Stateless clustering using OSCAR and PERCEUS

Abhishek Kulkarni
Computer Science Dept
Indiana University
Bloomington
adkulkar@cs.indiana.edu

Dr. Andrew Lumsdaine
Open Systems Lab
Indiana University
Bloomington
lums@osl.iu.edu

Abstract

OSCAR has matured over the years into a very capable, yet simple high-performance clustering platform solution. Ease of building and administrating a cluster, effective resource management and scalable performance are some of the issues which have gathered a lot of focus in the HPC domain lately. This technical paper describes the integration of OSCAR and PERCEUS, which aims to enable the deployment of stateless nodes in a cluster. This promises to fill a much needed niche in performing a diskless cluster installation with OSCAR. The single point of administration and ease-of-use would help novice users as well as experienced cluster administrators to deploy clusters faster.

1. Introduction

With high-performance compute clusters reaching the petaflops mark, addressing the large administration needs of these systems is an interesting challenge faced by cluster management toolkits. Cluster software has come a long way from the traditional approach taken of configuring each node individually[3] to platforms which allow easier building, managing and scaling of cluster systems. These cluster toolkits are expected to best utilize the advances in cluster architectures, high-performance processors, high-speed affordable interconnects and faster centralized storage systems to offer a seamless HPC platform.

The emerging proliferation of HPC in industry and labs for research has necessitated the use of advanced high-performance hardware and software tools. According to a study conducted on HPC application software solutions in 2005, there is an urgent need of capable HPC application software[4]. Independent software vendors (ISVs), national laboratories and universities are stepping forward to offer turn-key based enterprise clustering solutions that provide an out-of-box experience.

The OSCAR toolkit is a project developed by the OSCAR working group formed within the Open Cluster Group. Work on OSCAR began in April 2000 and over the years it has matured into a complete clustering platform solution offering support for various parallel libraries, schedulers, filesystems and other cluster monitoring and management utilities[7]. With downloads reaching almost 200,000[1], OSCAR is one of the most widely used cluster computing toolkits today.

Of the known cluster software paradigms, the adoption of diskless cluster configurations has been on a rise over the past few years. A study comparing diskfull and diskless clusters[5] confirms no observed performance differences between both in the presence of a suitably high-speed interconnect. Besides, diskless clusters are known to have a host of other advantages over the diskfull architectures. Power consumption and heat dissipation are some of the issues plagued by HPC clusters in general. Hard disk drives account for a significant proportion of the total power consumption of a cluster. Drive manufactures are pushed to take steps to reduce the power consumed by the disk drives by a variety of approaches. A diskless clustering solution considerably reduces the overall power consumption and reduces the number of moving parts in the cluster. This increases reliability and availability[16], thereby increasing the MTBF (mean time between failures).

The terms diskless and stateless are often used interchangeably to refer to the node-image provisioning method in a cluster. However, stateless clusters can have disks that could be used for additional swap storage if the need be. Stateless provisioning serves a fresh file system to the nodes on every boot, thereby offering non-persistent disk images. Compute clusters with stateless nodes are simpler to deploy, in a way that adding a node to the cluster only requires rebooting the node. The node acquires the software image and relevant configuration settings by itself, as soon as it is powered-on.

The Perceus toolkit is a successor to Warewulf[12],

[1] Statistics snapshot taken on January 14, 2008

1550-5243/08 $25.00 © 2008 IEEE
DOI 10.1109/HPCS.2008.12

which is one of the most widely used toolkits for diskless clustering. Developed by the creators of Warewulf[1], Perceus makes use of the ”best practices” techniques learned from Warewulf for large scale provisioning of stateless nodes. It uses a hybrid NFS-ramdisk approach[8] to provision nodes which allows better scaling and expansion of the cluster than the traditional NFS-only approach. This reduces the support effort considerably and offers a better control over the cluster software stack[9].

1.1. Motivation

A number of cluster installation suites based on different cluster paradigms[15] namely HA-OSCAR, SSI-OSCAR, SSS-OSCAR and Thin-OSCAR have been developed by the OCG working groups. These derivatives leverage the facilities provided by the core OSCAR framework and offer specialized cluster application environments based on specific needs. Until recently, no clustering paradigm associated with OSCAR has been able to provide a completely stateless clustering scheme successfully. Thin-OSCAR[13] which capacitated diskless cluster installation support is deprecated.

