Future Generation Computer Systems 55 (2016) 428–443

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Multi-criteria and satisfaction oriented scheduling for hybrid distributed computing infrastructures

Mircea Moca [a,*], Cristian Litan [a], Gheorghe Cosmin Silaghi [a], Gilles Fedak [b]

[a] Babeş-Bolyai University, Cluj-Napoca, România

[b] INRIA, Université de Lyon, France

HIGHLIGHTS

- We designed an overall multi-criteria task scheduling method for hybrid DCIs.
- The scheduling method allows a systematic integration of new scheduling criteria into it.
- We defined a methodology for finding optimal scheduling strategies.
- For the validation we consider both user and resource owners perspectives.
- We presented the experimental system built for the validation of the scheduling method.

ARTICLE INFO

Article history:
Received 30 April 2014
Received in revised form
24 March 2015
Accepted 27 March 2015
Available online 28 May 2015

Keywords:
Elastic computing infrastructures
Hybrid distributed computing infrastructures
Pull-based scheduling
Multi-criteria scheduling
Promethee scheduling

ABSTRACT

Assembling and simultaneously using different types of distributed computing infrastructures (DCI) like Grids and Clouds is an increasingly common situation. Because infrastructures are characterized by different attributes such as price, performance, trust, and greenness, the task scheduling problem becomes more complex and challenging. In this paper we present the design for a fault-tolerant and trust-aware scheduler, which allows to execute Bag-of-Tasks applications on elastic and hybrid DCI, following user-defined scheduling strategies. Our approach, named *Promethee scheduler*, combines a pull-based scheduler with multi-criteria Promethee decision making algorithm. Because multi-criteria scheduling leads to the multiplication of the possible scheduling strategies, we propose SOFT, a methodology that allows to find the optimal scheduling strategies given a set of application requirements. The validation of this method is performed with a simulator that fully implements the Promethee scheduler and recreates an hybrid DCI environment including Internet Desktop Grid, Cloud and Best Effort Grid based on real failure traces. A set of experiments shows that the Promethee scheduler is able to maximize user satisfaction expressed accordingly to three distinct criteria: price, expected completion time and trust, while maximizing the infrastructure useful employment from the resources owner point of view. Finally, we present an optimization which bounds the computation time of the Promethee algorithm, making realistic the possible integration of the scheduler to a wide range of resource management software.

1. Introduction

The requirements of distributed computing applications in terms of processing and storing capacities are continuously increasing, pushed by the gigantic deluge of large data volume to process. Nowadays, scientific communities and industrial companies can choose among a large variety of distributed computing infrastructures (DCI) to execute their applications. Examples of such infrastructures are Desktop Grids or Volunteer Computing systems [1] which can gather a huge number of volunteer PCs at almost no cost, Grids [2] which assemble large number of distributed clusters and more recently, clouds [3] which can be accessed remotely, following a pay-as-you-go pricing model. All these infrastructures have very different characteristics in terms of computing capacity, cost, reliability, consumed power efficiency and more. Hence, combining these infrastructures in such a way that meets users' and applications' requirements raises significant scheduling challenges.

* Corresponding author.
E-mail addresses: mircea.moca@econ.ubbcluj.ro (M. Moca), cristian.litan@econ.ubbcluj.ro (C. Litan), gheorghe.silaghi@econ.ubbcluj.ro (G.C. Silaghi), gilles.fedak@inria.fr (G. Fedak).

http://dx.doi.org/10.1016/j.future.2015.03.022
0167-739X/

The first challenge concerns the design of the resource management middleware which allows the assemblage of hybrid DCIs. The difficulty relies in the number of desirable high level features that the middleware has to provide in order to cope with: (i) distributed infrastructures that have various usage paradigms (reservation, on-demand, queue), and (ii) computing resources that are heterogeneous, volatile, unreliable and sometimes not trustee. An architecture that has been proved to be efficient to gather hybrid and elastic infrastructures is the joint use of a *pull-based scheduler* with pilot jobs [4–7]. The pull-based scheduler, often used in Desktop Grid computing systems [8,9], relies on the principle that the computing resources pull tasks from a centralized scheduler. Pilot jobs consist in resource acquisition by the scheduler and the deployment on them of agents with direct access to the central pull-based scheduler. In this way the scheduler can directly work with the resources, rather than going through local job schedulers. This approach exhibits several desirable properties, such as scalability, fault resilience, ease of deployment and ability to cope with elastic infrastructures, motivating us to use it in our scheduler.

