Sun Fire™ 3800–6800 Servers Dynamic Reconfiguration

Peter Gonscherowski

Sun BluePrints™ OnLine – April 2002

http://www.sun.com/blueprints

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054 USA
650 960-1300 fax 650 969-9131

Part No. 816-4560-10
Revision 1.0, 03/27/02
Edition: April 2002

Please
Recycle

Sun Fire™ 3800–6800 Servers Dynamic Reconfiguration

Dynamic Reconfiguration (DR) is a key component of the Solaris™ Operating Environment (Solaris OE). By using DR, hardware components can be added or removed from a system with minimal interruption. DR increases the overall uptime and availability of systems. While using DR for system upgrades, modifications, or service actions, the Solaris OE and user applications remain operating.

This document contains the following sections about DR:

- A general overview of DR
- DR implementation on Sun Fire™ 3800–6800 servers
- Best-practice guidelines for DR and command line procedures

DR Overview

To use DR on Sun Fire 3800–6800 servers, the domain must be running a minimum version of the Solaris OE and a minimum version of the firmware on the Sun Fire server system controller (Sun Fire SSC). These minimum versions are:

- Solaris 8 02/02 OE release (on the domain)
- ScApp 5.12.6 and RTOS 19 (on the Sun Fire SSC)

DR is automatically enabled on the domain with Solaris 8 02/02 OE release. The Sun™ Management Center (Sun MC) software includes a graphical user interface (GUI) for DR operations in its DR module. This module is provided in the Sun Management Center 3.0 update 4 software release, which is included in the Solaris 8 OE 02/02 release. The `cfgadm`(1M) command is the command line interface (CLI) for DR on the domains.

DR is performed on attachment points, which consists of two objects: a receptacle and an occupant. The state of the attachment point refers to the condition of the receptacle and occupant. You can use the `cfgadm`(1M) command or the Sun MC DR module to retrieve the status of attachment points, occupants, and receptacles, along with their condition. The Sun Fire 3800–6800 servers support the following attachment points:

- I/O assembly (PCI/cPCI assemblies)
- CPU/Memory boards
- cPCI cards
- System memory
- CPUs

Dynamic attachment points are attachment points that are on base attachment points. Dynamic attachment points do not have to support the same DR operations as their base attachment points. On the Sun Fire 3800–6800 servers, the CPU/Memory boards, I/O assemblies, and cPCI cards are base attachment points.

Connecting and Disconnecting Attachment Points

You can connect or disconnect attachment points. When you connect an occupant, its hardware is connected to the interconnect, and its OpenBoot™ PROM structures are created. The DR software performs the power-on self-test (POST) if it is supported by the occupant. In the connected state, the occupant is not available to the Solaris OE. When you disconnect an occupant, it is removed from the interconnect, and after powering off, it is ready for removal.

Configuring and Unconfiguring Attachment Points

You can configure or unconfigure attachment points. When you configure an occupant, the device tree structures are created, and the occupant is available to the Solaris OE. When you unconfigure an occupant, the device tree structures are removed, and the occupant is not usable to the Solaris OE.

DR on Sun Fire 3800–6800 Servers Domains

The Sun Fire 3800–6800 servers support multiple domains. You can allocate resources to the domains on a board level by assigning or unassigning them to or from domains by using DR operations. In addition, you can use DR operations on the domain or on the Sun Fire SSC to make an attachment point (for example, a CPU/Memory board or I/O assembly) available to the domain.

Attachment Point States for CPU/Memory Boards and I/O Assemblies

You can change the state of CPU/Memory boards or I/O assemblies by using Sun Fire SSC and DR commands. FIGURE 1 shows the different states of a board or assembly, from the perspective of DR on the Sun Fire SSC.

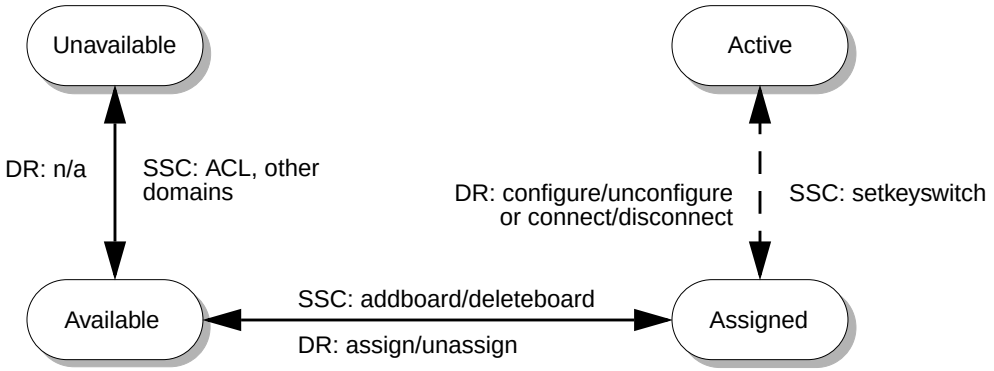

FIGURE 1 Board States From the Perspective of the Sun Fire SSC

The following table contains descriptions of the states:

TABLE 1 Board State Descriptions

State	Description
Unavailable	The board or assembly is not in the access control list (ACL) or assigned to another domain.

TABLE 1 Board State Descriptions

State	Description
Available	The board or assembly is listed in the ACL and is not in use by another domain.
Assigned	The board or assembly is assigned to a domain and is not available to any other domain.
Active	The board or assembly is being used by the domain.

Changing the State of the Board or Assembly

You can verify the state of the CPU/Memory board or I/O assembly on the Sun Fire SSC by using the `showboards` command or by using the DR module in the Sun MC software (FIGURE 2 shows the attachment points in the CPU board table). If an attachment point is not assigned to a domain, you can connect or configure it.

CPU/Memory board and I/O assembly attachment points are implicitly assigned to a domain as part of a connect or configure operation.

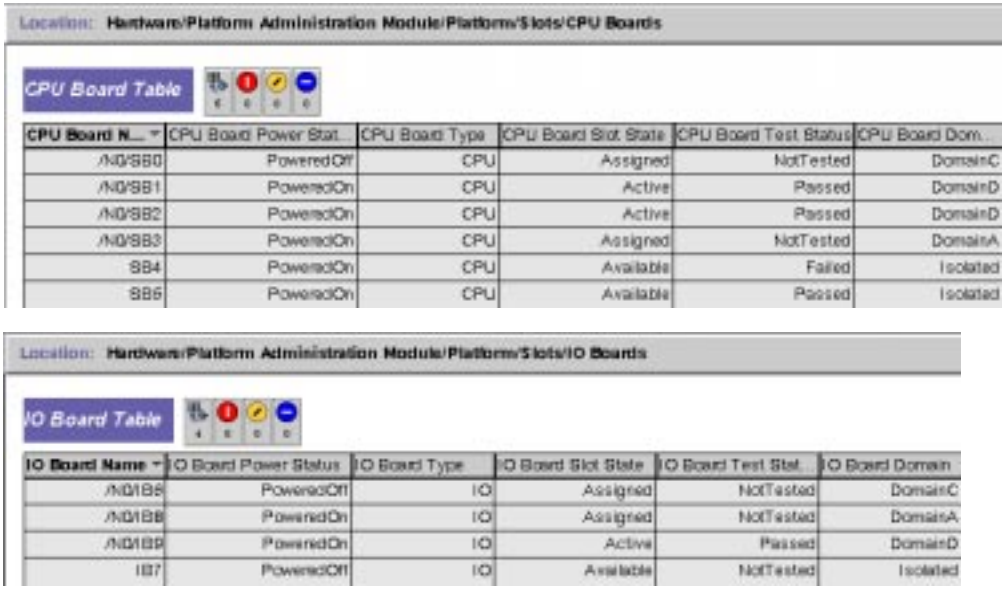

FIGURE 2 CPU Boards in the Sun MC Software

See the “Sample Output From the `cfgadm`(1M) and `showboards` Commands” on page 19 for an example of the typical output displayed when using the `cfgadm`(1M) command on the domain and the `showboards` command on the Sun Fire SSC.

DR on CPU / Memory Boards

CPU and memory boards can be connected, disconnected, configured, or unconfigured. You can view the status of these boards by using the DR module in the Sun MC software. In the following figure, boards `N0.SB1` and `N0.SB4` are in the disconnected, untested state, and are available to be configured into a domain.

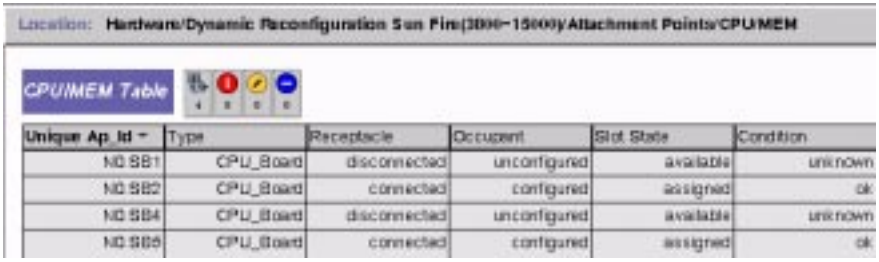

FIGURE 3 CPU / Memory Component Table in Sun MC

▼ To View DR Operations for CPU / Memory Boards

1. **Log in to the domain as superuser.**

2. **Use the `cfgadm`(1M) command with its help option (`-h`) and the name of the CPU/Memory board to view the DR operations for the CPU/Memory board.**

   The following example shows the output of the command.

```
# cfgadm -h NO.SB2
Usage:
cfgadm [-f] [-y|-n] [-v] [-o hardware_options] -c function ap_id [ap_id...]
cfgadm [-f] [-y|-n] [-v] [-o hardware_options] -x function ap_id [ap_id...]
cfgadm [-v] [-s listing_options] [-o hardware_options] [-a] [-l [ap_id|ap_type...]]
cfgadm [-v] [-o hardware_options] -t ap_id [ap_type...]
cfgadm [-v] [-o hardware_options] -h [ap_id |ap_type...]
Sbd specific commands/options:
cfgadm [-o parable] -l ap_id
cfgadm [-o unassign|nopoweroff] -c disconnect ap_id
cfgadm -t ap_id
cfgadm -x assign ap_id
cfgadm -x unassign ap_id
cfgadm -x poweron ap_id
cfgadm -x poweroff ap_id
```

Unconfiguring Memory

The entire amount of memory on a CPU/Memory board is a dynamic attachment point. The same DR mechanisms are used if you unconfigure memory individually or as part of an unconfigure-disconnect operation on a CPU/Memory board.

When unconfiguring memory, the DR operation differentiates between permanent and nonpermanent memory. Permanent memory is nonpageable (that is, it contains kernel or OpenBoot PROM structures). All of the other memory is pageable and is considered nonpermanent. DR uses different mechanisms for pageable and nonpageable memory.

You can determine if memory is permanent or nonpermanent from the CLI by using the `cfgadm -av | grep permanent` command or by using the Sun MC DR module, as shown in the following figure.

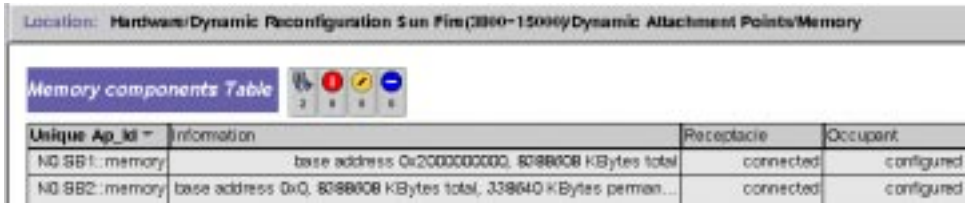

FIGURE 4 Memory Component Table in Sun MC

For nonpermanent memory, the memory pages are flushed back to a disk, moved to another memory location, or swapped out appropriately. The length of time this operation takes depends on the amount of memory being unconfigured and the system usage of the memory (for example, memory pages that are locked by processes). The Solaris OE and user applications are blocked from using these pages during the unconfigure process.

