OSCAR: Open Source Cluster Application Resources

Michael J. Brim (brimm@ornl.gov)
Timothy G. Mattson (timothy.g.mattson@intel.com)
Stephen L. Scott (scottsl@ornl.gov)

Abstract

Open Source Cluster Application Resources (OSCAR) is the packaging of current "best known practices" for Beowulf cluster computing into a fully integrated, easy to install software suite. Everything needed to install, build, maintain, and use a modest sized Linux cluster is included in the package. Thus, it is unnecessary to search for, download, or even install individual cluster components. OSCAR consists of RPM's, Perl scripts, libraries, and various cluster tools. OSCAR, and every component included in OSCAR, is available under one of the well known open source licenses or is freely redistributable. OSCAR may be found at `http://www.csm.ornl.gov/oscar`.

1 Introduction

In recent years, cluster computing has emerged as a highly available means for obtaining additional computational power in the scientific, commercial, and educational communities. As a result, much software has been developed or modified to utilize clusters, further promoting their attractiveness. Until lately, however, the total cost of ownership for clusters, although lower than traditional high performance computers, has been held relatively high when considering the time and expertise needed for their installation and maintenance. Previously, people were forced to do extensive research just to find out what components were needed to build a cluster. After obtaining this knowledge, they had to obtain each component, and then manually install and configure the software. Even after successfully building a cluster, they were still faced with the daunting task of ongoing maintenance and administration. The goal in providing OSCAR is to eliminate the need for such hardships by providing a complete clustering solution in a single package that greatly reduces the learning curve in building, using, and maintaining a cluster.

The rest of the paper is organized as follows. In the next section, the components of OSCAR are discussed. Section III provides a detailed overview of the cluster installation process using OSCAR. Finally, conclusions and future work are discussed in Section IV.

2 OSCAR Composition

As OSCAR is being developed as a complete clustering solution, there are many functional areas that must be covered by the included software components. These areas include installation, parallel processing environment, workload management, security, and general administration and maintenance. To satisfy the requirements in each area, the software components included with OSCAR were selected by investigating the practices of several independent cluster-computing sites. The result was a collection of software that is representative of the "best known practices" for creating a successful cluster environment. The software included for each area is discussed in the following paragraphs.

2.1 Installation

Probably the most difficult part of creating a successful cluster environment is the initial installation of the software responsible for making independent machines work together as a single computing resource. The software included with OSCAR to simplify the process is the 'Linux Utility for cluster Install' (LUI) [1], which is one of the many open source projects currently being developed at the IBM Linux Technology Center. 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.

2.2 Parallel Processing Environment

For parallel processing, OSCAR supports the message-passing paradigm and provides the two most common implementations, the 'Message Passing Interface' (MPI) [2] and the 'Parallel Virtual Machine' (PVM) [3]. Both MPI and PVM have substantial user bases throughout the world, which is why both are provided in OSCAR. As there are many versions of MPI available, the developer's decided to use MPICH from Argonne National Laboratory since it seems to be the most abundantly used.

2.3 Workload Management

Typically, clusters are used as a computing resource by multiple users, providing the necessity of workload management software. Such software is responsible for maximizing the utilization of the cluster resources based upon the application requirements specified by users. The software provided with OSCAR to manage workloads is the open source version of the 'Portable Batch System' (PBS) [4] from Veridian. PBS is responsible for accepting job requests from users, monitoring the state of the cluster and its resources, and running jobs when the resources required become available. Currently, OSCAR uses the default FIFO scheduler included with PBS, but work has already begun to substitute that scheduler with the Maui scheduler [5], which is known to outperform the default scheduler.

2.4 Security

Due to the wide variety of environments in which clusters are used, the developers of OSCAR felt it necessary to include secure methods for communicating to and inside of a cluster. As most clusters are attached to an external network through a gateway node, it is necessary to provide a secure login mechanism to prevent other users on the network from being able to access the cluster. In addition, user applications and data need to be kept private within the cluster. The software included in OSCAR to provide this security is OpenSSL [6] and OpenSSH [7]. OpenSSL is an open source implementation of the Secure Sockets Layer protocol, which provides secure communications over the Internet. OpenSSH is an open source implementation of the Secure Shell protocol, which provides secure login, file transfer, and connections forwarding. OpenSSH requires the use of external libraries provided by OpenSSL.

2.5 Administration and Maintenance

The software provided for cluster administration and maintenance is the 'Cluster Command & Control' (C3) [8] tool suite from Oak Ridge National Laboratory. The C3 tools are command-line scripts which provide the functionality necessary to efficiently manage clusters where each node contains its own copy of the operating system and software. The functionalities provided include cluster-wide command execution, file distribution and gathering, remote shutdown and restart, process status and termination, and system image update. The tools support operations on the entire cluster and subsets of cluster nodes.

3 Cluster Installation

Before diving into the cluster installation procedure, it is beneficial to give a description of the general layout for OSCAR clusters. 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 clients contain homogeneous hardware. The server node, along with serving as the gateway to the external network, contains the home directories of all users and runs the PBS server and scheduler. The clients each have a local copy of the operating system and other software, with the exception of NFS mounting the users' home directories from the server. All the nodes are connected by an internal Ethernet network, preferably through a dedicated switch.

3.1 LUI Concepts

In order to understand some of the steps in the installation procedure, it is essential to have a 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 three pieces of information that are kept for the server machine: name, IP address, and corresponding subnet mask. 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. 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, 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

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

postinstall - a user script that is run after client installation but before the machine is rebooted

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, users 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.

3.2 Installation Procedure

The rest of this section will give a detailed overview of the cluster installation procedure using OSCAR. For an even more in depth installation tutorial, refer to [9]. The following steps assume you have already physically put the cluster together, including the networking of nodes together.

The first step in creating a cluster using OSCAR is to install Linux on the machine to be used as the server. If you already have Linux installed on a machine that you want to be the server, you may use it as long as the version of Linux installed supports the RPM standard.

Once Linux is installed on the server, you will need to get the OSCAR distribution, which is currently available as a zipped tarball and can be obtained from the OSCAR site [10]. Copy the OSCAR tarball to a directory such as `/mnt` or `/tmp` on your server and unpack it. There is no required installation directory, except that you may not use

`/usr/local/oscar`, which is reserved for special use.

The next step is 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 600MB for the RPMs from your Linux distribution plus 5MB for each client. After creating the directories, copy the RPMs from your Linux distribution to the `/tftpboot/rpm` directory.

At this point, the first phase of server configuration is run. The first configuration is responsible for installing and configuring all of the software needed to perform the rest of the OSCAR cluster installation. The software installed includes LUI, DHCP, NFS, TFTP, and Syslinux. If the first phase configuration is successful, the OSCAR install wizard is started. The wizard is provided to guide users 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.

The first wizard step is definition of your server machine to LUI. In this step, you will fill in the name for your server, along with its internal IP address and cluster subnet mask. After the information is entered, LUI will create the corresponding server machine object.

The second step of the wizard is to collect the MAC addresses of the client nodes. There are two options for doing so. If you already know the addresses, you will enter the information in the standard MAC configuration file `/etc/MAC.info`. If you do not know the addresses, you will use the collection tool supplied with OSCAR. The collection tool requires that you network boot each client machine in sequence. The tool collects each address from the network boot request sent to the server, and makes the appropriate entry in the MAC configuration file.

Before starting the third step of the wizard, you will need to define your clients in the `/etc/hosts` file by entering their IP address, long hostname, and any aliases you wish to use. After editing the hosts file, you can begin the third wizard step, which defines your client machines to LUI. In this step, you will define a client group by entering the IP address of the first client as found in the hosts file, the cluster subnet mask, the number of clients to create, and a name for your client group. You may optionally specify the default route and gateway for your clients, the number of processors per each client, and a string used by PBS to distiguish among nodes when making scheduling decisions. After entering the information, LUI will create a client machine object for each client by using the information from the hosts and MAC configuration files in addition to the information supplied to the wizard.

In the fourth step of the wizard, you will define the resources used to install the client machines. There are a few resources that are required, and others that are optional. The required resources include a disk table specifying how to partition the clients' hard drives, a file resource for each of the partitions associated with a filesystem in the disk table, and a list of RPMs to install on the clients. Since these resources are required and generating the appropriate files may not be intuitive, OSCAR provides samples that users can modify or use directly. The optional resources are ones that enable users to create clusters that meet individual needs. For example, users can create a custom kernel and associated system map and have them as resources to be installed on the clients. Similarly, any files that users may want copied from the server to clients can be defined as source resources. All of the resources defined can be included in a resource group by specifying the name of the group when defining each resource.

The fifth wizard step is to allocate the resources defined to your client machines. If you used a group when defining resources, this is accomplished by simply naming the client machine group and the resource group. Otherwise, you will need to allocate each resource to the client group.

