# Cluster Management Software for Open Object Oriented Systems *

Andrzej Uszok, Andrzej Król, Krzysztof Zieliński

University of Mining & Metallurgy
Institute of Computer Science, Al. Mickiewicza 30
30-059 Kraków, Poland
email: kz@daisy.ics.agh.edu.pl

This paper addresses an issue of a cluster management software which supports parallel jobs processing in an object-oriented environment. A job submitted for computation represents a collection of communicating objects that cooperate with each other to perform a task.

The cluster management software consists of two related parts, i.e. load monitoring (LM) system and object dispatcher, called load balancer (LB). The following assumptions have been taken as a design decisions: the cluster management software has been designed as an object-oriented framework for applying various monitoring strategies and load sharing algorithms; the monitoring functions [1] have been limited only to external activity of computational objects expressed in terms of their resource requirements; user written application software should run without any changes under the proposed management system.

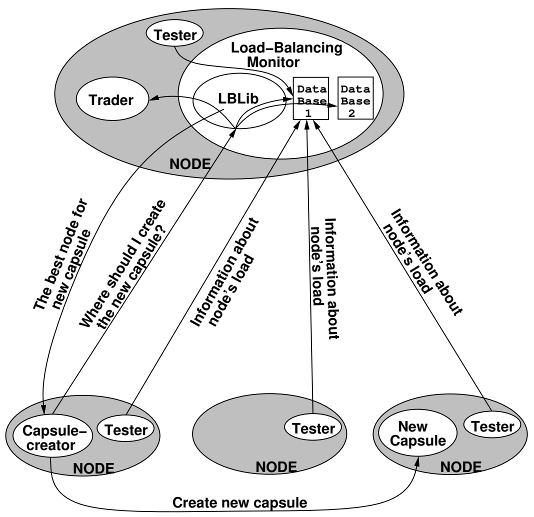


Fig. 1 Overview of Cluster Managment System

An implementation of the system is based on ANSA [2] and represents an extension of the original functionality of the software's version 4.0 with monitoring and LB modules. The described version of the implementation has centralized architecture (Fig. 1). LM system has been written as an ANSA application. Load, treated as a vector of parameters is measured on every computer in the cluster. This is done by special capsules, called *Tester*, which use *rstat* function or read directly from the system's kernel. *Testers* send collected information about system state to a central data base. Another data base contains information about nodes' resources. These data are used for calculation of various metrics of aggregated nodes' load. Various updating strategies of these data

* This work was sponsored by KBN under grant no.: 8 S503 015 06

bases (periodically, on request and event-driven) may be applied during work of this system.

LB module – *LBLib*, using metrics calculated by LM, makes decision about placement of each new object created in the system. When an application sends a request for a new service, the server object may be created on the best node according to the currently active load-sharing strategy.

An extended version of the LB system in which objects can migrate to less loaded nodes during their activity has also been designed and is being implemented.

Two programming styles are provided for the application programmer: transparent and non-transparent usage of management functions. The transparent mode makes it possible for all applications to use provided management functions in implicit form. Hence any existing ANSA programs may use the management software without any changes. For this purpose an application program is processed with the specially designed preprocessor which automatically finds the operations of new objects creation and replaces them with the calls to the management system. In the non-transparent mode an application programmer should explicitly insert calls to the management functions into a program. Available functions are: *BestNode*, *NodesList*, *GoodNodes*, *NewGoodNodes*, *BadNodes* and *StillGoodNodes*. These operations return subset of nodes according to actual arguments and the active load-sharing strategy. Achieved information may be useful in making decision about the creation of new objects in a system.

The implemented management software has been tested for some typical applications for instance processing farm computation. To get more general evaluation of the implemented software and various monitoring and LB strategies applicability, a simulator of cluster load has been also implemented (in ANSA).

The presented idea of management software for a cluster of workstation may be easily implemented in a modern CORBA environment [3]. Most of our current solutions may be applied without any changes. Some extension are also possible because of a distributed nature of the CORBA (Orbix [4] – implementation of CORBA is used), availability of filtering mechanism and Dynamic Invocation Interface. Our current work concentrate on making this management system available not only in ANSA environment but also in other distributed computing system. An effective control of cluster's load will be possible only if all applications running in a network are manageable.

## References


1. Masoud Mansouri-Samani, Morris Sloman, Monitoring Distributed Systems, Imperial College Research Report No DOC92/23, April 1993.
2. Application Programmer's Manual – ANSAware 4.0, APM Ltd. Cambridge UK, March 1992.
3. Draft Common Object Request Broker Architecture Revision 1.1, OMG Report 91-12-1, OMG Inc., December 1991.
4. The Orbix Architecture, IONA Technologies Ltd., August 1993.


This article was processed using the LaTeX macro package with LLNCS style