OPEN

Compute Project

Proposed Track
For
Open Machine Management

Revision History

Date	Name	Description
12-5-2011	Grant Richard	Original draft based on the discussions before the NYC Open Compute Summit and at the Summit.

Overview

The focus of Open Compute is hardware and much of its effort concentrates on machine, cabinet and data centers design. Yet, part of scale computing is reducing the support burden by leveraging tools to maintain servers. But tools frequently do not scale or that differences in vendor cause users to create systems to unify the tools. The inability to properly maintain machines causes instability and faults. This is why it is important that OCP have an effort that if targeted at Machine Management.

Examples of the larger problem set are:

- There are no broadly adopted and consistently implemented capabilities around basic machine maintenance.
    - Vendor written maintenance tools with identical functions for the alerts/events, remote operations and firmware lifecycle are not interoperable. Vendor X tools can only be applies to their machines.
    - For standard based processes like event/alert/log data and remote management, even though the delivery may be the same (SNMP, IPMI and syslog) the message numberings, payloads or commands may be different.
- Software deployed across identical hardware with different firmware will appear to randomly fault due to lack of ability to test across all permutations and to discover toxic combinations of individual firmware payloads.
    - In the first three years of a machine introduction, there are typically many (3-10) firmware releases for standard motherboard components
    - Vendor specific firmware lifecycle solutions generally deploy firmware payload in a piecemeal fashion.
    - When vendors test their firmware as an integrated stack, PCI add-in components from different OEMs have a different revision release cadence and payload delivery system.
    - Firmware configuration is often embedded in firmware payloads.
    - No standards around deploying firmware bits or configurations.
- Over or inappropriately featured vendor remote management tools cause scale users to abstract their function
    - Scale users' needs a very few functions.

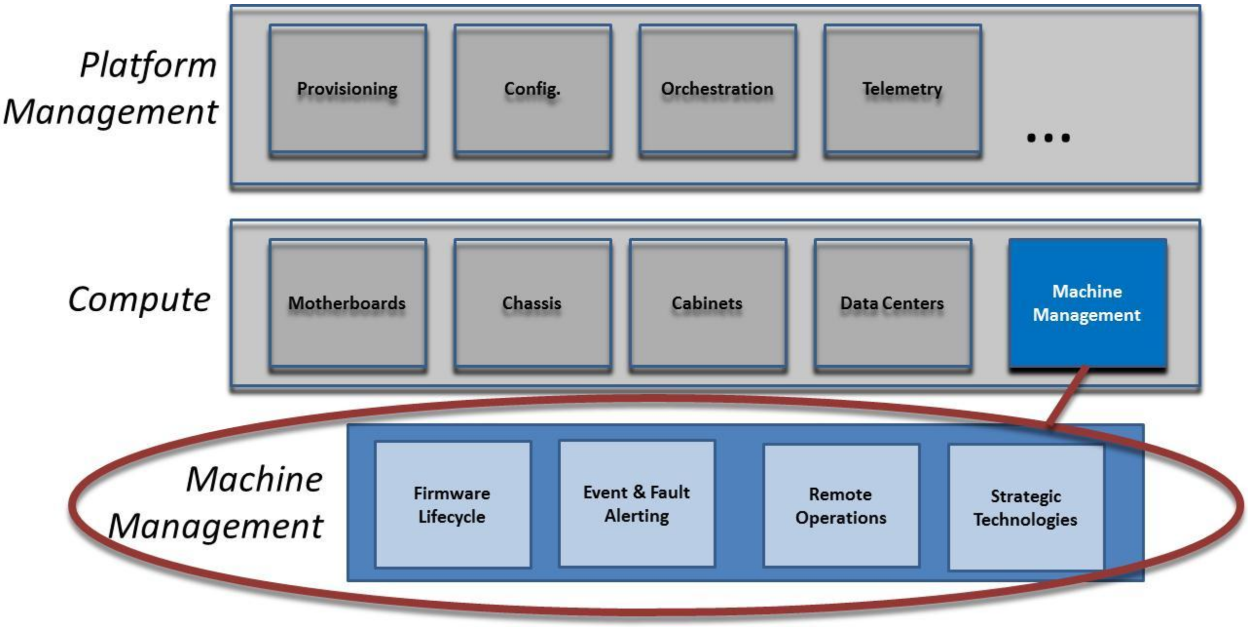
  - Complex remote management products can introduce more firmware and machine instability.

- Need to manage machines holistically as part of the overall data center management
  - No standard ways to integration of machine information into data center BMS/DCIS systems

These are some of the reasons it is important to include Open Machine Management as part of the Open Compute effort and to establish it as a Track. Standardization for API/Interface and consistent tools will only lower the barrier and ease scale computing.

Machine Management can be thought of is four areas:

1. Firmware Lifecycle
2. Event, Alert and Logs
3. Remote Operations
4. Strategic Enablement Technologies



It is equally important to understand what Open Machine Management is not. In this effort, we are making a distinction between Machine Management and Platform Management. The Machine Management effort is bounded by 1) the firmware on the machine, 2) event/alerting/logging about machine components and 3) basic remote management. Platform management encompasses all the other functions and is primarily about deploying and management operating systems and applications.

It is understood that these boundaries are somewhat arbitrary. Another, hopefully temporarily, difference is that the items identified as Platform Management have multiple cross platform open and closed sourced solutions so there are good options whereas most Machine Management tools there are not good options.

It is important that Open Compute should not forget to enable this technologies and Open Machine Management will contribute to this by providing an API/interface to access OMM functional uniformly across all OCP platforms.

Mission Statement

During our breakout session in the past Open Compute Summit, the participants created the following mission statement the frames our effort.

"Though the Open Machine Management initiative we will create uniform interfaces to manage all firmware, hardware failure event/alerting and local/remote management that are required to manage thousands of servers. Our focus will be on zero-touch maintainability, process automation and scalability by leveraging existing open standards whenever possible. We will coordinate with other groups within the OCP foundation to drive efforts."

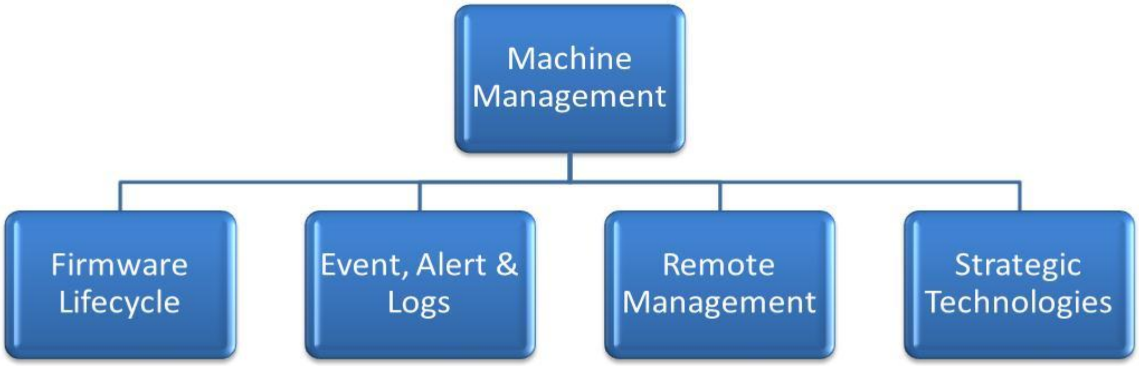
Approach

Using the Mission Statement, our approach to delivering and maintaining projects within this track are:

- Any solution must be scale to 100,000 of physical machines.
- Use existing standards whenever possible. If standard needs tweaking, work with standards bodies to make changes. Example of the standards would be:
    - DMTF – SMBIOS, CIM, WBEM, SMASH, ASF, WS-Man, SMI-S
    - Intel – (Broad adoption) DCMI, IPMI, IPMB, ICMB
    - AMD – OPMA
    - IETF – SNMP, NETCONF
    - SAForum – HPI
    - OASIS – DPWS, WSDM
    - SMIF – SMBUS – (Orig Intel), PMBUS
    - PICMG – ATCA (IPMI / IPMB / ICMP – mgt)
- Whenever possible, define interfaces/API and leave implementation to vendors/designers. Our belief is that this will increase time to market as this strategy avoids vendors having to place their code/IP into Open Source license.
- Encourage and work with Platform Management tools vendors to use OCP Hardware Management interfaces/APIs.
- Work with OCP Hardware designers to implement required functionality
- For each Sub-Project function, there will be "base" set that defines the minimal functionally to support scale computing. And there may be an "extended" set that may be developed that would serve scale compute provider with increased needs.
- As part of the every deliverable, a methodology to validate and maintain validity must be included.
- As it is common to have some non-scale platforms in a scale environment, encourage closed source and specialty hardware manufacturers to comply with Open Machine Management and to submit those platforms for validation.

Sub-Projects

Logically, the Machine Management Track divides into four Sub-Projects that integrate into common solutions for Machine Management.



