Table of Contents

Table of Contents

Administration Guide

Error: Macro TOC(None) failed

```
'NoneType' object has no attribute 'endswith'
```

back to Table of Contents

Chapter 1: Introduction

1.1 Overview

OSCAR version 6.0.1 is a snapshot of the best known methods for building, programming, and using clusters. It consists of a fully integrated and easy to install software bundle designed for high performance computing (HPC) cluster. Everything needed to install, build, maintain, and use a Linux cluster is included in the suite.

OSCAR is the primary project of the Open Cluster Group. For more information on the group and its projects, visit its website http://www.openclustergroup.org/.

This document provides details about the management tools which are provided to make administering an OSCAR cluster easier. It also describes the packages included in the OSCAR distribution, and an introduction to their usage. For detailed instructions on each package you should refer to its own internal documentation or the mailing lists for that project.

Please be sure that you have the latest version of this document. The PDF version of this document which is included in the distribution is a snapshot of the OSCAR Trac wiki AdminGuide (http://svn.oscar.openclustergroup.org/trac/oscar/wiki/AdminGuide) which may have been updated since this version was released.

Note that the OSCAR-6.0.1 version is not necessarily suitable for production. OSCAR-6.0.1 is actually very similar to KDE-4.0: this version is not necessarily "designed" for the users who need all the capabilities traditionally shipped with OSCAR, but this is a good new framework to include and develop new capabilities and move forward. If you are looking for all the capabilities normally supported by OSCAR, we advice you to wait for a later release of OSCAR-6.x.

Error: Macro TOC(None) failed

```
'NoneType' object has no attribute 'endswith'
```

back to Table of Contents

Chapter 2: OSCAR Management Wizard

2.1 Overview

The OSCAR Management Wizard GUI has the following functionality to allow manipulation of an existing OSCAR cluster:

- Build a different OSCAR Client image
- Add new nodes to the cluster
- Delete nodes from the cluster
- Test the cluster
- Reimage or test a node with the Network Boot Manager
- View the Ganglia Monitoring System

All of the above menu items are optional, and can be executed in any order. However, once started, a choice should be followed through to completion, e.g., after adding new nodes, the Complete Cluster Setup must be done.

If you wish to wipe out the cluster and start over, see the Starting Over section of the Install Guide.

2.2 Launching the OSCAR Wizard

Once the OSCAR cluster is deployed, start the OSCAR wizard:

```
# /usr/bin/oscar_wizard manage
```

The wizard, as shown in Figure 1, is provided to guide you through the rest of the cluster management. To use the wizard, you will complete a series of steps, with each step being initiated by the pressing of a button on the wizard. Do not go on to the next step until the instructions say to do so, as there are times when you may need to complete an action outside of the wizard before continuing on with the next step. For each step, there is also a <Help> button located directly to the right of the step button. When pressed, the <Help> button displays a message box describing the purpose of the step.

Figure 1: OSCAR Wizard.

Error: Macro Image(figure1_oscar_manage.png) failed

```
sequence item 0: expected string, NoneType found
```

In brief, the functions of the various buttons is as follows:

- **Build OSCAR Client Image**
    - This step allows the user to build an OS image using SystemInstaller. This image will then be pushed to the compute nodes as part of cluster installation.
- **Add OSCAR Clients**
    - Additional clients can be defined in this section.
    - **Step 1: Define OSCAR Clients**

        - ◊ The user can select hostnames for your compute nodes, number of nodes, etc.
    - ♦ **Step 2: Setup Networking**
        - ◊ This step allows the user to tie MAC addresses to defined clients (in the previous step) such that when they boot up, they will automatically be imaged. Installation mode is also set in this step - currently available modes are: systemimager-rsync (default), systemimager-multicast, systemimager-bt. After this mode is set, the user should then configure DHCP Server and also select Setup Network Boot.
    - ♦ **Monitor Cluster Deployment** [optional]
        - ◊ This step brings up the SystemImager monitoring widget si_monitortk which provides very useful information regarding image progress. The user can also invoke the Virtual Console by double-clicking on a node and this will bring up a window with console messages during the installation.
    - ♦ **Step 3: Complete Cluster Setup**
        - ◊ Perform this step after all your cluster nodes have successfully been imaged and rebooted. This step will initiate post-cluster installation fix-ups and get it ready for production.
- **Delete OSCAR Clients**
    - ♦ This button allows the user to delete OSCAR clients from the cluster. Services will be stopped on the nodes to be deleted and restarted on all the remaining nodes. The cluster will be re-configured without the presence of the deleted node entries in ODA.
- **Install/Uninstall OSCAR Packages**
    - ♦ This step allows the selection of non-core OSCAR Packages to be installed or removed. Typically these are resource manager/scheduling systems, parallel programming libraries as well as other tools that aid in cluster administration. Certain packages may conflict with each other and only allow either one of them to be installed (eg. SGE vs TORQUE/Maui).
- **Test Cluster Setup** [optional]
    - ♦ OSCAR provides some tests for its packages and this step invokes all these testing harness to ensure that your cluster is setup properly and ready for production runs.
- **Network Boot Manager**
    - ♦ This controls what the client does when they boot from the LAN. Choices are: Install, LocalBoot, Kernel-x, and Memtest.
- **Ganglia Monitoring System**
    - ♦ Brings up a webpage showing the status of all the nodes in the cluster.

Error: Macro TOC(None) failed

```
'NoneType' object has no attribute 'endswith'
```

back to Table of Contents

Chapter 3: OSCAR Management Commands

3.1 Overview

This section covers OSCAR management using a command line, where functionality does not exist within the GUI. Most of the commands are either mksi* or si_* (i.e., SIS commands).

Topics covered include:

- Deleting an image (mksiimage)

Note that since most of the commands are SIS commands. It means that these commands do not update the OSCAR database (ODA) and therefore may result in a de-synchronization of the SIS database, the actual file system and the OSCAR database.

3.2 Deleting node images

It is sometimes useful to be able to delete one or more of the node images which OSCAR uses to provision the client nodes or to change which image is sent to a node when it joins the cluster.

To delete an OSCAR image, you need to first unassign the image from the client(s) which are currently using that image and then run the command *mksiimage*.

There is currently no way to change which image is assigned to a node from within the OSCAR GUI, so first you will need to delete the client node(s) if you wish to change which image is used on a particular node. It is not necessary if the image simply changes, this procedure is only necessary to completely change a node to use a different image entirely. To do so, invoke the OSCAR Wizard and select "Delete OSCAR Clients...".

mksiimage is a command from SystemInstaller which is used to manage SIS images on the headnode (image server).

Assuming the name of your image is *oscarimage*, here are the steps you need to do to fully delete an OSCAR image.

First delete the client(s) associated with the image, then execute:

```
# mksiimage --delete --name oscarimage
```

If this command does not work for some reason, you can also use the command *si_rmimage* to delete the image, just pass it the name of the image as argument.

si_rmimage is a command from SystemImager, the system OSCAR uses to deploy images to the compute nodes. SystemImager images are typically stored in */var/lib/systemimager/images*.

Note: If you want to use the *si_rmimage* command, execute the following commands to delete all data:

```
# si_rmimage oscarimage -force
```

3.3 Managing Distribution Repositories

Distribution repositories contain the packages needed for

- building client node images
- resolving dependencies when installing OSCAR packages onto the master node or the client nodes

First of all, it is still possible to apply updates using the standard package management systems (*yum*, *aptitude* and so on). For instance, when updated packages (security or bugfixes) are made available by the distributors, these can normally be installed to the master node by using commands like *up2date* or *yum update*. When the master node is configured correctly, these commands will access a remote repository with updated packages, download them into a package cache and install them onto the master node.

Mostly client OSCAR nodes are not set up for connectivity to the internet, therefore they need to be updated a different way. The OSCAR way is to update the distribution repository and update the client nodes and images from it. This gives the cluster administrator the full control over which packages are updated, when and why, and avoids situations like a cluster being automatically updated over night with some untested package that breaks the installation. With a well maintained distribution repository updating the master node, the client nodes or the images is very easy: use the *yume* command. We do not currently provide any tool to ease the update of the distribution repository, system administrators have to perform this task manually.

3.4 Examples

The following paragraphs give examples of system administration on a RPM based system.

Example: Updating the master node

```
yume update
```

Example: Updating the image oscarimage

```
yume --installroot /var/lib/systemimager/images/oscarimage update <file>
```

If you updated the image kernel, you need to regenerate the ramdisk files for that kernel

- Just edit the systemconfig.conf file to point to the new kernel, located on the head node in: `/var/lib/systemimager/images/oscarimage/etc/systemconfig/` (if you didn't use the default name for your client image ("oscarimage"), edit the path appropriately)
  ```
  >> nano /var/lib/systemimager/images/oscarimage/etc/systemconfig/systemcon
  ```
- confirm that the following option is setup on the above systemconfig.conf file:
  ```
  >> CONFIGRD = YES
  ```
- Done! Now any newly created client that uses this image will boot by default into the new kernel

Example: Updating the client nodes (be careful when the cluster is in production!)

```
cexec yume -y update
```

If you want to avoid the update of certain packages, use the --exclude option should help:

```
yume -y --exclude="kernel*" update
```

Example: repository maintenance

The repository maintenance consists basically of three steps:

1. Download the updated packages to the repository.
2. Optional: remove old packages from repository, i.e. clean it up.
3. Regenerate the repository metadata cache. Execute the command `yume --prepare --repo PATH_TO_REPOSITORY` on the master node.

