Solaris™ Operating System Availability Features

Thomas M. Chalfant, Enterprise Server Group

Sun BluePrints™ OnLine—May 2004

http://www.sun.com/blueprints

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95045 U.S.A.
650 960-1300

Part No. 817-7039-10
Revision 1.0, 5/19/04
Edition: May 2004

Please

Adobe PostScript

Solaris Operating System Availability Features

The Solaris™ Operating System (OS) includes features aimed at enhancing availability by helping the system react better to certain types of normally occurring error conditions. These features have been improved in recent kernel updates. Solaris 8 Kernel Update patch 108528-20 and Solaris 9 Kernel Update patch 112233-06 introduce an enhanced L2 Cache error-handling technique called Processor Offlining. Solaris 8 Kernel Update patch 108528-24 and Solaris 9 Kernel Update patch 112233-11 introduced an enhanced memory DIMM error-handling technique called Page Retirement. Subsequent kernel update patches have modified and will continue to modify the behavior of the initial implementations. This article discusses the availability features implemented as of the release of Solaris 8 Kernel Update patch 117000-03 and Solaris 9 Kernel Update patch 112233-12. Where necessary, notes are provided to indicate where features of prior revisions differ. The Solaris 8 Kernel Update underwent a patch rejuvenation process, which changed its base patch number from 108528 to 117000. See Sun℠ Alert Notification 57489 for further details.

The article addresses the following topics:

- "Processor Offlining for L2 Cache Events" on page 2
- "Page Retirement" on page 13

This article is targeted at IT professionals interested in detailed technical information regarding the covered topics. Basic knowledge of memory and processor architecture is assumed.

Processor Offlining for L2 Cache Events

Processor offlining is implemented through the fixes to bug IDs 4740766 and 4740769. The behavior is further modified through the fixes to bug IDs 4832104, 4836134, 4846476, and 4833032. When an L2 Cache error checking and correcting (ECC) event is logged, the event specifics are also examined to see if the event meets the criteria for offlining the processor. Only UltraSPARC® III and UltraSPARC IV based systems have these changes, since the implementation is contained within the kernel code specific to those processors. For UltraSPARC IIIi systems, this feature is turned off by default, and for others it is turned on by default. In either case, the default can be changed through entries in the `/etc/system` file. Qualifying ECC events fall into three major categories:

- Single-bit correctable L2 Cache events
- UCC event with ME bit set
- Uncorrectable L2 Cache events

Category A: Single-Bit Correctable L2 Cache Events

There are four L2 Cache correctable events in this category:

- UCC – Software-correctable L2 Cache ECC error for instruction fetch or data access other than block load.
- CPC – Hardware-corrected L2 Cache ECC error for copyout (snoop request).
- WDC – Hardware-corrected L2 Cache ECC error for writeback.
- EDC – Hardware-corrected L2 Cache ECC error for store merge or block load. For UltraSPARC III Cu systems, a hardware-corrected L2 Cache ECC error for software or hardware prefetch access also generates EDC.

For each of these events the error is corrected, allowing the thread taking the trap to be restarted. However, repeated events make the CPU and its associated L2 Cache candidates for replacement. The event timestamp is fed into a Soft Error Rate Discrimination (SERD) algorithm that detects when three distinct events have occurred on the same processor in a 24-hour period. Upon detection of the third qualifying event, the processor becomes a candidate for offlining.

Category B: UCC Event With ME Bit Set

Prior to Solaris 8 Kernel Update patch 117000-01 and Solaris 9 Kernel Update patch 112233-12, the special combination of a UCC event with the multiple error (ME) bit set was treated as if three distinct UCC events, as described above, occurred in very rapid succession. In this case, the SERD algorithm was short-circuited, and the processor immediately became a candidate for offlining. Solaris 8 Kernel Update patch 117000-01 and Solaris 9 Kernel Update patch 112233-12 contain the fix to bug ID 4875077. This bug ID identifies a condition where a single bit flip could cause the ME bit to be set. The new behavior is to treat UCC+ME as a Category A type event, and not treat it as a special category by itself. It is no longer recommended to enable Category B offlining. You can disable the Category B processor offlining event using a tunable in the `/etc/system` file, as described in TABLE 1.

Category C: Uncorrectable L2 Cache Events

There are four L2 Cache uncorrectable events in this category:

- UCU – Uncorrectable L2 Cache ECC error for instruction fetch or data access other than block load.
- CPU – Uncorrectable L2 Cache ECC error for copyout (snoop request).
- WDU – Uncorrectable L2 Cache ECC error for writeback (victimization).
- EDU – Uncorrectable L2 Cache ECC error for store merge or block load. For UltraSPARC III Cu systems, an uncorrectable L2 Cache ECC error detected during a software or hardware prefetch access also generates EDU.

For each of the above events, the thread taking the trap can be restarted, but the processor becomes an immediate candidate for offlining. However, there are special syndromes which, depending upon the CPU type, can result in not offlining the processor. In these cases, the CPU that has discovered the error is not necessarily the offending CPU. For UltraSPARC IIIi systems, there is just one special syndrome: 0x3. For other UltraSPARC III based systems the special syndromes are 0x3, 0x71, and 0x11c.

Note that Solaris 8 Kernel Update patches prior to 108528-24 and Solaris 9 Kernel Update patches prior to 112233-09 do not consider the WDU and EDU events candidates for processor offlining. Therefore, no processor offlining is performed for WDU and EDU events in the specified kernel update releases. It is also possible in these kernel updates for CPUs to be offlined incorrectly due to memory UE and DUE errors (bug ID 4836134). If you cannot upgrade to at least Solaris 8 Kernel Update patch 108528-24 or Solaris 9 Kernel Update patch 112233-09, for this latter issue you can alternatively disable the Category C class of processor offlining events using a tunable in the `/etc/system` file. See TABLE 1 for details.

Processor Offlining Method

After a processor becomes a candidate for offlining, attempts are made to take the processor offline using the common OS interface `cpu_offline()`. This is similar in effect to running the command `psradm` and is subject to the same restrictions as to when and if it will be successful.

If the error logging (and therefore this code) happens to be executing on the candidate processor, an attempt is made to cause a different processor to perform the offlining, if one is available. If the offlining attempt fails, the attempt is repeated after waiting for an interval of time. For instance, an attempt might fail because the processor has threads running on it that cannot be moved immediately to another processor, or because it is the only CPU in a processor set. If interrupts are assigned to this processor that cannot be moved to another, the processor might still run interrupt handler code and therefore be subject to future events. If the number of offlining attempts exceeds a set limit, the algorithm stops trying and the processor is not offlined at that time. It remains eligible for future offlining consideration should additional qualifying L2 Cache ECC events occur. Both the maximum number of attempts and the interval between them are tunables that may be set in the `/etc/system` file. If a system administrator puts an offlined processor back online (for example, using `psradm`), it again becomes subject to this algorithm, just as any other processor in the system.

Once it is decided to offline a processor using these algorithms, a processor indictment message is sent to the system controller on Sun Fire™ hardware systems with the intention of allowing it to remove the processor from the configuration at the next `POST/reboot.` System controller software has been released for the following systems for to provide support for this feature and others. On midrange systems such as Sun Fire 6800, E6900, 4810, 4800, E4900, 3800, E2900, and V1280 servers, refer to the Sun BluePrints™ OnLine article entitled ”Sun Fire Hardware Midrange Servers System Auto Diagnosis and Recovery Enhancements” for more information. On high-end systems such as Sun Fire 15K/12K servers, refer to the Sun BluePrints OnLine article entitled "SunFire 15K/12K Auto Diagnosis and Recovery" for more information. On systems without the appropriate system controller software and kernel update patch, the resultant behavior is that the offlined processor will be part of the configuration again at the next `POST/reboot`, assuming no errors are encountered during that process.

Note – Solaris 8 Kernel Update patches prior to 108528-24 and Solaris 9 Kernel Update patches prior to 112233-09 do not send processor indictments to the system controller.

Processor Offlining and Dynamic Reconfiguration

Processor offlining does not utilize dynamic reconfiguration (DR). Automatic DR of an entire system board is not attempted when a single processor, or even all processors, are offlined on a board. If a system administrator manually performs DR on a system board containing offlined processors out of a domain and into the same or another domain, the processors are active again if the `POST` process is successful.

Processor Offlining and Capacity on Demand

There is no interaction between Capacity on Demand (COD) and processor offlining. The interaction remains the same as that seen if the system administrator had manually offlined a processor using the `psradm` command.

Example Messaging

The offlining algorithm has three possible scenarios. It could:

- Successfully offline a processor
- Initially fail but try again
- Give up after some number of failures (described in "Processor Offlining Method" on page 4)

A distinct message is output to the console and the system messages log file for each of these scenarios. All messages identify both the affected processor and the general category that prompted the attempt to offline it. This information is subsequently useful to the service personnel who are eventually called to replace the offlined hardware.

For each of the three categories defined above (A, B, and C), the following examples show the resulting messages from the offlining algorithm.

Category A Messages

Category A messages are demonstrated below.

```
Oct 23 11:37:05 sf15k-domc SUNW,UltraSPARC-III: [ID 962502
kern.info] NOTICE: [AFT0] WDC Event detected by CPU64 at TL=0,
errID 0x000000e3.2774aaf0
Oct 23 11:37:05 sf15k-domc     AFSR 0x00000040<WDC>.000001f0 AFAR
0x000000a1.f06b4000
Oct 23 11:37:05 sf15k-domc     Fault_PC 0x10152880 Esynd 0x01f0
SB2/P0/E0 J4400
Oct 23 11:37:05 sf15k-domc SUNW,UltraSPARC-III: [ID 189302
kern.info] [AFT0] errID 0x000000e3.2774aaf0 Data Bit 111 was in
error and corrected
Oct 23 11:37:05 sf15k-domc SUNW,UltraSPARC-III: [ID 860712
kern.info] [AFT2] errID 0x000000e3.2774aaf0 E$tag
PA=0x000000a0.00eb4000 does not match AFAR=0x000000a1.f06b4000
Oct 23 11:37:05 sf15k-domc SUNW,UltraSPARC-III: [ID 260100
kern.info] [AFT2] errID 0x000000e3.2774aaf0
PA=0x000000a0.00eb4000
Oct 23 11:37:05 sf15k-domc    E$tag 0x00000140.01000001 E$state_0
Shared
Oct 23 11:37:05 sf15k-domc SUNW,UltraSPARC-III: [ID 895151
kern.info] [AFT2] E$Data (0x00) 0x00000000.00000000
0x00000000.00000000 ECC 0x000
Oct 23 11:37:05 sf15k-domc SUNW,UltraSPARC-III: [ID 895151
kern.info] [AFT2] E$Data (0x10) 0x00000000.00000000
0x00000000.00000000 ECC 0x000
Oct 23 11:37:05 sf15k-domc SUNW,UltraSPARC-III: [ID 895151
kern.info] [AFT2] E$Data (0x20) 0x00000000.00000000
0x00000000.00000000 ECC 0x000
Oct 23 11:37:05 sf15k-domc SUNW,UltraSPARC-III: [ID 895151
kern.info] [AFT2] E$Data (0x30) 0x00000000.00000000
0x00000000.00000000 ECC 0x000
Oct 23 11:37:05 sf15k-domc SUNW,UltraSPARC-III: [ID 929717
kern.info] [AFT2] D$ data not available
Oct 23 11:37:05 sf15k-domc SUNW,UltraSPARC-III: [ID 335345
kern.info] [AFT2] I$ data not available
```

Upon encountering a second WDC event, messages are logged similar to those seen during the first WDC event.

```
Oct 23 11:38:05 sf15k-domc SUNW,UltraSPARC-III: [ID 137784
kern.info] NOTICE: [AFT0] WDC Event detected by CPU64 at TL=0,
errID 0x000000f1.217a63c1
.
.
.
```

Upon encountering the third WDC event, you see the normal messages just as in the first and second events, along with the following additional messages.

```
Oct 23 11:39:37 sf15k-domc SUNW,UltraSPARC-III: [ID 709559
kern.info] NOTICE: [AFT0] WDC Event detected by CPU64 at TL=0,
errID 0x00000106.a62214d3
.
.
.
Oct 23 11:39:38 sf15k-domc SUNW,UltraSPARC-III: [ID 732650
kern.notice] NOTICE: [AFT1] Failed to offline CPU64 due to more
than 2 xxC Events in 24:00:00 (hh:mm:ss), will try again
```

In this case, the CPU could not be immediately offlined. The system retries some number of seconds later.

```
Oct 23 11:40:08 sf15k-domc SUNW,UltraSPARC-III: [ID 915404
kern.notice] NOTICE: [AFT1] CPU64 offlined due to more than 2 xxC
Events in 24:00:00 (hh:mm:ss)
```

Now the CPU has been successfully offlined.

