

ENABLING HIGHLY AVAILABLE GRID SITES

Michael Bryant
Louisiana Tech University

Grid Computing and High Availability

What is High Availability?



Current State of High Availability in Grid Computing



Site-level High Availability of Grid Resources and Services

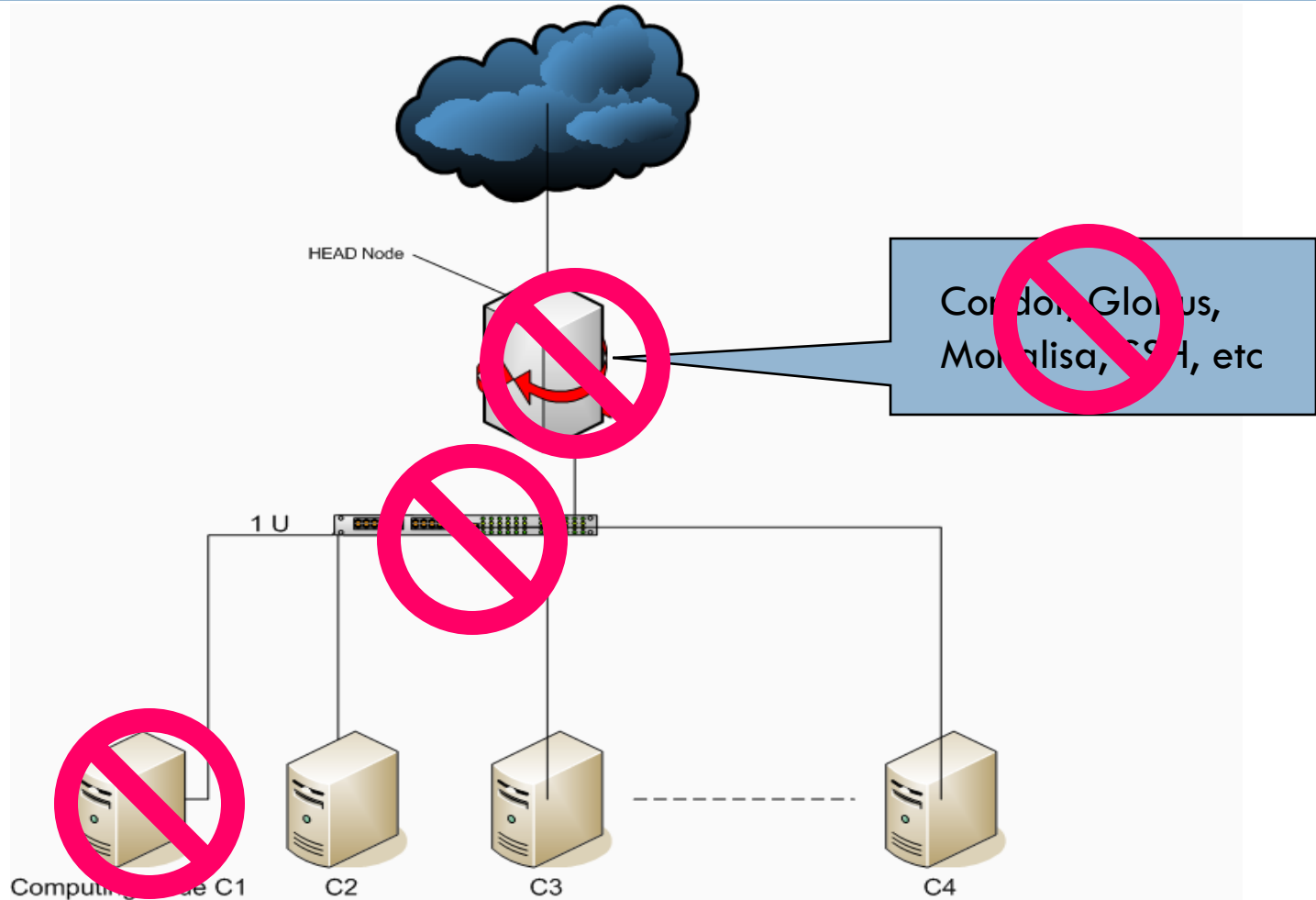
What exactly is High Availability?

High Availability is a system design protocol and associated implementation that ensures a certain absolute degree of operational continuity during a given measurement period.

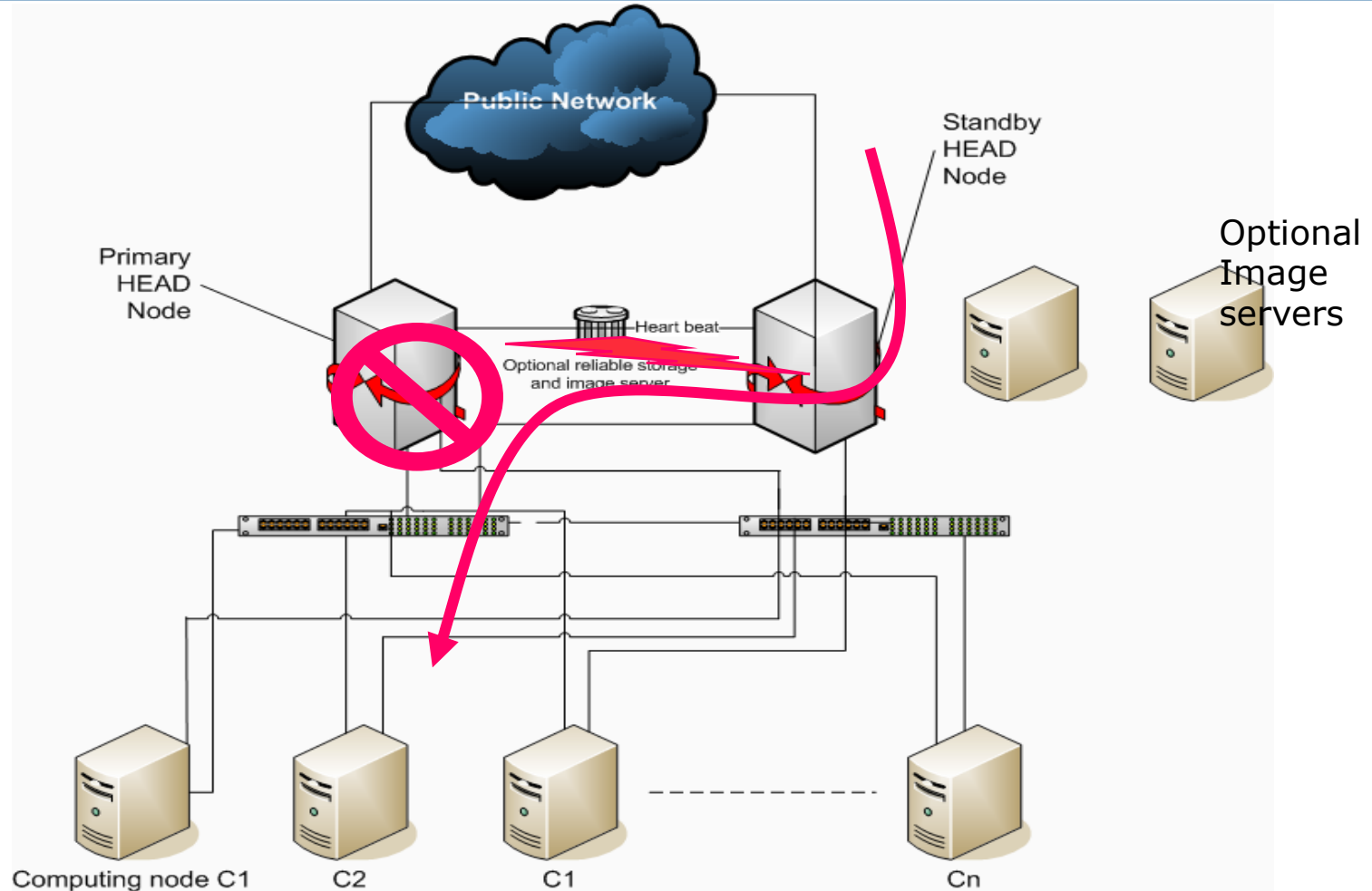
Simply put, high availability means increasing the **availability** of resources and services that a system provides.

In order to increase availability, you must first eliminate single points of failure...

Single Points of Failure: Single headnode scenario



Single Points of Failure: Dual headnode scenario



Current State of High Availability in Grid Computing

Where are HA solutions used and why not in grids?

Commonplace

- Clusters
 - ▣ Many commercial solutions
 - ▣ Few open source solutions
- Servers
 - ▣ Web/mail/storage/etc
 - ▣ Downtime is costly
- Mission-critical applications
 - ▣ HA more important with increased usage of Linux
 - ▣ Decreases cost

HA in Grids

- Most grid sites are not HA, but many do use some kind of fault-tolerant techniques (RAID, SAN)
- The complexity of grids slow adoption of HA
- Common HA techniques doesn't translate well to grids
 - ▣ Site-to-site failovers are problematic
 - Each site is different (hardware, networks, policies, etc...)
 - ▣ Grids connect over WANs
 - Short network timeouts possible (triggering false replication of services)
 - IP failover techniques don't work (different subnets)

So, are highly available grids even possible?

Most certainly.

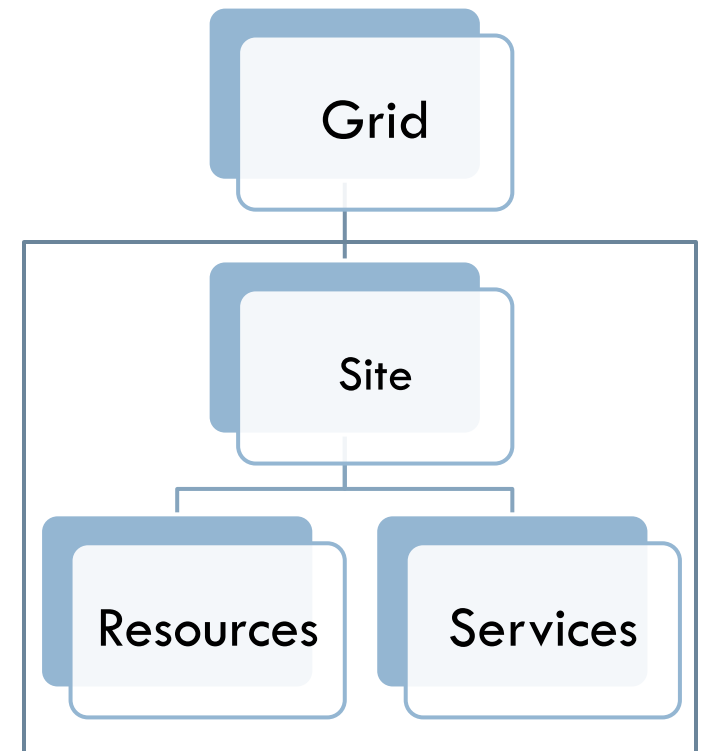
Site-level High Availability of Grid Resources and Services

Highly Available Grid Resources and Services

- Using component redundancy and self-healing capabilities, we can eliminate single points of failure at grid sites.
 - ▣ A site refers to an organization, such as an institution or corporation, that lends its computing **resources** to a grid.
 - ▣ These resources are accessible through **grid services** setup at the site.
- By eliminating single points of failure, we can make the grid resources and services of a particular site highly available.

What is site- vs grid-level HA?

- **Grid-level high availability**
 - ▣ Site-to-site failovers (Linux-HA claims it can in v2.0.8)
 - ▣ Failover of idle/running jobs to designated grid sites
 - ▣ Allowing for new jobs to be submitted to other sites (fault-tolerant grid resource selector)
- **Site-level high availability**
 - ▣ Removes single points of failure on cluster-based grid sites
 - ▣ Deals directly with the site's resources and services making them highly available



Thus, if grid sites are made highly available, it is then possible to have a grid that is highly available.

High Availability Techniques

Mon and Net-SNMP



Linux-HA



Application HA (Condor, PBS)

HA Technique: Mon and Net-SNMP

What is Mon and Net-SNMP?

- Underneath, HA-OSCAR uses two main components:
 - ▣ *mon* – “is a general-purpose scheduler and alert management tool used for monitoring service availability and triggering alerts upon failure detection.” (<http://www.kernel.org/software/mon/>)
 - ▣ *Net-SNMP* – “is a widely used protocol for monitoring the health and welfare of network equipment (eg. routers), computer equipment and even devices like UPSs.” (<http://net-snmp.sourceforge.net>)

How Mon and Net-SNMP work together...

- On standby headnode:
 - ▣ Mon uses fping every 15 sec to check if primary is down
 - Failover: If primary is down, mon calls a set of alert scripts that are used to confirm and act on the failure
 - The IP address for the standby changes to the primary's IP
 - All services running on primary are started on standby
 - PBS/Torque queue is transferred over using rsync and jobs are restarted from last checkpoint
 - Failback: When primary is back online, mon reverses process
- On primary headnode:
 - ▣ Mon uses Net-SNMP to check if a service is running
 - Mon will automatically restart any service that is not running

HA Technique: Linux-HA

What is Linux-HA?

- A high availability UNIX package called heartbeat which consists of many HA components:
 - ▣ Heartbeat program
 - ▣ Local resource manager
 - ▣ Cluster resource manager
 - ▣ Cluster information base
 - ▣ Stonith daemon (restarts failed node)
- Works similarly to Mon/Net-SNMP except it has more features and is more widely known.
- Comparable to most commercial HA solutions in terms of easy-of-use and features.

- Researching how it compares to the Mon/Net-SNMP technique

HA Technique: Application HA (Condor, PBS)

Highly Available Condor Daemons

- Condor has built-in HA mechanisms (*condor_had* daemon)
- Condor can failover/back both the central manager daemons (*condor_negotiator* and *condor_collector*) and the job queue (*condor_schedd*).
 - Central manager failover:
 - The CM daemons are the heart of the Condor matchmaking system and is critical to the pool's functionality.
 - Can handle network partitioning
 - Note: does **not** work with flocking!
 - Job queue failover:
 - Jobs caused to stop without finishing can be restarted from the beginning, or can continue execution using the most recent checkpoint.
 - New jobs can enter the job queue.
 - Uses a shared file system to keep a lock file to prevent multiple *condor_schedd* daemon from running at the same time.
- Researching possibility of using in the OSG

PBS/Torque Job Failover

- Developed here at Tech by graduate students
- In progress of polishing the package to include in HA-OSCAR
- Job queue failover:
 - ▣ PBS job queue is rsync'd from the primary to the standby headnode periodically – keeping both idle and running jobs.
 - ▣ Standby's jobs are put in the held state, while primary's jobs remains running.
 - ▣ When primary fails, jobs on standby are changed to a running state.
- Job checkpointing and failover:
 - ▣ Jobs are checkpointed on worker nodes and rsync'd to a file server.
 - ▣ If a worker node fails, the running jobs on that node are started on a spare node from the last checkpoint. Or restarted if no checkpoint.
 - ▣ Uses the BLCR (Berkeley Lab Checkpoint/Restart) checkpointing software

High Availability Solutions

HA-OSCAR

HA-ROCKS

Creating a grid-centric HA OSG package

HA Solution: HA-OSCAR

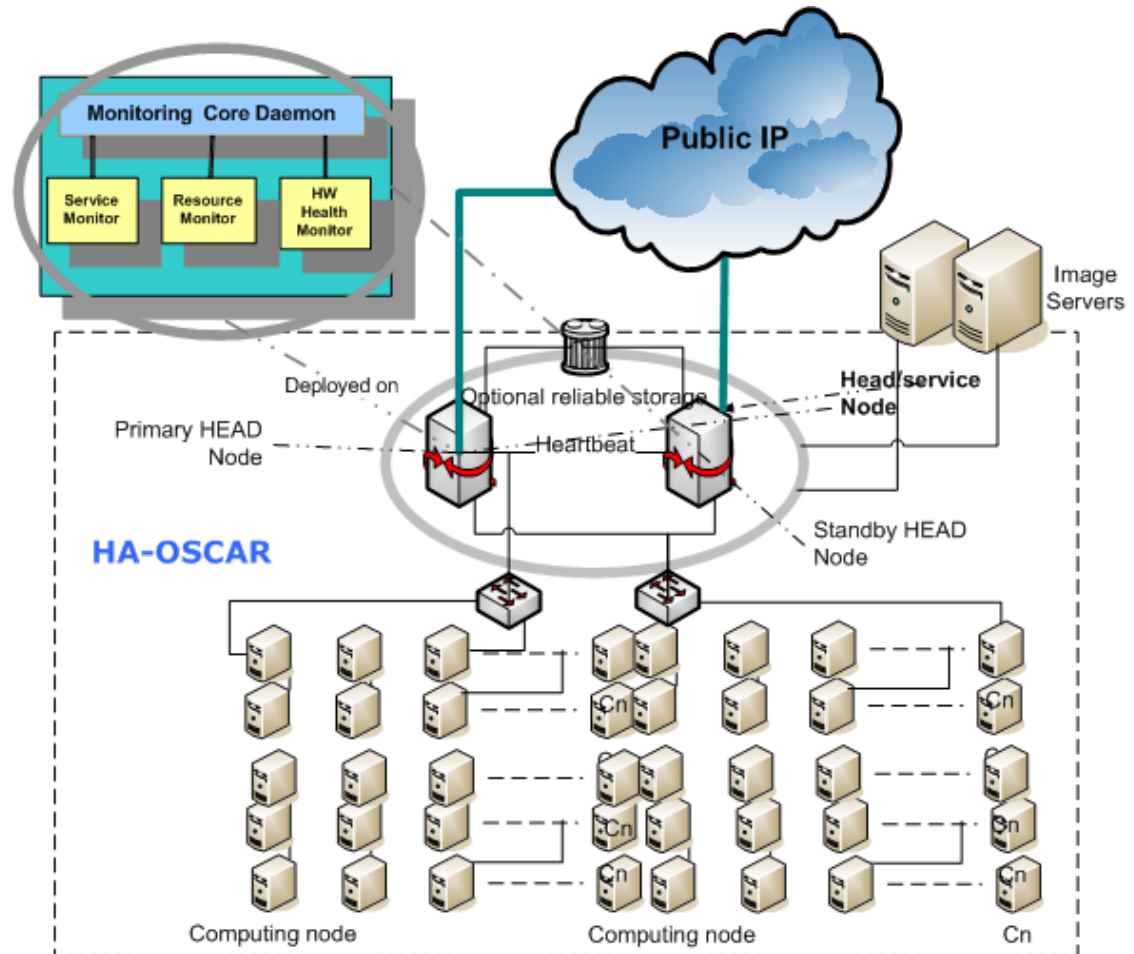
HA-enabled OSCAR Clusters

- ❑ Built on top of OSCAR (Open Source Cluster Application Resources), which is an open source cluster management tool (similar to ROCKS)
- ❑ Developed by Dr. Box and his students
- ❑ Provides an easy to install and setup open source HA solution
- ❑ OSCAR installs on RHEL, Fedora, SUSE, and others
- ❑ Component redundancy is adopted to eliminate single points of failure
- ❑ HA-OSCAR incorporates a self-healing mechanism; failure detection & recovery, automatic failover and fail-back
- ❑ Stable release 1.2 works with OSCAR 5.0

HA-OSCAR (continued)

Self-healing
with 5-20 sec
automatic
failover time

The first
known field-
grade open
source HA
Beowulf
cluster release



HA Solution: HA-ROCKS

HA-enabled ROCKS Clusters

- HA-ROCKS was in development by Dr. Box and students, but focused shifted back to OSCAR because of lack of interest.
- Uses SIS, the system installation suite, for cloning the primary headnode. This is what OSCAR uses, too.
- Plans to start back developing on when time permits

Creating a grid-centric HA OSG package

Bringing HA to all OSG sites

- Our first step to making a truly highly available grid is to develop a HA solution for the OSG, which will contain:
 - ▣ An easy to use installer to build a HA cluster
 - ▣ Standard HA package that can be installed on any Linux distribution (ROCKS, RHEL, Debian, SUSE, etc)
 - ▣ All HA functionality included in HA-OSCAR
 - ▣ Documentation on how to setup a HA cluster using OSCAR, ROCKS, RH Cluster Manager, etc.
- This HA solution will not depend on a cluster suite, so building highly available OSG headnodes without a cluster, such as LTU_CCT, will be possible.

In closing...

Conclusion



Questions/Comments?



Links

Conclusion

- Highly available grids can be enabled by reducing single points of failure at grid sites.
- At the same time, eliminating a lot of the single point failures will make grids more reliable.
- There are already existing open source HA tools, such as HA-OSCAR, that target mission critical applications.
- In the coming months, expect to see Louisiana Tech playing a bigger role in High Availability and Grid Computing.

Questions/Comments?

Links

- HA-OSCAR - <http://xcr.cenit.latech.edu/ha-oscar/index.html>
- Linux-HA - <http://www.linux-ha.org>
- Condor-HAD -
http://www.cs.wisc.edu/condor/manual/v6.8/3_11High_Availability.html
- Mon - <http://www.kernel.org/software/mon/>
- Net-SNMP - <http://net-snmp.sourceforge.net>