A Failure Predictive and Policy-Based High Availability Strategy For Linux HPC Cluster

• Box Leangsuksun, Tong Lui and Tirumal Rao, Louisiana Tech University
• Stephen Scott, Oakridge National Lab, DOE
• Richard Libby, Intel Corporation

Tux idol contestants

tux02 tux03
Outline

• Background and Motivation
• Related works
• HA-OSCAR and current release
• From detection to prediction?
• Summary and future work

Beowulf Cluster

Unavailable = No performance and functionality
Related Works

• Serviceability – easy Beowulf Builders
  – OSCAR, ROCKS etc

• High Availability
  – LinuxHA, Kimberlite, HP ServiceGuard, Redhat AS

• Monitoring
  – Mon, netSNMP.

HA-OSCAR Goals

• High Availability for HPC cluster
  – Fault-resilience and improve uptime at head
  – Self-healing

• Serviceability - Simplicity
  – Self-build (almost)
  – Self-configuration

• Preserve existing investments - No change required in usage, retrofitable

• Production-quality
Innovation and information technology

HA-OSCAR Beowulf

Optional Image servers

Self-healing Schemes
Adaptive recovery state diagram

Innovation and information technology

HA-OSCAR Cluster
Self-build installation

- How to build HA-OSCAR
  - Can retrofit an existing Linux Beowulf
  - Or start with OSCAR installation tool
  - HA-OSCAR GUI based installation tool

HA-OSCAR installation

- Adopt ease of build and operation same as OSCAR concept
- 30 min – 1.5 hrs installation (retrofit)
- Take almost the same time for disaster recovery

Step1

Step2 create head image

Step3 clone image

Step4 config & build Standby

Step5 (optional) web admin to add/config more services
Realities Checks

• Great! We got HA Beowulf!
• But How much improvement?
  – The total uptime?
  – Performance?
• Analytical model and prediction
  – Statistical technique to compare uptime
  – How many 9’s? (downtime per/year)
  – Stochastic Reward Net with SPNP package
  – Identical hardware parameters between Beowulf and HA-OSCAR multi-heads

Availability (uptime) improvement

HA-OSCAR(A) = 99.993 (36 min)  vs.  
Beowulf (A) = 99.65 (30 hr)
Hardware Management abstraction

- Ability to access and control detailed status for better management (CPU temp, baseboard, power status, system ID/ up/down etc.)
- IPMI (Intelligent Platform Management Interface)
- open IPMI and OpenHPI (SA forum)
- HW abstraction hints vendor specific
  - CPU
  - Power
  - Memory
  - Baseboard
  - Fan (cooling)

Example of IPMI event logs
Example of IPMI event logs

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Component</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>01/25/2004</td>
<td>00:31:19</td>
<td>Sys Fan 1</td>
<td>critical</td>
</tr>
<tr>
<td>01/25/2004</td>
<td>00:31:19</td>
<td>Sys Fan 3</td>
<td>critical</td>
</tr>
<tr>
<td>01/25/2004</td>
<td>00:31:19</td>
<td>Sys Fan 4</td>
<td>critical</td>
</tr>
<tr>
<td>01/25/2004</td>
<td>00:31:19</td>
<td>Processor 1 Fan</td>
<td>ok</td>
</tr>
<tr>
<td>01/25/2004</td>
<td>00:31:20</td>
<td>Processor 2 Fan</td>
<td>ok</td>
</tr>
</tbody>
</table>

Our early observations

- Can set thresholds in managed elements to trigger events with severity levels
- Automatic failure trend analysis -> prediction
A failure prediction & policy-based recovery
Cluster management

- Detections - the damage done!
- Predictions
  - trend analysis
  - Anticipate imminent failures
  - Better handling
  - More difficult for multiple events
- Example of IPMI events and trend analysis
  - E.g. CPU temp raising too fast with 5 min -> prepare to checkpoint, failover and restart
  - Memory bit error detected -> take a node out
DGAE Modules

- DGAE
- MON
- Net-SNMP
- OpenHPI
  - IPMI
  - IBM Blade Center
  - Watchdog
  - Sysfs
- Process.monitor
- Loadaverage.monitor
- Freespace.monitor
- Platform.monitor
- Fping.monitor

Process, Loadaverage and Freespace monitor are based on Net-SNMP. Freespace.monitor is based on OpenHPI.
Fping.monitor works with ICMP.