The sixth step of the wizard is responsible for performing the second phase of the automated server configuration. This phase installs the OpenPBS and C3 software, as well as configures the DHCP server in preparation for the client machine installations.

After completing the second server configuration phase, it is time for the client installations. During this phase, you will network boot your client nodes and they will automatically be installed by LUI. The actual steps taken in the client installation are as follows. Once each 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, which is `/tftpboot/pxelinux.bin`. The client downloads and processes the boot file, from which it obtains the name of the kernel to boot, which is `/tftpboot/bzImage`. The kernel is then downloaded and booted. During boot, the kernel mounts its client file systems from the server. The file systems for each client are created by LUI and are stored in `/tftpboot/<client IP>`. The clients also mounts `/usr` from the server, providing access to the routines of LUI. The last item started when the clients are processing their system startup scripts is the LUI clone script. The clone script is the installation workhorse, and is responsible for doing 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 resources from the server to the client

5. unmounts `/mnt`

Once clone completes, a client will show its login prompt, at which time you will reboot the node and let it do a standard boot from the hard drive.

After all the clients have completed their installation and have been rebooted from hard drive, the final step of the wizard should be run. The final step is responsible for completing the installation of software and configuration of the cluster. Software being installed or configured in this step includes MPI, PVM, OpenSSH, rsh, C3, and PBS.

At this point, users should have a fully functioning cluster that is ready to run parallel jobs. To make sure that everything is working as expected, OSCAR provides a method to test the functionality of the cluster once installed. The OSCAR Cluster Test software can be used to make sure that MPI, PVM, and PBS are working correctly.

4 Conclusions and Future Work

The developers of OSCAR have big plans for its future. There are numerous other components we are looking into adding to OSCAR, with the Local Area Machine (LAM) version of MPI [11] and the Maui scheduler as two requested additions under consideration. We are also exploring options to provide an easy to use single point configuration and increased automation. In addition, we are looking to expand the flexibility of OSCAR to make it easier to deviate from our canonical cluster. For example, a user should be able to easily omit installation of packages they don't plan to use.

In addition to the above specific plans, we are going to explore ways to extend OSCAR to a broader range of clusters. For example, it might be nice to use OSCAR for building high availability clusters. Another possibility is to extend OSCAR to handle diskless nodes.

The future goals for the project are quite hefty as well. We would like to see OSCAR become a starting point that companies will use to build supported cluster software stacks. We want academia interested in tools-research to use OSCAR as the core software upon which they build their software and tools. Basically, computer scientists spend way too much time re-inventing the wheel. We hope that OSCAR can help put an end to this cycle - at least in terms of the basic components of an open source cluster.

5 Acknowledgements

OSCAR is the first project by the Open Cluster Group [12], a collaboration between industry and government partners aimed at providing a complete open-source high performance clustering solution.

Members of the group who contributed to the development and testing of OSCAR are as follows (organized by company/institution):

Dell: Yung-Chin Fang, Jenwei Hsieh, Tau Leng

IBM: Michael Chase-Salerno, Richard Ferri

Intel: Timothy Mattson

MSC Software: Joe Griffin, David Lombard

NCSA: Jeremy Enos, Neil Gorsuch, Robert Pennington

ORNL: Michael Brim, Brian Luethke, Stephen Scott

SGI: John Hesterberg

Veridian: Bhroam Mann

PBS includes software developed by NASA Ames Research Center, Lawrence Livermore National Laboratory, and Veridian.

6 References

[1] LUI, `http://oss.software.ibm.com/lui`

[2] MPICH, `http://www-unix.mcs.anl.gov/mpi/mpich`

[3] PVM, `http://www.csm.ornl.gov/pvm`

[4] PBS, `http://www.openpbs.org`

[5] Maui High-Performance Batch Scheduler, `http://mauischeduler.sourceforge.net`

[6] OpenSSL, `http://www.openssl.org`

[7] OpenSSH, `http://www.openssh.com`

[8] C3, `http://www.csm.ornl.gov/torc/C3`

[9] "How to install an OSCAR cluster" by The Open Cluster Group. Available at `http://www.csm.ornl.gov/oscar/papers.html`.

[10] OSCAR, `http://www.csm.ornl.gov/oscar`

[11] LAM-MPI, `http://www.lam-mpi.org`

[12] The Open Cluster Group, `http://www.OpenClusterGroup.org`