Bright Cluster Manager & SLURM
Maximum Productivity, Minimum Complexity

Robert Stober
Systems Engineer
About Bright Computing

1. Develops and supports Bright Cluster Manager for HPC systems and server farms
2. Incorporated in USA (HQ in San Jose, California)
3. Backed by ING Bank as shareholder and investor
4. Sells through a rapidly growing network of resellers and OEMs world-wide
5. Customers and resellers in US, Canada, Brazil, Europe, Middle-East, India, Singapore, Japan, China
6. Installations in Academia, Government, Industry, ranging from 4 node to TOP500 systems
## Customers

### Academia
- CUNY: The City University of New York
- UH: University of Houston
- Drexel University
- Hong Kong Polytechnic University
- Georgia Institute of Technology
- Virginia Tech
- University of Oxford
- University of Cambridge
- EPFL: École Polytechnique Fédérale de Lausanne
- Technical University of Dresden
- University of Groningen

### Government
- Fermilab
- NERSC
- Sandia National Laboratories
- PSS Observatory
- Proudman Oceanographic Laboratory
- Office of Renewable Energy
- SINTEF

### Industry
- Boeing
- Saudi Aramco
- PACCAR
- ING
- REGENERON
- GL
- Vattenfall
- Delphi
- Zitrón
- vonRoll
- Componenta
- CD-Adapco
The Commonly Used “Toolkit” Approach

- Most HPC cluster management solutions use the “toolkit” approach (Linux distro + tools)
  - Examples: Rocks, PCM, OSCAR, UniCluster, CMU, etc.
  - Tools typically used: Ganglia, Cacti, Nagios, Cfengine, System Imager, xCAT, Puppet, Cobbler, Hobbit, Big Brother, Zabbix, Groundwork, etc.

- Issues with the “toolkit” approach:
  - Tools rarely designed to work together
  - Tools rarely designed for HPC
  - Tools rarely designed to scale
  - Each tool has its own command line interface and GUI
  - Each tool has its own daemon and database
  - Roadmap dependent on developers of the tools

- Making a collection of unrelated tools work together
  - Requires a lot of expertise and scripting
  - Rarely leads to a really easy-to-use and scalable solution
About Bright Cluster Manager

- Bright Cluster Manager takes a much more fundamental & integrated approach
  - Designed and written from the ground up
  - Single cluster management daemon provides all functionality
  - Single, central database for configuration and monitoring data
  - Single CLI and GUI for ALL cluster management functionality

- Which makes Bright Cluster Manager …
  - Extremely easy to use
  - Extremely scalable
  - Secure & reliable
  - Complete
  - Flexible
  - Maintainable
Architecture

Bright Cluster

- CMDaemon
  - procedure call
  - SOAP+SSL
  - event
- node001
  - procedure call
  - SOAP+SSL
  - event
- node002
- node003

Cluster Management

- GUI
- Shell
- Web-Based User Portal
- Third-Party Application
Workload Manager Integration

- Bright supports several workload managers including SLURM
- SLURM is now the default
- Automatic installation
- Automatic configuration
- Pre-job health checks
- Sampling and visualization of workload manager metrics
- Job monitoring and control
- Creation and editing of queues
- Integration of Bright and SLURM failover
Welcome to the Bright Cluster Manager Installer

License Information

- Version: 5.1
- Edition: Advanced
- Name: Bright 5.1 Cluster
- Organization: Bright Computing
- Unit: Development
- Locality: San Jose
- State: California
- Country: US
- Serial: 2158
- Valid from: 15 Aug 2010
- Valid until: 16 Nov 2010
- MAC address: ???:???:???:???:???:???
- Licensed nodes: 512

Installation mode

- Normal (recommended)
- Express

Remote Installation

Cancel  Go Back  Continue
SLURM Installation

SLURM is installed in a shared directory

- Current version is 2.2.4
- /cm/shared is mounted on the nodes by default

```
[root@atom-head1 apps]# cd /cm/shared/apps/slurm/current

[root@atom-head1 current]# ls -l
total 28
drwrxr-xr-x 2 root root 4096 Sep 2 18:32 bin
drwrxr-xr-x 4 root root 4096 Sep 3 02:10 cm
drwrxr-xr-x 2 root root 4096 Sep 3 02:10 etc
drwrxr-xr-x 3 root root 4096 Sep 2 18:32 lib64
drwrxr-xr-x 5 root root 4096 Sep 2 18:32 man
drwrxr-xr-x 2 root root 4096 Sep 2 18:34 sbin
drwrxr-xr-x 4 root root 4096 Sep 2 18:32 share
```
SLURM Server Role

- SGE Client Role
- Torque Client Role
- PBSPro Client Role
- SLURM Client Role
- SLURM Server Role

Scheduler: backfill

- SGE Server Role
- Torque Server Role
- PBSPro Server Role
- LSF Server Role
- LSF Client Role
- Submit Manager Role
- Root Role

EVENT VIEWER

All Events

- Cluster
- Source
- Message

- 20Sep2011 11:52:09 atom atom001 Check Devices Up is in state PASS on atom001
- 20Sep2011 11:50:02 atom atom001 Check Devices Up is in state FAIL on atom001
- 20Sep2011 11:47:00 atom atom001 Service named was restarted on atom001
- 20Sep2011 11:45:00 atom atom-head1 Service named was restarted on atom-head1
SLURM Client Role
SLURM Installation

Portions of the SLURM config files are autogenerated:
- Applies when a role has been assigned to a node
- Example: slurm.conf file

```plaintext
# BEGIN AUTOGENERATED SECTION -- DO NOT REMOVE
# Scheduler
SchedulerType=sched/backfill
# Master nodes
ControlMachine=atom-head1
ControlAddr=atom-head1
# Nodes
NodeName=atom[001-003]
# Partitions
PartitionName=defq Nodes=atom[001-003] Default=YES MinNodes=1 MaxNodes=UNLIMITED MaxTime=UNLIMITED AllowGroups=ALL Priority=1 DisableRootJobs=NO RootOnly=NO Hidden=NO Shared=NO
# END AUTOGENERATED SECTION -- DO NOT REMOVE
```
Bright Monitoring Framework

- Actions

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drain node</td>
<td>Remove a node from further use by the scheduler. Jobs running will be killed.</td>
<td>&lt;built-in&gt;</td>
</tr>
<tr>
<td>killprocess</td>
<td>Action which kills processes of pids found in STDIN.</td>
<td>/cm/local/apps/cmd/scripts/actions/killprocess.pl</td>
</tr>
<tr>
<td>Power off</td>
<td>Power off the device</td>
<td>&lt;built-in&gt;</td>
</tr>
<tr>
<td>Power on</td>
<td>Power on the device</td>
<td>&lt;built-in&gt;</td>
</tr>
<tr>
<td>Power reset</td>
<td>Power reset the device</td>
<td>&lt;built-in&gt;</td>
</tr>
<tr>
<td>Reboot</td>
<td>Reboot the node</td>
<td>&lt;built-in&gt;</td>
</tr>
<tr>
<td>remount</td>
<td>action which tries to fix broken fs mounts, e.g., when device is not mounted...</td>
<td>/cm/local/apps/cmd/scripts/actions/remount</td>
</tr>
<tr>
<td>SendEmail</td>
<td>Send an email to the address specified by the parameter in the monitorin...</td>
<td>&lt;built-in&gt;</td>
</tr>
<tr>
<td>Shutdown</td>
<td>Shutdown the node</td>
<td>&lt;built-in&gt;</td>
</tr>
<tr>
<td>testaction</td>
<td>action to test, it generates output in a file for e.g., debugging</td>
<td>/cm/local/apps/cmd/scripts/actions/testaction</td>
</tr>
<tr>
<td>Undertain node</td>
<td>Enable a node to start running jobs for the scheduler</td>
<td>&lt;built-in&gt;</td>
</tr>
</tbody>
</table>
Bright Monitoring Framework

- Health Checks
**Bright Monitoring Framework**

- **Metrics**

![Monitoring Configuration](image)

<table>
<thead>
<tr>
<th>RESOURCES</th>
<th>Monitoring Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>NODES</td>
<td></td>
</tr>
<tr>
<td>NODES Categories</td>
<td></td>
</tr>
<tr>
<td>default</td>
<td></td>
</tr>
<tr>
<td>Head Nodes</td>
<td></td>
</tr>
<tr>
<td>atom-head1</td>
<td></td>
</tr>
<tr>
<td>RACKS</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DGe3-3200</td>
<td></td>
</tr>
<tr>
<td>atom-head1</td>
<td></td>
</tr>
<tr>
<td>atom001</td>
<td></td>
</tr>
<tr>
<td>atom002</td>
<td></td>
</tr>
<tr>
<td>atom003</td>
<td></td>
</tr>
<tr>
<td>CHASSIS</td>
<td></td>
</tr>
<tr>
<td>Virtual SMP Nodes</td>
<td></td>
</tr>
<tr>
<td>NODES</td>
<td></td>
</tr>
<tr>
<td>atom001</td>
<td></td>
</tr>
<tr>
<td>atom002</td>
<td></td>
</tr>
<tr>
<td>atom003</td>
<td></td>
</tr>
<tr>
<td>GPU Units</td>
<td></td>
</tr>
<tr>
<td>OTHER DEVICES</td>
<td></td>
</tr>
<tr>
<td>CPU Idle</td>
<td></td>
</tr>
<tr>
<td>CPU Utilization</td>
<td></td>
</tr>
<tr>
<td>CPU Memory</td>
<td></td>
</tr>
<tr>
<td>EVENT VIEWER</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>All Events</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>Cluster</td>
<td>Source</td>
</tr>
<tr>
<td>20/09/2011 11:55:02</td>
<td>atom</td>
<td>atom001</td>
</tr>
<tr>
<td>20/09/2011 11:55:02</td>
<td>atom</td>
<td>atom001</td>
</tr>
<tr>
<td>20/09/2011 11:47:00</td>
<td>atom</td>
<td>atom-head1</td>
</tr>
<tr>
<td>20/09/2011 11:45:56</td>
<td>atom</td>
<td>atom-head1</td>
</tr>
<tr>
<td>20/09/2011 11:45:20</td>
<td>atom</td>
<td>Unknown</td>
</tr>
</tbody>
</table>
Bright collects the following basic workload manager metrics

- Running jobs
- Failed jobs
- Queued jobs
- **Average expansion factor**: This is by what factor, on average, jobs took longer to run than expected. The expectation is according to heuristics based on duration in past and current job queues, as well as node availability
- **Estimated delay**: Estimated Delay to execute jobs
- **Average run time (per queue)**
The Black Hole Node Syndrome
Pre-Job Health Checks

- Any Bright health check can be configured as a pre-job health check
Pre-Job Health Checks

- Bright prolog script (cmprolog) is configured as the SLURM prolog script
- The prolog script calls all the configured pre-job health checks
- It instructs the cmdaemon on the execution node, to execute all pre-job health checks.
- If any of the pre-job health checks fail
  - The cmprolog script exits with code 99
  - The node is drained
    `scontrol state=DRAIN NodeName=node001`
  - Administrator is notified of the failure.
### Monitoring Configuration

#### Metric Configuration

<table>
<thead>
<tr>
<th>Category</th>
<th>Item</th>
<th>Parameter</th>
<th>Threshold Bound</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Master Nodes</td>
<td>FreeSpace</td>
<td>/</td>
<td>&lt; 30 GB</td>
<td>NotifyVendor</td>
</tr>
<tr>
<td>All Master Nodes</td>
<td>FreeSpace</td>
<td>/</td>
<td>&lt; 30 GB</td>
<td>SendEmail</td>
</tr>
<tr>
<td>All Master Nodes</td>
<td>FreeSpace</td>
<td>/home</td>
<td>&lt; 30 GB</td>
<td>NotifyVendor</td>
</tr>
<tr>
<td>All Power Distribution</td>
<td>PDU.load</td>
<td>/</td>
<td>&gt; 32 A</td>
<td>SendEmail</td>
</tr>
<tr>
<td>slave</td>
<td>Temperature</td>
<td></td>
<td>&gt; 70</td>
<td>SendEmail</td>
</tr>
<tr>
<td>slave</td>
<td>Temperature</td>
<td></td>
<td>&gt; 70</td>
<td>Shutdown</td>
</tr>
</tbody>
</table>

#### Monitoring Rules Wizard

Select Category:
- All Power Distribution Units
- All Ethernet Switches
- All Myrinet Switches
- All IB Switches
- All Master Nodes
- All Rack Sensors
- All Generic Devices

- [Cancel](#)
- [Previous](#)
- [Next](#)
SLURM Configuration

SLURM Failover

- The SLURM failover role becomes available when a Bright failover node is configured
- When the failover occurs the SLURM DbdHost is changed to the secondary head node

Bright provides the capability to

- Monitor, kill, suspend, resume, hold and release jobs
- Add, remove and edit queues
- View, drain and undrain nodes
<table>
<thead>
<tr>
<th>Job</th>
<th>Name</th>
<th>Scheduler</th>
<th>User</th>
<th>Queue</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>fluent</td>
<td>torque</td>
<td>jodi</td>
<td>medium.q</td>
<td>queued</td>
<td></td>
</tr>
<tr>
<td>fluent</td>
<td>torque</td>
<td>jodi</td>
<td>medium.q</td>
<td>queued</td>
<td></td>
</tr>
<tr>
<td>fluent</td>
<td>torque</td>
<td>jodi</td>
<td>medium.q</td>
<td>queued</td>
<td></td>
</tr>
<tr>
<td>gromacs</td>
<td>torque</td>
<td>alex</td>
<td>long.q</td>
<td>running</td>
<td></td>
</tr>
<tr>
<td>gromacs</td>
<td>torque</td>
<td>alex</td>
<td>long.q</td>
<td>running</td>
<td></td>
</tr>
<tr>
<td>gromacs</td>
<td>torque</td>
<td>alex</td>
<td>long.q</td>
<td>running</td>
<td></td>
</tr>
<tr>
<td>gromacs</td>
<td>torque</td>
<td>alex</td>
<td>long.q</td>
<td>running</td>
<td></td>
</tr>
<tr>
<td>gromacs</td>
<td>torque</td>
<td>alex</td>
<td>medium.q</td>
<td>queued</td>
<td></td>
</tr>
<tr>
<td>hpcc</td>
<td>torque</td>
<td>kate</td>
<td>long.q</td>
<td>queued</td>
<td></td>
</tr>
<tr>
<td>hpcc</td>
<td>torque</td>
<td>kate</td>
<td>long.q</td>
<td>queued</td>
<td></td>
</tr>
<tr>
<td>hpcc</td>
<td>torque</td>
<td>kate</td>
<td>long.q</td>
<td>queued</td>
<td></td>
</tr>
<tr>
<td>magmasteel</td>
<td>torque</td>
<td>james</td>
<td>medium.q</td>
<td>queued</td>
<td></td>
</tr>
<tr>
<td>magmasteel</td>
<td>torque</td>
<td>james</td>
<td>medium.q</td>
<td>queued</td>
<td></td>
</tr>
<tr>
<td>magmasteel</td>
<td>torque</td>
<td>james</td>
<td>medium.q</td>
<td>queued</td>
<td></td>
</tr>
<tr>
<td>magmasteel</td>
<td>torque</td>
<td>james</td>
<td>medium.q</td>
<td>queued</td>
<td></td>
</tr>
<tr>
<td>xhpi</td>
<td>torque</td>
<td>matthew</td>
<td>short.q</td>
<td>running</td>
<td></td>
</tr>
<tr>
<td>xhpi</td>
<td>torque</td>
<td>matthew</td>
<td>short.q</td>
<td>running</td>
<td></td>
</tr>
<tr>
<td>xhpi</td>
<td>torque</td>
<td>matthew</td>
<td>short.q</td>
<td>running</td>
<td></td>
</tr>
<tr>
<td>xhpi</td>
<td>torque</td>
<td>matthew</td>
<td>short.q</td>
<td>running</td>
<td></td>
</tr>
</tbody>
</table>
SLURM Configuration

- Example of editing an existing queue
Questions?

Robert Stober
robert.stober@brightcomputing.com
+1 209 986 9298