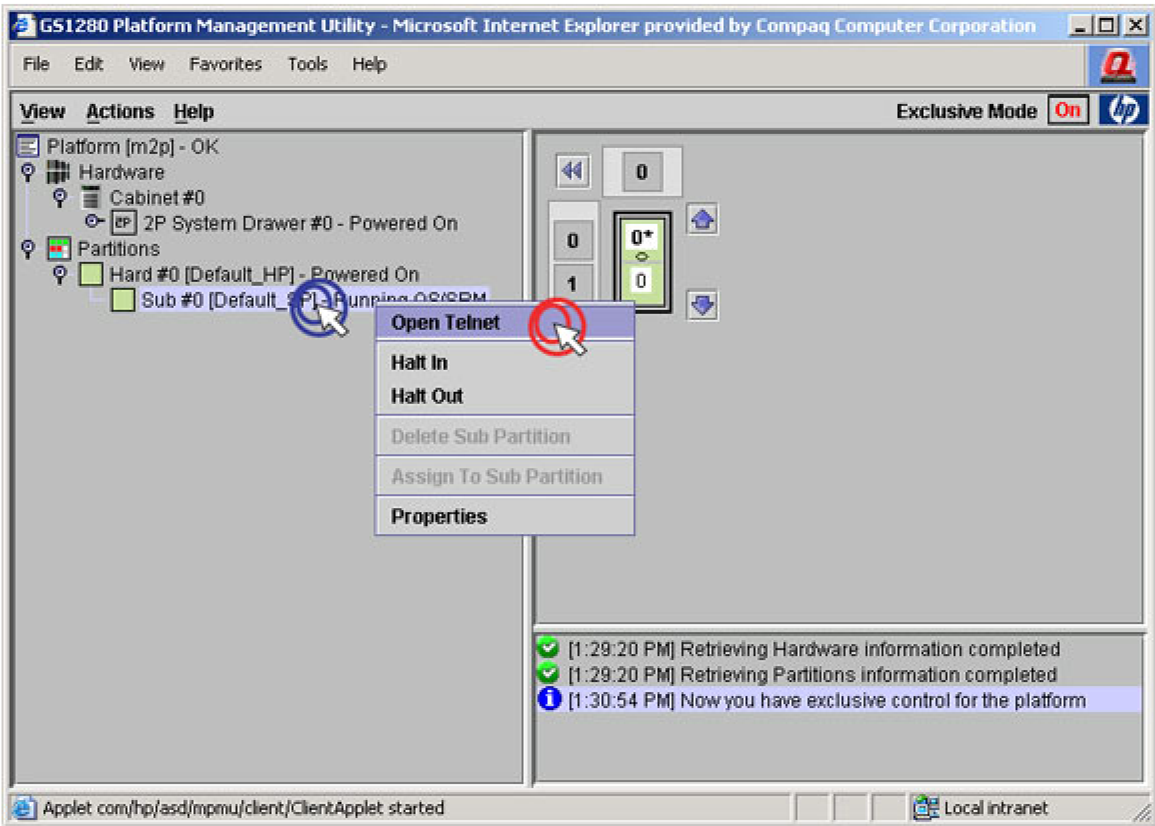
Figure 6-4 Press “Return” at the Black Telnet Screen



6.2 Accessing the SRM from the AMU

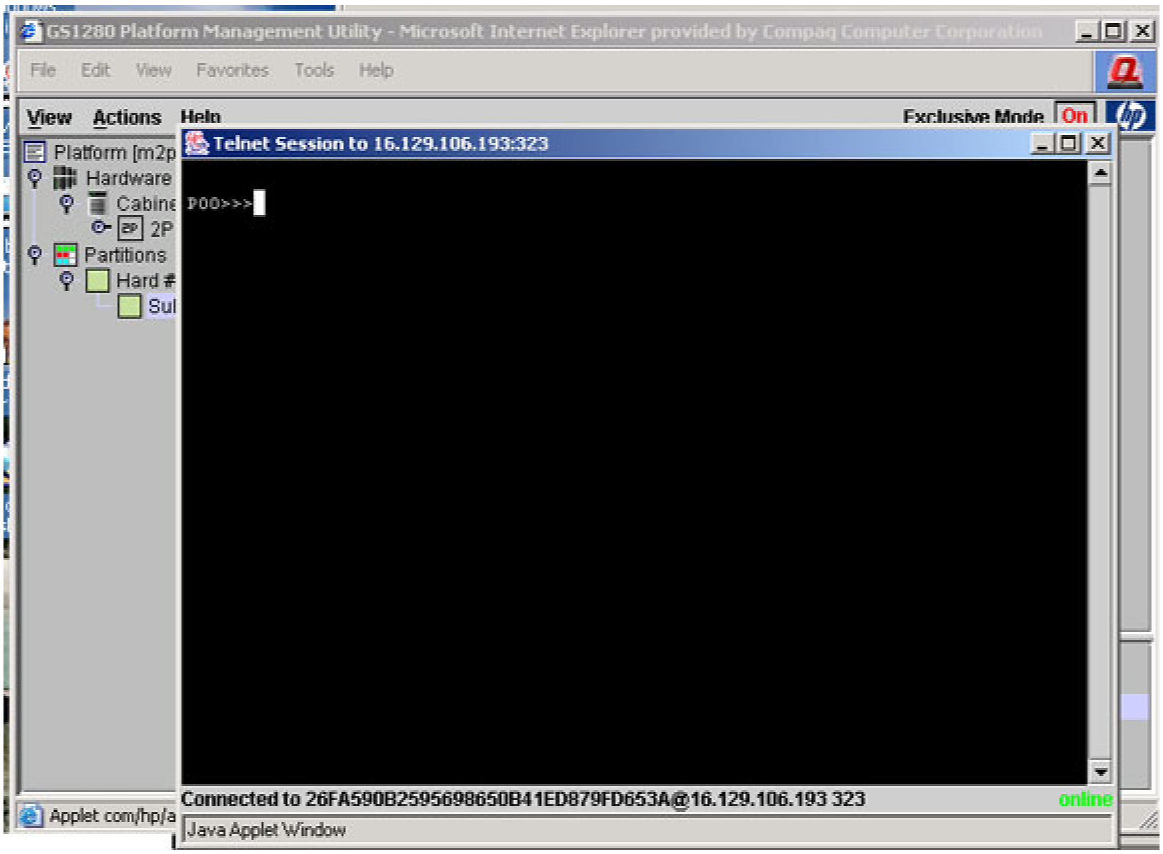
1 Right-click on the soft-partition name in the tree view and left-click on "Open Telnet". (If you have requested "Exclusive Access" from the AMU, the telnet access will be exclusive. If not, telnet access will be shared.)

