Future Generation Computer Systems 37 (2014) 309–320

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Multi-objective scheduling of many tasks in cloud platforms

Fan Zhang [a,b,*], Junwei Cao [b], Keqin Li [c], Samee U. Khan [d], Kai Hwang [e]

[a] Kavli Institute for Astrophysics and Space Research, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
[b] Research Institute of Information Technology, Tsinghua University, Beijing, 100084, China
[c] Department of Computer Science, State University of New York, New Paltz, NY 12561, USA
[d] Department of Electrical and Computer Engineering, North Dakota State University, Fargo, ND 58108-6050, USA
[e] Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089-2562, USA

HIGHLIGHTS

- We propose an ordinal optimized method for multi-objective many-task scheduling.
- We prove the suboptimality of the proposed method through mathematical analysis.
- Our method significantly reduces scheduling overhead by introducing a rough model.
- Our method delivers a set of semi-optimal good-enough scheduling solutions.
- We demonstrate the effectiveness of the method on a real-life workload benchmark.

ARTICLE INFO

Article history:
Received 21 May 2013
Received in revised form
26 July 2013
Accepted 5 September 2013
Available online 18 September 2013

Keywords:
Cloud computing
Many-task computing
Ordinal optimization
Performance evaluation
Virtual machines
Workflow scheduling

ABSTRACT

The scheduling of a many-task workflow in a distributed computing platform is a well known NP-hard problem. The problem is even more complex and challenging when the virtualized clusters are used to execute a large number of tasks in a cloud computing platform. The difficulty lies in satisfying multiple objectives that may be of conflicting nature. For instance, it is difficult to minimize the makespan of many tasks, while reducing the resource cost and preserving the fault tolerance and/or the quality of service (QoS) at the same time. These conflicting requirements and goals are difficult to optimize due to the unknown runtime conditions, such as the availability of the resources and random workload distributions. Instead of taking a very long time to generate an optimal schedule, we propose a new method to generate suboptimal or sufficiently good schedules for smooth multitask workflows on cloud platforms.

Our new multi-objective scheduling (MOS) scheme is specially tailored for clouds and based on the ordinal optimization (OO) method that was originally developed by the automation community for the design optimization of very complex dynamic systems. We extend the OO scheme to meet the special demands from cloud platforms that apply to virtual clusters of servers from multiple data centers. We prove the suboptimality through mathematical analysis. The major advantage of our MOS method lies in the significantly reduced scheduling overhead time and yet a close to optimal performance. Extensive experiments were carried out on virtual clusters with 16 to 128 virtual machines. The multitasking workflow is obtained from a real scientific LIGO workload for earth gravitational wave analysis. The experimental results show that our proposed algorithm rapidly and effectively generates a small set of semi-optimal scheduling solutions. On a 128-node virtual cluster, the method results in a thousand times of reduction in the search time for semi-optimal workflow schedules compared with the use of the Monte Carlo and the Blind Pick methods for the same purpose.

1. Introduction

Large-scale workflow scheduling demands efficient and simultaneous allocation of heterogeneous CPU, memory, and network bandwidth resources for executing a large number of computational tasks. This resource allocation problem is NP-hard [1,2]. How to effectively schedule many dependent or independent tasks on distributed sources that could be virtualized clusters of servers in a cloud platform makes the problem even more complex and challenging to solve, with a guaranteed solution quality.

The many-task computing paradigms were treated in [3–5]. These paradigms pose new challenges to the scalability problem, because they may contain large volumes of datasets and loosely coupled tasks. The optimization requires achieving multiple

* Corresponding author at: Kavli Institute for Astrophysics and Space Research, Massachusetts Institute of Technology, Cambridge, MA 02139, USA. Tel.: +1 617 682 2568.

E-mail addresses: f-zhang@tsinghua.edu.cn, f_zhang@mit.edu (F. Zhang), jcao@tsinghua.edu.cn (J. Cao), lik@newpaltz.edu (K. Li), samee.khan@ndsu.edu (S.U. Khan).

0167-739X/$ – see front matter
http://dx.doi.org/10.1016/j.future.2013.09.006

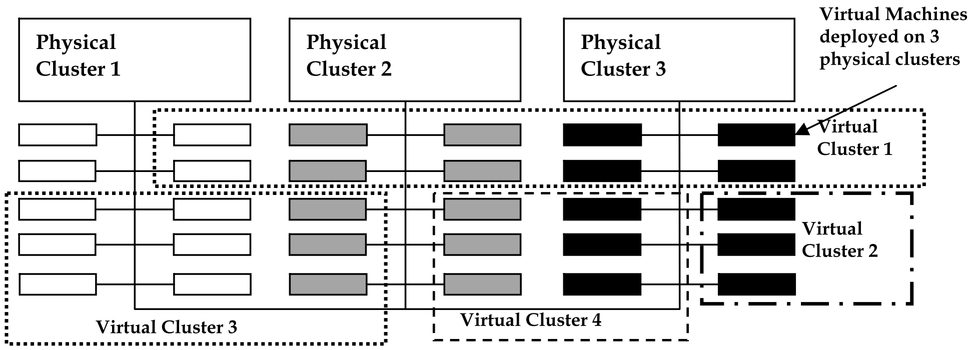

Fig. 1. A cloud platform built with four virtual clusters over three physical clusters. Each physical cluster consists of a number of interconnected servers, represented by the rectangular boxes with three different shadings for the three physical clusters shown. The virtual machines (VMs) are implemented on the servers (physical machines). Each virtual cluster can be formed with either physical machines or VMs hosted by multiple physical clusters. The virtual clusters boundaries are shown by four dot/dash-line boxes. The provisioning of VMs to a virtual cluster can be dynamically done upon user demands.

objectives. For example, it is rather difficult to minimize the scheduling makespan, the total cost, to preserve fault tolerance, and the QoS at the same time. Many researchers have suggested heuristics for the aforesaid problem [6].