Sub-Project	Description
Firmware Lifecycle	To provide a uniform interface to independently deploy and update firmware's binaries and configurations
Events, Alerts and Logs	Standard way for OCP machines to produce and format machine event, and logged messages.
Remote Management	Consistent way to remotely explore a machines configuration and perform systems operations such as reboot and open a remote console.
Strategic Technologies	Follows and encourage products and standards that may be available or commercial such as specialized system management buses and integration of machines to data center building management systems.

Sub-Project: Events, Alerts and Logs

Automation of standard machine condition is at the heart of a scale environment. For automation, Alerts, Events and Logs need to be reliable and standard.

For purposes of this document, they are defined as:

- ***Event*** is a recorded machine state change that has significance
- ***Alert*** is an urgent notification of event.
- ***Log*** contains a collection of events and also refers to the placing of events into a collection.

Events and Alerts can recorded a number of ways:

1. SNMP (Simple Network Management Protocol) (http://en.wikipedia.org/wiki/Simple_Network_Management_Protocol)
2. WS-MAN (WS-Management) (http://en.wikipedia.org/wiki/WS-MAN)

3. Syslog (http://en.wikipedia.org/wiki/Syslog)

Approach

- Define consistent event numbers and associated text payload information. Like event 501 would always be associated with a disk fault and the payload would contain the message “Disk fault – Drive %” where % is the drive identifier.
- Leverage SNMP/syslog for “base” functionality and SNMP/syslog/WS-Man for “extended” functionality.
- Whenever possible, define both a push and pull method for collecting event and alert information.
- Concentrate on the standardization of the events and not the actual implementations.
- Message number and message payloads will be consistent when delivered by different ways (like SNMP, WS-MAN & syslog).
- The approach should accommodate both in-band and out of band agents.

Sub-projects for Events, Alerts and Logs

Base Event Message Standardization

Task	Define a minimum (base) set of events, their event # and their accompanying messages
Effort	- Research and identify events/alerts - Determine payload format - Assign each event a unique number and create payload information - Draft proposal for member approval
Time	Two to four months

Machine Management SNMP MIB

Task	Create OCP SNMP MIB
Effort	- Review open source MIBs (like IF.MIB) that can be incorporated - Solicit MIB donations from existing hardware vendors - Integrate OMM base events messages in MIB - Draft proposal for member approval
Time	Three to five months

Syslog Standardization

Task	Standardize local and remote syslog and log using OCP Machine Management standard messages
Effort	- Research and document standard strategies for local and remote syslog - Integrate OMM base events into syslog - Draft proposal for member approval
Time	Three to six months

Wake-on-LAN / Wake-on-Reboot

Task	Document using existing standard an implementation of WOL and WOR.
Effort	- Research and document existing solution for WOL & WOR and performance gap analysis - Draft recommendation and proposal for member approval - Work with OCP Compute Track to implement recommendations
Time	Two to four months

Remote Machine Management

Scale environments require a way to perform operation that in non-scale environments are manually performed at the console. Basic scriptable operations should be available on every OCP machine as well as an extended set for those provides who either have increased complexity and/or greater service levels.

The examples of remote machines management operation that this Sub-Project is concerned with are:

- Remote power on / off
- Remote console
- Discover a machine's hardware/firmware configuration
- Soft reboot / shutdown
- Graphical console / VGA redirect
- Basic authentication / LDAP authentication

The approach for the Sub-Project would be:

- Delineate the remote management capabilities that fall with the Sub-Project.
- Categorize the remote management capabilities into two sets: 1) Basic that must be present in all machines and 2) Extended that may vary from platform to platform but will always have a uniform interface and performance.
- Survey existing remote management technologies and implementations to determine the best technology to leverage. Identify gaps between Remote Machine Management and existing standards. This will provide command line and API/interfaces.

Sub-projects for Remote Management

Below is a high level description of each sub-project.

Base Set – Remote Management through IPMI

Task	Standardize IPMI calls for Remote Power on/off and Remote Serial Console
Effort	- Review materials for IPMI's open source API/Interface and tools. - Draft proposal for member approval that includes recommendation, specification and certification. - Member discussion and vote - Work with OCP Compute Hardware track to include in spec
Time	Two to three months

Extended Set – Remote Management through IPMI

Task	Extend the Base Remote Management to include scale features that are not required by all scale compute users. These are features such as VGA redirect/remote graphical console, etc.
Effort	- Understand all the possible features - Need policy for compliance - Draft proposal for member approval - Work with OCP Compute track and closed source hardware vendors to implement
Time	Three to six months after Track Approval

Firmware Lifecycle

Most physical compute components have associated software most often referred to as firmware. Common examples of these include BIOS, NIC firmware and BMC firmware. Often and especially during the early part of the component's life, firmware is revised to fix bugs and introduce capabilities.

