Title	分散環境における可分的ワークロードのスケジュール方法
Author(s)	Loc, Nguyen The
Citation	
Issue Date	2007-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/3566
Rights	
Description	Supervisor:教授 片山　卓也 - Professor Takuya Katayama, 情報科学研究科, 博士

Abstract of Doctoral Dissertation

Nguyen The Loc

Foundations of Software laboratory, School of Information Science
Japan Advanced Institute of Science and Tecnology (JAIST)

Divisible Load is a kind of workload that can be divided into arbitrary, independent chunks. By definition, Divisible Load has the characteristic that it can be arbitrarily partitioned into any number of load fractions. We can see this kind of workload in many domains of science and technology such as:

- Protein sequence analysis
- Simulator of Cellular Micro physiology
- Parallel and Distributed Image Processing, Video Processing, and Multimedia

Because of total workload may be as much as a single computer can not execute itself, so we have to share the total workload between many available workstations in distributed environments such as Grids. At this point, a problem, called scheduling problem, arises to pose the following question: how to divide the total workload, residing at a computer called master, into many parts and assign them to computers of the Grid, hereafter called workers, so that the execution time (makespan) is minimum.

The main issue of scheduling process is to find an optimal division of total workload into workers. Up to now many approaches have been proposed. A simple solution, called *Single-Round*, divides the workload in as many part as workers, then sends each part to appropriate worker. Because each worker receives its load only one time, this method is named Single Round. Another approach, called *Multi-Round*, splits the overall process into many consecutive rounds. In each round, the master delivers chunks to each of workers in turn.

One apparent shortcoming in many scheduling algorithms that exist in the literature is the abandon of designing a solid selection policy for generating the best subset of available workers. Part of the reason is that the main focus of these algorithms is confined to the LAN environment, which makes them not perfectly suitable for a WAN environment such as the Grid. In the Grid, resource computing (workers) join and leave the computing platform dynamically. Unlike other algorithms, we cannot assume in the Grid that all available resources, which may be in thousands, must participate in the scheduling process. The more recent algorithms very tersely allude to this problem by proposing primitive intuitive solutions that are not back up by any analytical model.

In the first part of this dissertation, we propose a new scheduling algorithm, MRRS (inspired by an existing algorithm called UMR), which is better and more realistic. MRRS is superior to UMR with respect to two aspects. First, unlike UMR that relies primarily in its computation on the CPU speed, MRRS factors in several other parameters, such as bandwidth capacity and all types of latencies (computation and communication) which renders the MRRS a more realistic model. Second, the MRRS is equipped with a worker selection policy that finds out the best subset of workers. As a result, our experiments show that our MRRS algorithm outperforms previously proposed algorithms including the UMR.

However, all of above approaches assume that computational resources at workers are dedicated. This assumption renders these algorithms impractical in distributed environments such as Grids where computational resources are expected to serve local tasks in addition to the Grid tasks. The inevitable variation of workers' power in the Grid embodies a non-trivial challenge for scheduling (split and distribute) workloads to workers.

The second purpose of this dissertation is to develop an efficient multi-round scheduling method for non-dedicated environments such as Grids. In order to find the optimal division of workload in each round, we need to forecast, as accurate as possible, the available CPU power of each worker before the division happens. We develop a performance model to represent a worker's activity with respect to processing local and external tasks. This model helps us to estimate the computing power of a worker under the fluctuation of number of local and Grid applications in the system. Based on this model, we propose a new strategy for predicting the computing power of processors. After that we design a dynamic scheduling algorithm incorporates the performance model and the prediction strategy into the static algorithm MRRS that mentioned above.

As an alternative method we apply an existing prediction algorithm, Mixed Tendency-Based Prediction, in developing a new dynamic scheduling algorithm. The Mixed Tendency-Based Prediction is integrated into the static algorithm MRRS in order to partition the workload in non-dedicated environments.

At last, we describe the experiments for comparing between proposed dynamic and static scheduling algorithms as well as for comparing the proposed algorithms with the existing scheduling algorithm.