

AN  
SSC  
PUBLICATION

Open-Source Desktop Publishing: Scribus

# LINUX JOURNAL

LINUX  
JOURNAL  
★ READERS'  
CHOICE  
2003

The Monthly Magazine of the Linux Community • NOVEMBER 2003

**GENOME SCIENTISTS REPORT:**

## How we sequenced the SARS virus in five days

**Virtual security zones**

Highly available cluster  
management with **OSCAR**

Torture test your 1U servers

512-processor **NUMA** systems

Secure IMAP  
mail servers

Disaster plans for  
your web site

Design C++ programs  
for security

3-D visualization for  
submarine commanders

Work queues in the  
2.6 Kernel

USA \$5.00 CAN \$6.50

[www.linuxjournal.com](http://www.linuxjournal.com)





## **HA-OSCAR: the Birth of Highly Available OSCAR**

**Date:** Saturday, November 01, 2003

**Topic:** Networking

Ibrahim Haddad Chokchai Leangsuksun Stephen Scott

As clusters reach thousands of nodes, eliminating single points of failure becomes critical.

The seeds for the Open Cluster Group (OCG) and the resulting Open Source Cluster Application Resources (OSCAR) Project were planted when a number of like-minded individuals had the good fortune to sit together for dinner at a meeting sponsored by the Department of Energy on February 17, 2000. Over dinner, this group discussed the effort involved in deploying the software necessary to build a Beowulf high-performance computing cluster. Although this group agreed that it was simple to wire together commodity computers to build the cluster, they went on to agree that the amount of effort required to install and configure the requisite software stack on a Beowulf cluster was inordinately high. Challenging to some and tedious to all was the consensus description. Thus, the idea to simplify the process was born.

Further fertilization of this idea took place in Oak Ridge, Tennessee, in April that year at a meeting that included industry, academia and research labs. It was at this first formal meeting that the OCG was formed and work was begun on OSCAR, specifically the Beowulf cluster software stack and associated installation process. At this meeting the group agreed on three core principles:

1. The adoption of clusters for mainstream, high-performance computing is inhibited by a lack of well-accepted software stacks that are robust and easy to use by the general user.
2. The OCG embraces the open-source model. As a result, the OSCAR distribution must contain only freely redistributable codes, with a preference for the inclusion of source code under a Berkeley-style open-source license.
3. The OCG can accomplish its goals through the use of best-practices codes currently available.

Further details about the beginning of OCG and OSCAR can be found in an article by Richard Ferri in the June 2002 issue of *Linux Journal* titled "The OSCAR Revolution".

Throughout OSCAR's brief history, the group has managed to adhere to these three principles while allowing OSCAR to encompass other forms of software. For example, although the OSCAR distribution itself contains only freely redistributable codes, others are deploying an OSCAR package that, while not a part of the formal OSCAR distribution, may be installed or dropped into an existing OSCAR distribution for installation.

The OSCAR Project continues to contain a mixture of industry and academic/research members. The overall project is directed by a steering committee elected every two years from the current core organizations. This core list is composed of those actively contributing to project development. The 2003 core organizations include: Bald Guy Software (BGS), Dell, IBM, Intel, MSC.Software, Indiana University, the National Center for Supercomputing Applications (NCSA), Oak Ridge National Laboratory (ORNL) and Université de Sherbrooke (UdS).

In the past year, additional OCG working groups have been created to address other cluster environments. These new groups are working to leverage the technology provided by OSCAR when producing their cluster distributions. The two groups working today are Thin-OSCAR and HA-OSCAR. Thin-OSCAR is headed by the Université de Sherbrooke in Canada and is dedicated to delivering a diskless variant of OSCAR. The HA-OSCAR group is led by the authors of this article and is focused on providing a high-availability version of OSCAR.

## **HA-OSCAR: Mission, Goals and People**

In a July 2001 meeting at Ericsson Research Canada, Ibrahim Haddad made the case for high availability in cluster computing. Initially, the discussion centered on the necessity of high-availability computing for the telecom industry. As the discussions progressed, it became clear that with the anticipated tens of thousands of nodes in high-performance computing (HPC) clusters, high-availability techniques can provide some level of the fault tolerance desired by the HPC community.

Ibrahim's group at Ericsson Research worked primarily alone on the high-availability effort until the recent addition of Dr Chokchai Leangsuksun and his team at Louisiana Tech University and the continued interest in HA-OSCAR by Stephen Scott at ORNL. In 2002, the HA-OSCAR effort was recognized officially by OCG as another working group. The primary goal of the group is to leverage existing OSCAR technology and provide for new high-availability capabilities in OSCAR clusters. The anticipated customers of this technology include the telecom industry and HPC sites.

