The Modeling and Dependability Analysis of High Availability OSCAR Cluster System

Chokchai Leangsuksun[a], Lixin Shen[a], Hertong Song[a], Stephen L. Scott[b], Ibrahim Haddad[c]

[a]Department of Computer Science, Louisiana Tech University, Ruston, LA 71272, USA
[b]Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA
[c]Open Systems Lab, Ericsson Corporate Unit of Research, Town of Mont Royal, Quebec H4P 2N2, Canada

With over 85,000 downloads to date, OSCAR is the most widely used, freely available, open source software stack for the building and maintaining of high performance computation clusters. The desire for a high availability cluster computing solution is presently coming from two directions. First, traditional high performance computing clusters are growing in size beyond that which one can reasonably expect an application to complete prior to a hardware failure. Second, there is a growing demand for the use of high performance computing systems as platforms for mission critical or highly available computing. In this paper, the current OSCAR cluster computing system is presented. Reliability oriented and architectural features are then introduced to render a high availability OSCAR environment. Continuous Time Markov Chain models are then built and applied to these two systems; the classic OSCAR and the high availability OSCAR. Based on these models, the dependability of the two systems is analyzed and compared. The results show that the high availability path is reasonable and will significantly improve the availability of OSCAR cluster system.

Avec plus de 85000 téléchargements jusqu'à maintenant, OSCAR est la suite logicielle gratuite à code libre la plus utilisée pour l'installation et l'entretient des grappes de calcul de haute performance. La demande pour des grappes de calcul de haute disponibilité provient de deux tendances. Primo, les grappes de calcul de haute performance traditionnelles augmentent en taille, et l'on ne peut plus raisonnablement espérer compléter un calcul distribué avant que ne survienne un bris matériel. Secondo, il y a croissance de l'utilisation des systèmes de haute performance en tant que systèmes vitaux, ou encore en informatique de haute disponibilité. Dans cet article, nous décrivons l'environnement OSCAR. Nous lui ajoutons ensuite de nouvelles fonctions de fiabilité permettant d'obtenir un environnement OSCAR de haute disponibilité. Des modèles Continuous Time Markov Chain sont ensuite mis au point et appliqués à deux ensembles: la version classique de OSCAR et la version haute disponibilité de OSCAR. En se basant sur ces modèles, on analyse et compare la sécurité de fonctionnement de ces deux ensembles. Les résultats montrent que l'option haute disponibilité est intéressante et améliore notablement la stabilité de l'environnement OSCAR.

1. Introduction

Open Source Cluster Application Resources (OSCAR) [1, 6, 7] is a fully integrated, easy to install software package designed to facilitate the creation and operation of a Linux cluster for high performance computing. It has been widely used for building and maintaining Beowulf clusters, which are the fastest growing architecture choice for high-performance parallel computing systems [2]. High-performance systems are also in growing demand as platforms for mission critical applications. In many cases, high availability (HA) requirement becomes as critical as high performance. We anticipate that for OSCAR to be adopted for use in mission-critical and enterprise environments such as telecommunications or the defense industry will require the addition of high availability features [3].

From the HA view point, the current OSCAR release is not suitable since its system reliability components are exposed to multiple single points of failure. These failure rates will dominate the availability of the entire system. In order to increase system reliability, high availability requirements will be applied to several of these key components. One solution is to build redundancy in the system. Duplicating key components will significantly improve the availability of the redundant components and that of the whole system. However, the overall price/performance ratio of the system will most likely rise in direct relationship to the introduced redundancy and therefore may compromise the cost-conscious advantages of the Beowulf cluster system. It is also very possible that system redundancy will introduce additional system overhead, which will in turn reduce overall performance thereby reducing the price/performance ratio even further.

Dependability analysis demonstrates whether or not the HA requirements will be met without building an actual system during the design stage. It provides a good insight for examining the reliability characteristic of these systems. Continuous Time Markov Chain (CTMC) [6] is a useful modeling formalism for dependability analysis of computer systems. It can easily handle many of the interdependencies and dynamic relationships among the system components [4].

In this paper, we introduce reliability-oriented requirements and architecture that will render a high availability OSCAR cluster system. We then demonstrate the HA-OSCAR reliability improvement via CTMC technique. In the following section, we will discuss the architectures of the OSCAR cluster system and the proposed HA-OSCAR cluster system. Section three (3) describes the corresponding CTMC models for the two systems and is followed by their dependability analysis. The paper is ended with concluding remarks and a discussion of future work.

