Industry Standards for Managing the HPC Cluster Life Cycle

For organizations deploying high-performance computing (HPC) clusters, reducing total cost of ownership, maximizing cluster uptime, and expanding remote manageability are among the baseline IT requirements. This article discusses how LM sensor management, Wired for Management (WfM), Intelligent Platform Management Interface (IPMI), and other industry standards can help IT professionals manage the overall HPC cluster life cycle.

BY YUNG-CHIN FANG AND RIZWAN ALI

The design of a high-performance computing (HPC) cluster environment typically takes into account several requirements, including reduced total cost of ownership (TCO), maximized cluster uptime, optimized management efforts, and expanded remote manageability. Several industry-standard management specifications are designed to address these requirements, enabling many of the basic building blocks that IT organizations need to consider as they design an underlying HPC cluster environment. To create a cluster with the desired state of manageability, architects must understand these management specifications. This article discusses how industry standards, specifications, and implementations can affect the management of an HPC cluster throughout its life cycle.

Understanding key management specifications

To interface with server hardware, cluster monitoring and management applications use mainstream management standards and specifications such as LM sensor management,[1] Wired for Management (WfM),[2] Intelligent Platform Management Interface (IPMI),[3] and Extensible Firmware Interface (EFI).[4]

LM sensors manage hardware health

The core idea of LM sensor management revolves around using a dedicated management processor to monitor and manage hardware health conditions, so that the sensor monitoring and management effort will not consume host resources. When the management processor communicates with on-board sensors through its dedicated system management bus, the management traffic does not consume system (address, data, and control) bus bandwidth. The most common implementations of LM management features include monitoring CPU temperature and cooling fan speed.

LM sensor management was widely implemented in the 1990s and has since become a de facto standard, but has left room for independent implementation of this

[1] For more information about LM sensor management, see "Industry-Standard Specifications for Cluster Management" by Yung-Chin Fang; Jenwei Hsieh, Ph.D.; Victor Mashayekhi, Ph.D.; and Reza Rooholamini, Ph.D., in *Dell PowerSolutions*, Issue 3, 2001.

[2] For more information about the Wired for Management initiative, visit http://www.intel.com/labs/manage/wfm.

[3] For more information about the Intelligent Platform Management Interface, visit http://www.intel.com/design/servers/ipmi/index.htm.

[4] For more information about the Extensible Firmware Interface, visit http://www.intel.com/technology/efi/index.htm.

standard. Many varieties of LM-compliant management sensors and processors are available; more than ten semiconductor companies produce more than 40 unique implementations of LM chips. However, not all platforms can remotely monitor and manage hardware health conditions. As a result, cross–operating system (OS) and cross-platform interoperability with LM sensors can be problematic.

WfM provides remote management

The WfM specification defines services that enable remote server management in the deployment and operation phases of the cluster life cycle (see "Following the HPC cluster life cycle" for more information). WfM incorporates the Advanced Configuration and Power Interface (ACPI),[5] remote wakeup[6] (also known as Wake on LAN, or WOL), and Preboot Execution Environment (PXE)[7] specifications—which are typically helpful in the deployment phase—as well as interfaces to several popular management standards.

ACPI. The ACPI specification allows administrators to remotely power up a new cluster that does not have a resident OS, or to power cycle a hung node. ACPI also can be used with the OS to provide OS-directed power management (OSPM) functions. In the OSPM model, the OS determines when to provide power management and the BIOS determines how to provide it. Some manufacturers implement the ACPI specification on their platforms, whereas other manufacturers choose to implement the Advanced Power Management (APM) specification[8] or a proprietary remote power management specification.

Remote wakeup. The remote wakeup specification complements the ACPI power management feature. Remote wakeup enables cluster management tools to wake up a node by sending it a remote-wakeup packet. This process requires that the cluster node support either the ACPI or the APM specification and that the node's network interface card (NIC) support the remote-wakeup feature.

PXE. Typically, after a new node is powered up remotely through implementations of remote wakeup and ACPI, an OS is deployed to that node. PXE, which is usually implemented in the NIC option ROM and in the system BIOS, allows a server to boot remotely without a local OS. When a server boots, it loads the PXE code and sends out a Dynamic Host Configuration Protocol (DHCP) request to a remote boot server asking for an IP address. Once the server receives the IP address, its PXE routine interacts with the remote boot server to dynamically retrieve the requested boot image over the network. This functionality allows administrators to remotely install the OS and applications, and to remotely configure a new cluster without the presence of a technician.

> PXE allows administrators to remotely install the OS and applications, and to remotely configure a new cluster without the presence of a technician.

One of the most popular cluster computing packages, Open Source Cluster Application Resources (OSCAR),[9] uses PXE as a building block for the main remote-system deployment mechanism. In fact, PXE is widely used by IT departments to deploy and update images for mid- to large-scale computers.

Other specifications. WfM also includes interfaces to other management standards such as Boot Integrity Services (BIS),[10] Common Information Model (CIM),[11] Desktop Management Interface (DMI),[12] Network PC System Design Guidelines,[13] Simple Network Management Protocol (SNMP),[14] System Management BIOS (SMBIOS),[15] Web-Based Enterprise Management (WBEM),[16] and Windows® Management Instrumentation (WMI).[17] These standards provide OS-level interoperability.

IPMI enables cross-platform management

Interoperability among management components is important when a cluster comprises nodes from multiple vendors. WfM is designed to facilitate hardware management within the in-band management fabric; however, the in-band management traffic can potentially degrade cluster performance for certain communication-intensive

---

[5] For more information about the Advanced Configuration and Power Interface, visit http://www.acpi.info.

[6] For more information about remote wakeup, download the Wired for Management Baseline Version 2.0 specification at ftp://download.intel.com/labs/manage/wfm/download/base20.zip.

[7] The Preboot Execution Environment Version 2.1 specification is available at ftp://download.intel.com/labs/manage/wfm/download/pxespec.pdf.

[8] For more information about the Advanced Power Management v. 1.2 interface, visit http://www.microsoft.com/whdc/archive/amp_12.mspx.

[9] For more information about OSCAR, see "Open Source Cluster Application Resources (OSCAR): Design, Implementation and Interest for the [Computer] Scientific Community" by Benoît des Ligneris, Stephen Scott, Thomas Naughton, and Neil Gorsuch. Presented at The 17th Annual International Symposium on High-Performance Computing Systems and Applications, http://hpcs2003.ccs.usherbrooke.ca/papers/desLigneris_01.pdf.

[10] For more information about Intel Boot Integrity Services, visit http://www.intel.com/labs/manage/wfm/tools/bis.

[11] For more information about Common Information Model standards, visit http://www.dmtf.org/standards/cim.

[12] For more information about Desktop Management Interface standards, visit http://www.dmtf.org/standards/dmi.

[13] The Network PC System Design Guidelines are available at http://sunsite.rediris.es/sites/download.intel.nl/design/netpc/NETPC.PDF.

[14] For RFC 1270—SNMP Communications Services, visit http://www.faqs.org/rfcs/rfc1270.html.

[15] For more information about the System Management BIOS specification, visit http://www.dmtf.org/standards/smbios.

[16] For more information about the Web-Based Enterprise Management initiative, visit http://www.dmtf.org/standards/wbem.

[17] For more information about Windows Management Instrumentation, visit http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wmisdk/wmi/wmi_start_page.asp.

applications. To help solve that problem, a joint industry effort resulted in the development of IPMI—an out-of-band, cross-platform management specification.

IPMI defines the common commands, data structures, and message formats for all IPMI interfaces. IPMI also defines common management functions, including how the system event log (SEL) and sensor data records (SDRs) are managed and accessed; how the system interfaces work; how sensors operate; how control functions such as system power-up, power-down, and reset are initiated; and how the IPMI host-system watchdog timer operates.

