

# Looking Inside the **OSCAR Cluster** Toolkit

*By Thomas Naughton; Stephen L. Scott, Ph.D.; Brian Barrett;  
Jeff Squyres; Andrew Lumsdaine, Ph.D.; Yung-Chin Fang;  
and Victor Mashayekhi, Ph.D.*

Cluster computing is becoming increasingly practical for high-performance computing (HPC) research and development. This article describes the 1.3 release of the Open Source Cluster Application Resources (OSCAR) toolkit and explains how the tools facilitate installation, configuration, cluster and workload management, and security. A look ahead shows how developers are working to extend the flexibility and simplicity of future OSCAR releases.

As the number of nodes in cluster configurations grows, installation, configuration, and administration become more challenging. The Open Source Cluster Application Resources (OSCAR) project was founded to study the challenges of cluster management and to provide a solution. The result, the OSCAR package, includes the best standards-based tools for cluster installation and management in one software bundle. OSCAR works on commodity equipment and uses open-source software.

The OSCAR project began in early 2001 and released the first public version, OSCAR 1.0, in April of the same year. OSCAR 1.3, the latest version, was released in July 2002 and runs on the Red Hat® Linux® 7.1 and 7.2 and MandrakeSoft Mandrake™ Linux 8.2 operating systems. Currently, the only hardware configuration supported by OSCAR is multiple compute nodes networked to a single master node. As the design of OSCAR progresses, more cluster configurations will be supported.

OSCAR was developed by the Open Cluster Group (OCG), a collaboration of major research centers and technology companies led by IBM, Indiana University, Intel, the National Center for Supercomputing Applications (NCSA) at the University of Illinois, and Oak Ridge National

Laboratory (ORNL). Other collaborators included Bald Guy Software, Dell, Ericsson, Lawrence Livermore National Laboratory (LLNL), MSC.Software, Silicon Graphics, Inc., and Veridian. The project is open to new collaborators.

## Simplifying cluster administration using OSCAR

The software components, or packages, included in OSCAR provide all the tools required to build and operate a high-performance computing (HPC) cluster. Four of these components are required for installation and configuration: System Installation Suite (SIS), the ORNL Cluster Command and Control (C3) tool suite,

Env-Switcher, and the OSCAR Wizard. Other components that enhance the functionality of the cluster include OSCAR Password Installer and User Management (OPIUM) and Secure Shell (SSH) configuration tools. OSCAR also includes commonly used HPC packages for message passing, workload management, and cluster monitoring.

## Automating installation and configuration with SIS

SIS is an image-based installer package that helps administrators install and configure cluster nodes

---

The OSCAR package  
includes the best  
standards-based tools  
for cluster installation  
and management in one  
software bundle.

---

over the network. SIS replaced Linux Utility for Cluster Installation (LUI) in OSCAR 1.2. Teams from the SystemImager and IBM LUI projects collaborated to develop SIS, which comprises the System Installer, SystemImager, and System Configurator tools.

System Installer creates an image<sup>1</sup> of a node's file system locally on the master node. Then SystemImager propagates the generated image over a network to various nodes. Finally, System Configurator modifies each propagated image with unique configuration information so that, upon reboot, a node will appear on the network as a useful working machine.

Administrators use SIS to bootstrap node installations—kernel boot, disk partitioning, file system formatting, and base operating system (OS) installation. They can also use the installation image to maintain the cluster nodes. Modifying a previously deployed image is as straightforward as modifying a local file system. An administrator can update the image and then use `rsync`<sup>2</sup> to update the local file system on the cluster nodes. This method can be used to install and manage an entire cluster.

### Administrating single and multiple clusters with C3

The C3 power tools, a product of the scalable systems research at ORNL, offer a command-line interface (CLI) for cluster system administration. These tools facilitate parallel execution and let users type a single command that can run on all cluster nodes in parallel, such as `cexec hostname`. These tools also allow scatter/gather operations so that file distribution and collection can occur across all cluster nodes; for example, `cpush myfile.txt` and `cget mydat.log`.

