High Availability and Fault Management in Objective Architecture Systems

Steve Mills, Senior Systems Architect
Outline

• Standards/COTS and the Mission-Critical Requirement
• Defining High Availability (HA)
• What is (OA) Open Architecture?
• Introduction to SA Forum Services
• Conceptual Alignment between OA and SA Forum
• Summary
Introduction

• Mission Critical Systems

• System architecture requires modularity and scalability

• COTS Solutions in general gaining traction

• The Service Availability Forum High Availability standards are gaining momentum for addressing mission critical requirements
  – DOD Information Technology Standards Registry (DISR), DOD-wide mandated standard
  – USN PEO IWS Combat System Product Line Architecture Description Document
Mission Critical Requirements

• Maintain continuous availability
  – Distributed infrastructure to support cooperating redundant applications
  – Infrastructure oversight of compute nodes and health awareness

• Automated, policy-based fault management cycle
  – Configuration of fault detection criteria
  – Deterministic fault recovery modeling
  – Sub-second MTTR recovery for real-time applications

• Distributed system management – framing a single coherent system
  – Central configuration and administrative control
  – Centralized alarm, notification, and log support

• Modular Open Systems Approach (MOSA)
  – Open standards, COTS based solutions
  – Reduce costs and risks
The Essence Of The High Availability Requirement

• Goal: Continuous availability (99.999% or better) despite
  • Hardware, software, human, and communication failures
  • Planned maintenance (migration, upgrade)
• Achieved via no single point of failure
  • Redundancy of mission critical applications
  • Failure detection at HW, OS, Node and App layers
  • Automated, policy-based real-time recovery and repair:
    • Sub-second, stateful failover
    • Alarms and Notifications
    • Custom policies
• Proactive system monitoring with preventative measures
• Clear integration points and modular design
Open Architecture Goals

Many meanings, but core is:

- Flexibility, openness, and re-use
- Layered infrastructure with common, open functions and interfaces
- Decouple domains of concern
- Modular, scalable
- Increased industry competition
- COTS, open standards
OA Reference Model and Infrastructure

- Critical boundary – OA infrastructure and OA-compliant applications
- Applications – modular components that plug into OA reference model
- Common, open features at this boundary include:
  - Vertical integration - Interfaces for components to access infrastructure services
  - Horizontal integration - Interfaces for components to bind to each other
  - External integration – Interface to external world for configuration and administration
  - Modeling – Explains component capabilities and policy driven behaviors
- The better an OA solution enables components to explain their properties, resource needs, and policies:
  - The less such logic needs to be repeated within the applications
  - The more consistent the OA infrastructure can manage those issues
Open Architecture - Infrastructure

• Layered to better partition responsibilities and decouple dependencies
• When open architecture is applied to a specific domain, it is often referred to as an “Objective Architecture”
• In our example here, the layers are:
  – Platform layer – hardware abstraction layer
  – Node layer – operating system abstraction layer
  – Middleware layer – set of services that treat distributed nodes as a logical world of a single system
    – NOT a physical place or location
      • Includes functions such as: logging, notification, stateful fail-over, fault detection, and…
  – Application integration layer – component integration layer with integration methods
• This OA has a mission critical requirement – which could be inherent in the problem domain (ex: combat system), or based on a mixed system where mission critical needs arise because of system density
SA Forum Availability Middleware Services

- Public Integration Points
  - System Model
  - Public APIs for each service
  - Formal alarms and notifications
  - Formal configuration access

- Key SA Forum Services
  - Availability Management Framework (AMF)
  - Platform Management (PLM)
  - Cluster Membership (CLM)
Availability Management Framework
Model the distributed, redundant applications

Mission Critical Apps
Distributed redundant applications configured in Active/Standby relationships

System Model Policies
Deterministic policies drive the AMF Fault Mgmt Policy Engine

Mission Critical System
- Critical applications
- OS, MW Cluster Nodes
- Hardware & LRU inventory

Logical Modeling Entities
- Service Unit (SU)
- Service Group (SG)

GoAhead OpenSAFFire

- Power
- Cooling
- Shelf Mgr

SM

SG (2N) SU A SU S

SG (N+M) SU A SU A SU S

Linux Linux OS OS OS
Cluster Membership
Model the cluster node members & state

A cluster made up of a chassis and other compute nodes

Three clusters partitioned over several chassis

A cluster can consist of the same or heterogeneous devices such as chassis, rack mount servers, etc. as long as each has visibility to each other through (redundant) communications paths
PLM Manages Hardware Resources
- Model HW elements and dependencies
- Automated validation of LRU Inventory
- Hot swap management
- Configurable power and temperature threshold alarms
- Power on, off and reset of HW resources

PLM Manages Execution Environments (Operating System)
- Model execution environments
  - Hosted directly by computer
  - Hosted by hypervisor
- Administrative control of operating system state
Two Models – Conceptual Alignment

Open Architecture Reference Model

- **Track**: Hypothetical Application Domain
- **Target**
  - ID
  - Schedule
  - Record

- **Middleware**
  - Application Integration
  - OA Public Integration Methods
- **Node Participation**
  - SA Forum, DDS, Corba
  - Node/OS clustering as “system”
- **Hardware Platform**
  - LRU, CPU, Memory

SA Forum Services

- **Applications**
  - SA Forum Public Integration Points
    - System Model
    - Public APIs
    - Alarms & Notifications
    - Configuration
  - SA Forum Availability Middleware Services
    - Software Mgmt Framework
    - Availability Management Framework
    - Information Model Management
    - Platform Management
    - Notification
    - Log
  - SA Forum Cluster Membership
  - Operating System
  - SA Forum Hardware Platform Interface (HPI)
    - Hardware Platform A
    - Hardware Platform B
    - Hardware Platform C
    - Hardware Platform D
Summary

- SA Forum and objective architecture align well
- Core Principles of open architecture reflected in SA Forum
- SA Forum Standards – the ONLY proven open standard for mission-critical systems
- SA Forum standards can be integrated into new systems or legacy scenarios
- Deployed in programs such as Aegis Weapon System, Littoral Combat Ship, Common Processing System, Deepwater
Thank You

Stephen Mills
GoAhead Software
Senior Systems Architect

s@goahead.com
781-654-7273