Future Generation Computer Systems 55 (2016) 29–40

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Low-time complexity budget–deadline constrained workflow scheduling on heterogeneous resources

Hamid Arabnejad [a], Jorge G. Barbosa [a,*], Radu Prodan [b]

[a] LIACC, Departamento de Engenharia Informática, Faculdade de Engenharia, Universidade do Porto, Portugal

[b] University of Innsbruck, Institute of Computer Science, Innsbruck, Austria

HIGHLIGHTS

- A review of multiple QoS parameter workflow scheduling.
- A new multiple QoS algorithm with quadratic complexity for workflow scheduling.
- Similar performances of search-based algorithms in a small fraction of the time.
- Results for randomly generated graphs as well as for real-world applications.

ARTICLE INFO

Article history:
Received 28 January 2015
Received in revised form
2 June 2015
Accepted 30 July 2015
Available online 18 August 2015

Keywords:
Quality of Service
Grids
Clouds
List scheduling
Search-based algorithms

ABSTRACT

The execution of scientific applications, under the utility computing model, is constrained to Quality of Service (QoS) parameters. Commonly, applications have time and cost constraints such that all tasks of an application need to be finished within a user-specified Deadline and Budget. Several algorithms have been proposed for multiple QoS workflow scheduling, but most of them use search-based strategies that generally have a high time complexity, making them less useful in realistic scenarios. In this paper, we present a heuristic scheduling algorithm with quadratic time complexity that considers two important constraints for QoS-based workflow scheduling, time and cost, named Deadline–Budget Constrained Scheduling (DBCS). From the deadline and budget defined by the user, the DBCS algorithm finds a feasible solution that accomplishes both constraints with a success rate similar to other state-of-the-art search-based algorithms in terms of the successful rate of feasible solutions, consuming in the worst case only approximately 4% of the time. The DBCS algorithm has a low-time complexity of $O(n^2.p)$ for n tasks and p processors.

1. Introduction

Utility computing is a service provisioning model that provides computing resources and infrastructure management to consumers as they need them, as well as a payment model that charges for usage. Service-oriented grid and cloud computing, which supply frameworks that allow users to consume utility services in a secure, shared, scalable, and standard network environment, have become the basis for providing these services.

Computational grids have been used by researchers from various areas of science to execute complex scientific applications. Utility computing has been rapidly moving towards a pay-as-you-go model, in which computational resources or services have different prices with different performance and Quality of Service (QoS) levels [1]. In this computing model, users consume services and resources when they need them and pay only for what they use. In this context, cost and time become two of the most relevant user concerns. Thus, the cost/time trade-off problem for scheduling workflow applications has become challenging. Scheduling consists of defining an assignment and mapping of the workflow tasks onto resources. In general, the scheduling problem belongs to a class of problems known as NP-complete [2].

Most research on workflow QoS aware scheduling considers the optimization of one QoS parameter, such as time, constrained to another QoS parameter, such as cost [3,4]. Other approaches consider a bi-objective approach that consists in optimizing two QoS parameters, such as time and cost simultaneously [5–7], the

* Corresponding author.
E-mail addresses: hamid.arabnejad@fe.up.pt (H. Arabnejad), jbarbosa@fe.up.pt (J.G. Barbosa), radu@dps.uibk.ac.at (R. Prodan).

http://dx.doi.org/10.1016/j.future.2015.07.021
0167-739X/

constraints being mainly related to processor availability and load. Other combination of QoS parameters may be considered, such as time and reliability [8].

Workflow scheduling to satisfy multiple QoS parameters is becoming an active research area in the context of utility computing. Many algorithms have been proposed for multi-objective scheduling, but in most of them, meta-heuristic methods or search-based strategies have been used to achieve good solutions. However, these methods based on meta-heuristics or search-based strategies usually need significantly high planning costs in terms of the time consumed to produce good results, which makes them less useful in real platforms that need to obtain map decisions on the fly. In this paper, a low-time complexity heuristic, named Deadline–Budget Constrained Scheduling (DBCS), is proposed to schedule workflow applications on computational heterogeneous infrastructures constrained to two QoS parameters. In our model, the QoS parameters are time and cost. The objective of the proposed DBCS algorithm is to find a feasible schedule map that satisfies the user defined deadline and budget constraint values. To fulfill this objective, DBCS implements a mechanism to control the time and cost consumption by each task when producing a schedule solution. To the best of our knowledge, the algorithm proposed here is the first low-time complexity heuristic for a bounded number of heterogeneous resources addressing two QoS parameters that obtains similar performances to higher-time complexity scheduling algorithms in a small fraction of the scheduling time.

