Science & Technology Facilities Council

Technical Report
DL-TR-2010-002

Using Moab Cluster Suite for Cluster Management

DJ Cable

June 2010

Enquiries about copyright, reproduction and requests for additional copies of this report should be addressed to:

Chadwick Library
Science and Technology Facilities Council
Daresbury Laboratory
Daresbury Science and Innovation Campus
Warrington
WA4 4AD

Tel: +44(0)1925 603397
Fax: +44(0)1925 603779
email: library@dl.ac.uk

Science and Technology Facilities Council reports are available online at: http://epubs.stfc.ac.uk/

ISSN 1362-0207

Neither the Council nor the Laboratory accept any responsibility for loss or damage arising from the use of information contained in any of their reports or in any communication about their tests or investigations.

Using Moab Cluster Suite for cluster management

Dave Cable, Distributed Computing Group

Computational Science and Engineering Department, STFC Daresbury Laboratory, Daresbury Science and Innovation Campus, Daresbury, Warrington, Cheshire WA4 4AD

June 2009

Introduction

Moab Cluster Suite (MCS) is a resource scheduler from Cluster Resources Inc[1]. Cluster Resources (CR) claim that MCS can also be used for cluster management purposes, either on a single cluster or a "grid" of clusters. As the System Operations and Distributed Computing groups at Daresbury Laboratory have multiple clusters, each with its own, vendor-dependent, management software it could be advantageous to use just one toolset to manage them all. With that in mind the Distributed Computing Group (DisCo) have conducted an evaluation of the functionality and usefulness of MCS's cluster management capabilities. An underlying assumption is that, since MCS was originally written as a sophisticated job scheduler, and as many HPC-SIG members use it for that purpose, there is no need to investigate its scheduling ability.

MCS

MCS is a set of three software packages that provide job scheduling (including policies, reservations and backfill), cluster management, and remote job submission. It is designed to interface to multiple resource managers including Torque[2], Sun Grid Engine (SGE)[3], IBM LoadLeveler[4] and SLURM[5]. MCS interfaces most closely with Torque since that is also a CR product.

The three components of MCS are:

- Moab Workload Manager (MWM). This is the core component and comprises a mix of binaries and Perl scripts. It is installed from source on the cluster machine.
- Moab Cluster Manager (MCM). This is the remote cluster management GUI and is installed on one or more workstations. It is possible to install it on a cluster head node; however the documentation advises that in this case it must only be accessed from a local terminal, because X forwarding severely impairs performance of the node. The software is provided as a Java .jar file, and therefore requires a functioning Java installation, as well as an MWM target to connect to.
- Moab Access Portal (MAP). This application allows users to submit jobs to a MWM-enabled cluster via a web browser. It must be installed on a suitable server, not the cluster itself. The software is in the form of Java servlets and comes complete with Tomcat[6] to serve these.

Other CR products

These include a grid-enabled version of MCM called Moab Grid Manager, an "adaptive cluster suite" which provisions different operating systems on compute nodes in response to workload, and a usage accounting tool called GOLD. See http://www.clusterresources.com/products.php.

[1] http://www.clusterresources.com
[2] http://www.clusterresources.com/pages/products/torque-resource-manager.php
[3] http://gridengine.sunsource.net/
[4] http://www-03.ibm.com/systems/clusters/software/loadleveler/index.html
[5] https://computing.llnl.gov/linux/slurm/
[6] http://tomcat.apache.org/

Software License

CR initially provided a 45-day evaluation license for Moab, which ran out on May 22nd 2009. Subsequently they delivered a Grid-enabled license for both test clusters with an expiry date of July 15th.

Community Support

Mailman mailing lists for CR products, monitored by CR staff, can be found at: http://www.clusterresources.com/resources/mailing-lists.php

MWM Installation

MWM was installed on two machines with differing configurations as seen in table 1. The available documentation is clear and effective, but is fragmented and occasionally inaccurate. For instance, integration with SGE is not described in the main administrator's guide but is available in a separate document – not listed on the documentation page!