Category B Messages

Category B messages are demonstrated below.

```
Oct 23 18:58:20 sf68-doma SUNW,UltraSPARC-III+: [ID 135428
kern.info] NOTICE: [AFT0] First Error UCC Event detected by CPU5
in Privileged mode at TL=0, errID 0x0001487b.327ee8b0
Oct 23 18:58:20 sf68-doma     AFSR 0x00100400<PRIV,UCC>.00000031
AFAR
0x00000000.02c4e8f0
Oct 23 18:58:20 sf68-doma    Fault_PC 0x104e860 Esynd 0x0031 /N0/
SB1/P1/E1 J5300
Oct 23 18:58:20 sf68-doma SUNW,UltraSPARC-III+: [ID 782357
kern.info] [AFT0] errID 0x0001487b.327ee8b0 Data Bit 40 was in
error and corrected
Oct 23 18:58:20 sf68-doma SUNW,UltraSPARC-III+: [ID 669499
kern.info] [AFT2] errID 0x0001487b.327ee8b0
PA=0x00000000.02c4e8c0
Oct 23 18:58:20 sf68-doma     E$tag 0x00000000.0b249249
E$state_3 Shared
Oct 23 18:58:20 sf68-doma SUNW,UltraSPARC-III+: [ID 895151
kern.info] [AFT2] E$Data (0x00) 0x80a0a000.32680024
0xc45fa7f7.7ffff353 ECC 0x067
Oct 23 18:58:20 sf68-doma SUNW,UltraSPARC-III+: [ID 895151
kern.info] [AFT2] E$Data (0x10) 0x90100003.c85fa7ef
0x1080001b.c45fa7f7 ECC 0x1e8
Oct 23 18:58:20 sf68-doma SUNW,UltraSPARC-III+: [ID 895151
kern.info] [AFT2] E$Data (0x20) 0xc45fa7f7.86100000
0x90100004.92100012 ECC 0x12b
Oct 23 18:58:20 sf68-doma SUNW,UltraSPARC-III+: [ID 895151
kern.info] [AFT2] E$Data (0x30) 0x94100011.8778b401
0x9938e000.7ffff1fb ECC 0x066
Oct 23 18:58:20 sf68-doma SUNW,UltraSPARC-III+: [ID 929717
kern.info] [AFT2] D$ data not available
Oct 23 18:58:20 sf68-doma SUNW,UltraSPARC-III+: [ID 335345
kern.info] [AFT2] I$ data not available
Oct 23 18:58:31 sf68-doma SUNW,UltraSPARC-III+: [ID 391224
kern.info] NOTICE: [AFT0] UCC Event detected by CPU5 in Privileged
mode at TL=0, errID 0x0001487b.327ee8b0
Oct 23 18:58:31 sf68-doma    AFSR 0x00300400<ME,PRIV,UCC>.00000031
AFAR
0x00000000.02c4e8f0
Oct 23 18:58:31 sf68-doma     Fault_PC 0x104e860 Esynd 0x0031 /N0/
SB1/P1/E1 J5300
Oct 23 18:58:31 sf68-doma SUNW,UltraSPARC-III+: [ID 782357
kern.info] [AFT0] errID 0x0001487b.327ee8b0 Data Bit 40 was in
error and corrected
Oct 23 18:58:31 sf68-doma SUNW,UltraSPARC-III+: [ID 669499
```

```
kern.info] [AFT2] errID 0x0001487b.327ee8b0
PA=0x00000000.02c4e8c0
Oct 23 18:58:31 sf68-doma     E$tag 0x00000000.0b249249 E$state_3
Shared
Oct 23 18:58:31 sf68-doma SUNW,UltraSPARC-III+: [ID 895151
kern.info] [AFT2] E$Data (0x00) 0x80a0a000.32680024
0xc45fa7f7.7ffff353 ECC 0x067
Oct 23 18:58:31 sf68-doma SUNW,UltraSPARC-III+: [ID 895151
kern.info] [AFT2] E$Data (0x10) 0x90100003.c85fa7ef
0x1080001b.c45fa7f7 ECC 0x1e8
Oct 23 18:58:31 sf68-doma SUNW,UltraSPARC-III+: [ID 895151
kern.info] [AFT2] E$Data (0x20) 0xc45fa7f7.86100000
0x90100004.92100012 ECC 0x12b
Oct 23 18:58:31 sf68-doma SUNW,UltraSPARC-III+: [ID 895151
kern.info] [AFT2] E$Data (0x30) 0x94100011.8778b401
0x9938e000.7ffff1fb ECC 0x066
Oct 23 18:58:31 sf68-doma SUNW,UltraSPARC-III+: [ID 929717
kern.info] [AFT2] D$ data not available
Oct 23 18:58:31 sf68-doma SUNW,UltraSPARC-III+: [ID 335345
kern.info] [AFT2] I$ data not available
Oct 23 18:58:32 sf68-doma SUNW,UltraSPARC-III+: [ID 489146
kern.notice] NOTICE: [AFT1] CPU5 offlined due to UCC Event with ME
set
```