HA-OSCAR measurements

- 3-5 sec Manual failover time
- 0.9% CPU usage at each monitoring interval

Comparison of network usages for HA-OSCAR different polling sizes
Summary

• HA-OSCAR beta was released in March 2004
• Failure Perdition and policy based recovery proof of concept
• Prediction for known problems or events
  – Single event type can be anticipated
  – Active monitoring
  – Head node is achievable
  – Challenging for very large scale compute nodes and multi-events

Summary (continued)

• HA-OSCAR 1.0 Beta release (March 2004)
  – The first known field-grade HA Beowulf cluster release
  – Self-configuration Multi-head Beowulf system
  – HA and HPC clustering techniques to enable critical HPC infrastructure
  – Self-healing with 3-5 sec automatic failover time
  – 1-1.5 hour to self-build failover headnodes w/o preloaded OS
  – Optional Image Server for disaster recovery
  – Support existing HPC App(e.g. MPI) without any modification
Future work

- For unknown problems
  - Keep building knowledge base
    - Unknown ones -> known
  - Data mining and Machine learning may help
- Combined OS, HW, APP events/Trend analysis

Roadmap

- More sophisticated trend analysis and predictions (multi event correlations)
- Grid-aware HA-OSCAR
- Support other cluster distributions (ROCS, SCYLD)
- Multi-head n+1 active-active
- Hardware abstraction and policy-based recovery management
- Hot-upgrade cluster (OS/CMS)
- Fault-tolerant applications/services framework
- FCAPS Management & carrier grade
### Three flavors

<table>
<thead>
<tr>
<th>Monitor</th>
<th>HA-OSCAR Active-Hot Standby</th>
<th>HA-ROCKS Active-Hot Standby (lab grade)</th>
<th>HA-OSCAR 2+1 Active-Active (lab grade)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBS, MAUI, NFS, HTTPD</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>SGE</td>
<td></td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>NIS</td>
<td></td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Gmond, Gmetad</td>
<td></td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Heartbeat (3 sec)</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>

### Three flavors (continued)

<table>
<thead>
<tr>
<th>Monitor</th>
<th>HA-OSCAR Active-Hot Standby</th>
<th>HA-ROCKS Active-Hot Standby (lab grade)</th>
<th>HA-OSCAR 2+1 Active-Active (lab grade)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Fan Speed</td>
<td>✔️, IPMI option</td>
<td>✔️, IPMI option</td>
<td>✔️, IPMI option</td>
</tr>
<tr>
<td>CPU Temperature</td>
<td>✔️, IPMI option</td>
<td>✔️, IPMI option</td>
<td>✔️, IPMI option</td>
</tr>
<tr>
<td>CPU status</td>
<td>✔️, IPMI option</td>
<td>✔️, IPMI option</td>
<td>✔️, IPMI option</td>
</tr>
<tr>
<td>Memory bit error</td>
<td>✔️, IPMI option</td>
<td>✔️, IPMI option</td>
<td>✔️, IPMI option</td>
</tr>
</tbody>
</table>
Q&A

Thank You

tux02

tux03

Backup
Network monitoring (sample)

Enhanced HA-OSCAR stacks

HA-OSCAR Management Module
Policy-Based Failover Management (PBFM)
High Reliable-Network File System (HR-NFS)
Data Gathering and Analysis Engine (DGAE)

OS (Linux)

OpenHPI

device drivers (other hardware interface)

Hot-Swap/Component Redundant Hardware Platform