Core functions of the highly available application-ready platform: Don't promise availability without them

By Dr. Asif Naseem

As standards-based hardware platforms such as AdvancedTCA, MicroTCA, and BladeCenter gain increased application in the telecommunications market, Telecom Equipment Manufacturers (TEMs) are looking for preintegrated, pretested application-ready platforms. These platforms need to include standards-based, sophisticated high availability management capabilities. The value of such capabilities lies in ensured service continuity in the face of various hardware and software failures. Current approaches demonstrate that using a standard set of services to implement highly available network elements can save TEMs a significant amount of effort, cost, and time to revenue. Asif describes a few of the key concepts and services required in such an endeavor and presents a real life example.

Modeling the system

One of the key features of a highly available system is its ability to represent the configured system resources – managed objects – in a dynamic system model that can be used in implementing sophisticated availability management policies. The system represents each resource that is to be managed as a managed object in the system model. It also captures resource dependencies, including intricate relationships that form a given service. The system model ensures that project teams can easily maintain and/or change a system configuration over time. The key features of the system model include the following:

- Management representation of resources based on relevant industry standards, such as ITU Recommendation X.731
- Scaling to 10,000+ managed objects
- Defined attributes for health, operational state, administrative state, role, availability status, and dependencies
- Methods for access/control, monitoring, and configuration
- Logical representation of redundant resources and any arbitrary collection thereof
- Implementation of important redundancy policies such as 2N, N+1, N+M, and Active/Active

Highly available systems must provide redundancy of the system model to ensure uninterrupted availability management. Systems configured with more than one node can automatically replicate the system model from the active node to a standby node. This ensures the system can quickly fail over and continue service in case the manager node experiences a failure or can no longer communicate with the other nodes in the system.

Creating and maintaining a dynamic system model for managing system availability requires a set of sophisticated platform resource management services. Let us look at a few that are essential.

Platform resource management

The Platform Resource Management Service (PRMS) offers a standard abstraction layer that provides the ability to integrate with the hardware resources of a particular platform when integrated with the Service Availability Forum (SA Forum) Hardware Platform Interface or HPI. (SA Forum documentation is available at: www.saforum.org/specification.) PRMS can provide expansive platform management capabilities in a way that is both standard and hardware agnostic. Let us look at each in some detail.

Discovery

A discovery capability allows user applications to automatically discover and enumerate the set of hardware resources present within the system, along with the respective management capabilities those resources possess. This service enables discovery and monitoring of real-time dynamic information on all the configured hardware resources including compute cards, networking boards, fans, and power supplies. Furthermore, this service collects and maintains inventory management information about various hardware entities, which typically includes information such as the manufacturer ID, product name, product version, and serial number. This information is used to populate the system model, manage the platform resources, and apply appropriate availability management policies.

Monitoring

This capability of the PRMS/HPI service includes detailed mechanisms to monitor the health and performance of various hardware resources. The notification and logging capabilities provide a mechanism to monitor, communicate, and log various events occurring in the system. Key examples include the following:

- *Sensor events* that identify the change in the state of a sensor, such as a temperature or a voltage gauge exceeding or dropping below one of its predefined thresholds
- *Hot swap events* that identify a change in the hot swap state of a Field Replaceable Unit (FRU)
- *Resource failure events* that identify whether a resource has failed, has been restored to a healthy state, or has been added to the system

Management

Manageable hardware resources – known as HPI entities – generally have one or more management instruments associated with them. Some key examples include sensors, controls, watchdog timers, and annunciators. Sensors provide information on an entity through the measurement of a critical hardware entity attribute, such as a voltage sensor or a temperature sensor. As the states of sensors change, this service will also send event notifications to all subscribing applications identifying the change in sensor state, such as a voltage or temperature sensor exceeding a critical threshold. Controls provide read/write access to control devices associated with hardware entities such as LEDs, dry contact closures, LCD displays, and audible alarm indicators. In addition controls allow a user application to customize the manner in which information (on alarms for example) is communicated to the system administrator. Watchdog timers monitor the health of a system by ensuring that critical aspects of the system, for example BIOS operations or the loading of the operating system, are progressing well. Annunciators provide abstracted controls that can have a set of alarm conditions associated with them. The annunciators ensure that the alarms are properly annunciated through the platform's and the entity's alarm indicators.

Hot swap management

Hot swap ability is a critical requirement of highly available systems. Such systems provide Hot Swap Management Service (HSMS) to manage insertion and extraction sequences for the FRUs in the system. Although HSMS typically builds upon the functionality of the PRMS, it goes beyond PRMS and HPI capabilities. For example, HPI provides a standardized interface for accessing and managing hardware resources but lacks more advanced management functionality, such as managing resource(s) that depend on a hardware resource or FRU that can be dynamically inserted or extracted from a system.

HSMS, on the other hand, in conjunction with PRMS, fully manages hot swap insertion and extraction sequences within the system. HSMS assists in each step of hot swap insertion and extraction sequences to allow the operator to manage and control the hardware resources being inserted into the system, as well as coordinating the controlled shutdown of software resources, such as applications, that depend on a hardware resource being extracted from the system. Sophisticated hot swap management services provide this management functionality with minimal programming. HSMS typically includes predefined hot swap management policies and actions for each of the key steps in the managed hot swap sequences. The operator can choose from among the predefined set to quickly and easily establish the hot swap management policies. For systems requiring more complex or custom policies, HSMS provides programmatic interfaces to extend the predefined HSMS policies and actions.

Alarm management

Alarm management is the process of monitoring a system for conditions that may jeopardize healthy operations and using this information to implement policies to take appropriate action in response to various alarm conditions. In a highly available system, the Alarm Management Service (ALMS) deals with alarm conditions found at the software as well as the hardware level. ALMS custom code enhances and further integrates the particular hardware platform. Working together, PRMS, HSMS, and ALMS offer a powerful set of functionality required to create a highly available application-ready platform. Although the ALMS is designed to work as a standalone service, it provides enhanced functionality when running in conjunction with PRMS. ALMS includes the following functionality:

- Maintains an active alarm list for the system that can be queried by the management applications to get the latest alarm status information
- Enables user creation/deletion of alarm conditions, including automatic alarm annunciation on the system platform(s); allows the management applications to identify additional alarms in nonhardware resources and still have these software-related alarms annunciated on the system platform(s) consistently with other alarm conditions
- Allows acknowledgement of alarm conditions including automatic updating of the system alarm annunciation
- Automatically identifies alarm conditions based on changes in system state, which enables the operator to define the associated alarm level for different events in the system and have ALMS automatically generate the alarm conditions based on the particular configuration settings
- Redundant service design to eliminate single points of failure and to ensure ALMS functionality is continually accessible on the system
- Detection, generation, and annunciation for resources undergoing a state transition to a state of failed or no resource, for resources that do not belong in the system model, and for resources whose sensor indicates a temperature or voltage threshold violation
- Configurable severity levels – critical, major, and minor – to adjust how the system reacts to an alarm condition, and configurable actions to be taken in the event that sensor temperature or voltage thresholds are crossed

Vendor specific integration

Often various hardware platforms provide a differentiated set of monitoring and management capabilities. Such items as sensors, control, special displays, and annunciators can be employed to tailor platform management for particular applications or simply enhance management capabilities. It is helpful to have platform management services with extensibility. This allows vendor-specific capabilities to be included to augment the standard set of management capabilities. HPI includes these provisions, and services such as PRMS, HSMS, and ALMS can be designed to take advantage of these extensions.

An example implementation – RadiSys and GoAhead

GoAhead Software, Inc. and RadiSys Corporation have worked together to build an application-ready platform for equipment manufacturers from a variety of markets, who all must ensure carrier grade (99.999 percent and above) service availability. Makers of next-generation networks and IP multimedia subsystems are among those looking toward preintegrated platforms as a way to ensure availability and accelerate time to market for new services. By leveraging a solution based on standards, these equipment manufacturers can repurpose their original investment many times across multiple applications, gaining significant cost efficiencies. Some examples of applications that leverage this carrier grade application-ready platform include Session Initiation Protocol (SIP)-enabled applications, media gateways, and softswitches.

Such a platform has been created by utilizing the suite of high availability and platform management services available in GoAhead SelfReliant and on the RadiSys Promentum AdvancedTCA hardware platform running the Linux operating system, as shown in Figure 1. The critical integration elements to create this platform are depicted in Figure 2.

PRMS along with HPI, HSMS, and ALMS form key components of this integration. PRMS has been integrated with the HPI libraries implemented in the hardware. Additionally HPI resource discovery capabilities enumerate all the resources to create an initial system model, and to dynamically update this model as the various resources' roles and states change while different system events take place. These resources – or managed objects – can either be local to the cluster or remote, in which case they are represented by a remote adapter. A Remote Adapter Factory (RAF) is employed to maintain the current state of such managed objects. A state engine – depicted as a *role* – establishes if a PRMS instance is active, standby, or unassigned. A Hardware Abstraction Layer (HAL) translates standard HPI calls into actions that are specific to the underlying hardware. Conversely it translates hardware events such as hot swap events, into appropriate HPI notifications that are sent upstream to the management application. HAL also uses an error-handling component to identify and recover from errors that may occur during HPI calls. These error-handling policies are customizable through an XML file.

The integration of these capabilities is carried out to ensure portability across different hardware capabilities. As such a platform specific library provides abstraction from any differences in the underlying platform, that is, any differences in HPI implementation, such as OEM specific data types, platform specific configuration, and the like. Finally, a console component is used to implement a browser-based console to access and display various platform attributes in a readable format.

Conclusion

As we have discussed, there are fundamental elements that must be present in order to have a robust highly available system: a dynamic mechanism for enumerating system resources; platform resource management that enables discovery, monitoring, and management of hardware resources; comprehensive and graceful hot swap and alarm management; and integration with vendor-specific hardware sensors and controls. While selecting standards-based hardware and high availability management middleware offers unquestionable value, much work remains to be done to integrate the two to the level described here. A preintegrated solution based on standards takes this effort out of the equation and enables equipment manufacturers to focus on differentiation at the application layer by leveraging an application-ready platform that has all of this functionality already baked in. This offers significant time-to-market improvements. Moreover, because the platform is standards-based, allowing application portability, it enables TEMs to optimize and protect their investments and ensure that platform changes can be made later without a significant loss in already-developed platforms.

Dr. Asif Naseem, *Chief Technology Officer and Chief Operating Officer, GoAhead Software, is also the president of the Service Availability Forum. He has more than 19 years of experience in the computer and communications industry. Asif is a veteran speaker having presented at national and international events such as ITU Telecom Geneva, GSM World Congress, CTIA, and numerous other events. He has also presented papers at conferences organized by ACM, IEEE, and others, and is a frequent author of papers published in several technical journals and magazines. He has an MS in Electrical Engineering and a PhD in Computer Engineering from Michigan State University.*

To learn more, contact Asif at:

GoAhead Software
10900 N.E. 8th Street, Suite 1200
Bellevue, WA 98004-1455
425-453-1900
anaseem@goahead.com
www.goahead.com

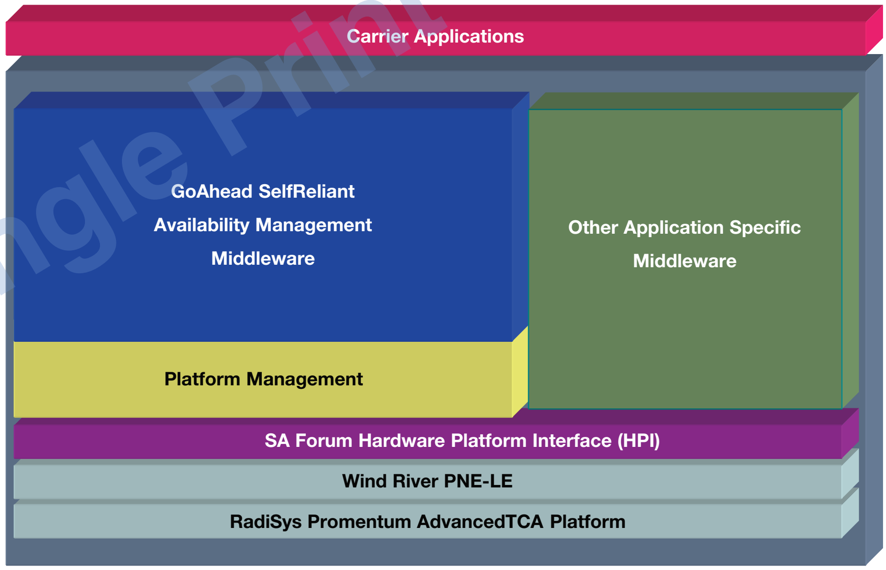

Figure 1

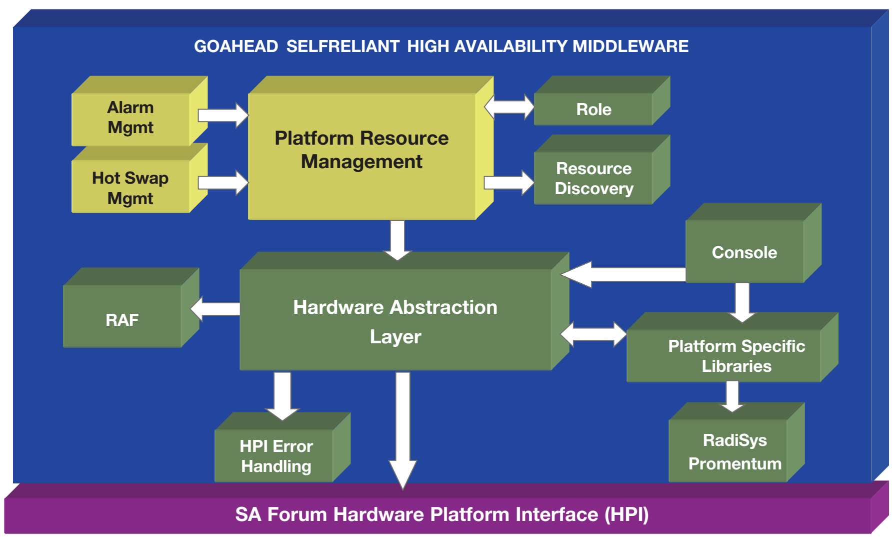

Figure 2