Firmware Innovations Towards Cloud

Intel’s Implementation for Open System Firmware

Song, Edmund | Software Architect
25th June, 2019
CONTENTS

01 UEFI Based Open System Firmware
02 Firmware Innovations Towards Cloud
03 Call To Action
UEFI Based Open System Firmware

MinPlatform:
Platform (board) Specific Code at
https://github.com/tianocore/edk2-platforms

Firmware Support Package:
Intel binaries for board invariant Si code at
https://github.com/intelfsp

EDKII:
Existing upstream/open source core at
https://github.com/tianocore/edk2

Interfaces:
Platform interface tables to support OS boot
https://uefi.org

Cloud OS
(Infrastructure and Resource Management)

Workload
- Data Analytic
- E-commerce
- Database
- AI

HW & FW integrated as cloud infrastructure capabilities

Agile, Open and Standard Firmware Design Model to Support Cloud Requirements
MinPlatform

- Open source package with minimum set of platform code needed to realize server with white box configuration
- Offer buildable and bootable “white box” configuration using Intel® FSP
- Reduce volume of “closed source” needed to support Server products
FSP (Firmware Support Package)

- Binary package to provide processor and chipset initialization easily be incorporated into industry boot loader framework (e.g. core boot, Tiano Core etc.)
- To abstract the complexity of silicon initialization and publicly distribute binaries of silicon code

Firmware Innovations Towards Cloud

Power and Performance
- Fine-grained HW Knobs. Mgmt.
- Runtime Configuration
- Workload Optimal Platform Profile
- Platform Telemetry

Service Availability & Reliability
- Minimized Downtime (Non-reboot, avoid reset, fast boot)
- Firmware Resiliency
- Enhanced RAS Capabilities

Sustainability & Maintainability
- Scalable Configuration and Update
- Remote Diagnostics
- Autonomous Error Collection

Service Availability & Reliability
- Minimized Downtime (Non-reboot, avoid reset, fast boot)
- Firmware Resiliency
- Enhanced RAS Capabilities

Sustainability & Maintainability
- Scalable Configuration and Update
- Remote Diagnostics
- Autonomous Error Collection

Firmware Innovations Towards Cloud

Power and Performance
- Fine-grained HW Knobs. Mgmt.
- Runtime Configuration
- Workload Optimal Platform Profile
- Platform Telemetry

Service Availability & Reliability
- Minimized Downtime (Non-reboot, avoid reset, fast boot)
- Firmware Resiliency
- Enhanced RAS Capabilities

Sustainability & Maintainability
- Scalable Configuration and Update
- Remote Diagnostics
- Autonomous Error Collection
Platform Runtime Mechanism (PRM)

SMM (System Management Mode)
- Operating mode all threads/cores execution suspended
- SMM latency increase with more core count - implicated performance degradation
- Security concerns due to higher SMM privilege

PRM
- Enable firmware runtime functions in kernel space instead of SMM
- ACPI abstract layer to allow OS to invoke runtime code w/o awareness of platform specific details

Sample PRM handler and ACPI Bridge Driver available in GitHub: https://github.com/tianocore/edk2-staging/tree/PRMCaseStudy

PRM moving SW SMI (w/o SMM privilege) to kernel space
Boot Time Optimization

**Boot Flow**

- ACM
- SEC uCode Load, CPU Init
- UPI Discovery Setup
- MRC Discovery, Training, ECC Init, Memory Test, Address Map
- MP Init, Platform Init
- PCle Device Enum.
- Publish Memory Map, ACPI, SMBIOS
- Option ROM
- Boot Device Selection
- OS Handoff

**Optimized Boot Time with MinPlatform Package on Mt. Olympus (OCP Board)**

<table>
<thead>
<tr>
<th>Phase</th>
<th>Duration (ms)</th>
<th>Elapsed Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 SEC</td>
<td>69.8667</td>
<td>0.0000</td>
</tr>
<tr>
<td>1 PEI</td>
<td>1519.7666</td>
<td>69.8667</td>
</tr>
<tr>
<td>2 DXE</td>
<td>823.9279</td>
<td>1589.6533</td>
</tr>
<tr>
<td>3 BDS</td>
<td>2238.0843</td>
<td>2413.5812</td>
</tr>
<tr>
<td>4 Total(UEFI)</td>
<td>4651.6654</td>
<td>~6000</td>
</tr>
<tr>
<td>5 Windows Boot Logo</td>
<td>~18000</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Phase</th>
<th>Duration (ms)</th>
<th>Elapsed Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 SEC</td>
<td>2449.9778</td>
<td>0.0000</td>
</tr>
<tr>
<td>1 PEI</td>
<td>3552.6876</td>
<td>2449.9778</td>
</tr>
<tr>
<td>2 DXE</td>
<td>820.3926</td>
<td>6002.6654</td>
</tr>
<tr>
<td>3 BDS</td>
<td>2989.2274</td>
<td>6823.0580</td>
</tr>
<tr>
<td>4 Total(UEFI)</td>
<td>9812.2854</td>
<td>~11000</td>
</tr>
<tr>
<td>5 Windows Boot Logo</td>
<td>~25000</td>
<td></td>
</tr>
<tr>
<td>6 Windows Login</td>
<td>~25000</td>
<td></td>
</tr>
</tbody>
</table>
Firmware Activation

OS Constructs for Runtime Update
- Unix/Linux – kexec
- Windows – Memory Preserving Maintenance

Firmware Activation Mechanisms
- Pause/Preserve Services (VM, Containers etc.) and OS
- Invoke Modified Reset Flow
- Update and Activate New FW modules
- Activate new FW modules
- Resume OS and Services

System reboot affects the service availability
Reduce service interruption time

Intel working with partners in OCP to improve FW Update
Web-scale Configuration

- Need consistent HW Management API model across In band interface and Out of band interface
- Extend Redfish Model for Firmware Configuration Interface, including boot, power, performance, update etc.
- DMTF Redfish Host Interface between Host CPU and Out of Band Management Controller

EDKII Redfish Host Interface support available in GitHub: https://github.com/tianocore/edk2-staging/tree/UEFI_Redfish
Call To Action

• Get involved into Open System Firmware Project:
  OCP-OSF: https://www.opencompute.org/projects/open-system-firmware

• Engage with Intel on MinPlatform and FSP:
  MinPlatform: https://github.com/tianocore/edk2-platforms

• Accelerate innovations through industry collaboration (OCP, UEFI, Redfish etc.)
Thank you