IPMI has two supporting specifications: Intelligent Platform Management Bus (IPMB) and Intelligent Chassis Management Bus (ICMB). IPMB defines an internal management expansion bus that is typically used to link chassis management features with the motherboard management subsystem. ICMB defines the dedicated external management bus between IPMI-enabled platforms.

> The design of an HPC cluster environment typically takes into account several requirements, including reduced TCO, maximized cluster uptime, optimized management efforts, and expanded remote manageability.

In the IPMI architecture, the central intelligence is provided by a microcontroller called the Baseboard Management Controller (BMC). The BMC operates on standby power and can automatically poll system health status. When the BMC detects any predefined exception or threshold violation, it can react to the violation condition based on preset rules—for example, log the event, generate alerts, and perform error recovery schemes such as power cycling. IPMI works by specifying common, abstracted message-based interfaces to the BMC. This abstraction isolates software from the hardware implementation and provides further interoperability.

The BMC also manages the storage of SDRs, the SEL, and field replaceable unit (FRU) information in nonvolatile memory. The SDRs describe the number and type of monitoring and control capabilities available in a given platform. These records allow software to discover and automatically adapt to the monitoring and control features offered by each platform. The FRU includes serial numbers and part numbers used to identify different serviceable or failed entities in a system.

IPMI 1.5 builds on the proven technology from IPMI 1.0 and exposes the same capabilities through new interfaces, allowing both local and remote software to automatically configure itself for multiple systems. This feature facilitates the creation of cross-platform management software. IPMI 1.5 added many features to the previous version, including:

- Serial over LAN (SOL), which uses IPMI messages encapsulated in User Datagram Protocol (UDP) packets
- IPMI over a serial/modem connection, including Basic Mode for highest speed with automated remote consoles, Point-To-Point Protocol (PPP) Mode, and Terminal Mode for limited access by dumb terminals in legacy environments
- Platform Event Filtering (PEF) to generate selectable actions when a new event matches a configurable set of event filters
- LAN alerting, which sends SNMP traps in the Platform Event Trap (PET) format to a specified destination
- Serial/modem alerting, which includes numerical paging, Telocator Alphanumeric Protocol (TAP) paging, and PPP alerting
- Alert policies that support alerts to multiple destinations
- Serial port sharing for enabling a single serial connector to be shared between the motherboard's serial controller and the serial connection to the BMC
- Boot option to direct the system boot process
- Support for a Peripheral Component Interconnect (PCI®) management bus, which defines commands and a protocol for sending and receiving IPMI messages over the proposed PCI management bus
- Definition of user privileges and authentication access to the serial/modem and LAN interfaces

More than 165 companies[18] have adopted or promoted IPMI, which was introduced in 1998. Designers can select a management processor for implementing IPMI in their platforms, and these platforms can provide varying levels of interoperability through the management framework and through different protocols.

EFI provides standard boot environment

All 64-bit Intel® architecture (IA-64) and some 32-bit Intel architecture (IA-32) platforms implement EFI. Defined by Intel, EFI creates a layer between the host OS and the platform firmware. It is implemented as a pre-OS boot environment and can be used to load EFI-level drivers. EFI provides boot and service calls to OS loaders and operating systems, so designers can implement an OS loader with little knowledge of hardware and firmware.

Following the HPC cluster life cycle

The HPC cluster life cycle consists of four phases: design, deployment, operation, and retirement (see Figure 1). Differently scaled HPC clusters have disparate needs and considerations in all

[18] For a list of IPMI industry promoters, adopters, and contributors, visit http://developer.intel.com/design/servers/ipmi/adopterlist.htm.