For permanent memory, a copy-rename operation is used because permanent memory contains critical kernel structures, so it cannot be swapped out. Before you can unconfigure permanent memory, nonpermanent memory on another CPU/Memory board must be unconfigured. The amount of nonpermanent memory must be at least as large as the amount on the board with permanent memory. Only the size of the memory is relevant, the memory layout is not.

After the DR software has a place for the permanent memory, the Solaris OE and user applications are quiesced (suspended), and the memory is copied to the new location. The memory controllers are then renamed appropriately. During the quiesce, all memory activity by the operating system is stopped, and all I/O operations and thread activity is paused. For most domains, only one CPU/Memory board contains permanent memory.

Before unconfiguring nonpermanent memory, you must do the following:

- Change the memory interleaving to either within-board or within-cpu if the domain was not booted with memory interleaving set to one of these values, then reboot the domain.

Before unconfiguring permanent memory, you must do the following:

- Stop all real-time processes.
- Ensure that all of the device drivers are Device Driver Interface (DDI) compliant.

The following table contains the requirements for the DDI functions on loaded device drivers with different types of attachment points.

TABLE 2 DDI Requirements for Device Drivers

Support	Permanent Memory	Nonpermanent Memory	I/O Devices
For Unconfiguring:			
DDI_DETACH	Yes	Yes	Yes
DDI_SUSPEND	Yes		
DDI_RESUME	Yes		
For Configuring:			
DDI_ATTACH	Yes	Yes	Yes

The following table contains a description of the DDI, IPMP, and Traffic Manager support for common Sun drivers. See your service representative for updates to this list.

TABLE 3 DDI, IPMP, and Traffic Manager Support for Drivers

Driver	DDI	IPMP	Traffic Manager
`hme`	Yes	Yes	N / A
`isptwo`	Yes	N / A	TBD
`qfe`	Yes	Yes	N / A
`ba`	Yes	TBD	N / A
`qlc`	Yes	N / A	Yes
`glm`	Yes	N / A	TBD

Unconfiguration Commands

The unconfiguration commands for DR are the same for CPU / Memory boards with or without permanent memory. When you unconfigure permanent memory, you will see a message about the memory, and you must confirm the operation.

Unconfiguration Example

In this example, SB1 has 833 Mbytes of permanent memory, and SB2 has nonpermanent memory. If you unconfigure or disconnect SB1, or if you unconfigure the memory on SB1, the operating system will be quiesced. If you unconfigure or disconnect SB2, or unconfigure the memory on SB2, the system will not be quiesced.

FIGURE 5 shows the dynamic attachment points memory in the domain. The location of the permanent memory depends on previous DR operations, the size of the kernel, and the memory distribution within the domain; therefore, the permanent memory does not have to be on the lowest numbered CPU/Memory board in the domain.

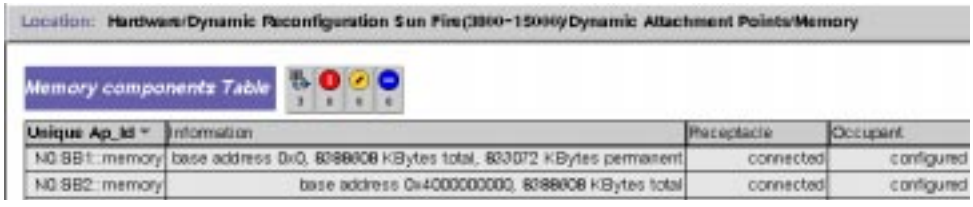

FIGURE 5 Memory Information for the Domain

FIGURE 6 shows the memory output after the memory on SB1 has been unconfigured. Comparing this figure with the previous figure, you can see that the memory base address for SB2 has changed from `0x2000000000` to `0.x0`. You can also see that the permanent memory has been moved from SB1 to SB2. The memory base address for SB1 is now the prior memory base address for SB2. The permanent memory base address, after the unconfigure operation, is still `0x0`.

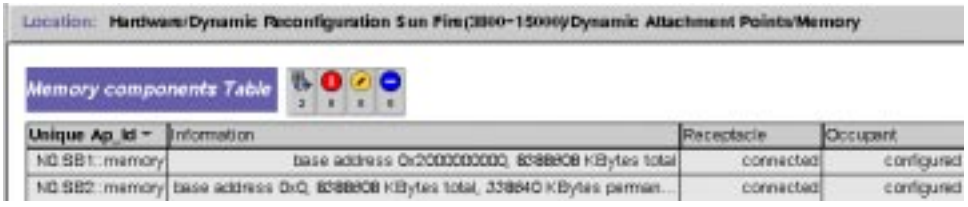

FIGURE 6 Memory Information After an Unconfigure Operation

Configuring Memory

Configuring memory into the domain is a simple operation that adds the memory to the Solaris OE memory structures. This memory can then be used by the Solaris OE and by user applications. You must set up interleaving within the board before configuring a CPU/Memory board or its memory into the domain.

DR Operation Time

The total amount of time used for unconfiguring and disconnecting a CPU/Memory board depends on two segments of time. The first segment is the length of time it takes for CPU/Memory resources to be removed from a domain. Understanding this length of time helps you determine the time required for a CPU/Memory board to be disconnected prior to performing upgrades or service actions.

The second segment of time is the length of time it takes for the system to be quiesced when permanent memory is unconfigured or disconnected. The quiesce period depends on the amount of time the DR software takes to perform the copy-rename operation and on the amount of time it takes to suspend and resume device drivers and processes. The total time is affected by the number of processors, the number of I/O device drivers that have to be suspended and resumed, and the amount of memory that needs to be copied.

When you unconfigure or disconnect a CPU/Memory board with used memory pages, the pages must be freed up. This process is known as draining the memory. The drain time depends on the use of the page by the operating system, the amount of pages that are locked by processes, and the amount of memory being unconfigured. System load and locking methods are specific to the system or application using the pages.

The total amount of time required for the drain and quiesce operations depends on the system configuration and the applications running on the domain.

The total amount of time used for connecting and configuring a CPU/Memory board depends on the time it takes to run the POST operation.