Cluster	Architecture	Resource Manager / Job Scheduler	Installation quirks	Other comments
csecell.dl.ac.uk	ppc64	Torque/Maui	Only 32-bit MWM build available, so Torque had to be recompiled in 32-bit mode to match.	Maui disabled as it is replaced by MWM.
cseem64t.dl.ac.uk	x86_64 (Woodcrest)	SGE	Required changes to SGE config and some Perl coding.	MWM configured to gather information from Ganglia, and to use IPMI.

Table 1: Target systems

MWM is distributed as a (compressed) tarball containing a mix of binaries, libraries, Perl scripts and other text files. The installation process is slightly confusing, since it requires a "`./configure ...; make install`" procedure as if it were a source-code compilation. The configure script will quietly accept any valid option, even if that option is precluded or negated by the particular binary build chosen. For example, "`./configure –with-sge`" is meaningless if a "Torque" or "generic" version of MWM is downloaded. It was also found to be helpful to get the latest build from the "snapshots" page, rather than the main download page, because the SGE integration document was correct only for a later version (5.3 rather than 5.2). The implication is that the software changes frequently.

MWM integrates most closely with Torque since CR develops them both. SGE integration requires the definition of a complex_variable, "`nodelist`", which must be given a default value. This value is then over-written by MWM at job submission. Although the integration document specified that short hostnames should be used by MWM to populate `nodelist`, long hostnames were required on cseem64t due to SGE's configuration. To accommodate this minor edits were made to two of the MWM Perl scripts - `/var/spool/moab/etc/config.sge.pl` and `/opt/moab/tools/node.query.sge.pl`.

A selection of `/etc/init.d` scripts, for different Linux distros, is provided in the contrib directory. All users must have `$MOABHOMEDIR` set in their environment, and `$PATH` must include the path to MWM binaries, so appropriate entries were added to `/etc/profile.local`.

MCS configuration

MWM (and by extension, MCM) is configured largely by the contents of the file `$MOABHOMEDIR/moab.cfg`. This is a straightforward text file comprising "`key=value[,value,…]`" pairs. The installation routine provides a basic configuration, which is enough to start MWM. However, careful study of the administrator's guide is required, as there are many, many options. For instance, MWM and MCM both provide means to power nodes off/on – which doesn't work unless Moab is configured to work with external management tools, such as IPMI[1] or xCAT[2].

These external interfaces are configured by means of `RMCFG` lines in `moab.cfg`. Such lines point to a script or text file. For example:

```
RMCFG[ipmi] TYPE=native CLUSTERQUERYURL=exec://$TOOLSDIR/ipmi.mon.pl
```

configures MWM to work with IPMI using a Perl script supplied by CR, and:

```
RMCFG[gmond] TYPE=NATIVE CLUSTERQUERYURL=file:///var/spool/moab/spool/gmond.out
```

extracts Ganglia metrics from a flat text file which is generated by a DisCo Perl script, triggered at regular intervals by a cron job.

Integration with Ganglia was particularly difficult and not entirely successful. It seems that MWM will only accept a small subset of Ganglia metrics since it expects to obtain most of this information from Torque – even when Torque is not present. Hence the additional script mentioned above. Furthermore, unless Ganglia returns exactly the same node names, MWM is easily confused and can mis-report the number of nodes whilst also assuming some of them are down.

In this evaluation, adding IPMI and Ganglia configurations to moab.cfg appeared to cause MWM some confusion, and led to the `moabd` daemon dying (and core-dumping) every time MCM was used to connect to cseem64t. The resolution provided by CR support was to alter MWM's configuration, so that the internal database was disabled and external text files used instead. On busy clusters this might lead to a loss of performance.

MWM commands

MWM provides a selection of command line tools for submitting and monitoring jobs and queues, such as "`showq`", "`showstate`" and "`msub`". These were not particularly useful during the evaluation. For example, `showq` returns less information about jobs than regular `qstat` does. `showstate` summarises the cluster status and provides a text-based diagram of the cluster, roughly equivalent to the visual cluster view in MCM (see "MCM Usage" below). `msub` could be useful for submitting jobs in a Moab-grid-enabled environment, where Moab will take care of which real resource manager to submit the job to, and thus which physical cluster. However, according to the documentation, msub does not recognise SGE-style qsub parameters, potentially requiring a learning curve on the part of users.

Two other MWM commands did prove helpful in diagnosing configuration issues. `mdiag` is a configuration analysis tool, which takes various options. For instance, "`mdiag –C`" validates the syntax of `moab.cfg`, and includes each line of the file in the output. "`mdiag –R`" reports the status of the various resource managers defined with "`RMCFG[]=`" lines in `moab.cfg`. `checknode` reports on the status of an individual node, which is useful for checking that the node is visible to MWM and that, for example, Ganglia metrics are being returned for that node.

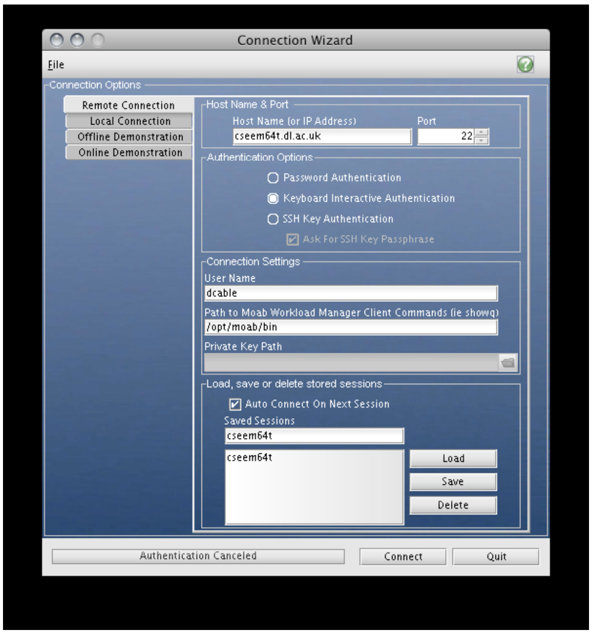
It may be necessary to add additional flags to user job submission scripts to enable MWM to return more meaningful information in the output of `showq` and other commands. For instance, if a user does not request a specific walltime, MWM reports a requested walltime of 99 hours 99 minutes.

[1] http://www.intel.com/design/servers/ipmi/
[2] http://xcat.sourceforge.net/

MCM installation

MCM couldn't be easier to install, provided Java is installed and mentioned in `$PATH`. Simply unpack the tarball somewhere sensible, and run "`mcm`". This starts a connection wizard, with four main options – local connection, remote connection, offline demo and online demo. Local connection is used when MWM and MCM are installed on the same machine; remote connection is used otherwise. Offline demo connects to a previously recorded MCM session; online demo allows connection to a CR demonstration cluster (or simulation – it's not clear which).

"Keyboard interactive authentication" was required to connect to cseem64t, and "Password authentication" to connect to csecell. Both apparently use ssh and require a password. The connection wizard allows different sessions to be saved and recalled, in a similar manner to PuTTY[1].



Connection Wizard

Connection Options: Remote Connection

Host Name (or IP Address): cseem64t.dl.ac.uk

Port: 22

Authentication Options:
○ Password Authentication
● Keyboard Interactive Authentication
○ SSH Key Authentication
☑ Ask For SSH Key Passphrase

Connection Settings

User Name: dcable

Path to Moab Workload Manager Client Commands (ie showq): /opt/moab/bin

Private Key Path:

Load, save or delete stored sessions

☑ Auto Connect On Next Session

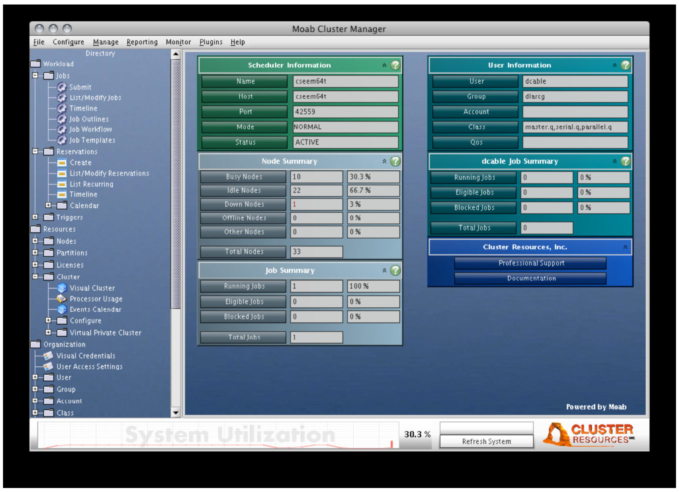
Saved Sessions: cseem64t

Authentication Canceled

Figure 1: MCM Connection Wizard

MCM usage

As it is a point-and-click GUI, utilising a navbar on the left, MCM is easy to use. It allows the user to examine/modify jobs, queues, reservations etc. in a multitude of different ways, including options to select the fields to be displayed, and the sort order. In fact it can be used to configure most of the scheduling options. Assuming, of course, that the user has appropriate permissions. These are set using the `ADMINCFG[n]` directive in `moab.cfg`, where `n` represents a level of access. Users not explicitly named in `moab.cfg` have the lowest level of access, which allows them only to view and submit jobs.



Moab Cluster Manager

Scheduler Information: Name cseem64t; Host cseem64t; Port 42559; Mode NORMAL; Status ACTIVE

User Information: User dcable; Group dlarcg; Account; Class master.q,serial.q,parallel.q; Qos

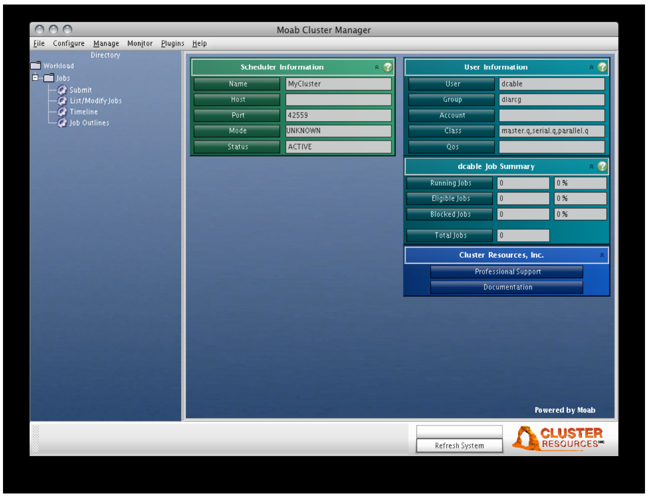
Node Summary: Busy Nodes 10, 30.3 %; Idle Nodes 22, 66.7 %; Down Nodes 1, 3 %; Offline Nodes 0, 0 %; Other Nodes 0, 0 %; Total Nodes 33

dcable Job Summary: Running Jobs 0, 0 %; Eligible Jobs 0, 0 %; Blocked Jobs 0, 0 %; Total Jobs 0

Job Summary: Running Jobs 1, 100 %; Eligible Jobs 0, 0 %; Blocked Jobs 0, 0 %; Total Jobs 1

System Utilization 30.3 %

Figure 2: MCM main window (administrative user)



Moab Cluster Manager

Scheduler Information: Name MyCluster; Host; Port 42559; Mode UNKNOWN; Status ACTIVE

User Information: User dcable; Group dlarcg; Account; Class master.q,serial.q,parallel.q; Qos

dcable Job Summary: Running Jobs 0, 0 %; Eligible Jobs 0, 0 %; Blocked Jobs 0, 0 %; Total Jobs 0

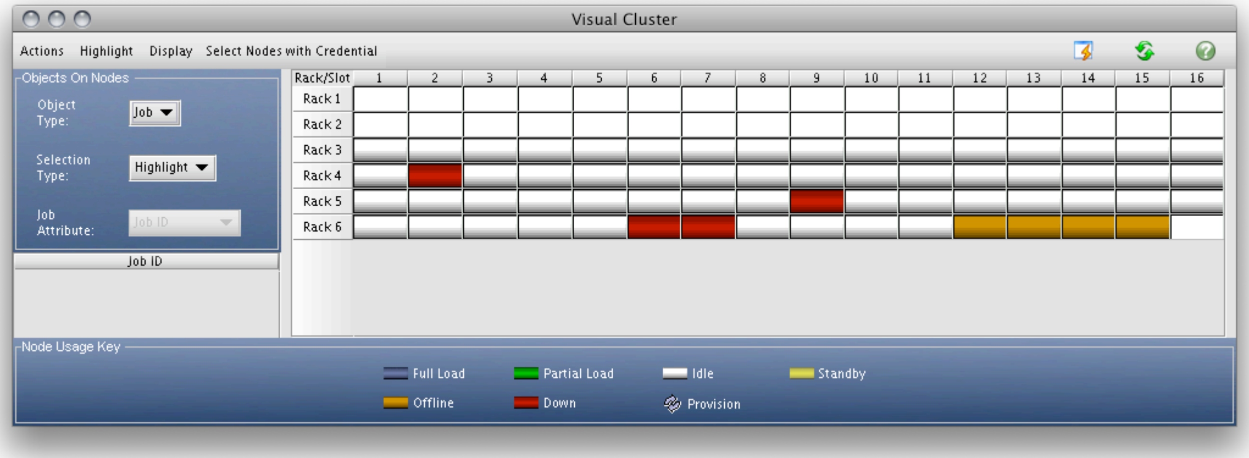
Figure 3: MCM main window (default user)

A useful feature is the command window, where MWM commands can be run directly and their output viewed. This was used in online demonstration mode to run the "`mdiag –C`" command on the CR machine. Thus it was possible to view the configuration of the demonstration machine - a useful starting point for configuration of cseem64t.

There is also a "Visual Cluster" display. This shows the cluster nodes in a grid formation. Whilst it's similar to the fashionable "physical" view of cluster nodes that other management tools provide (cf.

[1] http://www.chiark.greenend.org.uk/~sgtatham/putty/

Streamline CMA[1], ClusterVision OS 4[2]), it's actually just an arbitrary layout based either on positional information for each node entered in `moab.cfg`, on Torque's assumptions about node layout, or Moab's own assumptions. It might not be very useful unless working with a mixed batch of hardware, in which case it could be used to present a view of nodes grouped by type.



Object Type: Job

Selection Type: Highlight

Job Attribute: Job ID

Job ID

Rack/Slot	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Rack 1																
Rack 2																
Rack 3	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
Rack 4	⬜	🟥	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜
Rack 5	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	🟥	⬜	⬜	⬜	⬜	⬜	⬜	⬜
Rack 6	⬜	⬜	⬜	⬜	⬜	🟥	🟥	⬜	⬜	⬜	⬜	🟧	🟧	🟧	🟧	

Figure 4: MCM Visual Cluster window. For some reason MWM is reporting two racks of empty nodes. Mis-reporting of true node numbers is something that MWM appears to do well. Both cseem64t and csecell are represented in this image.

Moab Grid Manager

Connecting two Moab-aware clusters is relatively straight-forward. Using instructions provided by CR, the following changes were made to `moab.cfg` on cseem64t:

```
# Connect also to csecell
RMCFG[mg01] TYPE=moab SERVER=csecell.dl.ac.uk:42559
```

And on csecell:

```
# Connect also to cseem64t
RMCFG[cseem64t] TYPE=moab SERVER=cseem64t.dl.ac.uk:42559
```

This is a peer-to-peer configuration. It is possible to build a master-slave configuration instead.

The grid-aware version of MCM is Moab Grid Manager (MGM). As with MCM, it is a Java app and easily run from a desktop machine. It is visually very similar to MCM, although the fonts and colours used make it more visually attractive. Again there is a connection wizard. The main change is the addition of a Grid section to the navbar, and the Visual Cluster has become the Visual Grid window. On opening the Visual Grid window the user is presented with a representation of the clusters in the grid (see Fig. 5). Clicking "Display all nodes" results in the display shown in Fig. 6, which is very similar to the original Visual Cluster window.

[1] http://www.streamline-computing.com/index.php?wcId=118&xwcId=118
[2] http://www.clustervision.com/products_clustervisionos.php

