

# HA-OSCAR: Highly Available Linux Cluster

Latia Laura Shumpert  
Fayetteville State University  
shumpertll@ornl.gov

Research Alliance in Mathematics and Science

Mentors:

Dr. Stephen L. Scott

Dr. Daniel Okunbor

Mr. John Mugler

Mr. Thomas Naughton

Computer Science and Mathematics Division

Network and Cluster Computing

**August 11, 2005**

# Table of Contents

- High Performance Cluster Computing (Beowulf)
- OSCAR (Open Source Cluster Application Resources)
- HA (high-availability)
- HA-OSCAR Architecture



# Who Is Interested in Clusters & HA

## Clusters

- High Performance Computing
- Low cost “Supercomputing” for the commoner
- Reasonable scalability potential

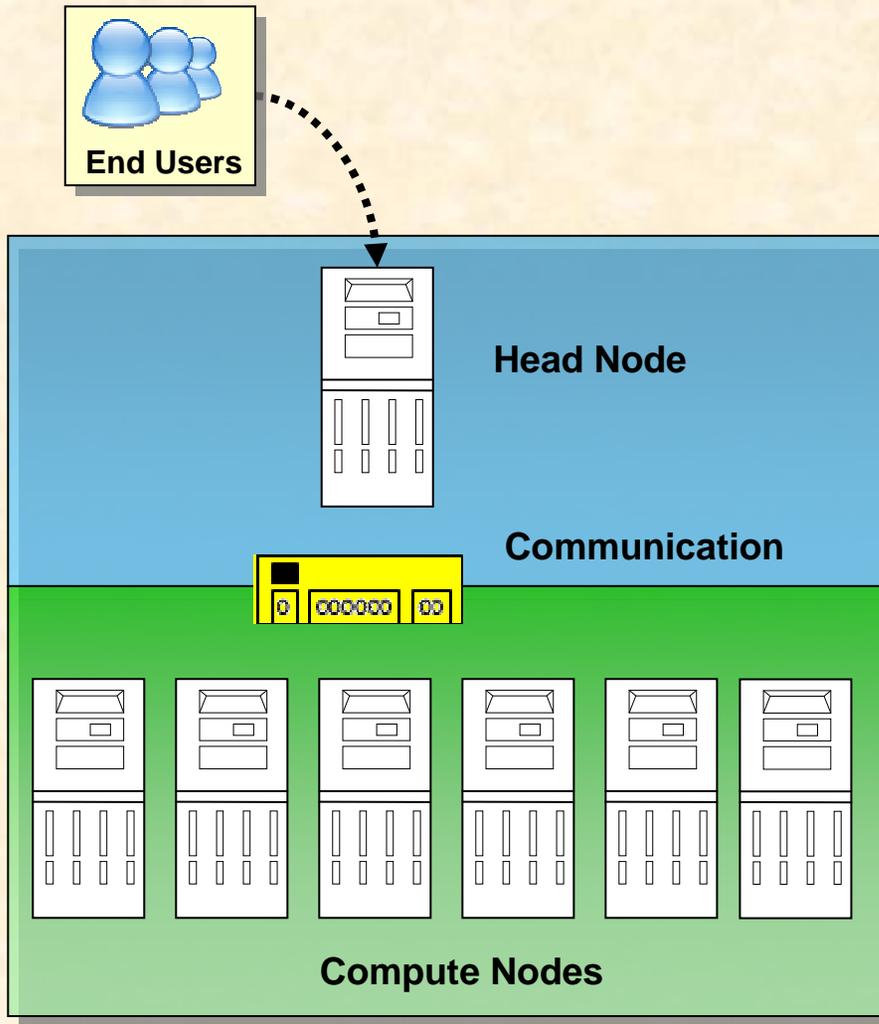
## High Availability

- HPC (many parts HW/SW - which fail)
- Telco
- Power Plants
- Web Server Farms
- Paid for continuous(non-stop) computer services

