

BladeCenter systems management software

G. Pruett
A. Abbondanzio
J. Bielski
T. D. Fadale
A. E. Merkin
Z. Rafalovich
L. A. Riedle
J. W. Simpson

The management software for the IBM eServer™ BladeCenter® platform combines multiple management tools and technologies to create an integrated solution for managing, configuring, and deploying the chassis components and attached storage area networks. The IBM software also integrates BladeCenter platform management capabilities with other enterprise system management tools. This paper describes the architecture and capabilities of the IBM Director, Cluster Systems Manager, and Virtualization Engine™ tools, which enable customers to configure, manage, and provision the BladeCenter system. These tools configure and monitor the chassis using out-of-band communications with the BladeCenter management module. Management agents installed on each processor blade provide in-band operating system monitoring and integration into enterprise management applications, such as IBM Tivoli® software. This paper includes a description of the management technologies and design innovations that assist the system administrator with management tasks such as discovery, blade and switch module configuration, asset inventory, out-of-band control, alerting and event actions, storage configuration, operating system installation, application deployment, and provisioning.

Introduction

At the time the IBM eServer* BladeCenter* system entered the server marketplace, there were demands for new levels of centralization, remote management, and scalability [1]. Personal computer (PC) servers were being consolidated into centralized data centers, sometimes hosted by a service provider and located hundreds of miles from system administrators. The increasing adoption of Linux** was accelerating the growth of PC servers, with large compute-intensive applications transitioning from legacy UNIX** servers onto lower-cost, industry-standard scale-out clusters. These shifts in the server industry necessitated new remote management, deployment, and clustering technologies to help deal with the growing complexity.

With the advent of the BladeCenter server, density increased and cabling and other physical installation costs decreased. However, system administrators were still faced with challenges in configuring, deploying, and

managing a large number of pluggable components.

The BladeCenter system reduced acquisition costs by providing a shared chassis infrastructure—shared power supplies, compact disk–read-only memory (CD-ROM), floppy disk drive, and Universal Serial Bus (USB) ports for all of the processor blades in the chassis. This shared infrastructure also helped drive increasing requirements for network-based deployment and configuration tools.

The BladeCenter platform itself was designed for remote management [2], with a built-in management module and remote console support. The form factor and cabling were optimized for large, scale-out deployments of blades. To support the BladeCenter environment and the industry trends of increased physical consolidation and remote management, the systems management software had to evolve rapidly. Remote storage configuration and operating system (OS) deployment software became increasingly important, as did clustering, provisioning, and workload management tools to support the BladeCenter and other eServer platforms.

©Copyright 2005 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to *republish* any other portion of this paper must be obtained from the Editor.

0018-8646/05/\$5.00 © 2005 IBM

A key challenge for the management software was to support integrated in-band OS management and out-of-band management through the BladeCenter management module. Existing systems management products—such as IBM Tivoli* Configuration Manager [3], Computer Associates Unicenter** [4], NetIQ AppManager** [5], and Hewlett-Packard OpenView** [6]—provide management at the OS level and provide network device discovery. Hardware-specific tools, such as the IBM Hardware Management Console [7], provide sophisticated out-of-band management but have no integrated OS management. Additional challenges involved the need to manage the BladeCenter integrated network and storage components and to support increasingly larger and more dynamically changing scale-out environments.

This paper describes the architecture and the evolution of some key IBM systems management tools supporting the BladeCenter design, including IBM Director [8], Remote Deployment Manager [9], Cluster Systems Manager [10], and Tivoli Provisioning Manager [11]. It describes the development of a methodology for integrating out-of-band management and in-band management. It also discusses innovations to simplify storage configuration and bare-metal deployment, and describes the provisioning technology developed to automate the entire environment.

Life-cycle management

IBM Director provides hardware and platform management for the entire BladeCenter life cycle. Life-cycle management includes the initial setup of new hardware, ongoing management through the operational life of the hardware, and secure disposal of the hardware at its end of life.

The life cycle begins with the initial setup, starting with the discovery of new BladeCenter chassis and blades. Once these have been discovered, the management tools query the topology of the components in the chassis using the out-of-band management module interface. Next, the tools can be used to configure all of the chassis components, including the management module, the Ethernet switch modules, Fibre Channel switch modules, and other storage elements. To make the BladeCenter operational, the management tools can deploy the latest firmware updates and install operating systems and applications onto bare-metal blades.

After deployment, the management tools are used for operational maintenance and monitoring of hardware status and events. Updates may be applied to operating systems or applications via software distribution. Servers may also be reprovisioned to more efficiently allocate hardware resources to different roles.

To complete the life cycle, the management tools must also assist with hardware disposal. Disposal procedures involve securely erasing all data from any storage media to prevent unauthorized access after removal from service.

IBM Director architecture

The Director architecture [12] required significant enhancements to support the life-cycle management of the new BladeCenter paradigm. Extending some of the key functional areas of the Director architecture enabled a single user interface that consolidated BladeCenter discovery, topology, hardware configuration, storage configuration, deployment, health monitoring, and problem determination. This architecture provides a common platform for managing servers with different platform architectures, including Intel x86 HS20 and HS40 blades and IBM Power Architecture* JS20 blades.

IBM Director consists of three components that are designed to be extensible:

- *IBM Director Server* is the main daemon—or aggregation point—for discovering and managing resources, communicating with the console and endpoint agents, and authenticating and regulating access privileges. It is also the container for all task-based services and the engine for receiving events and dispatching automated actions.
- *Director Agent* runs in the OS of the endpoint server platforms being managed. Director Server can discover and manage the operating systems in which Director Agent is installed. The Agent gathers information and listens for events using standard protocols and interfaces.
- *Director Console* provides a graphical user interface with a consistent look and feel for all server platforms and devices maintained by the management server.

The BladeCenter Director release introduced a significant shift to providing more out-of-band management and control, allowing Director to use the BladeCenter management module to discover, inventory, and monitor the hardware without requiring agents to be installed in the operating systems running on the blades. Several key architectural enhancements were made to Director Server to support the new BladeCenter and out-of-band management paradigm. Each of the main server subsystems and the associated enhancements are described below.

The *topology engine* is the core component of Director Server that provides the infrastructure for representing managed resources or objects such as systems and devices, associations between these resources, and tasks

that can target these resources. Typically, each managed object has an associated managed object factory for discovering, creating, and restoring managed objects for a particular managed resource type. By extending the base classes and interfaces of the topology engine, additional managed objects were added to Director.

A base *chassis-managed object* class was created to represent and allow for the extensibility of BladeCenter and other types of chassis enclosures. This base class and its corresponding factory contain a general chassis topology model [13]. The chassis-managed object was subclassed to create the *BladeCenter chassis-managed object* representing specific BladeCenter devices and functions. A new *platform-managed object* (PMO) class was created to represent physical and logical containers that can host operating systems. The processor blade is represented by a *physical-platform-managed object*. Interaction with physical platform objects is performed using out-of-band communication or bare-metal deployment operations. *Native-managed objects* (NMOs) represent OS instances where a Director Agent is running. Simple Network Management Protocol (SNMP) *device-managed objects* represent devices running an SNMP agent. These objects are used to represent switch modules in the BladeCenter chassis.

