System Management Interrupt Free Hardware

Keith Mannthey
kmannth@us.ibm.com

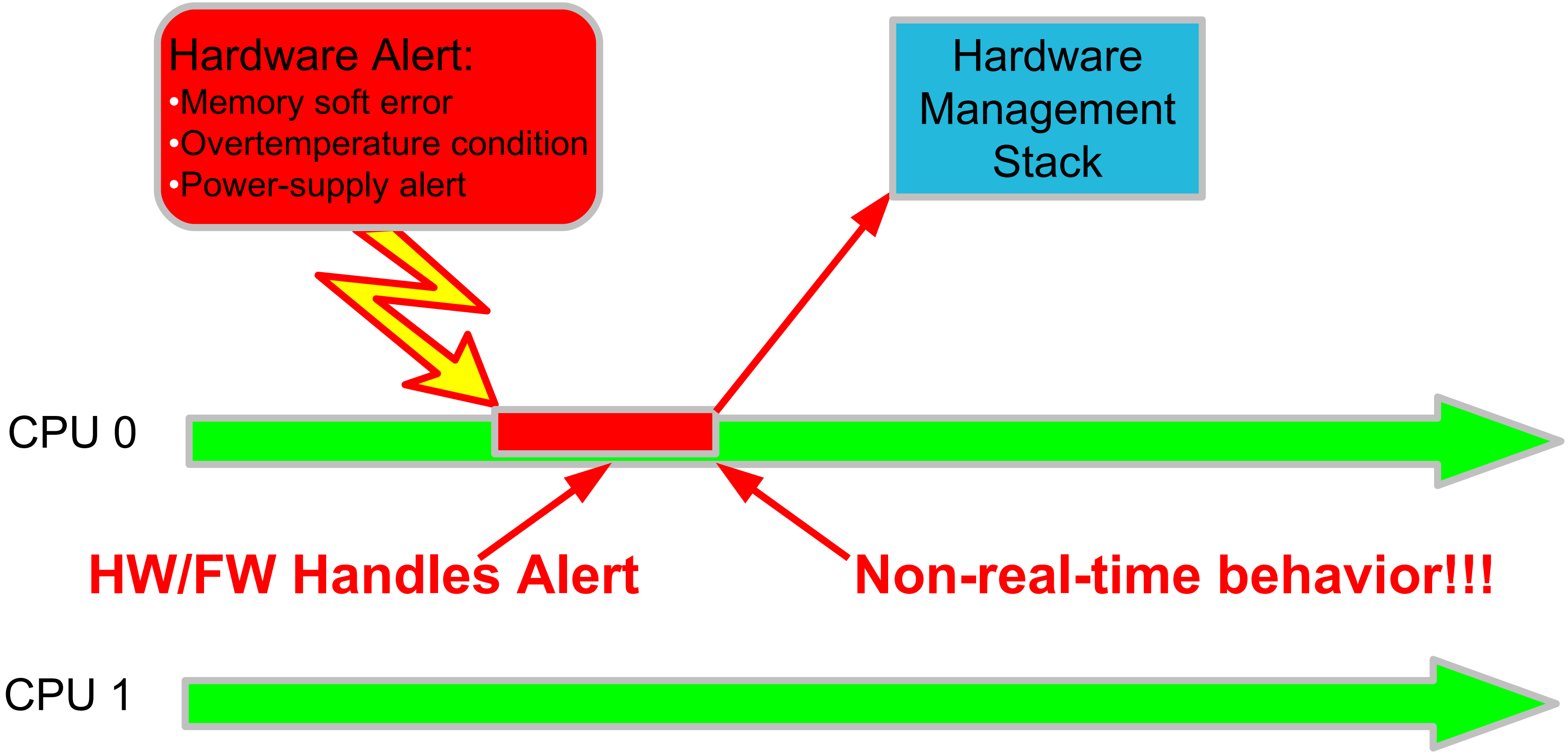
Agenda

- Overview of System Management Interrupts (SMI)
- Overview of SMI-Free Solution
- Firmware support -- BIOS
- Firmware support – Baseboard Management Controller (BMC)
- Operating System Support
- Linux Interactions
- Final Overview