2. System Architecture

Before diving into the dependability analysis, system characteristics are investigated. We first examine the current OSCAR architecture and its anatomy. Our goal is to identify the potential single-point-of-failure components. This will provide an opportunity to introduce system level redundancy to produce a high availability initiated improvement over the existing OSCAR cluster framework. However, only a brief description of the proposed architecture is provided here. Additional HA-OSCAR details may be found in [3].

2.1 Oscar Cluster Architecture

Figure 1 shows the architecture of an OSCAR cluster system supported by the current release (i.e. release 2.1, at the time of writing). Each individual machine within a cluster is referred to as a node. There are two types of nodes: server node (also called "head node") and client node (also known as "computing node" or simply "node"). The server node is responsible for providing service requests and routing appropriate tasks to client nodes. A client node is primarily dedicated to computation. The present OSCAR cluster architecture consists of a single server node and a number of client nodes, where all the client nodes contain somewhat homogeneous hardware. The server node and client nodes all communicate on a private Ethernet network not exposed to the outside world [2]. To provide cluster access to the external network, the server node is dual ported with a second network interface card connected to the outside world via a public network address. The server node file system is NFS mounted, contains the home directories of all users, and runs various service-providing servers, such as the PBS queue system and MAUI scheduler. Each client has a complete local copy of the operating system and other software, with the exception of aforementioned NFS mounted users' home directories on the server storage.

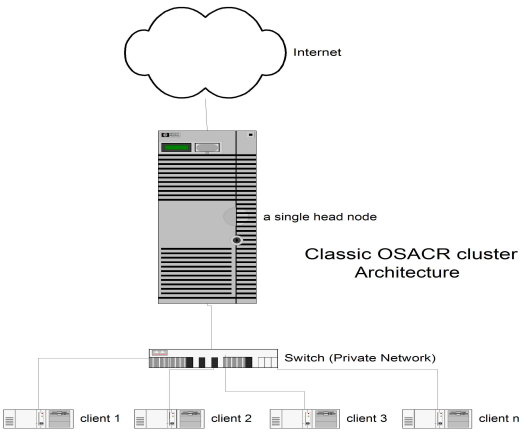

Fig. 1. Architecture Supported by Current OSCAR Release

2.2 HA OSCAR Cluster Architecture

The OSCAR working group of the Open Cluster Group continues to update the packages in OSCAR to ensure that it represents the best-known practices for building, using, and managing clusters [1]. The current release of OSCAR is not suitable for mission critical systems as it contains several individual system elements that are a single-point-of-failure. In order to support HA requirements, clustered systems must provide ways to eliminate single-point-of-failures. This will then allow them to be considered for adoption in environments that require high reliability and high availability such as the Telecommunications industry, Network or Application Service Providers (ASP), National Security, Military, etc.

Hardware duplication and network redundancy are common techniques utilized for improving the reliability and availability of computer systems. To achieve an HA-OSCAR cluster system, we must first provide a duplication of the cluster head node. There are different ways for implementing such an architecture, which includes Active-Active, Active-Warm Standby and Active-Cold Standby [7]. Currently, the Active-Active configuration is the model of choice since both head nodes can be simultaneously active to provide services. The dual master nodes will run redundant DHCP, NTP, TFTP, NFS and SNMP servers. In the event of a head node outage, all functionalities provided by that node will fail-over to the second redundant head node and will be served at a reduced performance rate (i.e. in theory, 50% at the peak or busy hours).

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

Figure 2 shows the HA-OSCAR cluster system architecture. Each of the duplicate server nodes is connected to the external network by two or more different links. These redundant links will keep the system connected to the external environment if one of the network systems should fail. Inside the cluster connectivity, on the private network, each server node is separately connected to two switches and each client node is connected to both switches providing two redundant switching fabrics via the Ethernet connections.

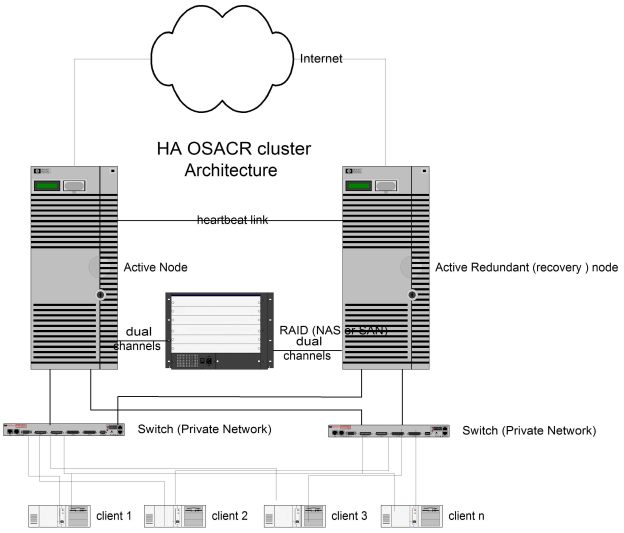

Fig. 2. A HA OSCAR Cluster System

3. System Model

Based on our previous study [8], we select the state-space model for reliability improvement evaluation. Our analysis focuses on servers and switches that dominate cluster availability.

We made several assumptions to reduce the size of the state-space as follows:

- Time of failure for each component is exponentially distributed, with the parameters being λ_{SV} for the servers and λ_{SW} for the switches, respectively.
- Failed components can be repaired.
- Times to repair a server and switches are exponentially distributed with parameters μ and β.
- When the system is down, no further failure can take place. Hence, for OSCAR cluster, when the server is down, no further failure can take place on the switch. Similarly, when the switch is down, no further failure can take place on the server. For the HA-OSCAR cluster, when both servers are down, no further failure can take place on the switches. Likewise, when both switches are down, no further failure can take place on the servers.

The following section contains more details regarding each of the OSCAR system models.

3.1 Classic OSCAR Cluster System Model

The CTMC model corresponding to the OSCAR cluster system is shown in Figure 3. At the state 1, both server nodes and switches are functioning properly. The transition to state 2 occurs if a server node has a failure and the transition from state 1 to state 3 occurs when a switch has a failure. The system will be available for service in state 1, and will be unavailable for state 2 and state 3. The system goes from state 1 to state 2 when server failure occurs at rate λ_{SV}, and from state 1 and to state 3 when switch failure occurs at rate λ_S. After server recovery at rate μ, the system is back in state 1 from state 2. Moreover, after switch recovery at rate β, the system is back in state 1 from state 3. Ultimately, we must try to keep system in the state 1 as long as possible.

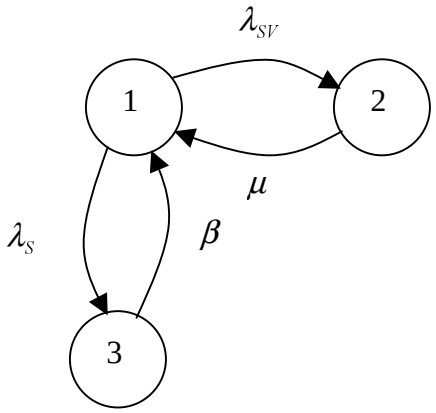

Fig. 3. CTMC for Current OSCAR Cluster System

3.2 HA OSCAR Cluster System Model

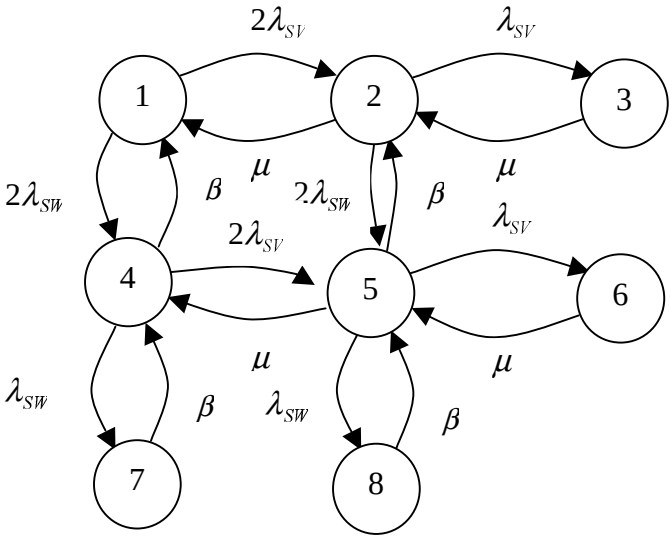

Fig. 4. CTMC for HA-OSCAR Cluster System

The CTMC model corresponding to HA-OSCAR cluster system is shown in Figure 4. The states and their corresponding system status are shown in Table 1. The system will be available for service in states 1, 2, 4 and 5, and will be unavailable during states 2, 6, 7 and 8. The system goes from one state to another at the rates mentioned in the corresponding arrow lines in Figure 4.

Table 1. System Status

State Number	Number of Servers (Up)	Number of Switches (Up)	System Status
1	2	2	Up
2	1	2	Up
3	0	2	Down
4	2	1	Up
5	1	1	Up
6	0	1	Down
7	2	0	Down
8	1	0	Down

4. Availability Analysis

Let π_i be the steady-state probability of state *i* of the CTMC. They will satisfy the following equations:

$$\pi Q = 0$$

and

$$\sum_{i \in \Omega} \pi_i = 1$$

where *Q* is the infinitesimal generator matrix [5]. Let *U* be the set of up states, the availability of the system *A* is

$$A = \sum_{k \in U} \pi_k$$

4.1 OSCAR Cluster System Analysis

We compute the steady-state probabilities by balance equations, and finally we have the steady-state availability

$$A = \pi_1 = \frac{1}{E} \qquad (1)$$

where

$$E = 1 + \frac{\lambda_{SV}}{\mu} + \frac{\lambda_{SW}}{\beta}.$$

4.2 HA OSCAR Cluster System Analysis

We compute the steady-state probabilities by balance equations, and finally we have the steady-state availability

$$A = \pi_1 + \pi_2 + \pi_4 + \pi_5$$

$$= \frac{1 + \frac{2\lambda_{SV}}{\mu} + \frac{2\lambda_{SW}}{\beta} + \frac{4\lambda_{SV}\lambda_{SW}}{\mu\beta}}{E} \qquad (2)$$

where

$$E = 1 + \frac{2\lambda_{SV}}{\mu} + \frac{2\lambda_{SW}}{\beta} + \frac{2\lambda_{SV}^2}{\mu^2} + \frac{2\lambda_{SW}^2}{\beta^2} + \frac{4\lambda_{SV}\lambda_{SW}}{\mu\beta} + \frac{4\lambda_{SV}^2\lambda_{SW}}{\mu^2\beta} + \frac{4\lambda_{SV}\lambda_{SW}^2}{\mu\beta^2}$$

4.3 Comparison and Example

For a hypothetical system, we assume that $\lambda_{SV} = 0.001\ hr^{-1}$, $\lambda_{SW} = 0.0005\ hr^{-1}$, $\mu = 0.5\ hr^{-1}$, and $\beta = 1.0\ hr^{-1}$. By formula 1 and 2, we can calculate the availability of the system. The availability for OSCAR cluster is 0.996, and the availability for the HA-OSCAR cluster is 0.99999. The downtime of the two systems in a year is 39.2 hours and 4.45 minutes, respectively. Typically, a high-availability system is one that has a downtime that does not exceed "five-nines" or 0.99999.

5. Conclusions and Future Work

Through the analysis and comparison of the theoretical dependability of OSCAR and HA-OSCAR cluster systems, we are able to conclude that the availability of HA-OSCAR cluster systems can be significantly higher than that of OSCAR cluster systems without going through the actual cluster construction process. This encouraging result has convinced us to pursue an experimental development of a prototype HA-OSCAR system infrastructure. We plan to explore additional ways to extend the availability, robustness and reliability of an HA-OSACR system already in development. We expect that a successful HA-OSCAR implementation will spur an increased adoption of OSCAR in the enterprise and mission-critical computing environments.

6. References

1 The Open Cluster Group, *OSCAR: A packaged Cluster Software stack for High Performance Computing*, http://www.openclustergroup.org/

2 S.L. Scott, T. Naughton, B. Barrett, J. Squyres, A. Lumsdaine, Y.C. Fang, and V. Mashayekhi, *Looking Inside the OSCAR Cluster Toolkit*, Dell Power Solution magazine (online and printed editions), http://www.dell.com/us/en/esg/topics/power_ps4q02-oscar.htm, November 2002.

3 I. Haddad, F. Rossi, C. Leangsuksun, S. L. Scott, *Telecom/High Availability OSCAR Suggestions for the 2nd Generation OSCAR, Technical Report TR-LTU-12-2002-01*, Computer Science Program, Louisiana Tech University, December 2002.

4 J. Muppala, M. Malhotra, K. S. Trivedi, *Markov Dependability Models of Complex Systems: Analysis Techniques, Reliability and Maintenance of Complex Systems*, S. Ozekici (ed.), pp. 442-486, Springer-Verlag, Berlin, 1996.

5 K. S. Trivedi, *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*, John Wiley and Sons, New York, 2001.

6 http://www.ee.duke.edu/~kst

7 P. S. Weygant, Cluster for high availability: a primer of HP solutions, 2nd ed., 2001, Hewlett-Packard Company, Prentice-Hall, Inc.

8 S. Lixin, and C. Leangsuksun, Techniques for System Dependability Evaluation, *Technical Report TR-LTU-09-2002-01*, Computer Science Program, Louisiana Tech University, September 2002.