The Journal of Systems and Software 113 (2016) 1–26

Contents lists available at ScienceDirect

The Journal of Systems and Software

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

Cost optimization approaches for scientific workflow scheduling in cloud and grid computing: A review, classifications, and open issues

Ehab Nabiel Alkhanak[a,*], Sai Peck Lee[a], Reza Rezaei[a], Reza Meimandi Parizi[b]

[a] Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia
[b] School of Engineering and Computing Sciences, New York Institute of Technology, Nanjing Campus, China

ARTICLE INFO

Article history:
Received 6 April 2015
Revised 12 November 2015
Accepted 14 November 2015
Available online 2 December 2015

Keywords:
Scientific workflow
Scheduling
Cloud computing

ABSTRACT

Workflow scheduling in scientific computing systems is one of the most challenging problems that focuses on satisfying user-defined quality of service requirements while minimizing the workflow execution cost. Several cost optimization approaches have been proposed to improve the economic aspect of Scientific Workflow Scheduling (SWFS) in cloud and grid computing. To date, the literature has not yet seen a comprehensive review that focuses on approaches for supporting cost optimization in the context of SWFS in cloud and grid computing. Furthermore, providing valuable guidelines and analysis to understand the cost optimization of SWFS approaches is not well-explored in the current literature. This paper aims to analyze the problem of cost optimization in SWFS by extensively surveying existing SWFS approaches in cloud and grid computing and provide a classification of cost optimization aspects and parameters of SWFS. Moreover, it provides a classification of cost based metrics that are categorized into monetary and temporal cost parameters based on various scheduling stages. We believe that our findings would help researchers and practitioners in selecting the most appropriate cost optimization approach considering identified aspects and parameters. In addition, we highlight potential future research directions in this on-going area of research.

1. Introduction

Efficient resources utilization remains a key issue in parallel and distributed computing environments. To resolve this issue, an organization needs to focus on finding the most suitable allocation of an application's tasks to available computational resources. This notion is generally referred as scheduling (Wang et al., 2013; Liu et al., 2010a). Optimal scheduling problem is known to be an NP-complete problem (Wu et al., 2013b; Bittencourt and Madeira, 2011; Ramakrishnan et al., 2011). There is no proposed scheduling approach that can achieve an optimal solution within the polynomial time, especially in the case of scheduling large-size tasks (Abrishami and Naghibzadeh, 2013; Yu and Buyya, 2006b). Users can employ different available computational resources to execute the tasks in an efficient manner. However, current limited computational resources lack in accomplishing users' demands (e.g., strict service completion deadline, and vast amount of required storage) due to the tremendous increase in complexity and size of today's applications. Consequently, users need to determine an appropriate computational environment that provides the required storage space and computational resources for processing large-scale complex applications.

Grid computing and cloud computing resources can provide an optimal solution that can meet the user's requirements by providing scalable and flexible solutions for considered applications (Wu et al., 2013b). The cloud computing based task scheduling differs from the grid computing based scheduling in the following two ways:

- Resource sharing: Cloud computing offers advanced services by sharing resources using the virtualization notion with the help of internet technologies. Consequently, it supports real-time allocation to fully utilize the available resources while improving elasticity of cloud services. Thus, the scheduler in a cloud workflow system needs to consider the virtualization infrastructure (e.g., virtual services and virtual machines) to efficiently facilitate the computational processes. In contrast, grid computing allows allocating a large cluster of resources in a shared mode. Therefore, it supports batch processing and resources will be available once they are released by other users.
- Cost of resource usage: Cloud computing provides a flexible costing mechanism in considering the user's requirements (i.e. pay-as-you-go and on-demands services). On the other hand, grid computing follows a quota strategy to determine the accumulated cost of requested services (Foster et al., 2008). Therefore,

* Corresponding author. Tel.: +60 162429667.
E-mail addresses: ehabsoa@gmail.com, ehabsoa@siswa.um.edu.my (E.N. Alkhanak), saipeck@um.edu.my (S.P. Lee), rezarezaei@siswa.um.edu.my (R. Rezaei), rparizi@nyit.edu (R.M. Parizi).

http://dx.doi.org/10.1016/j.jss.2015.11.023
0164-1212/

grid computing has no flexible costing mechanism as in cloud computing.

In the literature, researchers have categorized task-scheduling strategies into two main categories: (i) job-based, and (ii) workflow-based (Liu et al., 2010b; Deng et al., 2011a; Czarnul, 2013; Ma et al., 2009; Viana et al., 2011). Job-based scheduling usually focuses on scheduling a set of independent tasks to be executed in a sequence or parallel manner (Sharif et al., 2013; Varalakshmi et al., 2011). In contrast, workflow-based scheduling (or global task scheduling) aims at mapping and managing the execution of inter-dependent (i.e. precedence constraints) tasks on shared resources for applications with higher complexity (Kaur et al., 2011). The workflow can be defined as multiple steps or activities, which are necessary to complete a submitted task. The components of these activities can be any executable instances (e.g. load sets, report sets, programs, and data) with different structures (e.g. process, pipeline, data distribution, data aggregation, and data redistribution). The workflow scheduling attained more attention of researchers compared to job scheduling, since workflow-based scheduling is able to efficiently determine an optimal solution for large and complex applications by considering precedence constraints between potential tasks. Motivated by this, we focused on reviewing workflow-based scheduling in cloud and grid computing. Workflow-based scheduling is commonly represented using a Directed Acyclic Graph (DAG) model (Wu et al., 2013b; Talukder et al., 2009; Pandey et al., 2010; Yu and Buyya, 2006a; Liu et al., 2011b). The DAG is usually represented by:

$$DAG = \{T, E\} \tag{1}$$

where T (vertex) is a set of tasks (a task can be any program that the user would like to execute in a workflow application) and E is a set of directed edges between the vertices.

