A Performance-Based Approach to Dynamic Workload Distribution for Master-Slave Applications on Grid Environments

Wen-Chung Shih[1], Chao-Tung Yang[2,*], and Shian-Shyong Tseng[1,3]

[1] Department of Computer and Information Science,
National Chiao Tung University,
Hsinchu 30010, Taiwan, R.O.C.
{gis90805, sstseng}@cis.nctu.edu.tw
[2] High-Performance Computing Laboratory,
Department of Computer Science and Information Engineering,
Tunghai University,
Taichung 40704, Taiwan, R.O.C.
ctyang@thu.edu.tw
[3] Department of Information Science and Applications,
Asia University,
Taichung 41354, Taiwan, R.O.C.
sstseng@asia.edu.tw

Abstract. Effective workload distribution techniques can significantly reduce the total completion time of a program on grid computing environments. In this paper, we propose a dynamic performance-based workload partition approach for master-slave types of applications on grids. Furthermore, we implement two types of applications and conduct the experimentations on our grid testbed. Experimental results showed that our method could execute more efficiently than traditional schemes.

Keywords: Workload distribution, master-slave paradigm, grid computing, parallel loop scheduling, data mining, Globus, NWS.

1 Introduction

As inexpensive personal computers and Internet access become available, much attention has been directed to grid computing [2, 3, 4, 8, 9, 14, 21, 22, 23, 24, 25, 33]. The basic idea of grid computing is to share the computing and storage resources all over the world via Wide Area Networks. In this way, computational jobs can be distributed to idle computers far away, probably in other countries. Moreover, remote data can be accessed for large-scale analysis.

Master-slave paradigms are commonly utilized to model the task dispatching processes in parallel and distributed computing environments [38]. This model designates one computing node as the master, which holds a pool of tasks to be dispatched to other slave nodes. Divisible Load Theory (DLT) [1, 16, 17, 30] addresses

* Corresponding author.

Y.-C. Chung and J.E. Moreira (Eds.): GPC 2006, LNCS 3947, pp. 73 – 82, 2006.

the case where the total workload can be partitioned into any number of independent subjobs. This problem has been discussed in the past decade, and a good review can be found in [15]. In [19], a data distribution method was proposed for host-client type of applications. Their method was an analytic technique, and only verified on homogeneous and heterogeneous cluster computing platforms. In [20], an exact method for divisible load was proposed, which was not from a dynamic and pragmatic viewpoint as ours.

In this paper, we focus on the problem of dynamic distribution of workload for master-slave applications on grids. We implement two types of applications, Parallel Loop Self-Scheduling [18, 27, 34, 35, 36] and Association Rule Data Mining, with MPI directives, and execute them on our grid testbed. Experimental results show that effective workload partitioning can significantly reduce the total completion time.

Our major contributions can be summarized as follows. First, this paper proposes a new performance-based algorithm to solve this dynamic workload distribution problem. Second, we implement the algorithm and apply it to both loop self-scheduling and data mining applications on our grid testbed. Consequently, experimental results show the obvious effectiveness of our approach. To the best of our knowledge, this is the first paper to consider dynamic workload distribution within a program on grid environments.

Our previous work [31, 32] presents different heuristics to the parallel loop self-scheduling problem. This paper generalizes their main idea and proposes to solve the dynamic workload distribution problem. This approach is applied to both the parallel loop self-scheduling application and the association rule mining application. There have been a lot of researches of parallel and distributed data mining [12, 13, 26, 37]. However, this paper focuses on workload distribution, instead of proposing a new data mining algorithm.

The rest of this paper is organized as follows. Our approach is proposed in Section 2. Then, Section 3 shows the experimental results on our grid testbeds. Finally, we conclude this paper in Section 4.

2 Our Approach

Our performance-based approach is based on the estimated performance of each slave node and each link to distribute the corresponding workload. In this section, the system model and the concept of performance ratio are explained first. Then, we present the heuristics which the algorithm is based on. Finally, the algorithm is described.

2.1 System Model

Our system model and cost model are extended from the framework in [15]. The master-slave model for a grid is represented by a star graph $G = \{P_0, P_1, \ldots, P_n\}$, as shown in Figure 1. In this graph, P_0 is the master node and the other n nodes, P_1, …, P_n, are slave nodes. In addition, there is a virtual communication link L_i connecting the master node and the slave node P_i.