Predictive Fault Monitoring in Sun Fire™ Servers

Dave Re, Product Technical Support

Kumar Loganathan, Product Technical Support

Sun BluePrints™ OnLine—April 2005

http://www.sun.com/blueprints

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

Part No. 819-2261-10
Revision 1.0, 3/21/05
Edition: April 2005

Copyright 2005 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, California 95045 U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at `http://www.sun.com/patents` and one or more additional patents or pending patent applications in the U.S. and in other countries.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, Sun BluePrints, SunSolve, SunSolve Online, docs.sun.com, JumpStart, N1, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

U.S. Government Rights—Commercial use. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2005 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, Californie 95045 Etats-Unis. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuels relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plus des brevets américains listés à l'adresse `http://www.sun.com/patents` et un ou les brevets supplémentaires ou les applications de brevet en attente aux Etats - Unis et dans les autres pays.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Certaines parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque enregistree aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, Sun BluePrints, SunSolve, SunSolve Online, docs.sun.com, JumpStart, N1, et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REPONDRE A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.

Please
Recycle

Predictive Fault Monitoring in Sun Fire™ Servers

This Sun BluePrint™ article describes various Predictive Fault Monitoring tools that Sun Fire™ Servers implement to help increase the overall system availability and reliability. This document contains the following sections:

Introduction

Modern enterprise class computer systems represent increasingly complex application delivery vehicles. Over time, system downtime has become increasingly expensive, driving a demand for highly reliable hardware platforms and more robust operating systems and application suites. As the cost of downtime continues to rise, it becomes increasingly desirable to be able to predict incipient failures in these complex systems (with the associated benefit of scheduling convenient downtime to effect repairs), as well as to be able to gracefully and reliably handle unpredictable failures.

About Predictive Fault Monitoring

Predictive Fault Monitoring features allow a system to track and monitor variables—such as correctable or uncorrectable error rates and locations, or system environmentals (temperature, voltage, and so on)—and then compare that data to set thresholds, or analyze that data with complex decision making algorithms. The results of those analyses can then be used to warn system operators of an incipient failure or, in some cases, remove the offending *Field Replaceable Unit* (*FRU*) from the system configuration before it can cause a system failure.

About This Document

This document describes several new Predictive Fault Monitoring features in Sun's enterprise class Sun Fire server platforms (V1280-E25K) and in Sun's Solaris™ operating system (Solaris OS), including discussion about how these features operate and what action should be taken based on their output. The intention of this BluePrint document is to educate the reader on the functionality of these features so that the reader can use these new features to increase overall uptime in Sun's enterprise class systems.

The document assumes a basic understanding of Sun Fire server platforms and the terminology associated with them. It also assumes a basic understanding of Solaris OS administration and maintenance issues.

Processor Core Voltage Telemetry Monitor

In order to facilitate ease of repair, processors are mated to the system board (Uniboard) using a fixture known as a *socket*. The socket's job is to act as an electrical conductor between the processor and Uniboard, and yet allow the processor to be removed from the Uniboard without requiring a potentially damaging desoldering operation. Failures within the socket appear as a connectivity problem between the processor and Uniboard. This type of failure has been determined to exhibit a very gradually increasing electrical resistance within the socket's conductor columns, between the processor package itself and the printed circuit board (PCB) of the Uniboard.

Measuring this increasing resistance involves monitoring each individual connection between the processor and the PCB which, practically speaking, is an impossible task. However, physics comes to the rescue. Electrical current consumed by the processor remains relatively constant. Ohm's Law declares that, if current remains constant and resistance increases, voltage must also increase:

Voltage equals Current times Resistance (`V = IR`).

Voltage on the supply lines to each processor, known as the *core voltage* (*Vcore*), is monitored by simple circuits on the Uniboard.

The *Core Voltage Telemetry Monitor* (referred to as *CVTM* in this document) is functionality that is integrated into the system controller software to track the Vcore signal of a processor. CVTM provides a warning mechanism for processors that show a gradual increase in Vcore voltage—a phenomenon caused by the increasing resistance in the connection between the processor and the PCB. At the time that this document was written, the CVTM feature was available only on the V1280-E6900 systems.

CVTM Availability and Compatibility

CVTM first appeared in firmware version 5.18.0 for Sun Fire V1280-E6900 servers. CVTM operates on all Sun Fire servers in these classes. Only UltraSPARC-III class processors are monitored. Due to a physical design change, the UltraSPARC-IV processors are not vulnerable to the issue monitored by CVTM. CVTM is enabled by default upon installation of 5.18.0 or higher firmware. It is possible—though not recommended—to disable CVTM (under direct supervision of Sun personnel) should a problem arise in its operation.

CVTM Functional Algorithms

Accurately monitoring trends in processor Vcore presents a unique problem. The 8-bit analog to digital (A/D) converters used to sample the analog electrical signal cannot resolve voltage changes smaller than one hundredth (.01) of a volt. The A/D converter must quantize the analog signal into an 8-bit value, which is then interpreted by the *System Controller Application* (*ScApp*) to a valid voltage. The process inherently results in a loss of resolution, as voltage fluctuations typically occur in the thousandth (.001) of a volt range, and a given value sampled by the A/D and reported by ScApp can actually represent a relatively wide range of analog voltage values. To add to the challenge, voltages (and other environmental signals) are expected to fluctuate slightly over time as load on the processor changes, which makes tracking a trend all the more difficult. The ScApp command `showenvironment` displays the values that have been most recently sampled.