The command `$OSCAR_HOME/scripts/repo-update` simplifies steps 1 and 2 of the repository maintenance. All you need is to find an URL pointing to the updated RPMs on the internet. This location must be repomd compliant, i.e. compatible with yum usage, because repo-update uses the remote metadata cache for finding the updated package versions.

```
Usage:
    repo-update [--url URL_TO_PACKAGES] [--repo LOCAL_PATH] [--prim PRIMARY.XML
    [--check] [--rmdup] [--verbose|-v]
```

Download packages from an on-line repository to the local repository LOCAL_PATH or the current directory. If the `repodata/primary.xml` file from the remote repository has already been downloaded and unpacked, it can be passed to the program with the *--prim* option. *--check* only lists the files which would be downloaded but does not start the wget transfer.

The *--rmdup* option leads to the removal of old versions of packages, keeping only the latest version. If the --url option is not specified, i.e. no downloads are required, the --rmdup option removes the duplicate packages (older versions) in the repository specified by --repo. If the --check option is specified, the packages which would be removed are listed.

Example: Repository update

Check packages which would be downloaded from a FC4 updates mirror site::

```
repo-update --url http://mirrors.dotsrc.org/fedora/updates/4/i386/ --check \
  --repo /tftpboot/distro/fedora-4-i386
```

Download updates to current directory (which could be the repository) and remove older packages::

```
repo-update --url http://mirrors.dotsrc.org/fedora/updates/4/i386/ --rmdup
```

Remove duplicate rpms (old package versions) from the repository (usefull when one has copied the packages over from */var/cache/yum/*/packages/*)

```
repo-update --rmdup --repo /tftpboot/distro/fedora-4-i386
```

Once your repository is fully updated, before you really update(yume update) your cluster with the new repository, you may need to update the repository metadata too by the following command:

```
yume --prepare --repo /tftpboot/distro/fedora-4-i386
```

Error: Macro TOC(None) failed

```
'NoneType' object has no attribute 'endswith'
```

back to Table of Contents

4 Package Notes

4.1 Torque Resource Manager

License: OpenPBS (Portable Batch System) v2.3 Software License

4.1.1 Overview

The Torque resource manager consists of three major components:

- The Torque server.
    - ♦ This runs on the OSCAR head node.
    - ♦ It controls the submission and running of jobs.
    - ♦ It also tracks the current state of cluster resources.
- A "mom" daemon on each cluster node.
    - ♦ responsible for actually starting and stopping jobs on the client nodes.

Torque also has a first-in-first-out scheduler which it can use, but by default OSCAR uses the Maui Scheduler as it is more flexible and powerful. The two programs are quite tightly integrated, but there is some overlap in functionality due to the fact that they were designed originally as separate utilities.

From the point of view of the user, most interaction takes place with the Torque resource manager, as that is the entry point for job script submission.

4.1.2 Commands

All Torque commands can be found under `/opt/pbs/bin` on the OSCAR head node, but are included in the path in a standard OSCAR installation. There are man pages available for these commands, but here are the most popular with some basic options:

- `qsub`: submits job to Torque
- `qdel`: deletes Torque job
- `qstat [-n]`: displays current job status and node associations
- `pbsnodes [-a]`: displays node status
- `pbsdsh`: distributed process launcher
- `xpbs`: X-windows Torque client. Simplifies both user and administrative tasks.

4.1.3 Submitting a Job

The qsub command is not necessarily intuitive. Here are some handy tips to know:

- Be sure to read the qsub man page.
- qsub only accepts a script filename for a target.
- The target script cannot take any command line arguments.
- For parallel jobs, the target script is only launched on ONE node. Therefore the script is responsible for launching all processes in the parallel job.

- One method of launching processes is to use the pbsdsh command within the script used as qsub?s target. pbsdsh will launch its target on all allocated processors and nodes (specified as arguments to qsub). Other methods of parallel launch exist, such as mpirun, included with each of the MPI packages.

Here is a sample qsub command line:

```
$ qsub -N my_jobname -e my_stderr.txt -o my_stdout.txt -q workq -l \
  nodes=X:ppn=Y:all,walltime=1:00:00 my_script.sh
```

Here is the contents of the **script.sh** file:

```
#!/bin/sh
echo Launchnode is ?hostname?
pbsdsh /path/to/my_executable
# All done
```

Alternatively, you can specify most of the qsub parameters in script.sh itself, and the qsub command would become:

```
$ qsub -l nodes=X:ppn=Y:all,walltime=1:00:00 script.sh
```

Here is the contents of the script.sh file:

```
#!/bin/sh
#PBS -N my_jobname
#PBS -o my_stdout.txt
#PBS -e my_stderr.txt
#PBS -q workq
echo Launchnode is ?hostname?
pbsdsh /path/to/my_executable
# All done
```

Notes about the above examples:

- "all" is an optional specification of a node attribute, or "resource".
- "workq" is a default queue name that is used in OSCAR clusters.
- 1:00:00 is in HH:MM:SS format (although leading zeros are optional).

4.1.4 Torque LAM/MPI Sample Script

```
#!/bin/sh
#PBS -N  C10-0.0
#Add other PBS options here.
#redirect error msg to err_pbs.txt
#PBS -e /home/oscartest/lamtest/err_pbs.txt
#redirect pbs output msgs to err_pbs.txt
#PBS -o /home/oscartest/lamtest/out_pbs.txt
#PBS -q workq

lamboot -v $PBS_NODEFILE
cd /home/oscartest/lamtest
DATE=\`date +%c\`
```

```
echo Job C10-0.0 started at $DATE
mpirun -np 16 ./eqn10p.x
lamhalt -v $PBS_NODEFILE
DATE=\`date +%c\`
echo Job finished at $DATE
echo

#Comments only after execution section
#to run this job alone type in directory
#/home/oscartest/lamtest:
#qsub -l nodes=8:ppn=2 ./qscript.sh
```

This script uses LAM-MPI to execute the program `/home/oscartest/lamtest/eqn10p.x` on the nodes provided by Torque. After running lamboot on the nodes provided, the script moves to the target directory and runs eqn10p.x using mpirun with 16 processes. Then the script cleans up after itself by running lamhalt. The 16 processes are started regardless of how many nodes Torque assigns.

4.1.5 Sample Script for MPICH

```
#!/bin/sh
#PBS -N  MPICH_Test
# Add other PBS options here.
#redirect error msg to err_pbs.txt
#PBS -e /home/oscartest/mpich/err_pbs.txt
#redirect pbs output msgs to err_pbs.txt
#PBS -o /home/oscartest/mpich/out_pbs.txt
#PBS -q workq

cd /home/oscartest/mpich
DATE=\`date +%c\`
echo Job MPICH_Test started at $DATE
time mpirun -np 16 -machinefile $PBS_NODEFILE ./ring
DATE=\`date +%c\`
echo Job finished at $DATE

#Comments only after execution section
#to run this job alone type in directory
#/home/oscartest/mpich/
#qsub -l nodes=8:ppn=2 ./qscript.sh
```

This script uses MPICH to execute the program `/home/oscartest/mpich/ring` on the nodes provided by Torque. The script moves to the target directory and runs the program ring using mpirun with 16 processes using the file $PBS_NODEFILE as the machinefile as the node list. Initialization and clean-up steps are not necessary when using MPICH. As with LAM the 16 processes are started regardless of how many nodes Torque assigns.

4.1.6 Environment Variables

There are a number of predefined environment variables which the Torque resource manager makes available to scripts run through it. These can be useful for debugging Torque setup files as well as data organization.

The following environment variables relate to the submission machine:

Option	Description
PBS_O_HOST	The host machine on which the qsub command was run.
PBS_O_LOGNAME	The login name on the machine on which the qsub was run.
PBS_O_HOME	The home directory from which the qsub was run.
PBS_O_WORKDIR	The working directory from which the qsub was run.

The following variables relate to the environment where the job is executing:

Option	Description
PBS_ENVIRONMENT	This is set to PBS_BATCH for batch jobs and to PBS_INTERACTIVE for interactive jobs.
PBS_O_QUEUE	The original queue to which the job was submitted.
PBS_JOBID	The identifier that PBS assigns to the job.
PBS_JOBNAME	The name of the job.
PBS_NODEFILE	The file containing the list of nodes assigned to a parallel job.

4.1.7 Additional Resources

More information about using and configuring Torque is available on the Cluster Resources website at http://www.clusterresources.com/wiki/doku.php?id=torque:torque_wiki

4.2 MAUI Scheduler

Official website: http://www.clusterresources.com/pages/products/maui-cluster-scheduler.php

License: MAUI License

4.2.1 Overview

The Maui scheduler takes care of scheduling jobs on the cluster based on a series of sophisticated algorithms which take into account current system resources, cluster usage rates, past user activity, and many other factors. These algorithms are very flexible and are configurable by the cluster administrator.

4.2.2 Maui Commands

There are several commands available to the Maui Scheduler which can aid in the debugging of problems with Torque or Maui configurations. These commands are located in `/opt/maui/bin` and generally need to be run with root permissions. Refer to maui documentation for further detail about these commands.

- diagnose: displays information about nodes, jobs and other resources
- checkjob: displays less verbose information about particular jobs
- showstats: shows usage stats for scheduled jobs

4.2.3 Additional Resources

Additional documentation and resources for Maui are available on the project's website: http://www.clusterresources.com/pages/resources/documentation.php

4.3 C3

Official website: http://www.csm.ornl.gov/torc/c3/

License:

> Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted provided that the above copyright notice appear in all copies and that both the copyright notice and this permission notice appear in supporting documentation.

4.3.1 Overview

The Cluster Command Control (C3) tools are a suite of cluster tools developed at Oak Ridge National Laboratory that are useful for both administration and application support. The suite includes tools for cluster-wide command execution, file distribution and gathering, process termination, remote shutdown and restart, and system image updates.

A short description of each tool follows:

- cexec: general utility that enables the execution of any standard command on all cluster nodes
- cget: retrieves files or directories from all cluster nodes
- ckill: terminates a user specified process on all cluster nodes
- cpush: distribute files or directories to all cluster nodes
- cpushimage: update the system image on all cluster nodes using an image captured by the SystemImager tool
- crm: remove files or directories from all cluster nodes
- cshutdown: shutdown or restart all cluster nodes
- cnum: returns a node range number based on node name
- cname: returns node names based on node ranges
- clist: returns all clusters and their type in a configuration file

The default method of execution for the tools is to run the command on all cluster nodes concurrently. However, a serial version of cexec is also provided that may be useful for deterministic execution and debugging. Invoke the serial version of cexec by typing cexecs instead of cexec. For more information on how to use each tool, see the man page for the specific tool.

4.3.2 File Format

Specific instances of C3 commands identify their compute nodes with the help of *cluster configuration files*: files that name a set of accessible clusters and describe the set of machines in each accessible cluster. */etc/c3.conf*, the default cluster configuration file, should consist of a list of *cluster descriptor blocks*: syntactic objects that name and describe a single cluster that is accessible to that system?s users. The

following is an example of a default configuration file that contains exactly one cluster descriptor block: a block that describes a cluster of 64 nodes:

```
cluster cartman {
  cartman-head:node0 #head node
  node[1-64] #compute nodes
}
```

The cluster tag denotes a new cluster descriptor block. The next word is the name of the cluster (in this example, *cartman*). The first line in the configuration is the head node. The first name is the external interface followed by a colon and then the internal interface (for example, an outside user can login to the cluster by ssh?ing to *cartman-head.mydomain.com*). If only one name is specified, then it is assumed to be both external and internal. Starting on the next line is the node definitions. Nodes can be either ranges or single machines. The above example uses ranges ? node1 through node64. In the case of a node being offline, two tags are used: exclude and dead. exclude sets nodes offline that are declared in a range and dead indicates a single node declaration is dead. The below example declares 32 nodes in a range with several offline and then 4 more nodes listed singularly with 2 offline.

```
cluster kenny {
  node0 #head node
  dead placeholder #change command line to 1 indexing
  node[1-32] #first set of nodes
  exclude 30 #offline nodes in the range
  exclude [5-10]
  node100 #single node definition
  dead node101 #offline node
  dead node102
  node103
}
```

One other thing to note is the use of a place holder node. When specifying ranges on the command line a nodes position in the configuration file is relevant. Ranges on the command line are 0 indexed. For example, in the cartman cluster example (first example), node1 occupies position 0 which may not be very intuitive to a user. Putting a node offline in front of the real compute nodes changes the indexing of the C3 command line ranges. In the kenny cluster example (second example) node1 occupies position one.

For a more detailed example, see the **c3.conf** man page.

4.3.3 Specifying Ranges

Ranges can be specified in two ways, one as a range, and the other as a single node. Ranges are specified by the following format: *m-n*, where *m* is a positive integer (including zero) and *n* is a number larger than *m*. Single positions are just the position number. If discontinuous ranges are needed, each range must be separated by a ",". The range "0-5, 9, 11" would execute on positions 0, 1, 2, 3, 4, 5, 9, and 11 (nodes marked as *offline* will not participate in execution).

There are two tools used to help manage keeping track of which nodes are at which position: cname(1) and cnum(1). cnum assumes that you know node names and want to know their position. cname takes a range argument and returns the node names at those positions (if no range is specified it assumes that you want all the nodes in the cluster). See their man pages for details of use.

NOTE: ranges begin at zero!

4.3.4 Machine Definitions

Machine definitions are what C3 uses to specify clusters and ranges on the command line. There are four cases a machine definition can take. First is that one is not specified at all. C3 will execute on all the nodes on the *default cluster* in this case (the *default cluster* is the first cluster in the configuration file). An example would be as follows:

```
$ cexec ls -l
```

the second case is a range on the default cluster. This is in a form of <*:range*>. An example *cexec* would be as follows:

```
$ cexec :1-4,6 ls -l
```

This would execute ls on nodes 1, 2, 3, 4, and 6 of the default cluster. The third method is specifying a specific cluster. This takes the form of <*cluster name:*>. An example *cexec* would be as follows:

```
$ cexec cartman: ls -l
```

This would execute ls on every node in cluster cartman. The fourth (and final) way of specifying a machine definition would be a range on a specific cluster. This takes the form of <*cluster name:range*>. An example *cexec* would be as follows:

```
$ cexec cartman:2-4,10 ls -l
```

This would execute ls on nodes 2, 3, 4, and 10 on cluster cartman. These four methods can be mixed on a single command line. for example

```
$ cexec :0-4 stan: kyle:1-5 ls -l
```

is a valid command. it would execute ls on nodes 0, 1, 2, 3, and 4 of the default cluster, all of *stan* and nodes 1, 2, 3, 4, and 5 of *kyle* (the *stan* and *kyle* cluster configuration blocks are not shown here). In this way one could easily do things such as add a user to several clusters or read */var/log/messages* for an event across many clusters. See the *c3-range* man page for more detail.

4.3.5 Other Considerations

In most cases, C3 has tried to mimic the standard Linux command it is based on. This is to make using the cluster as transparent as possible. One of the large differences is such as using the interactive options. Because one would not want to be asked yes or no multiple times for each node, C3 will only ask *once* if the interactive option is specified. This is very important to remember if running commands such as "*crm --all -R /tmp/foo*" (recursively delete */tmp/foo* on every node in every cluster you have access too).

Multiple cluster uses do not necessarily need to be restricted by listing specific nodes; nodes can also be grouped based on role, essentially forming a meta-cluster. For example, if one wishes to separate out PBS server nodes for specific tasks, it is possible to create a cluster called *pbs-servers* and only execute a given command on that cluster. It is also useful to separate nodes out based on things such as hardware (e.g., fast-ether/gig-ether), software (e.g., some nodes may have a specific compiler), or role (e.g., *pbs-servers*).

4.4 LAM

Official website: http://www.lam-mpi.org/

License:

Indiana University provides no reassurances that the source code provided does not infringe the patent or any other intellectual property rights of any other entity. Indiana University disclaims any liability to any recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise.

LICENSEE UNDERSTANDS THAT SOFTWARE IS PROVIDED "AS IS" FOR WHICH NO WARRANTIES AS TO CAPABILITIES OR ACCURACY ARE MADE. INDIANA UNIVERSITY GIVES NO WARRANTIES AND MAKES NO REPRESENTATION THAT SOFTWARE IS FREE OF INFRINGEMENT OF THIRD PARTY PATENT, COPYRIGHT, OR OTHER PROPRIETARY RIGHTS. INDIANA UNIVERSITY MAKES NO WARRANTIES THAT SOFTWARE IS FREE FROM "BUGS", "VIRUSES", "TROJAN HORSES", "TRAP DOORS", "WORMS", OR OTHER HARMFUL CODE. LICENSEE ASSUMES THE ENTIRE RISK AS TO THE PERFORMANCE OF SOFTWARE AND/OR ASSOCIATED MATERIALS, AND TO THE PERFORMANCE AND VALIDITY OF INFORMATION GENERATED USING SOFTWARE.

Indiana University has the exclusive rights to license this product under this license.

4.4.1 Overview

LAM (Local Area Multicomputer) is an MPI programming environment and development system for heterogeneous computers on a network. With LAM/MPI, a dedicated cluster or an existing network computing infrastructure can act as a single parallel computer. LAM/MPI is considered to be "cluster friendly," in that it offers daemon-based process startup/control as well as fast client-to-client message passing protocols. LAM/MPI can use TCP/IP and/or shared memory for message passing.

LAM features a full implementation of MPI-1, and much of MPI-2. Compliant applications are source code portable between LAM/MPI and any other implementation of MPI. In addition to providing a high-quality implementation of the MPI standard, LAM/MPI offers extensive monitoring capabilities to support debugging. Monitoring happens on two levels. First, LAM/MPI has the hooks to allow a snapshot of process and message status to be taken at any time during an application run. This snapshot includes all aspects of synchronization plus datatype maps/signatures, communicator group membership, and message contents (see the XMPI application on the main LAM web site). On the second level, the MPI library is instrumented to produce a cummulative record of communication, which can be visualized either at runtime or post-mortem.

4.4.2 Notes about OSCAR's LAM/MPI Setup

The OSCAR environment is able to have multiple MPI implementations installed simultaneously ? see Section 2.8 (page 13) for a description of the switcher program. LAM/MPI is configured on OSCAR to use the Secure Shell (ssh) to initially start processes on remote nodes. Normally, using ssh requires each user to set up cryptographic keys before being able to execute commands on remote nodes with no password. The OSCAR setup process has already taken care of this step for you. Hence, the LAM command lamboot should work with no additional setup from the user.

4.4.3 Setting up switcher to use LAM/MPI

In order to use LAM/MPI successfully, you must first ensure that switcher is set to use LAM/MPI. First, execute the following command:

```
$ switcher mpi --show
```
If the result contains a line beginning with "default" followed by a string containing "lam" (e.g., "lam-7.0.6"), then you can skip the rest of this section. Otherwise, execute the following command:

```
$ switcher mpi --list
```
This shows all the MPI implementations that are available. Choose one that contains the name "lam" (e.g., "lam-7.0.6") and run the following command:

```
$ switcher mpi = lam-7.0.6
```
This will set all *future* shells to use LAM/MPI. In order to guarantee that all of your login environments contain the proper setup to use LAM/MPI, it is probably safest to logout and log back in again. Doing so will guarantee that all of your interactive shells will have the LAM commands and man pages will be found (i.e., your *$PATH* and *$MANPATH* environment variables set properly for LAM/MPI). Hence, you will be able to execute commands such as "*mpirun*" and "*man lamboot*" without any additional setup.

4.4.4 General Usage

The general scheme of using LAM/MPI is as follows:

- Use the *lamboot* command to start up the LAM run-time environment (RTE) across a specified set of nodes. The *lamboot* command takes a single argument: the filename of a hostfile containing a list of nodes to run on. For example:
  ```
  $ lamboot my_hostfile
  ```
- Repeat the following steps as many times as necessary:
  - Use the MPI "wrapper" compilers (*mpicc* for C programs, *mpiCC* for C++ programs, and *mpif77* for fortran programs) to compile your application. These wrapper compilers add in all the necessary compiler flags and then invoke the underlying "real" compiler. For example:
    ```
    mpicc myprogram.c -o myprogram
    ```
- Use the *mpirun* command to launch your parallel application in the LAM RTE. For example:
  ```
  $ mpirun C myprogram
  ```

  The mpirun command has many options and arguments ? see the man page and/or "mpirun -h" for more information.

- If your parallel program fails ungracefully, use the lamclean command to "clean" the LAM RTE and guarantee to remove all instances of the running program.
- Use the *lamhalt* command to shut down the LAM RTE. The *lamhalt* command takes no arguments.

Note that the wrapper compilers are all set to use the corresponding GNU compilers (*gcc*, *g++*, and *gf77*, respectively). Attempting to use other compilers may run into difficulties because their linking styles may be different than what the LAM libraries expect (particularly for C++ and Fortran compilers).

4.4.5 Other Resources

The LAM/MPI web site (http://www.lam-mpi.org/) contains much, much more information about LAM/MPI, including:

- A large Frequently Asked Questions (FAQ) list
- Usage tutorials and examples
- Access to the LAM user?s mailing list, including subscription instructions and web archives of the list

Make today a LAM/MPI day!

4.5 MPICH

4.5.1 Overview

Official Website: http://www-unix.mcs.anl.gov/mpi/mpich1/

MPICH is another MPI implementation which is installed by OSCAR. It is lighter weight than LAM/MPI and other implementations because it does not require any daemon processes to run on the compute nodes. This has historically made it a popular choice with people doing benchmarking and production runs of well known codes. It works in a very similar fashion to LAM/MPI but using mpirun requires an explicit list of nodes to be suplied at runtime, either from the command line or from a default file.

MPICH example scripts sutable for a standard OSCAR install are included in the Torque scripting section.

4.6 OpenMPI

Official website: http://www.open-mpi.org/

License: BSD

4.7 The OSCAR Password Installer and User Management (OPIUM)

Official website: N/A.

License: GPL v.2

4.7.1 Overview

The OPIUM package includes facilities which synchronize the cluster?s accounts and configures ssh for users. The user account synchronization may only be run by root, and is automatically triggered at regular intervals. OPIUM configures ssh such that every user can traverse the cluster securely without entering a password, once logged on to the head node. This is done using ssh user keys, in the .ssh folder in your home directory. It is not recommended that you make changes here unless you know what you are doing. If you change your password, make sure to change it on the OSCAR head node, because changes propagate to the cluster nodes from there.

4.8 Packet Filtering with pfilter

Official webiste: N/A.

License: GPL v.2

4.8.1 Overview

When the pfilter packet filtering system is turned on, the default OSCAR settings allow any network communication between the machines in the cluster, and allow ssh and http access to the cluster main machine from the outside.

Communication between cluster machines and the outside network are limited to outgoing connections only. Incoming network connections to cluster machines are blocked. To allow outside network connections to ports on the cluster machines, special rules will have to be added to the pfilter configuration. See your cluster administrator for help on this.

4.9 PVM

Official website: http://www.csm.ornl.gov/pvm/

License: Freely distributable.

4.9.1 Overview

PVM (Parallel Virtual Machine) is a software package that permits a heterogeneous collection of Unix and/or Windows computers hooked together by a network to be used as a single large parallel computer. Thus large computational problems can be solved more cost effectively by using the aggregate power and memory of many computers. The software is very portable. The source, which is available free thru netlib, has been compiled on everything from laptops to CRAYs.

PVM enables users to exploit their existing computer hardware to solve much larger problems at minimal additional cost. Hundreds of sites around the world are using PVM to solve important scientific, industrial, and medical problems in addition to PVM?s use as an educational tool to teach parallel programming. With tens of thousands of users, PVM has become the de facto standard for distributed computing world-wide.

4.9.2 Using PVM

The default OSCAR installation tests PVM via a Torque/PBS job (see also: Section 2.9 on page 18). However, some users may choose to use PVM outside of this context so a few words on usage may be helpful (the examples in this section assume a shared filesystem, as is used with OSCAR.) The default location for user executables is $HOME/pvm3/bin/$PVM ARCH. On an IA-32 Linux machine, this is typically of the form: `/home/sgrundy/pvm3/bin/LINUX` (replace "LINUX" with "LINUX64" on IA-64). This is where binaries should be placed so that PVM can locate them when attempting to spawn tasks. This is detailed in the pvm intro(1PVM) manual page when discussing the environment variables PVM PATH and PVM WD.

The "hello world" example shipped with PVM demonstrates how one can compile and run a simple application outside of Torque/PBS. The following screen log highlights this for a standard user sgrundy (Solomon Grundy).

```
# Crate default directory for PVM binaries (one time operation)
sgrundy: $ mkdir -p $HOME/pvm3/bin/$PVM_ARCH
# Copy examples to local ?hello? directory
sgrundy: $ cp $PVM_ROOT/examples/hello* $HOME/hello-example
sgrundy: $ cd $HOME/hello-example
# Compile a hello world, using necessary include (-I) and library
# (-L) search path info as well as the PVM3 lib.
sgrundy: $ gcc -I$PVM_ROOT/include hello.c -L$PVM_ROOT/lib/$PVM_ARCH \
> -lpvm3 -o hello
sgrundy: $ gcc -I$PVM_ROOT/include hello_other.c -L$PVM_ROOT/lib/$PVM_ARCH \
> -lpvm3 -o hello_other
# Move the companion that will be spawned to the default
# PVM searchable directory
sgrundy: mv hello_other $HOME/pvm3/bin/$PVM_ARCH
```

At this point you can start PVM, add hosts to the virtual machine and run the application:

```
# Start PVM and add one host "oscarnode1".
sgrundy: $ pvm
pvm> add oscarnode1
add oscarnode1
1 successful
                        HOST          DTID
                  oscarnode1         80000
pvm> quit
quit

Console: exit handler called
pvmd still running.
sgrundy: $

# Run master portion of hello world which contacts the companion.
sgrundy: $ ./hello
i?m t40005
from t80002: hello, world from oscarnode1.localdomain

# Return to the PVM console and terminate (?halt?) the virtual machine.
sgrundy: $
sgrundy: $ pvm
pvmd already running
pvm> halt
halt
Terminated
sgrundy: $
```

An alternate method is to use options in the hostfile supplied to pvm when started from the commandline. The advantage to the hostfile options is that you don?t have to place your binaries in the default location or edit any ".dot" files. You can compile and run the "hello world" example in this fashion by using a simple hostfile

as shown here.

The example below uses the same "hello world" program that was previously compiled, but using a hostfile with the appropriate options to override the default execution and working directory. Remember that the "hello" program exists in the `/home/sgrundy/helloexample` directory:

```
sgrumpy: $ cat myhostfile
*   ep=/home/sgrundy/hello-example   wd=/home/sgrundy/hello-example
oscarnode1
```

The options used here are:

- * ? any node can connect
- ep ? execution path, here set to local directory
- wd ? working directory, here set to local directory
- *nodes* ? a list of nodes, one per line

Now, we can startup pvm using this *myhostfile* and run the *hello* application once again.

```
# Now, we just pass this as an argument to PVM upon startup.
sgrundy: $ pvm myhostfile
pvm> quit
quit
Console: exit handler called
pvmd still running.
# The rest is the same as the previous example
sgrundy: $ ./hello
i?m t40005
from t80002: hello, world from oscarnode1.localdomain
sgrundy: $ pvm
pvmd already running
pvm> halt
halt
Terminated
sgrundy: $
```

4.9.3 Resources

The OSCAR installation of PVM makes use of the env-switcher package (also see Section 2.8, page 13). This is where the system-wide $PVM_ROOT, $PVM_ARCH and $PVM_RSH environment variable defaults are set. Traditionally, this material was included in each user?s ".dot" files to ensure availability with noninteractive shells (e.g. rsh/ssh). Through the env-ewitcher package, a user can avoid any ".dot" file adjustments by using the hostfile directive or default paths for binaries as outlined in the Usage Section 2.6.1.

For additional information see also:

- PVM web site: `http://www.csm.ornl.gov/pvm/`
- Manual Pages: *pvm(1)*, *pvm intro(1)*, *pvmd3(1)*
- Release docs: `$PVM_ROOT/doc/*`

4.10 System Installation Suite (SIS)

Official websites

> SystemImager: http://wiki.systemimager.org/index.php/Main_Page SytemInstaller?: http://wiki.systemimager.org/index.php/SystemInstaller SystemConfigurator: http://wiki.systemimager.org/index.php/System_Configurator

License: GPLv2 or later.

4.10.1 An overview of SIS

The System Installation Suite, or SIS, is a tool for installing Linux systems over a network. It is used in OSCAR to install the client nodes. SIS also provides the database from which OSCAR obtains its cluster configuration information. The main concept to understand about SIS is that it is an image based install tool. An image is basically a copy of all the files that get installed on a client. This image is stored on the server and can be accessed for customizations or updates. You can even chroot into the image and perform builds.

Once this image is built, clients are defined and associated with the image. When one of these clients boots using a SIS auto-install environment, either on floppy, CD, or through a network boot, the corresponding image is pulled over the network using rsync. Once the image is installed, it is customized with the hardware and networking information for that specific client and it is then rebooted. When booting the client will come up off the local disk and be ready to join the OSCAR cluster.

4.10.2 Building an Image

Normally, an OSCAR image is built using the <Build OSCAR Client Image> button on the OSCAR wizard. This button brings up a panel that is actually directly from the SIS GUI tksis. Once the information is filled in, the SIS command mksiimage is invoked to actually build the image.

In addition to building an image, you can use tksis or mksiimage to delete images as well. Images can take a fair amount of disk space, so if you end up with images that you aren?t using, you can delete them to recover some space.

4.10.3 Managing SIS Images

Much like the OSCAR image creation, the <Define OSCAR Clients> button actually invokes a tksis panel. There are a couple of SIS commands that are used to manage the client definitions. mksirange is used to define a group of clients. More importantly, mksimachine can be used to update client definitions. If, for example, you needed to change the MAC address after replacing one of your clients, you could use mksimachine.

4.10.4 Maintaining Client Software

There are many different ways to maintain the software installed on the client nodes. Since SIS is image based, it allows you to also use an image based maintenance scheme. Basically, you apply updates and patches to your images and then resync the clients to their respective images. Since rsync is used, only the actual data that has changed will be sent over the network to the client. The SIS command updateclient can be

run on any client to initiate this update.

4.10.5 Additional Information

To obtain more detailed information about SIS, please refer to the many man pages that are shipped with SIS. Some of the more popular pages are:

- tksis
- mksiimage
- mksidisk
- mksirange
- mksimachine
- systemconfigurator
- updateclient

You can also access the mailing lists and other docs through the sisuite home page, http://sisuite.org/.

4.11 Switcher Environment Manager

Official website: N/A.

License: Freely distributable.

4.11.1 Overview

Experience has shown that requiring untrained users to manually edit their "dot" files (e.g., `$HOME/.bashrc`,`$HOME/.login`, `$HOME/.logout`, etc.) can result in damaged user environments. Side effects of damaged user environments include:

- Lost and/or corrupted work
- Severe frustration / dented furniture
- Spending large amounts of time debugging "dot" files, both by the user and the system administrator

However, that it was a requirement for the OSCAR switcher package that advanced users should not be precluded - in any way - from either not using switcher, or otherwise satisfying their own advanced requirements without interference from switcher.

The OSCAR switcher package is an attempt to provide a simple mechanism to allow users to manipulate their environment. The switcher package provides a convenient command-line interface to manipulate the inclusion of packages in a user?s environment. Users are not required to manually edit their "dot" files, nor are they required to know what the inclusion of a given package in the environment entails.3 For example, if a user specifies that they want LAM/MPI in their environment, switcher will automatically add the appropriate entries to the $PATH and $MANPATH environment variables.

Finally, the OSCAR switcher package provides a two-level set of defaults: a system-level default and a user-level default. User-level defaults (if provided) override corresponding system-level defaults. This allows a system administrator to (for example) specify which MPI implementation users should have in their environment by setting the system-level default. Specific users, however, may decide that they want a different implementation in their environment and set their personal user-level default.

Note, however, that switcher does not change the environment of the shell from which it was invoked. This is a critical fact to remember when administrating your personal environment or the cluster. While this may seem inconvenient at first, switcher was specifically designed this way for two reasons:

1. If a user inadvertantly damages their environment using switcher, there is still [potentially] a shell with an undamaged environment (i.e., the one that invoked switcher) that can be used to fix the problem.
2. The switcher package uses the modules package for most of the actual environment manipulation (see `http://modules.sourceforge.net/`). The modules package can be used directly by users (or scripts) who wish to manipulate their current environment.

The OSCAR switcher package contains two sub-packages: modules and *env-switcher*. The modules package can be used by itself (usually for advanced users). The *env-switcher* package provides a persistent modules-based environment.

4.11.2 The modules package

The modules package (see `http://modules.sourceforge.net/`) provides an elegant solution for individual packages to install (and uninstall) themselves from the current environment. Each OSCAR package can provide a modulefile that will set (or unset) relevant environment variables, create (or destroy) shell aliases, etc.

An OSCAR-ized modules RPM is installed during the OSCAR installation process. Installation of this RPM has the following notable effects:

- Every user shell will be setup for modules - notably, the commands "module" and "man module" will work as expected.
- Guarantee the execution of all modulefiles in a specific directory for every shell invocation (including corner cases such as non-interactive remote shell invocation by rsh/ssh).

Most users will not use any modules commands directly - they will only use the *env-switcher* package. However, the modules package can be used directly by advanced users (and scripts).

4.11.3 The env-switcher package

The *env-switcher* package provides a persistent modules-based environment. That is, *env-switcher* ensures to load a consistent set of modules for each shell invocation (including corner cases such as non-interactive remote shells via rsh/ssh). *env-switcher* is what allows users to manipulate their environment by using a simple command line interface - not by editing "dot" files.

It is important to note that *using the switcher command alters the environment of all *future* shells / user environments*. switcher does not change the environment of the shell from which it was invoked. This may seem seem inconvenient at first, but was done deliberately. See the rationale provided at the beginning of this section for the reasons why. If you?re really sure that you know what you?re doing, you can use the "switcher-reload" command after changing your switcher settings via the switcher command. This will change your current shell/environment to reflect your most recent switcher settings.

env-switcher manipulates four different kinds of entities: tags, attributes, and values.

- Tags are used to group similar software packages. In OSCAR, for example, "mpi" is a commonly used tag.
- Names are strings that indicate individual software package names in a tag.
- Each tag can have zero or more attributes.
- An attribute, if defined, must have a single value. An attribute specifies something about a given tag by having an assigned value.

There are a few built-in attributes with special meanings (any other attribute will be ignored by *env-switcher*, and can therefore be used to cache arbitrary values). "default" is probably the most-commonly used attribute - its value specifies which package will be loaded (as such, its value is always a name). For example, setting the "default" attribute on the "mpi" tag to a given value will control which MPI implementation is loaded into the environment.

env-switcher operates at two different levels: system-level and user-level. The system-level tags, attributes, and values are stored in a central location. User-level tags, attributes, and values are stored in each user?s $HOME directory.

When *env-switcher* looks up entity that it manipulates (for example, to determine the value of the "default" attribute on the "mpi" tag), it attempts to resolves the value in a specific sequence:

1. Look for a "default" attribute value on the "mpi" tag in the user-level defaults
2. Look for a "default" attribute value on the "global" tag in the user-level defaults
3. Look for a "default" attribute value on the "mpi" tag in the system-level defaults
4. Look for a "default" attribute value on the "global" tag in the system-level defaults

In this way, a four-tiered set of defaults can be effected: specific user-level, general user-level, specific system-level, and general system-level.

The most common *env-switcher* commands that users will invoke are:

- `switcher --list` List all available tags.
- `switcher <tag> --list` List all defined attributes for the tag <tag>.
- `switcher <tag> = <value> [--system]` A shortcut nomenclature to set the "default" attribute on <tag> equal to the value <value>. If the --system parameter is used, the change will affect the system-level defaults; otherwise, the user?s personal user-level defaults are changed.
- `switcher <tag> --show` Show the all attribute / value pairs for the tag <tag>. The values shown will be for attributes that have a resolvable value (using the resolution sequence described above). Hence, this output may vary from user to user for a given <tag> depending on the values of user-level defaults.
- `switcher <tag> --rm-attr <attr> [--system]` Remove the attribute <attr> from a given tag. If the --system parameter is used, the change will affect the system level defaults; otherwise, the user?s personal user-level defaults are used. Section 2.8.3 shows an example scenario using the switcher command detailing how to change which MPI implementation is used, both at the system-level and user-level. See the man page for switcher(1) and the output of switcher --help for more details on the switcher command.

4.11.4 Choosing a Default MPI

OSCAR has a generalized mechanism to both set a system-level default MPI implementation, and also to allow users to override the system-level default with their own choice of MPI implementation. This allows multiple MPI implementations to be installed on an OSCAR cluster (e.g., LAM/MPI and MPICH), yet still provide unambiguous MPI implementation selection for each user such that "mpicc foo.c -o foo" will give deterministic results.

4.11.5 Setting a system-level default

The system-level default MPI implementation can be set in two different (yet equivalent) ways:

1. During the OSCAR installation, the GUI will prompt asking which MPI should be the system-level default. This will set the default for all users on the system who do not provide their own individual MPI settings.
2. As root, execute the command:
   ```
   # switcher mpi --list
   ```

This will list all the MPI implementations available. To set the system-level default, execute the command:

```
# switcher mpi = name --system
```

where "name" is one of the names from the output of the --list command.

NOTE: System-level defaults for switcher are currently propogated to the nodes on a periodic basis. If you set the system-level MPI default, you will either need to wait until the next automatic "push" of configuration information, or manually execute the `/opt/sync files/bin/sync` files command to push the changes to the compute nodes.

NOTE: Using the switcher command to change the default MPI implementation will modify the PATH and MANPATH for all **future** shell invocations - it does not change the environment of the shell in which it was invoked. For example:

```
# which mpicc
/opt/lam-1.2.3/bin/mpicc
# switcher mpi = mpich-4.5.6 --system
# which mpicc
/opt/lam-1.2.3/bin/mpicc
# bash
# which mpicc
/opt/mpich-4.5.6/bin/mpicc
```

If you wish to have your current shell reflect the status of your switcher settings, you must run the "switcher-reload" command. For example:

```
# which mpicc
/opt/lam-1.2.3/bin/mpicc
# switcher mpi = mpich-4.5.6 --system
# which mpicc
/opt/lam-1.2.3/bin/mpicc
# switcher-reload
```

```
# which mpicc
/opt/mpich-4.5.6/bin/mpicc
```
Note that this is only necessary if you want to change your current environment. All new shells (including scripts) will automatically get the new switcher settings.

4.11.6 Setting a User Default

Setting a user-level default is essentially the same as setting the system-level default, except without the --system argument. This will set the user-level default instead of the system-level default:

```
$ switcher mpi = lam-1.2.3
```
Using the special name none will indicate that no module should be loaded for the mpi tag. It is most often used by users to specify that they do not want a particular software package loaded.

```
$ switcher mpi = none
```
Removing a user default (and therefore reverting to the system-level default) is done by removing the default attribute:

```
$ switcher mpi --show
user:default=mpich-1.2.4
system:exists=true
$ switcher mpi --rm-attr default
$ switcher mpi --show
system:default=lam-6.5.6
system:exists=true
```

4.11.7 Use switcher with care!

switcher immediately affects the environment of all future shell invocations (including the environment of scripts). To get a full list of options available, read the switcher(1) man page, and/or run switcher --help.

back to Table of Contents

5 Licenses and Copyrights

1. C3
2. Disable Services
3. LAM/MPI
4. Maui scheduler
5. MPICH
6. pFilter
7. PVM
8. SIS
9. Switcher
10. Torque

back to Table of Contents

5.1 C3 Copyright

The C3 OSCAR package contains the following copyrights and licenses.

The C3 package contains:

C3 version 4.0: Cluster Command & Control Suite Oak Ridge National Laboratory, Oak Ridge, TN, Authors: M.Brim, R.Flanery, G.A.Geist, B.Luethke, S.L.Scott (C) 2001 All Rights Reserved

NOTICE Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted provided that the above copyright notice appear in all copies and that both the copyright notice and this permission notice appear in supporting documentation. Neither the Oak Ridge National Laboratory nor the Authors make any representations about the suitability of this software for any purpose. This software is provided "as is" without express or implied warranty.

The C3 tools were funded by the U.S. Department of Energy.

back to Table of Contents

5.2 Disable Services License

The disable-services package contains the following copyrights and licenses.

Copyright (c) 2002 The Trustees of Indiana University.

All rights reserved.

Indiana University has the exclusive rights to license this product under the following license.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1) All redistributions of source code must retain the above copyright notice, the list of authors in the original source code, this list of conditions and the disclaimer listed in this license;

2) All redistributions in binary form must reproduce the above copyright notice, this list of conditions and the disclaimer listed in this license in the documentation and/or other materials provided with the distribution;

3) Any documentation included with all redistributions must include the following acknowledgement:

"This product includes software developed at the Pervasive Technology Labs at Indiana University. For technical information contact Andrew Lumsdaine at the Pervasive Technology Labs at Indiana University. For administrative and license questions contact the Advanced Research and Technology Institute at 1100 Waterway Blvd. Indianapolis, Indiana 46202, phone 317-274-5905, fax 317-274-5902."

Alternatively, this acknowledgement may appear in the software itself, and wherever such third-party acknowledgments normally appear.

4) The name "disable-services" or shall not be used to endorse or promote products derived from this software without prior written permission from Indiana University. For written permission, please contact Indiana University Advanced Research & Technology Institute.

5) Products derived from this software may not be called "disable-services", nor may "disable-services" appear in their name, without prior written permission of Indiana University Advanced Research & Technology Institute.

Indiana University provides no reassurances that the source code provided does not infringe the patent or any other intellectual property rights of any other entity. Indiana University disclaims any liability to any recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise.

LICENSEE UNDERSTANDS THAT SOFTWARE IS PROVIDED "AS IS" FOR WHICH NO WARRANTIES AS TO CAPABILITIES OR ACCURACY ARE MADE. INDIANA UNIVERSITY GIVES NO WARRANTIES AND MAKES NO REPRESENTATION THAT SOFTWARE IS FREE OF INFRINGEMENT OF THIRD PARTY PATENT, COPYRIGHT, OR OTHER PROPRIETARY RIGHTS. INDIANA UNIVERSITY MAKES NO WARRANTIES THAT SOFTWARE IS FREE FROM "BUGS", "VIRUSES", "TROJAN HORSES", "TRAP DOORS", "WORMS", OR OTHER HARMFUL CODE. LICENSEE ASSUMES THE ENTIRE RISK AS TO THE PERFORMANCE OF SOFTWARE AND/OR ASSOCIATED MATERIALS, AND TO THE PERFORMANCE AND VALIDITY OF INFORMATION GENERATED USING SOFTWARE.

Indiana University has the exclusive rights to license this product under this license.

back to Table of Contents

5.3 LAM/MPI License

The LAM/MPI OSCAR package contains the following copyrights and licenses. Software License for LAM/MPI

Copyright (c) 2001-2003 The Trustees of Indiana University.

All rights reserved.

Copyright (c) 1998-2001 University of Notre Dame. All rights reserved.

back to Table of Contents

5.5 MPICH License

The MPICH OSCAR package contains the following copyrights and licenses.

COPYRIGHT

The following is a notice of limited availability of the code, and disclaimer which must be included in the prologue of the code and in all source listings of the code.

Copyright Notice

+ 1993 University of Chicago + 1993 Mississippi State University

Permission is hereby granted to use, reproduce, prepare derivative works, and to redistribute to others. This software was authored by:

Argonne National Laboratory Group

1. Gropp: (630) 252-4318; FAX: (630) 252-5986; e-mail: gropp@mcs.anl.gov
2. Lusk: (630) 252-7852; FAX: (630) 252-5986; e-mail: lusk@mcs.anl.gov Mathematics and Computer Science Division Argonne National Laboratory, Argonne IL 60439 Mississippi State Group
3. Doss: (601) 325-2565; FAX: (601) 325-7692; e-mail: doss@erc.msstate.edu
4. Skjellum:(601) 325-8435; FAX: (601) 325-8997; e-mail: tony@erc.msstate.edu Mississippi State University, Computer Science Department & NSF Engineering Research Center for Computational Field Simulation P.O. Box 6176, Mississippi State MS 39762

GOVERNMENT LICENSE

Portions of this material resulted from work developed under a U.S. Government Contract and are subject to the following license: the Government is granted for itself and others acting on its behalf a paid-up, nonexclusive, irrevocable worldwide license in this computer software to reproduce, prepare derivative works, and perform publicly and display publicly.