Figure 6-5 Open Telnet Session from AMU



2 When the black telnet screen appears, press "Return" to see the SRM console prompt.

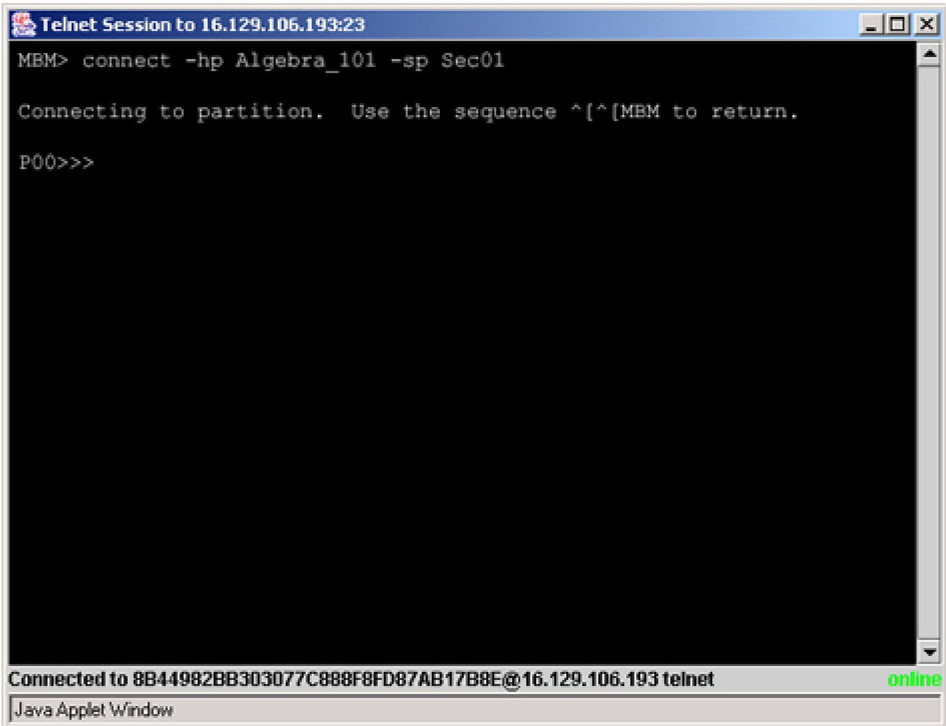
Figure 6-6 Press "Return" at the Black Telnet Screen



6.3 Accessing the SRM from the CLI

However you reached the Server Management CLI interface (see Accessing the CLI, Section 4.1), use the **connect** command, specifying the hard partition (and subpartition, if you have created soft partitions within a hard partition).

Figure 6-7 Telnet to the SRM from the CLI

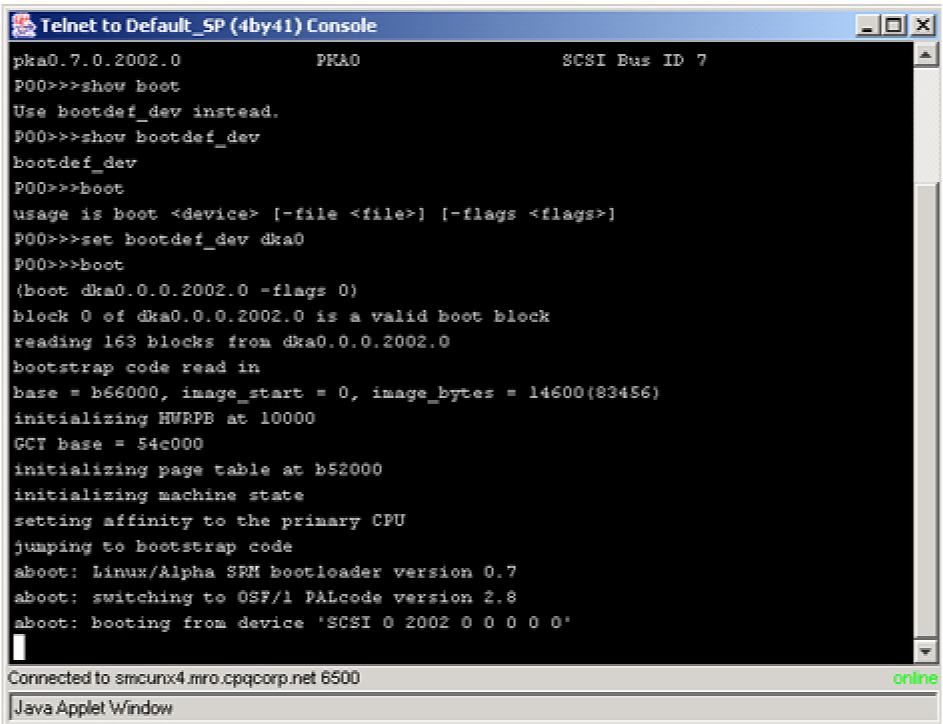


6.4 Useful SRM Commands

Having reached the SRM console for a given partition, you may want to ensure that the boot device and operating system type have been defined during the operating system installation process. The example below illustrates such commands, as well as the boot command to start the operating system.

The SRM **boot** command is described in detail, along with other SRM commands, in the AlphaServer *ES47/ES80/GS1280 SRM Reference* documentation.

Figure 6-8 Example of SRM Commands



Appendix A
CLI Partitioning Command Formats

The Server Management CLI commands which are useful in partitioning are described here for your convenience. (All CLI commands are described in the *hp AlphaServer CLI Reference.*)

A.1 assign component

Assigns CPU and IO resources to a partition.

Note: A CPU and its associated IOs cannot be separated across hard partitions. In order to move the IOP to a different hard partition, move its associated CPU to the targeted partition, and the IOP will automatically follow the CPU to the same destination. The IOP can be moved around between sub partitions within the same hard partition.

Syntax

assign component	[-cabinet <cabinet> -drawer <drawer>] {CPU<n>, IOP<n>, SBB} -hp <hard_partition> [-sp <sub_partition>]

Arguments

CPU<n>	The CPU by cabinet and drawer number. *n* refers to the CPU number within the 8P drawer and ranges from 0 to 7.
IOP<n>	The IOP by cabinet and drawer number. *n* refers to the IOP of the 8P drawer and ranges from 0 to 7. A user can assign the IOP only to a subpartition.
SBB	The SBB by cabinet and drawer number.
-hp <hard_partition>	Hard partition name.
-sp <sub_partition>	Subpartition name. Assign to subpartition 0 if no subpartition name is given.

Options

-cabinet <cabinet>	Cabinet number in hex. If not specified, the number defaults to 0.
-drawer <drawer>	Drawer number in hex. If not specified, the number defaults to 0.

Examples

1. Add the entire 8P drawer to the hard partition *account_dept.*

```
MBM> assign comp –ca 0 –dr 3 sbb –hp account_dept
MBM>
```

2. Delete the IO attached to CPU2 of the subpartition *acct_usa.*

```
MBM> assign comp –ca 0 –dr 3 IOP2 –hp account_dept -sp Free_Pool
MBM>
```

3. Add CPU 1 to the subpartition *acct_usa.*

```
MBM> assign comp –ca 0 –dr 3 cpu1 –hp account_dept –sp
Free_Pool
MBM> assign comp –ca 0 –dr 3 cpu1 –hp account_dept –sp acct_usa
MBM>
```

4. Move IOP1 from a different hard partition, resulting an error.
5.

```
MBM> assign comp –ca 0 –dr 3 cpu1 –hp account_dept –sp acct_usa


~CLI-E-(tCLImodem) cannot move IOP across different HP
MBM> [2003/04/11  10:21:36]
~PCO-W-(pco)01)  Command handler rtn failed for Id:5622 cmd:41a
```

See Also

show partition, show system, create partition, assign memory

A.2 assign memory

Assigns a portion of the hard partition's physical memory to a subpartition. Explicitly assigning memory to a hard partition is unnecessary because the RIMM population associated with each CPU determines the partition's memory size.

Users cannot assign memory to the Free_Pool.

The smallest memory chunk allowed is 8MB.

To deassign memory, specify 0MB as the memory size.

Syntax

assign memory	<memory_size> -hp <hard_partition> {-sp <sub_partition>, -com}

Arguments

<memory_size>	The memory size in units of GB or MB; must be multiples of 8MB.
-hp <hard_partition>	Hard partition name.
-sp <sub_partition>	Subpartition name.
-com	Assign the memory to the community.