OSCAR KernelPicker, a utility to handle client kernels allows a user to build a ramdisk image and easily deploy custom kernels from various sources[17]. However, it does not manage building ramdisks for diskless nodes since there is no networking support in the ramdisk system.

This technical paper discusses the integration of OSCAR toolkit with PERCEUS to support stateless provisioning of nodes. Integration of the two will fill a much needed niche in cluster computing. Moreover, the stated advantages of stateless clustering can be leveraged by OSCAR to offer different modes of cluster installation to the user. The architecture, concept and detailed discussion for the integration are provided in this paper. Since the work is still in alpha stage, the actual performance results have not been discussed. The future work section considers the porting of this project to varied systems like the Sony PlayStation 3.

2. Background

This section provides an introduction to OSCAR and the PERCEUS toolkit. The remainder of the document assumes a familiarity with their architecture and operation. The integration details and the interaction between both the cluster management toolkits are illustrated in the later sections.

2.1. OSCAR

OSCAR is a self-contained cluster installation utility which can facilitate the setup, administration and maintenance of a Beowulf style cluster typically used for high-performance computations. The core OSCAR framework relies on the SIS framework (SystemImager Suite) for building virtual images of the target system which are then pushed on to the nodes in the cluster. The OSCAR database stores the metadata and the information central to a cluster which can be accessed using the OSCAR Database Access API (ODA). Additional tools include the Env-Switcher which manages the environment settings for the cluster and the C3 (Cluster Command and Control) power tool, a parallel and distributed shell to issue commands to the nodes in the cluster[15].

The latest upcoming release of OSCAR (OSCAR 5.1) introduces a better meta-package management system called OPKG which relies on the native package distribution infrastructure of the underlying OS. These OSCAR packages (OPKG) include the binary software packages, as well as the configuration scripts, documentation and tests for that package[14]. The cluster utilities and tools that are provided with OSCAR are encapsulated as OPKGs so that the installation, removal and management of these packages become easier. This modularity in the packaging system of OSCAR has led to a number of OSCAR derivatives which are based on the core OSCAR framework. These extend OSCAR’s functionality to enable the installation of improved fault-tolerant clusters, offer single system-wide image view and support diskless clustering[13].

2.2. PERCEUS

Perceus is an enterprise-level cluster toolkit for installation and management of stateless clustering platforms irrespective of their size and architecture. It uses a hybrid NFS and ramdisk model to enable faster deployment and provisioning of nodes in a cluster. With a single point-of-administration and central management interface, the support effort for maintaining a cluster is reduced.

The Perceus cluster architecture consists of a central head node called the ’Perceus Master’ which manages and monitors the slave nodes of the cluster (Refer Figure 1). The slave nodes are usually booted using PXE (Preboot Execution Environment) over the network interface card which requests for an IP address from the Perceus master via DHCP (Dynamic Host Configuration Protocol). The Perceus Network Manager internally uses DNSmasq which acts as the TFTP (Trivial File Transfer Protocol) and DHCP server to facilitate the network booting of nodes.

The Perceus master serves a Perceus boot kernel to the nodes via TFTP. Consequently, the Perceus boot kernel loads the node’s runtime kernel that communicates with a provisioning daemon (provisiond) located on the master server to prepare the nodes for their provisionary states. Once a contact with the node has been established, the Perceus daemon adds the node details to its database.

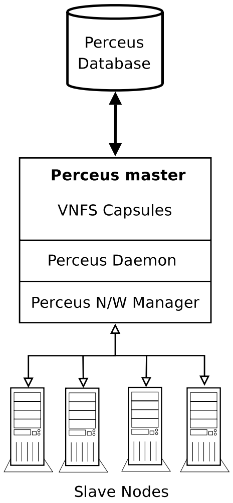


Figure 1. High-level view of Perceus architecture

A Perceus VNFS (Virtual Node File System) image is mounted onto the nodes to prepare them for running stateless. The nodes operate in a completely stateless way, such that their configuration data and state are not stored locally. The information pertaining to these nodes is maintained in a database on the Perceus master which monitors the status of the nodes in the cluster. The VNFS image is a chrooted image containing the complete copy of an actual filesystem that is mounted and/or pushed on to the client nodes. Thus, it is easier to upgrade or make changes to the software environment of the cluster simply by making changes to the VNFS image. These VNFS images are packaged with other scripts and utilities into a VNFS capsule which ensures proper stateless provisioning of the slave nodes [8].

2.3. Integration of OSCAR and PERCEUS

This project maintains maximum integrity of both the toolkits, while taking the "best practices" techniques out of them. OSCAR provides an excellent and easy-to-use GUI to configure and set up a cluster which can be leveraged to provide an easier installation and management process. Since Perceus acts as the node software provisioning backend for OSCAR, upcoming features in Perceus like stateful or hybrid provisioning and virtualization can be used along with OSCAR to offer promising solutions.

3. Architecture

The functional architecture for the integration of OSCAR and Perceus lays emphasis on the idea that maximum interaction between both the toolkits should be achieved, while still maintaining their complete integrity. Functional components have been inserted wherever required to 'plug' the differences between the two toolkits. The figure below illustrates a top-level design for this integration.

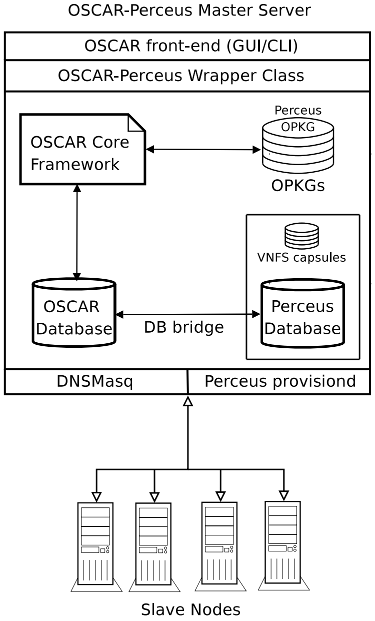


Figure 2. Functional architecture for integration of OSCAR and Perceus

OSCAR acts as a front-end for the installation and management of the cluster. Additional GUI interface to tweak the Perceus configuration parameters have been integrated within OSCAR's graphical user interface (Refer Figure 4). Previously, the SIS (SystemImager Suite) framework handled the complete node image generation, deployment and maintenance. With this integration, the complete SIS infrastructure has been replaced by Perceus. The stateless provisioning mechanism for nodes is provided by Perceus which acts as a backbone for the new infrastructure. To maintain consistent interaction between the two, the middleware layer is modified to make use of methods and services offered by OSCAR and Perceus.

The general flow of events during the installation of OSCAR/Perceus master server is depicted in Figure 3. The control changes back and forth between the two during the installation process.

4. Implementation

For the integration, the system was identified into functional modules which aimed at easing out the interaction

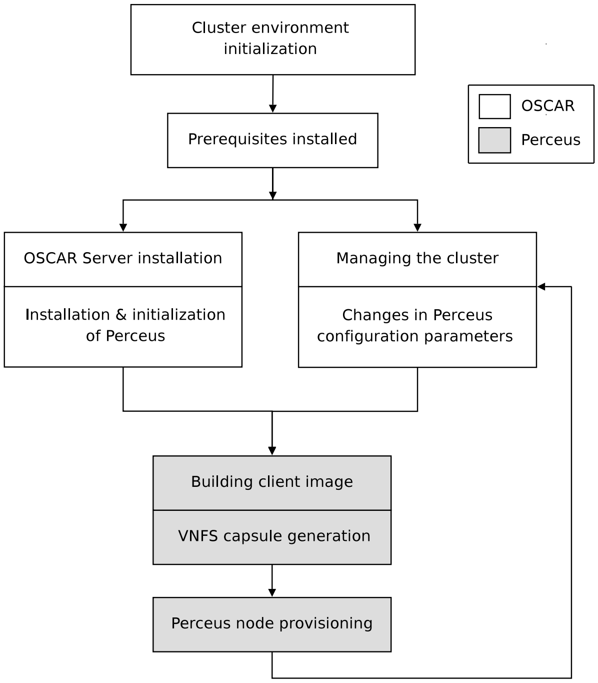

Cluster environment initialization
Prerequisites installed
OSCAR
Perceus
OSCAR Server installation
Installation & initialization of Perceus
Managing the cluster
Changes in Perceus configuration parameters
Building client image
VNFS capsule generation
Perceus node provisioning

Figure 3. Flow of events during the master server installation

between the two toolkits thus avoiding any profound modifications in their functioning. This would ensure the coexistence of both Perceus and SystemImager to perform diskfull and diskless cluster installations respectively. A flag `--diskfull` or `--diskless` can be specified during the execution of installation process to decide the provisioning backend to be used.

4.1. Perceus OPKG

Packaging Perceus as a third-party software with OSCAR involves creation of a Perceus OPKG (OSCAR package). The OPKG allows developers to describe software with a cluster-wide view. OSCAR users typically do not have to care about the installation of Perceus as it is handled internally by the OPKG framework. The configuration and initialization of Perceus is done by scripts included in the OPKG. The package dependency management infrastructure is effectively used by the OPKG to ensure that the head node OS meets all the package dependency requirements.

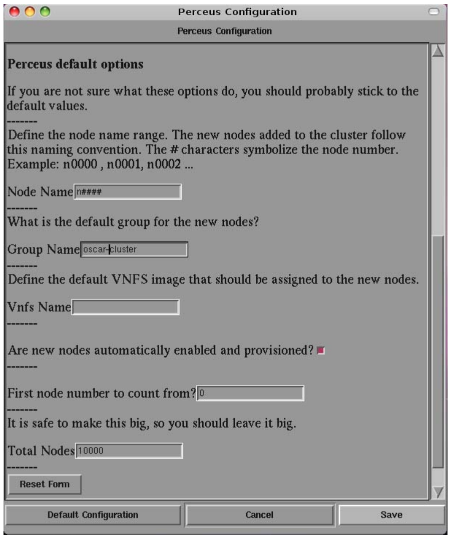
The Perceus OPKG consists of

- config.xml
- pre/post installation scripts
- documentation for the package
- scripts for testing the package

The config.xml file complies to a standard XML schema dictated by the OPKG API documentation[2]. The steps to be taken for proper installation and configuration have been translated into the Perceus OPKG API scripts which are executed at several stages during the installation.

4.2. Perceus configuration and initialization

Based on the default parameters for the installation of a Perceus cluster, a default template file for configuring Perceus is included in the OSCAR package. The GUI installation of OSCAR offers the users ability to change configuration parameters through the OSCAR configurator (Figure 4). Any changes in the default parameters are reflected back to the original Perceus configuration files typically located at */etc/perceus/perceus.conf* and */etc/perceus/defaults.conf*



Perceus default options

If you are not sure what these options do, you should probably stick to the default values.

Define the node name range. The new nodes added to the cluster follow this naming convention. The # characters symbolize the node number.
Example: n0000 , n0001, n0002 ...

Node Name n####

What is the default group for the new nodes?

Group Name oscar-cluster

Define the default VNFS image that should be assigned to the new nodes.

Vnfs Name

Are new nodes automatically enabled and provisioned? ☑

First node number to count from? 0

It is safe to make this big, so you should leave it big.

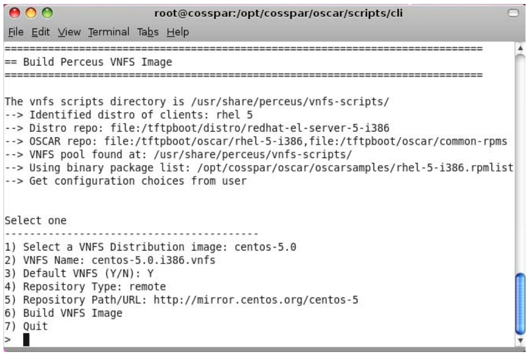
Total Nodes 10000

Figure 4. The Perceus Configurator

The first time Perceus is run, it goes through an initialization sequence which collects the information about the installation environment for Perceus. The network interface parameters, SSH keys, server configurations are updated during this step executed in the OPKG API post-installation script stage.

4.3. Client VNFS Image Generation

VNFS capsule generation for Perceus is done manually using pre-built Perceus VNFS (Virtual Node File System) generation scripts. These are shell scripts freely distributed with Perceus which build a chroot image of the node system. The ”Build Client VNFS Image” stage of OSCAR installation calls these scripts for the client image generation. Additional parameters can be passed to these scripts to choose the method of generation. For instance, the ability to generate images from a local or a remote distribution repository is available. Based on these parameters, OSCAR patches these scripts and facilitates the generation of custom VNFS images ready to be provisioned. The VNFS generation scripts hold the map for construction of compute node images. These scripts use the package manager YUM to retrieve the packages required to build the image of a compute node. Therefore, building a custom image involves editing these scripts manually. Our interface provides a way to customize these scripts automatically to a certain extent.


root@cosspar:/opt/cosspar/oscar/scripts/cli
File Edit View Terminal Tabs Help

```
==============================================================================
== Build Perceus VNFS Image
==============================================================================

The vnfs scripts directory is /usr/share/perceus/vnfs-scripts/
--> Identified distro of clients: rhel 5
--> Distro repo: file:/tftpboot/distro/redhat-el-server-5-i386
--> OSCAR repo: file:/tftpboot/oscar/rhel-5-i386,file:/tftpboot/oscar/common-rpms
--> VNFS pool found at: /usr/share/perceus/vnfs-scripts/
--> Using binary package list: /opt/cosspar/oscar/oscarsamples/rhel-5-i386.rpmlist
--> Get configuration choices from user


Select one
-----------------------------------------
1) Select a VNFS Distribution image: centos-5.0
2) VNFS Name: centos-5.0.i386.vnfs
3) Default VNFS (Y/N): Y
4) Repository Type: remote
5) Repository Path/URL: http://mirror.centos.org/centos-5
6) Build VNFS Image
7) Quit
>
```

Figure 5. Perceus VNFS image generation CLI

The OSCAR-Perceus wrapper class allows to set these images to be served to the slave nodes. These images can also be assigned to different node groups to maintain multiple software distributions on a cluster. Simple distribution swapping can help in switching cluster software stacks (similar to rolls in Rocks) for specialized roles of a HPC cluster.

4.4. OSCAR – Perceus Wrapper Class

The main goal of the OSCAR - Perceus wrapper class is to maintain consistent interaction between both the toolkits and maximize the abstraction of their interface. The end-user does not have to bother about the commands for adding, removing nodes; generation/configuration of VNFS images etc. OSCAR provides an intuitive ”one-click” method for installing and managing the cluster. For the expert users, the CLI provides a way to pass in additional parameters while issuing commands to OSCAR and Perceus.

The wrapper class is a module written in Perl (*Cosspar.pm*) which parses and passes the commands to Perceus and/or OSCAR as desired. Consistency of the cluster metadata is essential for monitoring the cluster over time. The wrapper class ensures this by hooking on to the OSCAR-Perceus DB bridge to store or retrieve this data. The OSCAR wizard interfaces with this class such that corresponding wrapper functions get called when OSCAR installation wizard steps are ”clicked” by the user.

4.5. OSCAR – Perceus DB bridge

With the introduction of the OSCAR database API [11], it has become easy to interact with the OSCAR database. The ODA creates an abstraction layer around the OSCAR database and thus the database-related routine tasks are easily accessible during installation and configuration of the cluster. Owing to the fact that a number of OSCAR derivatives have been developed (even for non-HPC environments) the previous version of ODA, deemed unsuitable to handle the growing database needs, was replaced with the new ODA infrastructure [10].

OSCAR uses MySQL (with support for PostGRE SQL) as the back-end database while Perceus uses the flat-file Oracle BerkeleyDB at its core. To provide a sufficient abstraction layer between these, the database bridge is written as a Perl module which utilizes the methods defined in ODA (*Oda.pm*) and Perceus’s Database API (*Perceus/Db.pm*). A daemon (cosspard) running in the background captures the events in the cluster, like those initiated by addition of new nodes, and duplicates this information for both the toolkits. The commonalities in the database APIs of both the clustering toolkits are identified, and an interface to consistently interact with the databases is provided under the purview of this database bridge.

The database synchronization during start up ensures that the initial environment information is reflected properly across OSCAR and Perceus. Data duplication or redundancy can be eliminated using a common database, however this would affect the integrity of OSCAR or Perceus and require profound modifications in their respective architectures.

5. Developmental roadmap

While still in alpha quality, a lot remains to be done for enabling seamless compatibility and integration between OSCAR and Perceus. Taking into account the active development on both the toolkits over the last few months and considering the significant changes in the code-base for OSCAR’s next major release, the development of this project has turned out to be slower than expected.

5.1. Completed modules

Barring further changes in the OSCAR package infrastructure, the following modules should successfully lead to the automatic installation, configuration and initialization of the Perceus package. The configuration parameters can be tweaked or changed through the OSCAR configurator. Generation of client node images (VNFS) is possible through a GUI and CLI interface. The text-based installation has been completed while the integration of GUI wizard still remains to be done.

The following modules have been completed in entirety, and their extensive testing has been done.

- Perceus OPKG
- Perceus configuration and initialization
- Client VNFS Image Generation
- OSCAR - Perceus DB bridge

5.2. Remaining modules

The Perceus-OSCAR wrapper class is a Perl module which acts as a middleware and abstracts the interface between the two for installation and management of the cluster. The OSCAR GUI wizard and command-line interface are independent modules but a goal of the OSCAR CLI project was instead to have the GUI call the CLI[6]. The wrapper class integrates the two modes such that the GUI calls execute command-line wrapper functions in the background. The DB bridge forms a reliable communication channel between the OSCAR database and the Perceus database using ODA bindings and the Perceus database interaction module respectively. The DB-bridge class hooks on to the wrapper class to ensure that a consistent cluster-wide view of information exists.

Since OSCAR is tightly coupled to the SIS (SystemImager Suite) framework, the OPKG scripts for a number of third party packages bundled with OSCAR (like Ganglia, OpenMPI, MPICH) have been written around the common SIS functions. To support these packages, the OPKG scripts need to be changed so that they use the stateless provisioning scheme.

The modules below have been developed only partially, that much of the desired functionality and testing on these still remains.

- OSCAR - Perceus Wrapper Class
- Support for all OSCAR 3rd-party packages
- Testing

6. Future Work

While a few of the integration features still remain alpha quality, the goal of basic integration of OSCAR with Perceus was attained. A lot of work pertaining to the tighter integration between both the toolkits, supporting of a variety of commonly used cluster-related packages, porting the project to different architectures/systems and better cluster resource management still remains to be done. A few of these are discussed in the following sections.

6.1. Tight Coupling

The current approach creates a wrapper class around Perceus which is used by OSCAR to communicate with the nodes. This is done to ensure that the maximum integrity of both the toolkits is maintained. Tighter coupling and minimal redundancy between the functional components of OSCAR and Perceus can help in improving the efficiency of the integrated package and to make it light-weight and faster. For instance, a single database can be used to maintain the cluster-wide state information of the nodes. Currently, this would be difficult to accomplish considering the active development being carried out on both the toolkits.

OSCAR has a mature and configurable framework for constructing the compute node images. VNFS image generation support can be added to this framework to build stateless images. This involves changing the package-list file and adding pre and post image generation scripts to add stateless capabilities to the image. This would give the user more flexibility in configuring and building the stateless images than that given by Perceus's VNFS image generation scripts.

6.2. PS3 clustering

Lately, a lot of focus has been gathered towards PlayStation 3 clustering considering the computational capabilities of the CELL processor used in PS3. Many turnkey PS3-based cluster solutions are available from cluster vendors. A number of Linux distributions like Yellow Dog Linux, Fedora, Gentoo and openSUSE among others are offering support for the PS3. By generating a PS3 Linux VNFS image, Perceus can be used to provision diskless PS3 nodes.

The Perceus kernel makes use of kexec to boot a new kernel in a much similar way as kboot does. The kboot on the PS3 nodes communicates with the PS3 Perceus master to fetch the boot kernel. Once the Perceus boot kernel is obtained, it loads and transfers the control to the host runtime operating system kernel. Lastly, all the compilers, tools and utilities used by the various programming models available on the PS3 can be bundled into a custom VNFS for provisioning diskless PS3 nodes.

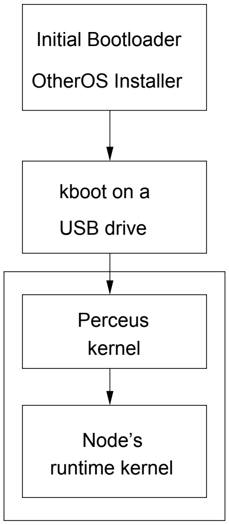


Figure 6. Perceus on PS3 - Boot sequence

7. Conclusions

The software architecture of Perceus and its cluster provisioning details were studied. The integration of both the toolkits was described, also mentioning the future work that is on the developmental roadmap.

Since the motive of this project is to demonstrate diskless cluster installation procedure using OSCAR, no real performance metrics were obtained. These metrics vary depending on the cluster hardware setup and the type of interconnect and storage system used. A rough estimate of performance comparisons between diskless and diskfull clusters can be observed from a study conducted by Dell[5].

Considering the fact that many of the compute cluster architectures are going the "diskless" way, the ability to do diskfull, diskless and hybrid provisioning of nodes would make it easier to install and maintain clusters with OSCAR. This is a good step towards 'easy-to-setup and manage' clusters for the large userbase of OSCAR toolkit.

8. Acknowledgment

The authors would like to thank Tim Mattox for providing invaluable advice and inspiration (and ofcourse a testbed cluster to try things on). Thank you to DongInn Kim and Greg Kurtzer for their guidance, continued support and motivation.

References

[1] Infiscale. `http://www.infiscale.com`.

[2] Oscar developer documentation for the latest upcoming major release oscar 5.1. `http://svn.oscar.openclustergroup.org/trac/oscar/wiki/DevelDocs`.

[3] The Beowulf HOWTO. `http://tldp.org/HOWTO/html_single/Beowulf-HOWTO/`.

[4] Accelerating innovation for competitive advantage: The need for better hpc application software solutions. Technical Report High Performance Computing Software Workshop Report, The Council on Competitiveness, July 2005.

[5] Baris Guler and Munira Hussain and Tau Leng Ph.D. and Victor Mashayekhi Ph.D. The advantages of diskless hpc clusters using nas. Technical Report Dell Power Solutions, Dell, November 2002.

[6] W. Bland, T. Naughton, G. Vallee, and S. L. Scott. Design and implementation of a menu based oscar command line interface. In *HPCS '07: Proceedings of the 21st International Symposium on High Performance Computing Systems and Applications*, page 25, Washington, DC, USA, 2007. IEEE Computer Society.

[7] B. des Ligneris, S. Scott, T. Naughton, and N. Gorsuch. Open Source Cluster Application Resources (OSCAR): design, implementation and interest for the [computer] scientific community.

[8] Infiscale. Perceus user guide, November 2007.

[9] G. Jung. The LBNL Perceus Cluster Infrastructure. Internet2 Fall Conference, October 2007. Next Generation Cluster Provisioning and Management.

[10] D. Kim, J. M. Squyres, and A. Lumsdaine. Revamping the OSCAR database: A flexible approach to cluster configuration data management. In I. Kotsireas and D. Stacey, editors, *19th International Symposium on High Performance Computing Systems and Applications*, pages 326–332, Guelph, Ontario, Canada, May 2005. IEEE Computer Society.

[11] D. Kim, J. M. Squyres, and A. Lumsdaine. The Introduction of the OSCAR Database API (ODA). In *Proceedings of the 20th International Symposium on High-Performance Computing in an Advanced Collaborative Environment (HPCS'06)*, page 39. IEEE Computer Society, May 14-17 2006. Session track: 4th Annual OSCAR Symposium (OSCAR'06).

[12] G. M. Kurtzer. Warewulf: The Cluster Node Management Solution. SuperComputing 2003. Lawrence Berkeley National Laboratory.

[13] B. Ligneris and F. Giraldeau. Thin-oscar : Design and future implementation, 2003.

[14] J. Mugler, T. Naughton, and S. Scott. Oscar meta-package system. *High Performance Computing Systems and Applications, 2005. HPCS 2005. 19th International Symposium on*, pages 353–360, 15-18 May 2005.

[15] J. Mugler, T. Naughton, S. L. Scott, B. Barrett, A. Lumsdaine, J. M. Squyres, Benoit des Ligneris, Francis Giraldeau, and C. Leangsuksun. OSCAR Clusters. In *Proceedings of the Ottawa Linux Symposium (OLS'03)*, Ottawa, Canada, July 23-26 2003.

[16] Panasas. Los Alamos National Lab Diskless Cluster deploys Panasas parallel storage, April 2005. Case Study.

[17] J. Parpaillon. Oscar kernelpicker: Handling clients kernels. *High Performance Computing Systems and Applications, 2007. HPCS 2007. 21st International Symposium on*, pages 27–27, May 2007.