Moab Workload Manager on Cray XT3

Don Maxwell (ORNL)
Michael Jackson (Cluster Resources, Inc.)
MOAB Workload Manager on Cray XT3

- Why MOAB?
- Requirements
- Features
- Support/Futures
Why Moab?

- Existing platforms
  - Cray X1E/XT3 and SGI Altix running PBSPro
  - Opteron-based visualization cluster running SLURM
  - IBM Power 4 running LoadLeveler

- Traditional resource management features missing
  - Dynamic Backfill for high resource utilization
  - “Run this Job Next” (e.g. Ilfavorjob)
  - Diagnostics too limited (e.g. determine “topdog”)

- More flexible scheduling for NLCF resources
  - Integration with resource allocation system (RATS)
  - Dynamic prioritization of jobs

- Reputation of Cluster Resources, Inc.
  - Developers of leading scheduling systems Maui and Moab
  - New ORNL staff with prior experience with Cluster Resources
Cluster Resources’ Solution Space

- Cluster (Intelligence and Orchestration)
  - Workload Management
  - Resource Management
  - Allocation Management
  - Event and Condition Management

- Grid
  - Grid Workload Management
  - Grid Job Submission (Grid Portal)
  - Grid Allocation Management

- Utility/Hosted Computing

- Services
  - Support
  - Consulting (Solution Design, Optimization, etc.)
  - Integration
  - Development
  - Training
Solution Framework: Where Does it Fit?
Integration Points:
Moab – XT3 Integration at ORNL

ORNL Environment

Identity Manager
(Security, Policies, Prioritization, etc.)

Other Services
(Grid & Peer Services, Allocation Manager, Etc.)

Virtualization Interface

OS & Middleware

Cray XT3

Operating System

XTAdmin Database
(Node, resource and file system allocation and statistics)

Resource Manager
(PBS Pro, LSF, TORQUE, etc.)

File System (e.g., Lustre)

Moab

Moab Workload Manager

Native Resource Manager Interface

Node Monitor
Job Monitor
Job Start
Job Cancel
Job Requeue
Other Job Commands …

Config File

• Holistic Intelligence
• Multi-resource Manager
• Tunable Prioritization
• Fault Protection (file system, rogue processes, bad nodes, etc.)

• Event Policy Engine
• Advanced Reporting
• Job Control
• Co-allocation

Moab Cluster Manager
(GUI – Manage, Monitor & Report)

Moab Access Portal
(Web-base Job Management)

Administrators manage via CLI or GUI.

Managers get reports.

End Users apply existing scripts via RM.

Compute Nodes
yod Nodes
Login Nodes
Requirements

- Port Moab to XT3, deploy it, integrate it with 3 tools, test it and go production in 3 Weeks!

- No disruption of production work
  - No change in the interface to the batch system on NLCF machines
  - Monitor and Simulation Mode allows evaluation of priority changes without impacting the current job queue

- Dynamic Backfill
  - Static backfill (a.k.a. FIFO backfill) = underutilization
  - Cathy Scheduler

- More control over workload to meet objectives better
  - “run-this-job-next”

- More flexible prioritization supporting fairshare

- Preemption for visualization workload
Moab Features for a System Administrator

- **Monitor and Simulation mode**
  - Installing new versions
  - Changing policies
Moab: Safe Evaluation and Deployment

- **Monitor Mode**
  - Moab monitors all information, processes and policies, decides what it would do, but does not implement it.

- **Simulation Mode**
  - Moab imports historical or scenario based information and allows offline evaluation of changes to resources, workload and policies.

- **Interactive Mode**
  - Moab fully schedules workload, but asks administrator to approve each decision before implementing it.

- **Partition Mode**
  - Moab applies full workload management to a subset of resources, all other resources are unaffected. Also allows for mixed production and test environments.

- **Normal Mode**
  - Normal production mode with full capabilities.
Moab Features for a System Administrator

- **Monitor and Simulation mode**
  - Installing new versions
  - Changing policies

- **Diagnostics**
  - Load the scheduler is placing on the server
  - Accounts and associated attributes (priorities, QOS, etc.)
  - Job priorities
  - Prior failures and reasons
  - Configuration problems
Moab Features for a System Administrator

- **Monitor and Simulation mode**
  - Installing new versions
  - Changing policies

- **Diagnostics**
  - Load the scheduler is placing on the server
  - Accounts and associated attributes (priorities, QOS, etc.)
  - Job priorities
  - Prior failures and reasons
  - Configuration problems

- **Admin levels**
  - User support could see all and influence jobs
  - Operators could see all but not change anything
Moab Features for a System Administrator

- **Monitor and Simulation mode**
  - Installing new versions
  - Changing policies

- **Diagnostics**
  - Load the scheduler is placing on the server
  - Accounts and associated attributes (priorities, QOS, etc.)
  - Job priorities
  - Prior failures and reasons
  - Configuration problems

- **Admin levels**
  - User support could see all and influence jobs
  - Operators could see all but not change anything

- **SSH X11 forwarding**
  - Multiple login and yod nodes complicated with batch
  - Moab perl interface (Virtualization Layer)
    - Easy placement of jobs
    - Knowledge of submitting host and interactive state retrieved from PBS Pro
Moab – SSH X11 Interactive Sessions

1. End User submits an Interactive Job

2. Moab sees that it is Interactive

3. Moab gets host information of the node that submitted the job

4. Moab ensures the yod service and X11 applications are started where the SSH tunnel was created.

End Users apply existing scripts via RM.
Moab Features for a System Administrator

- **Monitor and Simulation mode**
  - Installing new versions
  - Changing policies

- **Diagnostics**
  - Load the scheduler is placing on the server
  - Accounts and associated attributes (priorities, QOS, etc.)
  - Job priorities
  - Prior failures and reasons
  - Configuration problems

- **Admin levels**
  - User support could see all and influence jobs
  - Operators could see all but not change anything

- **SSH X11 forwarding**
  - Multiple login and yod nodes complicated with batch
  - Moab perl interface (Virtualization Layer)
    - Easy placement of jobs
    - Knowledge of submitting host and interactive state retrieved from PBS Pro

- **Discovered Lustre node recovery state missing in CPA and PBS Pro**
  - Diagnostics revealed jobs were failing
  - Moab modified to consult Lustre recovery table in SDB (database)
Non Moab – Lustre Clean Up Experience

1. End User submits job
2. Resource Manager starts job on XT3
3. After job completes Lustre begins its file lock recovery process.
4. Resource manager and XTAdmin Database report nodes as available.
5. Lustre continues recovery process.
6. Jobs are started on these nodes, but fail, because the nodes are not truly available.
7. Scheduling decisions are made improperly and backfill does not work correctly.

End Users apply existing scripts via RM.

Moab

Node Monitor
Job Monitor
Job Start
Job Cancel
Job Requeue
Other Job Commands...

Native Resource Manager Interface

Native Resource Manager

Config File

• Holistic Intelligence
• Multi-resource Manager
• Tunable Prioritization
• Fault Protection (file system, rogue clients, bad nodes, etc.)

Load Manager

• Event Policy Engine
• Advanced Reporting
• Job Control
• Co-allocation

Access Portal

(GUI – Manage, Nodes)

Virtualization Interface

Cray XT3

Utilization Loss

Compute Nodes

Nodes

Operating System

XTAdmin Database

(Node, resource and file system allocation and statistics)

Resource Manager

(PBS Pro, LSF, TORQUE, etc.)

File System (e.g. Lustre)

ORNL Environment
Moab – Lustre Clean Up Experience

1. End User submits job
   - Moab Cluster Manager (GUI – Manage, Monitor & Report)
   - Moab Workload Manager
   - Native Resource Manager Interface
   - XTAdmin Database
     - Operating System
     - File System (e.g., Lustre)
     - Login Nodes
   - Compute Nodes
   - yod Nodes

2. Moab tells resource manager to start job on XT3

3. After job completes Lustre begins its file lock recovery process.

4. Moab checks job status.

5. Moab checks status of Lustre recovery to see if the nodes are truly available.

6. Moab sees that Lustre is still cleaning up and schedules and backfills jobs properly around the nodes.

7. Once Lustre status is back to normal, additional jobs are applied to the nodes.
Rogue Process Issues

1. End User submits job

2. Resource Manager starts job on XT3

3. Resource or other failure causes job to fail.

4. Resource manager believes jobs are still running and that nodes are busy, thus blocking nodes.

5. Failure condition repeats causing additional blockage.

End Users apply existing scripts via RM.

Moab

- Holistic Intelligence
- Multi-resource Manager
- Tunable Prioritization
- Fault Protection (file system, rogue processes, bad nodes, etc.)

Resource Manager

- Event Policy Engine
- Advanced Reporting
- Job Control
- Co-allocation

XTAdmin Database

- Node, resource and file system allocation and statistics

Operating System

Native Resource Manager Interface

Node Monitor
Job Monitor
Job Start
Job Cancel
Job Requeue
Other Job Commands...

Config File
Moab – Rogue Process Recovery

1. End User submits job

2. Moab tells resource manager to send job to XT3

3. Resource or other failure causes jobs to fail.

End Users apply existing scripts via RM.

Administrators

1. Moab issues hold status on impacted jobs.

2. Moab checks job status.

3. Moab checks status of yod service.


5. Moab checks job status.


7. Moab associates issue information to the job

8. Moab is able to send email notice of job issue to Admin

9. Moab realizes nodes can now be used and additional jobs are applied to the nodes.

ORNL Environment

Good Recovery

Operating System

Resource Manager

(yod Nodes)

Login Nodes

File System

XTAdmin Database

(Node, resource, and file system allocation and statistics)

Job

Monitor

Job

Monitor

Job

Start

Cancel

Job

Request

Other Job Commands...

Config File

Identity Manager (Security, Policies, Prioritization, etc.)

Other Services (Grid & Peer Services, Allocation Manager, Etc.)

Moab Cluster Manager (UI – Manage, Monitor & Report)

Moab Access Control (Web-based Job Management)

Node, resource and file

storage, etc.)

571x738
19

183169476141672431017277277483800518336603297457173578656851858136729408917261240676864695146247101846491489269760121178530499151317910266989281262427764983262883573043194288888852775367081984x271680389672493440754452543831401989630537120784870209184440305714711025961918806775156779631307212970758994448799059407304635728634580539120083480772430535751675483379625130187455242015080448

4991562979862215068319081543580407087127378885722574702552154816008597893587881744832184234548252205801727519289276241839696005646422146428840859928373852139780355549932247269311176723784531968x66938824116329828383112884248645355884346639464357031449638564636052451814525025066471121310054682530377162900323696012546586454249441052975265546196192748782280228255103043603846023484538880

4319252182346148091083988123104084679701869920535775083144728937691016797129469189148307085140212923483218271359906920693077261305456998083354726750124141544424808864435702053263205990970425344x369268605464636218922376147380229836909358510544568703937259110346835576915328978768327534817349847777248017023572419289010525705927824588347587481966410536786714564974612412206520179038879744
## ORNL’s Moab Priority Implementation

<table>
<thead>
<tr>
<th>Factor</th>
<th>Unit of Weight</th>
<th>Actual Weight (Minutes)</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality of Service</td>
<td># of days</td>
<td>1440</td>
<td>High (7)</td>
</tr>
<tr>
<td>Account Priority</td>
<td># of days</td>
<td>1440</td>
<td>Allocated Hours (0)</td>
</tr>
<tr>
<td>Queue</td>
<td># of days</td>
<td>1440</td>
<td>Debug (5)</td>
</tr>
<tr>
<td>Job Size</td>
<td>1 day / 1000cpu</td>
<td>1</td>
<td>Provided by Moab</td>
</tr>
<tr>
<td>Queue Time</td>
<td>1 minute</td>
<td>1</td>
<td>Provided by Moab</td>
</tr>
</tbody>
</table>

The resulting priority is a simple calculation.

\[
1440 \times 5 + 2 + 96 = 7298
\]
Moab’s Flexibility w/ Multi-Factor Prioritization

- Fairshare
  - User
  - Group
  - Account
  - Class
  - Quality of Service
  - Jobs per User
  - Processor Seconds Per User
  - Processors Per User

- Resources
  - Node
  - Disk
  - Processor
  - Memory
  - Swap
  - Processor Seconds
  - Processor Equivalent
  - Walltime

- Usage
  - Consumed
  - Remaining
  - Percentage Consumed

- Service Levels
  - Queue Time
  - XFactor
  - Policy Violation
  - ByPass

- Target Levels
  - Queue Time
  - XFactor

- Credential Based
  - User
  - Group
  - Account
  - Class
  - Quality of Service

- Attribute
- State
- And More….
Moab Features for an End User

- **showbf**
  - Determine the size and length of a job that will backfill at any instance in time

- **checkjob**
  - “Why is my job not starting?”
  - Has it tried to run and failed due to a system problem?

- **showstart**
  - An estimated start and end time for jobs currently sitting in the queue
Moab Features for Management

- Difficult to Visualize Cluster
  - Usage
  - Performance
  - Historical Data (Big Picture and Capacity Planning)
Moab Reports

- Service Monitoring and Management
Support/Future

- Port for XT3 completed in about three weeks
  - X1E ported and running monitor mode currently

- Quick turnaround for bugs and features

- Next step issue resolution or feature requests
  - Job with top priority sometimes fails
    - Feels like CPA timing problem (CPA_NO NODES)
    - New feature for multi-dimensional MAXIJOB
Conclusion

- Rollout has gone very well on multiple architectures (XT3, X1E, Altix, Opteron), OSs, Resource Managers (PBS Pro, SLURM), and Interconnects (XT3, Crossbar, Quadrics, Numalink)

- ORNL policies are now properly represented and enforced
- Admin staff time is reduced
- Utilization is increased
- Progress on future projects has accelerated
- Users are happier

- More science is being delivered!
Questions?

Or Contact Us Afterwards

•Don Maxwell:
  www.ornl.gov
  maxwellde@ornl.gov

•Michael Jackson:
  www.clusterresources.com
  michael@clusterresources.com
  +1 (801) 873-3400