Category C Messages

Category C messages are demonstrated below.

```
Oct 23 11:42:31 sf15k-domc SUNW,UltraSPARC-III: [ID 798832
kern.warning] WARNING: [AFT1] WDU Event detected by CPU65 at TL=0,
errID 0x0000012f.1b1a7f3a
Oct 23 11:42:31 sf15k-domc     AFSR 0x00000020<WDU>.0000017a AFAR
0x000000a1.d9698000
Oct 23 11:42:31 sf15k-domc     Fault_PC 0x10152880 Esynd 0x017a
SB2/P1/E0 J5400
Oct 23 11:42:31 sf15k-domc SUNW,UltraSPARC-III: [ID 990173
kern.notice] [AFT1] errID 0x0000012f.1b1a7f3a Two Bits were in
error
Oct 23 11:42:31 sf15k-domc SUNW,UltraSPARC-III: [ID 974137
kern.info] [AFT2] errID 0x0000012f.1b1a7f3a E$tag
PA=0x000000a0.00e98000 does not match AFAR=0x000000a1.d9698000
Oct 23 11:42:31 sf15k-domc SUNW,UltraSPARC-III: [ID 264775
kern.info] [AFT2] errID 0x0000012f.1b1a7f3a
PA=0x000000a0.00e98000
Oct 23 11:42:31 sf15k-domc    E$tag 0x00000140.01000001 E$state_0
Shared
Oct 23 11:42:31 sf15k-domc SUNW,UltraSPARC-III: [ID 895151
kern.info] [AFT2] E$Data (0x00) 0x00000000.00000000
0x00000000.00000000 ECC 0x000
Oct 23 11:42:31 sf15k-domc SUNW,UltraSPARC-III: [ID 895151
kern.info] [AFT2] E$Data (0x10) 0x00000000.00000000
0x00000000.00000000 ECC 0x000
Oct 23 11:42:31 sf15k-domc SUNW,UltraSPARC-III: [ID 895151
kern.info] [AFT2] E$Data (0x20) 0x00000000.00000000
0x00000000.00000000 ECC 0x000
Oct 23 11:42:31 sf15k-domc SUNW,UltraSPARC-III: [ID 895151
kern.info] [AFT2] E$Data (0x30) 0x00000000.00000000
0x00000000.00000000 ECC 0x000
Oct 23 11:42:31 sf15k-domc SUNW,UltraSPARC-III: [ID 929717
kern.info] [AFT2] D$ data not available
Oct 23 11:42:31 sf15k-domc SUNW,UltraSPARC-III: [ID 335345
kern.info] [AFT2] I$ data not available
Oct 23 11:42:31 sf15k-domc unix: [ID 321153 kern.notice] NOTICE:
Scheduling clearing of error on page 0x000000a1.d9698000
Oct 23 11:42:32 sf15k-domc SUNW,UltraSPARC-III: [ID 277554
kern.notice] NOTICE: [AFT1] Failed to offline CPU65 due to xxU
Event, will try again
Oct 23 11:42:37 sf15k-domc unix: [ID 221039 kern.notice] NOTICE:
Previously reported error on page 0x000000a1.d9698000 cleared
```

In this case the CPU could not be immediately offlined. The system retries some number of seconds later.

```
Oct 23 11:43:02 sf15k-domc SUNW,UltraSPARC-III: [ID 966792
kern.notice] NOTICE: [AFT1] CPU65 offlined due to xxU Event
```

Now the CPU has been successfully offlined.

Due to reasons stated previously, it is possible that a CPU cannot be offlined at all. The following messages depict this situation.

```
Oct 23 15:41:17 sf15k-domc SUNW,UltraSPARC-III: [ID 277554
kern.notice] NOTICE: [AFT1] Failed to offline CPU64 due to xxU
Event, will try again
Oct 23 15:41:23 sf15k-domc SUNW,UltraSPARC-III: [ID 277554
kern.notice] NOTICE: [AFT1] Failed to offline CPU64 due to xxU
Event, will try again
Oct 23 15:41:53 sf15k-domc last message repeated 6 times
Oct 23 15:41:58 sf15k-domc SUNW,UltraSPARC-III: [ID 277554
kern.notice] NOTICE: [AFT1] Failed to offline CPU64 due to xxU
Event, will try again
Oct 23 15:43:14 sf15k-domc last message repeated 15 times
Oct 23 15:43:19 sf15k-domc SUNW,UltraSPARC-III: [ID 324082
kern.notice] NOTICE: [AFT1] Failed to offline CPU64 due to xxU
Event, giving up
```

This system was configured to attempt to offline the processor 24 times before giving up.

Tunables

The following `/etc/system` variables and their possible values are listed here for reference only. Changing the variables to values other than their defaults should only be done under the guidance of an authorized Sun Microsystems service provider.

TABLE 1 Processor Offlining Variables and Values

Variables	Values
`set automatic_cpu_removal=0`	Disables processor offlining.
`set automatic_cpu_removal=1`	Enables only Category A offlining.
`set automatic_cpu_removal=2`	Enables only Category B offlining. It is no longer recommended to enable category B offlining.
`set automatic_cpu_removal=4`	Enables only Category C offlining.
`set automatic_cpu_removal=5`	Enables both Category A and C but not Category B offlining. Note: The values are expressed in decimal format, but if you consider the variable value in binary format, it is formed by ORing together three bits. A value of `1` in binary is `001`, `2` is `010`, and `4` is `100`. To get both Category A and C, set the bit positions for both `1` and `4`, which gives `101` in binary or `5` in decimal. As of Solaris 8 Kernel Update patch 117000-01 and Solaris 9 Kernel Update patch 112233-12, if processor offlining is turned on for a specific processor type, this is the default.

TABLE 1 Processor Offlining Variables and Values *(Continued)*

Variables	Values
`set automatic_cpu_removal=7`	Enables all three categories of offlining. Prior to Solaris 8 Kernel Update patch 117000-01 and Solaris 9 Kernel Update patch 112233-12, if processor offlining was turned on for a specific processor type, this was the default.
`set cpu_remove_retry_seconds=30` `set cpu_remove_retry_attempts=2400`	When processor offlining is unsuccessful, retry again in `cpu_remove_retry_seconds`. Additionally, keep trying for `cpu_remove_retry_attempts`. Note: These are the default values for Solaris 8.
`set cpu_remove_retry_seconds=5` `set cpu_remove_retry_attempts=24`	When processor offlining is unsuccessful, retry again in `cpu_remove_retry_seconds`. Additionally, keep trying for `cpu_remove_retry_attempts`. Note: These are the default values for Solaris 9.

Page Retirement

Solaris maintains a list of pages not being used. This list of pages is known as the *freelist*. When a new page is needed, one is taken off the freelist. To remove a page from use permanently, it is sufficient to ensure that when it is no longer being used by the system, it is not returned to the freelist. Essentially, it will remain unused until the system is rebooted and the OS starts over with a new freelist. When a page is no longer being used, the OS calls the function `page_free()` on that page.

This section discusses some of the reasons for page retirement, and how faulty pages are handled.

Page Retirement for Correctable DIMM Errors

Page Retirement is implemented through the fixes to bug IDs 4484338, 4504686, 4880360, and 4915531. A memory DIMM which is experiencing repeated correctable (single-bit) errors might have an increased probability of experiencing an uncorrectable (multi-bit) error. Likewise, the probability of a memory error condition that could result in system downtime also increases.