Overview of System Management Interrupts (SMI)

- SMIs are used to perform a variety of tasks at the CPU level
  - Reporting of hardware errors (fatal and nonfatal)
  - Thermal throttling, Power capping, External Policies
  - Remote Consoles, System Health Checks
  - Programed by FW developers
- The nature of these interrupts causes latencies
  - Not optimal for Real Time Systems
  - No Operating System (OS) notification or control
  - Hard to detect, process of elimination detection only.
  - Source of unwanted/unaccounted latencies in a Real Time Systems

Non-Real-Time Hardware Error Behavior



Hardware Alert:
- Memory soft error
- Overtemperature condition
- Power-supply alert

HW/FW Handles Alert

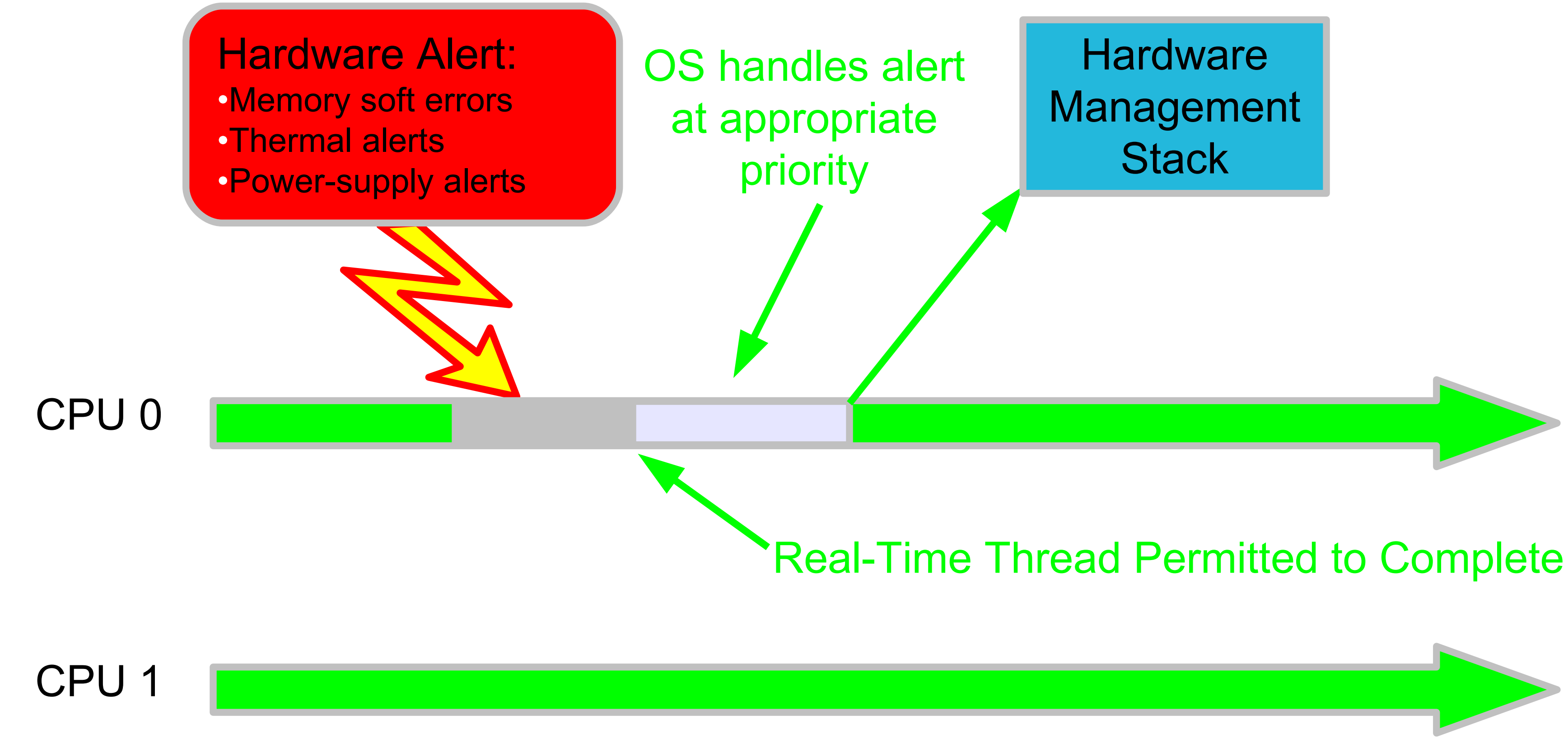
Non-real-time behavior!!!