The execution of a large-scale workflow encounters a high degree of randomness in the system and workload conditions [7,8], such as unpredictable execution times, variable cost factors, and fluctuating workloads that makes the scheduling problem computationally intractable [9]. The lack of information on runtime dynamicity defies the use of deterministic scheduling models, in which the uncertainties are either ignored or simplified with an observed average.

Structural information of the workflow scheduling problem sheds a light on its inner properties and opens the door to many heuristic methods. No free lunch theorems [10] suggest that all of the search algorithms for an optimum of a complex problem perform exactly the same without the prior structural knowledge. We need to dig into the prior knowledge on randomness, or reveal a relationship between scheduling policy and performance metrics applied.

The emerging cloud computing paradigm [11–13] attracts industrial, business, and academic communities. Cloud platforms appeal to handle many loosely coupled tasks simultaneously. Our LIGO [14] benchmark programs are carried out using a virtualized cloud platform with a variable number of virtual clusters built with many virtual machines on fewer physical machines and virtual nodes as shown in Fig. 1 of Section 3. However, due to the fluctuation of many task workloads in realistic and practical cloud platforms, resource profiling and simulation stage on thousands of feasible schedules are needed. An optimal schedule on a cloud may take an intolerable amount of time to generate. Excessive response time for resource provisioning in a dynamic cloud platform is not acceptable at all.

Motivated by the simulation-based optimization methods in traffic analysis and supply chain management, we extend the *ordinal optimization* (OO) [15,16] for cloud workflow scheduling. The core of the OO approach is to generate a *rough model* resembling the life of the workflow scheduling problem. The discrepancy between the rough model and the real model can be resolved with the optimization of the rough model. We do not insist on finding the best policy but a set of suboptimal policies. The evaluation of the rough model results in much lower scheduling overhead by reducing the exhaustive searching time in a much narrowed search space. Our earlier publication [17] indicated the applicability of using OO in performance improvement for distributed computing systems.

The remainder of the paper is organized as follows. Section 2 introduces related work on workflow scheduling and ordinal optimization. Section 3 presents our model for *multi-objective scheduling* (MOS) applications. Section 4 proposes the algorithms for generating semi-optimal schedules to achieve efficient resource provision in clouds. Section 5 presents the LIGO workload [18] to verify the efficiency of our proposed method. Section 6 reports the experimental results using our virtualized cloud platform. Finally, we conclude with some suggestions on future research work.

2. Related work and our unique approach

Recently, we have witnessed an escalating interest in the research towards resource allocation in grid workflow scheduling problems. Many classical optimization methods, such as opportunistic load balance, minimum execution time, and minimum completion time are reported in [19], and suffrage, min–min, max–min, and auction-based optimization are reported in [20,21].

Yu et al. [22,23] proposed economy-based methods to handle large-scale grid workflow scheduling under deadline constraints, budget allocation, and QoS. Benoit et al. [24] designed resource-aware allocation strategies for divisible loads. Li and Buyya [25] proposed model-driven simulation and grid scheduling strategies. Lu and Zomaya [26] and Subrata et al. [27] proposed a hybrid policy and another cooperative game framework. J. Cao et al. [28] applied a queue-based method to configure a multi-server to maximize profit for cloud service providers.

Most of these methods were proposed to address single objective optimization problems. Multiple objectives, if considered, were usually being converted to either a weighted single objective problem or modeled as a constrained single objective problem.

Multi-objective optimization methods were studied by many research groups [29–33,22,34] for grid workflow scheduling. To make a summarization, normally two methods are used. The first one, as introduced before, is by converting all of the objectives into one applying weights to all objectives. The other one is a cone-based method to search for a non-dominated solution, such as the Pareto optimal front [35]. The concept of a layer is defined by introducing the Pareto-front in order to compare policy performances [36]. An improved version [37] uses the count that one particular policy dominates others as a measure of the goodness of the policy. Our method extends the Pareto-front method by employing a new noise level estimation method as introduced in Section 4.2.

Recently, Duan et al. [1] suggested a low complexity game-theoretic optimization method. Dogan and Özgüner [29] developed a matching and scheduling algorithm for both the execution time and the failure probability that can trade off them to get an optimal selection. Moretti et al. [38] suggested all of the pairs to improve usability, performance, and efficiency of a campus grid.

Wieczorek et al. [6] analyzed five facets which may have a major impact on the selection of an appropriate scheduling strategy, and proposed taxonomies for multi-objective workflow scheduling. Prodan and Wieczorek [30] proposed a novel dynamic constraint