Open. Together.
OCP's Rack Manager Controller subproject (OpenRMC)

John Leung, Principle Engineer
Intel Corporation

Alfie Lew, Sr. Solutions Architect
Inspur
Motivation for OpenRMC

1. System Firmware (BIOS)
   • OCP System Firmware project

2. BMC Firmware
   • OpenBMC governed by Linux Foundation

3. Rack Manager Software/Firmware
   • OCP OpenRMC

   • With OpenBMC, the industry unified the various repositories in 2018
   • With rack manager, OCP will provide a source repository and prevent splintering
The rack manager controller can be hosted in various locations:

- **OpenRACK** (within power shelf)
- **EIA, OpenRACK** (within switch)
- **Olympus** (standalone)
The OpenRMC Charter

Specify the Rack Manager Controller service architecture
- Northbound interface to datacenter manager (spec)
- Southbound interface to OCP platforms in the rack (requirements)

Deliver a Rack Manager implementation
- Available as open source

Outside of charter
- The hardware designs will be within the charter of the other OCP platform projects
Logistics

- OpenRMC is a subproject of the Hardware Management project
  - Interim co-chairs: John Leung (Intel) and Alfie Lew (Inspur)
- Wiki
- Participation
  - ARM, Microsoft, Facebook, Huawei, Inspur, Nokia, Intel, etc
Status

Regular meetings since Nov 2018
• Facebook, Inspur and Microsoft have presented their RMC architectures
• Comparison of interfaces
• https://drive.google.com/file/d/1AU8NCCL-kYstK2iChD88hxbSAxp5jCLK

Initial draft of Northbound API (Redfish)
• https://drive.google.com/file/d/1CyGBKLSAtdIuUbwtF5FR71qc77e8-KRD

Microsoft submitted their Olympus RMC source
• The submittal will be updated with latest source changes
• https://github.com/opencomputeproject/Rack-Manager
Northbound API Specification

Redfish-based Interfaces are specified by

• A set of URI to the resources
• Contents of JSON document (i.e. resource properties)
• Behavior of the interaction via the API

The Northbound API shall be specified with

• An OCP Profile (resources and resource properties)
• An interface behavior specification
Redfish Resources from Service Root

Top level resources

- Each top level resource may have subordinate resources
- Each resource is represented as a JSON document (name-value pairs)

Resource Path (URI) | OpenRMC | RSD RMM | Olympus
---|---|---|---
/redfish/v1 (Service Root) | x | x | x
/redfish/v1/Chassis | x | x | x
/redfish/v1/Managers | x | x | x
/redfish/v1/Systems | ? | x | x
/redfish/v1/AccountService | | x | x
/redfish/v1/EventService | x | x | x
/redfish/v1/TaskService | x | x | x
/redfish/v1/TelemetryService | ? | x | x
/redfish/v1/UpdateService | | x | x
Rack Manager Scope Vectors

• Manages a single rack or multiple racks?
• What rack-level manageability capabilities are supported?
  • Reboot, power usage, power limit, temperature profile, firmware update?
• How do RMC Clients and RMC Services interact?
  • During primary operational manageability or during debug?
  • Via RMC abstraction or directly accessing server?
What rack-level manageability capabilities are supported?

**Rack Level (northbound)**
- Inventory
- Power-on/Reboot/Shutdown
- Power usage
- Power limit
- Temperature (profile?)
- Update firmware on all servers
- Status/Health

**Node Level (southbound)**
- Inventory
- Power-on/Reboot/Shutdown
- Power usage
- Power limit
- Temperature (profile?)
- Update firmware on all devices
- Status/Health

The southbound Redfish interface should be aligned with the OCP platform profiles.

---

Open. Together.
How do RMC Clients and RMC Services interact?

Two extrema exists for the interaction models

1. The RMC Client manages the server, via the RMC abstraction model
   - Does not see the underlying complexity

2. Upon failure, the RMC Client accesses the server directly

1. The RMC Client manages the server, directly

2. Upon failure, the RMC Client accesses the RMC Service, to reboot or diagnose the server
Call to Action

Participate in the OpenRMC project

• Attend the OpenRMC meetings - Wiki¹
• Join the discussion - mail-list²
• Help priorities the rack-level manageability needs
• Help develop and test the OpenRMC implementation

¹https://www.opencompute.org/wiki/Hardware_Management/Open_RMC
²https://ocp-all.groups.io/g/OpenRMC