Options

None

Examples

1. Assign 16GB of memory to the subpartition *acct_usa.*

```
MBM> assign mem 16GB –hp account_dept -sp acct_usa
MBM>
```

2. Assign 2GB of memory to the community.

```
MBM> assign mem 16GB -hp account_dept -com
MBM>
```

3. Deassign memory from the subpartition *acct_usa*. Memory for this subpartition becomes unspecified.

```
MBM> assign mem 0MB -hp account_dept -sp acct_usa
MBM>
```

See Also

show partition, create partition, assign component

A.3 create partition

Creates a partition. This command adds a new hard partition by providing the hard partition name, maximum number of CPUs, subpartition type, and optional modifiers. It creates a subpartition if the subpartition name is given. No attributes are necessary for a subpartition. (You use assign component or assign memory to define the components of a partition or subpartition.)

If no hard partitions are defined, a Default_HP and Default_SP are created with all CPUs and IOs belonging to the Default_SP. The creation of the first user-created hard partition causes the Default_HP to be deleted. The components of the deleted Default_HP go to the Global Free_Pool. Whenever a hard partition is created, a Default_SP along with the Free_Pool is created under that hard partition. When the first user-defined subpartition is created under a hard partition, the Default_SP is deleted and its components go to the Global Free_Pool.

Syntax

create partition {{-hp <hard_partition> <max_CPUs> <partition_type> [-stripe] },
{-hp <hard_partition> –sp <sub_partition>}}

Arguments

-hp <hard_partition>	Hard partition name. The name is limited to 19 characters (alphanumeric and underscore only).
-sp <sub_partition>	Subpartition name. The name is limited to 19 characters (alphanumeric and underscore only). A hard partition is created if no subpartition name is given.
max_CPUs	The maximum number of CPUs that can ever be placed in the hard partition. This value is used to determine the proper routing and memory algorithm.
partition_type	The type of subpartitions in this hard partition: soft soft partitions firm firm partitions semi semi-firm partitions

Options

-stripe Specify striping memory.

Examples

1. Create a hard partition *account_dept* with a maximum of 8 CPUs, define any subsequently created subpartitions as soft partitions, and request striping memory.

```
MBM> create partition –hp account_dept 255 soft -stripe
MBM>
```

2. Create a subpartition under the hard partition *account_dept.*

```
MBM> create partition –hp account_dept –sp acct_usa
MBM>
```

See Also

show partition, save partition, delete partition, assign component, assign memory

A.4 delete partition

Destroys a partition. Returns all resources of the specified partition to the Free_Pool. The partition(s) being destroyed must be in the halt or power-off state.

When the last user-defined subpartition within a hard partition is deleted, a Default_SP within that hard partition is created with no components. When the last user-defined hard partition in the system is deleted, a Default_HP and Default_SP are created with all CPUs and IOs belonging to the Default_SP. The user cannot delete the Free_Pool.

Syntax

delete partition {-hp <hard_partition>
[-sp <sub_partition>], -all}

Arguments

-hp <hard_partition>	Hard partition name.
-sp <sub_partition>	Subpartition name. The hard partition and all of its subpartitions are deleted if no explicit subpartition is given.
-all	All partitions.

Options

None

Examples

1. Delete the subpartition *asia_branch* of hard partition *account_dept*

```
MBM> delete partition –hp account_dept –sp asia_branch
MBM>
```

2. Attempt to delete the subpartition Free_Pool

```
MBM> delete part -hp account_dept -sp Free_Pool
~CLI-E-(tCLImodem) disallow deleting sub partition Free_Pool
Usage: delete partition {-hp <hard_partition> [-sp <sub_partition>],
-all}
MBM>
```

3. Delete the entire hard partition *account_dept*

```
MBM> delete partition –hp account_dept
MBM>
```

4. Delete all partitions. A Default_HP and Default_SP are created.

```
MBM> delete partition -all
MBM>
```

See Also

show partition, save partition, create partition

A.5 help

Displays information about server management CLI commands. Describes command conventions and available commands; includes a detailed description and syntax for a given command.

Syntax

help [<command>, *]

Arguments

command	The command to be displayed.
*	Display all commands.

Options

None

Examples

1. List the help topics.

```
MBM> help
NAME
hel[p]
FUNCTION
Display information about CLI commands.
SYNOPSIS
help [<command>, *]
Command synopsis conventions:
<item> Implies a placeholder for user specified item.
<item>... Implies an item or list of items.
[] Implies optional keyword or item.
{a,b,c} Implies any one of a, b, c.
{a|b|c} Implies any combination of a, b, c.

The following help topics are available:
assign component    assign memory       bootline            build fru
clear               clear alert         clear display       clear error
clear log           clear port          clear srmenv        config cables
connect             create partition    d                   delete partition
deposit             disable alert       disable remote      disable test
e                   el                  enable alert        enable remote
enable test         examine             halt in             halt out
hangup              help                init modem          locate
logout              power off           power on            reset
save partition      set                 set alert           set baud
set dial            set escape          set flow            set init
set membership      set password        set sys_serial_num  set time
show                show cables         show duo            show error
show fru            show grid           show log            show membership
show memory         show modem          show network        show partition
show power          show system         show sys_serial_num show time
show version        start partition     telnet              test
test alert          test led            update              uptime
MBM>
```

2. Help on a particular command. The characters in square brackets ([]) are optional.

```
MBM> help show cable
NAME
sh[ow] cab[le]
FUNCTION
Display the cable configuration.
SYNOPSIS
show cable [-ip | -io]

MBM>
```

A.6 modify partition

Modify existing hard partition attributes. This command will change the hard partition attributes, including maximum CPUs and memory striping.

Syntax

modify partition -hp <hard_partition> {CPU <max_CPUs | stripe {on, off}}

Arguments

-hp <hard_partition>	Hard partition name
-CPU <max_CPUs>	The maximum number of CPUs that can ever be placed in the hard partition. This value is used to determine the proper routing and memory algorithm.
-stripe {on, off}	Turn on or turn off memory striping

Options

None.

Examples

1. Modify hard partition of account_dept by allowing more CPUs.

```
MBM> modify partition -hp account_dept -cpu 64
MBM>
```

See Also

show partition, create partition

A.7 power off

Turns off the power to a platform, component, or partition. This command powers off a non-partitioned platform if no arguments are given, or it powers off a specific hard partition or component.

Use the **show power** command to confirm the power state.

Syntax

power off [{[-cabinet <cabinet> -drawer <drawer>] {DUO<n>, PCI, SBB}},
{-hp <hard_partition>, -all}]

Arguments

DUO<n>	The dual-CPU module by cabinet and drawer number. *n* refers to the module within the 8P drawer and ranges from 0 to 3.
PCI	The PCI drawer by cabinet and drawer number.
SBB	The entire system building block.

Options

-cabinet <cabinet>	Cabinet number in hex. If not specified, the number defaults to 0.
-drawer <drawer>	Drawer number in hex. If not specified, the number defaults to 0.
-hp <hard_partition>	Hard partition name.
-all	All partitions.

Examples

1. Power off a platform with only one hard partition. This is the only case in which the **power off** command can be used without arguments.

```
MBM> power off
[2003/04/10  03:09:43]
~PCO-I-(pco 04)  Preparing to power off partition.  HP:
 Default_HP
MBM>
```

2. Issuing the **power off** command without arguments returns an error if the platform has more than one hard partition.

```
MBM> p off
~CLI-E-(tCLImodem) need to specify partition or component
Usage: power off [{[-cabinet <cabinet> -drawer <drawer>]
{DUO<n>, PCI, SBB}},
               {-hp <hard_partition>}, {-all}]
MBM>
```

3. Power off hard partition *account_dept*.

```
MBM> power off -hp account_dept
[2003/04/10  03:09:43]
~PCO-I- (pco 01)  Preparing to power off partition.  HP:
 account_dept
MBM>
```

4. Power off all partitions.

```
MBM> power off –all
[2003/04/10 03:09:43]
~PCO-I-(pco_04) Preparing to power off partition. HP: HP0
[2003/04/10 03:09:44]
~PCO-I-(pco_00) Preparing to power off partition. HP: HP1
[2003/04/10 03:09:44]
~PCO-I-(pco_03) Preparing to power off partition. HP: HP2
[2003/04/10 03:09:45]
~PCO-I-(pco_01) Preparing to power off partition. HP: HP3
MBM>
```

5. Power off dual-processor module 3 in drawer 2 of cabinet 0.

```
MBM> power off -ca 0 -dr 2 duo3
MBM>
```

6. Power off 8P drawer 2 of cabinet 0.

```
MBM> power off -ca 0 -dr 2 sbb
MBM>
```

7. Power off PCI drawer 2 of cabinet 0xA.

```
MBM> power off -ca 1 -dr 2 pci
MBM>
```

See Also

power on, show power

A.8 power on

Turns on the power to a system, component, or partition.