# Beowulf Cluster

- Beowulf was one approach to clustering Common Off The Shelf (COTS) components to form a high performance computer
- Beowulf cluster is a collection of COTS computers networked together to harvest high performance computing
- Typical Beowulf cluster has:
  - a single head node
  - multiple identical client nodes

# Beowulf Cluster



## HeadNode

- ✓ Entry point to the cluster
- ✓ Responsible for serving user requests
- ✓ Distributes jobs to compute clients via scheduling and queuing software

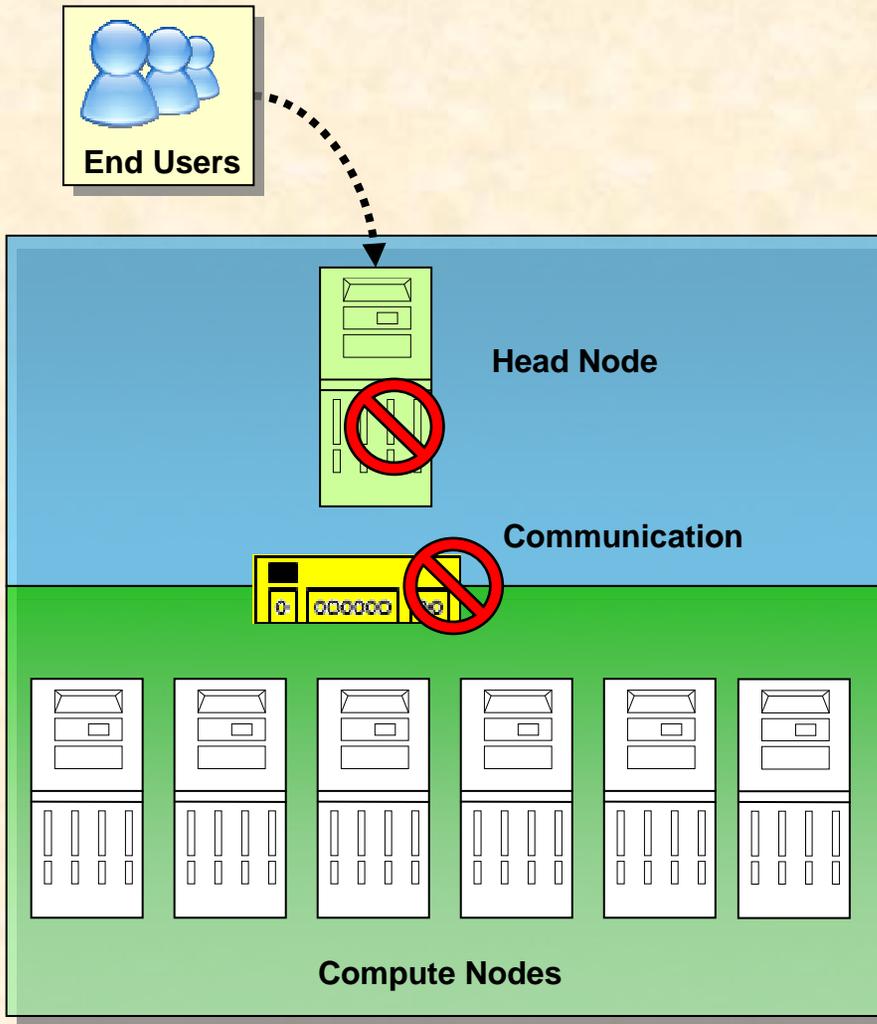
## Communication

Using Ethernet network and/or fast connectivity: Myrinet, Infiniband, etc.

## Compute Clients

- ✓ Dedicated for computation

# Beowulf Cluster – Issues



- Single head node architecture
  - Vulnerable for Single Point of Failure (SPOF)
- Single communication path architecture
  - Vulnerable for SPOF
- Compute nodes are not accessible after above threat occurs, or when cluster services or OS upgrade takes place

# OSCAR

## (Open Source Cluster Application Resources)



# Open Source Cluster Application Resources

## What is OSCAR?



- Framework for cluster installation configuration and management
- Common used cluster tools
- Wizard based cluster software installation
  - Operating system
  - Cluster environment
    - Administration
    - Operation
- Automatically configures cluster components

Step 8 Done!

