A List-based Heuristic Algorithm for Static Task Scheduling in Heterogeneous Distributed Computing Systems

Hadi Gholami*
Department of Computer Engineering,
Ayatollah Amoli Branch,
Islamic Azad University,
Amol, Iran
gholamihd@gmail.com

Reza Zakerian
Department of Computer Engineering,
Faculty of Mahmoudabad,
Technical and Vocational University,
Mazandaran, Iran
zakerianr@gmail.com

***Abstract*— Executing complicated computations in parallel increases the speed of computing and brings user delight to the system. Decomposing the program into several small programs and running multiple parallel processors are modeled by Directed Acyclic Graph. Scheduling nodes to execute this task graph is an important problem that will speed up computations. Since task scheduling in this graph belongs to NP-hard problems, various algorithms were developed for node scheduling to contribute to quality service delivery. The present study brought a heuristic algorithm named looking ahead sequencing algorithm (LASA) to cope with static scheduling in heterogeneous distributed computing systems with the intention of minimizing the schedule length of the user application. In the algorithm proposed here, looking ahead is considered as a criterion for prioritizing tasks. Also, a property called Emphasized Processor has been added to the algorithm to emphasize the task execution on a particular processor. The effectiveness of the algorithm was shown on few workflow type applications and the results of the algorithm implementation were compared with two more heuristic and meta-heuristic algorithms.**

***Keywords*— *Distributed Computing; Makespan; Static Scheduling; Workflow Scheduling* .**

I. INTRODUCTION

Large applications (or programs) with large volumes of data and the need for fast processing are the topics that have been discussed in recent years. To this end, Heterogeneous Distributed Computing Systems (HTDCS) interconnected by high-speed networks are promising for faster computing processing. One of the important advantages of a distributed (multi-processor) computing system is the high speed of executing programs because a large application can use multiple processors (or computers) to run itself simultaneously by breaking itself down into smaller subtasks. In order to take advantage of the computing power of a multi-processor system as well as to minimize makespan (or schedule length), each subtask must be scheduled for processing in such a way that dependencies and priority constraints are maintained. Since the scheduling of multiprocessor systems is known as an NP-hard problem [1], numerous heuristic algorithms have been developed to minimize such goals as makespan or energy.

The Directed Acyclic Graph (DAG) is used to represent parallel applications with priority constraint relations between tasks [2]. Each node in this graph represents a task that has a numeric label indicating how long it takes for a processor to perform it and each edge has a weight that represents the cost of communication between tasks.

The solutions expecting to illuminate the problem are partitioned into two common categories: static scheduling and dynamic scheduling [3]. In the former, ordering is done at compile-time, suggesting that tasks are assigned to the processors before the program is run. In the latter, however, the order of tasks is determined when the program is run [4]. Heuristic algorithms that work for static scheduling can be classified into three categories: list-based algorithms, clustering-based algorithms, and task duplication algorithms [3]. List-based algorithms are commonly-used algorithms that produce high quality and cost-effective schedules. These types of algorithms have two main phases: the prioritization phase, which defines the priority for each task, and the processor selection phase, where a suitable processor, i.e. one that improves the objective function, is assigned to the task based on a policy. Clustering-based algorithms which have more time complexity than list-based algorithms also include clustering and merging phases [5]. Finally, in the task duplication algorithms, to decrease the waiting time of dependent tasks, the parent tasks are identically copied on more than one processor. These algorithms have a high time complexity [6].

The problem addressed in this study is the static scheduling of a task graph on heterogeneous processors to minimize the makespan in order to achieve a better performance in the execution of the program. In the present study, the scheduling problem is considered as a deterministic model, including the cost of executing each task, the cost of communication between tasks, as well as the graph of tasks that indicate the tasks and the priority of performing them. Hence, this study follows two methods in the prioritization phase. It also introduces a concept called Emphasized Processor which, forces tasks to run on a particular processor. It tries to balance the workload between the processors to improve algorithm's overall performance.

The study continued and organized as follows. In Section II the related works on list-based scheduling algorithms for heterogeneous distributed systems are reviewed. Section III presents the motive and the statement of the problem. In Section IV, the proposed algorithm is described, and Section V presents the simulation with performance evaluation. The paper ends with concluding remarks presented in Section VI.

II. RELATED WORKS

To develop an efficient algorithm, studies on the static task scheduling problem in HTDCS were investigated. The literature is replete with a lot of works on the static task graph algorithms. For instance, the paper [7] introduces one of the most famous list-based heuristic algorithm called

978-1-7281-1051-6/20/$31.00 ©2020 IEEE