How to install an OSCAR cluster

Software Version 1.0
Documentation Version 1.0

By: The Open Cluster Group
www.OpenClusterGroup.org
oscar@OpenClusterGroup.org

I. Introduction

The OSCAR cluster installation HowTo is provided as an installation guide to users, as well as a detailed explanation of what is happening as you install. This document does not describe what OSCAR is however. For an overview of OSCAR and the intentions behind it see the `oscar_introduction` document located in the *docs* subdirectory. A list of software and hardware requirements for OSCAR can be found in the `oscar_requirements` document as well. In addition, there is no "Quick Start" guide for OSCAR. Due to the complicated nature of putting together a high-performance cluster, it is strongly suggested that you read this document straight through, without skipping any sections, as it has been organized to give a thorough understanding to users of all experience levels. Novice users will be comforted to know that anyone who has installed and used Linux can successfully navigate through the OSCAR cluster install. Although experienced cluster users & administrators may be tempted to skip ahead, they will only end up coming back to the earlier sections to answer their many questions.

Let's start with a few basic terms and concepts, so that everyone is starting out on the same level. The most important is the term *cluster*, which when mentioned herein refers to a group of individual computers bundled together using hardware and software in order to make them work as a single machine. Each individual machine of a cluster is referred to as a *node*. Within the OSCAR cluster to be installed, there are two types of nodes, *server* and *client*. A *server* node is responsible for servicing the requests of *client* nodes. A *client* node is dedicated to computation. The OSCAR cluster to be installed will consist of one server node and a number of client nodes, where all the client nodes have homogeneous hardware. The software contained within OSCAR does support doing multiple cluster installs from the same server, but no documentation is provided on how to do so.

The rest of this document is organized as follows. First, an overview is given for the installation software used in OSCAR, known as LUI. Second, an outline is given of the entire cluster install procedure, so that users have a general understanding of what they will be doing. Next, the cluster installation procedure is presented in much detail. The level of detail lies somewhere between "the install will now update some files" and "the install will now replace the string 'xyz' with 'abc' in file some_file." The reasoning behind providing this level of detail is to allow users to fully understand what it takes to put together a cluster, as well as to allow them to help troubleshoot any problems, should some arise. Last, but certainly not least, are the appendices. *Appendix A* covers the topic of network booting client nodes, which is so important that it deserved its own section. *Appendix B* gives instructions for creating initial ramdisks. *Appendix C* provides curious users an overview of what really happens during a client install. Finally, *Appendix D* tackles troubleshooting, providing fixes to known problems and where to find help for unknown problems.

II. Overview of LUI

The first question you may have is what is LUI. The Linux Utility for cluster Install (LUI), is a cluster installation tool from the open-source section of IBM. The main reason LUI was chosen as the installation mechanism was that it does not require that client nodes already have Linux installed. Nor does it require a disk image of a client node, which is the case for other installation techniques. LUI also has many other distinguishing features that make it the mechanism of choice. The most highly used quality of LUI in the OSCAR install is the cluster information database that it maintains. The database contains all the information on each node needed to both install and configure the cluster. A second desirable quality is that LUI makes use of the Red Hat Package Manager (RPM) standard for software installation, which simplifies software installation tremendously. Another quality, which up to this point has not been taken advantage of in OSCAR, is the heterogeneous nature of LUI, allowing clients to contain not only heterogeneous hardware, but heterogeneous software as well. An obvious application of

the future use of this quality is in the installation of heterogeneous clusters, allowing certain clients to be used for specialized purposes.

In order to understand some of the steps in the upcoming install, you will need knowledge of the main concepts used within LUI. The first concept is that of a machine object. In LUI, a machine object is defined for each of your cluster nodes. There are two types of machine objects, server and client, corresponding to the two cluster node types. The server machine is responsible for creating the cluster information database and for servicing client installation requests. There are four pieces of information that are kept for the server machine: name, IP address and corresponding netmask, and the default client reboot behavior. Note that the IP and netmask are for the internal cluster subnet, not the server machine's external network. The client machines are the ones to be installed. In addition to the information that is kept for the server, the following information is kept for each client: long hostname, MAC address, default route, default gateway, number of processors, and a PBS string. Each of these pieces of information will be discussed further as part of the detailed install procedure. The second concept is that of a resource object. Resource objects are used to characterize the essential things that define a client machine, and are the key component to support for heterogeneous machines in LUI. There are several types of resource objects, including *disk, file, kernel, map, postinstall, ramdisk, rpm, source*, and *exit*. A description of each type follows:

disk – a disk table used for partitioning the client hard drive and specifying network mounted file systems
file – a file system, such as `/home` or `/usr`
kernel – a custom kernel
map – a system map file to be used with a custom kernel
postinstall – a user script to run after the client is installed but before it is rebooted from local disk
ramdisk – an initial ramdisk, used for configuring special hardware not supported by the kernel at boot
rpm – a list of RPMs to install on the client
source – a file to be copied from the server to the client after installing the RPMs
exit – a user exit script that is run on the first boot after client installation

By allocating these resources to client machines, LUI enables custom clients to be built. Within LUI, there is also the ability to create groups of clients and groups of resources. With such a facility, you are able to assign a group of resources to a group of clients in one swift action. For a homogenous cluster such as the OSCAR cluster to be installed, this is a very useful mechanism.

For additional information on the concepts in LUI and how to use it, you should refer to the html documentation installed with LUI during the OSCAR install. In addition, you can visit the LUI web site at http://oss.software.ibm.com/lui for recent updates.

III. Outline of Cluster Installation Procedure

Notice: You do not perform many of the steps in this outline, as they are automated by the install. It will be clear in the detailed procedure what exactly you are expected to do.

A. Server Installation & Configuration
   1. install Linux on server machine
   2. get OSCAR distribution
   3. configure ethernet adapter for cluster
   4. create needed directories
B. Initial OSCAR Server Configuration
   1. creates OSCAR directories
   2. installs necessary software
      - LUI

- services:
  a) Network File System (NFS)
  b) Dynamic Host Configuration Protocol (DHCP)
  c) Trivial File Transfer Protocol (tftp-hpa)
- Syslinux (for network booting using PXE)
- Etherboot (for network booting using floppies)

3. updates some system files
4. updates system startup scripts

C. Cluster Definition
1. define server
2. collect client MAC addresses
3. define clients
4. define resources
5. allocate resources to clients

D. Secondary OSCAR Server Configuration
1. uses cluster information database to update DHCP configuration
2. installs ORNL's Cluster Command & Control (C3) tools
3. installs VA's SystemImager

E. Client Installations
1. network boot clients to start install
2. check log for completion status
3. reboot each client when finished installing

F. Cluster Configuration
1. updates hosts files on server & clients
2. configures user rsh capabilities
3. configures ssh on server & clients
4. installs & configures the Message Passing Interface (MPI-CH) from Argonne National Laboratory
5. installs & configures the Parallel Virtual Machine (PVM) from Oak Ridge National Laboratory
6. updates clients' C3 configuration
7. installs Veridian's Portable Batch System (PBS)
8. updates clients' PBS configuration
9. synchronizes clients' date & time with server

IV. Detailed Cluster Installation Procedure

Note: All actions specified herein should be performed by the *root* user.

A. Server Installation & Configuration

During this phase, you will prepare the machine to be used as the server for using OSCAR.

1. **Install Linux on the server machine.** *If you have a machine you want to use that already has Linux installed, you may use it and continue with step two*. When installing Linux, it is required that you use a distribution that is based upon the RPM standard. Furthermore, it should be noted that all testing up to this point has been done using the Red Hat 6.2 distribution. As such, use of distributions other than Red Hat 6.2 will require a porting of OSCAR, as many of the scripts and software within OSCAR are dependent on Red Hat. Do not worry about doing a custom install, as OSCAR contains all the software on which it

depends. The only other installation requirement is that some X environment such as GNOME or KDE must be installed. Therefore, a typical workstation install is sufficient.

2. **Get a copy of OSCAR and unpack on the server.** If you are reading this, you probably already have a copy. If not, go to http://www.csm.ornl.gov/oscar and download the latest OSCAR tarball, which will be named something like **`oscar-`*`version`*`.tgz`**. The version used in these instructions is 1.0, which you should replace with the version you download in any of the sample commands. Copy the OSCAR tarball to a directory such as */mnt* or */tmp* on your server. There is no required installation directory, except that you may not use */usr/local/oscar*, which is reserved for special use. Do **not** unpack the tarball on a Windows based machine and copy the directories over to the server, as this will convert all the scripts to the dreaded 'DOS' format and will render them useless under Linux. Assuming you placed the OSCAR tarball in */mnt*, open a command terminal and issue the following commands to unpack OSCAR:

```
cd /mnt
tar –zxvf oscar-1.0.tgz
```

The result is the creation of an OSCAR directory structure that is laid out as follows: (again assuming */mnt*)

/mnt/OSCAR-1.0/	- the base OSCAR directory
/mnt/OSCAR-1.0/c3/	- contains files for C3 installation
/mnt/OSCAR-1.0/benchmarks	- contains the OSCAR Benchmarks
/mnt/OSCAR-1.0/docs/	- OSCAR documentation directory
/mnt/OSCAR-1.0/install_cluster	- main installation script
/mnt/OSCAR-1.0/lui/	- contains files for LUI installation
/mnt/OSCAR-1.0/oscarResources/	- contains sample OSCAR resources
/mnt/OSCAR-1.0/oscarRPM/	- contains RPMs for software installed
/mnt/OSCAR-1.0/pbs/	- contains files for PBS installation
/mnt/OSCAR-1.0/prog_env/	- contains files for MPI & PVM installations
/mnt/OSCAR-1.0/README.first	- README first document
/mnt/OSCAR-1.0/scripts/	- contains scripts that do most of the work
/mnt/OSCAR-1.0/testing/	- contains OSCAR Cluster Test software

3. **Configure the ethernet adapter for the cluster.** Assuming you want your server to be connected to both an external network and the internal cluster subnet, you will need to have two ethernet adapters installed in the server. It is preferred that you do this because exposing your cluster may be a security risk, and certain software used in OSCAR such as DHCP may conflict with your external network. Once both adapters have been physically installed and you have booted Linux into an X environment, open a terminal and enter the command:

```
netconf &
```

The network configuration utility will be started, which you will use to configure your network adapters.

Press the **`Basic Host Information`** button to bring up the server's current network configuration. At this point, the server's hostname should have been filled in by the standard Linux installation. Additionally, the network information for your external ethernet adapter will have been completed, which can be verified by clicking on the **`Adapter`** tabs. Next, find

the tab for the newly added adapter, on which the **Net device** and **Kernel module** fields should already be filled in as a result of the Linux hardware detection facilities. If no such tab exists, go to the first **Adapter** tab that contains no information. Enter the following required information for your server: internal hostname and domain in the **Primary name + domain** field, a private IP address[1] and corresponding netmask[2]. If not already completed, select the appropriate device in the **Net device** field. If your external adapter is configured on device '*eth0*', then you should most likely select '*eth1*' as the device, assuming you have no other adapters installed. You must also specify the **Kernel module** for the internal ethernet adapter. If the internal adapter is the same type as the external one, use the module specified on that adapter's tab. Press the **Accept** button at the bottom to complete the change, press **Quit** on the main screen, and select **Activate the changes** when prompted.

Now reboot your machine to ensure that all the changes are propagated to the appropriate configuration files. To confirm that all ethernet adapters are in the **UP** state, once the machine has rebooted, open another terminal window and enter the following command:

```
/sbin/ifconfig –a
```

You should see **UP** as the first word on the third line of output for each adapter. If not, there is a problem that you need to resolve before continuing. Typically, the problem is that the wrong module is specified for the given device. Try using the network configuration utility again to resolve the problem.

4. **Create some necessary directories.** You need to create the **/tftpboot** and **/tftpboot/rpm** directories if they don't already exist. These directories will hold all the information needed for LUI to install the client nodes. As a result, the directories need to be placed on a partition that will have sufficient free space. A good estimate to the amount of space required is 650MB for the RPMs plus 5MB for each client. You can check the amount of free space on your drive's partitions by issuing the command **df –h** in a terminal. The result for each file system is located below the **Avail** heading. If your root (**/**) partition has enough free space, enter the following commands in a terminal:

   ```
   mkdir /tftpboot
   mkdir /tftpboot/rpm
   ```

   If your root partition does not have enough free space, create the directories on a different partition that does have enough free space and create links to them from the root (**/**) directory. For example, if the partition containing **/usr** contains enough space, you could do so by using the following commands:

   ```
   mkdir /usr/tftpboot
   mkdir /usr/tftpboot/rpm
   ln –s /usr/tftpboot /tftpboot
   ```

[1] There are three private IP address ranges: 10.0.0.0 to 10.255.255.255; 172.16.0.0 to 172.31.255.255; and 192.168.0.0 to 192.168.255.255. Additional information on private internets is available in RFC 1918.
[2] The netmask 255.255.255.0 should be sufficient for all OSCAR clusters.

5. **Copy distribution RPMs to /tftpboot/rpm.** In this step, you need to copy the RPMs included with your distribution into the **`/tftpboot/rpm`** directory. Insert and mount the CD for your Linux distribution using the following command:

   ```
   mount /dev/cdrom
   ```

   Locate the directory that contains the RPMs. In Red Hat, the RPMs are located on the distribution CD in the *`RedHat/RPMS`* directory. After locating the RPMs, copy them into **`/tftpboot/rpm`** with a command such as:

   ```
   cp  /mnt/cdrom/RedHat/RPMS/*.rpm  /tftpboot/rpm
   ```

   You may now unmount and eject your distribution CD.

B. Initial OSCAR Server Configuration

During this phase, the software needed to run OSCAR will be installed on the server. In addition, some initial server configuration will be performed.

1. **Change to the OSCAR directory and run 'install_cluster'.** If the OSCAR directory was placed in */tmp* for example, you would issue the following commands:

   ```
   cd  /tmp/OSCAR_1.0
   ./install_cluster ethernet-device
   ```

   In the above command, substitute the device name (e.g., eth0) for your server's internal ethernet adapter. The *`install_cluster`* script will first run the part one server configuration script, which does the following:

   a) creates the OSCAR directories */usr/local/oscar, /usr/local/oscar/lui, & /usr/local/oscar/lui_sources*
   b) installs system services (NFS, DHCP, tftp-hpa)
   c) installs syslinux (for network booting using PXE)
   d) installs LUI in */usr/local/lui*
   e) updates */etc/hosts* with OSCAR aliases
   f) updates */etc/exports* and restarts NFS
   g) updates */etc/inetd.conf* to use tftp-hpa and restarts the INET daemon
   h) updates system startup *(/etc/rc.d/init.d)* scripts

   If the part one script finishes successfully, *`install_cluster`* will then start the OSCAR wizard. The wizard, as shown in Figure 1, is provided to guide you through the rest of the cluster installation. 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 must 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.

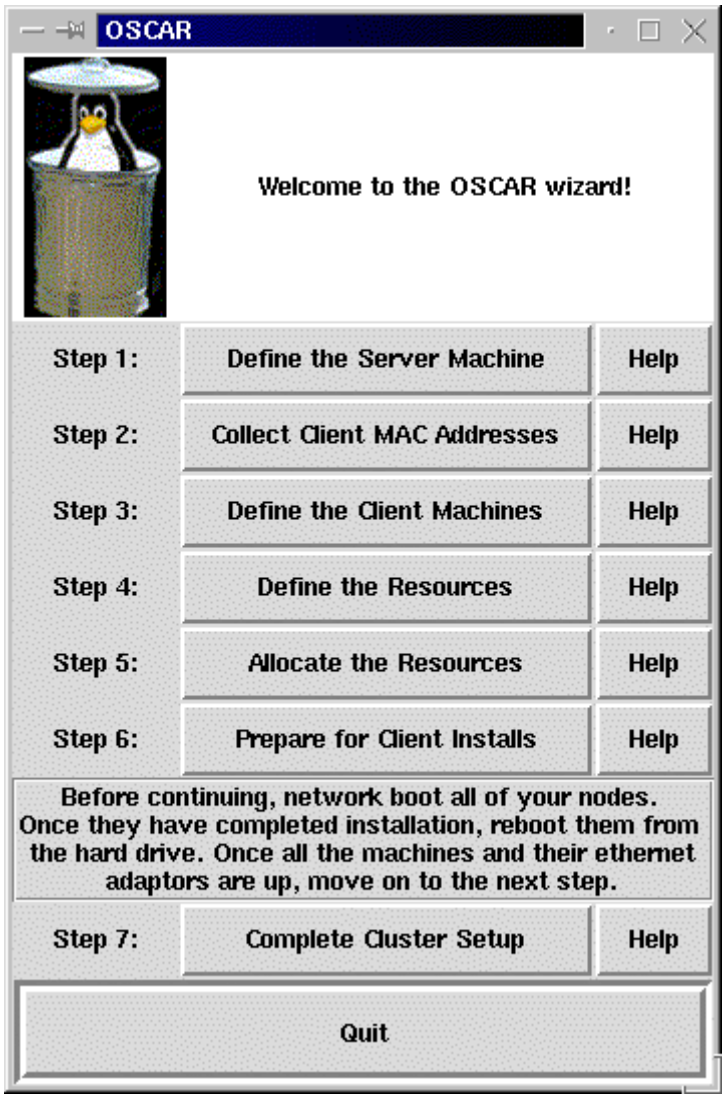

Figure 1: OSCAR Wizard

As each of the steps are performed, there is output generated that is displayed to the user. Currently, there are two very annoying properties of these output windows. The first is that the output is not shown until each command has completed in full. As some commands take minutes to complete, you may think that nothing is happening, when in fact the command is still running. In the instructions to follow, a notice will be given for steps where the output will not be displayed immediately. The second annoyance is that the standard output and standard error streams are displayed disjointedly, with the standard output being displayed first followed by the standard error. This causes problems in relating error messages to actions, as well as confuses users into thinking that some of the operations failed, since the error stream is displayed last. As a result, you should look through the output intently for completion messages before assuming an operation has failed.

C. Cluster Definition

During this phase, you will complete steps one through five of the OSCAR wizard in defining your cluster.

1. **Define the server machine.** Press the Step 1 button of the wizard entitled **`Define the Server`**. In the dialog box that is displayed, enter a name for the LUI server machine. For consistency purposes, you should use the same name as was used when configuring your internal ethernet adapter. Next enter the internal server IP address and cluster netmask. The last piece of information you need to provide is the default reboot action for all clients associated with the server after they have successfully completed their installation. You can either have them automatically reboot by selecting 'true', or show a prompt by selecting 'false.' It is important to note that if you wish to use automatic reboot, you should make sure the BIOS on each client is set to boot from the local hard drive before attempting a network boot by default. If you have to change the boot order to do a network boot before a disk boot to install your client machines, you should not use automatic reboot. Press the **`Apply`** button when finished. A sample dialog with input and successful output is shown in Figure 2. Press the **`Close`** button to complete the step.

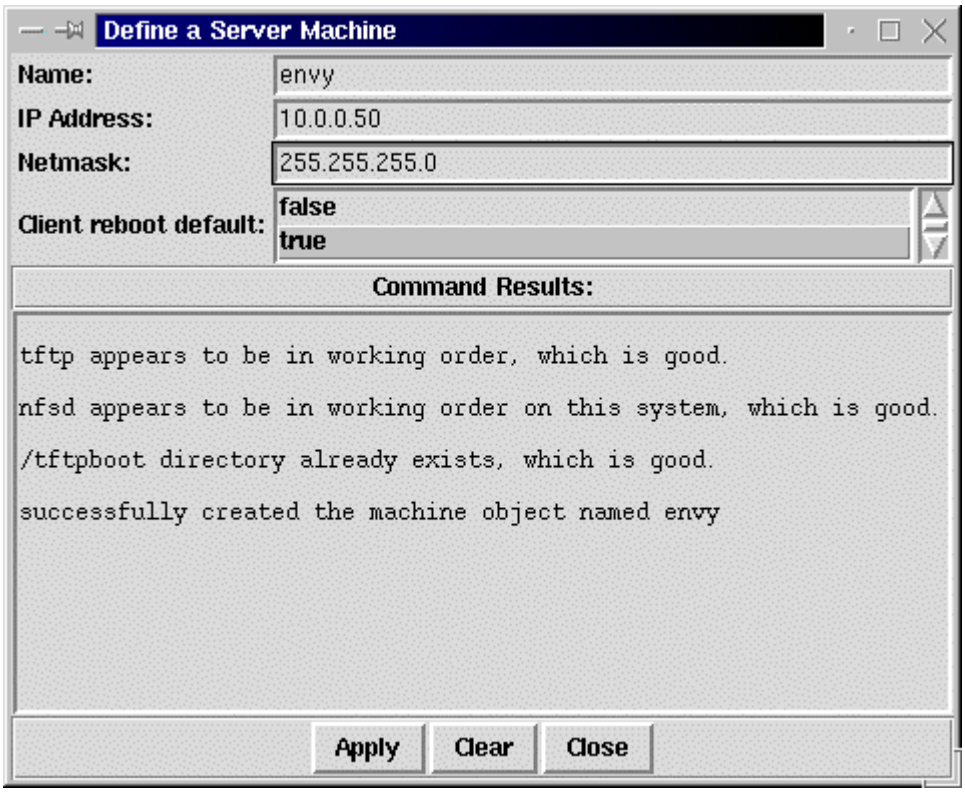

Figure 2: Define the Server

2. **Collect client MAC addresses.** What you do at this time is dependent upon whether you already know the MAC addresses of your clients or not. The MAC address of a client is a twelve hex-digit hardware address embedded in the client's ethernet adapter. MAC addresses look like 00:0A:CC:01:02:03, as opposed to the familiar format of IP addresses.

   a) If you do know the addresses, you should edit the file */etc/MAC.info* using your favorite editor. The format for the file is one line for each client, where each line is of the form:

node-id MAC-address

The node id is simply a symbolic tag used by different collection tools to refer to the MAC address. Typically, one would use a node id such as 'node0'. Enter a node id and MAC address for each of your clients. Be sure to write down the node id for the first client you enter, which will be used later in defining the client machines.

b) If you do not know the client MAC addresses, or do not want to edit the */etc/MAC.info* file manually, press the Step 2 button of the wizard entitled **Collect Client MAC Addresses**. The OSCAR MAC address collection utility dialog box will be displayed. To start the collection, press the **Collect** button and then network boot the first client. Follow the directions given in the dialog output window to collect the rest of your clients' addresses. Please do not press **Collect** more than needed, as each time it is pressed the utility will wait until a valid MAC is collected before continuing. In order to use this tool, you will need to know how to network boot your client nodes. For instructions on doing so, see *Appendix A*. A sample dialog that has been used to collect four client MAC addresses is shown in Figure 3. When you have collected the addresses for all your client nodes, press **Done**.

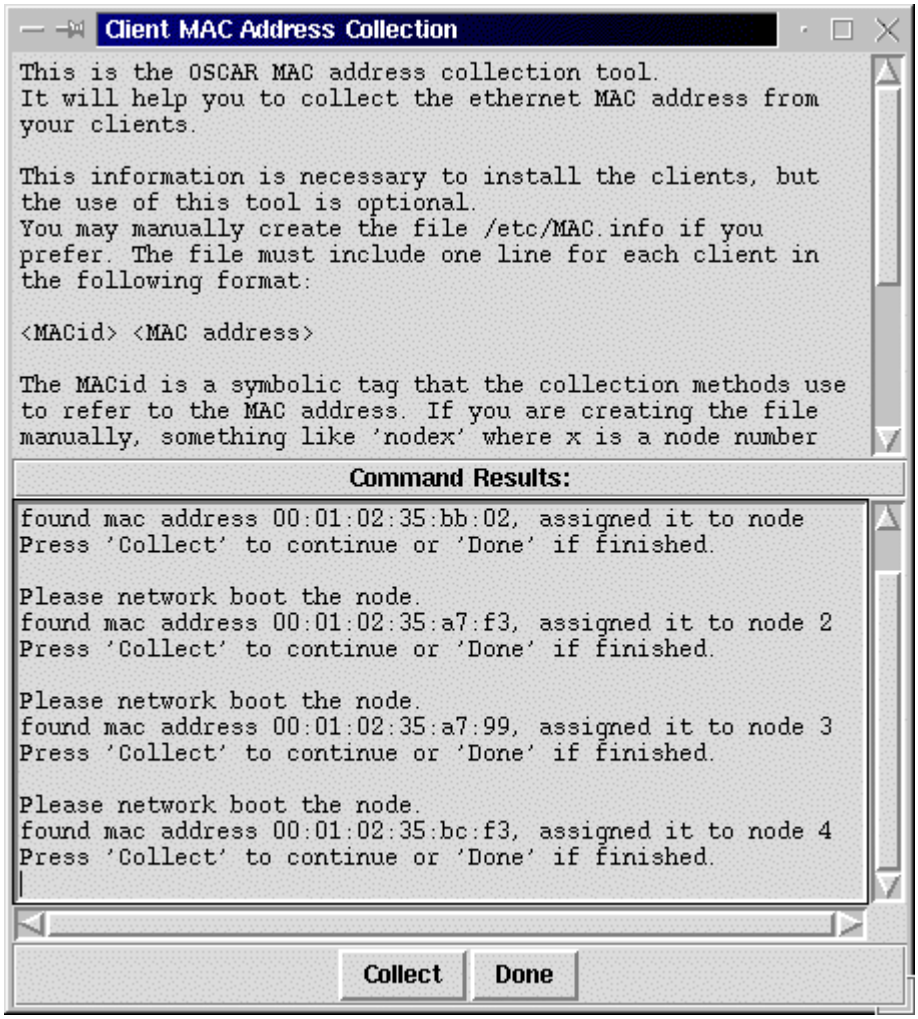

Figure 3: Collect Client MAC Addresses

3. **Edit */etc/hosts* to add your client information.** You will now need to edit the */etc/hosts* file using your favorite editor. For each client, you will need to assign a private IP address similar to that used for the internal server IP. Add a line of the form