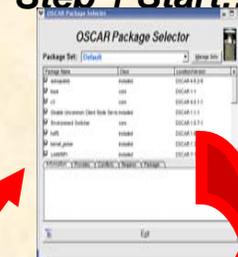
Step 1 Start...



Complete Cluster Setup

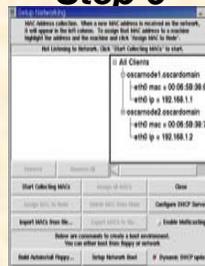


Test Cluster Setup



Select packages to install

Step 6



Setup Networking



OSCAR Wizard



Configure Selected OSCAR packages

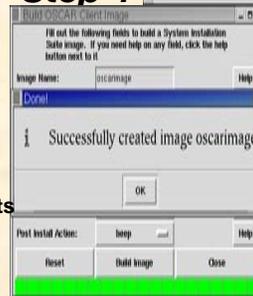
Step 2

Step 5



Define OSCAR Clients

Step 4



Build OSCAR Client Image



Install OSCAR Server packages

# HA (High Availability)



# What is HA Clustering?

- **High Availability**

- Enhanced the uptime of computer-based communications systems
- Isolates or reduces the impact of a failure in the machine, resources, or device through redundancy and fail over techniques.

- **Goal with HA-Clusters was to ensure service availability**

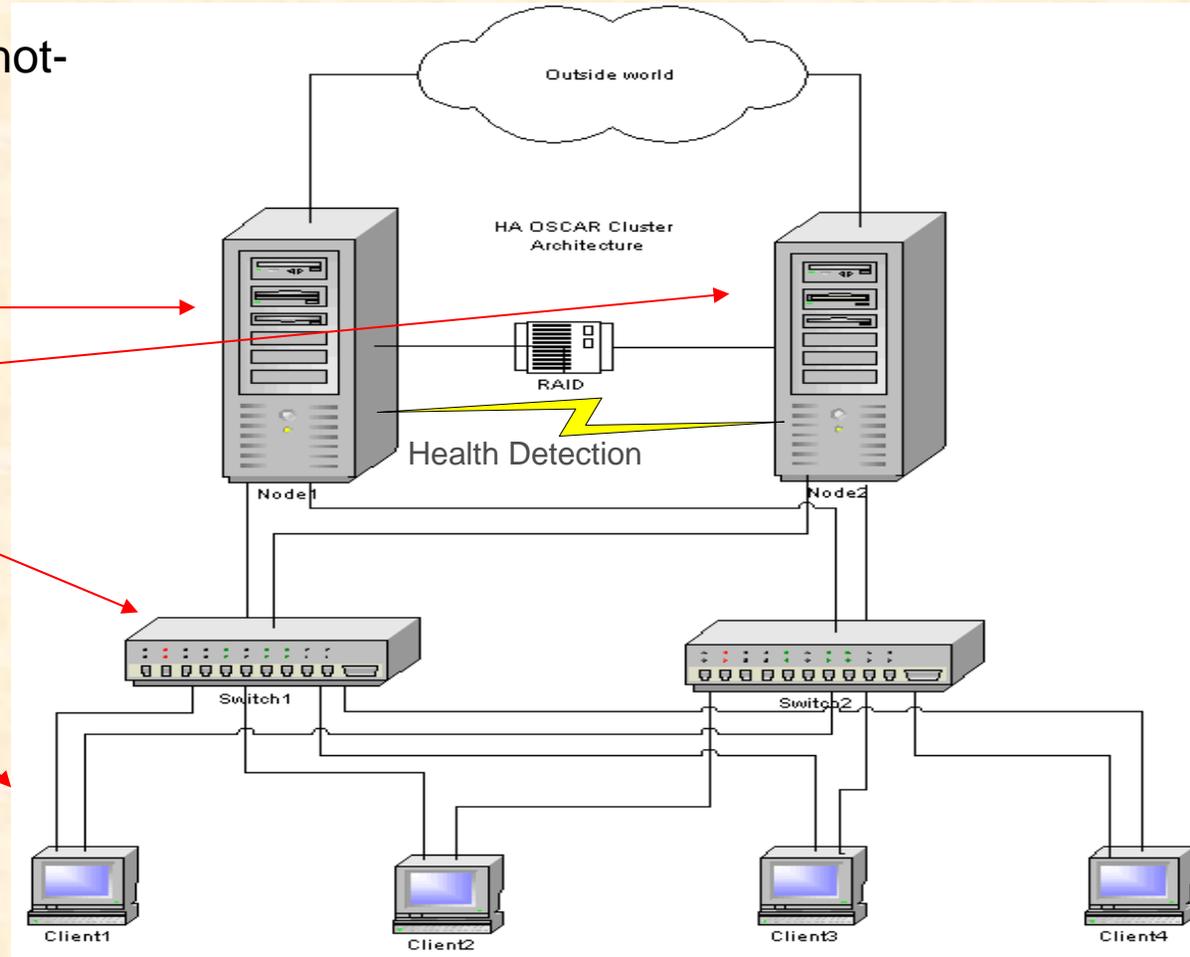
- Ability to continue serving clients even if one (or more) server node fails and becomes unavailable