In order to overcome this limitation, CVTM makes use of the fact that a properly functioning processor (one that does not exhibit a voltage ramp and is not suffering from the problem that CVTM monitors), has a nominally flat Vcore over time, and only a small number of quantized values are expected to be sampled from the A/D converter. For example, if one were to run the `showenvironment` command once a second and write down the unique values seen for a particular processor, a typical result might be "1.63V, 1.64V, 1.65V, 1.66V". Further, if the monitored processor is functioning properly, then the distribution of those values should remain constant over a large number of samples—the ratio of each discrete value to the total number of samples should stay constant over time. In a processor affected by increasing connection resistance, the higher quantized values will occur more frequently over time, causing a change in the observed distribution of quantized values.

Examples of VCore Samplings

The following figures show how this might appear visually. The quantized values sampled by ScApp (the same values that might be monitored with `showenvironment`) have been smoothed (made less jagged) with a moving average to aid in the visualization in both examples.

Example of Vcore Values From a Properly Functioning Processor

Figure 1 shows the plot of Vcore values from a properly functioning processor. The same signal can be viewed as a sequence of discrete random values. At any point in time, one can construct the distribution of discrete values obtained since the beginning of sampling.

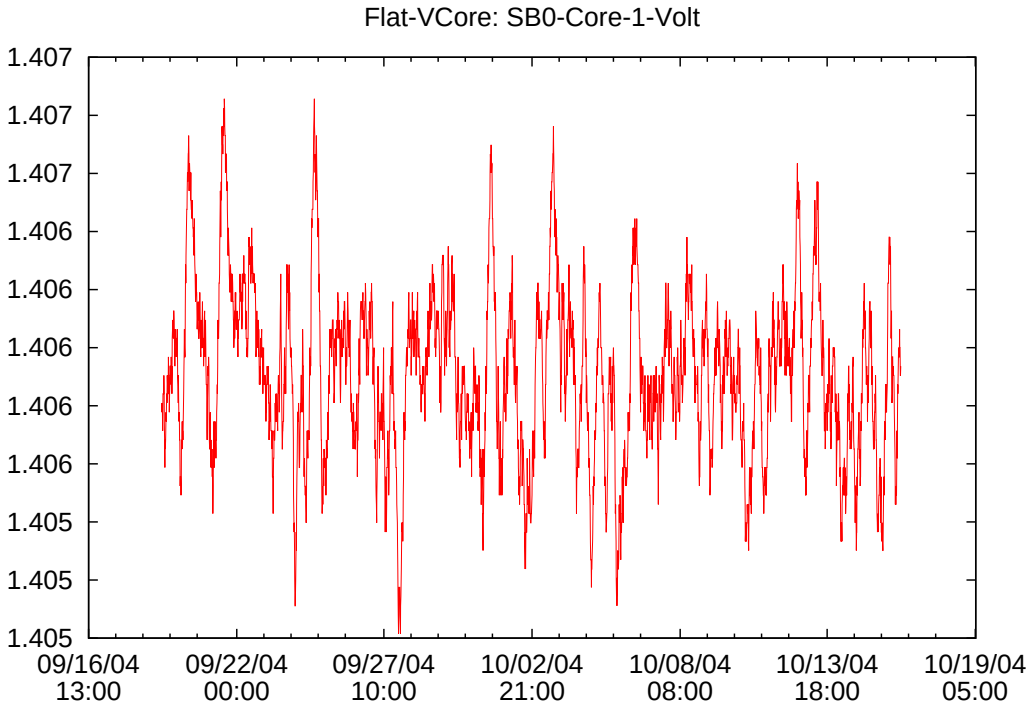

FIGURE 1 Plot of the Smoothed Core Voltage Signal for a Properly Functioning Processor Over Time

Figure 2 shows how this time-dependent distribution (also called a *moving histogram*) evolves as samples are taken.

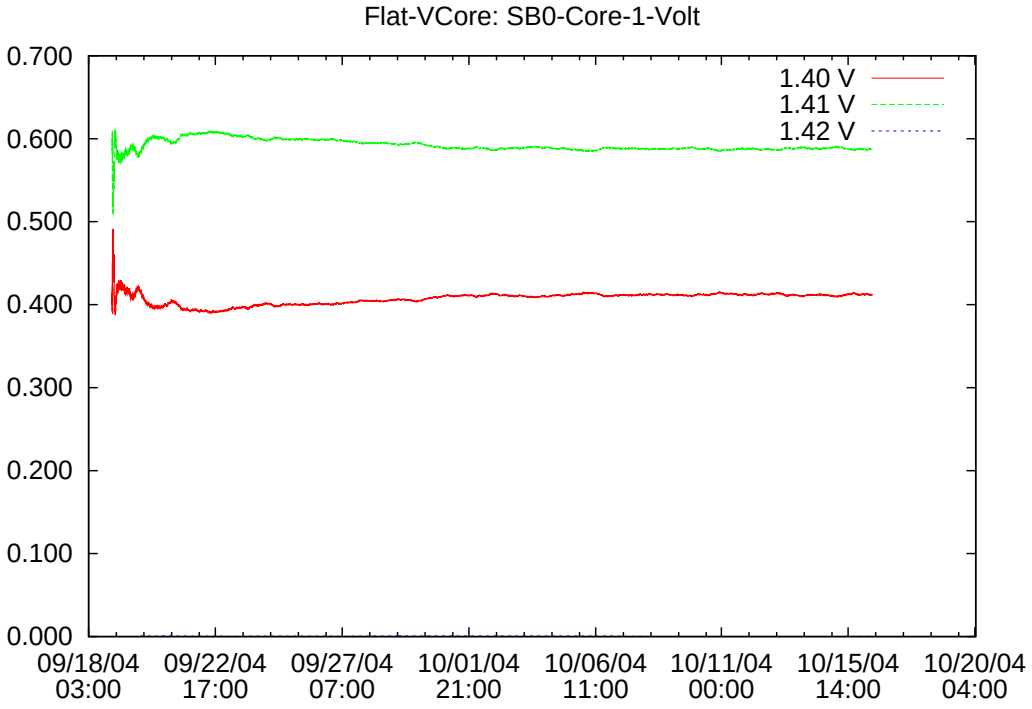

FIGURE 2 Moving Histogram Plot of Quantized Values Seen Over Time

Figure 2 has three lines—one for each of the discrete values sampled (1.40 V, 1.41 V, and 1.42 V).

- The top line for the 1.41 V value shows that, for this signal, approximately 60% of the samples taken are 1.41 V.
- The second line shows that the frequency of 1.40 V values is 40%.
- The third line for the 1.42V samples is actually flush with the X-axis.

Note that, once a sufficient number of samples have been taken, these three lines are relatively flat. This is what one would expect from a core voltage signal that is stationary—one whose distribution does not change over time.

Example of Ramping Vcore Values

Contrast Figures 1 and 2 with the plots in Figures 3 and 4 for a ramping core voltage signal.

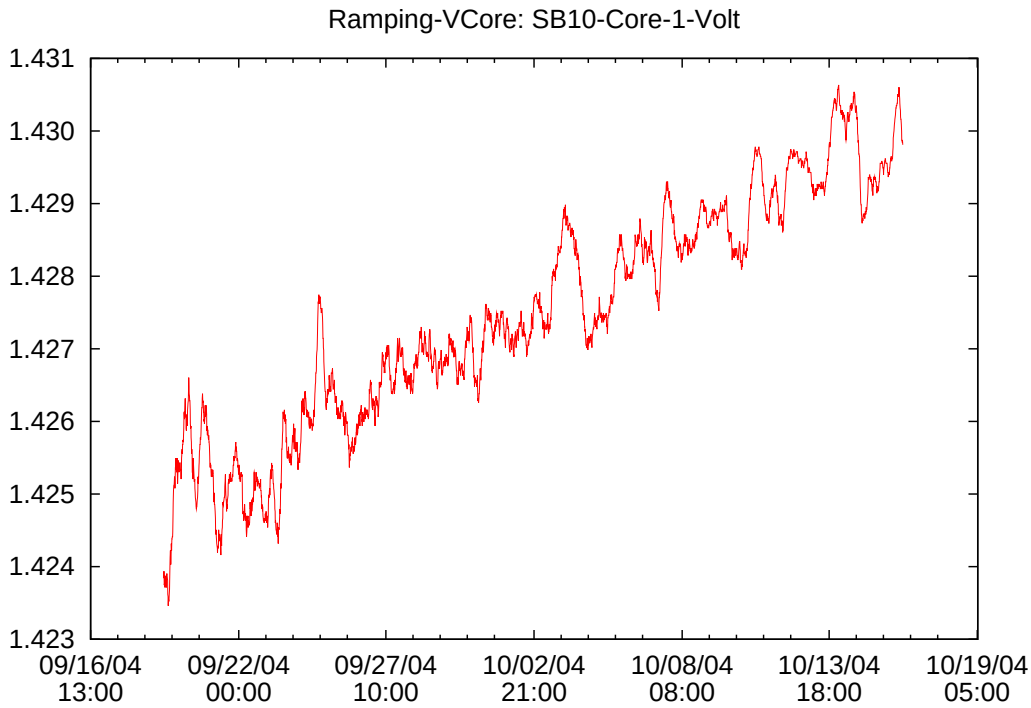

FIGURE 3 Plot of a Ramping Core Voltage Signal

The moving histogram plot in Figure 4 clearly shows that the 1.43 V values are occurring at an increasing rate over time, at the expense of the lower-valued 1.42 V and 1.41 V samples. The fact that the distribution lines are not flat indicates that the distribution is changing, which means that the underlying core voltage signal is changing as well.

