

OSCAR Cluster User's Guide
Software Version 3.0
Documentation Version 3.0

<http://oscar.sourceforge.net/>
oscar-users@lists.sourceforge.net

The Open Cluster Group
<http://www.openclutergroup.org/>

November 25, 2003

Contents

1	Introduction	4
2	Individual Packages	4
2.1	Using C3	4
2.1.1	Overview	4
2.1.2	Basic File Format	5
2.1.3	Specifying ranges	6
2.1.4	Machine Definitions	6
2.1.5	Other Considerations	7
2.2	HDF5	7
2.3	Using LAM/MPI	7
2.3.1	Notes about OSCAR's LAM/MPI Setup	8
2.3.2	Setting up <code>switcher</code> to use LAM/MPI	8
2.3.3	General Usage	8
2.3.4	More Information	9
2.4	The OSCAR Password Installer and User Management (OPIUM)	9
2.5	Using The Portable Batch System (PBS)	10
2.5.1	PBS Commands	10
2.5.2	Submitting a PBS Job	10
2.6	Packet Filtering with <code>pfilter</code>	11
2.7	Parallel Virtual Machine (PVM)	12
2.7.1	Using PVM	12
2.7.2	Other details	14
2.8	An overview of SIS	14
2.8.1	Building a SIS image	14
2.8.2	Managing SIS clients	15
2.8.3	Maintaining your client software	15
2.8.4	Additional information	15
2.9	An overview of <code>switcher</code>	15
2.9.1	The <code>modules</code> package	16
2.9.2	The <code>env-switcher</code> package	17
2.9.3	Which MPI do you want to use?	18
2.9.4	Setting the system-level default	18
2.9.5	Setting the user-level default	19
2.9.6	Use <code>switcher</code> with care!	19
3	Package Licenses and Copyrights	19
3.1	C3	20
3.2	Disable Services	21
3.3	HDF5	22
3.4	LAM/MPI	24
3.5	MPICH	26
3.6	PBS	27
3.7	<code>pfilter</code>	28
3.8	PVM	29
3.9	SIS	29

3.10 Switcher 31

1 Introduction

OSCAR version 3.0 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 cluster computing. Everything needed to install, build, maintain, and use a modest sized Linux cluster is included in the suite, making it unnecessary to download or even install any individual software packages on your cluster.

OSCAR is the first project by the Open Cluster Group. For more information on the group and its projects, visit its website <http://www.OpenClusterGroup.org/>.

More information about OSCAR can be found at the OSCAR home page:

<http://oscar.sourceforge.net/>

Section 2 contains the user installation for OSCAR clusters. Cluster users will find general instructions on how to use the software installed on the OSCAR cluster. Additionally, the licenses and copyrights of each of the individual software packages contained in OSCAR are included in Section 3.

2 Individual Packages

The following sections provide general user-level information on the packages installed on OSCAR clusters.

2.1 Using C3

2.1.1 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. To invoke the serial version of `cexec`, type `cexecs` instead of `cexec`. For more information on how to use each tool, see the man page for the specific tool.

2.1.2 Basic 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 1.

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

2.1.3 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!

2.1.4 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.

2.1.5 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`).

2.2 HDF5

HDF5 is a Hierarchical Data Format product consisting of a data format specification and a supporting library implementation.

HDF5 includes the following features which make it a widely-used scientific data format:

- supports a user-defined hierarchical grouping structure with varied datatypes and attributes to organize large and varied data in research, development, and production environments,
- runs on parallel computing platforms using MPI I/O (note that the compiled version of HDF5 supplied in OSCAR currently only supports serial operation),
- supports an all-encompassing variety of datatypes, including compound and user-defined datatypes,
- supports very large files (theoretically terabyte and larger, though the outer limits have not been tested)

HDF5 is fully documented – see `/opt/hdf5-oscar-1.4.4-post2/doc/hdf5/index.html` for the documentation that accompanied the release of HDF5 that is installed with this OSCAR distribution. For documentation of the current release of HDF5 as served from the HDF Group’s website, see <http://hdf.ncsa.uiuc.edu/HDF5/doc/>.

