Applied Soft Computing 24 (2014) 500–510

Contents lists available at ScienceDirect

Applied Soft Computing

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

Hybrid real-coded genetic algorithm for data partitioning in multi-round load distribution and scheduling in heterogeneous systems

S. Suresh [a,*], Hao Huang [a], H.J. Kim [b]

[a] School of Computer Engineering, Nanyang Technological University, Singapore
[b] CIST, Korea University, Seoul, South Korea

ARTICLE INFO

Article history:
Received 11 July 2013
Received in revised form 2 May 2014
Accepted 28 July 2014
Available online 11 August 2014

Keywords:
Data partitioning
Scheduling
Genetic algorithm
Divisible loads
Parallel computing

ABSTRACT

Data partitioning and scheduling is one the important issues in minimizing the processing time for parallel and distributed computing system. We consider a single-level tree architecture of the system and the case of affine communication model, for a general m processor system with n rounds of load distribution. For this case, there exists an optimal activation order, optimal number of processors m^* ($m^* \leq m$), and optimal rounds of load distribution n^* ($n^* \leq n$), such that the processing time of the entire processing load is a minimum. This is a difficult optimization problem because for a given activation order, we have to first identify the processors that are participating (in the computation process) in every round of load distribution and then obtain the load fractions assigned to them, and the processing time. Hence, in this paper, we propose a real-coded genetic algorithm (RCGA) to solve the optimal activation order, optimal number of processors m^* ($m^* \leq m$), and optimal rounds of load distribution n^* ($n^* \leq n$), such that the processing time of the entire processing load is a minimum. RCGA employs a modified crossover and mutation operators such that the operators always produce a valid solution. Also, we propose different population initialization schemes to improve the convergence. Finally, we present a comparative study with simple real-coded genetic algorithm and particle swarm optimization to highlight the advantage of the proposed algorithm. The results clearly indicate the effectiveness of the proposed real-coded genetic algorithm.

1. Introduction

One of the primary objectives in the area of parallel and distributed systems is to partition and schedule the processing load among the processors, such that the processing time of the entire processing load is a minimum. Minimizing the processing time of the entire processing load involves the design of efficient data partitioning and scheduling algorithms. Design of efficient data partitioning algorithms for heterogeneous systems is more difficult because, the processors in the system may have different computation rate, and the communication links in the system may have different communication rate. One way to design efficient data partitioning algorithms is from the *Divisible Load Theory* [1,2] view point in 1988 and 1990. The theory is shown in [3] that "the divisible load theory offers a tractable and realistic approach to scheduling that allows integrated modelling of computation and communication in parallel and distributed computing systems". The research work on divisible load scheduling is presented in [4,5]. Divisible load theory has served as an effective modelling tool for data-intensive application in [6] and a large amount of work has been published for various network structures, including linear networks [7,63], bus networks [8,72], single and multi-level tree networks [9–11,22,64], mesh networks [12,13], hypercubes [14], nonlinear loads [29–31], static load balancing [24] and arbitrary graphs [15] with different constraints. The theory of divisible load distribution has applications in a wide range of areas such as compute vision [16], large scale data file processing [17], data intensive applications [18], query processing in database systems [19], scientific computing [20], video and multimedia applications [21,23], numerical computing [25], biomedicine and bioinformatics [26]. Divisible load distribution has applications in filtering of radio communications, encryption for secure communications, and coding for digital communications [27]. Aerospace applications include

* Corresponding author. Tel.: +65 67906185.
E-mail addresses: suresh99@gmail.com, ssundaram@ntu.edu.sg (S. Suresh), khj-@korea.ac.kr (H.J. Kim).

http://dx.doi.org/10.1016/j.asoc.2014.07.021
1568-4946/

satellite signal and image processing [28], radar and infrared tracking [29], and data reporting and aggregation and processing in wireless sensor networks [32].

Divisible load scheduling problem will be more difficult, when practical issues like the communication latencies are considered. For divisible load scheduling, the affine model of communication in which the communication latencies (start-up times) are included, is studied in [33–35], for both single and multi-round load distribution in heterogeneous systems. For a heterogeneous single-level tree network (master-slave system), load scheduling strategies are discussed and is also shown the methodology leads to a bandwidth centric scheduling strategy in [34]. The effect of start-up delays in multi-round load distribution is studied in [35]. A linear programming formulation for the divisible load scheduling with communication latencies is given in [5]. It is pointed out in [33], that this linear programming methodology is valid only if all the processors in the system are participating in the computation process. The computational complexity of the divisible load scheduling problem is presented in [36]. The problem of scheduling divisible loads in both linear and affine communication model is considered and the open problems in multi-round load distribution are brought out [37]. In [38], various distribution strategies have been studied including, multi-installment of load distribution. Further, in [39,40], scheduling strategies without knowledge of network resources are examined.

In [41], scheduling algorithms using DLT on heterogeneous systems by considering different types of variation of the perturbation parameters, larger problem sizes, and different types of application characteristics. In [42], the resource model and the job model are presented to achieve a minimum processing time. A network aware divisible load algorithm extends DLT to optimally allocate both computational and networking resource of an optical grid in [43]. In [44], an uniform framework for parallel image processing on heterogeneous network systems, Single Program Multiple Data (SPMD) tasks where each image processing task can be computed as multiple SPMD subtasks on various kinds of computing nodes. The uniform interface has been created for connect existing hardware and software resources to the user application, which makes complexity of parallel execution system transparent. In [45], GA-BT a genetic algorithm based peer selection optimization strategy for efficient content distribution in Bit-Torrent networks. GA-BT employs the divisible load theory to dynamically predict an optimal fitness value to speed up the convergence process in producing optimal or near optimal schedules in peer selection.

A method of partitioning divisible loads known as “multi-installment technique” (multi-round) is presented in [46]. In [46], a homogeneous system (all the processors are identical and all the communication links are identical) and a linear model for communication and computation are used. The theoretical aspect of multi-installment scheduling algorithm for multiple loads in bus network is illustrated in [47] In practical heterogeneous systems, the communication model is not linear and communication latencies (start-up delays) are to be included. These communication latencies play a major role in data partitioning and scheduling. The effect of communication latencies (start-up time) and its influence on the processing time is first studied in [48] for a linear network architecture of the distributed computing system. In this study, it is shown that the processing load assigned to processors beyond a certain number of processors is zero. In other words, only processors up to a certain number will participate in the computation process. The effect of start-up delays in communication and computation for a bus network is presented in [49,50] and in these studies also processors beyond a certain number are not included in the load distribution process. In [51], multi-installment technique is presented for homogeneous tree network with start-up delaying computation process. In [51], the effect of result collection in multi-installment scheduling algorithm for homogeneous network is presented. Also, a closed form expression for processing time is obtained using rational expansion theorem. In [52,53], the issue of scheduling is formulated as a linear programming framework. The major benefit of using this uniform multi-installment strategy is that, we get an approximate optimal number of installments required to process the load such that the processing time is a minimum. In [54], a novel algorithm called robust uniform multi-round (RUMR) algorithm is proposed. In RUMR algorithm, the uncertainty in the communication and computation times, which can be caused by the platforms (e.g., when resources are non-dependent) are handled by increasing the chunk size (load fraction size) in the initial rounds and decreasing the chunk size towards the end. In [55], Drozdowski and Lawenda analyze multiple divisible loads on a star-connected system consisting of homogeneous processors. This work left one open question of the problem NP membership. As it is impossible to show that this problem is in NP it is necessary to prove that communication pattern can at least be recorded in polynomial time.

In this paper, we consider a single-level tree (master-slave or star-shaped) architecture of the distributed computing system. We consider the case of affine communication model, for a general m processor system with n rounds of load distribution. For this case, there exists an optimal activation order, optimal number of processors m^* ($m^* \leq m$), and optimal rounds of load distribution n^* ($n^* \leq n$), such that the processing time of the entire processing load is a minimum. This is a difficult optimization problem because for a given activation order, we have to first identify the processors that are participating (in the computation process) in every round of load distribution and then obtain the load fractions assigned to them, and the processing time. There are 2^{mn} combinations of participating processors for every activation order. There are $m!$ activation orders are possible. Hence, we have to consider $m! * 2^{mn}$ solutions to obtain the optimal processing time.

So for this problem, we propose a real-coded genetic algorithm to search the set of participating processors, and the load fractions assigned to them in every round, such that the processing time is a minimum, for a given activation order. We also show that from the load fractions obtained for any activation order, we can search for a better activation order. Real-coded genetic algorithm uses hybrid crossover and mutation operators. unlike, normal real-coded genetic algorithm, the four crossover operators proposed in this paper always satisfy the equality constraints and hence the solution are always valid. In addition, zero mutation operator helps in identifying optimal activator order effectively. Using the solutions from proposed real-coded genetic algorithm, we propose a heuristic approach to find the optimal activation order, optimal number of processors m^* ($m^* \leq m$), and optimal rounds of load distribution n^* ($n^* \leq n$), such that the processing time of the entire processing load is a minimum.

It is known that the solution obtained from genetic algorithms cannot be guaranteed to be optimal; i.e., no formal proof of optimality. But, for a large class of difficult optimization problems, it has been shown that the genetic algorithm produces solution that are very close to the optimal or among the best available solutions.

This paper is organized as follows: in Section ‘Multi-round divisible load scheduling problem’, we present the multi-round load distribution problem and present some motivating numerical examples to show the combinatorial nature of the problem. Section ‘Real-coded genetic algorithm for obtaining load fractions’, presents the real-coded genetic algorithm describing the solution representation, genetic operators and other details. In Section ‘Simulation results’, we present numerical results and show that the genetic algorithm gives the optimal number of rounds, optimal number of processors and also the optimal activation order for the