By creating new *association classes*, relationships between managed objects can be rendered on Director Console. The association between both processor blades and switch modules and the chassis is an important one. Hence, a new association was created to view the BladeCenter system and its chassis members.

Director *task services* provide extensibility for components located on Director Agent, Director Server, and/or Director Console. A *server task extension* defines a new module that runs on the management server. Some tasks are interactive, requiring user input, while others are noninteractive and can simply be executed or scheduled. Also, some tasks support only single targets, while others may support multiple targets or may not be targeted. New tasks were created by extending the base task classes and associating them with managed objects representing the BladeCenter system or its components. New tasks include the BladeCenter assistant, switch module launcher, deployment wizard, and Remote Deployment Manager (RDM).

The Director *inventory service* consists of an extensible set of data collection classes, each of which is capable of defining inventory tables for the inventory database. For BladeCenter support, the data collection classes were extended to populate and maintain inventory tables with data from the chassis and its subsystems. Most of the data collected relates to identification of the various devices in the chassis. This data is also referred to as the vital product data (VPD) of the chassis components. Using the

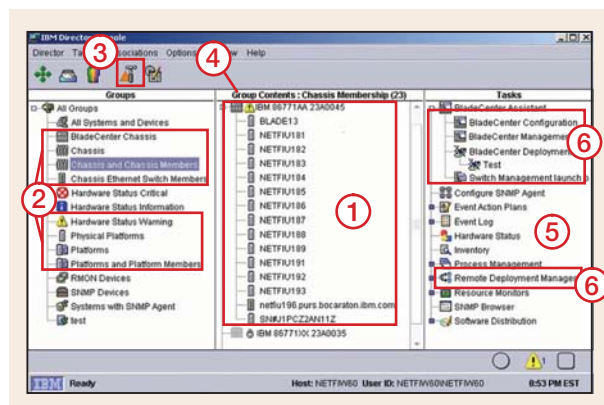


Figure 1

IBM Director Console screen. (See text for legend.)

data collected into the inventory database, *dynamic groups* can be created to filter and organize objects managed by the BladeCenter system.

The *event-management service* provides an extensible framework for the routing and processing of events generated by various Director subsystems, including the topology engine. The BladeCenter extension publishes and listens for events generated from the BladeCenter chassis. These events are mapped to Director native events and associated with the managed object, which represents the source of the event. The events are then passed to the event subsystem for subsequent processing. After being delivered to the event subsystem, events are compared with a set of configured event filters that correspond to an event action plan. If a match is found, a set of one or more actions is taken.

The Director Console screenshot in **Figure 1** depicts how the architecture was extended to integrate the systems management of the BladeCenter chassis and its components from a single point of control.

1. After discovering and authenticating with the BladeCenter management module, managed objects are created representing the topology view of the chassis and its pluggable components.
2. Each of the managed objects or BladeCenter components can be organized into dynamic groups using data from the inventory database.
3. The event action plan builder tool allows the user to create event action plans that can be used to define custom actions to respond to BladeCenter events.
4. The BladeCenter hardware status provider monitors the status of the system and its components to display colored status icons on the managed objects.

An aggregate status and the number of affected objects is shown in the lower right-hand corner of the console.

5. Director tasks appear in the right-hand pane. Basic tasks such as inventory and event log viewer apply to all managed object types.
6. New system-specific tasks were added to configure and manage the chassis. Additional tasks can be added to Director by installing additional components such as RDM.

Chassis discovery and topology

Life-cycle management begins with the discovery and inventory of the BladeCenter chassis and all of its components. This involves the ability to either manually create the chassis-managed object or automatically discover chassis-managed objects by broadcasting discovery packets over the out-of-band management network. The industry-standard Service Location Protocol (SLP) [14] provides the out-of-band protocol for chassis discovery. Based on the BladeCenter system design and the security model, the management module can be located by creating an SLP broadcast packet containing the SLP service type and a set of attributes to match [15]. The SLP response from the management module provides the management server with the Internet Protocol (IP) address and supported protocols available to connect out-of-band to the BladeCenter chassis. Additional inventory information for subsystems within the BladeCenter chassis requires authenticated communication with the management module.

SLP eliminates the requirement for a user to know the name of a network host supporting a service. Rather, the user supplies the desired type of service and a set of attributes that describe the service. On the basis of that description, the SLP resolves the network address of the service for the user. The SLP service format for management modules is as follows:

SLP URL: service:<service-type>://<addrspec>,

where <service-type> is management-hardware.IBM:management-module.

The attribute list included in the SLP response contains parameters that uniquely identify the chassis and management module, including both visual and programmatic identification parameters, such as serial number and universal unique identifier (UUID). It also contains parameters that list the supported protocols, such as SNMP and Secure Shell.

To accommodate BladeCenter discovery requirements, the Director topology engine added an SLP framework extension to enable automatic discovery of networked devices supporting SLP. Communication with a

chassis and the subsystems within it was handled by implementing a communication manager that is a Director Server task extension which runs on the management server. It provides the abstraction for authenticating and communicating with the BladeCenter management module. The BladeCenter chassis-managed-object factory—in conjunction with the SLP framework and the communication manager—discovers the chassis. An attempt is then made to gain access using the well-known default credentials of the management module. If that does not work, a padlock icon is placed beside the icon representing the chassis-managed object. The padlock icon is used in Director to indicate the need for user authentication before further management functions can be performed. When performing manual discovery, the user is required to provide proper authentication.

Once a user successfully authenticates Director Server with the management module, an out-of-band communication channel is established to discover all of the devices within the chassis. An inventory collection is initiated using this channel from Director Server to the BladeCenter management module. Physical-platform-managed objects and SNMP-device-managed objects are created to represent all of the components in the chassis topology, and the chassis membership association is populated showing the relationship of the blades and switch modules to the chassis-managed object. Information collected about the chassis components is also made available in the inventory database tables.

As part of the initial discovery process, the server task extension configures the event-forwarding tables on the management module to send events to the IP address of Director Server. The BladeCenter extension monitors alerts from the management module firmware to determine changes in the BladeCenter chassis hardware and configuration. These topology events include changes to the network configuration of the management module and the insertion or removal of processor blades and switch modules. When a processor blade is added to the BladeCenter chassis, a new corresponding platform-managed object is created and associated with the BladeCenter chassis.

The BladeCenter chassis-managed-object factory provides the implementation to maintain the hot-pluggable hardware associations between chassis subsystems and the dynamic, multilevel, scalable chassis topology. The Director topology engine was extended to provide generic-platform-managed objects to represent a server chassis that can host an OS. The BladeCenter extension creates physical-platform-managed objects to represent processor blades. Each of the processor blades represented in Director is a manageable entity with inventory, hardware status, events, and power control tasks. Other chassis components may be represented

using the SNMP-managed-object devices. For example, the BladeCenter extension creates an SNMP-managed object to represent a switch module that can further be categorized as an Ethernet switch module or Fibre Channel switch module. By using the hardware insertion or removal events, changes to chassis topology can be updated dynamically. Interfaces have been defined to allow for the topology to be traversed through the built-in associations and the acquisition of aggregate information on a particular chassis component.

Hardware status and event management

Director provides single-source monitoring of IBM hardware components through the hardware status task. This task integrates synchronous polling and asynchronous event notification to provide real-time health status across all managed systems. For convenience, status is displayed against managed systems using color-coded icons. This provides users with the ability to detect troubled systems with a glance at the management console. The hardware status task is multisourced and is implemented with a pluggable provider framework—i.e., the hardware status subsystem is extensible; additional providers can be written and added to Director to gather more events from additional sources.

The BladeCenter modular design creates a situation in which multiple computer systems become dependent on a set of shared hardware resources. Hardware failures no longer affect just one system; since they can potentially affect all of the blades in the chassis, hardware monitoring takes on greater importance. The hardware status provider uses synchronous polling to provide initial status and asynchronous events received from the management module to keep the status updated.

The BladeCenter extension that configures and monitors management module events uses a well-known User Datagram Protocol (UDP) port for events sent from the management module. When an event is received, it is mapped into Director using the process described above. The BladeCenter hardware status provider interprets the event and updates the status for the managed objects representing the system and its components.

Chassis configuration

IBM Director provides dedicated tasks that focus on a unique aspect of managing the BladeCenter system and components. The configuration task provides comprehensive configuration of the management module, processor blade service processors, and chassis policy. The management task is focused on active management of the enclosure, including monitoring event logs, viewing VPD, power control, boot sequence manipulation, and real-time sensor monitoring. The configuration and

management tasks are designed to interact with the chassis-managed object and the associated physical-platform-managed objects. The configuration changes and management actions initiated in these tasks take effect immediately. These tasks are also designed to manage multiple chassis simultaneously.

The deployment wizard provides a step-by-step task for configuring the management module, switch modules, and all of the blades in the chassis, and its detect-and-deploy capability automatically performs the configuration when a new component is added or discovered. Configuration functions include setting the IP addresses, user accounts, and management services for the management modules and switches, and initially deploying the OS onto the processor blades via the RDM task. These chassis configurations can be generated using the wizard interface and can also be saved as XML configuration files. These saved XML configuration files are incorporated into noninteractive tasks that can be launched later against a newly discovered BladeCenter chassis.

The switch management launch pad task provides a consistent interface and a convenient drag-and-drop mechanism to initiate management sessions to any switch module in the chassis. The management sessions include all applicable Web or Telnet sessions and any vendor-specific applications outside Director.

In addition to the above system-specific tasks, other Director tasks are supported on the chassis and its subsystems. These include hardware status (described earlier), power control, the location LED, and boot configuration. The power control framework enables other tasks within Director to power on, power off, shut down, and reset the blade servers using either in-band or out-of-band communications as part of their task flow. The location LED task provides a mechanism to turn on and off or flash a location LED on the BladeCenter chassis or blades. The boot configuration task provides a mechanism to reorder basic input/output system (BIOS) boot sequences, enabling tasks such as RDM to force a server to network boot. Each of these tasks extends the Director framework by providing a server extension to perform the requested actions against the system.

Remote deployment

After discovering and configuring the chassis, the next step in the life cycle is configuring and deploying the blades. Deployment is divided into two phases: the bare-metal deployment of firmware updates and operating systems onto the hardware and the post-installation maintenance and updates for the OS and applications. Post-installation maintenance is typically performed using Director software distribution or other industry-standard configuration and patch management tools. The

Director software distribution task includes a special wizard called *update assistant* that allows a user to import all driver and firmware updates from the IBM UpdateXpress CD into Director for mass distribution. Remote firmware updates for all of the components can also be performed using the update utility included with UpdateXpress.

RDM is an IBM Director Server extension that facilitates hardware configuration, firmware updates, and installation of bare-metal systems. RDM helps automate deployment tasks of x86-based platforms. RDM tasks include bare-metal OS installation (Microsoft Windows[®], Linux, VMware ESX Server[®] [16, 17]), application installation, firmware updates, BIOS setup, redundant array of independent disks (RAID) configuration, remote storage configuration, and secure data disposal for scrubbing data from retired systems. RDM also provides user-configurable custom tasks for special operations such as flashing adapter firmware.

RDM can deploy both blade and nonblade servers. BladeCenter-specific architectural features added to RDM included deploying blades discovered out-of-band, automatic object renaming based on blade slot, and power control of blades via the management module.

RDM works on bare-metal machines by Preboot eXecution Environment (PXE) [18] network booting a Disk Operating System (DOS) or Linux environment on the endpoint server to perform the required task. Multistep tasks can be created to automate the complete deployment process. For example, an RDM script task can be created to first update the firmware on the processor blades, configure zoning on the Fibre Channel switch module, create bootable logical unit numbers (LUNs) on the remote storage controller, then install an OS and applications onto that storage.

The first RDM task that must be run on all systems is the scan task. When an endpoint server PXE-boots for the first time, RDM executes the scan task on it. The scan task boots a small kernel that gathers inventory data from the endpoint and returns it to the RDM server. The scan task creates a PMO representing the endpoint if the object does not exist and updates the database with the inventory data. For the BladeCenter system, PMOs representing the bare-metal blades may already have been created by the out-of-band discovery process in Director. The RDM scan task can then be executed against these PMOs to collect additional inventory data, such as storage and network interface card data, required for OS installation tasks.

By default, newly scanned PMOs are automatically named according to their machine type model and serial number. The scan task was enhanced for the BladeCenter design to facilitate automatic renaming of the platform objects representing processor blades, using information

about the chassis and slot in which the processor blade is installed. With this feature enabled, processor blades can automatically be named according to the chassis name and slot number to allow for easier tracking of server locations.

RDM also integrated the out-of-band control capabilities in Director to provide power control for initiating the deployment operations. RDM can use either wake on LAN (local area network) or the out-of-band power control through the management module to power on a system and initiate a network boot.

Storage configuration

As mentioned above, storage configuration can be performed using RDM tasks in Director. These storage tasks automate the configuration of local RAID storage or storage attached via a storage area network (SAN), priming it for installation of a bootable OS. For the BladeCenter system, the RDM tasks can automate the configuration of certain storage controllers and Fibre Channel switch modules and perform the bare-metal OS and application installation on this storage.

Additional storage management and configuration capabilities are available through the IBM TotalStorage[®] Productivity Center (TPC). TPC is available as a separate IBM product to provide the ability to configure multiple storage devices from a single console, monitor and track the performance of SAN-attached storage, and manage advanced storage replication services such as peer-to-peer remote copy and flashcopy. For more detail on TPC, see [19].

The RDM remote storage configuration task automates the configuration of the SAN environment, enabling a diskless processor blade to boot from SAN. Manually setting up a remote boot drive for a processor blade is a time-consuming and error-prone process, requiring the user to use different tools to configure the host bus adapters (HBAs), switches, and storage servers. The remote storage configuration task within RDM automates this manual process so that customers need not understand or handle the low-level management details of the various subsystems for routine SAN administration, such as HBA configuration, LUN creation and mapping, or zoning.