## **HA-OSCAR Improvements over OSCAR**

HA-OSCAR introduces several enhancements and new features to OSCAR, mainly in areas of availability, scalability and security. Most of these features can be mapped to ITU (International Telecommunication Union), TMN (Telecommunication Management Network) and FCAPS (Fault-management, Configuration, Accounting, Performance and Security). These concepts are widely adopted in the telecom industry to manage its network elements.

## **Dual Master Nodes and Redundancy**

A typical cluster computing architecture consists of several nodes that can provide some degree of availability. However, it normally has a single-head node that is a simplex architecture and prone to single points of failure. The current release of OSCAR falls into this architectural category, which is unsuitable for mission-critical systems as it contains several individual system elements that have no redundancy for a backup or failover. In order to support HA requirements, clustered systems must provide ways to eliminate single points of failure.

Hardware duplication and network redundancy are common techniques utilized for improving the reliability and availability of computer systems. To build an HA-OSCAR cluster system, we first must provide a duplication of the cluster head node. Such an architecture can be implemented in different ways, including active-active, active-warm standby and active-cold standby.

The active-active model enables both performance and availability, because both head nodes simultaneously can provide services. However, its implementation is quite complicated and leads to data inconsistency when failures occur. Active-standby options mostly are adopted solutions. The standby server watches the primary server health and can take over control when it detects an outage. Currently, the active-warm standby configuration is the initial model of choice.

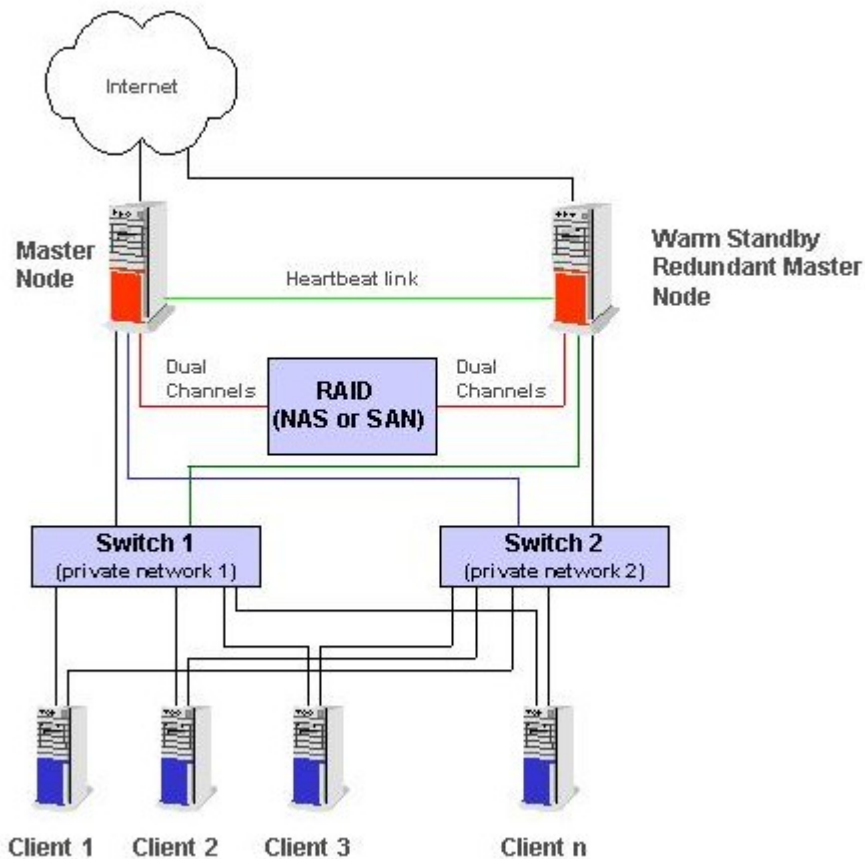


Figure 1. HA-OSCAR Cluster System Architecture

Figure 1 shows the HA-OSCAR cluster system architecture. We experimented with and planned to incorporate Linux Virtual Server and Heartbeat mechanisms into our initial active-hot standby HA-OSCAR distribution. Now, we plan to extend our initial architecture to support active-active HA after we release the hot-standby distribution. The active-active architecture can better utilize resources, because both head nodes can be simultaneously active to provide services. The dual master nodes then can run redundant DHCP, NTP, TFTP, NFS and SNMP servers. In the event of a head node outage, all functions provided by that node failover to the second redundant head node and are served at a reduced performance rate (in theory, 50% at the peak or busy hours).

Another HA functionality to support in HA-OSCAR is providing a high-availability network using redundant Ethernet ports on every machine. In addition, duplicate switching fabrics (network switches, cables, etc.) are used for the entire network configuration. This enables every node in the cluster to be present on two or more data paths within its networks. Backed with this Ethernet redundancy, the cluster achieves higher network availability. Furthermore, when both networks are up, improved communication performance may be achieved by using techniques such as channel bonding of messages across the redundant communication paths.

HA-OSCAR aims to reuse features from other implementations and existing projects, including the High-Availability Linux, Kimberlite and Linux Virtual Server projects. We then plan to contribute the added enhancements and functionalities back to the community.

## Support for Dual IP Stack (IPv4 and IPv6)

IPv6 is the next-generation protocol designed by IETF to replace the current version of the Internet Protocol, IPv4. Most of today's Internet uses IPv4, which has been remarkably resilient in spite of its age, but it is beginning to have problems. Most importantly, there is a growing shortage of IPv4 addresses, which are needed by all new devices connecting to the Internet. As a result, IETF defined IPv6 to fix the problems in IPv4 and to add many improvements for the future Internet. These improvements come in different areas, such as routing, autoconfiguration, security, QoS and mobility.

HA-OSCAR has support for IPv6 activated by default. Most of the ISPs and telecom companies already are experimenting with co-existence schemes for IPv4 and IPv6. All cluster nodes installed with HA-OSCAR provide support for IPv6 and basic IPv6 capabilities compiled directly in the network utilities and binaries.

## Recovering from Corrupted Disks

OSCAR assumes the client node disks on which it is installing are faultless. But, this is not always the case; some nodes may have corrupted disks. HA-OSCAR considers this issue and does not assume that all disks on all nodes are a good installation base. To this end, we support special scripts in our installs and software RAID in the kernel, in parallel with developing the necessary set of scripts needed to synchronize disk contents. As such, if a disk fails, data is not lost. In addition, our installation wizard first tries to fix the corrupted disk. HA-OSCAR also supports synchronous operation, disk removal and disk insertion. In addition, HA-OSCAR supports software RAID by default. By enabling software RAID, clusters powered by HA-OSCAR have increased data redundancy and better performance.

## Linux Virtual Server (LVS) and Heartbeat Integration

We already described the HA-OSCAR hardware architecture. A key enhancement is the addition of dual head nodes that provide a backup head node for a failover in case of a primary head node outage. However, a hardware redundancy solution alone is not sufficient to archive HA unless detection and recovery mechanisms are incorporated.

Few existing solutions provide outage detection and failover. We have evaluated and selected a failover LVS. The solution includes LVS, Linux Director Dæmon (ldirectord), Heartbeat and Coda. Linux Virtual Server is a software tool that directs network connections to multiple servers that share a workload. ldirectord is a standalone dæmon to monitor services. Heartbeat provides a primary node outage detection and failover mechanism through serial line and UDP connectivity. Coda is a fault-tolerant distributed filesystem. This solution not only provides HA capability, but load balancing as well. However, additional LVS services must be enhanced in HA-OSCAR, including SIP, PBS and Web services. An external Heartbeat (eHB) mechanism to a fault management system also has been added. eHB is a precaution in case of a total outage (for example, double head node failures) from which the fault management detects, raises an alarm and sends a page to a system administrator.

## Cluster-Wide Security

OSCAR currently is installed on clusters deployed mostly on private networks, where security is not a major concern. That is because these clusters are not connected to any networks outside the lab

boundaries. However, when HA-OSCAR is deployed on clusters connected to the Internet, security is vital. Security is a major concern for both OSCAR and HA-OSCAR not only because a hacker might access the cluster and the data sitting on it, but also because a malicious hacker also might disrupt the normal workings of the system and its availability.

Many security solutions exist, ranging from external solutions (firewalls) to internal solutions (integrity-checking software). Unfortunately, all of them are based on a single node approach and lack a homogeneous view of the cluster. Most of the time, administrators end up installing, patching, integrating and managing several security solutions. The increased management difficulty soon leads to decreased security, as interoperability issues increase with updates of the heterogeneous pieces.

Consequently, the Distributed Security Infrastructure (DSI) was initiated as an open-source project to provide an adequate security solution for carrier-grade clustered servers. DSI is a security framework that provides applications running on clustered systems with distributed mechanisms for access control, authentication, confidentiality and integrity of communications. It also provides auditing services with process-level granularity.

Therefore, HA-OSCAR can be more successful with telecom and other mission-critical sectors if it supports advanced security features. For this reason, HA-OSCAR adopted DSI from Ericsson.

## **Dynamic Addition and Removal of Nodes**

HA-OSCAR supports a mechanism that allows users to add and remove nodes from the cluster dynamically, in a transparent fashion, without affecting either the end-user experience or the running applications. Two open-source projects provide similar functionalities: Eddie, an Ericsson open-source initiative, and LVS. We currently are investigating the best mechanism and will implement it in HA-OSCAR. Our goal is to ensure that adding nodes to accommodate higher traffic or removing nodes for service purposes is a seamless operation and does not affect service availability.

## **Linux Kernel**

The subject of which kernel to adopt came in addition to the decision about whether to patch the HA-OSCAR kernel ourselves or try to have our patches accepted by the mainstream kernel tree. We decided to use the latest stable 2.4 kernel and submit the patches we create to the kernel mailing list. We are trying to provide a simplified kernel building tool for these HA-OSCAR users. Users can recompile based on their local configurations.

## **Support for Network Filesystems**

A network/distributed filesystem is an essential component for building clusters. A number of open-source projects aim to provide network filesystems for Linux clusters. Based on our previous research and lab testing, we ascertained that a different networked filesystem may be required, depending on the type of applications being run on the cluster. For instance, using the parallel virtual file system (PVFS) offers the advantage of high I/O performance for large files on a streaming video and audio server. On the other hand, sharing configuration files among cluster nodes can be achieved using the NFS without the need for high I/O. If it is desirable to maintain high availability and support storage area networks (SANs), OpenGFS, with its journaling capability, can handle such a task. Therefore, HA-OSCAR is working to support all the possible network filesystems that can be used in target environments.

## Fast Cluster Setup

One important factor to consider is the time it takes to build, boot and have the cluster ready to service requests. This is not a major issue for small clusters, but as we move to large installations of 256 nodes and higher, having the capabilities of installing and booting all cluster nodes in an automated and timely manner becomes an asset. HA-OSCAR is considering implementing hierarchical clustering by dividing the cluster into multiple zones. This type of experimentation also can be helpful in identifying the slow processes in the system installation procedure, which allows us to bring it up to speed. LinuxBIOS, for instance, can be included in place of the normal BIOS—with a little bit of hardware initialization and a compressed Linux kernel that can be booted from a cold start—to achieve faster startup times. The upcoming OSCAR release uses multicast technology, which was tested on about 500 nodes, to speed up install times and return impressive numbers. HA-OSCAR plans to adopt this method as the base install mechanism and improve on it.

## Selective Installs

Similar to the base OSCAR installation, users of HA-OSCAR have the freedom of deciding which application packages to install. By default, HA-OSCAR automatically installs the essential parts to build a cluster and then prompts the user to select the applications they want.

The installation procedure takes into consideration any existing configuration and the packages already installed on the node. Some packages are sensitive to certain system libraries, such as glibc. Users should be aware that installing HA-OSCAR may require them to upgrade their systems based on such dependencies. In the same manner, a de-installation procedure is provided to clean up every HA-OSCAR-specific addition without disturbing the system integrity. This option is important for users who want to test only HA-OSCAR.

It is also worth mentioning that package install and uninstall options are available in the base OSCAR release since v2.0, and a newly enhanced version is coming out soon.

## Network Upgrades

HA-OSCAR plans to investigate the possibility of providing mechanisms for selective network software upgrades without bringing down the system. Network upgrades are an interesting way of patching an operating system and its applications. As an example, most Linux distributions now come with an automatic network upgrade that eases this tedious administrative task. In the case of administrating a large cluster, HA-OSCAR users can use such a feature to upgrade their application version seamlessly, without service interruption. Network upgrade simplifies cluster administration and promotes better software management across all computing nodes.

In addition, HA-OSCAR provides a tool that allows users to change the configuration of the cluster at runtime by using a tool somewhat similar to LinuxConf. This is still a basic idea that will be investigated further in the near future.

## Backups, Restores and Disaster Recovery

Generally, one cannot trust a computing system if there is no backup or recovery mechanism. For mission-critical applications, including telecom applications, it is important to be able to recover from any software or hardware failure. Thus, providing efficient backup and recovery mechanisms is an

essential part of any HA system.

In case a disaster occurs, recovery ability and speed are critical. Every time HA-OSCAR is completely re-installed or the kernel updated, ghost images of before and after are saved in a designated location on a backup server and tape. Ghost for Unix takes a snapshot of an old and new kernel, gzips it and sends the image to the secondary head node as well as to a predefined disaster recovery site. Important data as well as application and configuration files also can be included in the ghost image. Normally, tape backup schedules include nightly snapshots for incremental images and weekly snapshots for full images. For faster recovery and highly reliable backups, ghost imaging, file journaling and data replication are implemented.

## Supporting Web Clusters

One goal of HA-OSCAR is to be deployed optionally as a Web server cluster providing highly available Web services to a large number of clients. One step toward this goal is to set up a Web server, such as Apache, on every node; Apache can be one of the packages copied to the nodes. Then, a single IP interface is provided for the cluster, possibly using LVS Direct Routing, because it has proven to be the scalable implementation.

## Support for Asynchronous Process Execution

Telecom applications must be built to face extreme or unplanned conditions of execution. Even in typical real-life situations, subscribers are putting a lot of pressure on carriers because of their high expectations regarding system performance and availability. Customers do not expect these applications to fail or their phone requests to be delayed beyond a typical threshold. This is increasingly true as telecom applications are providing additional services, some requiring real-time characteristics.

Carrier-grade applications must be designed with these subscribers' constraints in mind, taking into account the cost of software maintenance and upgrades, service availability and scalability. Complex distributed software demands a specific programming paradigm. It has been proven over the years that complex system interfaces tend to increase the time to debug and the probability of application failure.

AEM (asynchronous event mechanism) provides an event-driven methodology of development in order to provide robust applications with a mechanism that allows reacting quickly to system events by means of user-space callbacks. In the AEM implementation, the kernel plays a major role in handling events and increases the reliability of applications. For this reason, AEM provides a flexible solution for application designers, supplying an extensible framework that allows new functionalities to be added at runtime, without rebooting the system or restarting applications. In order to reach carrier-grade requirements, HA-OSCAR plans to supply efficient support for asynchronous events.

### Resources

AEM: [www.linux.ericsson.ca/aem](http://www.linux.ericsson.ca/aem)

CODA: [www.coda.cs.cmu.edu](http://www.coda.cs.cmu.edu)

DSI: [www.linux.ericsson.ca/dsi](http://www.linux.ericsson.ca/dsi)

Eddie: [eddie.sourceforge.net](http://eddie.sourceforge.net)

FCAPS: [www.iec.org/online/tutorials/ems/topic03.html](http://www.iec.org/online/tutorials/ems/topic03.html)

g4u: [www.feyrer.de/g4u](http://www.feyrer.de/g4u)

GFS: [www.globalfilesystem.org](http://www.globalfilesystem.org)

GPL: [www.gnu.org/copyleft/gpl.html](http://www.gnu.org/copyleft/gpl.html)

Linux HA: [linux-ha.org](http://linux-ha.org)

LVS: [www.linuxvirtualserver.org](http://www.linuxvirtualserver.org)

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

Open System Lab: [www.linux.ericsson.ca](http://www.linux.ericsson.ca)

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

The OSCAR Revolution: [www.linuxjournal.com/article/5559](http://www.linuxjournal.com/article/5559)

PVFS: [parlweb.parl.clemson.edu/pvfs](http://parlweb.parl.clemson.edu/pvfs)

Ibrahim Haddad ([Ibrahim.Haddad@Ericsson.com](mailto:Ibrahim.Haddad@Ericsson.com)) is a researcher at the Open System Lab, Ericsson Research Corporate Unit. He is coauthor, along with Richard Peterson, of the *Red Hat Linux Pocket Administrator* from McGraw-Hill, to be published in September 2003.

Chokchai Leangsuksun ([box@latech.edu](mailto:box@latech.edu)) is an associate professor of computer science at the Center for Entrepreneurship and Information Technology (CEnIT) at Louisiana Tech University. Prior to his academic career, he spent seven years in R&D with Lucent Technologies in system reliability and high-availability computing and telecommunication systems.

Stephen L. Scott ([scottsl@ornl.gov](mailto:scottsl@ornl.gov)) is a senior research scientist in the Computer Science and Mathematics Division of Oak Ridge National Laboratory, US. He is a founding member of OCG and presently is version 2 release manager. Previously he was the working group chair of the OSCAR Project.