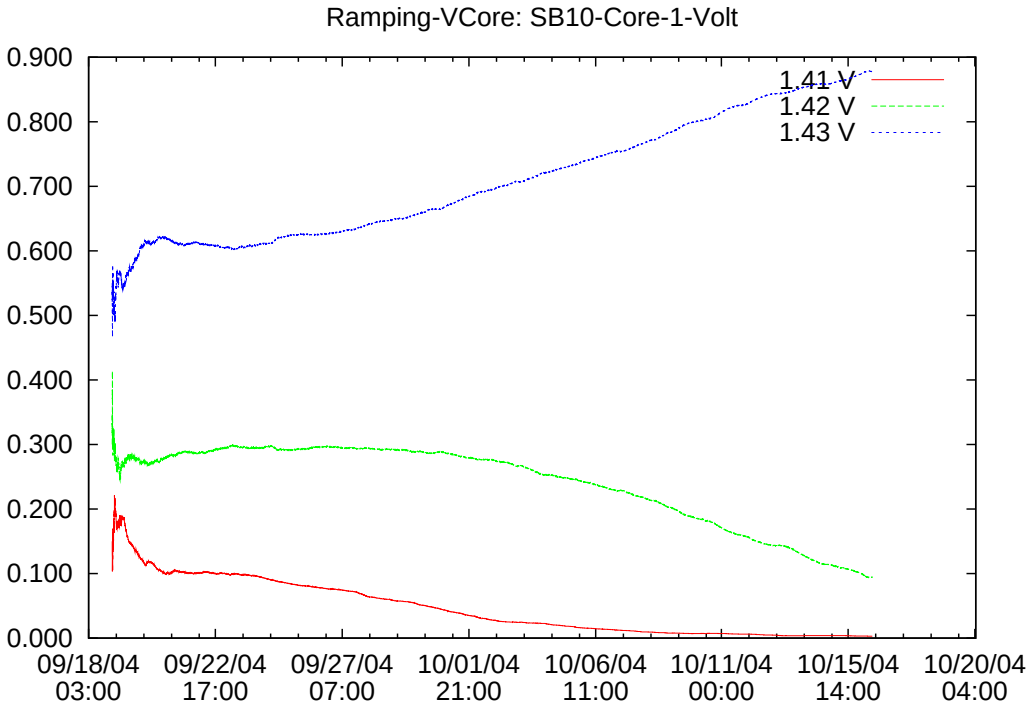

FIGURE 4 Moving Histogram Showing a Changing Quantized Value Distribution Over Time

Sequential Probability Ratio Test

Statistical techniques can be used to detect when the change in distribution is significant. This changepoint can be used to trigger the calculation of an index to assess the *severity* (*steepness*) of the core voltage ramp. A specific statistical analysis, called a *Sequential Probability Ratio Test* (*SPRT*), can be used for this purpose.

In statistical terms, the SPRT can be used to detect when a Gaussian random process starts to *drift away* from *stationarity*. Stationarity is a statistical term that describes a condition in which the properties of a given system do not change over time—a system that moves away from stationarity is one that is changing after having been stable. In common terms, this means that SPRT can be used to very accurately determine when a processor is experiencing increased connectivity resistance, once the data is prepared in a fashion that can be utilized by the SPRT algorithm.

SPRT must operate on a Gaussian distribution of data, so a direct feed of quantized values from the A/D converter will not suffice. Instead, the quantized data are transformed through a moving histogram on which the SPRT algorithm is applied. SPRT has several data requirements that CVTM must accommodate. CVTM must gather data on each targeted processor for a period of time, establishing an expected distribution of signals for each processor. Once a day, CVTM applies the SPRT algorithm to the collected data. If the SPRT algorithm detects a statistically significant change in the distribution of quantized values sampled for a given processor, a severity index is determined based on the slope of the curve in the Vcore voltage signal for that processor.

Note – *Severity* simply represents the slope of the Vcore ramp and does not imply a probability of failure on that particular CPU.

Phases of Data Collection and Analysis

The process of collecting and evaluating data is broken down into three phases—Blackout, Training, and Monitoring. During the Blackout and Training phases, CVTM learns the expected distribution of quantized voltage values for the processor. In the Monitoring phase, CVTM applies a statistical hypothesis test (the SPRT) to determine whether the observed distribution has deviated significantly from the distribution learned in the Blackout and Training phases.

Phase 1: Blackout

When a processor is first targeted for monitoring, core voltage data must be collected from the processor to establish initial conditions for the SPRT algorithm. Specifically, sufficient samples must be taken to give the distribution enough time to reach a steady state. As can be seen in Figure 2, the individual frequency distribution lines

for a Vcore voltage are not flat initially—they take time to stabilize. Once the moving histogram stabilizes, however, it is expected to remain constant for a properly functioning processor (it should remain constant throughout the Monitoring phase of the algorithm). The period of time in which the distribution is allowed to stabilize is called the *Blackout phase*. After extensively monitoring and testing many different systems, Sun's Reliability and Serviceability Computer Analysis Lab (RASCAL) has established that 24 hours of data (1440 samples, 1 per minute) are needed for the distribution to sufficiently stabilize. No determination is made during the Blackout phase as to whether the core voltage is ramping or not.

Each core voltage signal has an associated histogram data structure to keep track of the current distribution of values. This histogram data structure contains a set of bins, one for each observed quantized value (the lines in Figures 2 and 4). When a new value is sampled for a core voltage signal, the count of the corresponding bin is incremented. The frequency of the bin is calculated as the bin count divided by the number of samples processed. As new quantized values are seen for the first time in the Blackout phase, the set of bins is expanded to include the new quantized values. The sum of all the bin counts equals the number of samples processed, and the sum of the bin frequencies equals 100%. The bins reflect all samples seen throughout the Blackout phase.

Phase 2: Training

In order to make use of SPRT, two summary statistics (*mean* and *variance*) must be calculated for each bin frequency time series. The mean and variance characterize the baseline moving histogram for each processor. This baseline will then be compared with the actual moving histogram observed during CVTM's Monitoring phase. During the 30 hour Training phase, CVTM gathers 1800 signal samples per monitored processor to calculate the required summary statistics.

When a new sample is taken for a core voltage signal during the Training phase, the signal's histogram is updated in the same fashion as in the Blackout phase. A time series results when examining each bin's frequency after each sample (for example, the 1.41 V bin in Figure 2 has a frequency time series that looks something like 0.5995, 0.5992, 0.5990, 0.6012, 0.6011, 0.6134, ...).

At the end of the Training phase, CVTM calculates the mean and variance for the bin frequency time series. Once the Monitoring phase begins, it is expected that the bin frequency time series will adhere to the same distribution (characterized by the mean and variance) learned during the Training phase. If a deviation occurs, then the core voltage signal is known to be changing. Note that no determination is made during the Training phase as to whether a ramp exists in a particular processor's Vcore signal data.

In addition to updating the histograms and calculating the mean and variance of the bin frequencies, the mean value of the core voltage signal itself is calculated over the Blackout and Training phases. This value serves as a reference point when calculating the slope of a drifting core voltage signal in the Monitoring phase.