Note – Measuring the drain and quiesce time on production systems is important. It allows you to estimate the length of time needed for configuration changes and upgrades. It also helps to manage the impact of unconfiguring and disconnecting permanent memory.

Time Length Examples

In the following examples, the systems are configured to run only the Solaris OE. This condition allows the examples to show clearly the influencing factors on the length of time needed for the DR operations.

This first example, the number of CPU / Memory boards in the domain was increased stepwise, then the length of time for disconnecting the CPU / Memory board containing permanent memory was measured. The disconnect time is the sum of the time taken for memory drain and system quiesce. FIGURE 7 shows how adding CPUs to the domain increases the disconnect time.

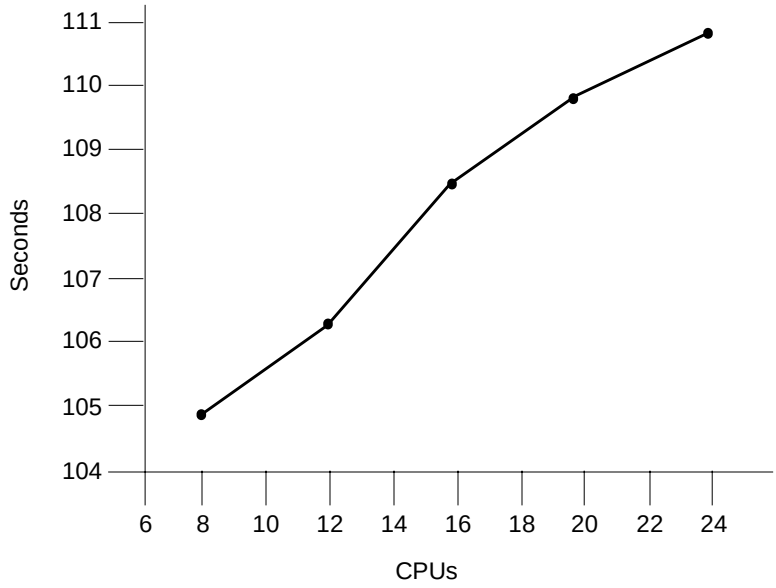

FIGURE 7 CPUs as a Factor of the Disconnect Time

Each step of four CPUs corresponds to a 2-Gbyte step in total memory. The disconnect times show that as you increase the number of CPUs, you increase the disconnect time. The increase in time is due to the fact that more CPUs have to be suspended and resumed during the copy-rename operation. The increase in time is not significant, so the increase of CPUs has a low impact on the disconnect time.

In the second example, the same configuration was used, but the number of CPUs was held constant at 24. The total amount of memory in the domain was increased stepwise, then the quiesce time was measured. To derive the quiesce time, debug kernels were used because with a production kernel, only the sum of the quiesce and drain time can be measured accurately.

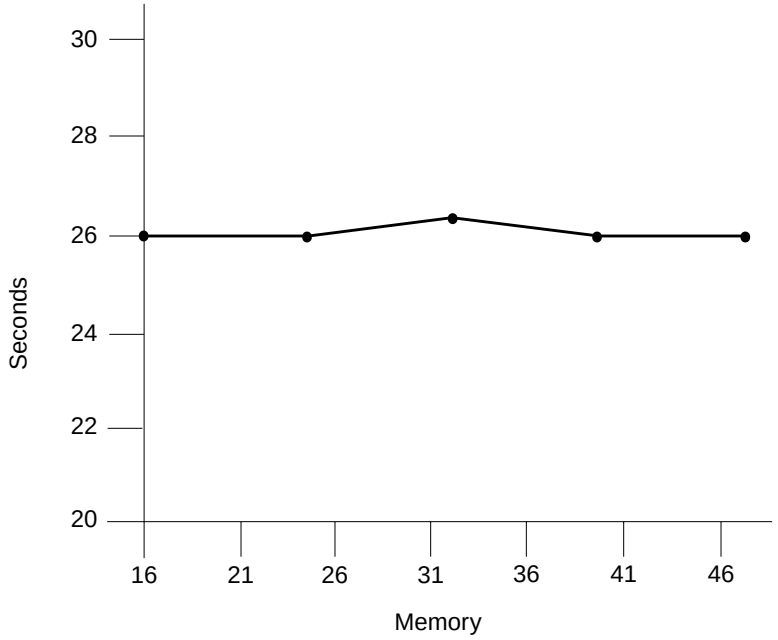

FIGURE 8 Memory as a Factor of the Quiesce Time

FIGURE 8 shows that a static number of CPUs does not significantly impact the amount of time for the quiesce operation.

Removing Individual CPUs and Memory Banks

CPUs and memory are dynamic attachment points on a CPU/Memory board. Individual CPUs or memory banks cannot be unconfigured independently because of the association of the memory banks and CPUs and the fact that the entire memory on a CPU/Memory board is treated as a single dynamic attachment point.

The memory controller is implemented on the UltraSPARC™ III CPUs. Each CPU controls two of the eight memory banks on the CPU/Memory board. FIGURE 9 shows the dynamic attachment points for the CPUs and memory on a CPU/Memory board.

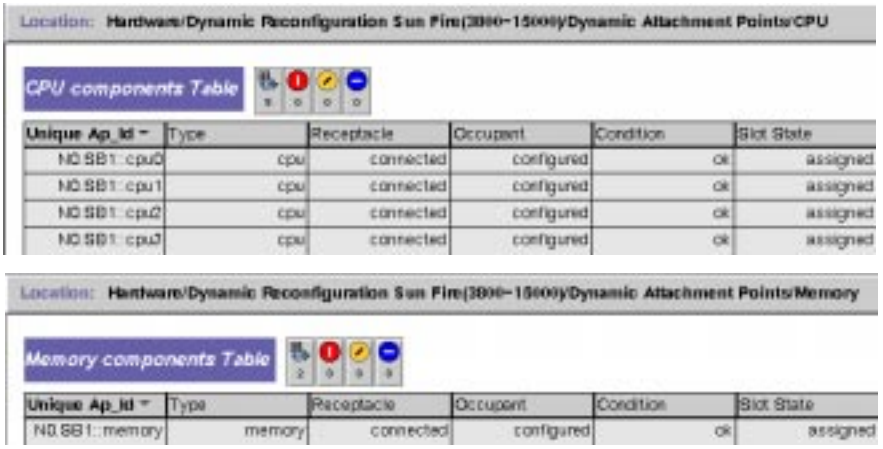

FIGURE 9 Dynamic CPU / Memory Attachment Points

However, with the use of DR and Sun Fire SSC commands, you can disable an individual memory bank or CPU, isolating it from domain usage.

