Case Study: Alternatives for SMM Usage in Intel Platforms

Spring 2019 UEFI Plugfest
April 8-12, 2019
Sarathy Jayakumar, Principal Engineer (Intel Corp.)
Agenda

- Problem Summary
- OS View of SMM
- Categories of SMM Handlers
- What about a Driver-based model
- Platform Runtime Mechanism
- Case Study: Using PRM for Correctable Error Handling
- Call to action
Problem Summary

• System Management Mode (SMM) issues to address
  – Degrades performance & quality of service (QoS)
    • SMM latency increases with core count
    • Firmware-based reliability of service (RAS) features
  – SMM model adds complexity to firmware
    • Multi-core asynchronous events, no concept of interrupt priority
      or reentrancy, race conditions, handler code, ...
  – Security concerns due to higher SMM privilege level
OS View of SMM

- OS / Drivers
  - ACPI Tables (ex. PCCT)
  - UEFI Runtime (RT) Services (ex: SetVariable)
  - ACPI DSM Methods (ex: ARS)

- SMM

- Platform Hardware (Processor, Memory, I/O, ...)

www.uefi.org
OS View of SMM

OS / Drivers

- ACPI Tables (ex. PCCT)
- UEFI Runtime (RT) Services (ex: SetVariable)
- ACPI DSM Methods (ex: ARS)

SMM

Platform Hardware (Processor, Memory, I/O, ...)

Software SMI triggers are transparent to the OS

Hardware SMI triggers are transparent to the OS

ACPI/RT services provide platform abstraction

Software SMM elimination strategy should not impact OS to ‘Platform Abstraction’ interface
Categories of SMM Handler

**Current Model**

1: Software SMI that do not require SMM privileges (ex: Address translation, NVDIMM DSMs, etc.)

2: Software SMI that require SMM privileges

3: Hardware SMI and RAS Handlers that do not require privileges

4: Hardware SMI and RAS Handlers that require privileges

- **ASL + PRM**
  - Capsule Update / OOB
  - Firmware Update
  - UEFI Variable Services, Firmware Update

- **OOB + PRM**
  - Capsule Update / OS Driver / OOB

- **OOB**

www.uefi.org
What about a Driver-based Model?

• Do not want platform knowledge in OS driver
• Requires intimate platform/silicon knowledge (ex: Address Translation for RAS)
• Variance between platform implementation / generation
Examples of Driver-based Issues

• **PSHED Plug-in:** Not a viable deployment model due to ACPI abstraction, which uses SMI for complex tasks.

• **Address Translation:** Originally pushed to EDAC drivers. OS vendors prefer ACPI to keep driver generic. ACPI relies SMM to handle complex algorithms.

• **NVDIMM Drivers:** Uses ACPI to keep NVDIMM drivers generic. Relies on ACPI (again) which (still) uses SMM to handle complex tasks (this is a trend).
Platform Runtime Mechanism (PRM)

- Mechanism to invoke native code from ACPI
- Uses ASL as a landing point for runtime events
- ASL will invoke PRM if required ("ASL Assist")

Note: PRM is not a new capability. It is based on combining existing capabilities.
Case Study: Using PRM for Correctable Error (CE) Handling

- Memory Error Subsystem
- ERR0 Pin
- Peripheral Controller Hub (PCH)
- GPIO
- SMI
- SCI
- Lxx Method
- PRM Build HEST
- APEI Notify to OS
- www.uefi.org
Memory correctable Error (CE)

SMI Based Firmware First CE Handling

Memory Subsystem → SMI Handler → SCI Notify to OS → OS Consumes HEST Table → Event Log

Build ACPI HEST Table
Call to Action

• Work together to accelerate SMM reduction.
• Move software SMM Handlers to PRM.

• Bridge driver and sample PRM handler available in GitHub:
  • https://github.com/tianocore/edk2-staging/tree/PRMCaseStudy

• Please review & provide feedback!
Glossary

PCCT – Platform Communication Channel Table
DSM – Device Specific Methods
ARS – Address Range Scrubbing
OOB – Out Of Band
PRM – Platform Runtime Mechanism
PSHED – Platform Specific Hardware Error Driver
EDAC – Error Detection And Correction
SCI – System Configuration Interrupt
HEST – Hardware Error Sources Table
APEI – ACPI Platform Error Interfaces

*Other names and brands may be claimed as the property of others
© Intel Corporation
Thanks for attending the 2019 Spring UEFI Plugfest

For more information on UEFI Forum and UEFI Specifications, visit http://www.uefi.org

presented by

www.uefi.org