Feature

OSCAR and Bioinformatics

With new cluster deployment and management tools, you can set up a 64-node cluster in an hour, then put it to work on your research.

by Bernard Li

The OSCAR (Open Source Cluster Application Resources) Project has been around for about four years. The concept initially was proposed in January 2000, and the first organizational meeting was held in April of the same year. The group acknowledged that cluster assembly is time consuming and repetitive. Thus, the project's goal was to create a toolkit to automate this process. In doing so, the group hoped to broaden the usage of clusters and adapt them for the academic and private sector.

The OSCAR Project is overseen by the advisory group OCG (Open Cluster Group), an informal group with open membership. The OCG strives to make cluster computing more practical for high-performance computing (HPC) research and development. The group, like the OSCAR Project, is directed by representatives from research/academia as well as industry. Key players of the group include Bald Guy Software, BC Genome Sciences Centre, Dell, Indiana University, Intel, Louisiana Tech University, Oak Ridge National Laboratory, Revolution Linux and Sherbrooke University.

OSCAR is one of the OCG working groups. Other projects include HA-OSCAR (high-availability), Thin-OSCAR (diskless) and SSS-OSCAR (Scalable Systems Software). To learn more about OCG and its projects, see the on-line Resources for this article.

The first release of OSCAR was in April 2001, and since then we have released two major versions. Our release cycle usually coincides with the SuperComputing Conference, which is held annually in November. The current version as of writing is 3.0, and we are aiming at releasing 4.0 by SuperComputing04.

The goal of OSCAR is to provide users with the best practices for installing, programming and maintaining HPC clusters. Many open-source components individually work well in an HPC environment but require specific setup routines. OSCAR acts as the glue to integrate all these components together to provide a working toolkit. The project targets mid-size clusters (50+ node clusters). Community feedback suggests this size represents the majority of clusters assembled today.

OSCAR has the following components:

- Administration: System Installation Suite (SIS), Cluster Command and Control (C3) and OPIUM (user management).

- HPC tools: the parallel programming libraries: MPICH, LAM/MPI and PVM; batch systems: OpenPBS/MAUI, Torque and SGE; monitoring tools: Ganglia and Clumon; and other third-party OSCAR packages.

- Core infrastructure/management: OSCAR Database (ODA) and OSCAR Package Downloader (OPD).

OSCAR developers are spread out over various geographical locations and weekly teleconferences are set up to discuss ongoing development issues. The group also holds annual general meetings to brainstorm new
features to be included in future releases. An annual symposium also is held, usually in conjunction with HPCS (International Symposium on High Performance Computing Systems and Applications) where users are encouraged to present papers on their experiences with OSCAR as well as other development work relating to HPC. The 2nd annual OSCAR Symposium commenced in May 2004 in Winnipeg, Canada, and the proceedings are now available.

Introduction to Bioinformatics

Bioinformatics is the marriage between biology and computer science/IT and is a rapidly growing field with high stakes in the HPC world. In simple terms, bioinformatics is concerned with the use of computer algorithms and systems to analyze biological data such as DNA, RNA, protein and regulatory elements.

Biological data are mainly string sequences. The analyses are usually string manipulations making Perl the programming language of choice for most bioinformaticians. Many open-source Perl programmers contribute to the Bioperl effort, which is a deposit of Perl modules specifically for performing bioinformatics analyses. Java is employed for larger projects and often for projects that involve graphical interfaces. Python is gaining a strong foothold in the field as more programmers learn about the ease of use and high readability of this relatively new but powerful programming language.

Linux clusters are very popular within the bioinformatics community, because a lot of the analyses tend to be long-running and repetitive. Linux clusters are ideal for running such embarrassingly parallel jobs that can be executed independently of one another. These are not considered true parallel programs because they do not need parallel programming libraries such as MPI. Bioinformatics routinely performed on clusters encompass running multiple scripts and algorithms on different inputs and can be executed on different CPUs, each with its own address space.

Walk-Through of a Typical OSCAR Installation

Installing a cluster with the OSCAR Toolkit is a straightforward process. If you have installed Linux before, you should have little trouble.

Currently, the OSCAR Project supports three Linux distributions: Red Hat 8.0, Red Hat 9.0 and Mandrake 9.0. The main Linux installation requirement is that an X windowing environment such as KDE or GNOME is installed; otherwise, a typical workstation install with software development tools should be sufficient.

After Linux is installed and configured on the head node, you can download the OSCAR tarball from the projects page, untar it and do the configure, make, make install routine.

OSCAR, by default, is installed to /opt/oscar. You can change this using the --prefix flag with configure. After OSCAR has been installed, you can start the OSCAR Wizard, which provides step-by-step installation menus for setting up your cluster.

To invoke the wizard, go to /opt/oscar and type ./install_cluster ethX. Here, ethX refers to the interface that is on the cluster network.

Figure 1. The OSCAR installation main menu. In less than ten steps the cluster is ready to compute!

OSCAR comes with many prebundled packages. Other packages available from the various repositories also may be of interest. To download those, simply click on Download Additional OSCAR Packages in the OSCAR Wizard and choose the package(s).
Next, you can select the packages you want to install. Packages have three main categories: core, provided and third party. Core packages must be installed and cannot be deselected. Provided packages are the ones the OSCAR team recommends you install, and third-party packages are all the remaining packages available from the repositories.

**Figure 2. OSCAR Package Downloader—additional packages can be downloaded in this menu.**

Configuration changes can be made to packages using the Configure Selected OSCAR Packages menu.

The next stage is to Install OSCAR Server Packages. This is non-interactive and basically sets up packages for use on the server. When it is done, you are alerted by a pop-up window.

**Figure 3. Non-interactive step where server packages are installed.**

Now the fun part begins. You can build a client image with the Build OSCAR Client Image step. In this step, you select a few options for the client image you want to build. This image then is pushed to your client nodes. You can provide a list of RPM packages to be installed on the base image, and you also can decide how to partition the hard drive and assign IP addresses. Lastly, you can choose the post-image action, such as rebooting the machine when imaging is done.

**Figure 4. Create a client image based on a user-provided package list and partition table.**

In the Define OSCAR Clients step, you can specify the domain name, base name of your clients, the number of nodes you want to bring up for this session and other network settings. After you click the Add clients button, these definitions are configured and the cluster is almost ready to be rolled out.
Next, you will want to set up networking for your cluster. Here, you can boot up your client nodes with PXE or floppy disk, and the OSCAR head node then collects the MAC addresses and you can assign them to particular hosts. Once this is done, the client nodes are imaged immediately. Typically, it takes anywhere from 10–30 minutes to image each node, depending on the speed of your hard drive. When deploying a cluster, multiple nodes can be imaged at the same time. We usually start up ten nodes to be imaged at a time so the head node does not get heavily loaded. With this staggered approach, you should be able to deploy a 64-node cluster within an hour.
Figure 6. Cluster nodes are network-booted, and their MAC addresses assigned to host entries.

After the nodes are imaged and rebooted, you can continue with the next step, which is to Complete the Cluster Setup. This again is a non-interactive step in which final installation configurations and other clean-up functions are performed.

**Figure 7. Non-interactive step where final installation configurations and clean-up functions are performed.**

Lastly, you may want to Test Cluster Setup. This runs a series of tests for the cluster install as well as for individual packages. If all goes well, you will pass all the tests, thus confirming your cluster setup is complete and ready to run computations.
The OSCAR Toolkit is easy to install and, in general, works with most hardware. However, if you run into problems, two mailing lists are available for help. The oscar-users list is the first place you should ask questions. Most of the core team frequently reads the list, and other users help out too. However, if you have questions regarding the development of OSCAR, there is the oscar-devel list. Both lists are closed lists; you need to subscribe before you can post to them.

New Features in OSCAR 4.0

We are planning to include various features in this upcoming release. They are partitioned into four main categories: NEST, node groups, Linux distributions and SIS.

NEST (Node Event and Synchronization Tools) is used to ensure that OSCAR packages are in sync with their centrally stored configurations across all the cluster nodes. Currently, when you install a new cluster node, post_install scripts for OSCAR and all packages have to be run on all the cluster nodes regardless of whether they need to. Although this model has worked on a medium scale, clearly a scalability limitation is imposed by it. The biggest change with NEST is that package configuration will be pulled from the server as opposed to being pushed to the clients. Operations will be executed only if they are necessary, which is a more elegant way than the brute-force execution scheme we are employing presently.

Node groups are arbitrary groupings of nodes in the cluster. With this new feature, it is possible to install and manage OSCAR packages selectively for these groups. In the upcoming release, we plan to support only the server and client node groups. In the future, however, users will be allowed to define their own groups.

One of the key features that defines OSCAR is our support for several Linux distributions. With the new release we hope to introduce support for Fedora Core 2 and 3, Red Hat Enterprise Server 3.0 and Mandrake 10. Supporting these distributions will also introduce support for IA-64 and x86-64 architectures.

System Installation Suite (SIS), which includes SystemImager, is the collection of programs that performs the image deployment of an OSCAR system. There are two main SIS-related improvements. First is the disk type autodetection. Traditionally, OSCAR cluster images are created with one type of hard disk in mind (either
IDE or SCSI). With this OSCAR-specific patch, you can use the same image to deploy on different machines with different types of hard disks, as long as the base hardware is similar.