In scale environments, it is important to reduce the number and combinations firmware. Over the years, there have been toxic combinations between different version of firmware on motherboard and components (e.g. conflicts between BIOS and NIC firmware) and firmware and OSs (like BIOS version X and Linux version Y). This is why the ability to update software to keep it at a known and tested level is critical.

To further complicate the task:

- Motherboard and component manufactures often ship the most recent version of the software went delivering components so, even for the same motherboard or component, there will be multiple versions.
- The cadence between firmware revisions is different for each manufacturer.
- Each manufacturer creates their environment for delivering the firmware.
- Manufactures firmware update tools do not always run in the same OS environment – causing any solution to have multiple boots to apply updates.
- The delivering of the firmware software update includes configuration as well.

For these reasons, it is important that deploying and updating both the firmware payload and its configuration be available for each OCP compute platform.

The approach for the Sub-Project would be:

- All components with firmware are in scope. Examples include motherboards, NICs, PCI SSD, HBAs and RAID Controllers.
- Develop architecture and requirements for the firmware lifecycle configuration, deployment, updating, security and auditing firmware.
- Any solution will have the capabilities of deploying the firmware’s configuration separately.
- Product should be available as a standalone offering.
- Specify management framework/API/interfaces for providing this service to permit Platform Tool provider’s access to integrate this with their products.

Sub-projects for Firmware Lifecycle

OCP Firmware Lifecycle

Task	Through member code donation and development, create a framework for independently distributing firmware binaries and configuration.
Effort	-
Time	Nine to twelve months to initial draft vote

Strategic Enabling Technology

Given the legacy of personal computers that extended to servers, the design of machines management was “bolted” on rather than designed into the platform. There are a number of efforts to provide a richer management network and stronger integration with external systems that have the promise of increasing efficiency and easing management.

Examples of these are:

- Separate Management Networks
    - I2c-Bus
    - SMBus
- Consolidated hardware management practices between components that comprise a cabinet (servers, storage, network, MOAs,etc.) through a variety of DCIM vendors.

Sub-projects for Strategic Enabling Technologies

Dedicated System Management Buses

Task	Investigate I2x-Bus and SMBus with OCP Compute Track
Effort	- Work OCP Compute track to understand technical and cost implications for including these. - ….
Time	

Integration with data center control systems

Task	Coordinate with Data Center Track to integrate into their management systems…
Effort	- Work with Data Center Track to understand integration opportunities - Research open and closed source solutions -
Time	

Organization & Meeting Cadence

Organization

Although there are various roles within the organization (like committer, etc.), there will be three main types of participates in the project:

1. Project Chairs – who will facilitate the flow of information, determine consensus and commit documents.
2. Working Group – are track member who are committed moving the project forward between meetings. Examples of contributions can be advice, specification and code.
3. Advisory – who are people who are engaged in monthly meetings and discussions.
4. General Assembly - who are people are following the topic and want to be part of the decision process.

Meeting Cadence

The formal meetings will have the following meeting schedules:

Working Group	Will meet as needed between other Project formal meeting . A notice of any meeting /conference call will be sent to the general list for anyone interested.
General Assemblies	Will be co-terminus with the Open Compute Summits. These meeting will be for a wider audience with update on the past efforts and anticipated progress.
Advisory	Will be take place approximately every month and will discuss the progress made, open issues and anticipated progress. These calls are intended provide direction/focus of efforts and approve any new projects.

It is anticipated the Sub-Project meeting cadences will follow this pattern although Sub-Projects may decide on different cadences based on their requirements.

Projects, Groups & Resources

As we discussed in our sessions, the following is a list of project that are believed to be ….

Sub-Projects in Priority Order

Based on discussions, below are the first five sub-projects listed in priority order:

1. Base - Remote Management through IPMI
2. Base – Message Standardization
3. Machine Management SNMP MIB
4. Syslog Standardization
5. Integration with data center control systems

Groups

Co-Chairs: Matthew Liste & Grant Richard

Working Group: Matthew Liste, Grant Richard, Joel Wineland, Markus Fischer

Advisory: TBD

Resource

- Teleconferencing / bridges
- …………