An Adaptive Scoring Job Scheduling algorithm for grid computing

Ruay-Shiung Chang *, Chih-Yuan Lin, Chun-Fu Lin

Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien, Taiwan

ARTICLE INFO

ABSTRACT

When human culture advances, current problems in science and engineering become more complicated and need more computing power to tackle and analyze. A supercomputer is not the only choice for solving complex problems any more as a result of the speed-up of personal computers and networks. Grid technology, which connects a number of personal computer clusters with high speed networks, can achieve the same computing power as a supercomputer does, also with a lower cost.

However, grid is a heterogeneous system. Scheduling independent tasks on it is more complicated. In order to utilize the power of grid completely, we need an efficient job scheduling algorithm to assign jobs to resources in a grid.

In this paper, we propose an Adaptive Scoring Job Scheduling algorithm (ASJS) for the grid environment. Compared to other methods, it can decrease the completion time of submitted jobs, which may compose of computing-intensive jobs and data-intensive jobs.

© 2012 Elsevier Inc. All rights reserved.

1. Introduction

When science and technology advance, the problems encountered become more complicated and need more computing power. In contrast to the traditional notion of using supercomputers, grid computing is proposed. Distributed computing supports resource sharing. Parallel computing supports computing power. Grid computing aims to harness the power of both distributed computing and parallel computing. The goal of grid computing is to aggregate idle resources on the Internet such as Central Processing Unit (CPU) cycles and storage spaces to facilitate utilization. The Search for Extra-Terrestrial Intelligence (SETI) experiment [31] is an early application of grids. The data Trans-Atlantic Grid project (TAG) [32] constructs a large-scale intercontinental grid testbed which focuses on issues of advanced networking and interoperability between these intercontinental grid domains, hence extending the capabilities of each and enhancing the worldwide program of grid development. In implementation, Globus Toolkit [33] is an open source and a fundamental enabling technology for grid. The latest version of Globus Toolkit is Globus Toolkit 5.2.0.

Grid can achieve the same level of computing power as a supercomputer does, but at a much reduced cost. Grid is like a virtual supercomputer. However, we need to consider about many conditions such as network status and resource status because the members of grid are connected by networks. Grid is also a heterogeneous system. Scheduling independent tasks on it is more complicated. In order to utilize the power of grid computing completely, we need an efficient job scheduling algorithm to assign jobs to resources. This paper focuses on the efficient job scheduling considering the completion time of jobs in a grid environment.

General task scheduling is an NP-Complete problem [17] and is an integral part of parallel and distributed computing [11]. How to schedule task in a grid environment efficiently is a new challenge because grid is a distributed and heterogeneous system. To shorten completion time and enhance the system throughput is the purpose of a job scheduling

---

* Corresponding author.
 E-mail address: rschang@mail.ndhu.edu.tw (R.-S. Chang).

algorithm. Because the status of grid environment may change at any time, the traditional job scheduling algorithm, e.g. "First Come Fist Serve" (FCFS), "Fist Come Last Serve" (FCLS), etc., may not adapt to the dynamic grid environment well.

This paper proposes a new framework and scheduling algorithm to decrease job's completion time in a grid environment. Computing intensive jobs and data intensive jobs are handled differently, reflecting the real time grid situations. We assign a new job to a resource depending on the result in the past job scheduling. We select the most appropriate resource for the current job. Local update and global update are used to get the newest status of resources in Grid environment. According to the local update and global update results, we can schedule jobs more dynamically and appropriately.

A new gauge structure for the current situation and a new job scheduling algorithm is proposed. The algorithm is called Adaptive Scoring Job Scheduling (ASJS) algorithm. We compare ASJS with Ant Colony Optimization (ACO) [21], Most Fit Task First scheduling algorithm (MFTF) [18] method and random selection method in the experiments. According to the results of experiments, ASJS is capable of decreasing the completion time of jobs better than other job scheduling algorithms mentioned above.

In recent years cloud computing has become an important part of computer systems [1,2]. Its centralized data center approach is different from grid computing approach. Its transaction based processing is also different from the batch processing of grids. That is, scheduling for thousands of processors in a data center is basically focused on response time and load balance. Scheduling algorithms for grid computing may not be able to be directly applied. However, when performing backup or load distributing between data centers, our scheduling algorithm could be a very useful reference [8].

The remainder of this paper is organized as follows. Section 2 gives an overview of previous work about job scheduling in grid environment. Section 3 introduces the framework and job scheduling algorithm we propose. The implementation and experiments are given in Section 4. Finally, Section 5 concludes the paper and proposes some future work.

2. Related work

In the literature, a lot of scheduling algorithms were proposed in the past. Some of them can be used in grid environment with appropriate modifications. The scheduling algorithms used in grid environment are most heuristics and are categorized into two types: batch mode and on-line mode [11]. We will introduce some scheduling algorithms of these two different types in the following.

2.1. Batch mode heuristic scheduling algorithms

Jobs are queued and collected into a set when they arrive in the batch mode. They will be scheduled afterwards by the scheduling algorithm. Batch mode heuristic scheduling algorithms are more appropriate for the environment with the same type of resources.

First-Come, First-Served Scheduling Algorithm (FCFS) is the simplest algorithm for job scheduling. Jobs are executed according to the sequence of job submitting. The second job will be executed when the first job is done, and therefore FCFS has a serious problem called convoy effect [16]. The convoy effect will happen when there is a job with large workload in the front of the job sequence. All other small workload jobs have to wait until the big one finishes.

Fastest Processor to Largest Task First Scheduling Algorithm (FPLTF) is a good representative for Bag-of-Tasks applications [14]. The strategy of the FPLTF scheduling algorithm is to schedule jobs according to the workload of jobs and computing power of resources. In Min–min scheduling algorithm [11], each job will be always assigned to the resource which can complete it earliest in order to spend less time completing all jobs. The Max–min scheduling algorithm [11] is similar to Min–min scheduling algorithm. It gives the highest priority to the job with the maximum earliest completion time.

Fairness is the key idea of the Round Robin (RR) scheduling algorithm. Workqueue (WQ) and Workqueue with Replication (WQR) Scheduling algorithm [13] is a knowledge-free scheduling algorithm. That is, the strategy of WQ scheduling algorithm is to schedule jobs without any information. Tasks are selected in a random order and sent to the resource, as soon as they become available. The WQR scheduling algorithm increases the chance of using the resource with high performance to execute jobs by adding replication into the WQ scheduling algorithm. In WQR scheduling algorithm, each task will be replicated a predefined numbers of times and transfer to the available resources. If there is a resource which completes the job mentioned above, all other replications which are executed by other resources will be canceled. Although the performance of the WQR scheduling algorithm is not bad, it wastes extra computing power on executing replications and takes lots of time to transfer the replications of jobs when the size of jobs is large.

2.2. On-line mode heuristic scheduling algorithms

Jobs are scheduled as soon as it arrives. Because a grid environment is heterogeneous with different types of resources, on-line mode heuristic scheduling algorithms are more appropriate for grid environment.

Dynamic FPLTF Scheduling Algorithm (DFPLTF) [11] is based on FPLTF scheduling algorithm and is modified to make the FPLTF scheduling algorithm more adaptive for grid environment. Most Fit Task First Scheduling Algorithm (MFTF) [18] mainly attempts to assign the most suitable resource to the task by a value called fitness. The range of fitness is from 100,000 to 0 and is defined as: