Asymmetric Active-Active High Availability for High-end Computing [*] [†]

C. Leangsuksun
V. K. Munganuru
Computer Science Department
Louisiana Tech University
P.O. Box 10348, Ruston, LA 71272, USA
Phone: +1 318 257-3291
Fax: +1 318 257-4922
box@latech.edu
vkm001@latech.edu

T. Liu
Dell Inc.
Tong_Liu@dell.com

S. L. Scott
C. Engelmann
Computer Science and Mathematics Division
Oak Ridge National Laboratory
scottsl@ornl.gov
engelmannc@ornl.gov

ABSTRACT

Linux clusters have become very popular for scientific computing at research institutions world-wide, because they can be easily deployed at a fairly low cost. However, the most pressing issues of today‘s cluster solutions are availability and serviceability. The conventional Beowulf cluster architecture has a single head node connected to a group of compute nodes. This head node is a typical single point of failure and control, which severely limits availability and serviceability by effectively cutting off healthy compute nodes from the outside world upon overload or failure. In this paper, we describe a paradigm that addresses this issue using asymmetric active-active high availability. Our framework comprises of n + 1 head nodes, where n head nodes are active in the sense that they provide services to simultaneously incoming user requests. One standby server monitors all active servers and performs a fail-over in case of a detected outage. We present a prototype implementation based on a 2 + 1 solution and discuss initial results.

Keywords

Scientific computing, clusters, high availability, asymmetric active-active, hot-standby, fail-over

[*]This work was supported by the U.S. Department of Energy under Contract No. contract No. DE-FG02-05ER25659.

[†]This research is sponsored by the Mathematical, Information, and Computational Sciences Division; Office of Advanced Scientific Computing Research; U.S. Department of Energy. The work was performed at the Oak Ridge National Laboratory, which is managed by UT-Battelle, LLC under Contract No. De-AC05-00OR22725.

1. INTRODUCTION

With their competitive price/performance ratio, COTS computing solutions have become a serious challenge to traditional MPP-style supercomputers. Today, Beowulf-type clusters are being used to drive the race for scientific discovery at research institutions and universities around the world. Clusters are popular, because they can be easily deployed at a fairly low cost. Furthermore, cluster management systems (CMSs) like, OSCAR [6, 18] and ROCKS [21], allow uncomplicated system installation and management without the need to individually configure each component separately. HPC cluster computing is undoubtedly an eminent stepping-stone for future ultra-scale high-end computing systems.

However, the most pressing issues of today‘s cluster architectures are availability and serviceability. The single head node architecture of Beowulf-type systems itself is an origin for these predicaments. Because of the single head node setup, clusters are vulnerable, as the head node represents a single point of failure affecting availability of its services. Furthermore, the head node represents also a single point of control. This severely limits access to healthy compute nodes in case of a head node failure. In fact, the entire system is inaccessible as long as the head node is down. Moreover, a single head node also impacts system throughput performance as it becomes a bottleneck. Overloading the head node can become a serious issue for high throughput oriented systems.

To a large extent, the single point of failure issue is addressed by active/hot-standby turnkey tools like HA-OSCAR [4, 5, 10], which minimize unscheduled downtimes due to head node outages. However, sustaining throughput at large scale remains an issue due to the fact that active/hot-standby solutions still run one active head node only. In this paper, we describe a paradigm that addresses this issue using *asymmetric active-active high availability*. Unlike typical Beowulf architectures (with or without hot-standby node(s)), our framework comprises of $n + 1$ servers. n head nodes are active in the sense that they provide services to simultaneously incoming user requests. One hot-standby server monitors all the active servers and performs a fail-over in case of a detected outage.

Since our research also focuses on the batch job scheduler that typically runs on the head node, our proposed high availability architecture effectively transforms a single scheduler system into a cluster running multiple schedulers in parallel without maintaining global knowledge. These schedulers run their jobs on the same compute nodes or on individual partitions, and fail-over to a hot-standby server in case of a single server outage.

Sharing compute nodes among multiple schedulers without coordination is a very uncommon practice in cluster computing, but has its value for high throughput computing.

Furthermore, our solution leads the path towards the more versatile paradigm of *symmetric active-active high availability* for high-end scientific computing using the *virtual synchrony* model for head node services [3]. In this model, the same service is provided by all head nodes using group communication at the back-end for coordination. Head nodes may be added, removed or fail at any time, while no processing power is wasted by keeping an idle backup server and no service interruption occurs during recoveries. Symmetric active-active high availability is an ongoing research effort and our asymmetric solution will help to understand the concept of running the same service on multiple nodes and the necessary coordination.

However, in contrast to the symmetric active-active high availability paradigm with its consistent symmetric replication of global knowledge among all participating head nodes, our asymmetric paradigm maintains only backups on one standby head node for all active head nodes.

This paper is organized as follows: First, we provide a review of related past and ongoing research activities. Second, we describe our asymmetric active-active high availability solution in more detail and show how the system handles multiple jobs simultaneously while enforcing high availability. Third, we present some initial results from our prototype implementation. Finally, we conclude with a short summary of the presented research and a brief description of future work.

2. RELATED WORK

Related past and ongoing research activities include cluster management systems as well as active/hot-standby cluster high availability solutions.

Cluster management systems allow uncomplicated system installation and management, thus improving availability and serviceability by reducing scheduled downtimes for system management. Examples are OSCAR and Rocks.

The Open Source Cluster Application Resources (OSCAR [6, 18]) toolkit is a turnkey option for building and maintaining a high performance computing cluster. OSCAR is a fully integrated software bundle, which includes all components that are needed to build, maintain, and manage a medium-sized Beowulf cluster. OSCAR was developed by the Open Cluster Group, a collaboration of major research institutions and technology companies led by Oak Ridge National Laboratory (ORNL), the National Center for Supercomputing Applications (NCSA), IBM, Indiana University, Intel, and Louisiana Tech University. OSCAR has significantly reduced the complexity of building and managing a Beowulf cluster by using a user-friendly graphical installation wizard as front-end and by providing necessary management tools at the back-end.

Similar to OSCAR, NPACI Rocks [21] is a complete "cluster on a CD" solution for x86 and IA64 Red Hat Linux COTS clusters. Building a Rocks cluster does not require any experience in clustering, yet a cluster architect will find a flexible and programmatic way to redesign the entire software stack just below the surface (appropriately hidden from the majority of users). The NPACI Rocks toolkit was designed by the National Partnership for Advanced Computational Infrastructure (NPACI). The NPACI facilitates collaboration between universities and research institutions to build cutting-edge computational environments for future scientific research. The organization is led by the University of California (UCSD), San Diego, and the San Diego Supercomputer Center (SDSC).

Numerous ongoing high availability computing projects, such as LifeKeeper [11], Kimberlite [8], Linux Failsafe [12] and Mission Critical Linux [16], focus their research on solutions for clusters. However, they do not reflect the Beowulf cluster architecture model and fail to provide availability and serviceability support for scientific computing, such as a highly available job scheduler. Most solutions provide highly available business services, such as data storage and data bases. They use "cluster of servers" to provide high availability locally and enterprise-grade wide-area disaster recovery solutions with geographically distributed server cluster farms.

HA-OSCAR tries to bridge the gap between scientific cluster and traditional high availability computing. High Availability Open Source Cluster Application Resources (HA-OSCAR [4, 5, 10]) is production-quality clustering software that aims toward non-stop services for Linux HPC environments. In contrast to previously discussed HA applications, HA-OSCAR strategies combine both the high availability and performance aspects making its methodology and infrastructure to be the first field grade HA Beowulf cluster solution that provides high availability, critical failure prediction and analysis capability. The project's main objectives focus on Reliability, Availability and Serviceability (RAS) for the HPC environment. In addition, the HA-ORCAR approach provides a flexible and extensible interface for customizable fault management, policy-based failover operation, and alert management.

An active/hot-standby high availability variant of Rocks has been proposed [14] and is currently under development. Similar to HA-OSCAR, HA-Rocks is sensitive to the level of failure and aims to provide mechanisms for graceful recovery to a standby master node.

Active/hot-standby solutions for essential services in scientific high-end computing include resource management systems, such as the Portable Batch System Professional (PBSPro [19]) and the Simple Linux Utility for Resource Management (SLURM [22]). While the commercial PBSPro service can be found in the Cray RAS and Management System (CRMS [20]) of the Cray XT3 [24] computer system, the Open Source SLURM is freely available for AIX, Linux and even Blue Gene [1, 7] platforms.

The asymmetric active-active architecture presented in this paper is an extension of the HA-OSCAR framework developed at Louisiana Tech University.

3. ASYMMETRIC ACTIVE-ACTIVE ARCHITECTURE

The conventional Beowulf cluster architecture (see Figure 1) has a single head node connected to a group of compute nodes. The fundamental building block of the Beowulf architecture is the head node, usually referred to as primary server, which serves user requests and distributes submitted computational jobs to the compute nodes aided by job launching, scheduling and queuing software components [2]. Compute nodes, usually referred to as clients, are simply dedicated to run these computational jobs.

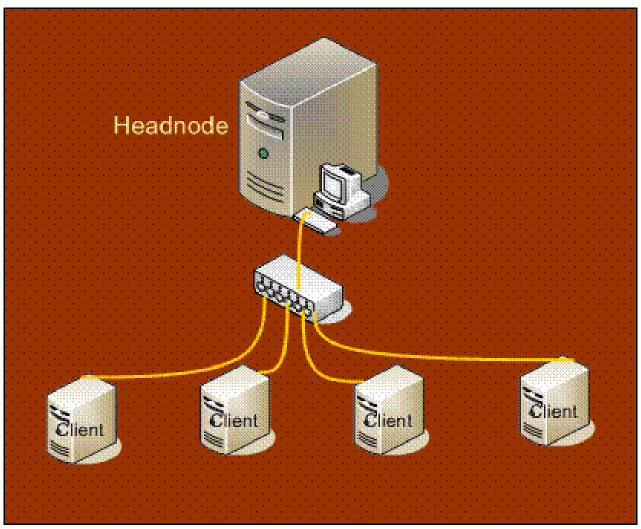

Figure 1: Conventional Beowulf Architecture

The overall availability of a cluster computing system depends solely on the health of its primary node. Furthermore, this head node may also become a bottleneck in large-scale high throughput use-case scenarios. In high availability terms, the single head node of a Beowulf-type cluster computing system is a typical single point of failure and control, which severely limits availability and serviceability by effectively cutting off healthy compute nodes from the outside world upon overload or failure.

Running two, or more, primary servers simultaneously and an additional monitoring hot-standby server for fail-over purpose, or asymmetric active-active high availability, is a promising solution, which can be deployed to improve system throughput and to reduce system down times.

We also note that preemptive measures for application fault tolerance, such as checkpointing, can introduce significant overhead even during normal system operation. Such overhead should be counted as down time as well, since compute nodes are not efficiently utilized.

We implemented a prototype of a $2+1$ asymmetric active-active high availability solution [13] that consists of three different layers (see Figure 2). The top layer has two identical active head nodes and one redundant hot-standby node, which simultaneously monitors both active nodes. The middle layer is equipped with two network switches to provide redundant connectivity between head nodes and compute nodes. A set of compute nodes installed at the bottom layer are dedicated to run computational jobs.

In this configuration, each active head node is required to have at least two network interface cards (NICs). One NIC is used for public network access to allow users to schedule jobs. The other NIC is connected to the respective redundant private local network providing communication between head and compute nodes. The hot-standby server uses three NICs to connect to the outside world and to both redundant networks. Compute nodes need to have two NICs for both redundant networks.

We initially implemented our prototype using a $2+1$ asymmetric active-active HA-OSCAR solution that consists of different job managers (see Figure 3), the Open Portable Batch System (OpenPBS [17]) and the Sun Grid Engine (SGE [23]), independently running on multiple identical head nodes at the same time. Additionally, one identical head node is configured as a hot-standby server, ready to takeover when one of the two active head node servers fails.

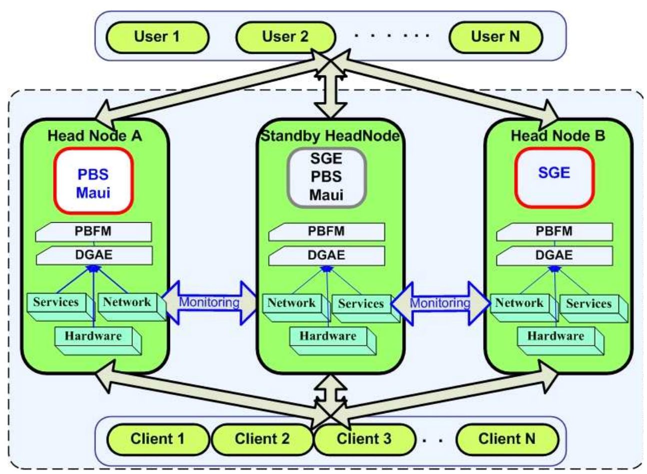

Figure 3: Normal Operation of 2 + 1 Asymmetric Active-Active HA-OSCAR Solution

Under normal system operating conditions, head node *A* runs OpenPBS and head node *B* runs SGE, both simultaneously serving users requests at tandem. Both active head nodes effectively employ the same compute nodes using redundant interconnects. Each active head node creates a different home environment for each of its resource manager, and prevents conflicts during job submission.

Upon failure (see Figure 4) of one active head node, the hot-standby head node will assume the IP address and host name of the failed head node. Additionally, the same set of services will transfer control to the standby node with respect to the same job management tool activated on the failed node masking users from this failure.

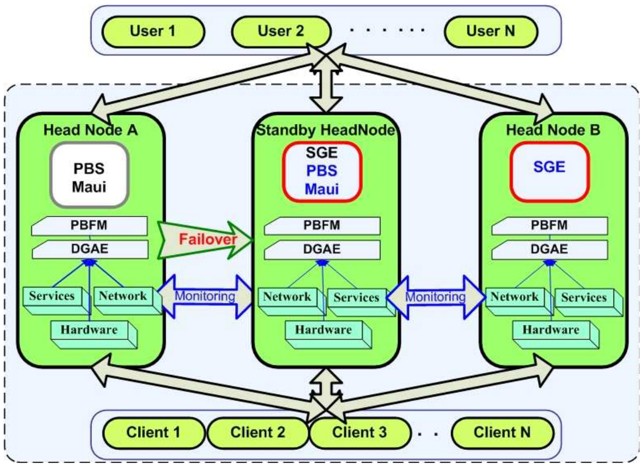

Figure 4: Fail-Over Procedure of 2 + 1 Asymmetric Active-Active HA-OSCAR Solution

The asymmetric active-active HA-OSCAR prototype is capable of masking one failure at a time. As long as one head node is down, a second head node failure will result in a degraded operating mode. Even under this rare condition of two simultaneous head node failures, our high availability solution provides the same capability as a regular Beowulf-type cluster without high availability features.

To ensure that the system operates correctly without unfruitful failovers, the system administrator must define a failover policy in the PBRM (Policy Based Recovery Management [9]) module, which allows selecting a critical head node (A or B). The critical head node has a higher priority and will be handled first by the hot-standby head node in case the DGAE (Data Gathering and Analysis Engine) detects any failures.

In the rare double head node outage event, there will not be a service failover from the lower priority server to the hot-standby head node. This policy ensures that critical services will not be disrupted by failures on the high priority head node. For example, if OpenPBS job management is the most critical service, we suggest setting the server running OpenPBS as the higher priority head node.

4. PRELIMINARY RESULTS

In our experimental setup, the head node A runs OpenPBS and Maui [15] and is assigned as the critical head node. The services on the head node A have the priority failover to the hot-standby head node. In case of a failure of head node A, the hot-standby head node takes over as head node A‘. Once head node A is repaired, the hot-standby head node will disable OpenPBS and Maui to allow those services failback to the original head node A. If head node B fails while head node A is in normal operation, the hot-standby head node will simply fail-over the SGE until the head node B is recovered and back in service again.

With our lab grade prototype setup, we experienced the same, if not better, availability behavior compared to an active/hot-standby HA-OSCAR system, if we consider the degraded operating mode of our prototype with two outages at the same time as downtime. Earlier theoretical assumptions and practical results (see Figure 5) using reliability analysis and tests of an active/hot-standby HA-OSCAR system could be validated for the asymmetric active/active solution. We obtained a steady-state system availability of 99.993%, which is a significant improvement when compared to 99.65%, from a similar Beowulf Linux cluster with a single head node.

If we consider the degraded operating mode with two outages at the same time not as downtime, availability of our prototype is even better than the standard HA-OSCAR solution. We are currently in the process of validating our results using reliability analysis. Furthermore, we also experienced an improved throughput capacity for scheduling jobs.

One of our initial concerns and one of the major challenges we encountered during prototype implementation and testing was that most if not all services in a high-end computing environment are not active-active aware, i.e. schedulers such OpenPBS and SGE do not support multiple instances on different nodes. Automatic replication of changes in job queues is not very well supported. The experience gained during implementation will be applied to future work on symmetric active-active high availability.

5. CONCLUSIONS

Our lab grade prototype of an asymmetric active-active HA-OSCAR variant showed some promising results and showed that the presented architecture is a significant enhancement to the standard Beowulf-type cluster architecture in satisfying requirements of high availability and serviceability. We currently support only $2 + 1$ asymmetric active-active high availability. However, ongoing work is currently investigating the extension of the implementation to $n+1$ active-active architectures.

Future work will be more focused on *symmetric active-active high availability* for high-end scientific computing using the *virtual synchrony* model for head node services [3]. In this architecture, services on multiple head nodes maintain common global knowledge among participating processes. If one head node fails, the surviving ones continue to provide services without interruption. Head nodes may be added or removed at any time for maintenance or repair. As long as one head node is alive, the system is accessible and manageable. While the virtual synchrony model is well understood, its application to individual services on head (and service) nodes in scientific high-end computing environments still remains an open research question.

6. REFERENCES

[1] ASCII Blue Gene/L Computing Platform at Lawrence Livermore National Laboratory, Livermore, CA, USA. http://www.llnl.gov/asci/platforms/bluegenel.

[2] C. Bookman. *Linux Clustering: Building and Maintaining Linux Cluster*. New Readers Publishing, Boston, 2003.

[3] C. Engelmann and S. L. Scott. High availability for ultra-scale high-end scientific computing. *Proceedings of 2nd International Workshop on Operating Systems, Programming Environments and Management Tools for High-Performance Computing on Clusters (COSET-2)*, 2005.

[4] HA–OSCAR at Louisiana Tech University, Ruston, LA, USA. At http://xcr.cenit.latech.edu/ha-oscar.

[5] I. Haddad, C. Leangsuksun, and S. Scott. HA-OSCAR: Towards highly available Linux clusters. *Linux World Magazine*, March 2004.

[6] J. Hsieh, T. Leng, and Y. C. Fang. OSCAR: A turnkey solution for cluster computing. *Dell Power Solutions*, pages 138–140, 2001.

[7] IBM Blue Gene/L Computing Platform at IBM Research. http://www.research.ibm.com/bluegene.

[8] Kimberlite at Mission Critical Linux. At http://oss.missioncriticallinux.com/projects/kimberlite.

[9] C. Leangsuksun, T. Liu, S. L. Scott T. Rao, and Richard Libby. A failure predictive and policy-based high availability strategy for Linux high performance computing cluster. *Proceedings of 5th LCI International Conference on Linux Clusters*, 2004.

[10] C. Leangsuksun, L. Shen, T. Liu, H. Song, and S. L. Scott. Availability prediction and modeling of high availability OSCAR cluster. *Proceedings of IEEE Cluster Computing (Cluster)*, pages 380–386, 2003.

[11] LifeKeeper at SteelEye Technology, Inc., Palo Alto, CA, USA. At http://www.steeleye.com.

[12] Linux FailSafe at Silicon Graphics, Inc., Mountain View, CA, USA. At http://oss.sgi.com/projects/failsafe.

[13] T. Liu. High availability and performance Linux cluster. *Master Thesis Report at Louisiana Tech University, Ruston, LA, USA*, 2004.

[14] T. Liu, S. Iqbal, Y. C. Fang, O. Celebioglu, V. Masheyakhi, and R. Rooholamin. HA-Rocks: A cost-effective high-availability system for Rocks-based Linux HPC cluster, April 2005.

[15] Maui at Cluster Resources, Inc., Spanish Fork, UT, USA. At http://www.clusterresources.com/products/maui.

[16] Mission Critical Linux. At http://oss.missioncriticallinux.com.

[17] OpenPBS at Altair Engineering, Troy, MI, USA. At http://www.openpbs.org.

[18] OSCAR at Sourceforge.net. At http://oscar.sourceforge.net.

[19] PBSPro at Altair Engineering, Inc., Troy, MI, USA. http://www.altair.com/software/pbspro.htm.

[20] PBSPro for the Cray XT3 at Altair Engineering, Inc., Troy, MI, USA. http://www.altair.com/pdf/PBSPro_Cray.pdf.

[21] ROCKS at National Partnership for Advanced Computational Infrastructure (NPACI), University of California, San Diego, CA, USA. At http://rocks.npaci.edu/Rocks.

[22] SLURM at Lawrence Livermore National Laboratory, Livermore, CA, USA. www.llnl.gov/linux/slurm.

[23] Sun Grid Engine Project at Sun Microsystems, Inc, Santa Clara, CA, USA. At http://gridengine.sunsource.net.

[24] XT3 Computing Platform at Cray Inc., Seattle, WA, USA. http://www.cray.com/products/xt3.

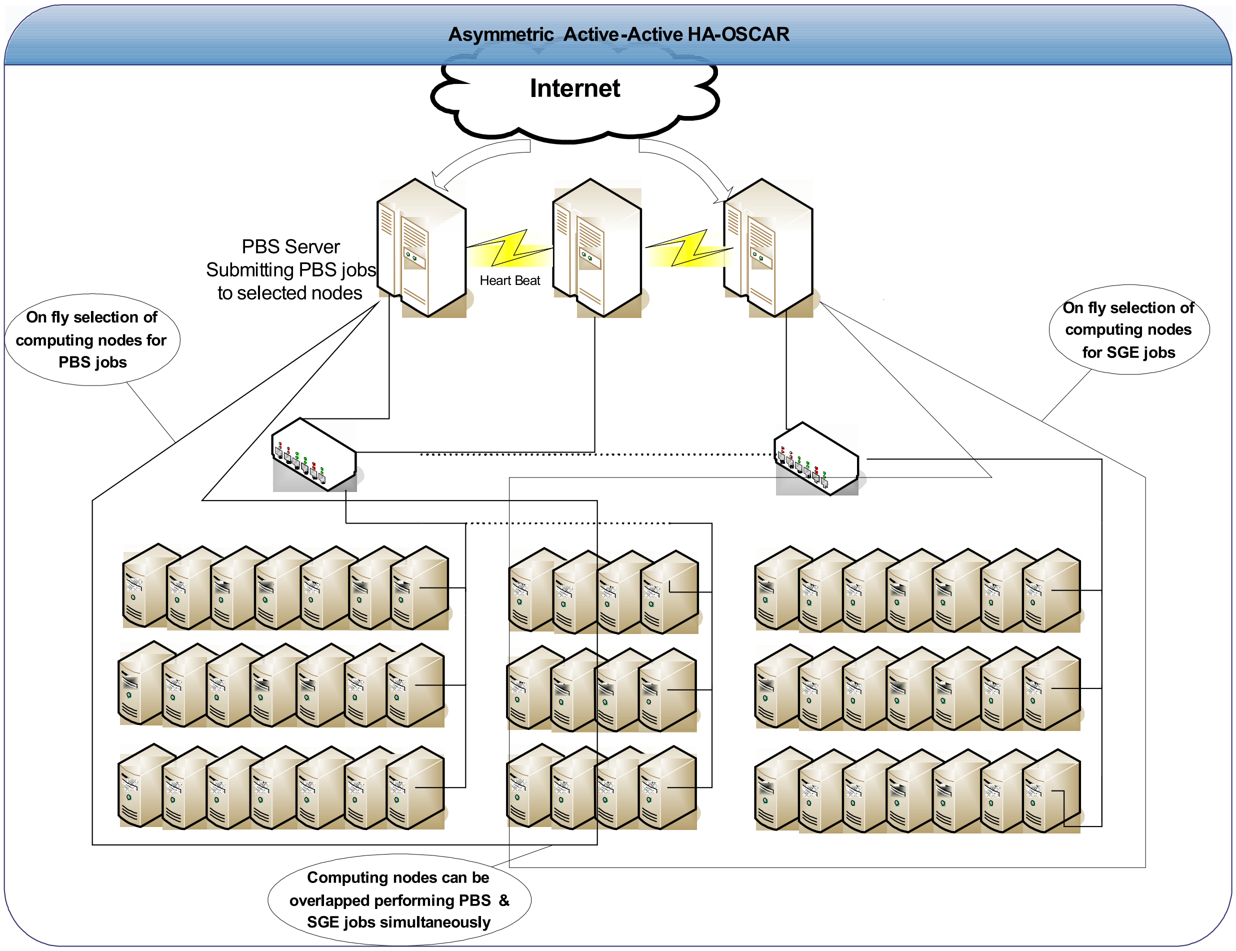

Figure 2: Asymmetric Active-Active High Availability Architecture

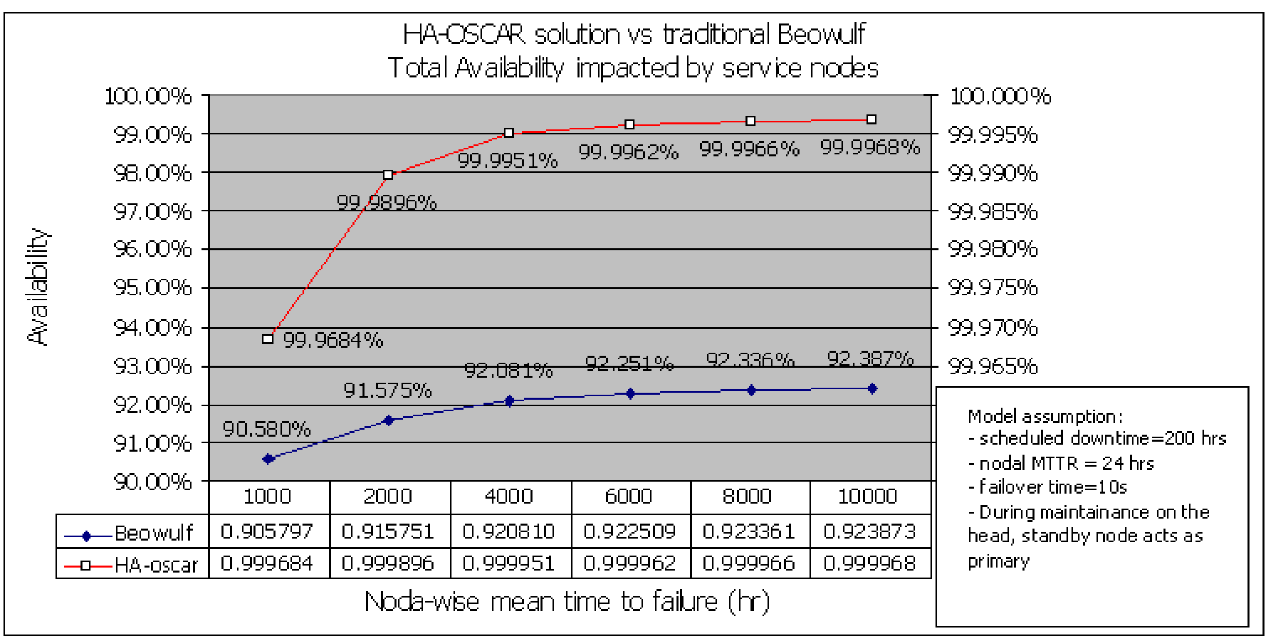

Figure 5: Total Availability Improvement Analysis (Planned and Unplanned Downtime): Comparison of HA-OSCAR with Traditional Beowulf-type Linux HPC Cluster