Asymmetric / Active-Active High-Availability for High-End Computing

C. Leangsuksun
V.K. Munganuru
Louisiana Tech University
Ruston, Louisiana - USA
{box, vkm001}@latech.edu

T. Liu
Dell Inc.
Austin, Texas - USA
Tong_Liu@dell.com

S.L. Scott
C. Engelmann
Oak Ridge National Laboratory
Oak Ridge, Tennessee - USA
{scottsl, engelmannc}@ornl.gov

Second International Workshop on Operating Systems, Programming Environments and Management Tools for High-Performance Computing on Clusters
June 19, 2005
Cambridge, Massachusetts (USA)
Outline

- Motivation
- Related Work: OSCAR
- HA-OSCAR: RAS Management for HPC
- Clusters: Self-awareness Approach
- Analysis & Experiment
- Summary & Future work
Motivation

- Cluster architecture dominates HPC community.
- Cluster architecture is prone to single-point-of failure (SPoF).
- Cluster size has significantly grown.
  - Size and reliability have inverse relationship...
- Self-aware Reliability, Availability and Serviceability management is needed.
Cluster “Beowulf” Architecture

Single Point of Failure
Single Point of Control
Availability of HEC Systems

- Today’s supercomputers typically need to reboot to recover from a single failure.
- Entire systems go down (regularly and unscheduled) for any maintenance or repair.
- Compute nodes sit idle while a head or service node is down.
- Availability will get worse in the future as the MTBI decreases with growing system size.
- Productive computation is not done during the checkpoint/restart process.
# Availability Measured by the 9’s

<table>
<thead>
<tr>
<th>9’s</th>
<th>Availability*</th>
<th>Downtime/Year</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>90.0%</td>
<td>36 days, 12 hours</td>
<td>Personal Computers</td>
</tr>
<tr>
<td>2</td>
<td>99.0%</td>
<td>87 hours, 36 min</td>
<td>Entry Level Business</td>
</tr>
<tr>
<td>3</td>
<td>99.9%</td>
<td>8 hours, 45.6 min</td>
<td>ISPs, Mainstream Business</td>
</tr>
<tr>
<td>4</td>
<td>99.99%</td>
<td>52 min, 33.6 sec</td>
<td>Data Centers</td>
</tr>
<tr>
<td>5</td>
<td>99.999%</td>
<td>5 min, 15.4 sec</td>
<td>Banking, Medical</td>
</tr>
<tr>
<td>6</td>
<td>99.9999%</td>
<td>31.5 seconds</td>
<td>Military Defense</td>
</tr>
</tbody>
</table>

- Enterprise-class hardware + Stable Linux kernel = 5+
- Substandard hardware + Good high availability package = 2-3
- Today’s supercomputers = 1-2
- My desktop = 1-2

* Based on (MTBI) – mean time between interrupt – both software and hardware interrupts.
Solution: Active Redundancy
Clustering High-Availability Models

- Active – Hot-Standby
- Asymmetric / Active – Active
- Symmetric / Active – Active
What is OSCAR?

- Framework for cluster installation configuration and management
- Common used cluster tools
- Wizard based cluster software installation
  - Operating system
  - Cluster environment
    - Administration
    - Operation
- Automatically configures cluster components
- Increases consistency among cluster builds
- Reduces time to build / install a cluster
- Reduces need for expertise
HA-OSCAR: Active – Hot-Standby

- Production-quality Open source Linux-cluster project
- HA and HPC clustering techniques to enable critical HPC infrastructure Self-configuration Multi-head Beowulf system
- HA-enabled HPC Services: Active / Hot-Standby
- Self-healing with 3-5 sec automatic failover time
- The first known field-grade open source HA Beowulf cluster release
HA-OSCAR Serviceability

- Self-Build and configuration Multi-head Beowulf system
- Adopt ease of build and operation same as OSCAR concept
- ~30 min – installation
- Take almost the same time for disaster recovery (that is, each disaster recovery – providing you are prepared)

---

**Step1**

**Step2 create head image**

**Step3 clone image**

**Step4 config Standby**

**Step5 web admin to add/config more services**
Adaptive Recovery State Diagram

- **working**
  - Detect
  - threshold reached

- **failure**
  - switch over & take control at the standby
  - previous state, # counter, recovery

- **Alert**
  - after # retry
  - previous state, # counter, recovery

- **Failover**
  - After the primary node repair, then optional Fallback

- **previous state, # counter, recovery**

- **working**
  - Detect
  - threshold reached
Monitoring & Self-healing cores

PBS, MAUI, NFS, HTTP services are monitored

eth0, eth0:1 interfaces are monitored

load_average, disk_usage, free_memory are monitored

Self-Healing Daemon

Service Monitor

Resource Monitor

Health channel Monitor
HA-OSCAR RAS Software Stack

- Redundant H/W platform
- Intelligent sensors
- HPI wrapper
- Operating System (OS) hardware Interface
- OS Application Services
- Monitoring and Self-healing Core
- HA-OSCAR Management layer
Asymmetric / Active-Active Architecture
Failover of: Asymmetric / Active-Active Architecture
Asymmetric/Symmetric Active/Active

This diagram shows only connectivity of management channel. Primary communication (data transfer) channels are omitted here and typically separated from management channel.
Great! We got Highly Reliable HPC system!

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 vs Unavailability

- Planned and unplanned downtime
  - Scheduled downtime = 200 hrs
  - Repair time = 24 hrs
  - Monitoring interval = 10 sec

- Ours 99.99% vs 91.+% 

- 1k vs 10m TFLOP (1T system)

- $70k vs $2m ($20m system)