When the RDM scan task is executed on an endpoint, the port World Wide Name (WWN) and fabric name of each Fibre Channel HBA port is automatically collected and entered into the Director inventory in the Fibre Channel adapter table. The user can then create an allocate-LUN task specifying the size, RAID level, storage server, and device prefix for all remote LUNs to be created. When this task is executed against a processor blade, the allocate-LUN task automatically creates (as necessary) a storage pool, LUN, host group, host, host

ports, and LUN-to-host mapping on the designated storage server. It creates zonesets, zones, and zone members on the Fibre Channel switch module if zoning is enabled. It then PXE-boots the client system and configures the QLogic Fibre Channel HBA to enable BIOS and enable boot from the new LUN.

In addition to allowing creation of allocate-LUN tasks, the remote storage configuration template provides the following built-in tasks:

- *Configure Fibre Channel HBA boot settings:* Configures the processor blade Fibre Channel HBA to boot from the first LUN 0 that is found mapped to the blade. If no LUN 0s are mapped, it will disable boot on the blade Fibre Channel HBA.
- *Delete mapping:* Unmaps remote drives from a processor blade without removing the drives themselves. This allows the drives to be remapped to another host later.
- *Delete mapping and LUNs:* Unmaps and deletes remote drives.
- *Get profile:* Captures a processor blade remote storage profile in an RDM image.
- *Put profile:* Assigns a remote storage profile image to a processor blade.

The *get profile* and *put profile* tasks support scenarios for dynamically binding blades to different bootable LUNs, including dual boot of different OS images, blade replacement and failover, and blade provisioning. For example, a processor blade may be provisioned to function as a Windows Web server during the day and a Linux payroll system at night. To accomplish this, the RDM *get profile* task can be used to capture the storage configuration for each scenario. The configurations can then be automatically applied by scheduling two *put profile* tasks: one to run each morning to assign the Windows-bootable image to a processor blade and one to run each evening to assign the Linux-bootable image to a processor blade.

For on demand provisioning, a pool of homogeneous profile images can be created. As more servers are required to meet processing demand, new blades can be provisioned with an available profile image using the *put profile* task. In this way, a provisioning manager can increase the processing capacity in just a few minutes. Similarly, by using the *delete mapping* task, blades can be taken out of commission as processing demand decreases.

Failing blades can be replaced with new blades and failing Fibre Channel adapters with new Fibre Channel adapters by simply remapping the old processor blade profile onto the new or upgraded blade, saving the time needed to allocate new LUNs and redeploy an OS and applications.

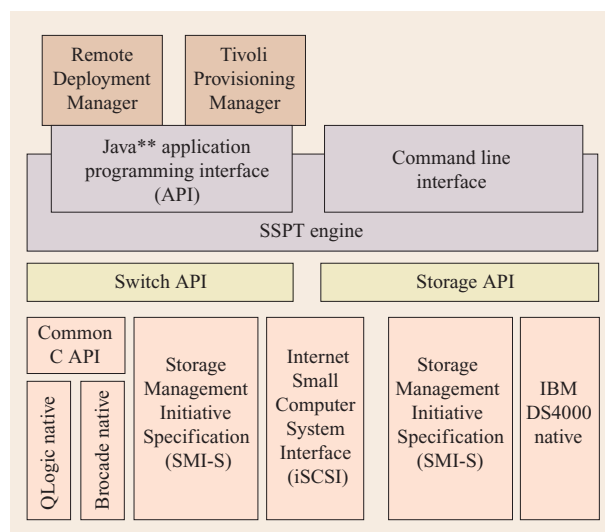


Figure 2

Architecture of the server storage provisioning tool (SSPT).

As shown in **Figure 2**, RDM employs the server storage provisioning tool (SSPT) as the remote storage and fabric interface. SSPT technology was recently developed by teams at the IBM Thomas J. Watson Research Center and Haifa Research Laboratory to automate and simplify the deployment and servicing of servers and remote storage, with a focus on the BladeCenter server. The SSPT communicates with the Brocade and QLogic Fibre Channel switches via proprietary libraries provided by the vendors. Communication with the IBM TotalStorage DS4000 series (formerly FASTT) [20] storage server is done via the system monitor command-line interface (SMcli). While currently supporting only the DS4000 line of storage servers, the SSPT is designed to be modular and pluggable so that other storage servers—Fibre Channel and Internet Small Computer System Interface (iSCSI)—can be supported in the future. Looking farther out, as the Storage Management Initiative Specification (SMI-S) becomes commonplace, a single SMI-S module will support all SMI-S-compliant IBM and non-IBM storage servers. Storage servers that are not compliant with SMI-S must be managed via proprietary interface modules.

Management agent and upward integration

While many BladeCenter management capabilities are available through the management module, some features still require an agent running in the OS on the processor blade. Director Agent provides in-band management of the OS and collects status and sends alerts from devices such as network interface cards (NICs) and local

integrated drive electronics (IDE) or SCSI hard drives. Processor temperature and voltage sensor status are also collected by the management agent, providing an alternate source for this information in the event that the out-of-band network is unavailable. Alerts from the in-band agent are issued against the native-managed-object representation of the processor blade, whereas out-of-band alerts surface on the platform-managed-object representation and on the native-managed object if one is present. For devices such as thermal and voltage sensors that are visible to both the agent and the management module, duplicate alerts from both sources are sent to Director Server. Event filtering and sorting on the management server can be used to correlate the common cause of the events.

Data and alerts from the processor blades are modeled using the Distributed Management Task Force (DMTF) Common Information Model (CIM) [21]. The CIM specification supports object-oriented modeling, allowing vendor-specific fields to be added to subclasses of standard objects when necessary without breaking interoperability with standard CIM clients. Director Agent also implements CIM association classes to describe relationships between objects in the model. These associations help describe relationships such as processor blades contained within a chassis and firmware versions attached to specific components. The CIM schema is closely tied to the version supported by the CIM server implementation shipped as part of the OS; for instance, Director Agent leverages Microsoft Windows Management Instrumentation (WMI) [22] on Windows and OpenPegasus [23] on Linux, IBM AIX*, and i5OS* as the CIM server implementations.

The well-defined schema and interoperable nature of CIM work to facilitate the exporting of blade data to third-party management servers besides IBM Director [24]. Management tools such as IBM Tivoli Enterprise Console*, Tivoli Configuration Manager, Computer Associates Unicenter, NetIQ AppManager, and HP OpenView can all collect server data and alerts from Director through a combination of agent-side data mappers and server-side scripts. Director Agent provides data mappers to translate data from the CIM model to SNMP [25] and Desktop Management Interface (DMI) [26] for integration with other management applications. For example, Director Agent includes utilities to translate and export CIM data in the DMI Management Information Format (MIF).

SNMP data mapping is performed by an SNMP subagent in Director Agent. At runtime, SNMP GET, GET NEXT, and SET requests for the Director Agent enterprise object identity (OID) are translated by the SNMP subagent into CIM client operations. CIM events are also dynamically mapped to SNMP traps

by the SNMP subagent and sent to all configured trap destinations. The Director SNMP subagent works as an extension to either the Microsoft SNMP agent or the Net-SNMP agent. Additionally, Director includes a set of management information bases (MIBs) that correspond to the instrumented CIM data and a set of property files that map CIM classes and properties to SNMP object types. For upward integration into certain management tools, server-side scripts are also required. These server-side scripts publish the data in the form of a MIB, BAROC (basic representation of object in C) event definitions, or other third-party proprietary formats for use by other management products. For products such as Tivoli Monitoring that consume CIM data natively, no mappers are needed—just a definition of CIM classes and properties to query for status.