Phase 3: Monitoring

Upon exiting the Training Phase, CVTM possesses enough data about each Vcore signal to begin Monitoring for ramps. During the Monitoring phase, each signal is sampled once per minute, and each signal's moving histogram is updated following each sample, just as it was in the Blackout and Training phases. Following each new sample, a new frequency is calculated for each bin, which is then compared with the reference distribution that was established in the Training phase using the SPRT algorithm.

If the SPRT indicates a significant change in the signal's distribution, then the algorithm raises an SPRT alarm. If seven of the last twelve samples have had SPRT alarms raised against them, CVTM calculates a severity index for the ramp (essentially, the slope of the upward curve in the signal) and issues a CVTM warning message to the platform and domain consoles, as well as associated loghosts (if configured).

Data Reset Events

Certain events affecting the processor have the tendency to induce an upward step change in the Vcore signal. If CVTM were allowed to use data gathered from the step change, then a false alarm could result. Therefore, several events will cause the data for a processor to be reset back to the Blackout phase.

All data used by CVTM resides in memory on the main system controller (SC). Due to the size of the data, and the frequency of change it experiences, a mechanism is not available to synchronize CVTM's data to the spare SC. Therefore, a reboot of the main SC, or a failover to the spare SC, will cause data for all processors within the system to be reset back to the Blackout Phase.

The following table describes the various data reset events.

TABLE 1 CVTM Data Reset Events

Event	Result
Board POST or DR	For all of the processors on the board, all data is reset and the processors revert back to the Blackout Phase.
Processor DR	No reset occurs. The processor continues in the current phase, its data is not changed, and CVTM continues to monitor the processor normally.
Board power off / power on	For all of the processors on the board, all data is reset and the processors revert back to the Blackout Phase.
SC Reset / Reboot / Failover	All data is cleared during one of these events. All processors in the system are reset to the Blackout Phase.

CVTM Messaging

Following identification of a ramping Vcore voltage, CVTM will issue an error message to the platform console and appropriate domain console once every 24 hours. The error messages take the following form:

```
[VCM] Event: <platform>.<initiator>.<cpu>.<sev_index>.<voltage>
                CSN: <csn> DomainID: <domain> ADInfo: <version>
                Time: <time>
                FRU-List-Count: 1; FRU-PN: <Part>; FRU-SN: <Ser>;
FRU-LOC:SBx/P<#>
                Recommended-Action: Service action required


     <platform> = SFxxxx where xxxx = 4800, 6900, 6800, etc.
     <initiator> = VCMON
     <cpu> = CPU having the issue (0..3)
     <sev_index> = severity index (2...10)
     <voltage> = last 24 hour average voltage without the decimal
(CCCC)
     <csn> = Chassis Serial Number,  <domain> = domain effected
    <version> = vcmon message version (MsgVer.Initiator.Major.Minor)
     <time> = time and date event was diagnosed
```

Use this form to decode any error messages. The following example shows an actual error message:

```
sc0:SC>
Sep 16 12:38:59 ds4-sc1 Platform.SC: [VCM] Event:
SF6800.VCMON.0.02.1663
         CSN: 036H3005 DomainID: B ADInfo: 1.VCMON.18.0
         Time: Thu Sep 16 12:38:59 PDT 2004
        FRU-List-Count: 1; FRU-PN: 5014362; FRU-SN: 004413; FRU-LOC:
/N0/SB3/P0
         Recommended-Action: Service action required
```

Here is an interpretation of what the error message means. This 6800 system has issued a CVTM error against SB3 processor 0 (or, CPU 12 from within the Solaris OS). The system's serial number is 036H3005. SB3 is part of domain B. The ADInfo field identifies that the system is running 5.18.0 firmware. The error message was issued by ScApp at the timestamp recorded in the Time field.

Recommended Action

A board flagged by CVTM should be targeted for replacement at the earliest convenient time. Contact your authorized Sun service provider for information on how to proceed with board replacement.

CPU Diagnostic Monitor

The *CPU Diagnostic Monitor* (*CDM*) is an online processor diagnostic program for platforms based on the UltraSPARC III family of processors. CDM monitors the *floating point unit* (*FPU*) of a processor by periodically running floating point math and FPU utilization tests as a background process in the Solaris OS. If any FPU failures are detected, appropriate actions are taken to prevent data corruption. This software continually tests the processor for FPU integrity issues and improves the overall reliability of the system. It is highly recommended that customers install and monitor CDM on production systems with UltraSPARC III and IV processors.

CDM contains a daemon (`cpudiagd`) and a test program (`cputst`). The daemon invokes the test program periodically based on a user configurable schedule, and then handles the resulting errors if a processor is found to be faulty.

The error detection and actions taken are logged using the syslog mechanism (which, by default, is logged in `/var/adm/messages`). In addition, errors are logged in a CDM-specific error log file (`/var/cpudiag/log/error.log`). Informational messages, such as test execution start, end, and elapsed time statistics, are logged in a CDM-specific information log file (`/var/cpudiag/log/info.log`).

For more information on configuring and using CDM, refer to the *Online CPU Diagnostics Monitor User Guide*, which is available from the Sun Download Center (after logging in) at:

http://javashoplm.sun.com/ECom/docs/Welcome.jsp?StoreId=8&PartDetailId=CPUDM-1.0-SSP-G-F&TransactionId=try.

