Future Generation Computer Systems 54 (2016) 132–143

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Multi-objective method for divisible load scheduling in multi-level tree network

Shamsollah Ghanbari [a,*], Mohamed Othman [a,b,**], Mohd Rizam Abu Bakar [a], Wah June Leong [a]

[a] Laboratory of Computational Science and Mathematical Physics, Institute For Mathematical Research, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor D.E., Malaysia

[b] Department of Communication Technology and Network, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor D.E., Malaysia

HIGHLIGHTS

- We have proposed a multi-objective divisible load scheduling method.
- The proposed method is able to estimate the actual computation rate of the processors.
- The proposed method improves the performance of divisible load when the processors cheat the algorithm.
- We investigate the effects of cheating problem on the total finish time.
- The proposed method has approximately 66% decrease in finish time in the best case.

ARTICLE INFO

Article history:
Received 1 September 2014
Received in revised form
3 March 2015
Accepted 15 March 2015
Available online 31 March 2015

Keywords:
Divisible load theory
Multi-objective method
Multi-level tree network
Divisible load scheduling

ABSTRACT

There is extensive literature concerning the divisible load theory. Based on the divisible load theory (DLT) the load can be divided into some arbitrary independent parts, in which each part can be processed independently by a processor. The divisible load theory has also been examined on the processors that cheat the algorithm, i.e., the processors do not report their true computation rates. According to the literature, if the processors do not report their true computation rates, the divisible load scheduling model fails to achieve its optimal performance. This paper focuses on the divisible load scheduling, where the processors cheat the algorithm. In this paper, a multi-objective method for divisible load scheduling is proposed. The goal is to improve the performance of the divisible load scheduling when the processors cheat the algorithm. The proposed method has been examined on several function approximation problems. The experimental results indicate the proposed method has approximately 66% decrease in finish time in the best case.

1. Introduction

The first published research concerning the divisible load theory appeared in 1988, [1,2]. Over the past two decades, the DLT has found a wide variety of applications in the area of distributed and parallel computing, e.g., linear algebra [3], image and vision processing [4,5], and data grid applications [6,7]. Moreover, the DLT was applied to a wide variety of interconnection topologies, including the daisy chain, bus, single-level tree, multi-level tree [8], hypercubes [9], three-dimensional meshes [10], k-dimensional meshes [11], and arbitrary graphs [12]. In addition, it has been applied in heterogeneous [13,14] and homogeneous platforms [15], grid-based job scheduling [16,17], and cloud-based job scheduling [18].

More recently, the other aspects of the DLT, e.g., limited memory [19], multi-criteria based divisible load scheduling [20], and incentive-based mechanism for the DLT [21] have been studied. A comprehensive review on the divisible load scheduling, including strategies, approaches, applications, and open problems can be found in [22].

* Corresponding author.
** Corresponding author at: Department of Communication Technology and Network, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor D.E., Malaysia

E-mail addresses: myrshg@gmail.com (S. Ghanbari), mothman@upm.edu.my (M. Othman).

http://dx.doi.org/10.1016/j.future.2015.03.015
0167-739X/

The DLT is based on the fact that, the load can be divided among parts, in which each part can be executed by an independent processor. The traditional DLT assumes that the processors report their true computation rates, i.e., they do not cheat the algorithm. In the real applications, the processors may cheat the algorithm. It means that the processors may not report their true computation rates. This issue was investigated by Thomas E. Carroll and Daniel Grosu in their research publications [21,23]. The results of their research indicate that the cheating reduces the performance of the divisible load model. In fact, the DLT only obtains its optimal performance if the processors report their true computation rates [21,23].

This paper focuses on the cheating problem and proposes a multi-objective divisible load method. The goal is to improve the performance of computing in the divisible load under the cheat processors. We use the adaptive strategy for this purpose. There is extensive literature concerning the adaptive divisible load method. The first adaptive strategy was introduced in [24], under the name of feedback strategy for divisible load allocation. Subsequently, adaptive approach was extended by other researchers [25–27].

However, the proposed method is a novel effort in both adaptive and cheat processing problems. We examine the proposed method on the multi-level tree (binary tree) network topology and investigate the effects of the multi-objective method on the total finish time (i.e., makespan), utility, and payment. The experimental results show that the proposed method is able to reduce the effects of processors cheating on the mentioned parameters. The rest of this paper is organized as the following sections: Section 2 provides some information about the problem and related works, Section 3 includes the preliminaries of this paper, Section 4 explains the proposed method and related algorithms, and Section 5 presents some experimental results to support the proposed method. We discuss the results in Section 6. Finally, Section 7 provides the conclusion.

2. Background

In general, the DLT assumes that the computation and communication can be divided into parts of arbitrary sizes, that these parts can be independently processed in parallel. The DLT assumes that initially amount V of load is held by the originator p_0. The originator does not do any computation. It only distributes the load into parts $\alpha_0, \alpha_1, \ldots, \alpha_m$ to be processed on worker processors $p_0, p_1, \ldots, p_m$. The condition for the optimal solution is that, all of the processors stop processing at the same time; otherwise, the load could be transferred from busy to idle processors to improve the solution time [28]. The goal is to calculate $\alpha_0, \alpha_1, \ldots, \alpha_m$ in the DLT timing equation.

2.1. Single-level tree network

The timing equation (i.e., closed form) for the divisible load scheduling under the single-level tree network topology can be depicted as the following equation:

$$T_j(\alpha) = \begin{cases} \alpha_j w_j & j = 0 \\ \sum_{r=1}^{j} \alpha_r z_r + \alpha_j w_j & j = 1, 2, \ldots, m \end{cases} \tag{1}$$

where $\alpha_0 + \alpha_1 + \cdots + \alpha_m = V$. Moreover, α_j can be calculated as the following equations:

$$\alpha_j = k_j \alpha_{j-1} \tag{2}$$

and

$$k_j = \frac{w_{j-1}}{z_j + w_j} \tag{3}$$

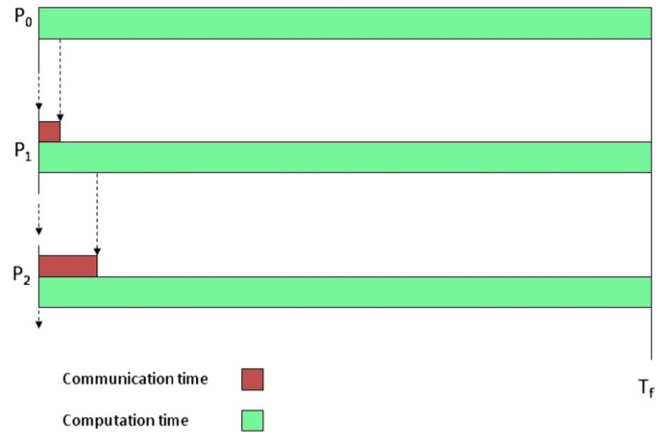

Fig. 1. Gantt chart-like timing diagrams for divisible load in a single level tree network.

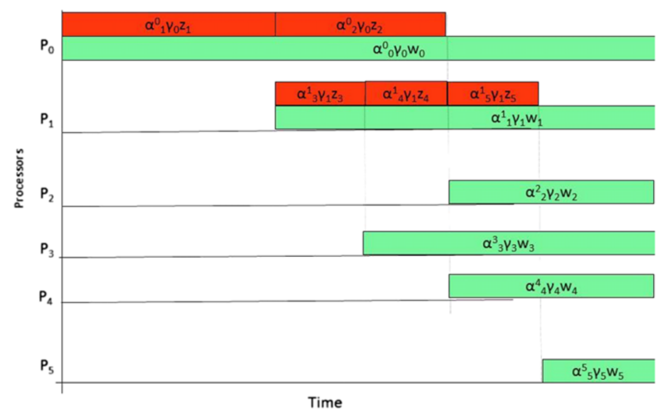

Fig. 2. Gantt chart-like timing diagrams for divisible load in a multi-level tree network.

and

$$\alpha_0 = \frac{V}{1 + \sum_{j=1}^{m} \prod_{r=1}^{j} k_r}. \tag{4}$$

The Gantt chart-like timing diagrams for this case can be depicted as Fig. 1.

2.2. Load allocation in multi-level tree network

In a multi-level tree, the load is distributed from top to the bottom, passing through each level. The optimal solution is obtained by traversing the tree from bottom to the top, replacing single level subtrees with single equivalent processors until total time is reduced to one processor, which is denoted by w_j^{eq} [21]. The Gantt chart-like timing diagram for multi-level tree divisible load is depicted in Fig. 2.

2.3. Related works

2.3.1. Processor cheating problem

The main idea of processor cheating refers to misreporting and time varying problems, which were investigated in respect of the divisible load scheduling in 1998, [29]. Subsequently, Thomas E. Carroll et al., focused on the application case of misreporting in divisible load scheduling. They proposed a strategyproof mechanism for the divisible load scheduling under various topologies, including the bus and multi-level tree network. In the