To help address this, new features have been implemented for UltraSPARC II-based, UltraSPARC III-based, and UltraSPARC IV-based systems. These features attempt to proactively predict which system memory components (DIMMs) have an increased probability of experiencing an uncorrectable error, and subsequently remove this memory from future use when it is no longer used by the kernel or any processes. Limits are placed on the number of memory pages that can be retired from use.

Note – Solaris 8 Kernel Update patches prior to 117000-03 and Solaris 9 Kernel Update patches 112233-12 and earlier disable page retirement for UltraSPARC IIIi-based systems by default. Track bug ID 4988523 for further details on implementation in Solaris 9.

A CPU receives notification that a correctable memory error has occurred using the trap mechanism. This initiates the following sequence of events:

1. The correctable error (CE) is scrubbed from the system using a sequence of address space identifier (ASI) memory accesses, cache line flushes, and so on.

Note – The operations to clear an error are CPU type specific.

   The CE can fall into one of three categories:

   - A CE is considered *intermittent* if the error is not detected upon a reread of the affected memory word. An intermittent CE is often referred to as a *transient soft error*.
   - A CE is considered *persistent* if the error is detected upon reread, but the scrubbing operation corrected it. A persistent CE is often referred to as a *temporary soft error*.
   - A CE is considered *sticky* if after scrubbing, the error is still present. A sticky CE is often referred to as a *stuck-at hard error*.

2. The physical address that caused the CE is read from the asynchronous fault address register (AFAR). From this, the DIMM that contains the affected memory cell is determined.

3. The "Leaky Bucket" SERD algorithm is invoked on this DIMM. This monitors the number and interval of CE occurrences on a system component. The algorithm ignores intermittent CE occurrences, and counts persistent CE occurrences. Sticky CE occurrences cause the page on which the CE occurs to be immediately marked as failing. This means a `PAGE_IS_FAILING` flag is set on the page. Until the limit of acceptable persistent CE occurrences on a DIMM has been exceeded, no pages on that DIMM are marked failing due to persistent CE occurrences. After the limit is exceeded, and for as long as it remains exceeded, every persistent CE on that

DIMM causes the containing page to be marked failing. Once the rate is again under the limit, an additional persistent CE no longer causes the containing page to be marked failing.

4. The physical page that has been marked failing is now retired from use if possible, and the system continues normal operations. Solaris 8 Kernel Update patch 117000-03 and Solaris 9 Kernel Update patch 112233-12 implement a more aggressive method of page retirement that is successful at retiring pages under a greater range of conditions. This is discussed in the section "Aggressive Page Retirement" on page 16.

Because of memory interleaving, every DIMM in the system has multiple pages associated with it. Conversely, a single physical page consists of portions of physical memory from multiple DIMMs.

Toxic vs. Failing Pages

The OS distinguishes between pages that have correctable errors and those that have uncorrectable errors. A page with an uncorrectable error that might be able to be cleared is marked as *toxic*. Pages mapped to a DIMM that has experienced multiple correctable errors are marked as *failing*.

If a page is marked toxic, the OS attempts to clean any errors from the page using a scrubbing algorithm when `page_free()` is invoked on that page. If it can verify that there are no errors on the page after it does its scrubbing, it allows that page to be returned to the freelist. This ensures that a single error does not cause a page to be removed from the system. If the scrubbing is unsuccessful, the page is marked as failing and is immediately retired.

If a page is marked failing, no attempt is made to clean the page by scrubbing. It is immediately retired if it is no longer in use by other threads.

The sequence of operations is as follows:

1. When the page is deemed to be failing, a flag is set on that page, `PAGE_IS_FAILING`.

2. When the page is no longer in use, `page_free()` is invoked on that page.

3. If the `PAGE_IS_FAILING` flag is set, `page_free()` moves the page to a special retired pages vnode, and the amount of available free memory in the system is decremented. The page is not returned to the freelist, and so will not be used again until reboot.

Aggressive Page Retirement

Solaris 8 Kernel Update patch 117000-03 and Solaris 9 Kernel Update patch 112233-12 implement a more aggressive method of page retirement, which is successful at retiring pages under a greater range of conditions. Prior to these kernel updates, if a page is found to be locked, dirty, or in a COPY_ON_WRITE status at the time the system attempts to retire it, the page could not have been retired until later during a `page_free()` as discussed in "Page Retirement" on page 13. Recall that some pages are allocated for a long lifetime and `page_free()` would not be called until the page was no longer in use. These patches implement a new algorithm to successfully retire pages under these conditions. This algorithm is referred to as Aggressive Page Retirement.

The algorithm first puts the page on a `bad_page_list` and signals a dispatch thread to let it know there are pages to process. The dispatch thread utilizes the standard Solaris `taskq` mechanism and puts the page in a `taskq` to be retired by a `page_retire_task`. This task acquires the necessary locks and relocates the contents of the original page to a new page. The new page is returned to its application, and the original, now unused, page is removed from use by `page_free()` as discussed in "Page Retirement" on page 13.

Page Retirement and DR

Page Retirement does not utilize DR. Automatic DR of an entire system board or its subcomponents is not attempted due to one or more pages being retired on a board. If a system administrator manually performs DR on a system board containing retired pages out of a domain and into the same or another domain, the pages will be active again if the `POST` process is successful.

There is currently no mechanism to inform `POST/OBP` of retired pages. For this reason, no method exists to remove a page from use permanently across reboots. The end result is that pages retired during one OS run become available again at the next boot.

Note – Solaris 8 Kernel Update patches prior to 108528-24 and Solaris 9 Kernel Update patches prior to 112233-11 provide a rudimentary implementation of page retirement, but it is disabled by default. You can enable the feature using the `/etc/system` file. However, due to bug IDs 4401262 and 4854496, which are present in those releases, it is not recommended to do so. The following two paragraphs describe these bug IDs in more detail.

Bug ID 4401262 describes a hang condition during dynamic reconfiguration. If a system board with DIMMs containing retired pages is subsequently dynamically reconfigured from the system, the DR operation hangs during the DR unconfigure

stage. Note that you might also encounter a memory error during DR that invokes the page retirement algorithms during the DR operation. This is expected to result in the same hang condition; however, test cases have not proven this. The message to understand regarding Solaris 8 Kernel Update patches prior to 108528-24 and Solaris 9 Kernel Update patches 112233-08 and earlier is that DR should not be used on a system board containing known retired pages. Service to the DIMMs should be postponed until such time as the OS can be taken down.

