Health Management System: An Application of Recovery Oriented Computing (ROC) Targeted at HPEC Systems
HA Approach: Redundancy

• Goal: Increase Mean Time To Failure
• Classes
  • Dual Redundancy (Hot Fail Over)
  • Triple Redundancy (Result Comparison)
• Redundancy at System/Component Level
• Drawbacks:
  • High Costs
  • Low Density
  • Additional Complexity
SKY's HAA Approach: Recovery Oriented Computing

Two Basic Tenets:
- Failure Rates of Both Software and Hardware are Non-Negligible and Increasing
- Systems Cannot be Completely Modeled for Reliability Analysis (thus their failure modes cannot be predicted in advance)

Goal:
- Decrease Mean Time to Repair

ROC Mechanisms:
- Detection (Sensing and Diagnostics)
- Isolation
- Use of Excess Capacity (if available)
- Repair/Recovery
Applying ROC to HPEC

Hardware:
- Quality Components
- Built-in Sensing of all Major Components
- Control of all Major Components (reset, etc.)
- Excess Capacity (where possible)

OS Middleware:
- Quality Components
- Built-in Sensing of all Major Components
- Control of all Major Components

Application:
- Quality Components
- Built-in Sensing of all Major Components
- Control of all Major Components
- Overall System Management (Sensing and Control)
Hardware Support

• HAA Support Blade
  - Tini Management Processor (Java Processor)
  - I2C Integration
  - TCP/IP External Access

• Compute/IO Blades
  - Out-of-band Management Controller
  - Temperature Monitoring
  - Voltage Monitoring
  - Heart Beat Monitor
  - Power Control/Reset
  - I2C Integration

• Chassis
  - Fan Monitoring
  - Voltage Monitoring
  - Power Control/Reset
Health Management System (HMS)

GOALS:

• Provide Capability to Instrument OS, Middleware, and Application (analogous to hardware instrumentation)
• Provide Uniform View of Entire System (hardware, OS, middleware, and application)
• Provide Integrated Diagnostics
• Provide Access Using Standard Interfaces
• Minimal Performance Impact
• Easily Extensible and Configurable (in order to meet individual application requirements)
Health Management System (HMS)

- Server Objects: Sensors, Controllers, and Timers
  - Embedded within the hardware, OS, middleware, and application
  - Combined into a Resource Object
- Clients: Application, Management Tools, and Users
- Communication: Event Driven, Request Driven, and Timer Driven Messaging
- Lookup Services
- Extensible
  - Can support an arbitrary number of servers and clients
  - Application developers can add application specific servers
- Configurable
  - Which servers and clients are to run
  - When and where they are to run
Example HMS Based System
Health Management System

• Used to Monitor Resource Usage (Development and Runtime)
  - Hardware (temperature, voltage, etc)
  - OS/Middleware (processor load, data throughput, etc)
  - Application (queue lengths, wait times, etc)
• Used to Manage These Resources
• Used to Detect and Isolate Faults
• Used to Predict Possible Future Faults
• Used to Gather Statistics on Resource Usage and Performance
• Used to Determine the Health of Resources (Diagnostics)
Future Directions

- Tight Integration with SKY Analysis Tools
- Tight Integration with SKY Development Tools
- Pattern-based Application Recovery Libraries
- Dynamic Insertion of Sensors/Controllers (Dynamic Probes)
- Support for Other Hardware Environments (Hot-Swap)