International Journal of Innovative
Computing, Information and Control
Volume **5**, Number **6**, June **2009**

ICIC International ©2009 ISSN 1349-4198
pp. **1665–1679**

A DEADLINE-BASED TASK SCHEDULING WITH MINIMIZED MAKESPAN

Li-Ya Tseng[1], Yeh-Hao Chin[1] and Shu-Ching Wang[2]

[1]Department of Computer Science
National Tsing Hua University
101, Section 2, Kuang-Fu Road, Hsinchu City, 30013, Taiwan
d918305@oz.nthu.edu.tw; yhchin@cs.nthu.edu.tw

[2]Graduate Institute of Informatics
Chaoyang University of Technology
168, Jifong E. Rd., Wufong Township, Taichung County, 41349, Taiwan
Corresponding Author: scwang@cyut.edu.tw

Received December 2007; revised June 2008

Abstract. *Many scheduling heuristics have been dedicated to either HC (Heterogeneous Computing) or Grid computing for the purpose of minimizing makespan. However, both the number and the heterogeneities of computers found in Grid computing are more complex than in HC. Two additional scenarios are included in this study to enhance the completeness of the experiment and to more closely approach real Grid computing situations. This study simulates experiments in a dynamic environment since the nature of Grid computing is dynamic. Min-min heuristic is the benchmark of such kinds of task and computer scheduling problems. In this study, the Min-minII dynamic scheduling heuristic is proposed that utilizes task deadline and task assignment time to testify that Min-minII can outperform Min-min makespan.*
Keywords: Dynamic heuristic, Scheduler, Makespan, Grid computing

1. **Introduction.** Khokhar *et al.* defined a HC system [14] that includes heterogeneous machines, high-speed networks, interfaces, operating systems, communication protocols, and programming environments, all combined to improve ease of use and performance. A HC system employs high-speed interconnection networks to link the different properties of high-performance computers in order to execute a job [14]. Academic thought regarding Grid computing categorizes it as a type of HC [16,22]. Foster [27] proposed that Grid computing includes a three-point checklist: 1) coordinate resources that are not subject to centralization; 2) use standard, open, and general-purpose protocols and interfaces; and 3) deliver nontrivial qualities of service. Although a distributed system achieves collaboration depends on transmission networks, Grid computing is more compatible with lower speed transmission than HC in current applications; and can be effectively utilized in idle computers, such as SETI@home [1].

In a Grid computing system, it is necessary to divide a job into several independent tasks. For instance, the computers used in the SETI@home project can download a small portion of information from the server, analyze and complete operational tasks locally, and then transmit the results back to SETI via the Internet [1]. The application scheduler that assigns the relevant data and communications to the involved computers is ordered temporally and based on the rules of the scheduling policy in a Grid computing system [29]. Nevertheless, mapping for independent tasks in HC systems is a well-known