Chapter 4
Dynamic Dependent Tasks Assignment for Grid Computing

Meriem Meddeber
University of Mascara, Algeria

Belabbas Yagoubi
University of Oran, Algeria

ABSTRACT

A computational grid is a widespread computing environment that provides huge computational power for large-scale distributed applications. One of the most important issues in such an environment is resource management. Task assignment as a part of resource management has a considerable effect on the grid middleware performance. In grid computing, task execution time is dependent on the machine to which it is assigned, and task precedence constraints are represented by a directed acyclic graph. This paper proposes a hybrid assignment strategy of dependent tasks in Grids which integrate static and dynamic assignment technologies. Grid computing is considered a set of clusters formed by a set of computing elements and a cluster manager. The main objective is to arrive at a method of task assignment that could achieve minimum response time and reduce the transfer cost, inducing by the tasks transfer respecting the dependency constraints.

INTRODUCTION

A *computational Grid* (Foster & Kesselman, 2004) is a hardware and software infrastructure that provides consistent pervasive and inexpensive access to high end computational capacity. An ideal grid environment should provide access to all the available resources seamlessly and fairly (Saleh, Deldari, & Dorri, 2008). *Grid computing* originated from a new computing infrastructure for scientific research and cooperation and is becoming a mainstream technology for large-scale resource sharing and distributed system integration. Current efforts towards making the global infrastructure a reality provide technologies on both grid services and application enabling (Cao, Spooner, Jarvis, & Nudd, 2005).

DOI: 10.4018/978-1-4666-2065-0.ch004

A *task* is defined to be a program segment that can be individually scheduled. A *grid computing element* is defined to be any processor that can receive tasks from a central scheduler and may be a single processor node or one of the processors within a multi-processor node. The problem of obtaining an optimal matching of tasks to machines in any distributed system is well known to be *NP-hard* even when the tasks are independent. The problem is much more difficult when the tasks have dependencies because the order of task execution as well as task-machine pairing affects overall completion time (Boyer & Hura, 2005).

A *precedence relation* from task *i* to task *j* means that *j* needs data from *i* before being started. If these two tasks are not assigned to the same computing element, a delay c_{ij} must be considered between the completion of *i* and the beginning of *j* to transfer the data.

Dynamic tasks assignment assumes a continuous stochastic stream of incoming tasks. Very little parameters are known in advance for dynamic tasks assignment. Obviously, it is more complex than *static tasks assignment* for implementation, but achieves better throughput. Also it is the most desired because of the application demand (Vidyarthi, Sarker, Tripathi, & Yang 2009).

In this paper, we propose a hybrid assignment strategy of dependent tasks in Grids which integrated static and dynamic assignment technologies. This strategy meets the following objectives:

1. Reducing, whenever possible, the average response time of tasks submitted to the grid,
2. Respecting the constraints of dependency between tasks, and,
3. Reducing communication costs by using a static tasks placement based on the connected components algorithm to minimize the delay c_{ij} between task *i* and task *j* and by favoring a dynamic tasks placement within the cluster rather than the entire grid.

The rest of this paper is organized as follows. We begin with the overview of some related works in Section 2. Section 3 describes the grid computing topology. In section 4, we present the tasks assignment problem. In section 5, we present Tasks assignment in Grid computing environments. In Section 6 we will presents our system model. Section 7 describes the main steps of the proposed assignment strategy. We evaluate the performance of the proposed scheme in Section 8. Finally, Section 9 concludes the paper.

RELATED WORK

There have been many heuristic algorithms proposed for the static and dynamic tasks assignment problem. Many of these algorithms apply only to the special case where the tasks are independent i.e. there are no precedence constraints (Kamalam, Maharajan, & Maheish Sundhar, 2010; Leal, Huedo, & Llorente, 2009; Jiang, Baumgarten, Zhou & Jin, 2009; Salcedo-Sanz, Xu, & Yao, 2006; Maheswaran, Ali, Siegel, Hensgen, & Freund, 1999). Many heuristic algorithms have been proposed for static scheduling of dependent tasks where task precedence constraints are modeled as a directed acyclic graph (DAG). Qu, Soininen, and Nurmi (2007) target dependent task models and propose three static schedulers that use different problem solving strategies. The first is a heuristic approach developed from traditional list based schedulers. It presents high efficiency but the least accuracy. The second is based on a full-domain search using constraint programming. It can guarantee to produce optimal solutions but requires significant searching effort. The last is a guided random search technique based on a genetic algorithm, which shows reasonable efficiency and much better accuracy than the heuristic approach. Boyer and Hura (2005), propose a non-evolutionary random scheduling (RS) algorithm for efficient matching and scheduling of inter-dependent tasks in a DHC system. RS is a succession of randomized

Related Content