DISCLAIMER

This computer code material was prepared, in part, as an account of work sponsored by an agency of the United States Government. Neither the United States, nor the University of Chicago, nor Mississippi State University, nor any of their employees, makes any warranty express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights.

back to Table of Contents

5.4 Maui Scheduler

The Maui Scheduler OSCAR package contains the following copyrights and licenses.

Maui Scheduler General Public License

This product includes software developed for The University of New Mexico High Performance Computing Education and Research Center for use in the Maui Scheduler software. Copyright (C) 2000 Science and Technology Corporation @ UNM, Software developed for The University of New Mexico. All Rights Reserved. Maui Scheduler is a trademark of Science & Technology Corporation @ UNM

THE SOFTWARE IS PROVIDED AS IS AND SCIENCE & TECHNOLOGY CORPORATION @ UNM (STC) AND THE UNIVERSITY OF NEW MEXICO (UNM) DISCLAIM ALL WARRANTIES RELATING TO THE SOFTWARE, WHETHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. NEITHER UNM, STC, NOR ANYONE INVOLVED IN THE CREATION, PRODUCTION, OR DELIVERY OF THE SOFTWARE SHALL BE LIABLE FOR ANY INDIRECT, CONSEQUENTIAL, OR INCIDENTAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE EVEN IF UNM OR STC HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR CLAIMS. IN NO EVENT SHALL UNM?S LIABILITY FOR ANY DAMAGES EXCEED THE CONSIDERATION PAID FOR THE LICENSE TO USE THE SOFTWARE, REGARDLESS OF THE FORM OF CLAIM. THE PERSON OR ENTITY USING THE SOFTWARE BEARS ALL RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOFTWARE.

By installing or using this software you are accepting a non-exclusive license to install, use, modify, and redistribute, and make modifications to the SOFTWARE ("LICENSE") pursuant to the following conditions:

1. All copies of the SOFTWARE, whether or not for redistribution and whether or not in source code or in binary form must include a conspicuous and appropriate publication of the above copyright notice and disclaimer.

2. Redistribution of the SOFTWARE in any form whatsoever, including parts of the code that are incorporated into other software programs, must include a conspicuous and appropriate publication of the following acknowledgement:

"This product includes software developed for The University of New Mexico High Performance Computing Education and Research Center for use in the Maui Scheduler software. Maui Scheduler is a trademark of Science & Technology Corporation @ UNM"

Any such modification of the SOFTWARE must, when installed, display the above language, the copyright notice, and the warranty disclaimer.

3. A copy of any modifications to the SOFTWARE beyond those necessary for the SOFTWARE to function with licensee?s system must be forwarded as source code to the Maui High Performance Computing Center at webmaster@mhpcc.edu. The forwarded modifications are to be accompanied with a brief explanation of why the modification was made and resulting performance of the modifications. Failure to do so renders the license invalid, as well as any licenses you grant to third parties subsequent to the incorporation of the modifications into the SOFTWARE.

4. Copyright owners of modifications to SOFTWARE hereby grant HPCERC a non-exclusive, royalty-free, world-wide, irrevocable right and license to install, use, distribute, sublicense, and prepare derivative works of said modifications.

5. The name "Maui Scheduler" or its variants must not otherwise be used to endorse or to promote products derived from the SOFTWARE without prior written permission from STC.

6. Products derived from or incorporating the SOFTWARE in whole or in part shall not contain as part of the product?s name any form of the terms "High Performance Computing Education and Research Center, "HPCERC", "Maui", "University of New Mexico", "UNM", "Maui High Performance Computing Center", or "MPHCC", unless prior written permission has been received from the University of New Mexico Patent Administration Office.

7. All advertising materials for products that use or incorporate features of the SOFTWARE must display the following acknowledgement:

"This product includes software developed for The University of New Mexico for use in the Maui Scheduler Software."

8. It is not required that you accept this LICENSE; however, if you do not accept the terms of this LICENSE, you are prohibited by law from installing, using, modifying or distributing the SOFTWARE or any of its derivative works. Therefore, by installing, using, modifying or distributing the SOFTWARE (or any of its derivative works), you have agreed to this LICENSE and have accepted all its terms and conditions.

9. Each time the SOFTWARE is redistributed (or any work based on the SOFTWARE), the recipient automatically receives this license from STC to copy, distribute or modify the SOFTWARE subject to these terms and conditions, and has the choice of accepting or declining the license. As the LICENSEE, you shall automatically provide the recipient with a copy of this license. Further restrictions are not to be imposed on recipients of the SOFTWAE by the LICENSEE beyond those expressly described herein. The LICENSEE is not responsible for enforcing compliance with this LICENSE by recipients.

10. If any portion of this LICENSE is held invalid or unenforceable under any particular circumstance, the balance of the LICENSE will continue to apply.

And the Moab package contains the following:

Moab Scheduling System General License

This product was created by the Supercluster Development Group,

Copyright (C) 1999-2003 Supercluster Development Group, all rights reserved. Moab Scheduling System (TM) is a trademark of Cluster Resources, Inc.

THE SOFTWARE IS PROVIDED AS IS AND THE SUPERCLUSTER DEVELOPMENT GROUP (SDG) AND ALL CONTRIBUTING PARTIES DISCLAIM ALL WARRANTIES RELATING TO THE SOFTWARE, WHETHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. NEITHER SDG NOR ANYONE INVOLVED IN THE CREATION, PRODUCTION, OR DELIVERY OF THE SOFTWARE SHALL BE LIABLE FOR ANY INDIRECT, CONSEQUENTIAL, OR INCIDENTAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE EVEN IF SDG HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR CLAIMS. IN NO EVENT SHALL SDG?S LIABILITY FOR ANY DAMAGES EXCEED THE CONSIDERATION PAID FOR THE LICENSE TO USE THE SOFTWARE, REGARDLESS OF THE FORM OF CLAIM. THE PERSON OR ENTITY USING THE SOFTWARE BEARS ALL RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOFTWARE.

By installing or using this software you are accepting a non-exclusive license to install, use, modify, and redistribute, and make modifications to the SOFTWARE ("LICENSE") pursuant to the following conditions:

1. Copyright

All copies of the SOFTWARE, in source code or in binary form, must include a conspicuous publication of the above copyright notice, this license, and the above disclaimer.

2. Usage

Source and/or binary forms of SOFTWARE may be freely used by any organization pursuant to the conditions of this license.

3. Distribution

Academic and government agencies may redistribute SOFTWARE subject to the condition that the distribution contain conspicuous publication of the following acknowledgement in both packaging and execution software:

"This product contains code developed by the Supercluster Development Group. Moab Scheduling System is a trademark of the Cluster Resources, Inc."

Commercial and other for profit organizations may not redistribute this code or derivations of this code in any form whatsoever, including parts of SOFTWARE incorporated into other software programs without express written permission from Cluster Resources, Inc.

4. Modifications

SOFTWARE may be freely modified as necessary to meet the needs of the licensee?s system. Licensees which wish to distribute their modifications (including government and academic institutions, and those with a written license to redistribute) must first send a copy of the modifications along with descriptive text to the Supercluster Development Group at support@supercluster.org. Those without license to redistribute may send modifications to the Supercluster Development Group for evaluation and possible incorporation into SOFTWARE. Failure to send a copy of distributed modifications renders the license invalid, as well as any licenses granted to third parties subsequent to the incorporation of the modifications into SOFTWARE.

Upon receipt of modification notice, Supercluster Development Group, may, at their disgression, send the submittor a nice hat or a short note stating in effect, ?Thank you very much. We really appreciate you making these changes. We hope you have a nice day."

Any such modification of the SOFTWARE must, when installed, display this license, the copyright notice, and the warranty disclaimer as described in paragraph 1.

5. Use of Modifications

Copyright owners of modifications to SOFTWARE hereby grant SDG a non- exclusive, royalty-free, world-wide, irrevocable right and license to install, use, distribute, sublicense, and prepare derivative works of said modifications.

6. Product Endorsement

The name "Moab Scheduling System" or its variants must not otherwise be used to endorse or to promote products derived from SOFTWARE without prior written permission from Cluster Resources, Inc.

7. Use of Trademarks

Products derived from or incorporating SOFTWARE in whole or in part shall not contain as part of the product?s name any form of the terms "Supercluster Development Group", "SDG", "Cluster Resources", "Moab", or "Moab Scheduling System" or "Moab Scheduler" unless prior written permission has been received from the Cluster Resources, Inc.

8. Advertising

All advertising materials for products that use or incorporate features of the SOFTWARE must display the following acknowledgement:

"This product includes software developed by The Supercluster Development Group for use in the Moab Scheduling System."

9. License Acceptance

It is not required that you accept this LICENSE; however, if you do not accept the terms of this LICENSE, you are prohibited by law from installing, using, modifying or distributing the SOFTWARE or any of its derivative works. Therefore, by installing, using, modifying or distributing the SOFTWARE (or any of its derivative works), you have agreed to this LICENSE and have accepted all its terms and conditions.