Bug ID 4854496 describes a panic condition that results from dereferencing a pointer that resides on a page already zeroed by page retirement. While no customers have experienced this panic condition outside of artificially created conditions, it is theoretically possible for it to occur.

Tunables

The following `/etc/system` variables and their possible values are listed here for reference only. Changing the variables to values other than their defaults should only be done under the guidance of an authorized Sun Microsystems service provider.

TABLE 2 Page Retirement Variables and Values

Variables	Values
`set ce_verbose_memory=[0/1/2]` `set ce_verbose_other=[0/1/2]`	A value of 0 indicates no logging. A value of 1 indicates that the messages are sent to the log file, but not the console. A value of 2 indicates that the messages are sent to the console and the log file. The default value is 1.

TABLE 2 Page Retirement Variables and Values *(Continued)*

Variables	Values
`set automatic_page_removal=[0/1]`	A value of 0 disables the page retirement feature. A value of 1 enables the page retirement feature. The default value depends upon the kernel patch release and system type. This is discussed in prior sections.
`set ecc_softerr_interval=1440` `set ecc_softerr_limit=2`	The interval measured in minutes and number of acceptable CEs within this interval. Used by the Leaky Bucket algorithm to determine when to begin page retirement. It is acceptable to have `ecc_softerr_limit` CEs within `ecc_softerr_interval` minutes. Beyond this limit, begin page retirement. These values are the defaults.
`set max_pages_retired_bps=10`	Limits the number of physical memory pages which can be retired. This number is a percentage of physical memory stored as basis points, where 100 basis points is 1%. The default is 10, or .1% of physical memory.

Example Messaging

The following is a sequential extract of messages from the system log of a system which experienced multiple errors on two DIMMs.

1. CE on Memory Module Board 4 J3401 (First DIMM, First Error)

```
Jan  7 04:13:29 pyre SUNW,UltraSPARC-II: [ID 194692 kern.notice]
[AFT0] Corrected Memory Error detected by CPU12, errID
0x0000003b.24cd6aea
Jan  7 04:13:29 pyre     AFSR 0x00000000.00100000<CE> AFAR
0x00000001.2db18000
Jan  7 04:13:29 pyre    AFSR.PSYND 0x0000(Score 05) AFSR.ETS 0x00
Fault_PC 0x78330400
Jan  7 04:13:29 pyre     UDBH Syndrome 0x64 Memory Module Board 4
J3401
Jan  7 04:13:29 pyre SUNW,UltraSPARC-II: [ID 898376 kern.notice]
[AFT0] errID 0x0000003b.24cd6aea Corrected Memory Error on Board
4 J3401 is Persistent
Jan  7 04:13:29 pyre SUNW,UltraSPARC-II: [ID 906141
kern.notice] [AFT0] errID 0x0000003b.24cd6aea ECC Data Bit  7 was
in error and corrected
```

2. CE on Memory Module Board 4 J3801 (Second DIMM, First Error)

```
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 338670 kern.notice]
[AFT0] Corrected Memory Error detected by CPU5, errID
0x00000043.f7fb26ef
Jan  7 04:14:07 pyre     AFSR 0x00000000.00100000<CE> AFAR
0x00000001.2bf6c000
Jan  7 04:14:07 pyre    AFSR.PSYND 0x0000(Score 05) AFSR.ETS 0x00
Fault_PC 0x78330be0
Jan  7 04:14:07 pyre     UDBH Syndrome 0xf2 Memory Module Board 4
J3801
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 282000 kern.notice]
[AFT0] errID 0x00000043.f7fb26ef Corrected Memory Error on Board
4 J3801 is Persistent
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 339990 kern.notice]
[AFT0] errID 0x00000043.f7fb26ef ECC Data Bit  9 was in error and
corrected
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x12bf6c000, Data 0x0eccf00d.ff250000, ECC 0xd5
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x12bf6c008, Data 0x0eccf00d.ff250000, ECC 0xd5
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x12bf6c010, Data 0x0eccf00d.ff250000, ECC 0xd5
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x12bf6c018, Data 0x0eccf00d.ff250000, ECC 0xd5
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x12bf6c020, Data 0x0eccf00d.ff250000, ECC 0xd5
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x12bf6c028, Data 0x0eccf00d.ff250000, ECC 0xd5
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x12bf6c030, Data 0x0eccf00d.ff250000, ECC 0xd5
```

3. CE on Memory Module Board 4 J3801 (Second DIMM, Second Error)

```
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 492729 kern.notice]
[AFT0] Corrected Memory Error detected by CPU5, errID
0x00000043.f9761088
Jan  7 04:14:07 pyre     AFSR 0x00000000.00100000<CE> AFAR
0x00000001.2bf6c040
Jan  7 04:14:07 pyre     AFSR.PSYND 0x0000(Score 05)
AFSR.ETS 0x00 Fault_PC 0x78330be0
Jan  7 04:14:07 pyre     UDBH Syndrome 0xf2 Memory Module Board 4
J3801
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 203173 kern.notice]
[AFT0] errID 0x00000043.f9761088 Corrected Memory Error on Board
4 J3801 is Persistent
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 429558 kern.notice]
[AFT0] errID 0x00000043.f9761088 ECC Data Bit  9 was in error and
corrected
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x12bf6c040, Data 0x0eccf00d.ff250040, ECC 0x5c
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x12bf6c048, Data 0x0eccf00d.ff250040, ECC 0x5c
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x12bf6c050, Data 0x0eccf00d.ff250040, ECC 0x5c
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x12bf6c058, Data 0x0eccf00d.ff250040, ECC 0x5c
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x12bf6c060, Data 0x0eccf00d.ff250040, ECC 0x5c
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x12bf6c068, Data 0x0eccf00d.ff250040, ECC 0x5c
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x12bf6c070, Data 0x0eccf00d.ff250040, ECC 0x5c
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x12bf6c078, Data 0x0eccf00d.ff250040, ECC 0x5c
```

4. CE on Memory Module Board 4 J3801 (Second DIMM, Third Error)

```
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 313390 kern.notice]
[AFT0] Corrected Memory Error detected by CPU5, errID
0x00000043.faed89ef
Jan  7 04:14:07 pyre     AFSR 0x00000000.00100000<CE> AFAR
0x00000001.2bf6c080
Jan  7 04:14:07 pyre    AFSR.PSYND 0x0000(Score 05) AFSR.ETS 0x00
Fault_PC 0x78330be0
Jan  7 04:14:07 pyre     UDBH Syndrome 0xf2 Memory Module Board 4
J3801
Jan  7 04:14:07 pyre unix: [ID 596940 kern.warning] WARNING: [AFT0]
3 soft errors in less than 24:00 (hh:mm) detected from Memory
Module Board 4 J3801
```