▼ To Remove Individual CPUs and Memory Banks

You can remove individual CPUs and memory banks by using the `disablecomponent` command on the Sun Fire SSC.

1. **Log in to the domain as superuser.**
2. **Use the `cfgadm`(1M) command or the Sun MC DR module to disconnect the CPU/Memory board.**

   This command removes the board from usage by the domain.
3. **Use the `disablecomponent` command on the Sun Fire SSC to disconnect the individual component.**

   You can use the `showcomponent` command on the Sun Fire SSC to verify which components are enable or disabled. The memory controlled by a CPU is not usable when the CPU is disabled.
4. **Use the `cfgadm`(1M) command or the Sun MC DR module to configure the CPU/Memory board back into the domain.**

   All of the components, except those that are disabled, are configured back into the domain.

Note – Using DR, unconfiguring individual CPUs will cause the entire memory on the CPU/Memory board to be lost. Using this procedure, you can minimize the losses of resources when you remove individual CPUs or memory banks. This procedure enables you to remove individual components so that scheduled maintenance can take place.

DR Operations on I/O Assemblies

An I/O assembly (PCI or cPCI) is an attachment point on the Sun Fire 3800–6800 servers. You can configure, unconfigure, connect, or disconnect them. Use the DR module in the Sun MC software to view the assigned status and the condition of the I/O attachment points. FIGURE 10 shows the status and condition of the I/O attachment points on a Sun Fire 6800 server.

FIGURE 10 I/O Attachment Points on a Sun Fire 6800 Server

Connecting and Configuring I/O Assemblies

An I/O assembly must be tested and pass the POST operation before it can be connected and configured into a running domain. The I/O assembly does not have the capability to perform the POST operation on its own. To test an I/O assembly, you need an available domain and a CPU/Memory board.

Certain requirements must be considered before disconnecting an I/O assembly and during the initial configuration phase of the server. For PCI assemblies, each PCI card change requires powering off the PCI assembly and perform a retest using the following procedure. Servers with cPCI assemblies do not require powering off to change the cPCI card. You must verify the need for a power-off because, only then, a retest is required.

▼ To Connect an I/O Assembly to a Running Domain

1. **Log in to the domain as superuser.**
2. **Assign the I/O assembly and a CPU/Memory board to an unused domain by using the** `addboard` **command on the Sun Fire SSC.**
3. **Verify that the boards have the same firmware level as the boards in the target domain.**
4. **Use the** `setkeyswitch` **command to run the POST operation.**

   The level of the POST operation can be set by using the `setupdomain` command. A high POST level must be used to ensure that only known good components are connected or configured into a domain.
5. **After the POST operation completes successfully, use the** `setkeyswitch` **command to set the assembly to standby.**

   The standby position preserves the test state so that the I/O assembly keeps its `OK` state.
6. **Use the** `deleteboard` **command to unassign the I/O assembly from the domain.**
7. **Optionally, use the** `addboard` **command on the Sun Fire SSC, the** `cfgadm`**(1M) command on the domain, or the Sun MC DR module to assign the I/O assembly to the final domain.**
8. **Connect and configure the I/O assembly into the domain.**

Disconnecting and Unconfiguring an I/O Assembly

An I/O assembly looses its POST state whenever it is powered off. For cases when an I/O assembly has to be replaced during a service action, or when a PCI card is removed or added to a PCI assembly, use the previous procedure above to connect and configure the assembly back into the domain. Unconfiguring an I/O device, it must be in an inactive state.

When no hardware changes on the I/O assembly occur or when the I/O assembly is moved from one domain to another, use the `nopoweroff` option during the disconnect process. This option preserves the POST state of the I/O assembly. If an I/O assembly is not powered off, it retains its POST state; it can be connected and configured into a domain without being retested.

You must unconfigure all of the cPCI cards in an assembly before unconfiguring or disconnecting a cPCI assembly. For PCI assemblies, the individual PCI cards are unconfigured as part of the unconfigure-disconnect operation of the assembly.

DR Operations on cPCI Cards

cPCI cards are base attachment points. They support DR operations on individual cards and the high-availability hot-swap model. These features enhance the DR capabilities of cPCI assemblies.

FIGURE 11 shows the base attachment points for cPCI assemblies on Sun Fire 3800–6800 servers.

FIGURE 11 cPCI Base Attachment Points for Sun Fire 3800-6800 Domains

The hot-plug and high-availability hot-swap models are supported. Both models are defined in the PCI specifications.

In the high-availability hot swap model, which is the default for Sun Fire 3800–6800 servers, DR operations are triggered by the insertion or removal of the cPCI card and by the engaging or ejecting of the lever on the cPCI card. This action enables the DR operations on cPCI cards without a system login. A cPCI card can be configured by engaging the lever and can be unconfigured by ejecting the lever. The status of the cPCI card can be seen in the LEDs on the card and the LEDs on the slot. The following tables and code examples show the change in LED status when a card is inserted or removed, and the corresponding `cfgadm`(1M) command output.

