

Getting to know the Solaris Fault Management Architecture (FMA)

Ryan Matteson

matty91@gmail.com

<http://prefetch.net>

Overview

- Tonight we are going to discuss the Solaris fault management architecture, as well as several tools that can be used to notify you when hardware fails
- I plan to split my 60-minutes into two parts. The first part will provide an overview of the software, and the second half will show FMA in action

The imperfect world

- Diagnosing hardware faults has historically been a royal PITA, and extremely prone to error (anyone ever replaced a DIMM when the CPU or power supply were actually faulty?)
- With advances in hardware error correction and reporting, you would think there would be an automated way to diagnose hardware faults, isolate faulty components, and send alerts to let operational staff know that a problem exists

Welcome to FMA

- The Fault Management Architecture (FMA) and Service Management Facility (SMF) were introduced in Solaris 10 to allow systems to self heal themselves when hardware and software fail
- FMA provides automated diagnosis of faulty hardware, and can take proactive measures to correct (e.g., offline a CPU) hardware-related faults
- SMF provides automated diagnosis of software faults, and can take proactive measures to correct (e.g., restart a process) software-related faults
- The remainder of this talk will focus on FMA

How does FMA work?

- Kernel sends error events to the fault manager daemon (fmd), which routes the events to modules based on subscriptions
- Two main types of modules:
 - **Diagnosis engines** take the raw error telemetry events and provide automated problem diagnosis based on the symptoms
 - **Agents** respond to a given diagnosis by taking one or more actions (e.g., offline a faulty CPU or memory page)
- When problems are diagnosed, the fault manager will log a fault diagnosis message that contains a case id (represented by a UUID) which references the problem, a description of the problem, and a link to a knowledge base article that describes the problem and the set of actions that will be required to fix the problem

How does FMA work? (cont.)

- The example below shows what a typical fault diagnosis message looks like:

SUNW-MSG-ID: SUN4U-8000-AC, **TYPE:** Fault, **VER:** 1, **SEVERITY:** Major

EVENT-TIME: Thu Feb 26 18:08:26 PST 2004

PLATFORM: SUNW,Sun-Fire-V440, **CSN:** -, **HOSTNAME:** boz

SOURCE: cpumem-diagnosis, **REV:** 0.1

EVENT-ID: 322fe6d5-fe14-6a73-b802-cc6c30b2afcd

DESC: The number of errors associated with this CPU has exceeded acceptable levels.

Refer to <http://sun.com/msg/SUN4U-8000-AC> for more information.

AUTO-RESPONSE: An attempt will be made to remove the affected CPU from service.

IMPACT: Performance of this system may be affected.

REC-ACTION: Schedule a repair procedure to replace the affected CPU.

What diagnosis engines are currently available?

- There are diagnosis engines available for a number of CPU architectures:
 - UltraSPARC III and above
 - UltraSPARC T1 and T2
 - Intel Xeons
 - AMD Opterons
 - AMD Athlons
- Diagnosis engines are also available for PCI and PCI express buses, as well as a number of lead node drivers (e.g., Ethernet adapters, HBA drivers, etc.)
- A disk-transport diagnosis engine was recently added to Nevada to daignose SATA and SCSI disk drives errors using SMART data

Which agents are currently available?

- AMD, Intel and SPARC agents are available to retire CPUs and memory pages
- Disk transport and I/O agents are available to retire disk drives and faulty I/O devices
- The ZFS agent allows the ZFS file system to enable hot spares in response to disk failures
- More agents to come ...

What diagnosis engines and agents are coming to an opensolaris near you?

- Sensor project will provide fault diagnosis based on sensor data (e.g., increase fan speeds in response to excessive heat, offline disk drives due to an unacceptable number of ECC errors, etc.)
- More leaf drivers will be hardened to send error telemetry data when they detect faults
- Software diagnosis engines will be developed to diagnose software faults, and take appropriate actions (e.g., restart a process that is using a page of memory that had uncorrectable ECC errors)

Viewing the active diagnosis engines and agents

- The `fmadm` utilities “`config`” option can be used to view the list of diagnosis engines and agents that are active on a system:

```
$ fmadm config
```