# Providing High Availability

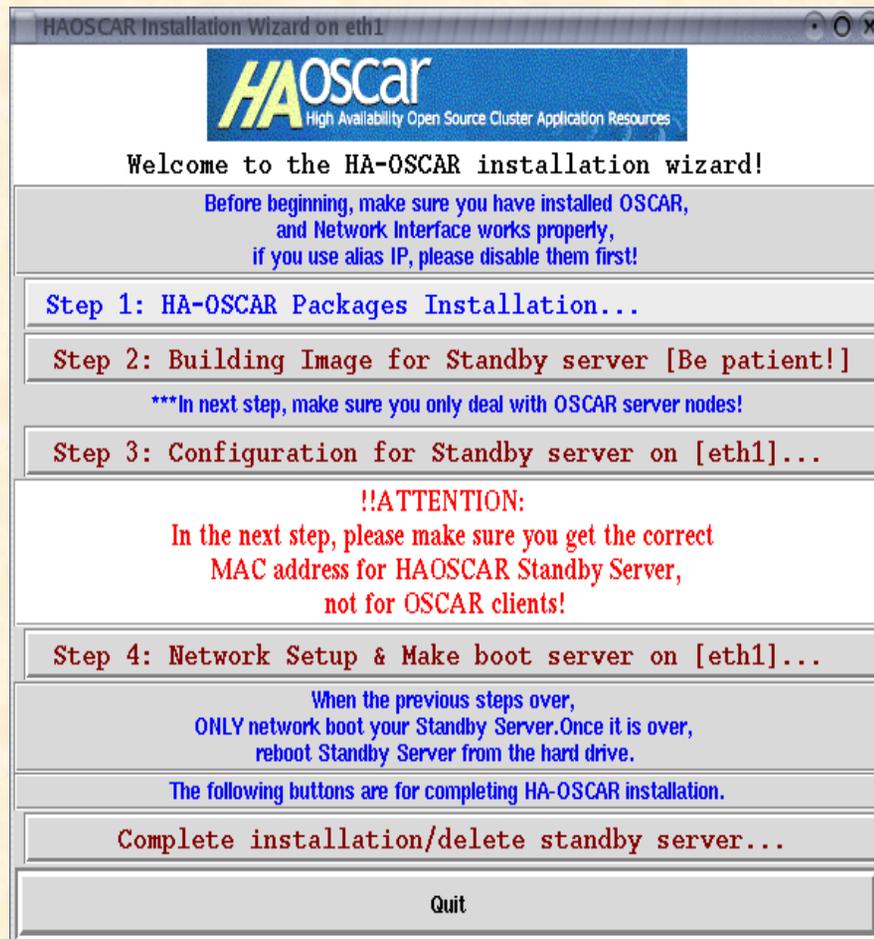
- **Complete HA solution requires close integration**
  - HA hardware
  - HA software solution
  - HA middleware
  - Application software that can cause failover to redundant systems
- **Other requirements**
  - Hot swap (hot insert, hot remove, identity maintenance)
  - Support diskless operation, ...
  - Options for booting compressed, remotely hosted kernel images
  - Support of compressed r/w and read-only Flash file systems
  - Accelerated boot and daemon start times
  - Fast shutdown / reboot
  - Eliminating costly file system operations with journaling file systems

