

## Research Article

# Scheduling Multilevel Deadline-Constrained Scientific Workflows on Clouds Based on Cost Optimization

Maciej Malawski,<sup>1</sup> Kamil Figiela,<sup>1</sup> Marian Bubak,<sup>1,2</sup> Ewa Deelman,<sup>3</sup> and Jarek Nabrzyski<sup>4</sup>

*Department of Computer Science, AGH University of Science and Technology, Aleja Mickiewicza 30, 32-065 Kraków, Poland*

*ACC CYFRONET AGH, Ulica Nawojki 12, 32-065 Kraków, Poland*

*USC Information Sciences Institute, Admiralty Way, Marina del Rey, CA 90292, USA*

*Center for Research Computing, University of Notre Dame, Notre Dame, IN 46556, USA*

Correspondence should be addressed to Kamil Figiela; kfigiela@agh.edu.pl

Received 15 May 2016; Accepted 15 November 2016

Academic Editor: Roman Wyrzykowski

Copyright © 2016 Maciej Malawski et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a cost optimization model for scheduling scientific workflows on IaaS clouds such as Amazon EC2 or RackSpace. We assume multiple IaaS clouds with heterogeneous virtual machine instances, with limited number of instances per cloud and hourly billing. Input and output data are stored on a cloud object store such as Amazon S3. Applications are scientific workflows modeled as DAGs as in the Pegasus Workflow Management System. We assume that tasks in the workflows are grouped into levels of identical tasks. Our model is specified using mathematical programming languages (AMPL and CML) and allows us to minimize the cost of workflow execution under deadline constraints. We present results obtained using our model and the benchmark workflows representing real scientific applications in a variety of domains. The data used for evaluation come from the synthetic workflows and from general purpose cloud benchmarks, as well as from the data measured in our own experiments with Montage, an astronomical application, executed on Amazon EC2 cloud. We indicate how this model can be used for scenarios that require resource planning for scientific workflows and their ensembles.

## 1. Introduction

Today, science requires processing of large amounts of data and use of hosted services for compute-intensive tasks [1]. Cloud services are used not only to provide resources, but also for hosting scientific datasets, as in the case of AWS public datasets [2]. Scientific applications that run on these clouds often have the structure of workflows or workflow ensembles that are groups of interrelated workflows [3]. Infrastructure as a service (IaaS) cloud providers offer services where virtual machine instances differ in performance and price [4]. Planning computational experiments requires optimization decisions that take into account both execution time and resource cost.

Research presented in this paper can be seen as a step towards developing a cloud resource calculator for scientific applications in the hosted science model [5]. Specifically, we address the cost optimization problem of large-scale scientific

workflows running on multiple heterogeneous clouds, using mathematical modeling with AMPL [6] and CML [7], and mixed integer programming. This approach allows us to describe the model mathematically and use a set of available optimization solvers. On the other hand, an attempt to apply this method to the general problem of scheduling large-scale workflows on heterogeneous cloud resources would be impractical due to the problem complexity and therefore simplified models need to be analyzed. In our previous work [8], we used a similar technique to solve the problem where the application consists of tasks that either are identical or vary in size within a small range. As observed in [9, 10], large-scale scientific workflows often consist of multiple parallel stages or levels, each of which has a structure of set of tasks; that is, the tasks in each level are similar and independent of each other. In the case of large workflows, when the number of tasks in the level is high, it becomes more practical to optimize the execution of the whole level instead of looking



