MODULE	VERSION	STATUS	DESCRIPTION
<code>cpumem-retire</code>	1.1	active	CPU/Memory Retire Agent
<code>disk-transport</code>	1.0	active	Disk Transport Agent
<code>eft</code>	1.16	active	eft diagnosis engine
<code>fmd-self-diagnosis</code>	1.0	active	Fault Manager Self-Diagnosis
<code>io-retire</code>	2.0	active	I/O Retire Agent
<code>snmp-trapgen</code>	1.0	active	SNMP Trap Generation Agent
<code>sysevent-transport</code>	1.0	active	SysEvent Transport Agent
<code>syslog-msgs</code>	1.0	active	Syslog Messaging Agent
<code>zfs-diagnosis</code>	1.0	active	ZFS Diagnosis Engine
<code>zfs-retire</code>	1.0	active	ZFS Retire Agent

Fault manager logs

- The fault manager maintains two log files:
 - The ***error log*** contains a list of errors events that have been sent to the fault manager daemon
 - The ***fault log*** contains a list of problems that have been diagnosed and repaired
- The fault log can be viewed by running fmdump:
\$ fmdump
- The error log can be viewed with fmdump's “-e” option:
\$ fmdump -e
- Fmdump also has a “-u” option to limit the output to a specific UUID, a “-T” option to display events that occurred during a specific timeframe, and “-v” and “-V” options to display verbose output

Viewing faulty components

- To view the faulted resources on a system, the `fmadm` utility can be run with the “faulty” option:

```
$ fmadm faulty
```

```
STATE RESOURCE / UUID
```

```
-----  
degraded dev:///pci@8,700000/pci@2  
a0461e5e-4356-ca7b-ee83-c66816b9caba
```

```
-----  
degraded dev:///pci@8,700000/pci@3  
a0461e5e-4356-ca7b-ee83-c66816b9caba
```

```
< ..... >
```

- The output above contains the faulted component, the unique case identifier, and the state (i.e., ok, unknown, degraded, faulted,) the component is currently in

Viewing all faults

- To view all of the faults (which includes silent faults) that have occurred on a system, you can run `fmadm` with the “`faulty`” and “`-a`” options:

```
$ fmadm faulty -a
```

- This will display all cached faults, including faults that don't necessarily indicate a component is faulty (e.g., a single page being retired)

Repairing faults

- Once a problem has been resolved (e.g., a CPU has been replaced), the `fmadm` utility can be run with the “repair” option and the UUID to repair

```
$ fmadm repair a0461e5e-4356-ca7b-ee83-c66816b9caba
```

- This will update the fault manager’s resource cache to indicate that no problems are present with the components associated with the UUID

Getting notified when things break

- The fault management architecture wouldn't be all that useful if it didn't provide methods to alert people when problems occur
- The fault manager logs diagnosis messages to syslog and the system console each time a fault is diagnosed, and can be configured to generate SNMPv1 traps or SNMPv2 notifications
- FMA currently doesn't have built-in support for email notifications, but third party tools are available to send email when faults occur

Enabling SNMP support

- To configure FMA to send SNMPv1 traps, you can add one or more ***trapsink*** directives to the SNMP daemon's `snmpd.conf` configuration file:

```
trapsink 192.168.1.100 public 162
```

```
trapsink 192.168.1.101 public 162
```

- To configure FMA to send SNMPv2 notification, you can add one or more ***informsink*** directives to the SNMP daemon's `snmpd.conf` configuration file:

```
informsink 192.168.1.100 public 162
```

```
informsink 192.168.1.101 public 162
```

Getting email when hardware faults occur

- Since FMA doesn't contain built-in support for email notifications, I developed a shell script to send email when the fault manager diagnoses a fault
- The script is designed to be run from cron and can be downloaded from my website:
<http://prefetch.net/code/fmadmnotifier>

Hardware testing utilities

- **SunVTS**
<http://www.sun.com/oem/products/vts/>
- **Memtest86+**
<http://www.memtest.org/>
- **Cpuburn**
<http://pages.sbcglobal.net/redelm/>
- **Smartmontools**
<http://smartmontools.sourceforge.net/>
- **Sys_basher**
http://www.polybus.com/sys_basher_web/

FMA Resources

- **FMA demo kit**
<http://opensolaris.org/os/community/fm/demokit/>
- **FMA programmer's reference manual**
<http://www.opensolaris.org/os/community/fm/FMDPRM.pdf>
- **FMA MIB**
<http://opensolaris.org/os/community/fm/mib/sun-fm-mib.mib>
- **Mike Shapiro's FMA presentation**
<http://blogs.sun.com/mws/resource/fma-osug.pdf>

Conclusion

- FMA is an incredibly powerful technology, and should make every admin smile
- Long gone are the days of confusing error messages, days of fruitless hardware debugging, extended downtimes and the frustration that goes along with it!

Questions?