   IP-address hostname alias-list

   for each of your clients. Be sure to write down the IP address of the first client you enter, which will be used later when defining the client machines. A sample */etc/hosts* for the cluster being defined in the sample figures would look like the following, where the entries for 'localhost' and the server 'envy' would have already been in the file before it was edited:

   ```
   127.0.0.1       localhost.localdomain   localhost
   10.0.0.50       envy.oscar.cluster      envy
   10.0.0.1        node1.oscar.cluster     node1
   10.0.0.2        node2.oscar.cluster     node2
   10.0.0.3        node3.oscar.cluster     node3
   10.0.0.4        node4.oscar.cluster     node4
   ```

   Once you have entered the information for all of your clients, proceed with the next step.

4. **Define your client machines.** Press the Step 3 button of the wizard entitled **Define the Client Machines**. In the dialog box that is displayed, enter the appropriate information.

   a) In the **Starting IP Address** field, enter the first client IP entered in */etc/hosts* in the previous step.
   b) For **Netmask**, enter the cluster netmask used when defining the server.
   c) If you decided to edit the */etc/MAC.info* file yourself, enter the node id of the first client in the **Starting MAC ID** field. Otherwise, you may leave this field blank.
   d) Enter the number of client machines to create in its field. The number of clients should be equal to the number of entries you added in */etc/hosts*.
   e) For **Default Route** & **Default Gateway**, enter the internal server IP address if you plan to use your server to route messages from your client nodes to the external network. Although OSCAR does not configure the server for routing of client messages, by entering the server IP now you can save yourself from having to manually update the clients later. If you have no plans to route internal messages to the external network, you may leave both fields blank.
   f) In the **PBS String** field, you may enter an arbitrary string of words to describe your cluster nodes. The words in this string can be used within PBS to make scheduling decisions. When all of your client machines will be homogeneous, you can safely leave this field blank. However, if you plan to do multiple cluster installations using the same server, you should enter a string to distinguish these client nodes from those you may define and install at a later time. A suggested string in this case is the client group name specified in the next field.
   g) In the **Machine Groups** field, enter a name for the group of clients to be created (e.g., myclients). Remember to write down your client group name, as it will be used later.
   h) Finally, choose whether or not the clients should use the default server setting for automatically rebooting after installation. If they should, select '*default*' from the **Client auto-reboot** list. If not, then select the desired action by choosing '*true*' or '*false*.'

When finished entering information, press the **`Apply`** button. A sample dialog with input and output is shown in Figure 4. After viewing the output, which may be delayed by several minutes, press the **`Close`** button and continue with the next step.

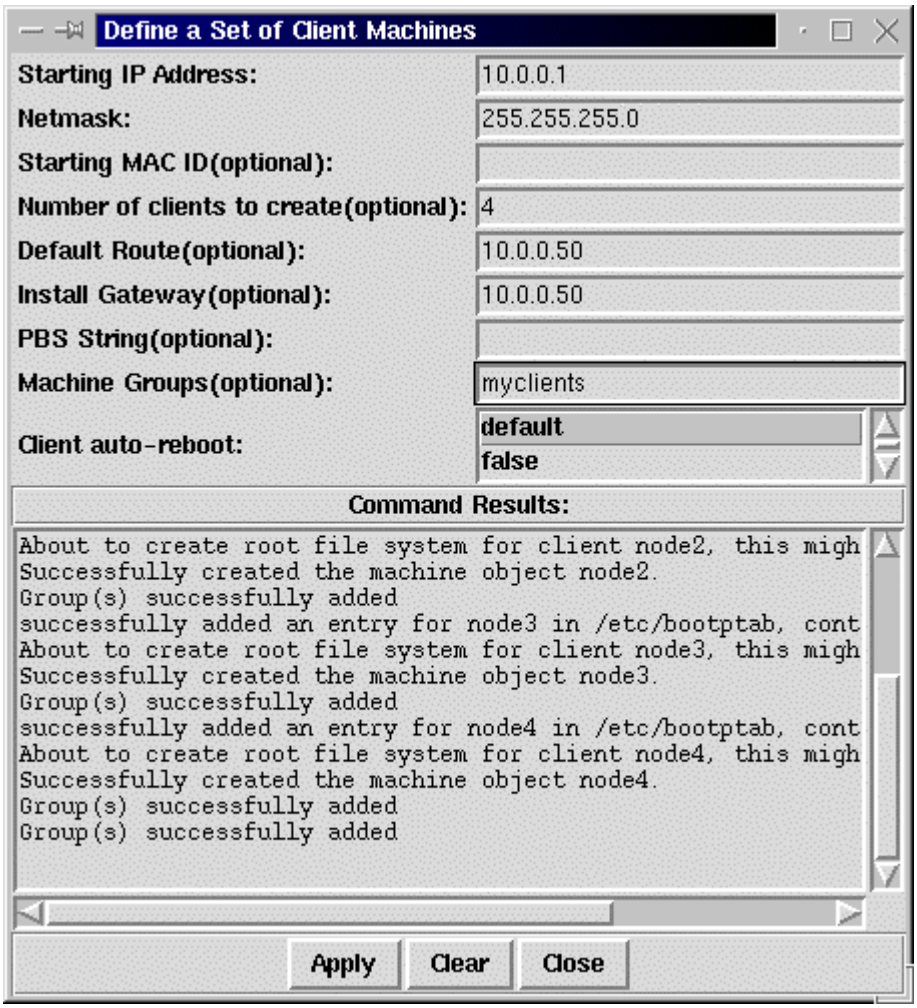

Figure 4: Define the Clients

5. **Define your resources.** Press the Step 4 button of the wizard entitled **`Define the Resources`**. The 'Define a Resource' dialog will be displayed, which is used to define all the resources to be allocated to the clients. When defining your resources, it is best to assign them to a resource group as you go, otherwise you will need to know the names of all your resources when allocating them to the client group defined in the previous step. The 'Define a Resource' dialog contains a **`Resource Groups`** entry field in which you can enter the name of the resource group (e.g., myresources) to add resources to. For each of the resources defined below, be sure to enter your resource group name in its field. Remember to write down your resource group name, as it will be used later.

   a) **Define a disk table.** Create a disk table that specifies how your client hard drives will be partitioned. Refer to the *`/usr/local/lui/README.disk`* file for instructions on creating a disk table. Additionally, OSCAR provides sample IDE and SCSI disk tables in

the *oscarResources* subdirectory. Be sure to include */home* in your disk table as an NFS partition. A sample disk table for an IDE drive with a 24MB boot partition, a 128MB swap partition, and the rest of the disk allocated to the root partition is shown below:

```
/dev/hda1    ext2        24     m     y     /boot
/dev/hda2    extended    *      m     n
/dev/hda5    swap        128    m
/dev/hda6    ext2        *      m     n     /
nfs          nfs         /home        rw    10.0.0.50
```

Once you have created your disk table, enter a name to refer to the disk resource by in the **Name** field, select '*disk*' in the resource type list, and enter the location of your disk table in the **Full Path Name** field. When finished, press the **Apply** button. A sample dialog showing a disk table being defined and the associated output is shown in Figure 5.

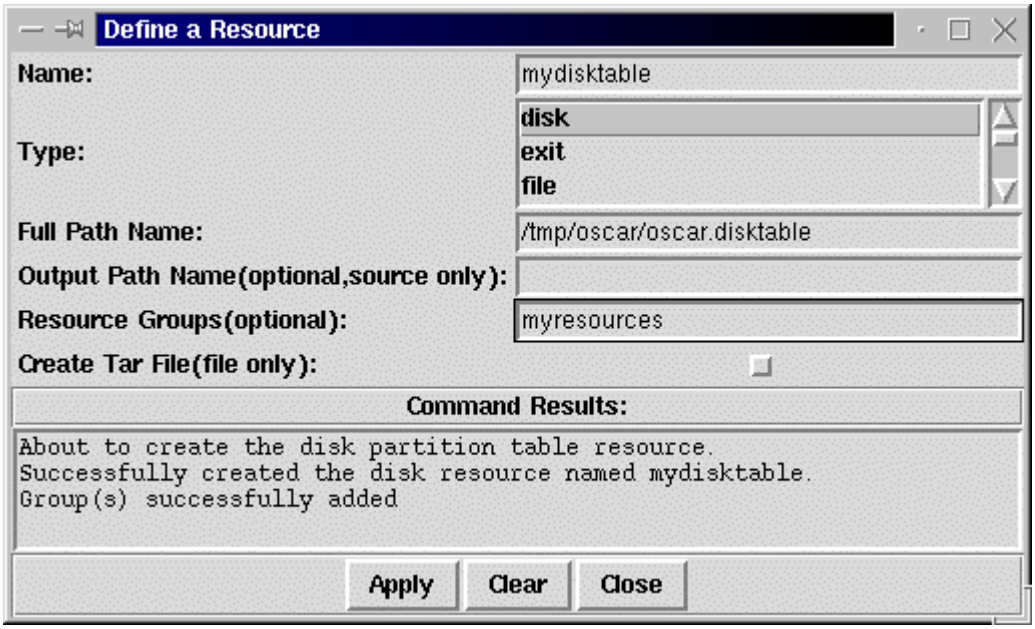

Figure 5: Define the Disk Resource

b) **Define the file systems.** For each of the partitions with associated file systems specified in your disk table, including */home*, create a file resource by entering the resource name, selecting the resource type '*file*', and specifying the file system path. Since OSCAR uses the RPM method of installation under LUI, you should not select the **Create Tar File** option. When finished, press the **Apply** button. A sample dialog showing the */home* file resource being defined and the associated output is shown in Figure 6.

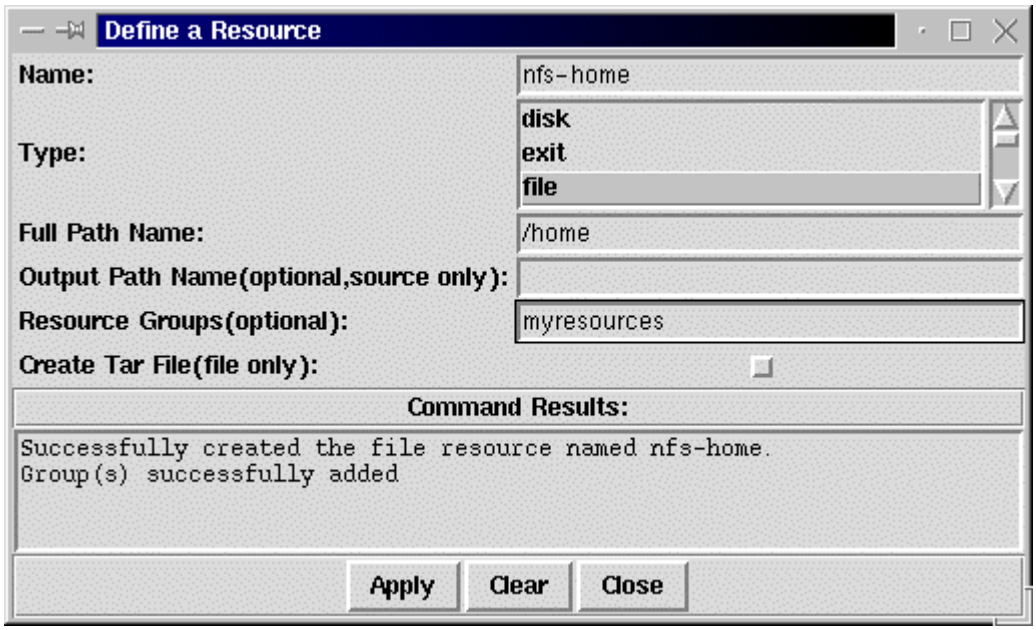

Figure 6: Define the /home File Resource

c) Define a custom kernel (Optional). If you would like to use the default kernel from your Linux distribution, do not define a kernel resource, as it will already be installed during the installation of RPMs. If you have a compiled a custom kernel you would like to use on your client machines instead, you need to define a custom kernel resource for it by specifying a resource name, selecting '*kernel*' from the resource types, and entering the path to the custom kernel. When finished, press the **`Apply`** button.

d) Define a system map (Optional). If you did not define a custom kernel resource, you do not need to define a system map resource. If you did define a custom kernel resource, you will need to have a system map file resource that is compatible with that kernel. Define the custom system map by entering a resource name, selecting '*map*' from the resource types, and entering the path to the map file. When finished, press the **`Apply`** button.

e) Define an initial ramdisk. The initial ramdisk resource is one of the most important for a modular kernel, as it is crucial in making sure that clients boot after installation. You will need to create an initial ramdisk to support the hardware of your client, including any SCSI devices and network cards. See Appendix B for instructions on creating an initial ramdisk. After creating the ramdisk, create a resource for it by entering a resource name, selecting '*ramdisk*' from the resource types, and specifying the path to the ramdisk. When finished, press the **`Apply`** button. A sample dialog showing a ramdisk being defined and the associated output is shown in Figure 7.

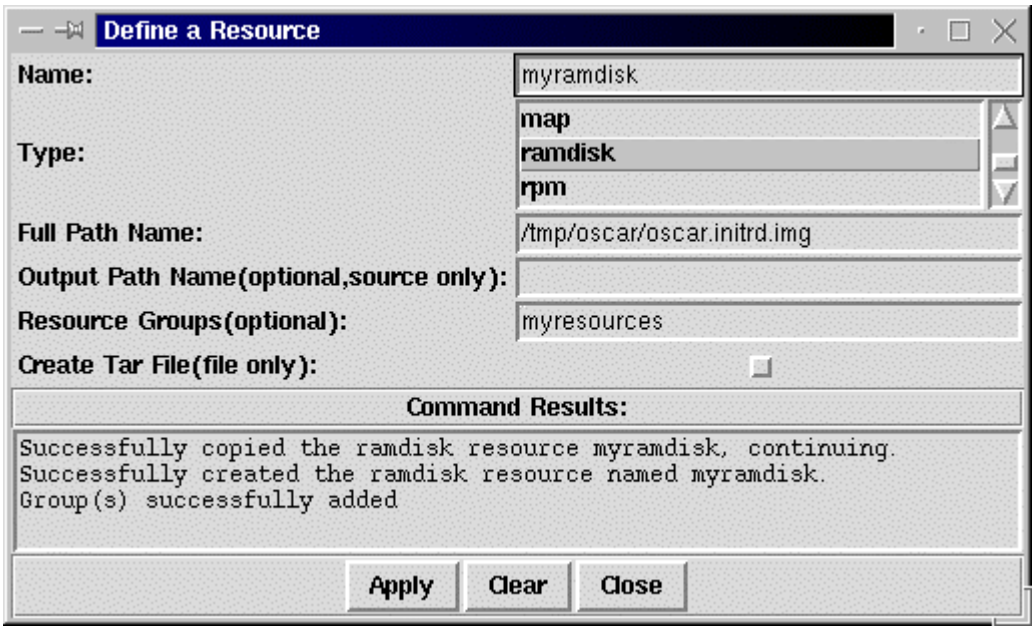

Figure 7: Define the Initial Ramdisk Resource

f) Define the RPM list. OSCAR provides a sample RPM list that contains a minimal set of RPMs needed to install RedHat 6.2 and the cluster software on a client machine. The list is located in the *oscarResources* subdirectory. If you are using another Linux distribution, you will need to create an RPM list that will work for that distribution. If you have additional RPMs for software that you would like installed on the clients, add them to the supplied list and copy the RPMs into */tftpboot/rpm*. Create the RPM list resource by specifying a resource name, selecting '*rpm*' from the resource types, and entering the location of the RPM list. When finished, press the **Apply** button. A sample dialog showing an RPM list being defined and the associated output is shown in Figure 8.

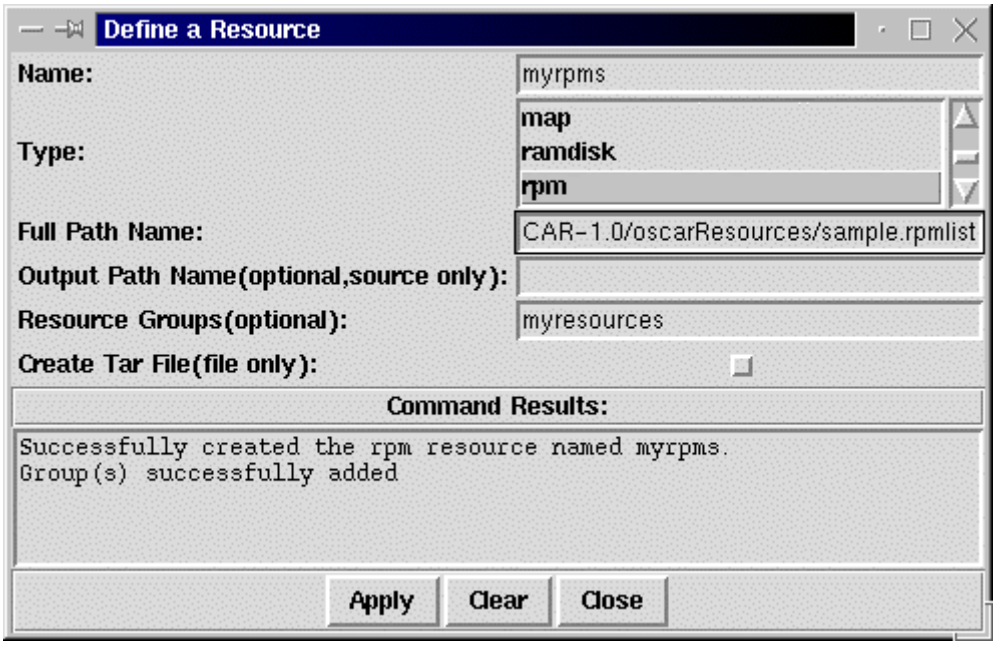

Figure 8: Define the RPM List Resource

g) **Define root .rhosts source file.** Create a root *.rhosts* source file resource by entering a resource name, selecting '*source*' from the resource types, and entering the source and destination paths given below. Press the **Apply** button to create the resource and **Close** when finished.

source: **/usr/local/oscar/lui_sources/root_rhosts**
destination: **/root/.rhosts**

A sample dialog showing the definition of the root *.rhosts* file is shown in Figure 9. Note that the file is created only on the clients and is used only for installation purposes. As soon as ssh is installed and functioning properly, the file is removed from each of the clients.

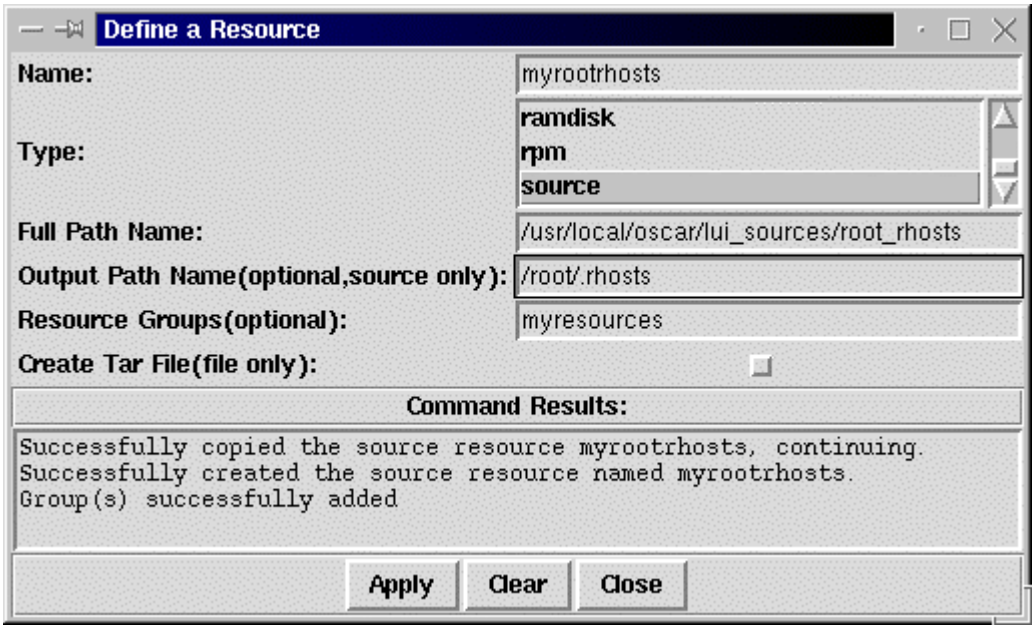

Figure 9: Define the /root/.rhosts Source Resource

6. **Allocate the resources to clients.** Press the Step 5 button of the wizard entitled **Allocate the Resources**. At this point you will need to know your client and resource group names. If you did not specify a resource group when defining your resources, see the paragraph below. Allocate the resources to the clients by entering the resource group name in the **Resource Groups** field and the client group name in the **Machine Groups** field. When finished, press the **Apply** button. A sample dialog with input and output is shown in Figure 10. After allocating the resources, continue on to the next step.

   If you did not use a resource group, you will need to know the names of all the resources defined in the previous step. If you forgot to write them down, you should be able to figure the names out from a listing of the */tftpboot/lim* directory. Each resource defined is represented by a file in the directory, with the file's name being of the form *resource_name.resource_type*. To allocate the resources to your clients, enter the resource names separated by commas in the **Resource Names** field and the client group name in the **Machine Groups** field. Be careful not to use spaces in the resource names field, as they are not allowed.

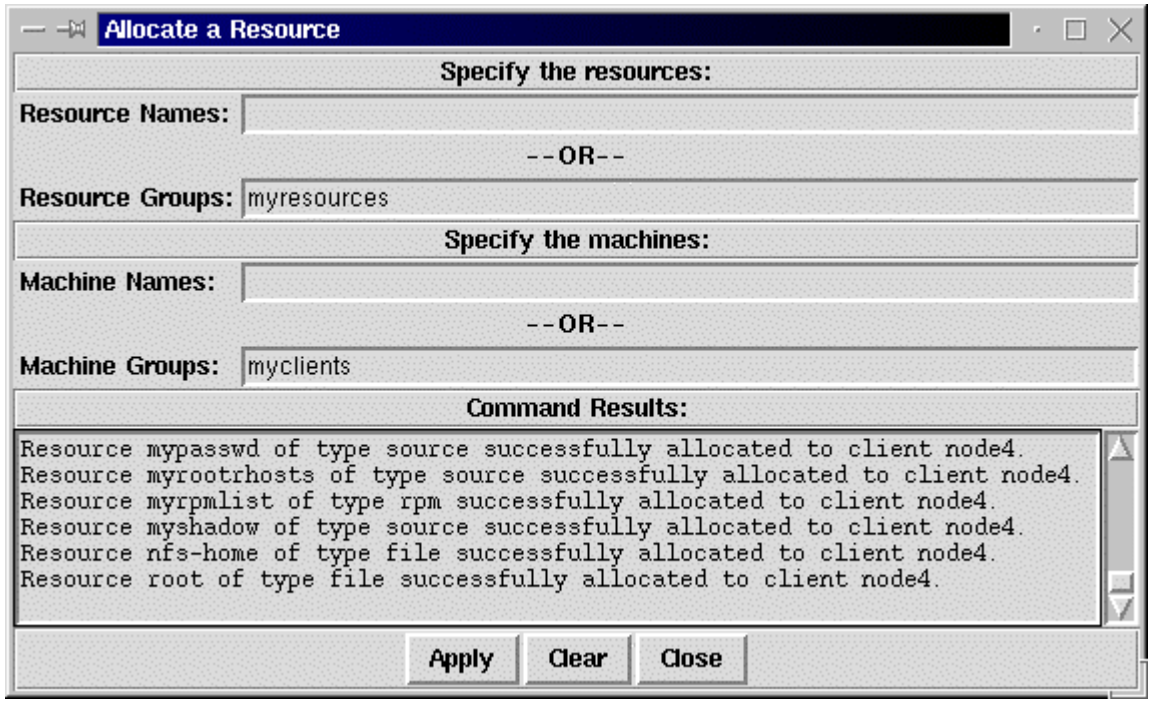

Figure 10: Allocate the Resource Group to the Client Group

D. Secondary OSCAR Server Configuration

During this phase, the server will be prepared for the client installs based upon the information entered in the 'Cluster Definition' phase.

1. **Run secondary server configuration.** Press the Step 6 button of the wizard entitled **`Prepare for Client Installs`**. This will run the pre client installation server configuration script, which does the following:
   a) uses cluster information database to update */etc/dhcpd.conf*
   b) installs C3 tools
      - puts tools in */usr/bin*
      - installs man pages
   c) installs SystemImager
   d) makes sure dhcpd server is running

2. **Check for successful completion.** In the output window for the above step, check for a message stating "**`Begin booting client nodes`**" before continuing. A sample successful output window is shown in Figure 11. At this point, you may continue with the client installations phase.

   If instead of the above message, you get a message stating that the "**`DHCP server is not running`**", see *Appendix D, Part A, Number 2* for help. Once you have the DHCP server running, you may continue with the client installations phase.

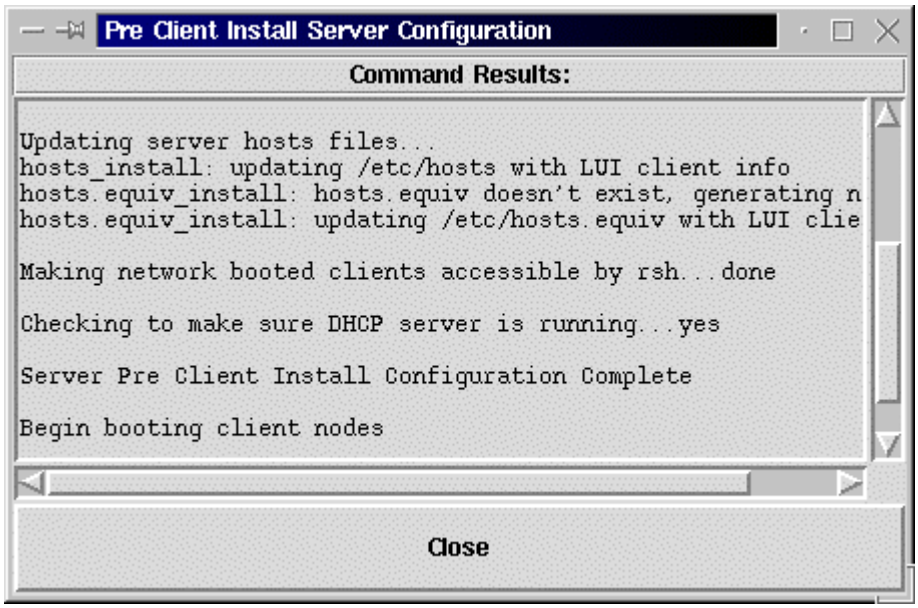

Figure 11: Successful Output Window for the 'Prepare for Client Installs' Step

E. Client Installations

During this phase, you will network boot your client nodes and they will automatically be installed and configured as specified in Section IV.C above. For a detailed explanation of what happens during client installation, see Appendix C.

1. **Network boot the client nodes.** See *Appendix A* for instructions on network booting clients.

2. **Check completion status of nodes.** For each client, a log is kept detailing the progress of its installation. The log files for all clients are kept on the server in `/tftpboot/lim/log`. When a client installation completes, the last line in the log for that client will read "**`installation is now complete, time to reboot!`**" Depending on the capabilities of your server and the number of simultaneous client installations, a client could take anywhere from five minutes to over an hour to complete its installation.

3. **Reboot the client nodes.** After confirming that a client has completed its installation, you should reboot the node from its hard drive. If you chose to have your clients auto-reboot after installation, they will do this on their own. If the clients are not set to auto-reboot, you must log in to the node and reboot it. After logging in, issue the command '**`reboot -f`**', which issues a reboot with the force option. The force option, which reboots the machine without shutting down any services, is needed so that the node does not hang on shutdown of its network. Since its file systems are network mounted during installation, the attempt to shutdown the network will hang the machine if the force option is not used. Note: If you had to change the BIOS boot order on the client to do a network boot before booting from the local disk, you will need to reset the order to prevent the node from trying to do another network install.

4. **Check network connectivity to client nodes.** In order to perform the final cluster configuration, the server must be able to communicate with the client nodes over the network. If a client's ethernet adapter is not properly configured upon boot, however, the server will

not be able to communicate with the client. A quick and easy way to confirm network connectivity is to do the following (assuming OSCAR installed in */tmp*):

```
cd /tmp/OSCAR-1.0/scripts
./ping_clients
```

The above commands will run the *ping_clients* script, which will attempt to ping each defined client and will print a message stating success or failure. If a client cannot be pinged, the initial ramdisk provided probably did not have built in module support for its ethernet adapter, and you will have to log in to the machine and manually configure the adapter. Once all the clients have been installed, rebooted, and their network connections have been confirmed, you may proceed with the next step.

F. Cluster Configuration

During this phase, the server and clients will be configured to work together as a cluster.

1. **Complete the cluster configuration.** Press the Step 7 button of the wizard entitled **`Complete Cluster Setup`**. This will run the *post_install* script, which does the following:
   a) updates hosts files on server & clients
   b) configures user rsh capabilities
   c) configures ssh on server & clients
      - *root* authentication
      - user authentication
   d) installs & configures MPI-CH in */usr/local/mpi*
   e) installs & configures PVM in */usr/share/pvm3*
   f) updates clients C3 configuration
   g) installs Veridian's Portable Batch System (PBS)
      - the PBS server & default scheduler are installed, but not the execution mom, since the server machine is not used for computation
   h) updates clients' PBS configuration
      - creates PBS execution mom startup script and adds to system startup scripts
   i) synchronizes the clients' time & date with the server

   Note: there will be a several minute delay before the output for this step will appear

2. **Check for successful completion.** In the output window for the above step, search for a message stating "**`Congratulations, your cluster is now ready for use`**." A sample successful output window is shown in Figure 12. If you do not find this message in the output, try to find out what may have failed by searching through the output for error messages.

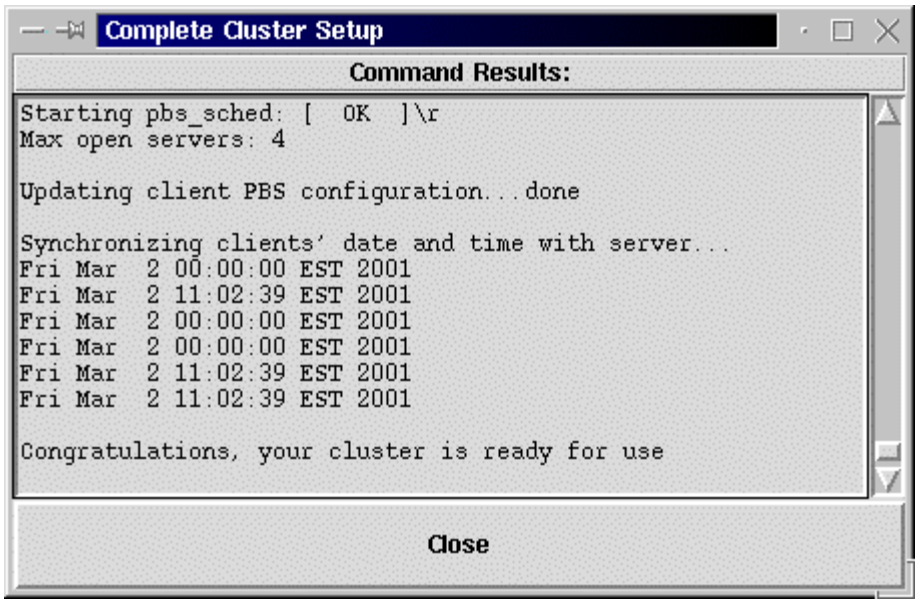

Figure 12: Successful Output Window for the 'Complete Cluster Setup' Step

3. **Test your cluster using the OSCAR Cluster Test software.** Provided along with OSCAR is a simple test to make sure the key cluster components (PBS, MPI, & PVM) are functioning properly. For information on installing and running the software, see the *oscar_testing* document in the *docs* subdirectory.

Appendices

Appendix A: Network Booting Client Nodes

There are two methods available for network booting your client nodes. The first is to use the Preboot eXecution Environment (PXE) network boot option in the client's BIOS, if available. If the option is not available, you will need to create a network boot floppy disk using the Etherboot package. Each method is described below.

1. **Network booting using PXE**. To use this method, your client machines' BIOS and network adapter will need to support PXE version 2.0 or later. The PXE specification is available at http://developer.intel.com/ial/wfm/tools/pxepdk20/index.htm. Earlier versions may work, but experience has shown that versions earlier than 2.0 are unreliable. As BIOS designs vary, there is not a standard procedure for network booting client nodes using PXE. More often than not, the option is presented in one of two ways.
   a) The first is that the option can be specified in the BIOS boot order list. If presented in the boot order list, you will need to set the client to have network boot as the first boot device. In addition, when you have completed the client installation, remember to reset the BIOS and remove network boot from the boot list so that the client will not attempt to do the installation again.
   b) The second is that the user must watch the output of the client node while booting and press a specified key such as 'N' at the appropriate time. In this case, you will need to do so for each client as it boots.

2. **Network booting using an Etherboot floppy.** The Etherboot package is provided with OSCAR just in case your machines do not have a BIOS network boot option. The directory for Etherboot is located in the `/usr/local/oscar/lui` directory. For instructions on creating a network boot floppy, read the `doc/txt/README.txt` in the Etherboot directory. The commands you will run will probably look like the following:

   (from within the Etherboot directory and with a floppy inserted)

   ```
   cd src
   make
   cat floppyload.pre.bin bin32/ethernet-card.lzrom > /dev/fd0
   ```

   After creating the floppy, you will have to make one more change in order to make OSCAR function correctly with booting from a floppy. By default, OSCAR assumes you will be using PXE to do the network boot, and sets up the DHCP configuration file accordingly. To correct this, you need to edit the configuration file, `/etc/dhcpd.conf`, and replace all the references to '`pxelinux.bin`' with '`vmlinuz`' in the `filename` entry for each client. After editing the file, you will need to restart DHCP by issuing the following command in a terminal:

   ```
   /etc/rc.d/init.d/dhcpd restart
   ```

   Once you have created the network boot floppy and updated the DHCP configuration, set your client's BIOS to boot from the floppy drive. Insert the floppy and boot the machine to start the network boot. Check the output for errors to make sure your network boot floppy is working properly. Remember to remove the floppy when you reboot the clients after installation.

Appendix B: Generating Initial Ramdisks

To create an initial ramdisk for your client machines, you will use the **mkinitrd** command. Using the command, you can create a ramdisk to support any special hardware your clients may contain. If the client machines contain SCSI disks, you will need to build support for the SCSI adapter into your ramdisk. In addition, you should build in support for your client's ethernet adapter. If your client machines contain the same hardware as your server, you can create a source resource for the */etc/conf.modules* file instead of building support for the ethernet adapter into the ramdisk.

Before creating the initial ramdisk, you should be aware of some important caveats. The first caveat is that the ramdisk you create must match the kernel to be installed on the clients. As LUI automatically installs the appropriate kernel (UP/SMP) based upon the number of processors in the client machine, you should create a ramdisk that matches this kernel. The second caveat is that in order to build a ramdisk for a particular kernel version, the kernel's associated modules must be located on the server in */lib/modules/kernel-version*.

Now that you are aware of the caveats described above, you are ready to build an initial ramdisk for your clients. A typical command using the kernel currently running on your server is as follows:

/sbin/mkinitrd –v –-with=*eth-module* client-initrd.img `uname –r`

In the above command, "eth-module" is the name of the module for the client's ethernet adapter, e.g., eepro100, "client-initrd.img" is the name of the ramdisk to create, and "uname –r" returns the version of the currently running kernel. The above command also assumes the client nodes use the same disk adapter(s) as the server system. If alternate adapters are used, specify them before the ethernet adapter with additional '**--with**' arguments.

If you defined custom kernel and system map resources, then be sure to specify the appropriate kernel version as the last argument. For example, the command

/sbin/mkinitrd –v --with=aic7xxx --with=eepro100 client-initrd.img 2.2.17

will create an initial ramdisk with support for the AIC7xxx series of Adaptec SCSI adapters and the Intel EtherExpress Pro 100 ethernet adapter using the modules located in */lib/modules/2.2.17*.

For additional information on how to use **mkinitrd**, see its man page, i.e., '**man mkinitrd**'.

Appendix C: What Happens During Client Installation

Once the client is network booted, it broadcasts a BOOTP/DHCP request to obtain the IP address associated with its MAC address. The DHCP server provides the IP, along with the name of a network boot file. The client downloads and processes the boot file, from which it obtains the name of the kernel to boot. The kernel is then downloaded and booted. During boot, the kernel mounts its file systems from the server. The file systems for each client are created by LUI and are stored in */tftpboot* in a directory whose name is the client's IP address. The client also mounts */usr* from the server, providing access to the routines of LUI. The last item started when the client is processing its system startup scripts is the LUI clone script. The clone script is the installation workhorse, and does the following:

1. partitions the disk as specified in the disk resource

2. mounts the newly created partitions on */mnt*
3. chroots to */mnt* and installs the RPMs specified in the rpm resource
4. copies any source resource files to the client's local disk
5. unmounts */mnt*

Once clone completes, the client will show its login prompt, at which time you should check the node log file for successful completion status as described in the 'Client Installations' phase.

Appendix D: Troubleshooting

A. Known Problems & Solutions

1. **Client nodes fail to network boot.** There are two causes to this problem. The first is that the DHCP server is not running on the server machine, which probably means the */etc/dhcpd.conf* file format is invalid. Check to see if it is running by running the command **'ps -A | grep dhcpd'** in the terminal. If no output is returned, the DHCP server is not running. See the problem solution for '**DHCP server not running**' below. If the DHCP server is running, the client probably timed out when trying to download its configuration file. This may happen when a client is requesting files from the server while multiple installs are taking place on other clients. If this is the case, just try the network boot again when the server is less busy. Occasionally, restarting the inet daemon also helps with this problem as it forces tftp to restart as well. To restart the daemon, issue the following command:

   ```
   /etc/rc.d/init.d/inet restart
   ```

2. **DHCP server not running.** Run the command '**/etc/rc.d/init.d/dhcpd start**' from the terminal and observe the output. If there are error messages, the DHCP configuration is probably invalid. A few common errors are documented below. For other error messages, see the **dhcpd.conf** man page.

   a) If the error message produced reads something like "**Can't open lease database**", you need to manually create the DHCP leases database, */var/state/dhcp/dhcpd.leases*, by issuing the following command in a terminal:

      ```
      touch /var/state/dhcp/dhcpd.leases
      ```

   b) If the error message produced reads something like "**Please write a subnet declaration for the network segment to which interface eth*x* is attached**", you need to manually edit the DHCP configuration file, */etc/dhcpd.conf*, in order to try to get it valid. A valid configuration file will have at least one subnet stanza for each of your network adapters. To fix this, enter an empty stanza for the interface mentioned in the error message, which should look like the following:

      ```
      subnet subnet-number netmask subnet-mask { }
      ```

      The subnet number and netmask you should use in the above command are the one's associated with the network interface mentioned in the error message.

3. **PBS is not working.** The PBS configuration done by OSCAR did not complete successfully and requires some manual tweaking. Issue the following commands to configure the server and scheduler:

```
/etc/rc.d/init.d/pbs_server start
/etc/rc.d/init.d/pbs_sched start
cd /mnt/OSCAR_1.0/pbs/config
/usr/local/bin/qmgr < pbs_server.conf
```

   Replace *'/mnt'* with the directory into which you unpacked OSCAR in the change directory command above.

B. What to do about unknown problems?

For help in solving problems not covered by this HowTo, send a detailed message describing the problem to the OSCAR users mailing list at oscar@openclustergroup.org. You may also wish to visit the OSCAR web site, http://www.csm.ornl.gov/oscar, for updates on newly found and resolved problems.

C. Starting Over or Uninstalling OSCAR

If you feel that you want to start the cluster installation process over from scratch in order to recover from irresolvable errors, you can do so with the *start_over* script located in the *scripts* subdirectory. This script is interactive, and will prompt you when removing components installed by OSCAR that you may not want to remove.

If you would like to remove all traces of OSCAR from your server, you may do so by running the *uninstall* script located in the *scripts* subdirectory. This will run the *start_over* script and then remove the OSCAR directory created by unpacking the tarball.