10. Perpetuation of License

Each time the SOFTWARE is redistributed (or any work based on the SOFTWARE), the recipient automatically receives this license from Cluster Resources, Inc. to copy, distribute or modify the SOFTWARE subject to these terms and conditions, and has the choice of accepting or declining the license. As the LICENSEE, you shall automatically provide the recipient with a copy of this license. Further restrictions are not to be imposed on recipients of the SOFTWAE by the LICENSEE beyond those expressly described herein. The LICENSEE is not responsible for enforcing compliance with this LICENSE by recipients.

11. License Enforcement

If any portion of this LICENSE is held invalid or unenforceable under any particular circumstance, the balance of the LICENSE will continue to apply.

back to Table of Contents

5.7 PVM License

The PVM OSCAR package contains the following copyrights and licenses.

> PVM version 3.4: Parallel Virtual Machine System University of Tennessee, Knoxville TN. Oak Ridge National Laboratory, Oak Ridge TN. Emory University, Atlanta GA. Authors: J. J. Dongarra, G. E. Fagg, G. A. Geist,

1. A. Kohl, R. J. Manchek, P. Mucci,
2. M. Papadopoulos, S. L. Scott, and V. S. Sunderam (C) 1997 All Rights Reserved

NOTICE Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted provided that the above copyright notice appear in all copies and that both the copyright notice and this permission notice appear in supporting documentation. Neither the Institutions (Emory University, Oak Ridge National Laboratory, and University of Tennessee) nor the Authors make any representations about the suitability of this software for any purpose. This software is provided ??as is?? without express or implied warranty. PVM version 3 was funded in part by the U.S. Department of Energy, the National Science Foundation and the State of Tennessee.

back to Table of Contents

5.8 SIS License

The SIS OSCAR package contains the following copyrights and licenses.

The SystemImager package contains: # Copyright (C) 1999-2001 Brian Elliott Finley # <brian.finley@baldguysoftware.com> The System Configurator and SystemInstaller packages contain: # Copyright (c) 2001 International Business Machines # This program is free software; you can redistribute it and/or modify # it under the terms of the GNU General Public License as published by # the Free Software Foundation; either version 2 of the License, or # (at your option) any later version. # This program is distributed in the hope that it will be useful, # but WITHOUT ANY WARRANTY; without even the implied warranty of # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU # General Public License for more details. # You should have received a copy of the GNU General Public License # along with this program; if not, write to the Free Software # Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 # USA The AppConfig? package contains: Copyright (C) 1998 Canon Research Centre Europe Ltd. All Rights Reserved. This module is free software; you can redistribute it and/or modify it under the same terms as Perl itself. The Perl-Tk package contains: # Copyright (c) 1992-1994 The Regents of the University of California. # Copyright (c) 1994 Sun Microsystems, Inc. # Copyright (c) 1995-1999 Nick Ing-Simmons. All rights reserved. # This program is free software; you can redistribute it and/or # modify it under the same terms as Perl itself, subject # to additional disclaimer in Tk/license.terms due to partial # derivation from Tk8.0 sources. The Syslinux package contains: SYSLINUX is Copyright 1994-2001 H. Peter Anvin, and is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, Inc., 675 Mass Ave, Cambridge MA 02139, USA; either version 2 of the License, or (at your option) any later version. The tftp-hp package contains: Copyright (c) 1983, 1993 The Regents of the University of California. All rights reserved.

back to Table of Contents

5.9 Switcher License

The Switcher OSCAR package contains the following copyrights and licenses.

The *env-switcher* package contains:

Software License for Env-switcher Copyright (c) 2002 The Trustees of Indiana University. All rights reserved. Indiana University has the exclusive rights to license this product under the following license. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1) All redistributions of source code must retain the above copyright notice, the list of authors in the original source code, this list of conditions and the disclaimer listed in this license; 2) All redistributions in binary form must reproduce the above copyright notice, this list of conditions and the disclaimer listed in this license in the documentation and/or other materials provided with the distribution; 3) Any documentation included with all redistributions must include the following acknowledgement: "This product includes software developed at the Pervasive Technology Labs at Indiana University. For technical information contact Andrew Lumsdaine at the Pervasive Technology Labs at Indiana University. For administrative and license questions contact the Advanced Research and Technology Institute at 1100 Waterway Blvd. Indianapolis, Indiana 46202, phone 317-274-5905, fax 317-274-5902." Alternatively, this acknowledgement may appear in the software itself, and wherever such third-party acknowledgments normally appear. 4) The name "Env-switcher" or shall not be used to endorse or promote products derived from this software without prior written permission from Indiana University. For written permission, please contact Indiana University Advanced Research & Technology Institute. 5) Products derived from this software may not be called "Env-switcher", nor may "Env-switcher" appear in their name, without prior written permission of Indiana University Advanced Research & Technology Institute. Indiana University provides no reassurances that the source code provided does not infringe the patent or any other intellectual property rights of any other entity. Indiana University disclaims any liability to any recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. LICENSEE UNDERSTANDS THAT SOFTWARE IS PROVIDED "AS IS" FOR WHICH NO WARRANTIES AS TO CAPABILITIES OR ACCURACY ARE MADE. INDIANA UNIVERSITY GIVES NO WARRANTIES AND MAKES NO REPRESENTATION THAT SOFTWARE IS FREE OF INFRINGEMENT OF THIRD PARTY PATENT, COPYRIGHT, OR OTHER PROPRIETARY RIGHTS. INDIANA UNIVERSITY MAKES NO WARRANTIES THAT SOFTWARE IS FREE FROM "BUGS", "VIRUSES", "TROJAN HORSES", "TRAP DOORS", "WORMS", OR OTHER HARMFUL CODE. LICENSEE ASSUMES THE ENTIRE RISK AS TO THE PERFORMANCE OF SOFTWARE AND/OR ASSOCIATED MATERIALS, AND TO THE PERFORMANCE AND VALIDITY OF INFORMATION GENERATED USING SOFTWARE. Indiana University has the exclusive rights to license this product under this license. The modules package contains: GNU GENERAL PUBLIC LICENSE Version 2, June 1991 Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

back to Table of Contents

5.10 Torque License

The Torque OSCAR package contains the following copyrights and licenses.

OpenPBS (Portable Batch System) v2.3 Software License Copyright (c) 1999-2002 Veridian Information Solutions, Inc. All rights reserved.
---------- For a license to use or redistribute the OpenPBS software under conditions other than those described below, or to purchase support for this software, please contact Veridian Systems, PBS Products Department

("Licensor") at: www.OpenPBS.org +1 650 967-4675 sales@OpenPBS.org 877 902-4PBS (US toll-free) ---------- This license covers use of the OpenPBS v2.3 software (the "Software") at your site or location, and, for certain users, redistribution of the Software to other sites and locations. Use and redistribution of OpenPBS v2.3 in source and binary forms, with or without modification, are permitted provided that all of the following conditions are met. After December 31, 2003, only conditions 3-6 must be met:

1. Commercial and/or non-commercial use of the Software is permitted provided a current software registration is on file at www.OpenPBS.org. If use of this software contributes to a publication, product, or service, proper attribution must be given; see www.OpenPBS.org/credit.html
2. Redistribution in any form is only permitted for non-commercial, non-profit purposes. There can be no charge for the Software or any software incorporating the Software. Further, there can be no expectation of revenue generated as a consequence of redistributing the Software.
3. Any Redistribution of source code must retain the above copyright notice and the acknowledgment contained in paragraph 6, this list of conditions and the disclaimer contained in paragraph 7.
4. Any Redistribution in binary form must reproduce the above copyright notice and the acknowledgment contained in paragraph 6, this list of conditions and the disclaimer contained in paragraph 7 in the documentation and/or other materials provided with the distribution.
5. Redistributions in any form must be accompanied by information on how to obtain complete source code for the OpenPBS software and any modifications and/or additions to the OpenPBS software. The source code must either be included in the distribution or be available for no more than the cost of distribution plus a nominal fee, and all modifications and additions to the Software must be freely redistributable by any party (including Licensor) without restriction.
6. All advertising materials mentioning features or use of the Software must display the following acknowledgment: "This product includes software developed by NASA Ames Research Center, Lawrence Livermore National Laboratory, and Veridian Information Solutions, Inc. Visit www.OpenPBS.org for OpenPBS software support, products, and information."
7. DISCLAIMER OF WARRANTY THIS SOFTWARE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND. ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT ARE EXPRESSLY DISCLAIMED. IN NO EVENT SHALL VERIDIAN CORPORATION, ITS AFFILIATED COMPANIES, OR THE U.S. GOVERNMENT OR ANY OF ITS AGENCIES BE LIABLE FOR ANY DIRECT OR INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. This license will be governed by the laws of the Commonwealth of Virginia, without reference to its choice of law rules.

back to Table of Contents

5.6 pfilter License

The pfilter OSCAR package contains the following copyrights and licenses.

% Copyright 2002 Neil Gorsuch

%

% pfilter is free software; you can redistribute it and/or modify

% it under the terms of the GNU General Public License as published by

% the Free Software Foundation; either version 2 of the License, or

% (at your option) any later version.

%

% pfilter is distributed in the hope that it will be useful,

% but WITHOUT ANY WARRANTY; without even the implied warranty of

% MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

% GNU General Public License for more details.

%

% You should have received a copy of the GNU General Public License

% along with this program; if not, write to the Free Software

% Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA