Cisco UCS Manager Architecture

What You Will Learn

The management infrastructure of a traditional blade server solution must deliver and control the power, cooling, network, and logic to the blades. The management process is made complex by the need for various products to achieve some level of integration to create the overall solution. Each component vendor (and often each of these vendor's products) has its own unique management interface and set of protocols. To bring order to this challenging situation, Cisco® UCS Manager v1.0.1 software manages all components of the Cisco Unified Computing System server solution through a single pane of glass.

Cisco UCS Manager serves as an embedded device manager for all Cisco Unified Computing System components. In essence, Cisco UCS Manager creates a unified management domain that serves as the central nervous system of the Cisco Unified Computing System. In one sense, Cisco UCS Manager takes the place of the system management tools associated with a traditional computing architecture by integrating computing, networking, and virtualization resources into one cohesive system. However, Cisco UCS Manager leaves cross-system and heterogeneous device management to classic system management tools and provides APIs to those tools. Cisco UCS Manager implements policy-based management using service profiles to help automate provisioning and increase agility.

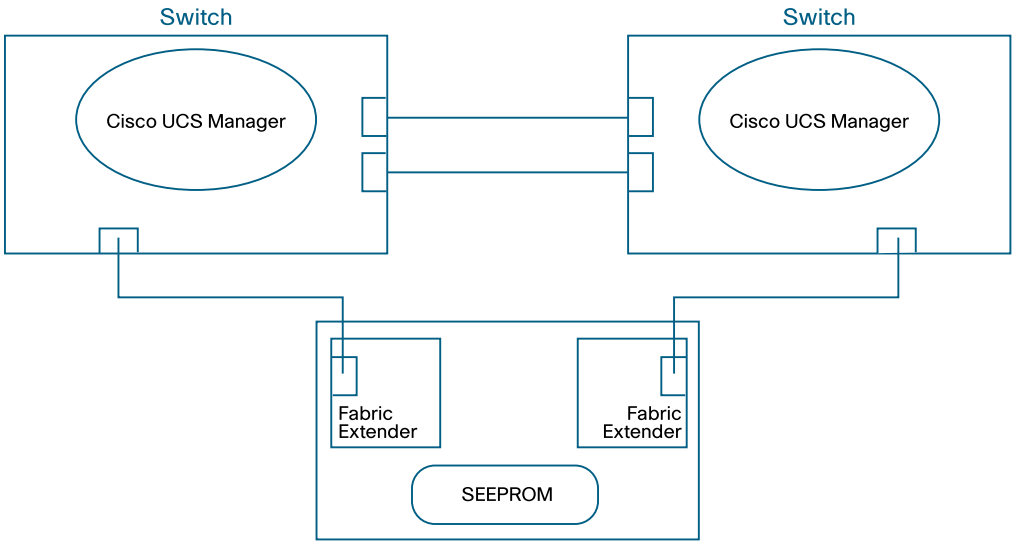
In this paper, you will learn about

- Cisco UCS Manager high availability
- Cisco UCS Manager clients
- Cisco UCS Manager's model-driven architecture

Cisco UCS Manager High Availability

The Cisco Unified Computing System supports the use of two interconnected switches for improved management redundancy and increased switch throughput. A valid high-availability configuration includes a pair of interconnected switches and two fabric extender modules per chassis (with one fabric extender attached to one switch, and the other fabric extender attached to the other switch), as shown in Figure 1.

Figure 1. High Availability Through Redundant Management



Active versus Standby

Each of the two Cisco UCS Manager instances will run as either the active or standby instance. This distinction applies to certain running processes (for example, the data management engine [DME]), but not to other processes (for example, the element managers). All Cisco UCS Manager processes running on both the active and standby instances are monitored by another process, which can restart them if they terminate abnormally.

A “floating” management IP address is configured on the active instance so that all GUI and command-line interface (CLI) connections and management operations are forced to initiate there. Configuration and operational state changes are then propagated over the private network from the active instance to the standby instance so that management information is synchronized.

Split Brain

The Cisco UCS Manager instance acting as the active instance will remain so either until it fails (failover) or until the administrator requests a role change (switchover). Each instance periodically sends a heartbeat message to the other to indicate that it is healthy. The situation in which an instance stops receiving heartbeats from the other node is referred to as a split brain. Cisco UCS Manager detects and resolves this split brain by using the SEEPROM on the chassis.

The split brain just described is also referred to as a partition in space, so named because the cluster was partitioned spatially through the loss of the private interconnect. The other type of split brain is referred to as a partition in time. It occurs when an instance attempts to start the cluster on an outdated configuration. Again, Cisco UCS Manager detects and resolves this type of split brain using the SEEPROM.