C3 version 3.1 includes 10 basic building blocks used by OSCAR for operations such as `sync_date`, which synchronizes the time across the cluster. The C3 tools were completely rewritten for version 3.x and offer many enhancements, including multi-cluster support—allowing command-line operations to execute across multiple clusters simultaneously—and improved documentation.

### Managing the environment with Env-Switcher

Created for OSCAR by members from Indiana University, the Env-Switcher package provides a simple CLI through which users can safely manipulate their environments. For example, when a user wants to add a package, Env-Switcher, not the user, modifies the environment variables, such as `PATH` and `MANPATH`,

OSCAR includes several packages that are commonly used on HPC clusters to enable message passing and other cluster operations.

as necessary. Using Env-Switcher, administrators can set up persistent system- and user-level attributes for various packages without needing to manually edit user dot files, such as `.bashrc`.

The OSCAR installation uses Env-Switcher to set system defaults, which can later be overridden by individual users. The canonical example is the selection of a default Message Passing Interface (MPI) implementation. At install time, an administrator can select a default, such as Local Area Multicomputer (LAM)/MPI or MPI Chameleon (MPICH), which allows OSCAR to set up the correct environment based upon the install time

choice. Based on this selection, OSCAR can set up the correct system and user environments. Additionally, Env-Switcher enables non-interactive shells such as `rsh` and `ssh` to receive appropriate environment settings, so administrators do not need to edit user-level shell configuration files.

### Installing a cluster using the OSCAR Wizard

The OSCAR Wizard provides a graphical user interface (GUI) to assist with cluster installation. It comprises a set of screens that guides a user through the process of creating an image, defining the number of nodes, configuring the network settings, and confirming the cluster setup was successful. Another set of screens lets users add and remove nodes from the cluster.

Current development intends to create a CLI, which future wizards will use to drive the installation. This CLI will improve scripting capabilities and offer another option to system administrators who do not wish to use the GUI-based wizard. This CLI will provide the necessary access to the OSCAR Data Repository (ODR) while performing the cluster configuration and installation tasks.

### Managing cluster accounts using OPIUM

The OPIUM package provides a mechanism to synchronize cluster account information. The standard user management tools (`user[add|del]`, `group[add|del]`) are made cluster-aware, meaning that an administrator creates an account as usual, but upon command completion, the account exists on all OPIUM-managed cluster(s). The OPIUM package also configures the system so that user accounts create SSH keys upon first login to the system. Having the keys in place lets users access cluster nodes without a password prompt.

<sup>1</sup> Here, *image* is defined to be a directory tree that comprises an entire file system for a machine.

<sup>2</sup> `rsync` is a tool to transfer files and is similar to `rcp` and `scp`.

## Securing the cluster with SSH

SSH provides a secure replacement for RSH. Added security increases configuration requirements, burdening system administrators and cluster users who wish to use SSH on their cluster. OSCAR installs OpenSSH and automatically sets up all the required configuration files for SSH, transparently replacing RSH with SSH.

## Enabling message passing and monitoring

OSCAR includes several packages that are commonly used on HPC clusters to enable message passing and other cluster operations, such as the following:

- ▶ **Parallel message passing libraries:** LAM/MPI, MPICH, and Parallel Virtual Machine (PVM)
- ▶ **Batch queuing system:** Open Source Portable Batch System (OpenPBS)
- ▶ **Scheduler:** Maui
- ▶ **Monitoring system:** Ganglia (CLI and Web-based view of cluster statistics)
- ▶ **Security:** pfilter (packet filter configured with reasonable settings)

## Installing OSCAR and running OSCAR Wizard

The OSCAR Wizard lets both novice users and seasoned system administrators build and configure an HPC cluster in a few easy steps. Before using OSCAR, the user must complete the following on the master node (the server from which the image will be deployed to the client nodes):

- ▶ Install an OS using standard methods for an OSCAR-supported distribution (for example, with the Red Hat CD-ROM).
- ▶ Build the server with X Window System support.
- ▶ Make sure the networking is configured and working for the server.
- ▶ Provide the internal cluster interface to the installation script (such as `./install cluster eth1`).
- ▶ Copy the RPM (Red Hat Package Manager) files from the distribution CD-ROM(s) into a directory on the server; the default location is `/tftpboot/rpm`.

