Table of Content

- Motivation
- Standards
- CLI Dependency Stack
- Current Status
  - Management Fabrics and CLIs
- The Need
- The Framework
- Future Work
Motivation

What’s missing in OSCAR?

HPCC users desire a remote hardware management framework

- Standard based
- One-to-many hardware management CLI
- Auto resolve runtime environment dependency
- Easy to use
- One CLI to cross vendors/generations of HW/FW/SW

Can we bridge the gap between HPCC HW management and management components from vendors.
• Deployment phase
  – Remote power up
    • ACPI (IPMI)
    • WOL
    • APM
  – Remote deployment
    • PXE (WfM)
    • EFI network boot
  – remote text console
    • SOL (IPMI) – (BIOS/OS console redirection required)
Standards-2

- Operational phase
  - Remote power cycle hung node
    - ACPI
  - Management protocols
    - SNMP
    - CIM
    - DMI

- Standards are flexible, implementations can be different

- Each implementation shall have its own HW/FW/SW dependency
# CLI Dependency Stack

<table>
<thead>
<tr>
<th>Connections (network, direct serial, SOL, serial over telnet)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPMI Proxy and CLI implementation</td>
</tr>
<tr>
<td>OS level components and versions dependency</td>
</tr>
<tr>
<td>OS kernel version dependency</td>
</tr>
<tr>
<td>IPMI FW version dependency</td>
</tr>
<tr>
<td>BIOS versions dependency</td>
</tr>
<tr>
<td>Management controller HW / FW dependency</td>
</tr>
<tr>
<td>BMC FW versions dependency</td>
</tr>
<tr>
<td>Platform architecture (IA32, EM64T, IA64, Monolithic, Blade)</td>
</tr>
</tbody>
</table>
Current Status

• Multiple solutions
  – 3 management fabrics
  – 10+ CLIs (command sets)
  – One HW/FW/SW dependency stack per CLI
  – Same specification, different interpretation/implementation

• Management cost & effort can be improved
  – Study HW/FW/SW spec, dependency and implementation
  – Test drive all CLI commands
  – Learn and memorize when to use what command on which fabric
The Need

• A single management framework
  – One CLI to cross platforms from multiple vendors
  – Auto dependency solver
    • Self-contained, unattended local/remote installer to solve dependency issue (version management)
  – Expandability
    • Grouping
    • One-to-many CLI engine
    • Add new dependencies/CLIs
    • Output XML
• **Proof of Concept**
  – Find out the best practice for CLI integration
  – Proof unified CLI technical feasibility
  – Create an unified interface across platform/architecture/fabric/CLI

• **Results**
  – Architected and implemented a unified CLI
  – Proved the unify CLI is technical feasible
  – Intelligence is needed
    • Command level failover
    • Power up management
The Framework

• Completed a prototype
  – CLI based
  – Automated-integrated installer mechanism
  – Case based on-line help
  – Grouping features
  – One-to-many engine
  – Across HW/FW/SW/CLI
  – Command level failover
  – Smart remote power-up

• RedHat 9 and Enterprise Linux 3
Future Work

- Enhance Input/Output Parser
- Intelligent HPCC Management GUI
- Command sets plug-In mechanism
- SNMP Command for device management
- OS level HPCC Agent
Q & A