The contributions of this paper are:

- a review of multiple QoS parameter workflow scheduling on heterogeneous resources;
- a new heuristic algorithm with quadratic complexity for workflow application scheduling, constrained to time and cost, on a bounded set of heterogeneous resources;
- similar performances of search-based state-of-the-art algorithms in a small fraction of the time that ranges from 0.004% to 4%;
- a realistic simulation considering a bounded multi-port model in which bandwidth is shared by concurrent communications;
- extensive evaluation with results for randomly generated graphs as well as for real-world applications.

The remainder of the paper is organized as follows. Section 2 describes related work. Section 3 defines the scheduling problem and describes the system model. Section 4 presents the proposed scheduling algorithm. Section 5 presents results, and Section 6 concludes the paper.

2. Related work

Workflow scheduling has been extensively investigated. The scheduling strategies can be classified into two main categories: single and multiple QoS parameters.

On the single QoS parameter, the execution time of a workflow application, also called *makespan*, has been the major concern in most of the scheduling strategies. In [9,10], a wide study on list scheduling algorithms is presented for *makespan* optimization.

The problem becomes more challenging when two or more QoS parameters are considered in the scheduling problem. Time, cost, energy and reliability are common QoS parameters considered in recent research work in this area. Many algorithms consider time and cost in their formulation but most of them perform: (a) optimization of one parameter constrained to the other; (b) optimization of both parameters in a bi-objective formulation; and (c) a consideration of an unlimited number of resources, in particular for cloud platforms, where the strategy to accomplish time constraints is by allocating new computational instances.

Concerning the optimization of one parameter constrained to the other, Mao et al. [11] proposed an auto-scaling mechanism that automatically scales computing instances based on workload information to minimize the cost of the scheduling map while meeting application deadlines on cloud environments. Zeng et al. [12] proposed *ScaleStar*, a budget-conscious scheduling algorithm to minimize the execution time of large-scale many-task workflows in Clouds with monetary costs. Yu et al. [13] proposed a QoS-based workflow scheduling algorithm utilizing a Markov Decision Process approach for the service Grid that minimizes the total cost of the application while meeting the deadline constraints imposed by the user. Their algorithm first categorizes tasks into two classes: synchronization tasks (the nodes that have more than one parent or child) and simple tasks. Then, the original workflow is partitioned into sub-workflows, and based on the two classes of tasks, sub-deadlines are assigned to each partition. Finally, the cost optimized mapping for each partition is obtained, guaranteeing the application deadline. In [14], Yuan et al. proposed a time–cost tradeoff dynamic heuristic scheduling strategy to optimize the cost and time of the whole workflow. In addition, [15] presented a heuristic scheduling algorithm called DET (Deadline Early Tree), which minimizes cost with a deadline constraint. The communication time between tasks is not considered in their model. Sakellariou et al. [16] presented the LOSS1 algorithm to construct schedules that optimize time constrained to a cost. The algorithm uses initial assignments made by other heuristic algorithms to meet the time optimization objective; then, a reassignment strategy is implemented to reduce cost and meet the cost constraint that is specified by the user budget. Zheng et al., in [4,17], proposed the algorithm Budget-constrained Heterogeneous Earliest Finish Time (BHEFT), which optimizes the execution time of a workflow application constrained to a budget. Arabnejad et al. [3] proposed a Heterogeneous Budget Constrained Scheduling (HBCS) algorithm that guarantees an execution cost within the users specified budget and that minimizes the application execution time similarly to BHEFT. The results presented show that the HBCS algorithm achieves lower makespans, with a guaranteed cost per application. The algorithm proposed in this paper extends the HBCS algorithm to consider deadline (time) and budget (cost) as constraints. Similar to HBCS, in this paper, we propose a quality measure for each processor that combines time and cost constraints, which is used for processor selection and may not necessarily select the processor that guarantees the earliest finish time. BHEFT uses a different formulation that, in two steps, selects the set of affordable processors, the cost factor, and then selects the processor that minimizes the processing time. LOSS1 and BHEFT are also selected for comparison to the algorithm proposed in this paper, DBCS, as an alternative approach that optimizes time constrained to a budget.

The following algorithms use a bi-objective formulation. Talukder et al. [18] proposed a workflow execution planning approach using Multi-objective Differential Evolution (MODE) to satisfy the user time and cost constraint parameters. Chen et al. [19] proposed an ant colony optimization (ACO) to schedule large-scale workflows with various QoS parameters such as reliability, time, and cost in computational grids. Garg et al. [20] proposed a multi-objective non-dominated sort particle swarm optimization (NSPSO) approach to find schedule maps minimizing the makespan and total cost under the specified deadline and budget constraints. In [21], Yu et al. proposed a genetic algorithm (GA) approach for scheduling workflow applications constrained to budget and deadline, on heterogeneous environments. Two fitness functions are used to encourage the formation of individuals who satisfy the deadline and budget constraints. Prodan et al. [5] proposed a general bi-criteria scheduling heuristic called the Dynamic Constraint Algorithm (DCA) based on dynamic programming to optimize two independent generic criteria for workflows, e.g., execution time