Cisco UCS Manager Clients

The two primary interfaces for users to interact with Cisco UCS Manager are the buffered CLI and the GUI. Both the CLI and GUI are written using a third interface to Cisco UCS Manager: the XML API.

The GUI provides a very simple and intuitive interface for users to manage the Cisco Unified Computing System. The Java Web Start executable code is pushed to the admin client at the time of connection to the management port, helping ensure that the most current version of the GUI is deployed. One of the benefits of using the GUI is that it provides several wizards to guide users through complex configuration tasks. For example, two types of wizard are available for configuring servers: simple and expert. The simple wizard hides configuration of certain properties while exposing only those that are absolutely necessary to provision a blade. By contrast, the expert wizard exposes all properties to the user who needs that level of configuration.

The CLI provides an interface for users who prefer working at the command line. One of the benefits of using the buffered CLI is that it provides the capability to group multiple configuration changes and store and run scripts. For example, if users perform a set of management operations on a regular basis, the associated commands can be stored in a script. Users can, whenever they want, run the script with a single command. In this way, users can effectively create their own custom configuration wizards by writing scripts. The CLI also comes equipped with standard enterprise features such as command history and syntactically guided command completion.

The XML API is a powerful interface for Cisco UCS Manager. Recall that the management information is stored in a hierarchical tree structure known as the management information tree. This tree is exposed in its entirety through the XML API, permitting users to manipulate management information in new and custom ways. Users who want to create their own front-end interfaces can do so using the XML API. For example, organizations that want to deploy the Cisco Unified Computing System in a multi-tenancy environment can create a custom portal for each tenant. Others may want to use the API to connect to an external configuration management database (CMDB) or to third-party management and provisioning tools.

Role-Based Access in UCS Manager

This flexibility in client interfaces facilitates role-based management while eliminating much manual coordination between multiple disciplines as it allows IT administrators from various disciplines to work with the interfaces and tools with which they are most familiar. Others may choose to adapt their tools to emerging needs. For example, one person may be responsible for servers, one for SAN configuration, one for LAN configuration, and one for managing the rest of the Cisco Unified Computing System infrastructure. In large-scale environments, teams of users may be responsible for managing each of these components across the entire organization, and within these teams, some individuals may even specialize in the management of subcomponents. For example, within the service profile management team, one user may be responsible only for configuring Ethernet networking for blades.

Standard roles (server, network, storage, etc) are built into the UCS Manager. Each role has a corresponding set of privileges that control write access to server configuration, internal and border LAN configuration, internal and border SAN configuration, and configuration of other Cisco Unified Computing System components, including configuration of RBAC itself.

Cisco UCS Manager further categorizes these high-level privileges into lower-level ones to support the more granular division of management responsibility found in large deployments. For example,

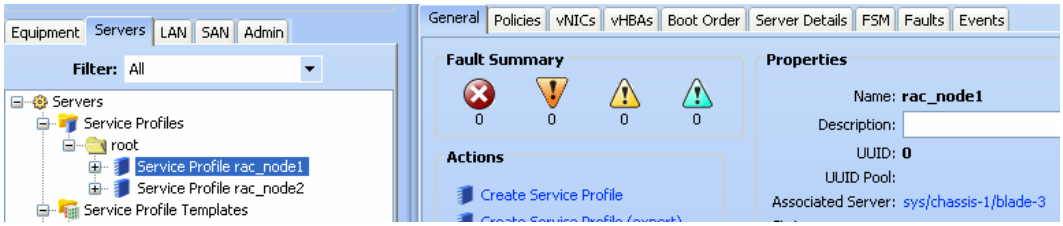
privileges for SAN configuration include the right to configure VSANs, Fibre Channel adapter policies, Fibre Channel quality of service (QoS), and SAN security. One special privilege, called "admin," grants the user the right to modify all components in a Cisco Unified Computing System. Another privilege, referred to as "read-only," grants all Cisco UCS Manager users the right to read the configuration of all components in a Cisco Unified Computing System.

Simplicity Through a Hierarchical Information Model

All the physical and logical components that comprise the Cisco Unified Computing System are represented in a hierarchical management information model referred to as a management information tree. Each node in the tree represents a managed object (or group of objects) that contains its configuration and operational state.

Hierarchically organizing the management information reflects an intuitive containment of the objects. It also facilitates parallel and asynchronous management operations on individual managed objects and on subtrees of managed objects. End users can get a glimpse of this tree by examining the organization and labeling of the management information in the GUI, as seen in Figure 2.