What Happens When a Faulty Processor is Detected

When a faulty processor is detected by `cputst`, it communicates the information about the faulty processor to `cpudiagd`. The following sequence of operations then occurs:

1. The daemon logs an error message about the fault using the `syslog(3C)` mechanism, which logs the error message into the `/var/adm/messages` file by default. In addition, the CDM specific error log is updated.
2. The daemon creates a bad processor history file (`/var/cpudiag/data/bad_cpu_id.`*X*, where *X* is the decimal processor ID of the faulty processor). This file is used by the daemon to recognize the faulty processor as a suspected bad processor across reboots until this file is manually deleted by the system administrator after the faulty processor is replaced.
3. The CDM software performs an initial attempt to offline the detected faulty processor. The offline attempt fails if any process is bound to the faulty processor.
4. If the user has provided a binary script to be run when a fault is detected, it is invoked. This script would, for example, notify the system administrator about the fault and shut down any user applications that might be explicitly bound to the faulty processor.
5. The software then attempts to offline the faulty processor again. This attempt to offline is likely to succeed if the binary/script has shutdown the user applications, and all processes bound to the faulty processor were terminated.
6. If the offline attempt fails and if the bad processor is still offline, CDM will reboot the system. Emergency category syslog messages are logged before the system is rebooted or halted. Messages include the specific cause of the problem with the processor ID that failed when the system was halted or rebooted.
7. On reboot, the `cpudiagd` daemon is run with the `-i` option from the startup script, which causes `cpudiagd` to run a special startup regimen. If there is a suspected faulty processor indicated by a bad processor history file (`/var/cpudiag/data/bad_cpu_id.`*X*) as created from Step 2, then the software runs `cputst` in high stress mode to test the processor. If the processor found is faulty, then the software performs Steps 2 through 5. If the faulty processor still cannot be taken offline, then the system will be halted to prevent indefinite looping on reboot.

Invoking cputst At Various Stress Levels

The `cputst` test program can be invoked with three levels of stress —low, medium, or high. The operations performed in all three modes are the same. The difference between low, medium, and high stress levels is in how many times an operation or set of operations is performed.

The `cputst` test program performs complex mathematical operations that involve complex floating point operations—such as operations with matrices—and compares the resulting answers with the known correct solutions to the tests. The stress level is also reflected in the size of the matrices with which `cputst` operates.

Low stress tests provide functional coverage of the floating point and caches in the processor. High stress tests simulate a floating point intensive application. Approximate memory and processor resources consumed by `cputst` for different stress levels are documented in the *Online CPU Diagnostics Monitor User Guide*.

The likelihood of catching problems is better in high stress mode than in low stress mode. However, processor run time increases with the stress level.

Fault Management Architecture in Solaris 10 OS

In addition to CDM, the Solaris 10 OS further enhances FPU error detection capabilities that are integrated into the *Fault Management Architecture* (*FMA*). When an FPU issue with a processor is detected by the Solaris 10 OS, the system will produce a set of FMA error events that will allow the Solaris Fault Manager to produce a fault event and offline the processor experiencing the FPU issue automatically. With the Service Management Facility in the Solaris 10 OS, any affected user process will be killed and restarted automatically if a copy to or from its address space was likely affected by FPU corruption.

Parity Error Handling

The Sun Fire architecture has *Error Checking and Correcting* (*ECC*) logic that allows for both the detection and correction of single bit data error disturbances. While ECC provides end-to-end data protection, parity checking is used throughout the system to allow specific detection of errors between critical logic points. For more information, refer to *The Sun Fireplane System Interconnect* white paper by Alan Charlesworth, which is available at:

http://www.sun.com/servers/white-papers/SunFireplane_Interconnect.pdf

ECC errors are detected by the Solaris OS and the SC independent of each other. The diagnostic engine in ScApp correlates ECC and parity information from the hardware to provide accurate diagnosis. When diagnosing memory and datapath errors, the engine examines ECC information from the hardware and information from the Solaris OS.

Data path faults in the system that report a parity error will also generate an ECC error message. These messages can be mistakenly diagnosed as memory faults, causing unnecessary memory DIMM replacements. While diagnosing data path errors, examine showerrorbuffer, showlogs, and loghost data. If further assistance is required, contact a Sun Service representative.

It is recommended that the `showchs` command also be used to determine whether the Component Health Status (CHS) of a component is marked SUSPECT to be sure that the correct component (memory versus system board) is considered for replacement. If a component is found marked SUSPECT, further analysis should be conducted on the failure that resulted in the SUSPECT status to determine whether replacement is required.

Persistent Indictment of Non-Fatal L2SRAM Errors

A *Soft Error Rate Discrimination* (*SERD*) algorithm is also used to detect when a specified number of distinct ECC events have occurred on the same processor in a 24-hour period. After the specified processor SERD events, the processor and its associated Level 2 SRAM cache (L2SRAM) become candidates for Solaris OE off-lining. The Solaris OS communicates these indictments of processor and L2SRAM components to the SC, which then updates the CHS of the affected L2SRAM/processor, preventing it from being used upon the next system reboot.

The Solaris OS offlines the processor when SERD indicates an L2SRAM correctable error threshold has been exceeded (for UCC, EDC, WDC, or CPC events) or immediately when an uncorrectable (XXU) event is encountered.

For more information, see the following Sun BluePrint: *Solaris™ Operating System Availability Features* by Thomas M. Chalfant (May, 2004), which is available at:

http://www.sun.com/blueprints/0504/817-7039.pdf

Examples of Processor Offlining for L2SRAM Events

A processor is offlined immediately for an uncorrectable ECC error:

```
SUNW,UltraSPARC-III+: WARNING: [AFT1] EDU:ST Event detected by CPU0
at TL=0, errID 0x0000fef1.966c308e

                AFSR 0x00000008<EDU>.00000012 AFAR
0x00000060.4b417810

                Fault_PC 0x10151f3c Esynd 0x0012 SB0/P0/E1 J4300

NOTICE: [AFT1] CPU99 offlined due to xxU Event
```

When a processor encounters three or more distinct L2SRAM CEs (UCC, CPC, WDC, EDC) within a 24 hour period, the processor becomes a candidate for offlining.

```
NOTICE: [AFT1] Failed to offline CPU34 due to more than 2 xxC Events
in 24:00:00 (hh:mm:ss), will try again

NOTICE: [AFT1] CPU34 offlined due to more than 2 xxC Events in
24:00:00 (hh:mm:ss)
```

If a processor experiences a UCC and the Multiple Error (ME) bit is set, the processor becomes a candidate for offlining immediately.

```
WARNING: [AFT1] First Error UCC Event detected by CPU34 in Privileged
mode at TL>0, errID 0x0000015c.c7a4ce9b

[AFT1] errID 0x0000015c.c7a4ce9b Data Bit 95 was in error and
corrected

WARNING: [AFT1] UCC Event detected by CPU34 in Privileged mode at
TL>0, errID 0x0000015c.c7a4ce9b

[AFT1] errID 0x0000015c.c7a4ce9b Data Bit 95 was in error and
corrected

NOTICE: [AFT1] Failed to offline CPU34 due to UCC Event with ME set,
will try again

NOTICE: [AFT1] CPU34 offlined due to UCC Event with ME set
```

The following example message was generated by ScApp:

```
[DOM] Event: SF6800.L2SRAM.SERD.f.1b.10040000000091.f4470000

      CSN: 044M347B DomainID: A ADInfo:

      1.SF-SOLARIS-DE.5_10_on10-gate:05/29/2003

      Time: Mon Jun 02 23:34:59 PDT 2003

      FRU-List-Count: 1; FRU-PN: 3704125; FRU-SN: 090K01; FRU-LOC: /
N0/SB3/P3/E0

      Recommended-Action: Service action required
```

For more information on processor offlining capabilities for L2SRAM events, refer to the following Sun BluePrint document: *Solaris Operating System Availability Features* by Tom Chalfant (May 2004).

Processor Offlining Tuneables

The following table describes the processor offlining Solaris OS tuneables (configured in the `/etc/system` file).

TABLE 2 Processor Offlining Tuneables

Tunable	Value
`automatic_cpu_removal`	One of the following settings: • 0—disables processor offlining. • 1—enables processor offlining for multiple L2SRAM CEs (WDC, EDC, CPC, UCC). • 4—enables offlining only for uncorrectable L2SRAM events (UCU, CPU, WDU, EDU). • 5 (default)—enables offlining for both multiple L2SRAM CEs and uncorrectable L2SRAM events. This variable is derived by ORing together the bits that are set individually for multiple L2SRAM and uncorrectable L2SRAM events.
`cpu_remove_retry_seconds` `cpu_remove_retry_attempts`	When processor offlining is unsuccessful, retry again in `cpu_remove_retry_seconds`. Additionally, keep trying for `cpu_remove_retry_attempts`. Default values differ between Solaris OS versions: • For Solaris 8 OS, the default values for `cpu_remove_retry_seconds` and `cpu_remove_retry_attempts` are 30 and 2400, respectively. • For Solaris 9 OS, the default values for `cpu_remove_retry_seconds` and `cpu_remove_retry_attempts` are 5 and 24, respectively.
`ecc_indictment_mailbox_disable`	One of the following values: • 0 (default)—Indictments OK. • 1—Indictments suspect. Will still send indictments, but SC will not update CHS. • 2—Indictments disabled. No mailbox message sent. This tunable affects the SC to domain mailbox communication with ECC indictments due to L2SRAM CEs and UEs.
`ecc_indictment_mailbox_flags`	One of the following values: • Bit 0 (0x1)—Send DIMM indictments (Off by default). • Bit 1 (0x2)—Send L2SRAM correctable indictments (On by default). • Bit 2 (0x4)—Send L2SRAM uncorrectable indictments (On by default). This tunable affects the SC to domain mailbox communication with ECC indictments due to L2SRAM CEs and UEs.

Memory Page Retirement

The Solaris OS will attempt to remove a virtual memory page from service when:

- An *Uncorrectable ECC error* (also known as *UE*, *multi-bit*) is detected within that virtual memory page.
- The number of *Correctable ECC errors* (also known as *CE* or *single-bit ECC errors*) detected on the DIMM (that contains the virtual memory page experiencing CEs) exceeds a system-defined threshold within a specific period of time. (For Solaris 8 OS and Solaris 9 OS, this threshold is three CEs or more in 24 hours; for Solaris 10 OS, this threshold is three CEs or more in 72 hours.)

Uncorrectable ECC Errors

When an uncorrectable ECC error occurs:

- If the page is in use by the kernel, there is no way to recover correct data for that page, and Solaris will panic.
- If the page is in use by a user process:
  - With Solaris 8 and Solaris 9, that process is terminated first, and Solaris will then reboot. If there is a UE—for example, in a situation where data is read out of memory, but not actually used, such as in the case of DUE—Solaris will report the error and run without rebooting or panicking. Pages will be retired in this case. However, if that data is accessed later, then Solaris will panic or reboot with a UE depending on whether the access was from userland or kernel.
  - With Solaris 10, user-land uncorrectable errors will no longer bring down the running Solaris instance. Instead, the affected subsystem will be terminated and restarted.

For example, pages will be retired when a UE occurs, such as a DUE, when data is read out of memory but not actually used. The system will report the error and run without interruption with the page retired. However, if that data is accessed later, the system will panic or reboot with a UE, depending on whether the access was from user space or kernel.

Correctable ECC Errors

A particular algorithm, known as a “Leaky Bucket” algorithm, is used to track CEs on a per-DIMM basis. The default threshold and period of time settings are three CEs within a 24 hour period. A per DIMM count (the “bucket”) is incremented each time a CE error event is reported on that DIMM. The algorithm then decrements that count periodically—in other words, the “bucket” slowly “leaks”. If the number of CE events exceeds the threshold defined above, regardless of the “leak”, page retirement is initiated.

Removal of the virtual page from service prevents that area of memory from being allocated for use by the Solaris OS for the lifetime of this instance of the Solaris OS. A reboot will clear the page from being retired. The Solaris OS also communicates this information to the SC, which updates the SoftError record in the DIMM's Dynamic FRU-ID record.

The “Leaky Bucket” is designed to allow the Soft Error Rate Discrimination to determine whether the CEs are not due to soft errors (radiation). The “leak” requires that the errors occur with a certain amount of frequency to exceed the threshold.

In addition to the “Leaky Bucket” SERD algorithm, Sun has also implemented the new DIMM replacement policy as detailed in the Sun InfoDoc #79928, “Sun Memory DIMM Replacement Policy,” which is available at:

http://sunsolve.sun.com/search/document.do?assetkey=1-9-79928-1

Memory Page Retirement Tuneables

The following table describes memory page retirement Solaris OS tuneables (configured in `etc/system`).

TABLE 3 Memory Page Retirement Tuneables

Tunable	Value
`automatic_page_removal`	One of the following values: • 0—disables the page retirement feature. • 1—enables the page retirement feature. The default value depends upon the kernel patch release and system type.
`ecc_softerr_interval` `ecc_softerr_limit`	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. The default values for these tuneables are 1440 and 2 respectively.

These variables can be changed through `/etc/system`, `mdb`, or `kadb`. Changing them via the `/etc/system` file is the preferred manner for ensuring that they remain persistent across reboots of the operating system.

For more information on the page retirement capability within the Solaris OS, refer to the following Sun BluePrint: *Solaris Operating System Availability Features* (May 2004), by Tom Chalfant.

Conclusion

Sun Fire Servers implement a number of Predictive Fault Monitoring tools that, when properly understood and utilized, increase the overall availability and reliability of these systems. These include monitoring tools that will detect and announce incipient failures within the hardware, and reactive algorithms that seek to remove a faulty FRU from the system configuration before it causes a system outage. Understanding the operation of these subsystems is critical to all personnel maintaining or operating Sun Fire Servers.

References and Related Sources

- Sun BluePrints

  http://www.sun.com/blueprints/
- Solaris Operating System Availability Features

  http://www.sun.com/blueprints/0504/817-7039.pdf
- Sun InfoDoc #79928, “Sun Memory DIMM Replacement Policy”

  http://sunsolve.sun.com/search/document.do?assetkey=1-9-79928-1

About the Authors

This section provides more information about the authors of this document.

Dave Re

Dave Re works for Sun as a member of the Sun Support Services Product Technical Support (PTS) team, responsible for the Sun Fire Midrange (3800-6800) product line. Before joining PTS, Dave supported Sun's mission critical customers at a local and area level, based out of Atlanta, GA. Prior to working for Sun, Dave worked as a Sun and IBM Mainframe systems administrator, and as a pre- and post-sales professional services engineer. Dave currently resides in Austin, TX, with his wife, three dogs, and horse. He enjoys playing golf, pursuing the artistic challenge of photography, and is an avid musician.

Kumar Loganathan

Kumar Loganathan is a Senior Staff Engineer in the PTS, Strategic Solutions Group. Kumar joined Sun in 1999. Prior to joining Sun, Kumar worked as Principal Systems Engineer at Proquest Information and Learning (formerly Bell & Howell). Prior to that, he worked at Ford Motor Company and General Motors where he held senior technical positions in software and systems engineering. Kumar has several years of experience with UNIX and Sun systems as a software and systems engineer. Kumar joined the HES/CTE/CPRE/PTS organization in 1999 where he has since spent most of his time. Kumar also managed a sustaining group in the Solaris OS organization before rejoining the PTS organization. Kumar has a Bachelor's degree in Electronics and Communications Engineering and a Master's degree in Computer Science.

Acknowledgements

The authors would like to recognize the following individuals for their contributions to this article:

- **Kenny C. Gross** is a Distinguished Engineer in the RAS Computer Analysis Laboratory, specializing in proactive fault monitoring and dynamic system characterization for improving the RAS and quality-of-service of enterprise computing systems. Kenny was the driving force in developing the technique on which CVTM is based to combine telemetry with advanced pattern recognition in high-end servers to monitor temperatures, voltages, currents, and a variety of performance metrics so problems could be detected before they resulted in a system failure.
- **Keith Whisnant** is a Research Staff Scientist, also in the RAS Computer Analysis Laboratory, who worked with Kenny in developing the CVTM concept, and in the design and implementation of this concept into the Continuous System Telemetry Harness (CSTH) and system controller software for mid-range and high-end servers.

  The work performed by these two gentlemen was indispensable in understanding the theory and operations of CVTM, and was relied upon heavily to provide the CVTM content in this document.
- **Domingo Goyena**, **Brian Jackson**, and **Lennart Karlsson** graciously proofread this document for us. Their assistance was crucial in providing a clear, concise, accurate document that conveys the content we set out to provide. Many thanks to these gentlemen!

Ordering Sun Documents

The SunDocs℠ 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`