The second challenge is to design task scheduling that are capable of efficiently using hybrid DCIs, and in particular, that takes into account the differences between the infrastructures. In particular, the drawback of a pull-scheduler is that it flattens the hybrid infrastructures and tends to consider all computing resources on an equal basis. Our earlier results [10,11] proved that a *multi-criteria scheduling* method based on the Promethee decision model [12] can make a pull-based scheduler able to implement scheduling strategies aware of the computing resources characteristics. However, in this initial work, we tested the method on single infrastructure type at a time, without considering hybrid computing infrastructures, and we evaluated the method against two criteria: expected completion time (ECT) and usage price. In this paper, we propose the following extensions to the Promethee scheduler: (i) we add a third criteria called *Expected Error Impact* (EEI), that reflects the confidence that a host returns correct results, (ii) we evaluate the Promethee scheduler on hybrid environments, (iii) we leverage the tunability of the Promethee scheduler so that applications developers can empirically configure the scheduler to put more emphasize on criteria that are important from their own perspective.

The third challenge regards the design of a new scheduling approach that maximizes satisfaction of both users and resource owners. In general, end users request to run their tasks quicker and at the cheapest costs, opposed to the infrastructure owners which need to capitalize their assets and minimize the operational costs. Thus, an overall scheduling approach should allow the resource owners to keep their business profitable and meantime, increase the end user satisfaction after the interaction with the global computing system.

The Promethee scheduler allows users to provide their own scheduling strategies in order to meet their applications requirements by configuring the relative importance of each criteria. However such configurable multi-criteria schedulers have two strong limitations: (i) there is no guaranty that the user preferences expressed when configuring the scheduler actually translates in an execution that follows the favored criteria, and (ii) the number of possible scheduling strategies explodes with the number of criteria and the number of application profiles, rapidly leading to an intractable situation by the user. We propose *Satisfaction Oriented FilTering* (SOFT), a new methodology that explores all the scheduling strategies provided by a Promethee multi-criteria scheduler to filter and select the most favorable ones according to the user execution profiles and the optimization of the infrastructure usage. SOFT also allows to select a default scheduling strategy so that the scheduler attains a high and at the same time stable level of user satisfaction, regardless the diversity of user satisfaction profiles.

In this paper, we introduce the design of the fault-tolerant and trust-aware Promethee scheduler, which allows to execute Bag-of-Tasks applications on elastic and hybrid DCI, following user-defined scheduling strategies. We thoroughly present the algorithms of the multi-criteria decision making and the SOFT methodology. Finally, we extensively evaluate the Promethee scheduler using a simulator that recreates a combination of hybrid, elastic and unreliable environment containing Internet Desktop Grid, public Cloud using Spot Instance and Best Effort Grid. Simulation results not only show the effectiveness of the Promethee scheduler but also its ability to meet user application requirements. We also propose an optimized implementation of the Promethee algorithms and perform real world experiments to validate the approach.

The remainder of the paper is organized as follows. In Section 2 we give the background for our work and define the key concepts, in Section 3 we explain our scheduling approach and define the performance evaluation metrics. In Section 4 we define SOFT, the methodology for optimal scheduling strategies selection. Then we present the architecture of the implemented experimental system in Section 5. In Section 6 we describe the experimental data, the setup and present the obtained results and findings. In Section 7 we discuss related work and finally Section 8 gives the concluding remarks and observations on this work.

2. Background

This section describes the multi-criteria scheduling on hybrid DCIs problem that we address in this work and defines the key concepts used in our discussion.

2.1. Description of the scheduling problem

In the considered context users submit their applications for execution on a system that aggregates the computing resources from at least three types of DCI: Internet Desktop Grids (IDG), Best Effort Grids (BEG) and Cloud. Each computing resource from the above mentioned infrastructures have different characteristics in terms of computing capacity, reliability, cost, consumed power efficiency, and trust. For instance, Internet volunteer Desktop PCs could be considered as free of charge but insecure and unreliable, while a Cloud resource can be costly but far more secure and reliable.

Users usually expect good execution performance but they are also concerned about other issues like cost, confidence and environmental footprint of the infrastructure. Thus, a relevant concern for task scheduling is to attain the *best* usage of the infrastructures in order to meet users' expectations and, *at the same time*, insure a convenient capitalization of the resources for their owners.

2.2. Key concepts

Users submit bag of work-units to a centralized scheduler and expect (after a while) the corresponding results. For each work-unit the scheduler creates at least one task and inserts it into a BoT (Bag of Task). During the execution the scheduler aims at emptying this BoT by scheduling tasks to hosts belonging to various types of computing infrastructure.

We use a *pull-based* scheduling strategy. Hence our scheduler is a centralized component (master) based on the *pull communication model* for the interaction with hosts (workers). The reason for this design choice is the requirement for *elasticity* and *adaptability* to structure disruptions that characterize DCIs like IDG and BEG. This model allows a complete independence of all system components [13]. The pull model allows workers to have the