Figure 2. View of Cisco UCS Manager GUI



Management Using a Model-Driven Framework

The state information for a managed object includes both its administrative state (how it is configured) and its operational state (how it is running). Cisco UCS Manager's information model is a logical abstraction of the hardware and software components in the Cisco Unified Computing System.

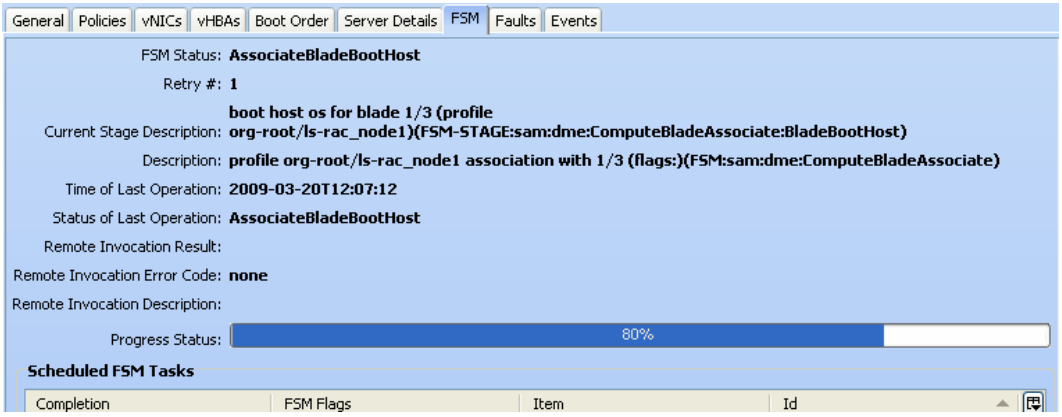
The information model is centrally stored and managed by the data management engine (DME), a user-level process running as a Cisco NX-OS Software plug-in. When a user initiates an administrative change to a Cisco Unified Computing System component (for example, applies a service profile to a server), the DME first applies that change to the information model, and subsequently the change is applied to the actual managed endpoint. This approach is called a model-driven framework.

With the model-driven framework, Cisco UCS Manager separates business logic from platform implementation. This approach lets programmers develop the business logic independent of the platform implementation, thus permitting more rapid support for new platforms. This approach also helps simplify software maintenance, since model logic errors should be easily distinguished from platform implementation errors. Additionally, troubleshooting and maintenance operations are simplified because natively verifying that a particular management operation has succeeded should be easy to accomplish. Finally, this separation permits the insertion of a software layer that emulates the platform components.

The entities that can change the administrative state of a managed object are referred to as stimuli. Management interfaces, such as the CLI and GUI, generate stimuli in a Cisco Unified Computing System. A less obvious example of a stimulus generator is another managed object. For instance, if a fabric extender fails, then the operational state of the network interface ports connected to that fabric extender will also be affected.

Administrators have limited visibility into the model-driven aspect of the Cisco UCS Manager framework. Finite state machines (FSMs) that are displayed in the user interface reflect the intersection of the information model and the managed endpoint, as depicted in Figure 3.