After setting up the server, download OSCAR from the project Web page and extract it. Then run the installation script, which copies necessary files to the server, sets up requisite services, and starts the OSCAR Wizard.

Figure 1 outlines the steps that the OSCAR Wizard performs. The steps change slightly among versions, but the basic process remains the same:

1. Prepare the server, possibly selecting defaults (such as default MPI implementation).
2. Create an image for the client nodes from a user-specified description, based on a package list and disk partitioning scheme.
3. Define a set of clients: number of nodes in cluster, naming scheme, networking settings (IP, netmask, and so forth), and image name to be installed on clients.
4. Collect Media Access Control (MAC) addresses (the identifier used to associate a node with its IP address) for the client nodes.
5. Install the compute nodes. First, configure the server to respond to DHCP requests from the client nodes. Then, boot the nodes using either a bootable floppy disk or a network-enabled BIOS boot mechanism such as Preboot Execution Environment (PXE). The nodes will contact the server, which will convey their identity. They will then perform the basic operations to complete the installation, such as partitioning the hard drives, formatting the file systems, and copying the files from the server.
6. Configure remaining packages and synchronize time.
7. Run the test suite to verify that key cluster components and services, such as OpenPBS, PVM, LAM/MPI, MPICH, and Network File System (NFS), are operating properly.

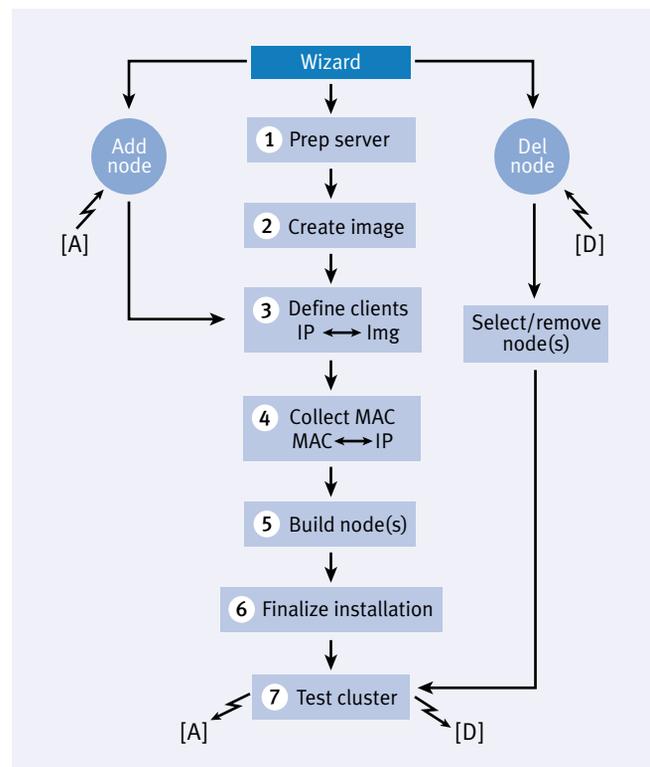


Figure 1. Installing a cluster with the OSCAR Wizard

Constructing a cluster is typically an ongoing task. As of OSCAR 1.3, the GUI lets users add and delete nodes just as they did when first installing the cluster. Users define a set of nodes and associate an image with it using the MAC-to-IP address mapping. This identity is then transferred to the client nodes as they boot from either a floppy or network boot mechanism. The process uses the same screens, and users can re-run the tests to verify the installation updates.

Removing a node is slightly simpler and requires notifying all relevant packages that their configurations must reflect the node deletion. The wizard modifies configuration files for tools such as C3 or OpenPBS, which maintain state about the cluster nodes and therefore must be updated when nodes no longer exist.

In general, the OSCAR Wizard reduces the time required to build a functional cluster, increases consistency among cluster builds, and reduces the expertise necessary to build an HPC cluster.

### Improving OSCAR through current development efforts

OSCAR offers a solid means for building and configuring a cluster. Current development efforts focus on increasing installation flexibility and extending cluster management capabilities after deployment. The following discussion highlights some of these features and goals.

#### Adding a standard interface for cluster management

Although SIS and C3 are powerful tools, administrators often need better high-level management tools. The OSCAR group seeks to remedy this issue in the form of a standard interface to a set of tools for node addition, deletion, and package management. The interface will mask the underlying mechanism, such as SIS or SIS plus C3, thus allowing others to extend or replace the management system.

#### Making the OSCAR architecture modular

As OSCAR has evolved, a clear burden has been the integration process required for a major release. Removing the tight coupling of all packages contained in an OSCAR release has eased integration; OSCAR 1.3 features a modular packaging system prototype that removes much of this coupling. Developers plan to extend this decoupling to the OS installation as well. A modular architecture will allow administrators to install a set of nodes using other means (such as CD-ROM or Red Hat Kickstart Configurator) and then use OSCAR for the remaining installation and configuration.

Not only does the modular packaging system remove the tight coupling between OSCAR packages, it also—and possibly more importantly—enables developers outside of the OSCAR team to contribute packages, extending the base components of OSCAR. The interface to the modular architecture requires a standard

package, which is currently the widely used RPM system from Red Hat. In addition to using RPM, the package creator may provide a set of scripts to perform configuration steps not possible within the RPM framework. As work progresses, the architecture document on the OSCAR Web site will include an application programming interface (API) for package maintainers.

#### Accessing reliable information through the ODR

Current OSCAR designs offer very limited data about the cluster to packages or to system administrators. The OSCAR Data Repository (ODR) is a generic interface to such data, which will be especially important as the flexibility of OSCAR grows. The ODR API will likely resemble a SQL interface. Access to the data will be available from any node in the cluster.

The API to the repository will be coupled with the improvements to the standard OSCAR Wizard. The modular packaging system will also allow scripts to query the ODR at specified times to obtain information such as number of nodes and master node IP address. This functionality will enable many cluster-aware packages to configure themselves, reducing the load on cluster system administrators.

#### Improving the OSCAR Wizard

The OSCAR GUI and companion CLI facilitate better usability through an intuitive, extensible interface. Administrators usually prefer to use CLIs for expediting complex commands and for creating new functionality. The GUI offers greater ease of use. The GUI and CLI enable administrators to function easily without regard for the underlying implementation. Offering both kinds of interface suits every administration style.

Adding an underlying CLI lets developers contribute different GUIs without overhauling the entire system. GUIs considered for OSCAR include Perl/Tk, Webmin, and Python/Tkinter. In the future, other developers might contribute an ncurses-based GUI that also uses the CLI—the underlying CLI would remain the same for accesses to the ODR.

### Looking ahead to OSCAR 2.0

The OSCAR project has emerged as a useful tool for cluster installation and administration. The introduction of SIS into OSCAR at version 1.2 greatly simplified the steps necessary to build and configure a cluster. Subsequent releases have brought other features such as a modular packaging system, support for multiple distributions, and support for Intel® 32-bit and 64-bit architectures. As the OSCAR project grows, developers seek to balance flexibility and simplicity. OSCAR 2.0 will offer improved cluster management, a modular architecture, enhanced ODR, and extended GUI and CLI Wizard tools.

As development progresses, the project will begin to extend, and even define, “best cluster practices.” These extensions will lead to improved cluster management, at both the node and package levels. In the meantime, the OSCAR project continues to be an effective cluster computing solution, providing powerful tools for cluster installation and management. 

## References

Fang, Yung-Chin, Tau Leng, Ph.D., Victor Mashayekhi, Ph.D., and Reza Rooholamini, Ph.D. “OSCAR 1.1: A Cluster Computing Update.” *Dell Power Solutions*, Issue 4, 2001.

Hsieh, Jenwei, Tau Leng, and Yung-Chin Fang. “OSCAR: A Turnkey Solution for Cluster Computing.” *Dell Power Solutions*, Issue 1, 2001.

IBM. *LUI Project: Summary*. <http://oss.software.ibm.com/developerworks/projects/lui>

Luethke, Brian, Thomas Naughton, and Stephen L. Scott. “C3 Power Tools: The Next Generation...” Paper to be presented at the Austrian-Hungarian Workshop on Distributed and Parallel Systems (DAPSYS 2002), Linz, Austria, September–October 2002.

Naughton, Thomas, Stephen L. Scott, Brian Barrett, Jeff Squyres, Andrew Lumsdaine, and Yung-Chin Fang. “The Penguin in the Pail—OSCAR Cluster Installation Tool.” Paper presented at the World Multiconference on Systemics, Cybernetics and Informatics (SCI 2002), Orlando, Fla., July 2002.

Oak Ridge National Laboratory. *Project C3: Cluster Command & Control (C3) home page*. <http://www.csm.ornl.gov/torc/C3>.

Sterling, Thomas, et al. “BEOWULF: A parallel workstation for scientific computation.” *Proceedings of the 24th International Conference on Parallel Processing, Volume I*. Boca Raton, Fla.: CRC Press, 1995.

“System Installation Suite Project.” <http://sisuite.sourceforge.net>  
Tridgell, Andrew, and Paul Mackerras. “The rsync algorithm.” *Technical Report TR-CS-96 05*. Canberra: Australian National University, Department of Computer Science, June 1996. See also: <http://rsync.samba.org>.

**Thomas Naughton** ([naughtont@ornl.gov](mailto:naughtont@ornl.gov)) is a research associate in the Computer Science and Mathematics Division, Oak Ridge National Laboratory. Thomas has a B.S. in Computer Science and a B.A. in Philosophy from the University of Tennessee–Martin, and an M.S. in Computer Science from Middle Tennessee State University.

**Stephen L. Scott, Ph.D.** ([scottsl@ornl.gov](mailto:scottsl@ornl.gov)) is a research scientist at the Computer Science and Mathematics Division, Oak Ridge National Laboratory. Stephen has a B.A. from Thiel College, Greenville, Penn., and an M.S. and Ph.D. from Kent State University.

**Brian Barrett** ([brbarret@osl.iu.edu](mailto:brbarret@osl.iu.edu)) is a graduate student at the Open Systems Laboratory, Indiana University, Bloomington. Brian has a B.S. from the University of Notre Dame.

**Jeff Squyres** ([jsquyres@osl.iu.edu](mailto:jsquyres@osl.iu.edu)) is a research associate at the Open Systems Laboratory, Indiana University, Bloomington. Jeff has a B.A. in English, a B.S. in Computer Engineering, and an M.S. in Computer Science and Engineering from the University of Notre Dame.

**Andrew Lumsdaine, Ph.D.** ([lums@osl.iu.edu](mailto:lums@osl.iu.edu)) is the associate director of the Open Systems Laboratory at Indiana University, Bloomington. Andrew has a Ph.D. from the Massachusetts Institute of Technology.

**Yung-Chin Fang** ([yung-chin\\_fang@dell.com](mailto:yung-chin_fang@dell.com)) is a member of the Scalable Systems team at Dell. Yung-Chin has a bachelor’s degree in Computer Science from Tamkang University and a master’s degree in Computer Science from Utah State University. He is currently working on his doctorate degree.

**Victor Mashayekhi, Ph.D.** ([victor\\_mashayekhi@dell.com](mailto:victor_mashayekhi@dell.com)) is a senior technical member of the Enterprise Computing Solutions Group at Dell. His product development responsibilities at Dell have included all the cluster product offerings from Dell. Victor has a B.A., M.S., and Ph.D. in Computer Science from the University of Minnesota.

Work by Thomas Naughton and Stephen L. Scott was supported by the U.S. Department of Energy, under Contract DE-AC05-00OR22725. Work by Brian Barrett was supported by a Department of Energy High Performance Computer Science Fellowship. Work by Jeff Squyres and Andrew Lumsdaine was supported by a grant from the Lilly Endowment.

## FOR MORE INFORMATION

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

**The OSCAR Project:** <http://oscar.sourceforge.net>

**Previous articles on OSCAR:**  
<http://www.dell.com/powersolutions>