# HA-OSCAR Architecture

- Version 1.1 was an active/hot-standby architecture with automatic failover
- Major components
  - Primary server
  - Standby server
  - Switches
  - Multiple clients



## Four steps to install HA-OSCAR



# Installation Walkthrough

2/5

1. Install server packages to build an HA-OSCAR base
2. Launches a fetch Image wizard by which primaryserver image is grabbed and stored on primaryserver.
  1. User can accept defaults values in this window
  2. Finally user clicks *Fetch Image* button and image is fetched

Fetch a System Installation Suite Image

Fill out the following fields to fetch a System Installation Suite image. If you need help on any field, click the help button next to it

Image Name:	<input type="text" value="serverimage"/>	Help
Client Name:	<input type="text" value="hapcoscar"/>	Help
SSH User Name:	<input type="text"/>	Help
IP Assignment Method:	<input type="text" value="static"/>	Help
Post Install Action:	<input type="text" value="beep"/>	Help
Reset		Fetch Image
		Close

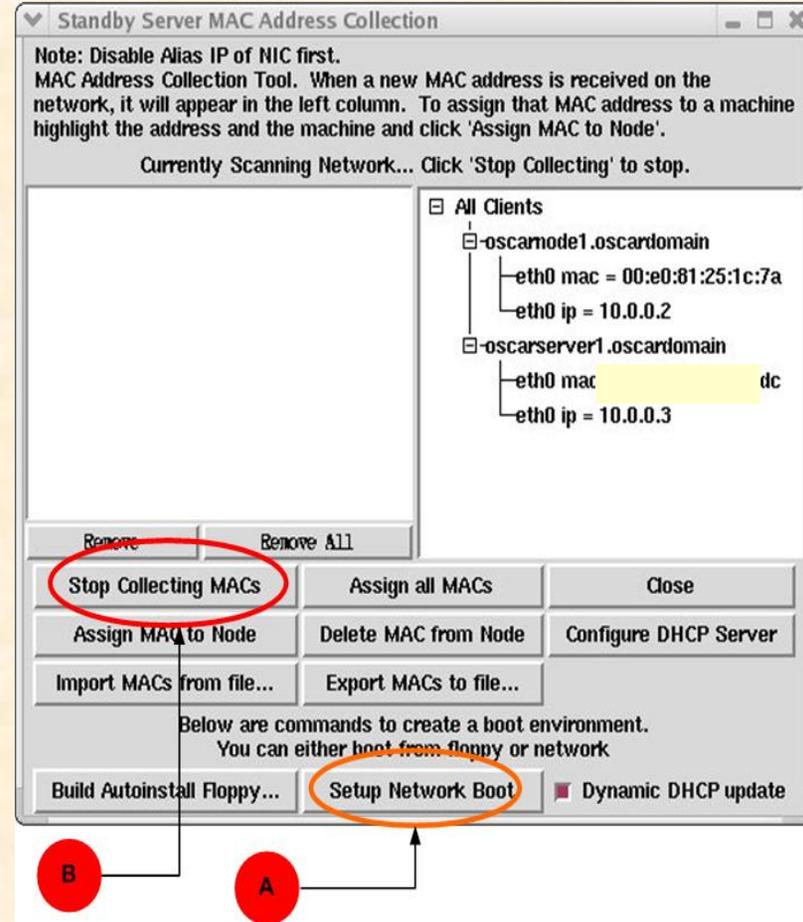
3. Next step involves configuration of standby server
- Image name from the previous step (*Serverimage*) is selected to install on Standbyserver
  - Standbyserver's local IP, public alias IP and gateway can be changed according to there network address
  - After entering all the fields, next, click on *AddStandby Server* button