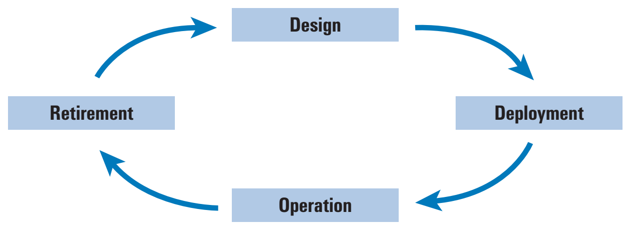

Figure 1. The HPC cluster life cycle

phases. For that reason, administrators must implement management specifications properly according to the size of their cluster installations.

Design phase: Investigating technologies and requirements

Designers must first understand the original problem that the cluster is required to solve, and then create the corresponding cluster based on that problem. Design phase tasks include understanding candidate platform architectures, understanding management specification implementation, and selecting the best-fit cluster architecture and components.

Designers can change the cluster software stack after a cluster is deployed, but they cannot change the implementation of hardware-based management specifications. Consequently, understanding industry management standards thoroughly and selecting the best-fit implementation are key tasks in this phase. Decisions made in this phase will affect all other phases.

Beyond knowing the specifications involved, designers must also understand the differences among implementations. For example, SOL can be implemented either using a shared NIC or using a dedicated NIC, and both implementations are IPMI compliant. For very large-scale clusters, the shared-NIC implementation has the potential to generate noticeable network traffic over the shared fabric. A dedicated-NIC implementation of SOL can avoid management traffic, but this implementation also increases the cost for a dedicated out-of-band management fabric.

The same results can be obtained using different specifications and implementations. For example, both EFI and IPMI implementations enable the examination of hardware health conditions without an OS. The following sections discuss the effects derived from selecting different management specifications and implementations.

Deployment phase: Constructing the cluster

The deployment phase includes racking and stacking hardware components, interconnecting all the in-band and out-of-band fabrics, enabling the cluster architecture to remotely verify the health condition of the bare-metal hardware, and deploying and configuring the cluster-computing software stack.

Certain specifications can obtain the SDRs and SEL remotely without a host OS; other specifications can enable remote firmware-level diagnostics without a host OS. When these types of specifications and an out-of-band implementation are selected, then the health of remote cluster hardware can be verified without the use of an OS. For mid- to large-scale clusters, this capability can effectively reduce cluster staging time.

During the deployment of the cluster-computing software stack, ACPI and remote wakeup can both be used to remotely power up a node without an OS. This implementation may require enabling the remote-wakeup bit on the NIC. IPMI can also be used to remotely power up a node by enabling the ACPI hardware implementation; this method does not depend on the host's NIC settings.

For the remote deployment of a cluster software stack, certain IA-32 platforms implement PXE, and other platforms provide EFI-level network boot. PXE has a dependency on the NIC option ROM or the system BIOS. The EFI-level network boot has an EFI firmware implementation dependency. In an IA-64 platform, EFI provides a network boot feature that is similar to the NIC-level PXE implementation. Both the PXE and EFI network boot implementations can help save deployment time.

The remote deployment process requires a remote console mechanism for monitoring deployment progress. IPMI defines OS-independent SOL and supports remote pre-OS serial console access over IP, providing Internet-level scalability. Another specification for remote consoles is the Intel Emergency Management Port (EMP). EMP is based on the serial-port connection and uses the standard RS-232 line interface. Pre-OS console access can reduce the difficulties of deploying the remote cluster-computing software stack.

Operation phase: Maintaining cluster operation

The operation phase includes tasks such as maintaining a cluster's operating status and optimizing resource utilization by distributing jobs properly (based on the node's health status acquired from remote management capabilities). An important goal in this phase is reducing overall operational costs by downtime reduction. Downtime can be reduced by preventing hardware failures and by implementing automatic failure recovery and unattended remote firmware provisioning.

Downtime can be reduced by preventing hardware failures and by implementing automatic failure recovery and unattended remote firmware provisioning.