TABLE 4 Led status for an Empty Slot and the `cfgadm`(1M) Output

Event	Action	Power LED (Slot)	Failed LED (Slot)	Safe to Remove LED (Slot)	Safe to Remove LED (Card)
Slot empty	None	Off	Off	On	N/A

```
Ap ID              Type     Receptacle  Occupant      Condition
pcisch1:sg9slot0   unknown  empty       unconfigured  unknown
```

TABLE 5 LED States for an Inserted Card and the `cfgadm`(1M) Output

Event	Action	Power LED (Slot)	Failed LED (Slot)	Safe to Remove LED (Slot)	Safe to Remove LED (Card)
Card inserted	Power on and connect	On	Off	Off	On

```
Ap ID              Type     Receptacle  Occupant      Condition
pcisch1:sg9slot0   unknown  connected   unconfigured  unknown
```

TABLE 6 LED States for Lever Engagement and the `cfgadm`(1M) Output

Event	Action	Power LED (Slot)	Failed LED (Slot)	Safe to Remove LED (Slot)	Safe to Remove LED (Card)
Lever engaged	Configure	On	Off	Off	Off

```
Ap ID              Type          Receptacle  Occupant    Condition
pcisch1:sg9slot0   stpcipci/fhs  connected   configured  ok
```

TABLE 7 LED States When the Red Button Is Pressed and the `cfgadm`(1M) Output

Event	Action	Power LED (Slot)	Failed LED (Slot)	Safe to Remove LED (Slot)	Safe to Remove LED (Card)
Red button pressed	Disconnected	On	Off	Off	Off

```
Ap ID              Type     Receptacle    Occupant      Condition
pcisch1:sg9slot0   unknown  disconnected  unconfigured  unknown
```

The following code example shows the console output when the red button is pressed.

```
# Jan 9 20:06:55
# sghsc: NOTICE: sghsco: node 0 / board 9 slot 0 unconfigured
```

TABLE 8 LED States for Card Removal and the `cfgadm`(1M) Output

Event	Action	Power LED (Slot)	Failed LED (Slot)	Safe to Remove LED (Slot)	Safe to Remove LED (Card)
Card removed	None	Off	Off	On	N/A

```
Ap ID              Type     Receptacle  Occupant      Condition
pcisch1:sg9slot0   unknown  empty       unconfigured  unknown
```

In the hot-plug model, DR operations are not triggered by the lever on the cPCI card; instead, the `cfgadm`(1M) command or the Sun MC DR module are used. The following code box lists the usage message for the `cfgadm`(1M) command for cPCI assemblies.

```
# cfgadm -h pci
Usage:
cfgadm [-f] [-y|-n] [-v] [-o hardware_options] -c function ap_id [ap_id...]
cfgadm [-f] [-y|-n] [-v] [-o hardware_options] -x function ap_id [ap_id...]
cfgadm [-v] [-s listing_options] [-o hardware_options] [-a] [-l [ap_id|ap_type...]]
cfgadm [-v] [-o hardware_options] -t ap_id [ap_type...]
cfgadm [-v] [-o hardware_options] -h [ap_id |ap_type...]
PCI hotplug specific commands:
-c [connect|disconnect|configure|unconfigure|insert|remove] ap_id [ap_type...]
-x enable_slot ap_id [ap_type...]
-x disable_slot ap_id [ap_type...]
-x enable_autoconfig ap_id [ap_type...]
-x disable_autoconfig ap_id [ap_type...]
-x led[=[fault|power|active|attn],mode=[on|off|blink]] ap_id [ap_type...]
```

The Sun MC DR module automatically displays the options that are applicable. Using the `enable_autoconfig` or the `disable_autoconfig` commands, you can configure the hot-plug or the high-availability hot-swap model for each cPCI card slot. This technique is useful for debugging or if automatic configuring or unconfiguring cPCI cards is not desired.

Using cPCI versus PCI cards increases the flexibility for changing or servicing I/O paths on the Sun Fire 3800–6800 servers. With the constraints on connecting an I/O assembly to a domain, you must decide before you set up your server if the PCI or cPCI cards fulfill your requirements.

Sample Output From the `cfgadm`(1M) and `showboards` Commands

This section contains several DR scenarios and the outcome of different DR or Sun Fire SSC actions.

State and Condition of Attachment Points

The following is a sample of the state and condition of attachment points.

```
# cfgadm -s “select=class(sbd)”
Ap ID   Type         Receptacle    Occupant      Condition
N0.SB0  CPU_Board    disconnected  unconfigured  unknown
N0.SB2  CPU_Board    connected     configured    ok
N0.IB6  CPCI_I/O_bo  connected     configured    ok
N0.IB8  CPCI_I/O_bo  disconnected  unconfigured  unknown

> showboards
Slot     Pwr  Component Type  State      Status      Domain
----     ---  --------------  -----      ------      ------
SB0      Off  CPU Board       Available  Not tested  Isolated
/N0/SB2  On   CPU Board       Active     Passed      A
/N0/IB6  On   CPCI I/O board  Active     Passed      A
IB8      Off  CPCI I/O board  Available  Not tested  Isolated
```

CODE EXAMPLE 1 State and Condition of Attachment Points

Powering on a System Board

This sample shows the results from powering on SB0. On the Sun Fire SSC, you must use the showboards command to view the state or condition changes.

```
# cfgadm -x poweron N0.SB0
# cfgadm -s "select=class(sbd)"
Ap ID   Type         Receptacle    Occupant      Condition
N0.SB0  CPU_Board    disconnected  unconfigured  unknown
N0.SB2  CPU_Board    connected     configured    ok
N0.IB6  CPCI_I/O_bo  connected     configured    ok
IB8     CPCI_I/O_bo  disconnected  unconfigured  unknown

> showboards
Slot     Pwr  Component Type  State      Status      Domain
----     ---  --------------  -----      ------      ------
SB0      On   CPU Board       Available  Not tested  Isolated
/N0/SB2  On   CPU Board       Active     Passed      A
/N0/IB6  On   CPCI I/O board  Active     Passed      A
IB8      Off  CPCI I/O board  Available  Not tested  Isolated
```

CODE EXAMPLE 2 Results From Powering on a System Board

Configuring a System Board

This sample shows the results from configuring `SB0`. The board is now connected, configured, and in the OK state-condition. The POST level is specified by using the `-o` option. The POST output is displayed on the domain console on the Sun Fire SSC. After the POST operation completes successfully, the status of the attachment point changes.

```
# cfgadm -o platform=dag=default -c configure N0.SB0
# cfgadm -s “select=class(sbd)”
Ap ID   Type         Receptacle    Occupant      Condition
N0.SB0  CPU_Board    connected     configured    ok
N0.SB2  CPU_Board    connected     configured    ok
N0.IB6  CPCI_I/O_bo  connected     configured    ok
IB8     CPCI_I/O_bo  disconnected  unconfigured  unknown

> showboards
Slot     Pwr  Component Type  State      Status      Domain
----     ---  --------------  -----      ------      ------
/N0/SB0  On   CPU Board       Active     Passed      A
/N0/SB2  On   CPU Board       Active     Passed      A
/N0/IB6  On   CPCI I/O board  Active     Passed      A
IB8      Off  CPCI I/O board  Available  Not tested  Isolated
```