The management module for the chassis also supports SNMP for interoperability with third-party management servers. In fact, the upward integration scripts that publish the Director Agent data also publish the BladeCenter SNMP trap types, so that Tivoli Enterprise Console and Unicenter can receive traps directly from the hardware. Various switch modules that plug into the chassis also provide their own vendor-specific MIB files that describe the SNMP data and traps available through the device.

However, there is currently no relationship between the management module MIB, the in-band agent MIB, or the MIBs of any switches that plug into the chassis. One goal for future manageability of the chassis is that all elements will be unified under a single, interrelated data model that can be intelligently traversed from one element to the next, both for IBM-specific hardware and third-party hardware. One way to provide some level of unification today is to manage all three components from Director and use one of the Director event actions to send an SNMP trap to another management server. Director Server then serves as a normalizer for incoming events from switches (SNMP traps), management module events (Director events), and agent events (CIM indications mapped to Director events). Each event is mapped into the Director Server SNMP MIB and sent to an upstream SNMP-based manager with the same enterprise OID.

Clustering

The Cluster Systems Management (CSM) tool is provided as an extension to IBM Director or as a standalone cluster management utility. CSM supports large Linux and AIX clusters containing thousands of nodes. It is optimized for executing operations in parallel across large numbers of nodes, such as OS and application deployment, out-of-band hardware control, distributed command execution, and software maintenance. Cluster management can be accomplished from a single point of

control at the management server, and the fully scriptable command-line interface offers flexibility for incorporation into a larger management environment. CSM can manage clusters containing nodes with mixed processor architectures, including Intel x86 and Power Architecture (JS20 blades) from the same management server.

On a BladeCenter system, CSM leverages the out-of-band discovery features of the management module to populate the node definition registry. The discovery agent running within the CSM management server broadcasts an SLP request and returns the SLP information for the management modules that exist on the management network. The command-line interface to the discovery agent allows the user to specify which broadcast network to search for BladeCenter servers or, by default, broadcasts on all network interfaces if none are specified. The information returned includes the IP address of the management module. This IP address may be used to collect further information about the chassis. Once the user knows the IP address of the management module, another command may be used to list the blades that exist within the chassis managed by the management module.

CSM supports network installation of the OS for the nodes. Installation of the OS is performed through either a PXE network boot of the node or, in the case of the JS20 Power Architecture blade, a directed BOOTP [27] boot request. Installation is assisted by the `getadapters` command, used to collect and update the CSM registry with the processor blade NIC attributes prior to network installation of the blade. CSM provides integrated power control through the hardware control subsystem to initiate the OS installation, and it provides connections to the serial ports on the endpoint servers using the serial-over-LAN connection to monitor the installation process.

CSM also provides a hardware control subsystem. The term *hardware control* refers to power control and other actions performed through an out-of-band hardware control point. For BladeCenter systems, the hardware control point is the management module. The CSM hardware control design is modular and allows for additional hardware support architectures to be plugged into the existing infrastructure. The user defines the architecture of the hardware within the cluster by specifying the type of power method attribute—blade, xSeries*, baseboard management controller (BMC), or hardware management console (HMC)—within the node object in the CSM registry. This power method is used by CSM to determine which library to use and which daemon to contact to access the hardware. With the BladeCenter system, a new blade power method was added to CSM. This method is used to communicate with

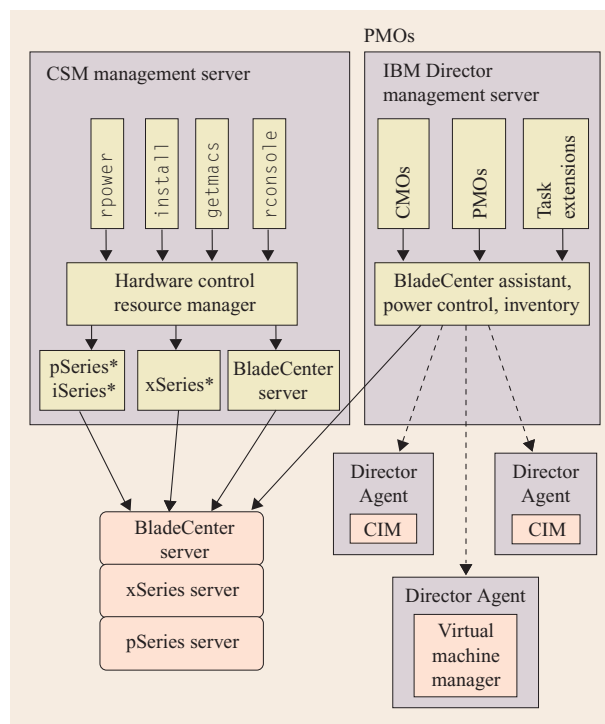


Figure 3

Architecture of the CSM management server and the hardware control subsystem.

the management module in the chassis to perform the requested hardware control task.

The hardware control architecture in CSM is optimized for parallel execution of commands across a large number of nodes in the cluster. **Figure 3** shows the architecture of the CSM management server and the hardware control subsystem. The administrator issues a CSM command to specify an operation for multiple nodes in the cluster. This operation is passed to the hardware control resource manager, which sorts the node operations according to the hardware control architecture and passes each group to the respective hardware control daemon. To provide scalability, multiple hardware control daemons exist in the architecture to offload operations from the management server. Each daemon accepts a stream of power control operations from the management server and queues the operations to be performed. The daemon then sorts operations according to hardware control point. Operations to each hardware control point are then executed sequentially on separate threads, allowing the maximum parallelism for the tasks.

Provisioning and workload management

In addition to physical management and consolidation, the BladeCenter system also focused on the problems of

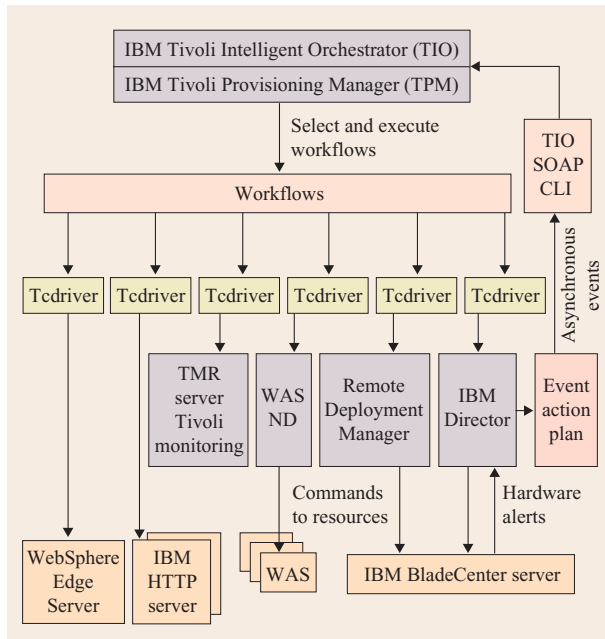


Figure 4

Example of an environment in which a Web infrastructure can be hosted within a BladeCenter chassis. (TMR: Tivoli management region; WAS: WebSphere Application Server; ND: Network Deployment; CLI: command line interface.)

consolidating workload and increasing server utilization. With the majority of PC-based servers sitting nearly idle and the increasing power requirements and demands on cooling in the data center [28], the need to further consolidate server workloads and reduce overprovisioning was rapidly growing in importance. The IBM Virtualization Engine* [29] product brought together and integrated the Director Multiplatform, Systems Provisioning, and Enterprise Workload Manager (EWLM) software components to create an on demand operating environment to control the management and allocation of server, network, and storage resources.