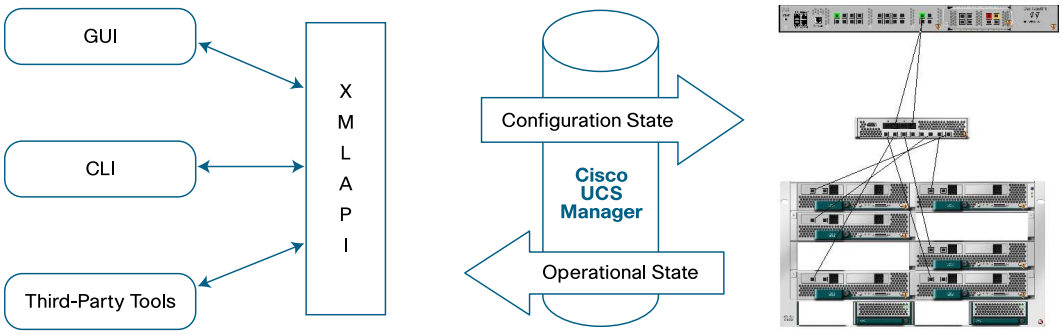
Figure 3. View of Finite State Machine



Three-Tiered Management Infrastructure

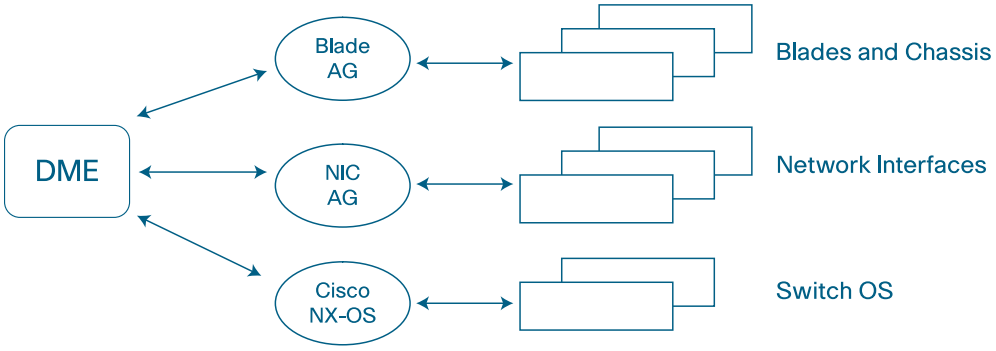
Cisco UCS Manager isolates the back-end platform implementation from the business logic of the information model and is designed with independent front-end user interfaces. In this regard, it is reasonable to consider the Cisco UCS Manager framework a three-tier architecture, as shown in Figure 4. The figure summarizes the direction of the state changes in Cisco UCS Manager. Configuration state changes originate in the user interface, are then applied through service profiles, to the managed object in the information model, and finally are deployed to the managed endpoint. Conversely, operational state changes originate in the managed endpoint, are applied to the managed object in the information model, and finally appear in the user interface.

Figure 4. Separation of Business Logic and Hardware Implementation



Not shown in Figure 4 are the entities, called application gateways (AGs) that bridge the DME and the managed endpoints. These application gateways run as multithreaded daemon processes on Cisco NX-OS. Application gateways implement the platform-specific details on each managed endpoint and so can be considered component-level device managers. In Cisco UCS Manager, the application gateways are implemented for the Cisco NX-OS, chassis, blades, ports, host agents, and network interface cards (NICs). The DME, application gateways, and managed endpoints comprise a three-tiered software architecture as well, as shown in Figure 5.

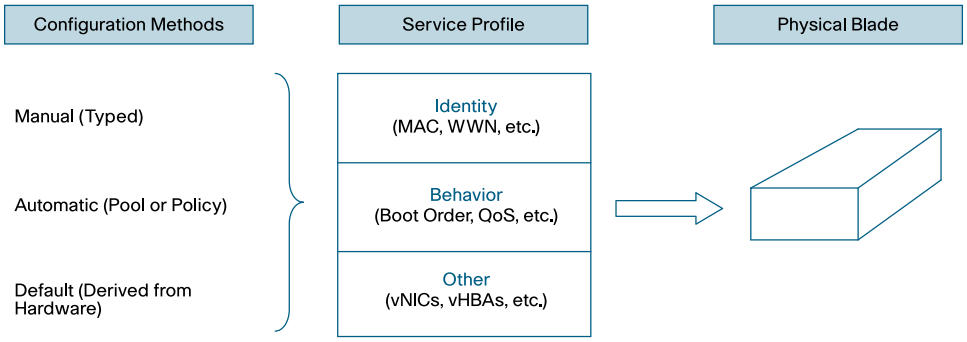
Figure 5. Three-Tiered Architecture Within Cisco UCS Manager



Example: Associating Service Profiles to Blades

For example, the administrative state of blades is managed with service profiles. A service profile contains values for a server's property settings: virtual network interface cards (vNICs), MAC addresses, boot policies, firmware revisions, and other elements. By abstracting these settings from the blade into a profile, a server can be deployed to any blade in the Cisco Unified Computing System. Furthermore, the profile can, at any time, be migrated from one blade to another. Figure 6 summarizes the information that is included in a service profile and shows how that information is populated in the profile and ultimately associated with a blade.

Figure 6. Service Profile Elements



A service profile can be associated with a blade either manually (using the GUI or CLI) or automatically (using a policy or the XML API). When a service profile is associated with a blade, the DME performs all necessary logical and physical mutations to fulfill the request. Specifically, a thread of the DME process, referred to as the "doer" thread, creates a service profile association workflow in the form of an FSM. The doer thread will mutate the managed objects in the management information tree according to the settings in the service profile. The doer thread will then call the appropriate application gateways to actually deploy the configuration changes to the managed endpoints. The application gateways will transition through platform-specific workflows to

help ensure that all endpoints are modified appropriately. The doer thread will not wait for the application gateway to complete its task. Instead, the doer thread will process the next task in its queue. When the application gateway completes its task, it will signal the doer thread, at which point the doer thread will transition to the next task in the FSM. The DME orchestrates all management stimuli serially and transactionally to guarantee that all requests to mutate the management information tree are successfully performed in the order they were requested.

