Active/Active HA Job Scheduler and Resource Management

a proof-of-concept implementation

Kai Uhlemann

Oak Ridge National Laboratory, USA
University of Reading, UK
University of Applied Sciences Berlin, Germany

March 15, 2006
Content

1 Overview
   - Project overview
   - Project goals

2 System design
   - Cluster system architecture
   - System concepts
   - External components
   - System design overview

3 Summary
   - Results
   - Future work
   - Conclusion

March 15, 2006
Content

1 Overview
   - Project overview
   - Project goals

2 System design
   - Cluster system architecture
   - System concepts
   - External components
   - System design overview

3 Summary
   - Results
   - Future work
   - Conclusion

March 15, 2006
Content

1 Overview
   - Project overview
   - Project goals

2 System design
   - Cluster system architecture
   - System concepts
   - External components
   - System design overview

3 Summary
   - Results
   - Future work
   - Conclusion
Research area of high availability for job scheduler
Efforts focused on the head node
Proposed solution uses the virtual synchrony paradigm
Proof-of-concept application shows possible design solution
Overview
System design
Summary

Project overview
Project goals

### Failure of Head Node

- Job submitted
- Job runs
- Job resubmitted
- Job runs
- Job done
- Job resubmitted
- Job runs
- Job done
- Job runs
- Job continues

#### Fault Tolerant Goal

- HA-OSCAR recovery/failover
- MTTR
- MTTR
- MTTR
- MTTR

#### Resubmission without Checkpointing

- MTTR
- Job resubmitted
- Job runs
- Job done

#### Resubmission with Checkpointing

- MTTR
- Job resubmitted
- Job runs
- Job done

March 15, 2006
Job scheduler service providing Active/Active HA
- No loss of scheduled job
- No loss of running job
- No restart of running jobs

Leading to uninterruptible service availability

Seamless failover (actually none)

Dynamic reconfiguration and recovery are completely masked

High transparency of HA capabilities from the user
Beowulf cluster architecture

Overview
System design
Summary
Cluster system architecture
System concepts
External components
System design overview

Beowulf cluster architecture:
- Head Node
- LAN
- Compute Nodes

March 15, 2006
HA-OSCAR cluster architecture

Active/Standby Head Nodes

Backup

LAN

Compute Nodes
Symmetric Active/Active HA cluster architecture

Active/Active Head Nodes

LAN

Compute Nodes

March 15, 2006
- Active state replication using virtual synchrony
- All messages are processed as uninterruptible events
- Global process state provided by external group communication system
- External replication used for resource and job management
- External components should be replaceable with new or similar ones
- Introduction of symmetric multi headed cluster architecture
- Any amount of head nodes possible $\Rightarrow$ scalable high availability
Overview

System design

Summary

Cluster system architecture

System concepts

External components

System design overview

MTBF 5000 hours  MTTR 72 hours

\[ t_{down} = 8760 \cdot (1 - A) \]

\[ A_{component} = \frac{MTBF}{MTBF + MTTR} \]

\[ A = 1 - (1 - A_{component})^n \]

<table>
<thead>
<tr>
<th>No.</th>
<th>HN</th>
<th>Availability</th>
<th>Est. downtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>98,580441640%</td>
<td>5d 4h 21min</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

March 15, 2006
Overview

System design

Summary

Cluster system architecture

System concepts

External components

System design overview

MTBF 5000 hours  MTTR 72 hours

\[ t_{\text{down}} = 8760 \cdot (1 - A) \]

\[ A_{\text{component}} = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}} \]

\[ A = 1 - (1 - A_{\text{component}})^n \]

<table>
<thead>
<tr>
<th>No.</th>
<th>HN</th>
<th>Availability</th>
<th>Est. downtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>98.580441640%</td>
<td>5d 4h 21min</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>99.979848540%</td>
<td>1h 45min</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

March 15, 2006
Overview
System design
Summary

Cluster system architecture
System concepts
External components
System design overview

MTBF 5000 hours  MTTR 72 hours

\[ t_{\text{down}} = 8760 \cdot (1 - A) \]

\[ A_{\text{component}} = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}} \]

\[ A = 1 - (1 - A_{\text{component}})^n \]

<table>
<thead>
<tr>
<th>No.</th>
<th>HN</th>
<th>Availability</th>
<th>Est. downtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>98,580441640%</td>
<td>5d 4h 21min</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>99,979848540%</td>
<td>1h 45min</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>99,999713938%</td>
<td>1min 30s</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Overview
System design
Summary

Cluster system architecture
System concepts
External components
System design overview

\[ t_{down} = 8760 \cdot (1 - A) \]

\[ A_{\text{component}} = \frac{MTBF}{MTBF + MTTR} \]

\[ A = 1 - (1 - A_{\text{component}})^n \]

<table>
<thead>
<tr>
<th>No.</th>
<th>HN</th>
<th>Availability</th>
<th>Est. downtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>98.580441640%</td>
<td>5d 4h 21min</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>99.979848540%</td>
<td>1h 45min</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>99.999713938%</td>
<td>1min 30s</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>99.999995939%</td>
<td>1s</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

MTBF 5000 hours  MTTR 72 hours
Overview

System design

Cluster system architecture

System concepts

External components

System design overview

MTBF 5000 hours  MTTR 72 hours

\[ t_{down} = 8760 \cdot (1 - A) \]

\[ A_{component} = \frac{MTBF}{MTBF + MTTR} \]

\[ A = 1 - (1 - A_{component})^n \]

<table>
<thead>
<tr>
<th>No.</th>
<th>HN</th>
<th>Availability</th>
<th>Est. downtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>98.580441640%</td>
<td>5d 4h 21min</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>99.979848540%</td>
<td>1h 45min</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>99.999713938%</td>
<td>1min 30s</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>99.999995939%</td>
<td>1s</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>99.999999942%</td>
<td>18ms</td>
<td></td>
</tr>
</tbody>
</table>
Group communication system transis

- Communication facilities for virtual synchrony
- Handles group membership changes

Resource management PBS TORQUE

- Enhanced version (>= version 2.0p1)
- Enabled multiserver architecture support

Job scheduler maui

- Commonly used in conjunction with PBS
Group communication system transis
- Communication facilities for virtual synchrony
- Handles group membership changes

Resource management PBS TORQUE
- Enhanced version (>= version 2.0p1)
- Enabled multiserver architecture support

Job scheduler maui
- Commonly used in conjunction with PBS
Group communication system transis
- Communication facilities for virtual synchrony
- Handles group membership changes

Resource management PBS TORQUE
- Enhanced version (>= version 2.0p1)
- Enabled multiserver architecture support

Job scheduler maui
- Commonly used in conjunction with PBS
Overview
System design
Summary

Cluster system architecture
System concepts
External components
System design overview

HA augmented cluster setup
Traditional cluster setup
HA augmented cluster setup

command line tools

Client using jcmd
External (optional)

invoke queue command

group communication (transis)
queue message
joshua
local queue operation
local resource manager (pbs_server)
local scheduler (maui)
head node

job submission and resource information exchange

jmutex enhanced jobscript

cluster nodes

March 15, 2006
Command line tools

- Perform queue operation such as
  - Job submission
  - Job status information
  - Job removal from the queue
- Designed for high transparency to the user
- Enables simple replacement for former job queue commands
- Achieved by message encapsulation of argument vector, stdin and environment
- Effective command is redirected to the joshua server daemon

March 15, 2006
Server application

- Daemons act as transparent gateway for user commands
- Queue commands are received globally, executed locally
- Output is redirected to the client
- All joshua daemons run in virtual synchrony provided by transis
- Daemons form a group of head nodes
- Using group communication the group reconfigures itself in cases such as
  - Daemon joins
  - Daemon failure
Cluster mutex

- Distributed mutual exclusion across all cluster nodes
- Realized in connection with virtual synchrony
- Executor sends request for execution
- First gets the permission to run the job
- All other are put on hold
- When job is finished the pending executors will be released
- Solves the single instance problem

March 15, 2006
- Reasonable cost for HA depends from number of head nodes
- Submission performance in linear scale
- Joshua handles parallel job submission, join and fail events as proposed
- Jmutex handles single instance execution problem
- Transis has some stability issues
- PSB is still focused on client-server model

March 15, 2006
Possible merge with HA-OSCAR project
Integration and development of more reliable group communication system
Use of more suitable external resource management system
Add missing job manipulation features (hold, release)
- Take advantage of multi-headed architecture and the symmetric Active/Active HA design
- Prototype implementation could prove design to improve the cluster HA
- System availability is scalable as proposed
- Approach leads to a significant increase of availability of cluster head nodes
- First step toward non-stop computation by introducing a fault tolerant job scheduler