Field	Value	Action
Image Name:	serverimage	Help
Primary server's Public Network Interface:	eth1	Help
Standby server's Public Network Interface:	eth1	Help
Standby server's Public Alias IP:	138.47.21.199	Help
Standby server's local IP:	192.168.3.200	Help
Subnet Mask:	255.255.255.0	Help
Default Gateway:	192.168.3.1	Help
<b>AddStandby Server</b>		Close

# Installation Walkthrough

4/5

4. Network setup (for PXE boot) to transfer the clone image on Primaryserver to remote Standbyserver
  - First click on *Setup Network Boot (A)*.
  - Configure Standbyserver boot sequence to network boot and reboot the Standbyserver.
  - Next *Collect MAC Address (B)* of Standbyserver.

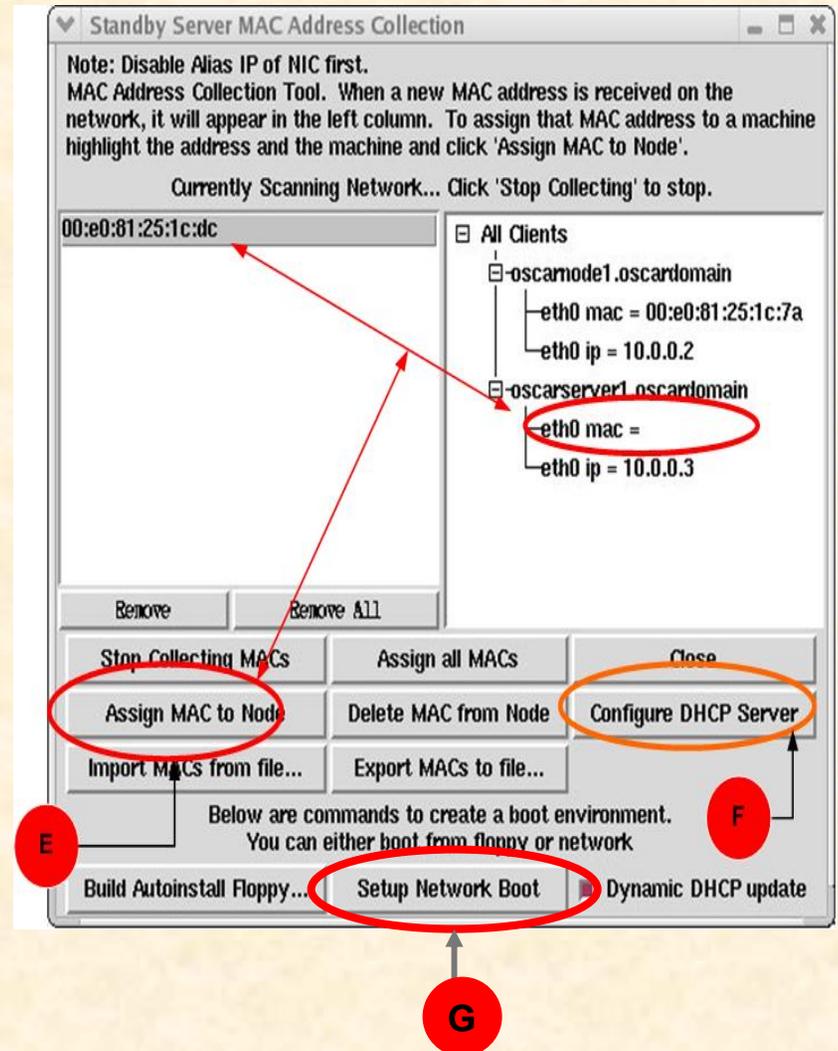


Note: For *Build Autoinstall Floppy* method refer to appendix 1

# Installation Walkthrough

5/5

- After MAC address is collected, it will be associated to IP address (from previous step) of Standbyserver by clicking on *Assign MAC to Node* (E).
- Then *Configure DHCP Server* (F) on primary node to assign IP address to Standbyserver.
- *Setup Network Boot* (G) is booted as PXE boot.
- Once the Standbyserver is up, final step *complete installation* finishes the HA-OSCAR setup.