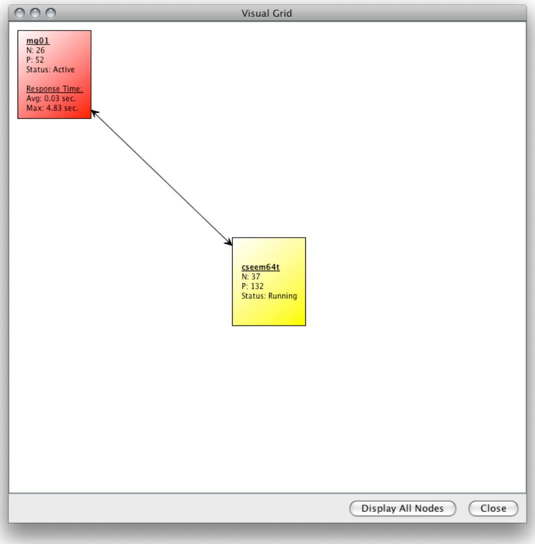
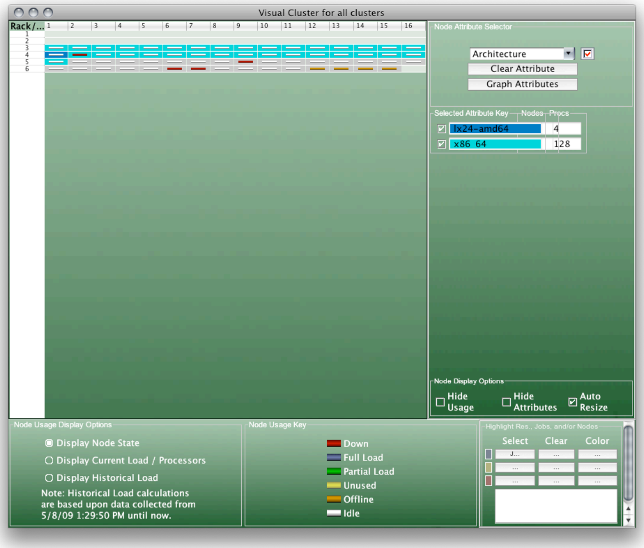


Figure 5: The Visual Grid window.



Note: Historical Load calculations are based upon data collected from 5/8/09 1:29:50 PM until now.

Figure 6: "Display all nodes". Note that this is basically the same as the Visual Cluster window, again with two phantom racks, but for some reason the icons used to represent nodes are much smaller. There appeared to be no way to make them bigger.

Chris Vaughan from CR stated in an email that: "From an administrative point there is a lot of value from having Moab set up in a grid configuration. It gives you a central point that aggregates information about job status, utilization, system performance and assists you in identifying issues throughout your infrastructure. From that central point you can set reservations and attach actions to those reservations, such as a software upgrade a diagnostics script etc. You can also perform actions on nodes so that you don't have to go out to the individual hosts if you want to reboot a node or power it off from Moab Grid Manager."

Clearly the focus is again on scheduling and resource management. The reporting tools are useful but inaccurate, at least in the configuration tried during the DisCo evaluation. Apart from remotely power-cycling nodes, there is little cluster management capability.

Desirable Functionality

If the MCS is to be genuinely useful as a cluster management tool, besides its undoubted abilities for resource management and job scheduling, it needs to be able to perform at least the following functions:

- Node imaging, re-imaging and image deployment
- A graphical utility equivalent to "`dsh`" or "`pexec`", which runs the same command across all nodes or groups of nodes
- Easy application deployment with "modules", integrated into the management tool.
- Node process management, providing an easy way to kill rogue user jobs (similar to, for example, "`pkilluser`" in ClusterVision OS)
- Use of IPMI to provide remote console windows

Costs

MCS is priced per compute node socket (not core) per year. The basic price is $85 per socket or $125 for the grid-enabled version. It attracts a 40% academic discount, and there are volume and multi-year discounts available.

Conclusion

MCS has strong credentials for job scheduling and provides a myriad of options for creating/modifying scheduling policies, triggers and backfill. The purpose of this evaluation however was to find out if it is also a good cluster management tool, particularly in a multi-cluster environment, since it would be advantageous to both DisCo and System Operations groups to have a single management console for all their clusters. It seems that MCS is not the product for this; it can require considerable effort to configure to work with existing tools, may not be completely reliable, and has some missing functionality. Dedicated cluster management products such as ClusterVision OS, Streamline CMA, ROCKS[1] and OSCAR[2] are more suitable, despite not having grid-enabled versions. Their management capabilities outweigh the disadvantage of (potentially) having to use several such products.

Finally, DisCo would like to acknowledge that Cluster Resources staff have been very helpful throughout this evaluation and the level of support provided was good.

[1] http://www.rocksclusters.org/wordpress/

[2] http://svn.oscar.openclustergroup.org/trac/oscar