Computers and Electrical Engineering 39 (2013) 690–696

Contents lists available at SciVerse ScienceDirect

Computers and Electrical Engineering

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

Static and dynamic job scheduling with communication aware policy in cluster computing ☆

A. Neela Madheswari [a,*], R.S.D. Wahida Banu [b]

[a] Department of Information Technology, KMEA Engineering College, Aluva, India
[b] Government College of Engineering, Salem, India

ARTICLE INFO

Article history:
Available online 1 March 2013

ABSTRACT

Parallel jobs submitted to processors should be efficiently scheduled to achieve high performance. Early scheduling strategies for parallel jobs make use of either space-sharing approach or time-sharing approach. The scheduling strategy proposed in this work, makes use of both the policies for parallel jobs while scheduling under clusters. Static and dynamic scheduling algorithms were developed for communication intensive jobs. The algorithms are used to handle different types of jobs such as serial, parallel and mixed jobs. For performance evaluation, the workload from Grid5000 platform is considered. The main objective is to achieve performance and power improvement. The dynamic scheduling algorithm with communication aware policy gives better performance when compared to static scheduling algorithm that is tested under the given workload.

1. Introduction

Parallel job scheduling is an important topic in the field of research. The main issue is how to share the available processors from the existing parallel computing environment to the tasks that are submitted by the users or processes. There are different parallel computing environments: (i) the parallel computers in which the systems come under are desktops and laptops with multi-core chips, (ii) the grids in which large scale heterogeneous distributed shared computing environment, (iii) the web servers, in which large scale latency sensitive online services are provided and (iv) the virtualization, in which the resource management is performed inside and among multiple virtual machines [1]. It is easily understood that the parallel computing environments are existing and rather continue to aid future technologies.

Clusters offer considerable computational power, which can be used to solve problems with large computational requirements. Proper scheduling is fundamental to performance in distributed systems. The most important aspect of a distributed system is the scheduling algorithm. The scheduling algorithm is responsible for the allocation of processors to jobs and also determines the order in which jobs will be executed on processors.

Scheduling of parallel jobs on parallel machines has a significant impact on the system utilization. A good task scheduling method is needed to reduce the total time taken for job execution. Scheduling algorithms can be classified into static and dynamic. The static algorithms are in need of some prior information before scheduling the jobs and dynamic algorithms can execute the jobs in runtime without the prior information cost.

Parallel jobs consist of independent tasks which can be executed on any processor and in any order or independent group of tasks or tasks which need to frequently communicate with each other – they start essentially at the same time and execute

☆ Reviews processed and proposed for publication to Editor-in-Chief by Guest Editor Dr. Sabu M. Thampi.
* Corresponding author.
E-mail addresses: neelu.madheswari@gmail.com, neela.madheswari@gmail.com (A. Neela Madheswari), drwahidabanu@gmail.com (R.S.D. Wahida Banu).

0045-7906/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compeleceng.2013.01.008

for the same amount of time. In this work, the communication intensive jobs are considered for parallel jobs. A new scheduling algorithm is developed that utilizes both time-sharing and space-sharing policies as well as to meet the requirements of communication intensive jobs by developing the communication aware policy. It is the extension of the work [2], in which the static algorithms are explained and this work proposed the dynamic scheduling algorithm for scheduling different kinds as well as different mix of jobs in clusters.

The plan of the paper is as follows: Section 2 presents the motivation for this work. Section 3 describes the system set up for experimental evaluation and the workload considered for job scheduling towards this work and explains about the communication aware policy. Section 4 describes the different types of scheduling algorithms developed and section 5 discusses the experimental output. Section 6 gives the detailed conclusion for this work. In this paper, processor is also denoted as node.

2. Motivation

While job scheduling is considered, two common approaches used namely (i) time-sharing and (ii) space-sharing. But when these approaches are used separately, the performance level is lagging. If time-sharing approach alone is used, the jobs may be slowed down due to communication behavior [3] and, if space-sharing approach alone is used, the jobs may be affected by a process called blockade situation [4], in which the shortest jobs are easily blocked by longest jobs and thus the average response time of shortest jobs may be high. Context switching can be more, if the communication-intensive jobs are scheduled to different processors at different times which are also an important parameter of reducing processor utilization [5]. If communication-intensive jobs are not scheduled properly, process thrashing may occur [6].

For running parallel jobs at the same time, two scheduling strategies are generally considered namely: (i) gang scheduling and (ii) batch scheduling [7]. Both of them are suitable for batch jobs and not for communication-intensive jobs. Batch jobs do not need response from the user, process or from I/O systems. But it is not the same case for communication-intensive jobs. There are numerous solutions for gang scheduling [8–13]. Gang scheduling performs poorly for communication-intensive numerical applications [14]. To avoid these problems, a new scheduling algorithm is introduced for both static and dynamic scheduling approaches. The performance of static scheduling with dynamic scheduling strategies are evaluated by finding the wait time of the jobs and idle time of the nodes for different mixtures of jobs.

There are a number of works which focus on how the processes are allotted to different processors simultaneously. A novel approach to co-scheduling packets, tasks and CPU speeds in reservation based-environments is proposed with the goal to meet packet deadlines. This approach proposed for wireless real-time systems, whereas this proposed work focuses for clusters [15]. A dynamic scheduling strategy for multi-core processor design was proposed in [16–18]. A novel algorithm for mapping parallel applications on multicore systems using Servet is proposed in [19]. A scheduler of master-slave real-time operating system to manage threads running for the distributed multicore systems is presented in [20], but this work focuses on cluster with homogenous single core processors.

3. System setup

The system model for this work is explained in section 3.1 and the workload consideration is given in section 3.2. The policy that the system must follow for scheduling the parallel job is named as communication aware policy and is described in section 3.3.

3.1. System model

The number of jobs to be submitted to the scheduler is considered as m. The scheduler node N_0 contains the details of all the jobs, which are arranged in a queue. Each parallel job is composed of a set of dependent tasks. There are totally N_{n+1} number of nodes. The node N_0 is the actual scheduler or head node which is responsible for assigning the work or job to the worker nodes from N_1 through N_n. The scheduling algorithm is implemented in the head node, which contains two queues for storing the incoming jobs. One of the queues is used to store the serial jobs and the other is used to store the parallel jobs. The system for the computer cluster assumed to be satisfied for the following conditions:

a. There are N_{n+1} number of nodes and they are fully connected.
b. Communication protocols are error-free and the communication cost of any two nodes is the same as any other two nodes.
c. Communication and computation can be executed simultaneously. This can be accomplished by following the non-blocking send and non-blocking receive protocol for communication purpose.
d. The messages are assumed to be sent to the receiving job at the time of completion of originating job and the receiving job cannot begin execution until the message received.
e. The jobs submitted to the system are assumed to submit with the type of job details i.e. whether the job belongs to serial or parallel.
f. Before scheduling a particular job, the system is assumed to know the runtime of that job.