Simulation Modelling Practice and Theory 17 (2009) 327–347

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat

Analytical modelling and simulation of small scale, typical and highly available Beowulf clusters with breakdowns and repairs

Enver Ever [a,*], Orhan Gemikonakli [a], Ram Chakka [b]

[a] Middlesex University, The Burroughs, Hendon, London NW4 4BT, UK
[b] Meerut Institute of Engineering and Technology, Meerut 250 002, India

ARTICLE INFO

Article history:
Received 10 July 2007
Received in revised form 22 May 2008
Accepted 30 August 2008
Available online 30 September 2008

Keywords:
Clustering
Markov processes
Performability
Queuing theory
High performance computing

ABSTRACT

Beowulf clusters are very popular because of the high computational power they can provide at reasonably low costs. However, the most pressing issues of today's cluster solutions are the need for high availability and performance. Cluster systems are clearly prone to failures. Even if cover is provided with some probability *c*, there would be reconfiguration and/or rebooting delays to resume the operation following a failure. In this paper, the performability modelling of both typical and highly available Beowulf multiprocessor systems is presented. The models developed provide a large degree of flexibility to evaluate the performability of typical and highly available Beowulf cluster systems.

1. Introduction

A Beowulf cluster is a multi-server system that consists of a tightly connected network of computers. These computers are dedicated to simultaneously provide service to incoming job requests. The idea behind this sort of parallel processing is to provide high performance, and reduce the time required to perform complex computations, by sharing the necessary work among the cluster's nodes. The resulting cluster is used for high performance computing (HPC). Until recently, the computational power generated for parallel processing has been prohibitively expensive on commercial supercomputers such as Cray, and not many organizations can afford to have them. Beowulf-style clusters allow the power of multiprocessor computing to be available for parallel or simultaneous processing at much lower costs [1,47].

A typical Beowulf cluster generally consists of two types of nodes: a head node server and multiple identical client nodes. The head node is responsible for serving user requests and distributing them to clients via scheduling/queuing software. Clients or identical nodes are normally dedicated to computation. Since the head node is responsible for the organization and distribution of jobs, the clients cannot compute if the head node is not operative [4,5,35,36]. Because of this single head node setup, clusters are vulnerable, as the head node represents a single point of failure affecting the availability of the cluster. Furthermore, the head node represents a single point of control [36]. This severely limits access to healthy identical nodes in case of head node failures. It is possible to introduce redundancy via a backup node for the head node of such systems. Systems with backup nodes are called highly available Beowulf systems [27,34]. In addition, in Beowulf clusters, the head node may or may not participate in computations, depending on the structure of the cluster.

* Corresponding author. Tel.: +44 0 7917724516.
E-mail address: eever@bradford.ac.uk (E. Ever).

1569-190X/$ - see front matter
doi:10.1016/j.simpat.2008.08.016

Multi-server system models are useful to model multiprocessor systems [28,45,46], nodes in communication networks, and flexible machine shops [7,18,38,42,43] in a manufacturing environment. Beowulf clusters are widely used all over the world. However, performance modelling of typical and highly available Beowulf clusters with one head and several identical processors has not been considered together with breakdowns and repairs. It is possible to extend various methodologies proposed in the literature to obtain an analytical model for the performability evaluation of Beowulf clusters. Such systems are prone to break-downs. To combat this, it is essential that effective repair strategies are used. Also in many multi-server systems, some delays are encountered when a failed processor is being mapped out of the system. Depending on the nature of failures, the system becomes non-operational (during the process of mapping out a failed processor) for a relatively shorter reconfiguration delay if the fault is covered and a longer rebooting delay if the fault is uncovered [46].

In this paper we develop approaches for performability evaluation of Beowulf multiprocessor systems with reconfiguration and rebooting delays, by suitably modelling their operation as quasi birth and death (QBD) processes. In this study, both typical and highly available systems, with computing/non-computing head nodes are considered. Some earlier works leading to the developments presented in this paper are in [8,11,14–16,21]. The models developed are applicable for systems with various diverse characteristics. Simulation results are also presented together with results obtained from analytical solutions.

The paper is organised as follows: the next section presents computing paradigms and Beowulf clusters followed by a section on modelling a Beowulf Cluster with breakdowns and repairs as a QBD process. The section on modelling reconfiguration and rebooting delays in Beowulf multiprocessor systems deals with typical Beowulf clusters with breakdowns, repairs, and reconfiguration and rebooting delays. Numerical results for the performability measures of typical Beowulf clusters are presented considering unbounded as well as bounded queuing facilities. This is followed by discussions on existing techniques to achieve high availability and models are developed for such highly available Beowulf clusters. Numerical results are presented for highly available systems with bounded as well as unbounded queuing capacities. The performability measures of typical and highly available Beowulf clusters are compared. Effects of using hot or cold standby replacement techniques are analysed under various conditions. Relevant conclusions are drawn.

2. Computing paradigms and Beowulf clusters

In computing clusters, arriving jobs can either be served independently (job-level parallelism) or split between available processors. The latter introduces overhead in terms of communication between processes contributing to the execution of the same job. This may have implications on system performance. Applications can be classified as coarse grained (i.e., $t_{\text{computation}} \gg t_{\text{communication}}$ where t represents time), medium grained (i.e., $t_{\text{computation}} > t_{\text{communication}}$), and less fine grained (i.e., $t_{\text{computation}} = t_{\text{communication}}$). The cost of the overhead changes between negligible to significant depending on the application [6]. Furthermore, in case of high throughput applications, when number of jobs in the system is greater than number of processors, job-level parallelism is preferable [29]. This section addresses the issue of job dependencies in Beowulf clusters and summarizes performance and availability studies reported.

2.1. Job dependencies in Beowulf clusters

Beowulf Computing Clusters can be employed for coarse grained applications (e.g. Monte Carlo calculations, statistical simulations, and data field explorations), medium grained applications (problems on a lattice) and less fine grained applications (cosmology, and molecular dynamics applications with long range interactions) [5,6]. For coarse and medium grained applications a computing paradigm called "embarrassingly parallel" computing is used. In this case, the computational work tends to consist of running a series of more or less independent jobs [5,17].

In [23,24], benchmark results are presented for various parallel computing environments, including, inter-process communication time for Beowulf cluster systems. The results clearly show that the time required for communication becomes negligible compared to the computation time especially when small scale multi-server systems are used.

Beowulf multi-server systems are commonly used in high throughput applications as well. In these applications the basic aim is not to maximise the performance of each individual job by using as many processors as possible, but rather to maximize the *throughput* of the compute resource [3,29]. Condor is an example of queuing and scheduling packages that allow a user to easily divide tasks to compute farms and to various extents balance the resource loads [3]. This package is being effectively used together with Beowulf multi-server systems [29,44]. Condor supports distributed job stream resource management emphasizing capacity or throughput computing. It schedules *independent* jobs on cluster nodes to handle large user workloads and provides many options in scheduling policy [44].

Beowulf clusters can also be used in grid computing [19,20,26,30]. Beowulf HA-OSCAR supports network file systems as well as web cluster facilities [4,27]. Both implementations are known with the use of job-level parallelism. This work is applicable to such systems with medium/coarse grained or high throughput applications as well as grid computing, HA-OSCAR, and small scale Beowulf systems.

2.2. Existing performance and reliability studies

As far as the authors are aware, there are no performability studies presented for Beowulf multi-server systems. Pure performance studies and availability studies are performed separately. In [37] benchmarking results are used together with