There is nothing that the OS or higher-level software can do to make up for this HW/FW non-realtime behavior.

Overview of SMI-Free Solution

- Provide a system that has no non fatal SMIs
  - Deal with correctable ECC memory errors
  - Disable external CPU throttling
    - Power consumption
    - Thermal protection
- Design Goals
  - Protect the health of the system
  - Correctly report errors for serviceability
  - Do not block the OS
- BIOS, BMC (firmware) and OS work together
  - OS manages firmware and reports errors
  - Firmware is involved with fatal errors

IBM System x Real-Time Hardware Error Behavior



Hardware Alert:

- Memory soft errors
- Thermal alerts
- Power-supply alerts

OS handles alert at appropriate priority

Hardware Management Stack

Real-Time Thread Permitted to Complete

The OS and higher-level software now see Real-Time behavior.

Firmware support -- BIOS

- BIOS no longer registers non fatal SMI handlers with the CPU
  - All non-fatal events are handled by the OS
    - Correctable ECC memory errors
  - Fatal events are still handled by the BIOS
    - Non recoverable hardware events
      - Non Correctable ECC memory errors
      - Fatal PCI bus errors
- BIOS provides a way to enter / exit SMI-free mode
  - Runtime state change
  - Interface is used by the OS
  - Currently a table in the Extend BIOS Data Area

Firmware support
Baseboard Management Controller (BMC)

- BMC == Service Processor
- High level polices enforced via FW are manged by the BMC
- BMC no longer requests to throttle the CPU
  - Throttling causes unacceptable latencies on real time systems
    - Power capping disabled
    - Thermal throttling disabled
    - Acoustical mode disabled
- BMC still protects the system from critical over temp
  - Hard power off; normal protection behavior
- BMC provides a way to enter / exit SMI-free mode
  - BMC runtime state change via the OS with IPMI
    - Yea for standard interfaces!

Operating System Support

- New OS service “ibm-prtm” manages entering and exiting the SMI-free state
  - Manages BMC and BIOS interfaces
  - Starts/stops OS daemon that reports ECC memory errors
  - Service is a non real time task
- Reports correctable ECC memory errors
  - Support standard service path via IPMI
    - On our system LED LightPath error indicators and entry in the BMC logs
- With EDAC drivers; detect and report other system errors
  - Reported in /var/log/messages and the system console
    - We really only care about incrementing ECC error counts but we get everything

Linux Kernel Interactions

- BIOS / UFI state change: IBM RTL driver
  - Creates a small sysfs interface
  - There is small table in the EDBA region that get manipulated
  - Still working on getting it upsteam :(
- Currently EDAC for ECC memory error detection
  - amd64_edac, k8_edac (old)
  - I5000, i7core_edac
  - Live error creation on current cpus, some development and plenty of test/debugging.