CODE EXAMPLE 3 Results From Configuring a System Board

Disconnecting a System Board

This sample shows the results from disconnecting SB2. The board is automatically powered off after the DR operation, and it is ready to be removed. You can also connect or configure it to another domain.

```
# cfgadm -o unassign -c disconnect N0.SB2
# cfgadm -s “select=class(sbd)”
Ap ID   Type         Receptacle    Occupant      Condition
N0.SB0  CPU_Board    connected     configured    ok
N0.SB2  CPU_Board    disconnected  unconfigured  unknown
N0.IB6  CPCI_I/O_bo  connected     configured    ok
IB8     CPCI_I/O_bo  disconnected  unconfigured  unknown

> showboards
Slot     Pwr  Component Type  State      Status      Domain
----     ---  --------------  -----      ------      ------
/N0/SB0  On   CPU Board       Active     Passed      A
SB2      Off  CPU Board       Available  Not tested  Isolated
/N0/IB6  On   CPCI I/O board  Active     Passed      A
IB8      Off  CPCI I/O board  Available  Not tested  Isolated
```

CODE EXAMPLE 4 Results From Disconnecting a System Board

Updating the ACLs

This sample shows the results from updating the ACLs on the Sun Fire SSC. The output of the `cfgadm`(1M) command reflects the removal of SB2 from the ACL for domain A.

```
# cfgadm -s “select=class(sbd)”
Ap ID   Type         Receptacle    Occupant      Condition
N0.SB0  CPU_Board    connected     configured    ok
N0.IB6  CPCI_I/O_bo  connected     configured    ok
N0.IB8  CPCI_I/O_bo  disconnected  unconfigured  unknown

> setupplatform -p acl
ACL for domain A [SB0 SB2 IB6 IB8]:-d sb2
ACL for domain B [SB0 SB2 IB6 IB8]:
ACL for domain C [SB0 SB2 IB6 IB8]:

> showplatform -p acl
ACL for domain A [SB0 IB6 IB8]:
ACL for domain B [SB0 SB2 IB6 IB8]:
ACL for domain C [SB0 SB2 IB6 IB8]:
```

CODE EXAMPLE 5 Results From Updating the ACLs

DR Best Practices

This section contains general guidelines and specific considerations you must remember before using the DR operations on the Sun Fire 3800–6800 servers.

General Guidelines

This section contains general guidelines you must follow when you are executing DR commands.

Using the cfgadm(1M) Command or the Sun MC Software for DR

You can use the cfgadm(1M) command on the domain or the DR module in the Sun MC software to perform DR operations. Normally, the granularity offered by the cfgadm(1M) command is not needed for day-to-day business needs of the DR software. Use the DR module in the Sun MC software for routine DR operations.

Using a Slot for the Boot Device

You should use the first host bus adapter in the OpenBoot PROM probe list to access the boot disk. This practice ensures that the boot path is fixed and will not change if the I/O cards are added to the system and the /etc/path_to_inst file is recreated during a boot operation (for example, boot -ra). The device tree structure to the boot device also remains fixed when the domain is booted from a CD-ROM drive or a networked software image.

Labeling Boot Devices

You should label the boot disk and the boot mirror disk by using the format(1M) command with the volname option. This technique enables you to easily identify these disks. The following format(1M) command output shows disk 0 is labeled as bootdisk, and disk 4 is labeled mir-disk.

```
# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
0. c0t0d0<SUN18G cyl7506 alt2 hd19 sec248> bootdisk
/ssm@0,0/pci@18,700000/pci@1/SUNW,isptwo@4/sd@0,0
1. c1t0d0<SUN18G cyl7506 alt2 hd19 sec248>
/ssm@0,0/pci@18,700000/pci@1/SUNW,isptwo@4/sd@0,0
2. c2t0d0<SUN18G cyl7506 alt2 hd19 sec248>
/ssm@0,0/pci@18,700000/pci@1/SUNW,isptwo@4/sd@0,0
3. c3t0d0<SUN18G cyl7506 alt2 hd19 sec248>
/ssm@0,0/pci@18,700000/pci@1/SUNW,isptwo@4/sd@0,0
4. c4t0d0<SUN18G cyl7506 alt2 hd19 sec248> mir-disk
/ssm@0,0/pci@1a,700000/pci@1/SUNW,isptwo@4/sd@0,0
Specify disk (enter its number):
```

Note – Changing or adding labels can be done when partitions are in use.

Path Independence

You need two or more paths to the same resources in order to use DR to service the I/O paths and still have access to the resources. The system needs an alternate path to the resources in case a single-point-of-failure occurs in the path. These paths must be as independent as possible.

For storage devices, you must use different I/O assemblies to host each switch and interface. On the storage device, you must have different I/O interfaces or channels. For complex multivendor storage solutions, you must verify the hardware dependencies so that the entire solution provides the needed independance.

Host Bus Adapters for Multipathed Devices

If you configure multipathed devices into a domain, all of the primary paths must be on the same I/O assembly. All of the secondary, or alternate, paths should be on the same I/O assembly, if possible. This practice ensures that a failure on a primary I/O assembly causes the system to fail over to the alternate path.

Application Configuration for Quiesce

You must configure applications so that they support a system quiesce. In client-server environments, set the client timeout values appropriately to accommodate a quiesce of a server. You must configure clients to respond to the `SIGHUP` signal, which is sent out after a quiesce.

Using cPCI Instead of PCI

The cPCI solution allows greater flexibility in making configuration changes. Individual cPCI cards can be unconfigured or disconnected. All of the PCI cards in a PCI assembly must be unconfigured or disconnected. The cPCI boards also support the high-availability hot-swap model. They can be unconfigured or configured without having a system login. Your decision on the use of the cPCI option should be made before you order the system.