For further information regarding HDF5, such as lists of current users, supporting agencies, and applications that employ the library, see <http://hdf.ncsa.uiuc.edu/HDF5/>.

OSCAR installs the HDF5 library at `/opt/hdf5-oscar-1.4.4-post2/lib/`. Please note that the HDF5 library included in OSCAR is presently only available for serial operation. Parallel operation will be enabled in a future version of OSCAR.

2.3 Using LAM/MPI

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 cumulative record of communication, which can be visualized either at runtime or post-mortem.

2.3.1 Notes about OSCAR's LAM/MPI Setup

The OSCAR environment is able to have multiple MPI implementations installed simultaneously – see Section 2.9 (page 15) 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.

2.3.2 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-6.5.9”), 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-6.5.9”) and run the following command:

```
$ switcher mpi = lam-6.5.9
```

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.

2.3.3 General Usage

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

1. 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
```

2. Repeat the following steps as many times as necessary:

- (a) Use the MPI “wrapper” compilers (`mpicc` for C programs, `mpic++` 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
```

- (b) 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.

- (c) 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.

3. 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 `gfortran`, 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).

2.3.4 More Information

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!

2.4 The OSCAR Password Installer and User Management (OPIUM)

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.

2.5 Using The Portable Batch System (PBS)

The PBS architecture consists of three major components:

- The PBS server. This runs on the OSCAR head node. It controls the submission and running of jobs.
- The Maui scheduler. This takes care of scheduling jobs across the cluster according sophisticated algorithms.
- A “mom” daemon on each cluster node. The moms are responsible for actually starting and stopping jobs on the client nodes.

2.5.1 PBS Commands

All PBS commands can be found under `/usr/local/pbs/bin` on the OSCAR head node. There are man pages available for these commands, but here are the most popular with some basic options:

- `qsub`: submits job to PBS
- `qdel`: deletes PBS job
- `qstat [-n]`: displays current job status and node associations
- `pbsnodes [-a]`: displays node status
- `pbsdsh`: distributed process launcher

2.5.2 Submitting a PBS 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.
- Job parameters can be specified to `qsub` on the command line, or within the submitted script. You can get a good start by looking at examples provided by the OSCAR test suite. Ask your system administrator if you would like to see these. They can likely be found in the home directory of the “`oscartst`” user.

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:

```
$ 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).

2.6 Packet Filtering with `pfilter`

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.

2.7 Parallel Virtual Machine (PVM)

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.

2.7.1 Using PVM

The default OSCAR installation tests PVM via a PBS job (see also: Section 2.5 on page 10). However, some users may choose to use PVM outside of this context so a few words on usage may be helpful ¹.

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 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 "oscardnode1".
sgrundy: $ pvm
pvm> add oscarndode1
add oscarndode1
```

¹Note, the examples in this section assume a shared filesystem, as is used with OSCAR.

```

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 command-line. 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/hello-example` 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: $

```

2.7.2 Other details

The OSCAR installation of PVM makes use of the `env-switcher` package (also see Section 2.9, page 15). 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 non-interactive 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 2.7.1. 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/*`

2.8 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.

2.8.1 Building a SIS 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.

2.8.2 Managing SIS clients

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`.

2.8.3 Maintaining your 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 update².

2.8.4 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/>.

2.9 An overview of switcher

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

²In OSCAR, when using `updateclient` to maintain a cluster, the last step of the wizard must be re-run to keep the system configurations in sync, i.e., re-run “Complete Cluster Setup”.

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 inadvertently 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.

2.9.1 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).

³Note, 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`.

2.9.2 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.

`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:

1. `switcher --list`
List all available tags.
2. `switcher <tag> --list`
List all defined attributes for the tag `<tag>`.

3. `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.

4. `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.

5. `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 2.9.3 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.

2.9.3 Which MPI do you want to use?

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.

2.9.4 Setting the 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 propagated 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/opium/bin/sync_users` 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
```

2.9.5 Setting the user-level 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
```

2.9.6 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`.