ECC Error Memory Mapping Fun

- Mapping what ever a given chipset/cpu reports it actual dimm number (the one printed on the board) is non trivial.
  - Every System is different
  - No standard table to describing the mapping
    - DMI table device order works in SOME systems
    - FW writers do have this information it is just not exported
  - Mappings currently developed with trial and error
    - Live debug dimm testing
  - Mappings have changed as drivers develop

Linux: Next steps

- New CPUs and Systems
  - Nehalem EX, MCE architecture; move away from EDAC?
  - Interrupt driven hardware error reporting
  - Explore cpu visualization features as they relate to SMIs
- Long term solution for User space bits
  - As the number of kernel versions increase and the number of systems increase the matrix of mapping increase
  - As ECC detection drivers change mappings change
- UFI based Real-Time state change
  - Presently FQ only supports “Legacy” BIOS EBDA state change method

Supported Hardware

- Lots of IBM Blades
  - LS21 (AMD Dual Core Rev F)
  - LS22 (AMD Quad Core Rev 10)
  - HS21xm (Intel i5000 Xeon)
  - HS22 (Intel i7core 55XX Xeon)
- 2 Rackable Systems
  - IBM x3650m2 2U (Intel i7core 55xx Xeon)
  - IBM x3550m2 1U (Intel i7core 55xx Xeon)
- OS's are RedHat MRG and SuSE SLERT

Final Overview

- FW and the OS work together to provide a serviceable solution for running without non fatal SMIs
  - Improved real time performance during non-fatal hardware events
- Currently supported OS
  - MRG, SLERT
  - Work is covered by the GPL
- Current hardware support
  - Blades and Rack mounts
  - AMD and Intel currently

Blade Center Hardware/Firmware Overview

- BladeCenter H (BCH)
  - Users interact with the BCH not with the blades
    - Power on/off
    - Hardware information
    - Thermal and Power Policy

- IBM Blade
  - BMC Communicates with the BCH and interacts with the CPU to set user policy
    - CPU throttling for power and thermal issues
  - BIOS Manages the CPU
    - Source of ALL SMIs
    - Error reporting

IBM Blade Center Specific Considerations

- IBM PowerExecutive features not supported
  - Thermal Considerations:
    - System will not throttle the system in an over temp situation
    - System will do a hard shutdown at critical temperature
    - SNMP and polling of the hardware can provide temperature status information
  - Power Considerations:
    - Systems in real time mode will not automatically throttle to reduce power usage

Legal Statement

- This work represents the view of the author and does not necessarily represent the view of IBM.
- IBM, IBM (logo), e-business (logo), pSeries, e (logo) server, and xSeries are trademarks or registered trademarks of International Business Machines Corporation in the United States and/or other countries.
- Linux is a registered trademark of Linus Torvalds.
- Other company, product, and service names may be trademarks or service marks of others.

System Management Interrupt Free Hardware

Keith Mannthey
kmannth@us.ibm.com

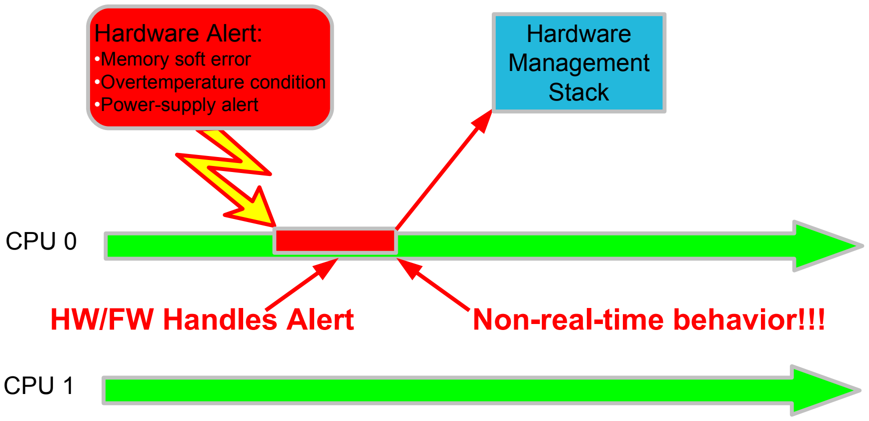
Agenda

- Overview of System Management Interrupts (SMI)
- Overview of SMI-Free Solution
- Firmware support -- BIOS
- Firmware support – Baseboard Management Controller (BMC)
- Operating System Support
- Linux Interactions
- Final Overview

Overview of System Management Interrupts (SMI)

- SMIs are used to perform a variety of tasks at the CPU level
  - Reporting of hardware errors (fatal and nonfatal)
  - Thermal throttling, Power capping, External Policies
  - Remote Consoles, System Health Checks
  - Programed by FW developers
- The nature of these interrupts causes latencies
  - Not optimal for Real Time Systems
  - No Operating System (OS) notification or control
  - Hard to detect, process of elimination detection only.
  - Source of unwanted/unaccounted latencies in a Real Time Systems

Non-Real-Time Hardware Error Behavior

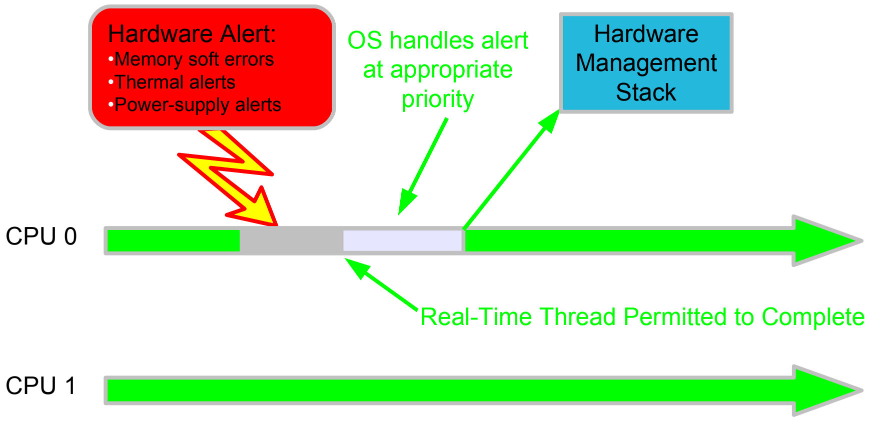


There is nothing that the OS or higher-level software can do to make up for this HW/FW non-realtime behavior.

Overview of SMI-Free Solution

- Provide a system that has no non fatal SMIs
  - Deal with correctable ECC memory errors
  - Disable external CPU throttling
    - Power consumption
    - Thermal protection
- Design Goals
  - Protect the health of the system
  - Correctly report errors for serviceability
  - Do not block the OS
- BIOS, BMC (firmware) and OS work together
  - OS manages firmware and reports errors
  - Firmware is involved with fatal errors

IBM System x Real-Time Hardware Error Behavior



Hardware Alert:

- Memory soft errors
- Thermal alerts
- Power-supply alerts

OS handles alert at appropriate priority

Hardware Management Stack

CPU 0

Real-Time Thread Permitted to Complete

CPU 1

The OS and higher-level software now see Real-Time behavior.

Firmware support -- BIOS

- BIOS no longer registers non fatal SMI handlers with the CPU
  - All non-fatal events are handled by the OS
    - Correctable ECC memory errors
  - Fatal events are still handled by the BIOS
    - Non recoverable hardware events
      - Non Correctable ECC memory errors
      - Fatal PCI bus errors
- BIOS provides a way to enter / exit SMI-free mode
  - Runtime state change
  - Interface is used by the OS
  - Currently a table in the Extend BIOS Data Area

Firmware support
Baseboard Management Controller (BMC)

- BMC == Service Processor
- High level polices enforced via FW are manged by the BMC
- BMC no longer requests to throttle the CPU
  - Throttling causes unacceptable latencies on real time systems
    - Power capping disabled
    - Thermal throttling disabled
    - Acoustical mode disabled
- BMC still protects the system from critical over temp
  - Hard power off; normal protection behavior
- BMC provides a way to enter / exit SMI-free mode
  - BMC runtime state change via the OS with IPMI
    - Yea for standard interfaces!

Operating System Support

- New OS service “ibm-prtm” manages entering and exiting the SMI-free state
  - Manages BMC and BIOS interfaces
  - Starts/stops OS daemon that reports ECC memory errors
  - Service is a non real time task
- Reports correctable ECC memory errors
  - Support standard service path via IPMI
    - On our system LED LightPath error indicators and entry in the BMC logs
- With EDAC drivers; detect and report other system errors
  - Reported in /var/log/messages and the system console
    - We really only care about incrementing ECC error counts but we get everything

Linux Kernel Interactions

- BIOS / UFI state change: IBM RTL driver
  - Creates a small sysfs interface
  - There is small table in the EDBA region that get manipulated
  - Still working on getting it upsteam :(
- Currently EDAC for ECC memory error detection
  - amd64_edac, k8_edac (old)
  - I5000, i7core_edac
  - Live error creation on current cpus, some development and plenty of test/debugging.

ECC Error Memory Mapping Fun

- Mapping what ever a given chipset/cpu reports it actual dimm number (the one printed on the board) is non trivial.
  - Every System is different
  - No standard table to describing the mapping
    - DMI table device order works in SOME systems
    - FW writers do have this information it is just not exported
  - Mappings currently developed with trial and error
    - Live debug dimm testing
  - Mappings have changed as drivers develop

Linux: Next steps

- New CPUs and Systems
  - Nehalem EX, MCE architecture; move away from EDAC?
  - Interrupt driven hardware error reporting
  - Explore cpu visualization features as they relate to SMIs
- Long term solution for User space bits
  - As the number of kernel versions increase and the number of systems increase the matrix of mapping increase
  - As ECC detection drivers change mappings change
- UFI based Real-Time state change
  - Presently FQ only supports “Legacy” BIOS EBDA state change method

Supported Hardware

- Lots of IBM Blades
  - LS21 (AMD Dual Core Rev F)
  - LS22 (AMD Quad Core Rev 10)
  - HS21xm (Intel i5000 Xeon)
  - HS22 (Intel i7core 55XX Xeon)
- 2 Rackable Systems
  - IBM x3650m2 2U (Intel i7core 55xx Xeon)
  - IBM x3550m2 1U (Intel i7core 55xx Xeon)
- OS's are RedHat MRG and SuSE SLERT

Final Overview

- FW and the OS work together to provide a serviceable solution for running without non fatal SMIs
  - Improved real time performance during non-fatal hardware events
- Currently supported OS
  - MRG, SLERT
  - Work is covered by the GPL
- Current hardware support
  - Blades and Rack mounts
  - AMD and Intel currently

Blade Center Hardware/Firmware Overview

- BladeCenter H (BCH)
  - Users interact with the BCH not with the blades
    - Power on/off
    - Hardware information
    - Thermal and Power Policy

- IBM Blade
  - BMC Communicates with the BCH and interacts with the CPU to set user policy
    - CPU throttling for power and thermal issues
  - BIOS Manages the CPU
    - Source of ALL SMIs
    - Error reporting

IBM Blade Center Specific Considerations

- IBM PowerExecutive features not supported
  - Thermal Considerations:
    - System will not throttle the system in an over temp situation
    - System will do a hard shutdown at critical temperature
    - SNMP and polling of the hardware can provide temperature status information
  - Power Considerations:
    - Systems in real time mode will not automatically throttle to reduce power usage

Legal Statement

- This work represents the view of the author and does not necessarily represent the view of IBM.
- IBM, IBM (logo), e-business (logo), pSeries, e (logo) server, and xSeries are trademarks or registered trademarks of International Business Machines Corporation in the United States and/or other countries.
- Linux is a registered trademark of Linus Torvalds.
- Other company, product, and service names may be trademarks or service marks of others.