System Testing DR Operations

You should test all of the DR operations before you put the system into your production network.

Documenting DR Operations

DR is normally used for system upgrades, changes, or maintenance. You should update your system runbooks to include all of the DR steps and special information for quick DR decisions.

The Reconfiguration Manager (RCM) provides a framework to integrate application dependencies into unconfigure and disconnect operations. Based on user defined scripts, unconfigure-disconnect operations on devices can be blocked. Application changes before unconfiguring and/or disconnecting a device can be set up. Using RCM simplifies automating and configuring user application DR dependencies (refer to the `rcmscript`(1M) man page for more information).

Specific Considerations for the Sun Fire 3800–6800 Servers

This section contains specific considerations for the Sun Fire 3800–6800 servers. The topics include domain set-up, access control lists, memory placement, POST level, power-off options for I/O assemblies, and firmware levels.

Domain Set-Up

You should always have at least two domains on the Sun Fire 3800–6800 servers. An available domain is required to perform the POST operation on the I/O assemblies. If you disconnect an I/O assembly on a server without an available domain, you will not be able to connect it to another running domain because you will not be able to test the assembly.

On the Sun Fire 6800 server, two domains can be created in a single segment mode, meaning that a free domain would not exist. To dynamically connect an I/O assembly on the Sun Fire 6800 server, the server must be configured in dual segment mode if two domains are set up.

Access Control List

The access control list (ACL) on the Sun Fire SSC prevents uncontrolled access to unassigned or available resources. It also ensures a domain has access only to its preallocated resources.

FIGURE 12 shows an ACL for a domain on a Sun Fire 6800 server. The domain has access to the CPU/Memory boards `SB1`, `SB2`, and `SB5`, as well as access to the I/O assemblies `IB7` and `IB9`. The other CPU/Memory boards and I/O assemblies on the system, that are not in the ACL for this domain, cannot be connected or configured in the domain, even when they are not being used by other domains.

FIGURE 12 Access Control List for a Domain

Memory Placement

The memory in the domain should be placed so that a minimum of CPU/Memory boards contain permanent memory and so that the requirements for the copy-rename mechanisms are fulfilled for minimum number of boards. Memory layouts can be divided into evenly and unevenly spread configurations. The layout you use will impact DR operations differently for memory operations.

Even Placement

The advantage of evenly placing the memory across all of the CPU/Memory boards is that every CPU/Memory board can be used as a target for a copy-rename operation.

The disadvantage is that more than one CPU/Memory board may contain permanent memory, which causes the copy-rename mechanism to apply to more than one CPU/Memory board.

Uneven Placement

The advantage of unevenly placing the memory across the CPU/Memory boards is that it can minimize the amount of CPU/Memory boards containing permanent memory. The copy-rename mechanism then only applies to those boards containing permanent memory.

The disadvantage is that only specific CPU/Memory boards can be used as a target of a copy-rename operation.

You may, however, prefer to use an even approach because the kernel, and therefore the permanent memory, grows dynamically.

Specifying the POST Level

You should specify the POST level on connect and configuration operations. When you add new hardware, the highest level of POST must be used so that you know only good components are connected.

To change the POST level, use the `-o` and `-x` options with the `cfgadm`(1M) command, as shown in the following code example.

```
# cfgadm -s “select=class(sbd)”
Ap ID   Type         Receptacle    Occupant      Condition
N0.IB6  PCI_I/O_boa  disconnected  unconfigured  unknown
N0.IB8  CPCI_I/O_bo  connected     configured    ok
N0.SB0  CPU_Board    disconnected  unconfigured  unknown
N0.SB2  CPU_Board    connected     configured    ok

# cfgadm -o platform=diag=default -c configure N0.SB0
# cfgadm -s “select=class(sbd)”
Ap ID   Type         Receptacle    Occupant      Condition
N0.IB6  PCI_I/O_boa  disconnected  unconfigured  unknown
N0.IB8  CPCI_I/O_bo  connected     configured    ok
N0.SB0  CPU_Board    connected     configured    ok
N0.SB2  CPU_Board    connected     configured    ok
```

CODE EXAMPLE 6 Changing the POST Level

Using the `NOPOWEROFF` Option

If I/O assemblies are moved between domains and no hardware is changed, you can use the `nopoweroff` option to ensure that the boards retain their POST status. In FIGURE 13, you can see the condition of the `IB7` I/O assembly. It is OK after being disconnected and unassigned from the domain. You can connect or configure `IB7` to another domain without having to test it.

FIGURE 13 I/O Attachment Points After Using the `nopoweroff` Option

Firmware Level

Any new resource, CPU/Memory board or I/O assembly, must have the same level of firmware as all of the other boards and assemblies in the domain.

About the Author

Peter Gonscherowski is a member of the Sun CPR Engineering Americas Group. He is also a member of the team responsible for providing engineering support for Sun midrange servers. Prior to his current role, Peter worked as a mission critical engineer and as a backline engineer for Sun Enterprise Services in Europe.

Bibliography and Recommended Reading

The following documents are at `http://docs.sun.com`:

- *Solaris 8 Reference Manual Collection*:
  - `cfgadm`(1M)
  - `cfgadm_pci`(1M)
  - `cfgadm_sb`(1M)
  - `libcfgadm`(3)
  - `modinfo`(1M)
  - `rcmscript`(4)
- *Sun Fire 3800-6800 Service Manual* (805-7363)
- *Sun Fire 3800-6800 System Controller Command Reference Manual* (805-7372)
- *Sun Fire 3800-6800 Systems Dynamic Reconfiguration Users Guide* (806-6783)
- *Sun Fire 3800-6800 Systems Platform Administration Manual* (805-7373)
- *Sun Management Center 3.0 Software Supplemental for Sun Fire 3800-6800 Systems* (806-5948)
- *Writing Device Drivers* (805-7378)

The following document is available at bookstores:

- T. Shanley and D. Anderson, *PCI System Architecture*, October 2000