The Virtualization Engine software leverages Tivoli Provisioning Manager (TPM) for systems provisioning. TPM provides the object model for describing data center resources and a workflow engine [30] to drive resource assignments, application configuration [31], and back-end network [32] and storage infrastructure changes. Tivoli Intelligent Orchestrator (TIO) is built on the same TPM technology, but adds a resource broker and prediction engine to drive autonomic provisioning operations [33].

The role of provisioning and orchestration tools is to manage the background allocation of resources. Provisioning operations typically take a significant amount of time—from five minutes to an hour or longer, depending

on the workflows involved. In an on demand, orchestrated environment, objective analyzers predict demand for additional server resources and allocate and provision those resources before user demand spikes upward. Using virtualization in an orchestrated environment provides additional flexibility, allowing all services to continue to operate (perhaps in a degraded state) even when a dearth of physical server resources is present.

A separate and distinct notion from provisioning is workload management. Workload management [34, 35] is provided by products such as the Application Workload Manager (AWM) for IBM Director and EWLM [36] in the Virtualization Engine. Workload management is typically concerned with optimizing the performance and utilization of a system with a fixed set of resources. In a BladeCenter environment, workload management involves managing resource allocations within a server—affecting the share-based entitlements for processor or memory resources among virtual machines or among processes within a single OS. Active workload management software helps increase the number of simultaneous application sessions that can be active on a given server. Increasing the application density per server has a significant effect on reducing the overall total cost of ownership [37]. In the event that a workload manager cannot satisfy its service-level agreements, it can notify a provisioning manager to allocate additional virtual or physical servers [38].

The BladeCenter on demand software comprises multiple management tools working together:

- TPM or TIO for the object model and workflow engine for provisioning the server, network, and storage resources.
- EWLM or AWM for workload management.
- IBM Director for hardware management, fault monitoring, and processor blade insertion/removal detection.
- IBM RDM for bare-metal OS deployment, agent installation, hardware configuration, and firmware updates.

TPM integrates with other management tools by delegating specific operations and coordinating the results through its workflow engine. **Figure 4** shows an example environment whereby a Web infrastructure can be hosted within a BladeCenter chassis. The entire hardware and application environment can be orchestrated using TIO [39]. The TPM architecture provides logical device operations (LDOs), or abstract methods, for each object in the data center model. These LDOs are implemented by workflows that interact with specific management tools and applications. Workflows implementing specific devices or applications are collected into separately

installable automation packages (also called *tcdrivers*). For example, TPM workflows can notify network load balancers, such as IBM WebSphere* Edge Server, of the addition or removal of server resources from the cluster, and workflows can deploy Sun Java** 2 Platform Enterprise Edition (J2EE**) applications into a WebSphere Application Server (WAS) environment.

For the BladeCenter environment, TPM workflows were implemented to interface with Director for power control of blades and with RDM for bare-metal deployment of blades. Additionally, workflows were created to interact with the BladeCenter switch modules, allowing TPM to modify the virtual local area network (VLAN) configuration and the SAN configuration as part of a server reprovisioning workflow.

BladeCenter and Director software interact with TPM in three ways. First, TPM can perform operations on the hardware by invoking interfaces in Director through a workflow. The workflow contains the necessary logic to perform the actual operations (such as power control or bare-metal installation) using the Director command-line interface. Second, Director interacts with TPM by initiating TPM workflows using the Simple Object Access Protocol (SOAP) interfaces. Workflows may be initiated automatically by a Director event action plan (EAP) to run a program on the management server in response to a BladeCenter chassis event, which in turn executes a SOAP command to initiate a TPM workflow. For example, Director Server can automatically initiate a workflow in TPM to update the data center model in the event that Director detects a blade insertion or removal event in a particular BladeCenter chassis. Finally, Director provides hardware topology information (discovery of chassis, blades, slots, switch modules, ports, network interfaces) to TPM. The Director inventory utility included in TPM facilitates the creation of the data center model using the detailed topology information collected by Director inventory.

For bare-metal installation of operating systems and applications, TPM can drive the RDM server. The RDM server is represented as a boot server in the TPM data center model. An image software stack that includes a variable containing the RDM deployment task name and a reference to the RDM boot server is defined. This image software stack is then associated with a cluster in the data center model. Whenever a server is moved into a cluster with an associated image software stack, the RDM operations are invoked to install the OS on the server and update the software associations for the server object in the DCM.

Conclusion

The management software for the BladeCenter platform combines multiple management tools and technologies

to create an integrated solution for managing and configuring the chassis hardware components and attached storage area networks. It also provides a unifying architecture for integrating with enterprise management tools and managing higher-level constructs, such as clusters. The BladeCenter tools also plug in as integral components in the Virtualization Engine architecture, providing workload management and provisioning capabilities to improve server utilization and efficient allocation of resources.

Acknowledgments

The authors gratefully acknowledge the constructive comments of Dr. Tom Bradicich, Dhruv Desai, and Dr. Steven Hunter, all of the IBM Systems and Technology Group. The authors recognize the significant intellectual and development contributions to the storage configuration and server storage provisioning tool work from IBM Research team members B. Abali, M. Banikazemi, K. Barabash, S. Guthridge, E. A. Henis, H. Kolodner, D. E. Poff, T. B. Smith, G. Valentin, and N. Wei. The authors also acknowledge B. Potter, J. Ellsworth, S. Sakolish, and W. LePera from the cluster systems management team for their intellectual contributions and constructive comments.

*Trademark or registered trademark of International Business Machines Corporation.

**Trademark or registered trademark of Linus Torvalds, The Open Group, Computer Associates International, Inc., NetIQ Corporation, Hewlett-Packard Development Company, L.P., Microsoft Corporation, VMware, Inc., or Sun Microsystems, Inc. in the United States, other countries, or both.