Second, a tool is available so users can use specific kernel modules for booting the nodes for imaging. Sometimes it is difficult to get newer hardware to work with OSCAR, because the SIS kernel boot image does not have the supported drivers. With this tool, you can use an existing kernel with known working modules as the SIS boot kernel and use that to boot up your client nodes so they can be imaged. This feature will be included in the next SystemImager development release, which we hope to include in OSCAR 4.0.

Creating Packages for OSCAR

OSCAR supplies packages for commonly used cluster-aware applications. They are simply RPM packages with corresponding metafiles and installation scripts. These packages are created and maintained by the OSCAR core team and package authors. If there is an application you would like to install on your cluster, but did not find it available from OPD (OSCAR Package Downloader), please create a package for it. The OSCAR team is open to contributions and possibly even hosting an OSCAR software package you might create. The team extends this invitation to software developers too.

OSCAR packages reside in package repositories, which are decentralized Web spaces provided by the package authors to host the package files. The URIs of these repositories are stored in a master repository list.

Creating OSCAR packages is generally a straightforward process. If an RPM is readily available, you are already halfway done. What remains is to create some files to store metadata about the package/RPMs and also scripts to propagate configuration files for the entire cluster. This is relatively easy to do because certain assumptions can be made about an OSCAR cluster.

If, however, an RPM package is not currently available, you need to package the application in RPM format before continuing. Creating an RPM may or may not be easy, depending on the complexity of the application. You need to create a spec file and build the RPM(s) and corresponding SRPM with the source tarball.

The first OSCAR package that the Genome Sciences Centre (GSC) has put out is for Ganglia, a cluster monitoring system. We have started working on our second package, which is for Sun Grid Engine, an open-source batch scheduling system sponsored by Sun Microsystems. This should be available from OPD at the GSC OSCAR Repository at a later date.

Bioinformatics Applications and OSCAR Cluster

The most commonly used bioinformatics program is BLAST, a sequence alignment/search tool. Querying genetic databases with multiple gene sequences is highly amenable to parallelization; it is very easy to split this problem into many sub-jobs, with each sub-job querying the database with one set of input. Solutions exist that perform such database/query splitting for you, most notably the open-source mpiBLAST as well as the commercial version from Paracel. Parallelized versions of BLAST scale quite well but, of course, reach a point where adding more nodes simply does not increase performance due to the overhead of starting separate and smaller jobs on multiple nodes.

FPC is another application we use a great deal and is used for assembling, editing and viewing fingerprint-based physical maps. The original parallelized version of this application was developed at the GSC, but it was not batch system-aware. We recently have integrated parallel FPC with Sun Grid Engine so that users can easily request a specific number of nodes to run this application.
We also are developing a peer-to-peer application called Chinook for sharing bioinformatics services. Currently, it is still under development, and we are working on integrating it with our cluster. This potentially could link up grids in the future and provide an alternative to the Globus toolkit.

Currently, our 200-node cluster is being heavily used to discover and classify regulatory elements in the human genome. The human genome consists of roughly 30,000 genes, and we are employing different algorithms to scan each gene. Genes are grouped into batches of 1,500, and a typical run would take about four days on an Opteron 2.0GHz machine. Without cluster technology, this kind of research would not be possible.

**Conclusion**

The OSCAR Toolkit has come a long way since its first release. More and more people have found it easy to use and deploy—the key to getting clustering technology more widely adopted. Bioinformatics will continue to grow with high-performance computing. Soon, it is likely that cluster toolkits geared toward the bioinformatics community will become more widely available—a solution that includes all the tools for running parallel bioinformatics applications and is easy to install and deploy.

**Acknowledgements**

The author would like to thank Mark Mayo, Asim Siddiqui and Steven Jones for giving him the opportunity to work on the Linux cluster at the GSC and for identifying OSCAR as the tool to use. The author also would like to thank the OSCAR core team, developers and users for creating a great community for sharing HPC knowledge and information. Last, but not the least, the folks at NCSA who contributed much time and effort into the OSCAR Project. Personal thanks to Jeremy Enos, René Warren and Martin Krzywinski for providing valuable comments and suggestions for this article.

**Resources for this article:** [www.linuxjournal.com/article/7760](http://www.linuxjournal.com/article/7760).

Bernard Li is a High-Performance Computing Specialist at Canada's Michael Smith Genome Sciences Centre. He spends time managing the Linux cluster infrastructure and integrating bioinformatics applications with the cluster. He is a core developer for OSCAR and a fan of Sun Grid Engine. He can be reached at bli@bcgsc.ca.