3 Package Licenses and Copyrights

Since OSCAR includes many packages, the licenses and copyrights for each of them is included here for reference.

3.1 C3

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.

The Python package contains:

This is Python version 2.1.1
=====

Copyright (c) 2001 Python Software Foundation.
All rights reserved.

Copyright (c) 2000 BeOpen.com.
All rights reserved.

Copyright (c) 1995-2001 Corporation for National Research Initiatives.
All rights reserved.

Copyright (c) 1991-1995 Stichting Mathematisch Centrum.
All rights reserved.

License information

See the file "LICENSE" for information on the history of this software, terms & conditions for usage, and a DISCLAIMER OF ALL WARRANTIES.

This Python distribution contains no GNU General Public Licensed (GPLed) code so it may be used in proprietary projects just like prior Python distributions. There are interfaces to some GNU code but these are entirely optional.

All trademarks referenced herein are property of their respective holders.

3.2 Disable Services

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.

3.3 HDF5

The HDF5 OSCAR package contains the following copyrights and licenses.

Copyright Notice and Statement for NCSA Hierarchical Data Format (HDF) Software Library and Utilities

NCSA HDF5 (Hierarchical Data Format 5) Software Library and Utilities
Copyright 1998, 1999, 2000, 2001, 2002 by the Board of Trustees of the University of Illinois. All rights reserved.

Contributors: National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana-Champaign (UIUC), Lawrence Livermore National Laboratory (LLNL), Sandia National Laboratories (SNL), Los Alamos National Laboratory (LANL), Jean-loup Gailly and Mark Adler (gzip library).

Redistribution and use in source and binary forms, with or without modification, are permitted for any purpose (including commercial purposes) provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the following disclaimer in the documentation and/or materials provided with the distribution.
3. In addition, redistributions of modified forms of the source or binary code must carry prominent notices stating that the original code was changed and the date of the change.
4. All publications or advertising materials mentioning features or use of this software are asked, but not required, to acknowledge that it was developed by the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign and to credit the contributors.
5. Neither the name of the University nor the names of the Contributors may be used to endorse or promote products derived from this software without specific prior written permission from the University or the Contributors, as appropriate for the name(s) to be used.
6. THIS SOFTWARE IS PROVIDED BY THE UNIVERSITY AND THE CONTRIBUTORS "AS IS" WITH NO WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED. In no event shall the University or the Contributors be liable for any damages suffered by the users arising out of the use of this software, even if advised of the possibility of such damage.

Portions of HDF5 were developed with support from the University of California, Lawrence Livermore National Laboratory (UC LLNL). The following statement applies to those portions of the product and must be retained in any redistribution of source code, binaries, documentation, and/or accompanying materials:

This work was partially produced at the University of California,

Lawrence Livermore National Laboratory (UC LLNL) under contract no. W-7405-ENG-48 (Contract 48) between the U.S. Department of Energy (DOE) and The Regents of the University of California (University) for the operation of UC LLNL.

DISCLAIMER:

This work was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any 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. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

3.4 LAM/MPI

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.

Copyright (c) 1994-1998 The Ohio State 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 Ohio Supercomputer Center at The Ohio State University, the University of Notre Dame and the Pervasive Technology Labs at Indiana University with original ideas contributed from Cornell 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 "LAM" or "LAM/MPI" 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 "LAM" or "LAM/MPI", nor may "LAM" or "LAM/MPI" 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.

3.5 MPICH

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

W. Gropp: (630) 252-4318; FAX: (630) 252-5986; e-mail:
gropp@mcs.anl.gov

E. 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

N. Doss: (601) 325-2565; FAX: (601) 325-7692; e-mail:
doss@erc.msstate.edu

A. 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.

3.6 PBS

The OpenPBS 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.

3.7 pfilter

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
```

3.8 PVM

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,
          J. A. Kohl, R. J. Manchek, P. Mucci,
          P. 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.

3.9 SIS

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.

3.10 Switcher

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.

...