OpenRMC Charter Revision Proposal: Evolving the Rack Management Architecture

Date: March 2026

Project: OCP Hardware Management Project – Open RMC Subproject

Overview

The OpenRMC subproject aims to evolve the rack management architecture to enable flexibility, disaggregation, and advanced management capabilities. This revised charter proposal updates the legacy hardware management charter to reflect the shift from rigid firmware to flexible software implementations, addressing the needs of modern, high-density, and disaggregated data center environments.

In-Scope Activities

1. Flexible Rack Management Implementation

- **Transition from Rigid Firmware:** Moving beyond traditional, dedicated Rack Management Firmware (RMFW) tied to specialized controllers.
- **Implementation Choices:** Defining and supporting multiple implementation models for rack management functionality:
  - Dedicated Firmware (Traditional RMC)
  - Software running within an Operating System (Application on host)
  - Software deployed in Containers (Portable, efficient)
- **Migration Capabilities:** Enabling the seamless migration of rack management functionality across different hosts and environments to support load balancing, maintenance, upgrades, and high availability (failover).

2. Scaling Domains and Hierarchical Structure

- **Rack as the Fundamental Unit:** Maintaining the single rack as the foundational element of management.
- **Multi-Rack Configurations:** Addressing immediate scaling requirements for multi-rack deployments.
- **Disaggregated Architecture:** Developing a long-term vision for fully disaggregated architectures connected over high-speed optical fabrics.
- **Domain Focus Areas:**
  - *Short-Term:* Managing power and thermal domains.
  - *Mid-Term:* Managing scale-up domains for high-density architectures.

    - *Long-Term:* Managing scale-up domains over disaggregated architectures via optics.
- **POD Layer Management:** Defining the critical grouping layer above the rack (POD) and standardizing its management representation.

3. Controller Hardware Candidates

- **Hardware Flexibility:** Defining management intelligence at the chip level to optimize for both flexibility and performance.
- **Supported Platforms:**
    - *Baseboard Management Controller (BMC):* Continuing support for traditional, low-level system monitoring and control.
    - *Computer-on-Module (COMe):* Standardizing powerful options for complex management and control plane functions.
    - *Other Intelligence Engines:* Exploring bespoke solutions (e.g., ASICs, FPGAs, advanced host processors) for optimized, real-time control in disaggregated environments.

4. Critical Operational Policies

- **Robust Policy Definition:** Establishing standardized policies to ensure reliability, security, and maintainability across the rack.
- **Key policies encompass, but are not limited to:**
    - Leak Policy
    - Firmware Update Policy
    - Power On/Off Policy
    - RAS (Reliability, Availability, Serviceability) Policy
    - Power Cap Policy
- **Automated Responses:** Defining automated actions, sequence management, state transitions, recovery actions, and thermal management responses based on these policies.

5. Predictive Failure Analysis

- **Proactive Management:** Developing requirements for predicting and mitigating hardware and software failures before they cause system outages.
- **Subsystem Focus Areas:**
    - *Fabric Management:* Telemetry for link stability, error rates, and latency; models for link degradation and automated rerouting to maintain high-speed interconnectivity.
    - *GPU:* Monitoring temperature, utilization, ECC errors, and voltage; establishing prediction thresholds for workload migration or retirement to ensure consistent compute performance.

    - *Memory Subsystem:* Analysis of multi-bit ECC logs; algorithms for DIMM failure prediction and memory isolation/hot-swap requirements to prevent data corruption.
    - *Power/Thermal:* Continuous monitoring of PSU health, fan speeds, and temperatures; predictive algorithms for fan/PSU derating and stepped thermal responses to maximize component lifespan.

6. API Considerations: Stability and Performance

- **Redfish Northbound Interface:** Maintaining the current Redfish specification for standard functionality to ensure backward compatibility and stability for existing integrations.
- **Telemetry Data Path Enhancement:** Exploring the inclusion of a "fast lane" or dedicated mechanism for high-volume, real-time operational telemetry.
- **Optimization Goal:** Reducing the latency and overhead introduced by standard Redfish to enable optimized transmission of critical metrics for monitoring and analytics.

7. Hardware Rack Management System-Level Design

- **Reference Architecture:** Defining a standardized, modular reference architecture for the OpenRMC that supports disaggregated and containerized deployment models.
- **Interoperability:** Specifying clear interfaces and protocols for component-to-component communication within the rack management system, ensuring compatibility across diverse vendor hardware.
- **Scalability Testing:** Establishing performance and stress testing requirements to validate the system's ability to manage hundreds of devices within a single rack and across multi-rack PODs without performance degradation.
- **Data Model Standardization:** Defining a unified data model for all hardware assets and their telemetry, simplifying integration with upper-level data center orchestration systems.

8. AI Agent Security Requirements

- **Secure Deployment and Integrity:** Specifying mechanisms for the secure provisioning and verification of AI agents, ensuring they run only trusted, signed code and configurations.
- **Isolated Execution Environment:** Requiring execution environments that isolate AI agents from critical hardware control paths and from each other (e.g., hardware-enforced sandboxing or secure containers).
- **Least Privilege Access:** Defining granular access control policies (Role-Based Access Control - RBAC) for AI agents, limiting their permissions strictly to the hardware and data necessary for their specific function (e.g., a Predictive Failure Analysis agent only needs read access to telemetry, not write access to firmware).
- **Data Privacy and Anonymization:** Implementing requirements for anonymizing or scrubbing sensitive operational data before it is used for training or inference by AI agents, particularly when data is transferred off-rack for centralized analysis.
- **Threat Detection and Response:** Integrating monitoring capabilities to detect anomalous behavior by AI agents (e.g., unauthorized resource utilization, attempts to

access restricted interfaces) and defining automated shutdown or quarantine procedures in response to security threats.

9. Advanced Intelligent Management and Optimization

- **Self-Healing Capabilities:** Developing requirements for automated diagnosis and correction of non-critical faults without human intervention (e.g., automatic component reset, power cycling non-responsive devices, adjusting resource allocation based on real-time loads).
- **Proactive Performance and Efficiency Optimization:** Utilizing AI/ML models to dynamically adjust power, cooling, and compute resource allocation across the rack based on **job/workload awareness, hardware status, component degrading trends, and real-time fabric traffic**. This comprehensive approach aims to optimize performance at scale and improve effective efficient training/working time.
- **Root Cause Analysis (RCA) Automation:** Implementing frameworks to automatically correlate telemetry and logs across multiple subsystems (power, fabric, compute, memory) and **job execution data** to pinpoint the precise root cause of complex, multi-component failures, drastically reducing troubleshooting time.
- **Management Plane Resiliency:** Defining standards for highly available and fault-tolerant management plane services, including active-active redundancy, automated state synchronization, and rapid failover mechanisms for all critical management functions (e.g., policy enforcement, API services).

Out-of-Scope Activities

- Platform-specific manageability capabilities not applicable to the broader rack or POD level.
- Software or application management beyond the scope of hardware and infrastructure management.