Evaluation of Bright Cluster Manager - Advanced Edition

Hussein N. El-Harake, Roberto Aielli, Thomas Schoenemeyer
Swiss National Supercomputing Centre (CSCS)
{hussein, aielli, thomas.schoenemeyer}@cscs.ch

Abstract: We evaluated the latest version 6.1. of Bright Cluster Manager on a small test cluster with one head-node and 20 compute nodes at CSCS. This study covers the functionality, scalability, capability, ease of use, health management and other essential requirements in HPC cluster management.

This evaluation study includes the test of the recently added Management and Monitoring features for Nvidia GPUs and Intel Xeon Phi.

Bright Cluster Manager matches our requirements to manage different accelerators and coprocessors in compute nodes. From our perspective it is an appropriate tool, if there are only one or two system managers in a computer center or an organization with several HPC systems to be managed.

1 Introduction

A wide variety of cluster tools are available today to help people get started provisioning, managing and monitoring HPC clusters. The tools differ in complexity, costs, supported devices and Linux distributions, provisioning methods, scalability, supported job schedulers and third party software add-ons.

There are currently three categories of Cluster Management Tools available:

The first category includes management solutions provided by vendors together with their own hardware. Examples within this group are

- HP Insight Cluster Management Utility [1]
- IBM Cluster Manager [2]
- NEC LXC3 [3]
- Megware ClustWare [4]
- IBM Platform HPC [5]

The second group consists of commercial Management and Provisioning tools that can be deployed on any HPC Cluster such as

- Bright Cluster Manager Standard and Advanced Edition [6]
- Puppet Enterprise Edition for more than 10 nodes [7]

A collection of freely available tools falls in the third category. Examples are

- Puppet Open Source and Puppet Enterprise (for less than 10 nodes) [8]
- Perceus Provisioning System [9]
- Pacemaker [10]
- Warewulf [11]
- xCat [12]

Comparison studies have been provided by Trangoni et al [13], where the authors compared various provisioning systems, according to their evaluation criteria, and Bright Cluster Manager received the highest score.

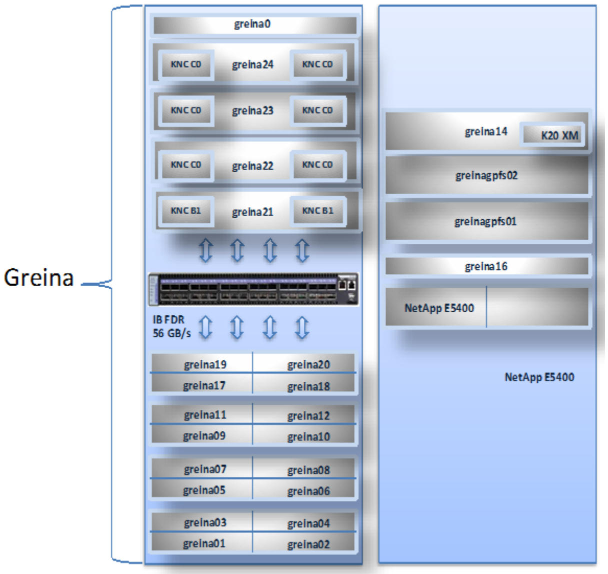
We also decided to evaluate Bright Cluster Manager Advanced Edition because of the new GPU and coprocessor support. CSCS has been investigating accelerator technologies for a few years now a number of GPUs have been integrated in production systems already.

For our study we used a small test cluster at CSCS as shown in Figure 1. The system consists of one head-node and 20 compute nodes.

Sixteen of the compute nodes each have two E5-2670 processors, four nodes have two E5-2650 v2 (IvyBridge 10 cores) connected to one InfiniBand FDR switch.

Of the 16 nodes, 4 nodes have two Intel Xeon Phi cards each and one node has one Nvidia Tesla K20X GPU.

A small GPFS parallel file system using one NetApp E5400 controller is integrated in the cluster.



Greina Cluster Figure 1

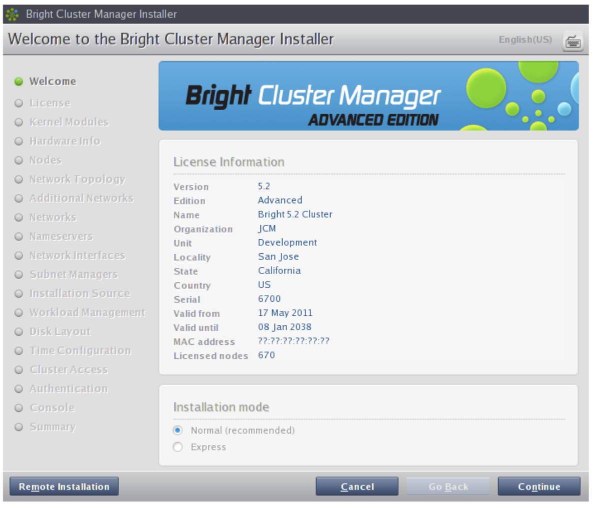
There are two different versions of the Bright Cluster Manager, standard and advanced. One of the main differences is the accelerator and coprocessor support.

We tested successfully RedHat Enterprise Linux (RHEL) versions 6.2, 6.3, 6.4 and Scientific Linux 6.1, 6.2, 6.3 and 6.4. According to the vendor information Bright Cluster Manager also supports SUSE Linux Enterprise Server (SLES) and CentOS, which was not part of this study.

2 Installation & Configuration

Bright Cluster Manager can be applied to a cluster with a preinstalled operating system on the head node. It is also possible to start from a clean head node by using the operating system provided with the Bright ISO image.

Bright Computing provides an administrator manual including an appendix with a quick start installation guide [14]. The quick start procedure is sufficient for experienced administrators and will allow having a system of our size up and running in less than four hours. In case of a preinstalled Operating System on the head node, you must verify the OS compatibility (Admin manual Section 1.1). It provides a welcome screen that displays core information about the version and edition as well as the license agreement (Figure 2). One can choose between an express or normal installation of which is recommended.



Welcome to the Bright Cluster Manager Installer

License Information

Version	5.2
Edition	Advanced
Name	Bright 5.2 Cluster
Organization	JCM
Unit	Development
Locality	San Jose
State	California
Country	US
Serial	6700
Valid from	17 May 2011
Valid until	08 Jan 2038
MAC address	??:??:??:??:??:??
Licensed nodes	670

Installation mode

● Normal (recommended)
○ Express

Bright Cluster Installer Welcome Screen Figure 2

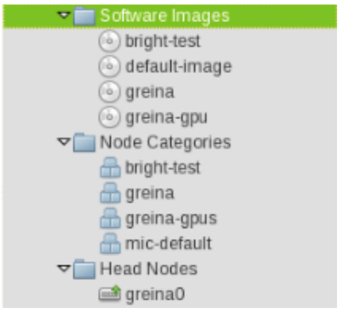
After the license screen, the Kernel Modules screen shows the kernel modules list generated by probing system hardware. The next step is the "Nodes screen" where the number of racks, nodes, name of node etc. are specified. This is followed by the "Network Topology" selection screen and allows for the selection of three different network topologies. Most of the configuration parameters can be adjusted using *cmgui* or *cmsh* after the initial installation.

Cmgui and *Cmsh* are tools used to manage the CMDaemon on Bright Cluster manager. As a graphical interface *Cmgui* provides easier access to different manager sections like dashboard, visualization, monitoring and event viewer. *Cmgui* offers closely equivalent capabilities to *Cmsh*, from our perspective *Cmgui* is more user friendly than *Cmsh* especially for new Bright Cluster Manager users.

Cmsh is the CLI provided with Bright Cluster Manager. By default it connects to the localhost. *Cmsh* allows sysadmins to manage resources faster than *Cmgui*. The power of *Cmsh* is the scripting part. It's possible to script it by submitting a set of commands from the shell. Both management tools allow you to configure services without the need of knowing them in depth.

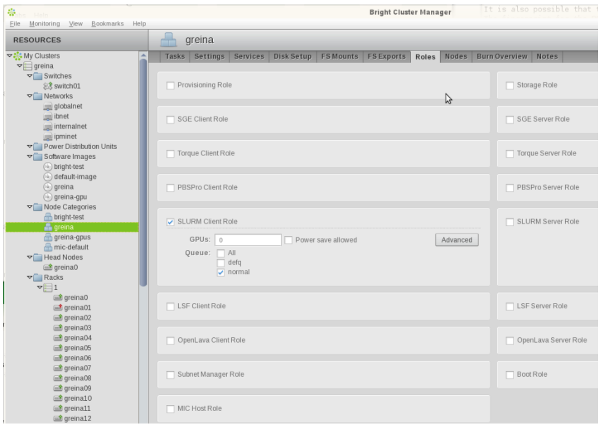
3 Software Images

The software image is a shared directory, which contains a complete Linux OS and provisioned to the compute nodes during boot. The default Linux version that comes with Bright is the same version used by the head node. It is always possible to create a new software image based on a different OS. We successfully tested the latest versions of Redhat 6.4 and Scientific Linux. The *cmgui* provides a screen to manage software images easily (Figure 4).



Software Image Figure 4

Different images and categories allowed us to create a heterogeneous cluster with different sets of hardware, accelerators and configurations. Node categories (Figure 5) comprise configurations and settings shared between groups of similar nodes; changes applied to a category will be applied on all nodes that belong to this category. Every setting under category could also be configured at node level. In case of mismatch between node category settings and individual node settings, node category settings will be used automatically.



□ Provisioning Role
□ Storage Role
□ SGE Client Role
□ SGE Server Role
□ Torque Client Role
□ Torque Server Role
□ PBSPro Client Role
□ PBSPro Server Role
☑ SLURM Client Role
GPUs: 0 □ Power save allowed
Queue: □ All □ defq ☑ normal
□ SLURM Server Role
□ LSF Client Role
□ LSF Server Role
□ OpenLava Client Role
□ OpenLava Server Role
□ Subnet Manager Role
□ Boot Role
□ MIC Host Role

Node Categories (Figure 5)

Node category includes roles, services, disk setup etc. Creating multiple software images and multiple categories is helpful in an heterogeneous environment as mentioned above but the necessity decreases in an homogenous environment.

3 Health Check and Monitoring

Bright Cluster Manager provides a health check manager; it is based on scripts and located under: /cm/local/apps/cmd/scripts/healthschecks/

These set of functions verify the functionality of services, reporting hardware problems such as faulty nodes, verification of mounted filesystem or checking the authentication system etc.. The health check helps keeping the downtime as short as possible; any intervention could be done immediately based on dashboard alerts, emails and logs.

It's also possible to set an action when health check reports any of the following cases: failures pass, unknown or state flapping. There is a complete list of defined actions available like drain nodes, reboot, reset, send email, kill process and remount file systems etc.

Bright Cluster Manager offers a mature monitoring framework for existing resources. It allows monitoring current or past problems and collects trends that help the administrator to predict prospective issues. It can trigger alerts when certain thresholds are exceeded and can also launch an immediate action.

Cmgui is used to access monitoring resources, to set and create monitoring graphs covering the CPU, disks, networks, batch system, failures, utilization, health checks and power information, if provided by the PDUs.

A dashboard conveys the most important relevant information at a glance and draws attention to items that are abnormal (Figure 6).

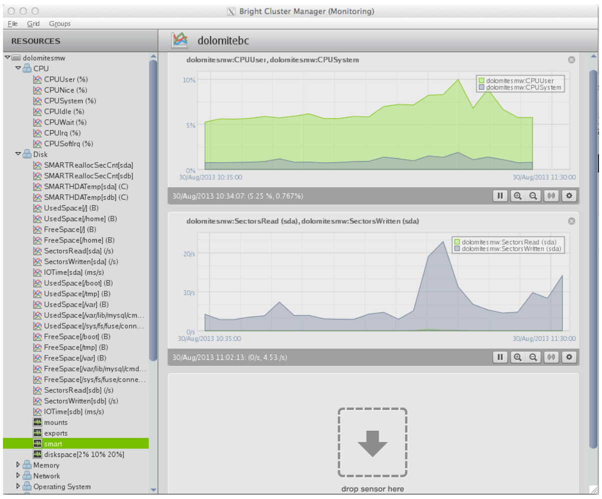


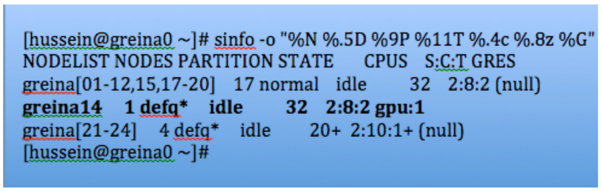
Figure 6

All Bright Cluster monitoring details are fully described in Chapter 11 of the Administration Manual.

4 Accelerator and Coprocessor Support

Bright Cluster Manager supports both Nvidia GPUs and Intel Xeon Phi. We successfully tested the Nvidia Tesla K20X GPU and the Intel Xeon Phi 5120D.

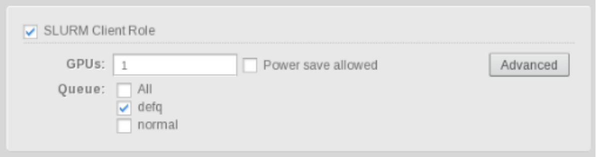
Bright provides two methods to access the Tesla K20s, interactively or through scheduling system; we used only SLURM Workload Manager in our tests. CUDA 5.5 and the generic driver packages were also provided.



```
[hussein@greina0 ~]# sinfo -o "%N %.5D %9P %11T %.4c %.8z %G"
NODELIST NODES PARTITION STATE      CPUS   S:C:T GRES
greina[01-12,15,17-20]   17 normal   idle        32   2:8:2 (null)
greina14    1 defq*    idle        32   2:8:2 gpu:1
greina[21-24]    4 defq*    idle        20+  2:10:1+ (null)
[hussein@greina0 ~]#
```

Query GPU via Slurm Figure 7

The output of the sinfo command (figure 7) shows that there is one GPU attached to node greina14; it uses the 'defq' queue and it is in IDLE state.



☑ SLURM Client Role

GPUs: 1 ☐ Power save allowed

Queue: ☐ All ☑ defq ☐ normal

Figure 8

The configuration of the GPU is quite easy; we installed through yum the CUDA and driver packages, added the number of GPUs under Node role of greina 14 (figure 8) and finally added the Nvidia GPU as generic resources *type GresTypes=gpu* through *Cmgui* or *Cmsh*. That allowed us to submit jobs to GPU by

adding –gres=gpu

in the batch script.

When we added the Xeon Phi cards to our cluster we were using BCM V6.0, and this version had no Xeon Phi support, so we had to use the Intel software stack and that required some tuning. Now Bright Cluster Manager 6.1 supports Intel Xeon Phi and provides the software stack including SLURM support. There are two ways to configure the Xeon Phi coprocessors (Figure 9), using *cmgui* or through *cm-mic-setup* script. We successfully tested both, *cmgui* created some stability issues. According to Bright Computing, these issues are solved now.

We used NFS as the file-system supported on the Xeon Phi so far. To test the basic functionality of Xeon Phi there are some test jobs available under: /cm/shared/examples/workload/slurm/jobscripts/.

According to Bright Computing, a Lustre client is also available from Intel and could be installed on the Xeon Phi card. This Lustre client was not tested in this study.

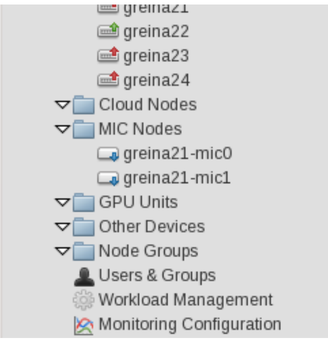


Figure 9: Xeon Phi Nodes

If SLURM is configured with more than one partition, it is important to assign the Xeon Phi host and the Xeon Phi coprocessor to the same logical set of nodes otherwise jobs will fail. It is mandatory to set "mic" as general resource in the slurm configuration with

GresTypes=mic.

Jobs will be allocated to generic resources requested at job time using the *--gres* option like this:

sbatch –gres=mic

By default users cannot access the Xeon Phi devices via ssh due to Bright Cluster Manager policy, a work around of this problem would be to modify */etc/init.d/mpss* by adding group access, or comment the statement that creates the condition. Finally configure the authentication on the Xeon Phi devices to provide interactive authentication.

5 Conclusion

Bright Cluster Manager 6.1 is a mature and reliable management system for HPC cluster and answers our requirements to manage different accelerators and coprocessors in compute nodes.

The list of available health check features is exhaustive and we were not able to validate all of them. Among those we have tested, the results were satisfying. A dashboard conveys the most important relevant information at a glance and draws attention to items that are abnormal.

Bright Cluster Manager is an appropriate tool if there are only one or two system managers in a computer center or an organization with several HPC systems. Assigning images to individual nodes or groups of nodes with a single command or a mouse click makes it very convenient to manage multiple clusters.

Bright Computing provides support for the product, which we believe, justifies the cost. In our case, we bundled the support and the license together, that means if the license expires the product ceases operation. We recently learned, that it is possible to buy perpetual licenses and the support contract separately.

6 Literature

[1] HP Inside Cluster Management Utility, PDF Datasheet , 2012

[2] IBM Cluster Manager, Datasheet, 2012

[3] NEC LXC3 Cluster Manager, Datasheet, 2013

[4] Megware ClustWare, Datasheet, 2013

[5] IBM Platform HPC, Datasheet, 2013

[6] Bright Cluster Manager, Datasheet, 2013

[7] Puppet Enterprise, Datasheet, 2012

[8] Puppet Open Source, Datasheet, 2012

[9] Perceus Provisioning Systems, Datasheet, 2012

[10] Pacemaker, Datasheet, 2012

[11] Warewulf, Website, 2013

[12] xCAT, Website, 2013

[13] Mario Trangoni, Matias Cabral: A comparison of Provision Systems for Beowolf Clusters, pdf, 2012.

[14] Bright Cluster Manager 6.1, Administrator Manual, Sep. 2013