References

1. D. M. Desai, T. M. Bradicich, D. Champion, W. G. Holland, and B. M. Kreuz, "BladeCenter System Overview," *IBM J. Res. & Dev.* **49**, No. 6, 809–821 (2005, this issue).
2. T. Brey, B. E. Bigelow, J. E. Bolan, H. Cheselka, Z. Dayar, J. M. Franke, D. E. Johnson, R. N. Kantesaria, E. J. Klodnicki, S. Kochar, S. M. Lardinois, C. A. Morrell, M. S. Rollins, R. R. Wolford, and D. R. Woodham, "BladeCenter Chassis Management," *IBM J. Res. & Dev.* **49**, No. 6, 941–961 (2005, this issue).
3. V. Gucer, O. Alti, C. Chan, M. Choiawala, T. Lamacchia, J. Raeymaeckers, N. Xaba, and Z. Veress, "All About IBM Tivoli Configuration Manager Version 4.2," *IBM Redbooks*, December 11, 2002; see <http://www.redbooks.ibm.com/abstracts/sg246612.html>.
4. Computer Associates, Unicenter Infrastructure Management Software; see <http://www.ca.com/infrastructure>.
5. NetIQ Corporation, NetIQ AppManager for IBM Director; see <http://www.netiq.com/products/am/modules/netfinity.asp>.
6. Hewlett-Packard Development Company, L.P., HP OpenView Management Software; see <http://www.openview.hp.com>.
7. K. Matsubara, M. Robbins, R. Barker, and T. Thitayanun, "Effective System Management Using the IBM Hardware Management Console for pSeries," *IBM Redbooks*, August 20, 2003; see <http://www.redbooks.ibm.com/abstracts/sg247038.html>.

8. D. Watts, E. Ciliendo, D. Feisthammel, and M. Gudmundsen, "Implementing Systems Management Solutions using IBM Director," *IBM Redbooks*, June 15, 2003; see <http://www.redbooks.ibm.com/abstracts/sg246188.html>.
9. R. Marchini, "IBM eServer BladeCenter Systems Management with IBM Director V4.1 and Remote Deployment Manager V4.1," *IBM Redbooks*, February 4, 2004; see <http://www.redbooks.ibm.com/abstracts/redp3776.html>.
10. D. Quintero, T. Braunbeck, O. S. Thye, A. Vlad, and P. Zutenis, "Cluster Systems Management Cookbook for pSeries," *IBM Redbooks*, December 1, 2004; see <http://publib-b.boulder.ibm.com/abstracts/sg246859.html>.
11. E. Manoel, S. C. Brumfield, K. Converse, M. DuMont, L. Hand, G. Lilly, M. Moeller, A. Nemati, and A. Waisanen, "Provisioning On Demand: Introducing IBM Tivoli Intelligent ThinkDynamic Orchestrator," *IBM Redbooks*, December 23, 2003; see <http://www.redbooks.ibm.com/abstracts/sg248888.html>.
12. V. Castelli, R. E. Harper, P. Heidelberger, S. W. Hunter, K. S. Trivedi, K. Vaidyanathan, and W. P. Zeggert, "Proactive Management of Software Aging," *IBM J. Res. & Dev.* **45**, No. 2, 311–332 (March 2001).
13. T. D. Fadale, "Chassis Environment Topology Mapping for System Management," IPCOM000028204D, May 3, 2004; see <http://www.ip.com>.
14. E. Guttman, C. Perkins, J. Veizades, and M. Day, "RFC: 2608—Service Location Protocol, Version 2, Network Working Group, The Internet Society, June 1999; see <http://www.rfc-editor.org/rfc/rfc2608.txt>.
15. E. Guttman, C. Perkins, and J. Kempf, "RFC 2609—Service Templates and Service: Schemes," Network Working Group, The Internet Society, June 1999; see <http://www.rfc-editor.org/rfc/rfc2609.txt>.
16. VMware, Inc., Configuring and Installing IBM BladeCenter; see http://www.vmware.com/pdf/esx21_IBM_blade.pdf.
17. B. Dufrasne, A. Groth, A. Meakin, P. Mescher, and A. Letts, "Implementing VMware ESX Server 2.1 with IBM TotalStorage FASTT," *IBM Redbooks*, September 24, 2004; see <http://www.redbooks.ibm.com/redbooks.nsf/0399260c229485daa85256ed90081878e?OpenDocument>.
18. Intel Corporation, Preboot Execution Environment (PXE) Specification, September 1999; see <http://pxes.sourceforge.net/pxe.html>.
19. M. Lovelace, I. Aliprandi, W. Andrews, J. A. Cooper, D. Demer, W. Egli, and P. Zerbin, "Managing Disk Subsystems Using IBM TotalStorage Productivity Center," *IBM Redbook*, SG24-7097-01 (September 2005); see <http://www.redbooks.ibm.com/abstracts/sg247097.html>.
20. B. Dufrasne, A. Groth, A. Meakin, P. Mescher, and A. Letts, "IBM TotalStorage FASTT Best Practices," *IBM Redbooks*, September 22, 2004; see <http://www.redbooks.ibm.com/abstracts/sg246363.html>.
21. Distributed Management Task Force, Inc., Common Information Model (CIM) Standards; see <http://www.dmtf.org/standards/cim>.
22. Microsoft Corporation, WMI: Windows Management Instrumentation; see <http://www.microsoft.com/whdc/system/pnppwr/wmi/default.msp.x>.
23. The Open Group, OpenPegasus; see <http://www.openpegasus.org>.
24. D. Watts, P. Engel, R. Marchini, and S. Pfeiffer, "Integrating IBM Director with Enterprise Management Solutions," *IBM Redbooks*, July 31, 2001; see <http://www.redbooks.ibm.com/abstracts/sg245388.html>.
25. J. Case, M. Fedor, M. Schoffstall, and J. Davin, "RFC 1157—Simple Network Management Protocol (SNMP)," Network Working Group, The Internet Society, May 1990; see <http://www.rfc-editor.org/rfc/rfc1157.txt>.
26. Distributed Management Task Force, Inc., Desktop Management Interface (DMI) Standards; see <http://www.dmtf.org/standards/dmi>.
27. B. Croft and J. Gilmore, "RFC 951—Bootstrap Protocol (BOOTP)," Network Working Group, The Internet Society, September 1985; see <http://www.rfc-editor.org/rfc/rfc951.txt>.
28. R. Schmidt, "Hot Spots in Data Centers," *Electron. Cooling Online* **9**, No. 3 (August 2003); see http://www.electronics-cooling.com/html/2003_august_a2.html.
29. Virtualization Engine, IBM eServer Software Information Center, http://publib.boulder.ibm.com/eserver/v1r1/en_US/index.htm?info/icmain.htm.
30. E. Manoel, M. Hicks, M. Moeller, I. Naick, M. Poulson, and J. Surmann, "Developing Workflows and Automation Packages for IBM Tivoli Intelligent ThinkDynamic Orchestrator," *IBM Redbooks*, May 17, 2004; see <http://www.redbooks.ibm.com/abstracts/sg246057.html>.
31. A. Abbondanzio, Y. Aridor, O. Biran, L. L. Fong, G. S. Goldszmidt, R. E. Harper, S. M. Krishnakumar, G. Pruett, and B.-A. Yassur, "Technical Forum: Management of Application Complexes in Multitier Clustered Systems," *IBM Syst. J.* **42**, No. 1, 189–195 (2003).
32. S. W. Hunter, N. C. Strole, D. W. Cosby, and D. M. Green, "BladeCenter Networking," *IBM J. Res. & Dev.* **49**, No. 6, 905–919 (2005, this issue).
33. M. Mihaescu and A. Trossman, "Increasing the Availability of Complex Systems Through Dynamic Automatic Reconfiguration," *Proceedings of the 9th International Symposium for Design and Technology of Electronic Packages (SITIME)*, September 19, 2003.
34. J. Aman, C. K. Eilert, D. Emmes, P. Yocom, and D. Dillenberger, "Adaptive Algorithms for Managing a Distributed Data Processing Workload," *IBM Syst. J.* **36**, No. 2, 242–283 (1997).
35. J. D. Turner, D. P. Spooner, J. Cao, S. A. Jarvis, D. N. Dillenberger, and G. R. Nudd, "A Transaction Definition Language for Java Application Response Measurement," *Intl. J. Computer Resource Meas.* **105**, 55–65 (December 2001); see <http://www.dcs.warwick.ac.uk/research/hpsg/documents/TurnerJ.TDLFJARM.pdf>.
36. P. Bari, C. Covill, K. Majewski, C. Perzl, M. Radford, K. Satoh, D. Tonelli, and L. Winkelbauer, "IBM Enterprise Workload Manager," *IBM Redbooks*, August 27, 2004; see <http://www.redbooks.ibm.com/abstracts/sg246350.html>.
37. J. R. Phelps, "Consolidating Servers Can Cut TCO Significantly—Sometimes," *Gartner Research* (March 2002); see http://www.gartner.com/DisplayDocument?doc_cd=105163.
38. K. Appleby, S. Fakhouri, L. Fong, G. Goldszmidt, M. Kalantar, S. Krishnakumar, D. P. Pazel, J. Pershing, and B. Rochwerger, "Oceano—SLA Based Management of a Computing Utility," *Proceedings of the 7th IEEE/IFIP International Symposium on Integrated Network Management*, May 2001, pp. 855–868.
39. R. Credle, A. Chechi, B. Fenoglio, J. Haskins, K. Kekessie, J. Kenefick, G. Pruett, B. Rajaraman, D. Ullrich, and T. Verhoeven, "IBM Web Infrastructure Orchestration," *IBM Redbooks*, November 19, 2003; see <http://www.redbooks.ibm.com/abstracts/sg247003.html>.

Received December 16, 2004; accepted for publication March 4, 2005; Internet publication October 7, 2005

Gregory Pruett *IBM Systems and Technology Group, 3039 Cornwallis Road, Research Triangle Park, North Carolina 27709 (pruett@us.ibm.com).* Mr. Pruett is a Senior Technical Staff Member in xSeries software solutions. He works as a software architect and strategist, specializing in provisioning and remote deployment solutions. He holds B.S. degrees in mathematics and computer science from Furman University and an M.S. degree in computer science from the University of North Carolina. He has eight years of experience in systems management and deployment software, working as lead developer and architect on the IBM Director and IBM Remote Deployment Manager products. Mr. Pruett has 30 issued or pending patents in areas including RF-ID, BladeCenter design, bare-metal OS installation, and on demand provisioning.

Anthony Abbondanzio *IBM Systems and Technology Group, 3039 Cornwallis Road, Research Triangle Park, North Carolina 27709 (danzio@us.ibm.com).* Mr. Abbondanzio is a Senior Engineer in the IBM Systems and Technology Group. He received a B.S. degree in computer science from the University of Central Florida in 1986. He has worked as a lead developer and architect on OS/2 and systems management software at IBM for 11 years. He is currently responsible for the design and architecture of eServer systems management, with a focus on the IBM Director product and heterogeneous platform management. Mr. Abbondanzio holds ten issued or pending patents associated with operating systems, clustering, systems management, and BladeCenter design.

Julianne Bielski *IBM Systems and Technology Group, 3039 Cornwallis Road, Research Triangle Park, North Carolina 27709 (bielsk@us.ibm.com).* Ms. Bielski is a Senior Engineer in the eServer software systems management organization. She received a B.S. degree in mechanical engineering from the University of Michigan in 1991 and an M.S. degree in biomedical engineering from the University of North Carolina in 1995. Since 2001, she has been part of the eServer organization, where most recently she has been an architect and designer for the IBM Director product. Ms. Bielski holds several patents spanning hardware and software.

Tushar D. Fadale *IBM Systems and Technology Division, 3039 Cornwallis Road, Research Triangle Park, North Carolina 27709 (fadalet1@us.ibm.com).* Dr. Fadale is an Advisory Software Engineer in eServer and Blade Systems Management development. He received a B.Tech. degree in aeronautical engineering from the Indian Institute of Technology, Kharagpur, in 1985, and M.S. and Ph.D. degrees in aeronautical and astronautical engineering from the University of Washington at Seattle in 1987 and 1993, respectively. He has authored and presented numerous journal and conference papers in the fields of stochastic analysis and inverse problems in thermal systems. Since joining IBM in 2000, he has been developing systems management software in IBM Director. Dr. Fadale has several pending patents associated with software and hardware.

Aaron E. Merkin *IBM Systems and Technology Division, 3039 Cornwallis Road, Research Triangle Park, North Carolina 27709 (amerkin@us.ibm.com).* Mr. Merkin works in eServer and Blade Systems Management development focusing on standards definition through the Distributed Management Task Force. He received a B.S. degree in mathematics from Florida Atlantic University in 1997 and an M.S. degree in computer science from North Carolina State University in 2000. Since joining IBM in 1998, he has been developing systems management software. Mr. Merkin has several pending patents related to application management and BladeCenter design.

Ziv Rafalovich *IBM Research Division, Haifa Research Laboratory, University Campus, Mount Carmel, Haifa 31905, Israel (zivr@il.ibm.com).* Mr. Rafalovich is a team leader and an architect in the Storage Management Technologies Group at the IBM Haifa Research Laboratory. His work is focused on server/storage integrated management and disaster recovery. He received a B.S. degree in computer science in 1998 and an M.B.A. degree in 2004, both from the Technion, Israel Institute of Technology. He has led the Server Storage Provisioning Task team in Haifa since he joined IBM in 2003. He has eight years of experience in the development of management solutions in various domains. Mr. Rafalovich has three issued or pending patents in the area of remote storage software.

Linda A. Riedle *IBM Systems and Technology Group, 3039 Cornwallis Road, Research Triangle Park, North Carolina 27709 (lriedle@us.ibm.com).* Ms. Riedle is a Senior Software Engineer in the xSeries Deployment Software Department. She received a B.S. degree in computer science from the University of Louisville in 1984, followed by an M.E. degree in computer engineering in 1985. Since 1996 she has been part of the xSeries organization, where most recently she has been a designer and developer for storage deployment for the Remote Deployment Manager product. Ms. Riedle has 19 issued or pending patents spanning RAID and remote storage software.

John W. Simpson *IBM Systems and Technology Group, 2455 South Road, Poughkeepsie, New York 12601 (john_simpson@us.ibm.com).* Mr. Simpson is the AIX/Linux server hardware Technical Chief Engineering Manager for the Poughkeepsie UNIX Development Laboratory, and hardware control architect for CSM. He received a B.S. degree in computer science from the State University of New York at Plattsburgh in 1982. From 2000 to 2004, Mr. Simpson was the CSM hardware control architect and team leader, responsible for designing cluster solutions for out-of-band control across the pSeries, eServer, BladeCenter, and xSeries product lines.