If no arguments are given, this command powers on a system with one hard partition. By using arguments, you can power on a specific hard partition or component.

When there is only one hard partition, the screen display shows the entire power up sequence like the grid and diagnostics. However, if more then one hard partitions is defined in the database, the hard partitions are powered on in parallel, a maximum of four hard partitions s at a time. The detailed power up progress messages like the grid and the diagnostics messages, are sent to the console session and do NOT come out on the MBM session.

Given two hard partitions (Yankees and RedSox) each with two sub-partitions (Jeter and Bernie in Yankees, and Pedro and Noma in RedSox) issuing either "`power on –all`", "`power on –hp RedSox`", or "`Power on –hp Yankees`" will power up the partitions and send the output to the console session. When a user connects to the console session (`conn –hp RedSox –sp Pedro`) the powerup messages (the grid and diagnostics) will be displayed. All of the sub-partition console buffers will contain the grid and diagnostics power up messages.

Any diagnostic failures from any hard partition are delivered to the MBM session for high visibility. They also are sent to the console session.

Use the **show power** command to confirm the power state.

Syntax

power on [{[-cabinet <cabinet> -drawer <drawer>] {DUO<n>, PCI, SBB}},
{-hp <hard_partition>, -all}] [-xsrom]

Arguments

DUO<n>	The dual-processor module by cabinet and drawer number. *n* refers to the module within the 8P drawer and ranges from 0 to 3.
PCI	The PCI drawer by cabinet and drawer number.
SBB	The entire system building block.

Options

-cabinet <cabinet>	Cabinet number in hex. If not specified, the number defaults to 0.
-drawer <drawer>	Drawer number in hex. If not specified, the number defaults to 0.
-hp <hard_partition>	Hard partition name.
-all	All partitions.
-xsrom	Power up until the level at which XROM tests complete.

Examples

1. Power on a system with only one hard partition. This is the only case in which the **power on** command can be used without arguments.

```
MBM> power on
[2003/04/16 00:06:30]
~PCO-I-(pco_01) Preparing to power on partition. HP: Default_HP
[2003/04/16 00:06:43]
~PCO-I-(pco_01)
Configuring for 22 CPUs for HP:0 Default_HP

    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
    .w..w..w..w..w..w..w..w.......................
0   wP--P--P--F--P--P--P--Fw......................
   .|..|..|..|..|..|..|..|.......................   P  Processor
   .|..|..|..|..|..|..|..|.......................   F  Filler
1   wP--P--P--F--P--P--P--Fw......................    |  connection
   .|..|..|..|..|..|..|..|.......................   -  connection
   .|..|..|..|..|..|..|..|.......................
2   wP--P--P--F--P--P--F--Fw......................    W <---------->
E
   .|..|..|..|..|..|..|..|....................... N ^ (0,0)
   .|..|..|..|..|..|..|..|.......................   |  .
3   wP--P--P--F--P--P--F--Fw......................    |    .
   .w..w..w..w..w..w..w..w.......................   |      .
   .............................................. S v      (ns,ew)
4   ..............................................
   ..............................................
   ..............................................
5   ..............................................
   ..............................................
   ..............................................
6   ..............................................
   ..............................................
   ..............................................
7   ..............................................
   ..............................................
    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
[2003/04/16 00:06:54]
~PCO-I-(pco_01) Running diagnostics on HP: Default_HP
Running test 10, Initialize RAMBUS ... on 22 EV7s
Running test 11, Initialize Memory ... on 21 EV7sRunning test 12, Data
Pattern March read/write ... on 21 EV7s
Running test 13, RAID channel Test ... on 11 EV7s
Running test 14, Single Bit Error ... on 21 EV7s
Running test 15, Double Bit Error ... on 21 EV7s
Running test 20, Init IO7 and Start Clocks ... on 4 EV7s
Running test 21, IO7 Data Path (Scratch CSR) ... on 4 EV7s
Running test 22, IO Single Bit Error checkers ... on 4 EV7s
Running test 23, IO Double Bit Error checkers ... on 4 EV7s
Running test 24, IO Timer Expirations ... on 4 EV7s
Running test 25, IO up-hose SBE checkers ... on 4 EV7s
Running test 26, IO up-hose DBE checkers ... on 4 EV7s
Running test 27, IO7 pass2 data mover test ... on 4 EV7s
Running test 30, Configure RBOX Routes ... on 22 EV7s
Running test 31, Clear Errors / Enable Routes ... on 22 EV7s
Running test 32, Route Test: N S E W ... on 22 EV7s (North)
Running test 32, Route Test: N S E W ... on 22 EV7s (South)
Running test 32, Route Test: N S E W ... on 22 EV7s (East )
Running test 32, Route Test: N S E W ... on 22 EV7s (West )
Running test 33, Inverse Route Setup ... on 22 EV7s
```

```
Running test 32, Route Test: N S E W ... on 22 EV7s (North)
Running test 32, Route Test: N S E W ... on 22 EV7s (South)
Running test 32, Route Test: N S E W ... on 22 EV7s (East )
Running test 32, Route Test: N S E W ... on 22 EV7s (West )
Running test 33, Inverse Route Setup ... on 22 EV7s
Running test 34, Single Bit Error checker ... on 22 EV7s
Running test 35, Double Bit Error checker ... on 22 EV7s
Running test 31, Clear Errors / Enable Routes ... on 22 EV7s
Running test 16, Interprocessor Memory Access ... on 21 EV7s
Running test 40, Local I/O Device Interrupts ... on 22 EV7s
Running test 41, Local Interval Timer Interrupts ... on 22 EV7s
Running test 42, Local Interprocess Interrupts ... on 22 EV7s
Running test 43, Software Alerts ... on 1 EV7s
Running test 46, Other Local Interrupt Bits ... on 22 EV7s
[2003/04/16 00:08:40]
~PCO-I-(pco_01) Diagnostics completed on HP: Default_HP
[2003/04/16 00:08:40]
~PCO-I-(pco_01) HP:Default_HP SP:Default_SP Primary is NS:0 EW:0 which
is cab:00 drw:0 cpu:0
Running test 50, Loop on Secondary Routine ... on 21 EV7s
(SP:Default_SP)
[2003/04/16 00:08:41]
~PCO-I-(pco_01) Loading SRM on Primary for HP: Default_HP, SP:
Default_SP.
Running test 50, Loop on Secondary Routine ... on 0 EV7s
(SP:Free_Pool)[2003/04/16 00:08:45]
~PCO-I-(pco_01) Powered On HP:Default_HP
MBM>
```

2. Issuing the **power on** command without arguments returns an error if the system has more than one hard partition.

```
MBM> p on
~CLI-E-(tCLImodem) need to specify partition or component
Usage: power on [{[-cabinet <cabinet> -drawer <drawer>] {DUO<n>, PCI, SBB}},
                   {{-hp <hard_partition>, -all} | [-xsrom]}]
MBM>
```

3. Power on hard partition *account_dept* in a multiple-partitioned system:

```
MBM> power on -hp account_dept
[2003/04/16 00:14:56]
~PCO-I-(pco_02) Preparing to power on partition. HP: account_dept
[2003/04/16 00:15:15]
~PCO-I-(pco_02) Running diagnostics on HP: account_dept
[2003/04/16 00:15:31]
~DIA-W-(pco_02) Test 17 [T12 with isolation info (MFG)] failed on cpu [NS: 0
EW: 1] which is cab:00 drw:0 cpu:2
[2003/04/16 00:16:21]
~PCO-I-(pco_02) Diagnostics completed on HP: account_dept
[2003/04/16 00:16:21]
~PCO-I-(pco_02) HP:account_dept SP:Default_SP Primary is NS:0 EW:0 which is
cab:00 drw:0 cpu:0
[2003/04/16 00:16:21]
~PCO-I-(pco_02) Loading SRM on Primary for HP: account_dept, SP: Default_SP.
[2003/04/16 00:16:25]
~PCO-I-(pco_02) Powered On HP:account_dept
MBM>
```

4. Power on all partitions.

```
MBM> power on –all
[2003/04/14 14:13:30]
~PCO-I-(pco_01) Preparing to power on partition. HP: hp0
[2003/04/14 14:13:31]
~PCO-I-(pco_02) Preparing to power on partition. HP: hp1
[2003/04/14 14:13:45]
~PCO-I-(pco_02) Running diagnostics on HP: hp1
[2003/04/14 14:13:48]
~PCO-I-(pco_01) Running diagnostics on HP: hp0
[2003/04/14 14:14:45]
~PCO-I-(pco_02) Diagnostics completed on HP: hp1
[2003/04/14 14:14:45]~PCO-I-(pco_02) HP:hp1 SP:Default_SP Primary is
NS:2 EW:0 which is cab:00 drw:1 cpu:0
[2003/04/14 14:14:45]~PCO-I-(pco_02) Loading SRM on Primary for HP:
hp1, SP: Default_SP.
[2003/04/14 14:14:48]
~PCO-I-(pco_01) Diagnostics completed on HP: hp0
[2003/04/14 14:14:48]
~PCO-I-(pco_01) HP:hp0 SP:Default_SP Primary is NS:0 EW:0 which is
cab:00 drw:0 cpu:0
[2003/04/14 14:14:48]
~PCO-I-(pco_01) Loading SRM on Primary for HP: hp0, SP: Default_SP.
[2003/04/14 14:14:50]
~PCO-I-(pco_02) Powered On HP:hp1
[2003/04/14 14:14:53]
~PCO-I-(pco_01) Powered On HP:hp0
MBM>
MBM> conn -hp hp0 -sp default_sp

Connecting to partition. Use the sequence ^[^[MBM to return.
~PCO-I-(pco_01) Preparing to power on partition. HP: hp0
~PCO-I-(pco_01)

Configuring for 8 CPUs for HP:0 hp0
     0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
     ................................................
0   wP--P--P--Pw....................................
    .|..|..|..|.....................................  P  Processor
    .|..|..|..|.....................................  F  Filler
1   wP--P--P--Pw....................................  |  connection
    ................................................  -  connection
    ................................................
2   ................................................   W <----------> E
    ................................................ N ^ (0,0)
    ................................................   |  .
3   ................................................   |    .
    ................................................   |       .
    ................................................ S v      (ns,ew)
4   ................................................
    ................................................
    ................................................
5   ................................................
    ................................................
    ................................................
6   ................................................
    ................................................
    ................................................
7   ................................................
    ................................................
     0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
```

```
~PCO-I-(pco_01) Running diagnostics on HP: hp0
Running test 10, Initialize RAMBUS ... on 8 EV7s
Running test 11, Initialize Memory ... on 8 EV7s
. . . .        (more progress messages)
P00>>>
```

5. Power on until xsrom completes.

```
MBM> power on -all -xsrom
MBM>
```

6. Power on dual-CPU module 3 in drawer 2 of cabinet 0.

```
MBM> power on -ca 0 -dr 2 duo3
MBM>
```

7. Power on 8P drawer 2 of cabinet 0.

```
MBM> power on -ca 0 -dr 2 ssb
MBM>
```

8. Power on PCI drawer 2 of cabinet 1.

```
MBM> power on -ca 1 -dr 2 pci
MBM>
```

See Also

power off, show power, reset, connect

A.9 show grid

Shows the power-on grid for a particular hard partition or all hard partitions. The power-on grid shows how the CPUs interconnect. Options may be omitted if there is only one hard partition.

Syntax

show grid [-hp <hard_partition>, -all]

Arguments

None

Options

-hp <hard_partition>	Specify the name of the hard partition for which information is displayed.
-all	Display information for all partitions.

Examples

Show grid for the case of only one hard partition.

```
MBM> sh grid
Powerup grid for partition Default_HP
      0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
      ...............................................
 0   .P--P--P--F....................................
     .|..|..|..|....................................   P  Processor
     .|..|..|..|....................................   F  Filler
 1   .P--P--P--F....................................   |  connection
     ...............................................   -  connection
     ...............................................
 2   ...............................................    W <----------> E
     ...............................................  N ^ (0,0)
     ...............................................    |  .
 3   ...............................................    |    .
     ...............................................    |       .
     ...............................................  S v      (ns,ew)
 4   ...............................................
     ...............................................
     ...............................................
 5   ...............................................
     ...............................................
     ...............................................
 6   ...............................................
     ...............................................
     ...............................................
 7   ...............................................
     ...............................................
      0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
MBM>
```

A.10 show partition

Displays partition attributes and resources. This command confirms the partition settings. By default it lists all partitions.

Syntax

show partition [{–hp <hard_partition> [-sp <sub_partition>]}, -all] [-brief]

Arguments

None

Options

-hp <hard_partition>	Hard partition name.
-sp <sub_partition>	Subpartition name. All subpartitions under the hard partition are displayed if no subpartition is specified.
-all	All partitions.
-brief	Highlight the partition attributes, number of CPUs and connected IOs

Examples

1. List all partitions.

```
MBM> show part

-----------------------------------------------------------------
Hard Partition : HP Name = USA_branch, HP No.= 0, SP count = 2
Attributes     : max CPUs = 32, SP type = soft, Non-stripe
Physical Memory: 8192MB (8.000GB)

       Community Memory: 0MB (0.000GB)
       Sub Partition: HP Name = USA_branch, HP No.= 0
                      SP Name = hr, SP No.= 0
                      State = Not Running, Telnet port = 323
               Assigned Memory: 1024MB (1.000GB)

               CPUs:
                   Cab  Drw  CPU   (NS,EW)   PID    Type
                    0    0    0    ( 0,0 )    0     Primary
                    0    0    2    ( 0,1 )    2     Non-primary
                    0    0    4    ( 0,2 )    4     Non-primary
                    0    0    6    ( 0,3 )    6     Non-primary
                    0    0    1    ( 1,0 )    1     Non-primary
                    0    0    3    ( 1,1 )    3     Non-primary
                    0    0    5    ( 1,2 )    5     Non-primary
                    0    0    7    ( 1,3 )    7     Non-primary
               IOPs:
                          SBB                           PCI Drawer
                   Cab  Drw  IOP   (NS,EW)            Cab  Drw  IOR
                    0    0    0    ( 0,0 ) ---------   1    1    0
                    0    0    2    ( 0,1 )
                    0    0    4    ( 0,2 )
                    0    0    6    ( 0,3 )
                    0    0    1    ( 1,0 )
                    0    0    3    ( 1,1 )
                    0    0    5    ( 1,2 )
                    0    0    7    ( 1,3 )
       Sub Partition: HP Name = USA_branch, HP No.= 0
                      SP Name = Free_Pool, SP No.= 255
                      State = Not Running
               Free Memory: 0MB (0.000GB)

               CPUs: None                IOPs: None

MBM>
```

2. Show only one subpartition.

```
MBM> sh part -hp USA_branch -sp hr

-----------------------------------------------------------------
Hard Partition : HP Name = USA_branch, HP No.= 0, SP count = 2
Attributes     : max CPUs = 32, SP type = soft, Non-stripe
Physical Memory: 8192MB (8.000GB)

       Community Memory: 0MB (0.000GB)

       Sub Partition: HP Name = USA_branch, HP No.= 0
                      SP Name = hr, SP No.= 0
                      State = Not Running, Telnet port = 323
               Assigned Memory: 1024MB (1.000GB)

               CPUs:
                   Cab  Drw  CPU   (NS,EW)   PID    Type
                    0    0    0    ( 0,0 )     0    Primary
                    0    0    2    ( 0,1 )     2    Non-primary
                    0    0    4    ( 0,2 )     4    Non-primary
                    0    0    6    ( 0,3 )     6    Non-primary
                    0    0    1    ( 1,0 )     1    Non-primary
                    0    0    3    ( 1,1 )     3    Non-primary
                    0    0    5    ( 1,2 )     5    Non-primary
                    0    0    7    ( 1,3 )     7    Non-primary

               IOPs:
                          SBB                            PCI Drawer
                   Cab  Drw  IOP   (NS,EW)           Cab  Drw  IOR
                    0    0    0    ( 0,0 ) ---------  1    1    0
                    0    0    2    ( 0,1 )
                    0    0    4    ( 0,2 )
                    0    0    6    ( 0,3 )
                    0    0    1    ( 1,0 )
                    0    0    3    ( 1,1 )
                    0    0    5    ( 1,2 )
                    0    0    7    ( 1,3 )
MBM>
```

3. Show partitions in brief format.

```
MBM> sh part -b
Partition Name       CPU Count   IO Count        State      Telnet Port
hp0                    32 max                  Power Off
 Default_SP           16          2           Not Running       323
hp1                    32 max                  Power Off
 Default_SP            8          1           Not Running       324
MBM>
```

See Also

create partition, delete partition

Appendix B
Routing

This appendix discusses some esoterics of routing, to answer questions about what happens when CPU modules are missing, the effect of filler modules in a partition, and so forth.

B.1 Routing Algorithm Considerations

The routing algorithm is designed to prevent deadlocks of requests and responses within hard partitions in the three-dimensional torus. The firmware determines the routes during a hard partition power-on or reset.

For each CPU in a hard partition, the router firmware creates a unique routing table to determine how to get from the CPU (including its memory and/or I/O) making a request to the CPU that will service the request. A memory read is an example of such a request.

The routing table is indexed by the *processor id*, or *PID*. (Section B.2 discusses how PIDs are assigned.) Each indexed entry contains all the information that will be used to get a request to a given destination (another PID) within the hard partition.

The path that a packet takes must be *minimal*; that is, it can never move further away from a destination, only closer. (So routing never goes "the long way around.") There must be at least one deadlock-free path between any two CPUs. There may be additional non-deadlock-free paths, but the first one must exist.

The concepts that define routing are described below.

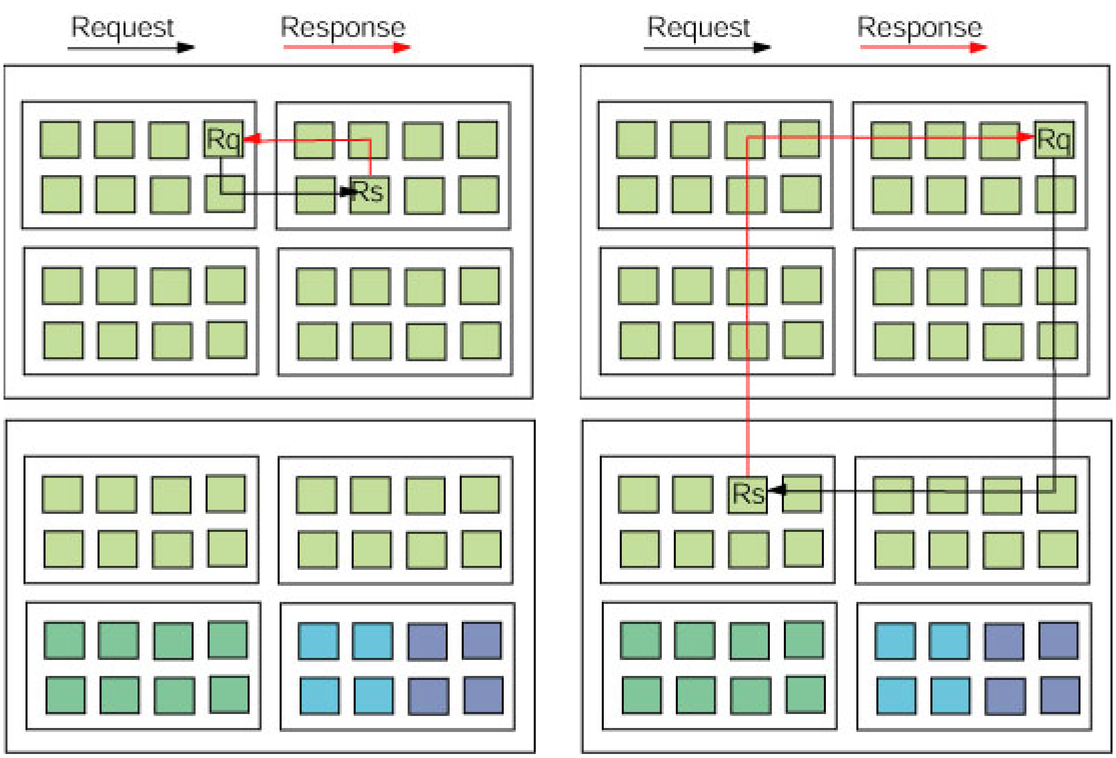
B.1.1 Primary Axis

Since hard partitions are by definition rectangular in shape (in the two-dimensional presentation), they have a *primary axis,* along the North-South direction or the East-West direction. The primary axis follows the direction of the longest side of the partition. The primary axis determines whether all routes from a given CPU begin in the North-South or East-West direction *first*. (Note: a square partition can use either.)

The router table entry for a CPU gives the North-South, East-West coordinate of the destination. This information, along with the primary axis information, determines which interprocessor port (North, South, East, or West) the packet will be sent out on. The packet will continue along the row or column of the primary axis until it needs to move along the opposite axis (if needed). It must then reach its destination. This is also referred to as "one-turn" routing. This method only works for a perfect torus or rectangle. Two examples are shown in Figure B-1. The "Rq" processor is the one making a request, say for a memory read. The "Rs" processor is the one responding to the request.

Note that a filler module will pass a packet along in the direction it is headed, but cannot be used to make a turn.

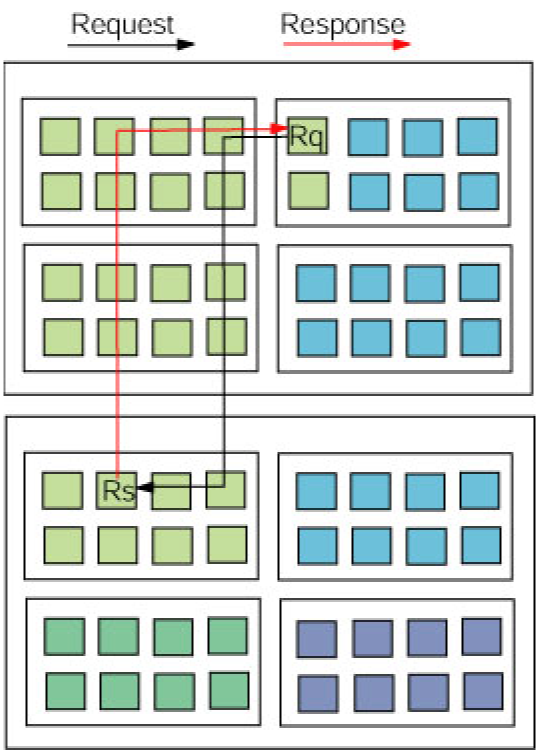
Figure B-1 One-Turn Routing Examples



B.1.2 Initial Hop

The "initial hop" feature allows a CPU in a non-rectangular configuration to make one jump from outside the rectangular region into the region. Within the rectangular region, only one-turn routing will work. Initial hop is specified in the router table entries. The hop can only occur in the North, South, East, or West direction. This essentially gives the routing one "free turn". Click here for an example. Here, the requesting CPU is allowed one hop into the rectangular region of the partition, where the "one-turn" rule applies.

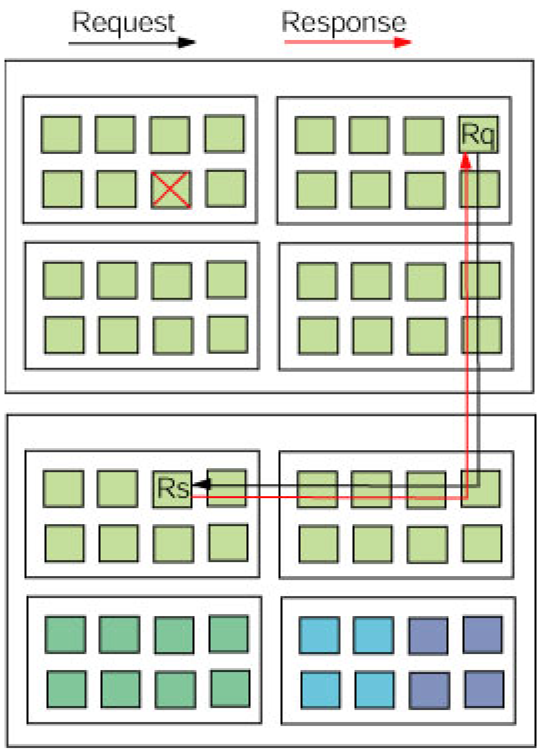
Figure B-2 Initial Hop Example



B.1.3 SRC Routing

This method takes advantage of a proof that showed that there are cases in which it is OK to flip the primary axis if all the other CPUs in the secondary row/column do it. This provides another way for an otherwise unroutable configuration to proceed. Figure B-3 shows an example. This can be useful if there are missing CPUs or filler modules in awkward positions within the hard partition. It should *not* be used just to create hard partitions that are not rectangular.

Figure B-3 SRC Routing Example



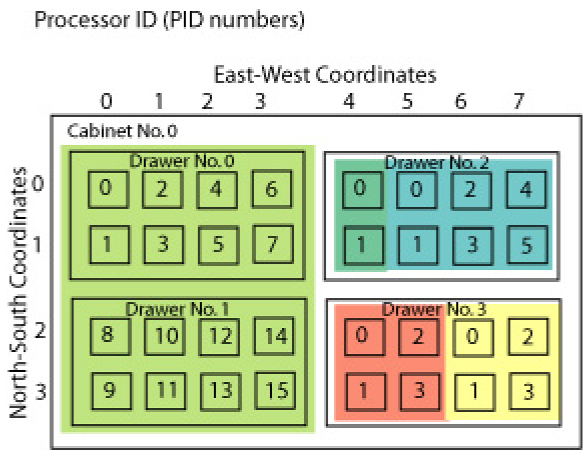
B.2 Processor ID (PID) Assignment

The processor id (or PID) is a number from 0 to 255 that is used to identify a particular CPU within a hard partition. It is the "CPU number" that is known to the SRM firmware and the operating system. It appears in some displays provided by the three partitioning tools, but it is ***not*** used in the user interfaces of these tools when you define partitions.

The PID is primarily used by the routing algorithm firmware to identify CPUs and define routing tables, as described in the topic Routing. PID numbers are assigned according to the coordinates on the torus, and PID numbers are assigned to empty slots, if a CPU module happens to be missing in a drawer.

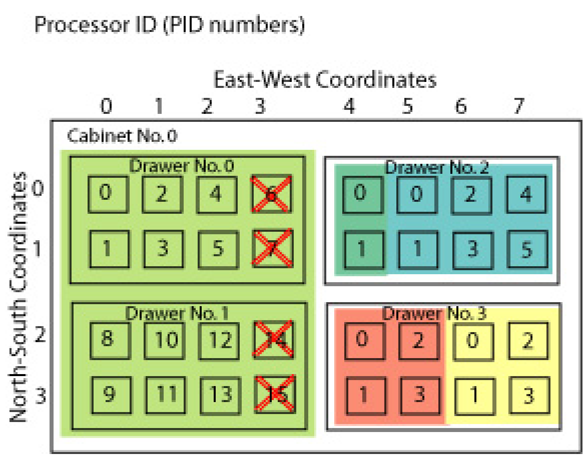
Some example PID numbers for partitions in a fully populated platform are shown in Figure B-4. PID number are assigned proceeding from the top left corner of the partition, and move in pairs according to CPU module interconnects.

Figure B-4 Example PID Numbers



For an example of how assigning PID numbers to empty slots can cause confusion, consider Figure B-5. The lime green partition has an edge of empty slots shown by the red crosses in the squares. Still, these slots are assigned PID numbers that show in displays. A user may consider that this partition contains nine processors and wonder why some of them have PIDs larger than nine. In this case, the answer is that PIDs are assigned to empty slots within the partition.

Figure B-5 PID Numbers Are Assigned to All Slots, Empty or Not



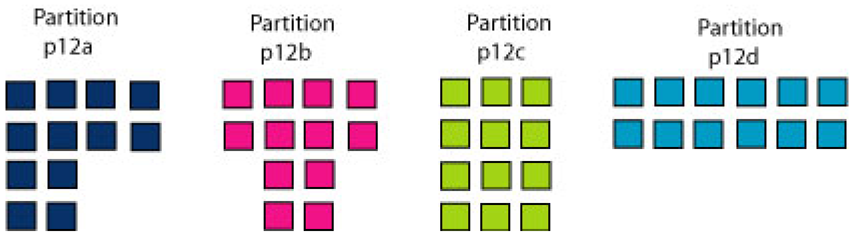
Appendix C
Performance Tests Relating to Partition Configuration Guidelines

This appendix shows performance test results relating to partition configuration guidelines.

C.1 Performance Data on Rectangular Shapes

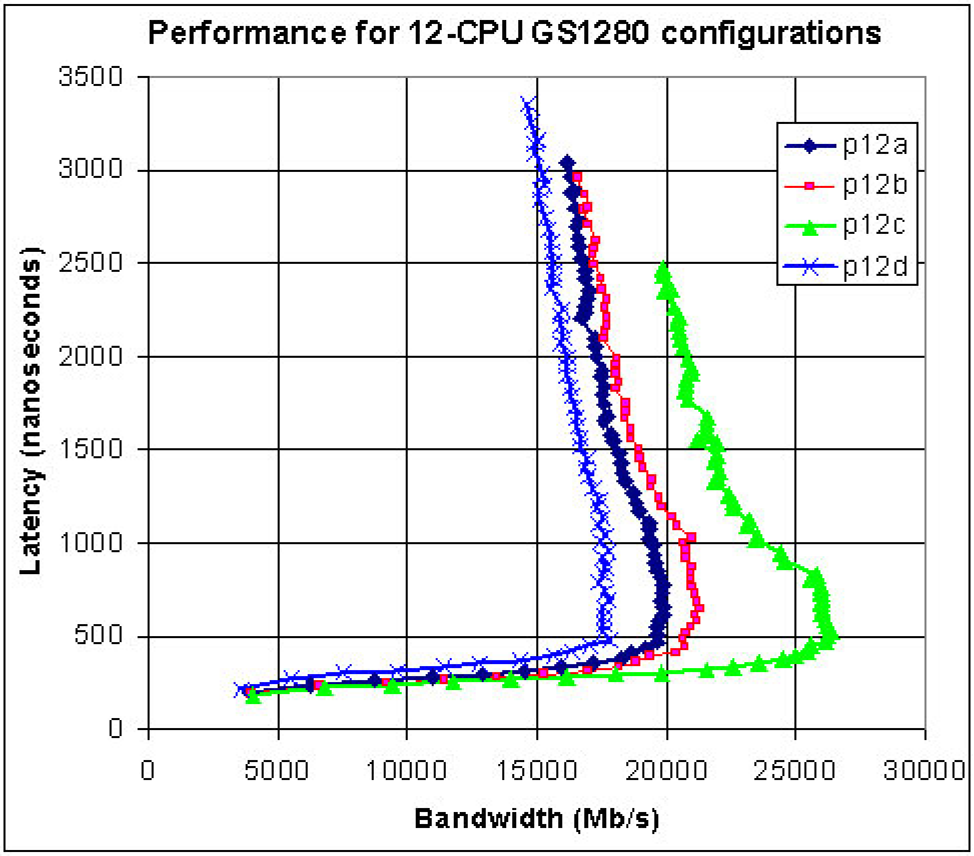
This topic shows performance comparisons for various 12-P shapes on a 64-P GS1280 system. The diagram below shows the shape of four different partitions tested. Note that no wraparound occurs for any of these partitions.

Figure C-1 Shapes of Partitions Tested



The results of the performance tests are shown in the graph below. The test is started with a single outstanding load (leftmost point on the graph). For each additional point, one outstanding request is added. In an ideal case, as the number of outstanding requests increases, the bandwidth increases too (moving to the right), and latency does not change (line stays low and flat).

Figure C-2 Performance Data



The results show that the 2x6 configuration (p12d) has the worst performance: the lowest bandwidth and the highest latency. The 4x3 configuration (p12c) has the best performance.

The performance difference between p12d and p12c is 15-20% at low loads (leftmost datapoints) and 30-50% at high loads. Note that the load is application-dependent: some applications stress the IP interconnect bandwidth substantially (for example, parallel OMP applications), while others do not (for example, independent/multistream jobs).

The reason for such a substantial difference in performance is the 1.5 times increase in the bi-directional bandwidth in p12c vs. p12d.

Recommendation #1:

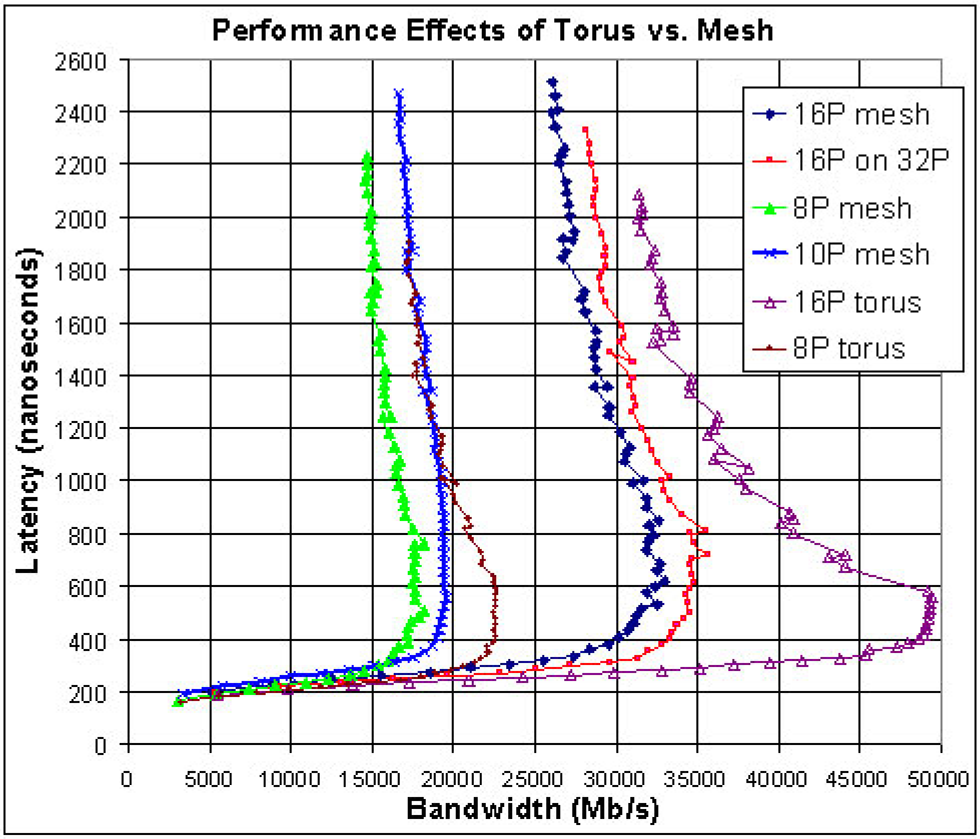
Avoid long and narrow shapes. Use shapes that are as close to the square as possible. The comparison of configurations 12a and 12b shows that is it better to ‘clump' or 'centralize' the CPUs as much as possible. The configuration 12a outperforms configuration 12b. The performance difference is 4-9%.

C.2 Performance Data on Torus Wraparounds

The graph in compares the performance of torus vs. mesh configurations. The performance improvement of torus vs. mesh is:

- 8-15% for low loads and 30-50% for high loads on 16P
- 5-10% for low loads and 20-30% for high loads on 8P

Figure C-3 Performance Data on Torus vs Mesh



Therefore, when a large GS1280 system is partitioned into partitions that do not use any of the wraparound links in the torus, the resulting partitions will typically have the lower performance than same-sized GS1280 models using the full torus connections.

For example, the performance of a 16P partition constructed by partitioning a GS1280 Model 64 into four 4 x 4 partitions is worse than the GS1280 Model 16.

In addition, even a partial loss of links results in a loss of performance, although much less significant. One such example is the 4 x 4 16P partition on a GS1280 Model32 (which has north-south connections on the torus) compared to the 16P GS1280 Model 16 in the graph below.

Recommendation # 2:

Use wrap-around wires whenever possible. For example, on a 4x8 system, a 4x2 partition can use the North/South wrap-around wires, whereas a 2x4 partition cannot.

Appendix D
CPU Speeds

A partition cannot contain CPUs that run at different speeds. The possibilities include those CPUs available for

- *AlphaServer* ESxx Platforms
- *AlphaServer* GS1280 Platforms

CPU Speeds for *AlphaServer* ESxx Platforms

AlphaServer ESxx platforms can use two different CPUS. A platform may run CPUs of different speeds, but only one type of CPU may run within a given hard partition.

CPU Speed	Part Number
1000 MHz	3X-KN73A-xx
1150 MHz	3X-KN73C-xx

CPU Speeds for AlphaServer GS1280 Platforms

AlphaServer GS1280 platforms can use two different CPUS. A platform may run CPUs of different speeds, but only one type of CPU may run within a given hard partition.

CPU Speed	Part Number
1150 MHz	3X-KN72C-xx
1300 MHz	3X-KN72D-xx

Index