$$T = \{t_0, \ldots, t_n\} \tag{2}$$

$$E = \{e_1, \ldots, e_m\} \tag{3}$$

Note that there is a data dependency between edges in E. For instance, if there is a directed edge e (*i.e.* $e \in E$) connecting t_i and t_j (denoted as $t_i \rightarrow t_j$), then t_i is considered as a parent and t_j as a child. The input data of task j depends on the produced data by the parent task i. Similarly, the complete path from t_0 to t_n can be represented as:

$$(t_0 \rightarrow t_1), (t_1 \rightarrow t_2), \ldots, (t_{n-2} \rightarrow t_{n-1}), (t_{n-1} \rightarrow t_n) \tag{4}$$

In order to execute workflow tasks in cloud and grid computing, it requires tasks mapping to the set of heterogeneous resources, which are commonly used in cloud as a set of Virtual Machines (VMs):

$$VM = \{vm_0, \ldots, vm_k\} \tag{5}$$

Furthermore, it is crucial to consider the computational cost (in terms of time) of executing the workflow tasks on available heterogeneous VMs along with the communication cost between these VMs.

Traditionally, the information technology staff manually executes workflow tasks, which requires knowledge about resource availability and the estimated starting time for each workflow task (Ranaldo and Zimeo, 2009; Wang et al., 2013; Miu and Missier, 2012). It is necessary to automate and optimize the workflow scheduling process in order to achieve an efficient Workflow Management System (WfMS). A WfMS defines, manages, and executes workflows on available computing resources, where the workflow execution order is driven by a computer representation of the workflow logic. The WfMS can be implemented for different purposes including process management, process re-design/optimization, system integration, achieving flexibility, and improving maintainability. The main stages of any WfMS are modeling stage, instantiation stage and execution stage (as depicted in Fig. 1) (Liu, 2012). In the modeling stage, scientific processes are redesigned based on cloud workflow specifications which should contain the task definitions, tasks structural representation (e.g. DAG), and user-defined QoS requirements. The cloud workflow service provider will negotiate with the service consumer to finalize Service Level Agreement (SLA). In the instantiation stage, the WfMS selects and reserves the suitable cloud services (from private cloud, public cloud, and hybrid cloud) based on the SLA in order to execute workflow activities as well as satisfy the defined QoS requirements. Finally, at the execution stage, the cloud workflow execution scheduler coordinates the data and control flows according to the workflow specifications obtained from the modeling stage, and employs the candidate software services reserved at the instantiation stage to execute all the workflow activities. The workflow scheduler (i.e. workflow engine) plays a crucial role in scheduling and allocating the given tasks to the available resources by considering their dependencies as modeled using a DAG.

WfMS in cloud and grid computing must have the ability to handle the requests from different application domains such as business workflow applications and scientific workflow applications. The business workflow application (also referred as transaction intensive workflow) has been defined by Workflow Management Coalition (WfMC) as the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules (e.g. bank transactions and insurance claim applications) (Wieczorek et al., 2009; de Oliveira et al., 2012; Chandrakumar, 2013; Frincu et al., 2014; Poola et al., 2014). Conversely, Scientific Workflow Application (SWFA) (also known as data and computational intensive scientific workflow) mostly implies data flows together with the tasks execution (Wieczorek et al., 2009; Ma et al., 2009; Malawski et al., 2014; Tolosana-Calasanz et al., 2012), including input scripts (scientific program or data), which can be used to produce, analyze and visualize output results. It can provide interactive tools to help scientists better execute their own workflows and view results in real time. In addition, the SWFA simplifies the process for scientists to reuse the same workflows and provide them with an easy-to-use environment to track and share the output results virtually. Thus, SWFAs have been used in different scientific applications including weather forecasting, bioinformatics, geoinformatics, cheminformatics, biomedical informatics, and astrophysics (Wu et al., 2013b; Malawski et al., 2012). To execute SWFA data, high performance resources, such as supercomputers, need to be delivered by the service provider (i.e. infrastructure as a service) (Wu et al., 2013b; Yan et al., 2013; Deelman et al., 2013; Bittencourt and Madeira, 2013; Malawski et al., 2012). Therefore, WfMSs using cloud and grid services enable scientists to define multi-stage computational and data processing pipelines that can be executed as resources with predefined quality of service. Consequently, the scheduling process can automate complex analyses, improve application performance, and reduce the time required to obtain the desired results (Sharif et al., 2013; Czarnul, 2013; Malik et al., 2013; Barrett et al., 2011; Hameed et al., 2014). Inspired by this, we surveyed the studies that focused on Scientific Workflow Scheduling (SWFS) in cloud and grid computing.

One of the most challenging problems with SWFS in cloud and grid computing is to optimize the cost of workflow execution (Abrishami and Naghibzadeh, 2013; Yu and Buyya, 2006a; Li et al., 2015). The cost optimization challenge of SWFS in cloud computing is a multi-objective cost-aware problem that requires consideration of three main aspects: (i) different users which usually compete for resources within the cloud or grid computing to satisfy QoS constraints, (ii) the inter-dependencies among workflow tasks, and (iii) high communication cost due to the inter-dependencies between the tasks (i.e. data needs to be transferred from one resource to another). However, considering all cost optimization problem related aspects makes the SWFS process more complicated and also requires a high amount of computational resources in terms of computational