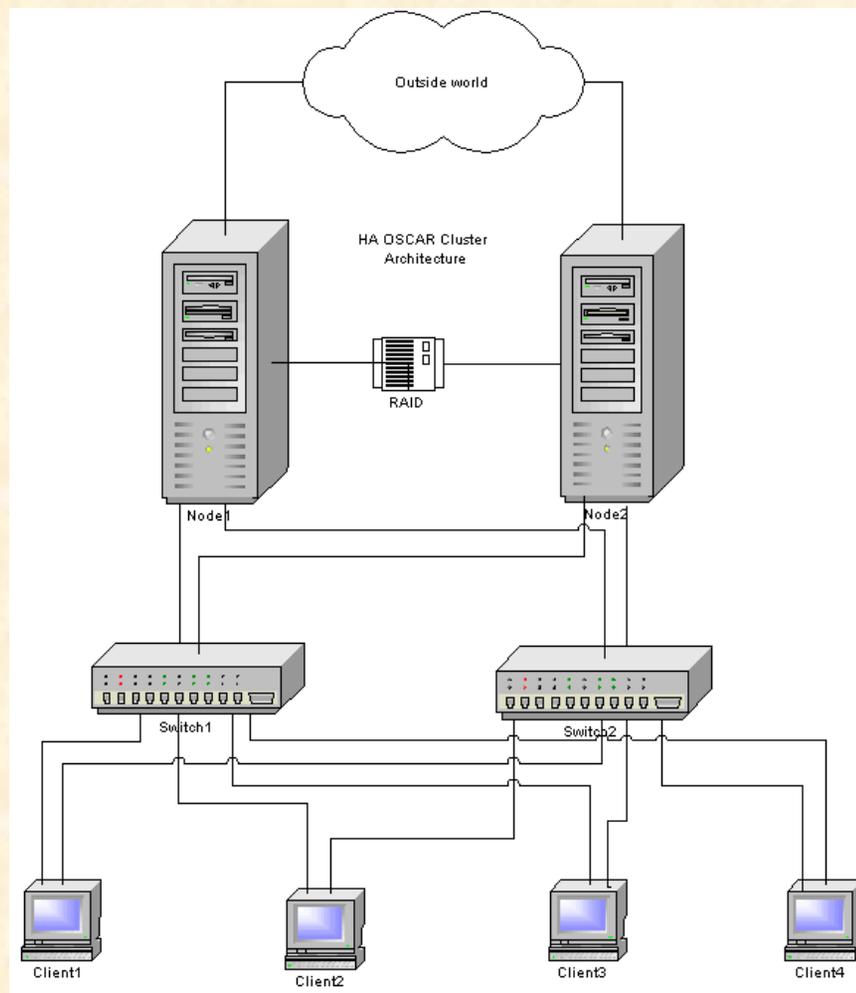
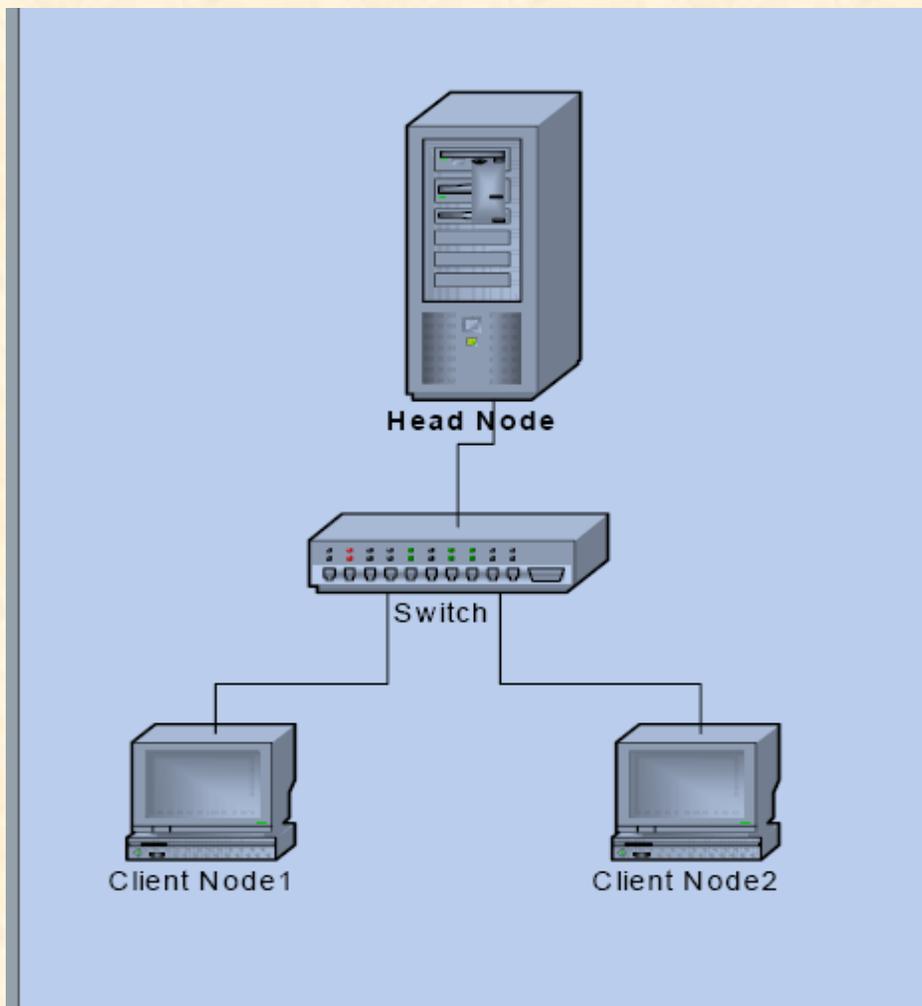
# Accomplished Goals for OSCAR

- Installed Linux on server machine (cluster head node)
  - workstation install w/ software development tools
  - 50+ page installation document!
    - (quick install available)
- Downloaded copy of OSCAR and unpack on server
- Configured and install OSCAR on server
  - readies the wizard install process
- Configured server Ethernet adapters
  - public
  - private
- Launched OSCAR Installer (wizard)

# Accomplished Goals for HA-OSCAR

- Downloaded copy of HA-OSCAR and unpack on server <http://xcr.cenit.latech.edu/ha-oscar>
- Extract the tar-file
- Launched HA-OSCAR Installer (wizard)

# OSCAR & HA-OSCAR Setups



# Resources

HA-OSCAR

[xcr.cenit.latech.edu/ha-oscar](http://xcr.cenit.latech.edu/ha-oscar)

OSCAR

[www.OSCAR.OpenClusterGroup.org](http://www.OSCAR.OpenClusterGroup.org)

Open Cluster Group

[www.OpenClusterGroup.org](http://www.OpenClusterGroup.org)

# Acknowledgments

- Louisiana Tech University
  - Chokchai “Box” Leangsuksun
- Oak Ridge National Laboratory
  - Stephen L. Scott
  - Thomas Naughton
  - John Mugler
- Open Source Development Labs
  - Ibrahim Haddad
- OSCAR
  - The entire OSCAR team, collaborators, and users.



# Results

**Successful OSCAR installation**

**Successful HA-OSCAR Installation**

## Special Thanks

Mathematical, Information, and Computational  
Sciences Division, Office of Advanced Scientific  
Computing Research, U.S. Department of Energy

**Question or Comments**