Managed Objects and Managed Endpoints

With Cisco UCS Manager, all configuration changes mandated by the service profile are first applied to the managed object in the information model (by the DME) and are then applied to the actual managed endpoint (by the appropriate application gateway). These deployments of management to managed objects are fully transactional and abide by the atomic, consistent, isolated, and durable (ACID) requirements of a standard database transaction. As far as the DME is concerned, the deployment of administrative state to the managed endpoint is just a side-effect of the database transaction. However, the deployment of administrative state to the managed endpoint is not transactional in nature—this is an important distinction between the model and the managed endpoint implementations.

Application gateways compare the administrative state of a managed object with the operational state of the managed endpoint and take appropriate action. Application gateways are stateless and their stimuli are idempotent, making Cisco UCS Manager a robust framework for deploying management, particularly in the event of failures. When an application gateway fails to deploy management to an endpoint, that failure is reflected in the operational state of the managed object. At the end of any transaction, whether it was successful or not, all affected objects are inspected, and appropriate managed object rules are applied.

Cisco UCS Manager Notifications

As previously mentioned, the DME centrally stores the administrative and operational states of all managed endpoints, making the DME a good candidate for notifying all interested client programs about state and configuration changes as they occur. Indeed, the DME was designed with a generic mechanism to publish notifications to clients so that the clients can maintain a consistent and up-to-date view of management information without ever having to poll Cisco UCS Manager. The GUI and CLI both subscribe to events so that they reflect the most up-to-date state and configuration of the system at all times.

Cisco UCS Manager makes use of generic object notifications in the management of events and faults. An event is a representation of something that momentarily occurred in the system; for example, insertion of a blade into a chassis is represented as an event. By contrast, a fault represents something that failed in the system: the failure of a fan module is represented as a fault. Events and faults are themselves objects that are managed by Cisco UCS Manager and are subject to the same base set of rules as other managed objects in Cisco UCS Manager. However, events and faults have additional rules that specifically apply to them. For example, event managed objects are immutable because they usually correspond to a momentary, nonpersistent condition that one would want logged. Fault managed objects, in contrast, are mutable because the operational state of a managed endpoint may transition between a failed state and a functioning state. Such a fault is known as a cleared fault. Cisco UCS Manager has a user-configurable policy for the handling of cleared faults. Cisco UCS Manager can automatically delete or retain cleared faults for a certain amount of time. For both events and faults, Cisco UCS Manager supports the concept of expiry in the lifecycle of their corresponding managed objects; for

example, after a certain period of time or when the logs have reached a certain size, Cisco UCS Manager can destroy the oldest managed objects.

Conclusion

The Cisco UCS Manager model-driven framework separates the Cisco UCS Manager logic from platform implementation, making the software cleaner and easier to maintain. The Cisco UCS Manager's information model is hierarchical, which makes manipulating the management information tree more intuitive for XML API users. Managed objects can be modified either through configuration state changes initiated by an administrator or through operational state changes that occur in the managed endpoint, both of which occur as ACID-compliant transactions in the DME. Users can use either the GUI or CLI to manage the Cisco Unified Computing System, or they can use the XML API to write their own custom interfaces. The Cisco Unified Computing System provides high availability to Cisco UCS Manager instances by running two switches in the environment.

Cisco Unified Computing Services

Using a unified view of data center resources, Cisco and our industry-leading partners deliver services that accelerate your transition to a unified computing architecture. Cisco Unified Computing Services help you quickly deploy your data center resources, simplify ongoing operations, and optimize your infrastructure to better meet your business needs. For more information about these and other Cisco Data Center Services, visit http://www.cisco.com/go/unifiedcomputingservices.

Why Cisco?

The Cisco Unified Computing System continues Cisco's long history of innovation in delivering integrated systems for improved business results based on industry standards and using the network as the platform. Recent examples include IP telephony, LAN switching, unified communications, and unified I/O. Cisco began the unified computing phase of our Data Center 3.0 strategy several years ago by assembling an experienced team from the computing and virtualization industries to augment our own networking and storage access expertise. As a result, Cisco delivered foundational technologies, including the Cisco Nexus™ Family, supporting unified fabric and server virtualization. The Cisco Unified Computing System completes this phase, delivering innovation in architecture, technology, partnerships, and services. Cisco is well-positioned to deliver this innovation by taking a systems approach to computing that unifies network intelligence and scalability with innovative ASICs, integrated management, and standard computing components.

For More Information

For more information about the Cisco Unified Computing System, visit http://www.cisco.com/go/unifiedcomputing or contact your local Cisco representative.

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV
Amsterdam, The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.

Printed in USA C11-525344-00 04/09