Future Generation Computer Systems 55 (2016) 1–16

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Reliability-driven scheduling of time/cost-constrained grid workflows

Somayeh Kianpisheh [a], Nasrolah Moghadam Charkari [a,*], Mehdi Kargahi [b,c]

[a] Faculty of Electrical and Computer Engineering, Tarbiat Modares University, Tehran, Iran
[b] School of Electrical and Computer Engineering, Faculty of Engineering, University of Tehran, Tehran, Iran
[c] School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran

HIGHLIGHTS

- The method maximizes the workflow reliability while respecting budget and deadline.
- Two heuristics find feasible schedule and one enhances reliability within Ant Colony.
- Real world workflows under various pair of constraints have been investigated.
- Feasibility ratio and workflow execution reliability have been improved.
- The feasibility ratio increment has reduced Grid profit loss.

ARTICLE INFO

Article history:
Received 9 August 2014
Received in revised form
24 May 2015
Accepted 27 July 2015
Available online 5 August 2015

Keywords:
Workflow scheduling
Reliability
Real-time systems
Grid computing
Constrained optimization
Ant colony optimization

ABSTRACT

Workflow scheduling in Grids and Clouds is a NP-Hard problem. Constrained workflow scheduling, arisen in recent years, provides the description of the user requirements through defining constraints on factors like makespan and cost. This paper proposes a scheduling algorithm to maximize the workflow execution reliability while respecting the user-defined deadline and budget. We have used ant colony system to minimize an aggregation of reliability and constraints violation. Three novel heuristics have been proposed which are adaptively selected by ants. Two of them are employed to find feasible schedules and the other is used to enhance the reliability. Two methods have been investigated for time and cost considerations in the resource selection. One of them assigns equal importance to the time and cost factors, and the other weighs them according to the tightness of satisfaction of the corresponding constraints. Simulation results demonstrate the effectiveness of the proposed algorithm in finding feasible schedules with high reliability. As it is shown, as an additional achievement, the Grid profit loss has been decreased.

1. Introduction

Grid computing has provided an infrastructure for coordination and sharing of large scale, heterogeneous and geographically distributed resources such as computing and data resources, supercomputers and instruments. Grid middlewares like Globus [1], Gridbus [2] and Legion [3] have facilitations for users to access resources over the network to carry out their scientific and business applications which are usually modeled by workflows. A workflow is defined as a composition of several atomic activities supposed to be executed according to a predefined order to achieve a specific goal [4]. Workflow related issues have been regarded as one of the most interesting research areas in Grid computing. Various **W**orkflow **M**anagement **S**ystems (WMS) like Pegasus [5], Kepler [6] and Condor DAGMan [7] have been emerged to present efficient solutions for executing Grid workflows.

Workflow scheduling is regarded as one of the major challenges in WMS. Scheduling is defined as mapping of activities to resources and specifying their execution start time to meet user requirements while realizing Grid objectives. This problem, which is essential for effective utilization of resources, has been proven to be NP-hard [8,9].

Conventional scheduling algorithms attempt to minimize the workflow makespan as the only quality measure. List based scheduling algorithms like Min–Min [10], Max–Min [10], HEFT [11], as well as evolutionary-based methods [12] have been applied to minimize the makespan. However, in Grids, other factors like workflow execution reliability and resource operational costs are

* Corresponding author.
E-mail addresses: s_kianpisheh@modares.ac.ir (S. Kianpisheh), moghadam@modares.ac.ir (N.M. Charkari), kargahi@ut.ac.ir, kargahi@ipm.ir (M. Kargahi).

http://dx.doi.org/10.1016/j.future.2015.07.014

also of quite importance. For example, the authors of [13] have applied particle swarm optimization to make some trade-off among makespan, cost and reliability. Increasing success ratio has been achieved through prediction of resource failure [14]. The ordinal optimization method has also been used in [15] to minimize the execution time and cost in workflow scheduling.

As another category of studies, workflow scheduling to optimize an objective function while satisfying some constraints, has attracted the attention of researchers in recent years. Cost minimization while meeting deadline has been considered in [16–18]. Makespan minimization while maintaining the workflow execution cost below some user defined budget has been investigated in [19]. Supporting these constraints is favorable for both users and Grid owners. From users' viewpoint, many scientific and business goals are realized when the execution of workflow is completed within some deadline [20,21]. Users usually have monetary constraints to execute their applications [21,22]. Form Grid owners' viewpoint, economic benefits will be gained; since the better the level of quality of service the more users will pay for the service. Two drawbacks are apparent in the related works; first, the constraints have been assumed loosely enough such that there are many feasible solutions. Thus, the concentration in those works is on optimization of the objective function. However, in the real world, the scheduling system copes with medium and tight constraints satisfaction problems. Thus, research is needed to be done to assess and improve the capability of a scheduling algorithm to find feasible solutions in various situations for the constraints. Second, in these works the resources have been assumed to be reliable. However, the resources may become unavailable due to canceling the Grid membership, link failure, power variation, and software/hardware failures [23,24].

In this paper, we propose a **C**onstrained **W**orkflow **S**cheduling algorithm called CWS. The scheduling is done such that the reliability of workflow execution is maximized while the makespan satisfies the user defined deadline and the execution cost is within the user defined budget. The problem is solved using ant colony system [25] to minimize an aggregation of violation function and workflow reliability. Violation function reflects the violation of budget and deadline constraints. Three novel heuristics are proposed to guide search direction. These heuristics include: **G**reedy **E**xploration (GE), **D**eadline and **B**udget **D**ecomposition based **E**xploration (DBDE) and **R**eliability **E**nhancement (RE). GE encourages ants to greedily select resources that minimize an aggregation of the execution cost and the response time of activities. DBDE encourages selection of resources that meet the sub-deadlines and sub-budgets achieved from decomposing deadline and budget among activities. These heuristics result in finding feasible solutions. RE increases the reliability of activities executions while preserving the constraints. The heuristics are used by ants through an adaptive strategy. The simulation results on actual workflows generated by Pegasus [26] show the algorithm is effective in finding feasible schedules with high reliability. The results also indicate that improving feasibility ratio considerably reduces the Grid profit loss.

The remainder structure of the paper is as follows. Section 2 discusses the related work. Section 3 describes the modeling of system, problem definition and the evaluation metrics. The details of our proposed algorithm will be presented in Section 4. In Section 5, the complexity of the algorithm is discussed. Section 6 illustrates an example. The simulation results will be presented in Section 7. Section 8 gives the conclusion and future work.

2. Related work

Few researches have been done in the area of constrained workflow scheduling. Some studies have considered budget constraint while minimizing makespan. In [19], a primitive scheduling that minimizes the makespan is constructed. If the cost is within the budget, the schedule will be finalized. Otherwise, activities are remapped so that the cheaper resources are selected to meet the budget constraint.

Some researchers attempt for cost minimization regarding a deadline. In [27], the cheapest resource is selected for each activity. In the case of deadline violation, back tracking is done to choose more expensive resources that help meeting the deadline. However, the repetition of backtracking is time-consuming. The authors of [18] employ a deadline decomposition strategy to compute a time window for each activity. The cheapest resource which executes the activity within the associated window is selected as the host. Similar solutions with different strategies for deadline decomposition have been presented in [16,17]. In [28], by analyzing the performance of resources, a batch of workflows is scheduled such that at least some of the workflows satisfy the associated deadline while minimizing the cost.

Few studies have considered both time and cost constraints within workflow scheduling. In [29] a hierarchical algorithm has been suggested for cloud computing such that a budget and deadline are satisfied for each activity. Using a random search, activities are mapped to data centers. Then, an evolutionary algorithm has been used to assign activities to virtual machines within a data center. Some researches assume a deadline and budget for the whole workflow. A genetic based algorithm has been proposed in [30]. Each individual represents a vector of assignment of activities to resources. The algorithm tries to find the proper individual that satisfies both constraints as much as possible.

In the mentioned works, it is supposed that the resources are always available. However, in real world, the resources may become unavailable due to reasons like canceling the Grid membership, link failure, power variation and software/hardware failures [23,24]. Thus, considering reliability is essential for efficient workflow scheduling to reduce the failure of workflow execution. In this paper, a scheduling algorithm is proposed such that the reliability of workflow execution will be maximized while the deadline and budget is met. Three novel heuristics are exploited within ant colony optimization to solve the problem. The results are compared with [21] that advocates a greedy approach toward optimization of time, reliability and cost of activities. The proposed algorithm, has improved the feasibility ratio for various constraints. The Grid profit loss has been reduced and the reliability has been enhanced.

3. System model, problem definition and metrics

This section illustrates how we model the workflow and Grid. Then the problem is described formally and the metrics for evaluation are defined.

3.1. Workflow and grid models

A common representation of a workflow as a directed acyclic graph (DAG) [31] is utilized. Let $G = (A, E)$ be the graph corresponding to the workflow where $A = \{a_1, a_2 \dots a_m\}$ is the set of activities. The number of instructions of a_i is indicated by I_i. E is the set of edges showing precedence relationship among activities. An edge $(a_i, a_j) \in E$ indicates that a_i is a parent of a_j. Two dummy entry and exit activities represented by a_1 and a_m are assumed at the start and bottom of workflow, respectively. $\boldsymbol{D}$ is a $m \times m$ matrix where d_{ij} shows the size of data transmitted from a_i to a_j. In a workflow, an activity can start the execution when the executions of all of its parents have been finished and its required data has been transmitted.

The Grid is modeled by a set of n heterogeneous resources: $RE = \{r_1, r_2 \dots r_n\}$. The computational speed of r_j is defined by