5. CE count on J3801 has exceeded maximum acceptable; page removal is attempted

```
Jan  7 04:14:07 pyre unix: [ID 618185 kern.notice] NOTICE:
Scheduling removal of page 0x00000001.2bf6c000
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 873457 kern.notice]
[AFT0] errID 0x00000043.faed89ef Corrected Memory Error on Board
4 J3801 is Persistent
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 185728 kern.notice]
[AFT0] errID 0x00000043.faed89ef ECC Data Bit  9 was in error and
corrected
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x12bf6c080, Data 0x0eccf00d.ff250080, ECC 0xb1
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x12bf6c088, Data 0x0eccf00d.ff250080, ECC 0xb1
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x12bf6c090, Data 0x0eccf00d.ff250080, ECC 0xb1
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x12bf6c098, Data 0x0eccf00d.ff250080, ECC 0xb1
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x12bf6c0a0, Data 0x0eccf00d.ff250080, ECC 0xb1
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x12bf6c0a8, Data 0x0eccf00d.ff250080, ECC 0xb1
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x12bf6c0b0, Data 0x0eccf00d.ff250080, ECC 0xb1
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x12bf6c0b8, Data 0x0eccf00d.ff250080, ECC 0xb1
```

6. CE on Memory Module Board 4 J3801 (Second DIMM, Fourth Error)

```
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 958426 kern.notice]
[AFT0] Corrected Memory Error detected by CPU5, errID
0x00000043.fc6c8d68
Jan  7 04:14:07 pyre     AFSR 0x00000000.00100000<CE> AFAR
0x00000001.2bf6c0c0
Jan  7 04:14:07 pyre    AFSR.PSYND 0x0000(Score 05) AFSR.ETS 0x00
Fault_PC 0x78330be0
Jan  7 04:14:07 pyre     UDBH Syndrome 0xf2 Memory Module Board 4
J3801
Jan  7 04:14:07 pyre unix: [ID 596940 kern.warning] WARNING: [AFT0]
4 soft errors in less than 24:00 (hh:mm) detected from Memory
Module Board 4 J3801
```

7. CE count on J3801 still in excess of maximum acceptable; page removal is attempted. Note: this is the same page as before, which could not be removed because it was still in use. This is the second attempt.

```
Jan  7 04:14:07 pyre unix: [ID 618185 kern.notice] NOTICE:
Scheduling removal of page 0x00000001.2bf6c000
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 716816 kern.notice]
[AFT0] errID 0x00000043.fc6c8d68 Corrected Memory Error on Board
4 J3801 is Persistent
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 236389 kern.notice]
[AFT0] errID 0x00000043.fc6c8d68 ECC Data Bit  9 was in error and
corrected
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x12bf6c0c0, Data 0x0eccf00d.ff2500c0, ECC 0x38
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x12bf6c0c8, Data 0x0eccf00d.ff2500c0, ECC 0x38
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x12bf6c0d0, Data 0x0eccf00d.ff2500c0, ECC 0x38
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x12bf6c0d8, Data 0x0eccf00d.ff2500c0, ECC 0x38
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x12bf6c0e0, Data 0x0eccf00d.ff2500c0, ECC 0x38
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x12bf6c0e8, Data 0x0eccf00d.ff2500c0, ECC 0x38
Jan  7 04:14:07 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x12bf6c0f0, Data 0x0eccf00d.ff2500c0, ECC 0x38
```

8. Page on Memory Module Board 4 J3801 is removed

```
Jan  7 04:14:12 pyre unix: [ID 693633 kern.notice] NOTICE: Page
0x00000001.2bf6c000 removed from service
```

9. CE on Memory Module Board 6 J3801 (Third DIMM, First Error)

```
Jan  7 04:14:20 pyre SUNW,UltraSPARC-II: [ID 298492 kern.notice]
[AFT0] Corrected Memory Error detected by CPU15, errID
0x00000046.fd74b4e8
Jan  7 04:14:20 pyre     AFSR 0x00000000.00100000<CE> AFAR
0x00000001.35bf4000
Jan  7 04:14:20 pyre    AFSR.PSYND 0x0000(Score 05) AFSR.ETS 0x00
Fault_PC 0x78330be0
Jan  7 04:14:20 pyre     UDBH Syndrome 0xf2 Memory Module Board 6
J3801
Jan  7 04:14:20 pyre SUNW,UltraSPARC-II: [ID 686754 kern.notice]
[AFT0] errID 0x00000046.fd74b4e8 Corrected Memory Error on Board
6 J3801 is Persistent
Jan  7 04:14:20 pyre SUNW,UltraSPARC-II: [ID 542367 kern.notice]
[AFT0] errID 0x00000046.fd74b4e8 ECC Data Bit  9 was in error and
corrected
Jan  7 04:14:20 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x135bf4000, Data 0x0eccf00d.ff250000, ECC 0xd5
Jan  7 04:14:20 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x135bf4008, Data 0x0eccf00d.ff250000, ECC 0xd5
Jan  7 04:14:20 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x135bf4010, Data 0x0eccf00d.ff250000, ECC 0xd5
Jan  7 04:14:20 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x135bf4018, Data 0x0eccf00d.ff250000, ECC 0xd5
Jan  7 04:14:20 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x135bf4020, Data 0x0eccf00d.ff250000, ECC 0xd5
Jan  7 04:14:20 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x135bf4028, Data 0x0eccf00d.ff250000, ECC 0xd5
```

10. CE on Memory Module Board 6 J3801 (Third DIMM, Second Error)

```
Jan  7 04:14:20 pyre SUNW,UltraSPARC-II: [ID 565072 kern.notice]
[AFT0] Corrected Memory Error detected by CPU15, errID
0x00000046.fef0581a
Jan  7 04:14:20 pyre     AFSR 0x00000000.00100000<CE> AFAR
0x00000001.35bf4040
Jan  7 04:14:20 pyre    AFSR.PSYND 0x0000(Score 05) AFSR.ETS 0x00
Fault_PC 0x78330be0
Jan  7 04:14:20 pyre     UDBH Syndrome 0xf2 Memory Module Board 6
J3801
Jan  7 04:14:20 pyre SUNW,UltraSPARC-II: [ID 190951 kern.notice]
[AFT0] errID 0x00000046.fef0581a Corrected Memory Error on Board
6 J3801 is Persistent
Jan  7 04:14:20 pyre SUNW,UltraSPARC-II: [ID 744456 kern.notice]
[AFT0] errID 0x00000046.fef0581a ECC Data Bit  9 was in error and
corrected
Jan  7 04:14:20 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x135bf4040, Data 0x0eccf00d.ff250040, ECC 0x5c
Jan  7 04:14:20 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x135bf4048, Data 0x0eccf00d.ff250040, ECC 0x5c
Jan  7 04:14:20 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x135bf4050, Data 0x0eccf00d.ff250040, ECC 0x5c
Jan  7 04:14:20 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x135bf4058, Data 0x0eccf00d.ff250040, ECC 0x5c
Jan  7 04:14:20 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x135bf4060, Data 0x0eccf00d.ff250040, ECC 0x5c
Jan  7 04:14:20 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x135bf4068, Data 0x0eccf00d.ff250040, ECC 0x5c
Jan  7 04:14:20 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x135bf4070, Data 0x0eccf00d.ff250040, ECC 0x5c
Jan  7 04:14:20 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x135bf4078, Data 0x0eccf00d.ff250040, ECC 0x5c
```

11. CE on Memory Module Board 6 J3801 (Third DIMM, Third Error)

```
Jan  7 04:14:20 pyre SUNW,UltraSPARC-II: [ID 372229 kern.notice]
[AFT0] Corrected Memory Error detected by CPU15, errID
0x00000047.005d5687
Jan  7 04:14:20 pyre     AFSR 0x00000000.00100000<CE> AFAR
0x00000001.35bf4080
Jan  7 04:14:20 pyre    AFSR.PSYND 0x0000(Score 05) AFSR.ETS 0x00
Fault_PC 0x78330be0
Jan  7 04:14:20 pyre     UDBH Syndrome 0xf2 Memory Module Board 6
J3801
Jan  7 04:14:20 pyre unix: [ID 596940 kern.warning] WARNING: [AFT0]
3 soft errors in less than 24:00 (hh:mm) detected from Memory
Module Board 6 J3801
```

12. CE count on J3801 has exceeded maximum acceptable; page removal is attempted

```
Jan  7 04:14:20 pyre unix: [ID 618185 kern.notice] NOTICE:
Scheduling removal of page 0x00000001.35bf4000
Jan  7 04:14:20 pyre SUNW,UltraSPARC-II: [ID 834227 kern.notice]
[AFT0] errID 0x00000047.005d5687 Corrected
Memory Error on Board 6 J3801 is Persistent
Jan  7 04:14:20 pyre SUNW,UltraSPARC-II: [ID 745085 kern.notice]
[AFT0] errID 0x00000047.005d5687 ECC Data Bit  9 was in error and
corrected
Jan  7 04:14:20 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x135bf4080, Data 0x0eccf00d.ff250080, ECC 0xb1
Jan  7 04:14:20 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x135bf4088, Data 0x0eccf00d.ff250080, ECC 0xb1
Jan  7 04:14:20 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x135bf4090, Data 0x0eccf00d.ff250080, ECC 0xb1
Jan  7 04:14:20 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x135bf4098, Data 0x0eccf00d.ff250080, ECC 0xb1
Jan  7 04:14:20 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x135bf40a0, Data 0x0eccf00d.ff250080, ECC 0xb1
Jan  7 04:14:20 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x135bf40a8, Data 0x0eccf00d.ff250080, ECC 0xb1
Jan  7 04:14:20 pyre SUNW,UltraSPARC-II: [ID 832828 kern.notice]
[AFT0]    Paddr 0x135bf40b0, Data 0x0eccf00d.ff250080, ECC 0xb1
```

13. Page for Memory Module Board 6 J3801 is removed

```
Jan  7 04:14:25 pyre unix: [ID 693633 kern.notice] NOTICE: Page
0x00000001.35bf4000 removed from service
```

About the Author

Tom Chalfant received his bachelor's degree in Computer Science from the University of Kentucky and a master's degree in Computer Science from the University of Dayton. Prior to joining Sun in 1996, Tom served in the U.S. Air Force, first as a Communications, Computer Systems, Programming, and Analysis Officer working on multi-platform software development, and later as a member of a civilian group which managed over 1800 Sun servers and desktops. Here, he had the pleasure of working on various projects including a favorite at the time, Firewalls and Network Security Design.

During his 7.5 years at Sun, Tom has worked as a System Support Engineer and Regional System Support Engineer for Enterprise Services, a Corporate Technical Escalation Engineer in the High End Server Group (HSG) within ESG, and is currently a staff engineer with the Strategic Solutions Group.

Acknowledgements

I would like to thank: Jim Quigley for content contributions; Steve Chessin and Kumar Loganathan for many detailed reviews; and finally Stephane Dutilleul and Gavin Maltby for help in generating example messages.

References

1. Wittsack, Tricia and Gonscherowski, Peter. "Sun Fire Hardware Midrange Servers Auto Diagnosis and Recovery Enhancements," *Sun BluePrints OnLine*, April 2004.

To access this article online, go to
`http://www.sun.com/blueprints/0404/817-6212.pdf`

2. Butala, Vasant. "SunFire 15K/12K Auto Diagnosis and Recovery", Sun BluePrints OnLine, February 2004.

To access this article online, go to
`http://www.sun.com/blueprints/0204/817-5206.pdf`

3. "Patch Rejuvenation Due to Size of Solaris 8 Kernel Update (KU) Being Very Large (37MB)", Free Sun Alert Notification 57489, February 2004.

To access this article online, go to
`http://sunsolve.sun.com`

Ordering Sun Documents

The SunDocs[SM] program provides more than 250 manuals from Sun Microsystems, Inc. If you live in the United States, Canada, Europe, or Japan, you can purchase documentation sets or individual manuals through this program.

Accessing Sun Documentation Online

The `docs.sun.com` web site enables you to access Sun technical documentation online. You can browse the `docs.sun.com` archive or search for a specific book title or subject. The URL is `http://docs.sun.com/`

To reference Sun BluePrints OnLine articles, visit the Sun BluePrints OnLine web site at: `http://www.sun.com/blueprints/online.html`