In this phase, an operator can poll SDRs and the SEL from time to time to check the cluster health condition. SDRs list sensor readings and the SEL carries system events that exceed a preset threshold. When exceptional conditions such as memory bit error, fan failure, and high CPU temperature occur and are noticed by a system administrator, the administrator can either schedule a maintenance shutdown or perform a task migration to prevent system failure. Administrators can also perform periodic OS-level or EFI-level remote hardware diagnostics to examine hardware health conditions and react accordingly to prevent hardware failures.

For implementations that use clusters or cluster nodes from different generations and different vendors, interoperability that enables administrators to manage individual nodes through one centralized management console is desirable. In an ideal situation, administrators use a centralized management console to monitor and manage cluster hardware and software conditions, which is achievable using an IPMI and SNMP/CIM implementation. This implementation can save cluster management time and reduce management software costs.

In certain implementations, a node's BMC or management processor will send out a heartbeat signal periodically through the management fabric to notify the centralized management console that this node is alive. Administrators can also use the BMC to monitor the on-board watchdog for auto-recovery of a hung OS. The most common auto-recovery scheme is to automatically power cycle the hung node. Certain BMC implementations can send a PET to a centralized management console for notification if a preset threshold is met while power cycling the hung node. The centralized management console can then e-mail or page an administrator or even execute code to repair the software issue.

In the operation phase, a cluster will typically need to update its firmware for a bug fix or for new features. Updating one piece of firmware usually requires one reboot, resulting in downtime for system maintenance. To reduce system downtime, the remote firmware provisioning capability—using out-of-band fabric to update all firmware in one reboot—is a highly desired feature. Many manufacturers provide software tools for this purpose, but no standardized specification exists; so interoperability for remote firmware provisioning is not currently possible. A next-generation management specification may be able to address this need.

Retirement phase: Migrating jobs to a new cluster

The cluster retirement phase requires all runtime jobs and queued jobs to seamlessly migrate to a new cluster. By selecting the proper implementations of industry standards, administrators can help to reduce deployment time and cost, prevent node hardware failures, reduce cluster downtime, reduce operating cost, and perform smooth migrations of runtime tasks to new hardware.

Administrators can apply a runtime task migration feature in this phase to move live jobs from a retiring cluster to a new cluster. After all jobs migrate to a new cluster, a hardware management utility can be used to poll for a final hardware health status, and then ACPI can be used to shut down the retiring cluster and complete the cluster life cycle.

Achieving remote manageability and interoperability

Industry standards can help to reduce TCO during the deployment and operation phases of a cluster's life cycle. Management specifications defined by industry standards can help provide ever-increasing levels of remote manageability and interoperability, and the hardware industry is continually enhancing these specifications. However, OS-level node and centralized console management specifications are not yet standardized enough to provide full interoperability and scalability.

Management specifications defined by industry standards can help provide ever-increasing levels of remote manageability and interoperability, and the hardware industry is continually enhancing these specifications.

Microsoft has built OS-level hardware management specifications into the Windows OS. These specifications include WMI and integrated management solutions such as Microsoft® Operations Manager (MOM), Automated Deployment Services (ADS), and Systems Management Server (SMS), all of which can be effective tools for cluster management. The Linux® domain does not have a comparable, interoperable common hardware-management specification. To effectively monitor and manage a large Linux cluster, administrators may need to run several stand-alone utilities to acquire needed information. However, many such open source utilities exist.

By investing time during the design phase to better understand the various management specifications and implementations, HPC cluster designers and administrators can properly build a system with the desired level of manageability.

Yung-Chin Fang (yung-chin_fang@dell.com) is a senior consultant of the Scalable Systems Group at Dell. He specializes in cyberinfrastructure resource management and high-performance computing. He also participates in open source groups and standards organizations as a Dell representative. Yung-Chin has a B.S. in Computer Science from Tamkang University and an M.S. in Computer Science from Utah State University.

Rizwan Ali (rizwan_ali@dell.com) is a systems engineer working in the Scalable Systems Group at Dell. His current research interests are performance benchmarking and high-speed interconnects. Rizwan has a B.S. in Electrical Engineering from the University of Minnesota.