ARTICLE IN PRESS

Egyptian Informatics Journal xxx (2017) xxx–xxx

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

ELSEVIER

EGYPTIAN Informatics JOURNAL

Full length article

An extended Intelligent Water Drops algorithm for workflow scheduling in cloud computing environment

Shaymaa Elsherbiny [a], Eman Eldaydamony [a], Mohammed Alrahmawy [b,*], Alaa Eldin Reyad [c]

[a] Department of Information Technology, Faculty of Computer and Information, Mansoura University, Egypt
[b] Department of Computer Science, Faculty of Computer and Information, Mansoura University, Egypt
[c] Department of Information System, Faculty of Computer and Information, Mansoura University, Egypt

ARTICLE INFO

Article history:
Received 19 September 2016
Revised 17 May 2017
Accepted 2 July 2017
Available online xxxx

Keywords:
Cloud computing
Scheduling algorithms
Resource management
Intelligent Water Drops
Workflow scheduling
Natural-based algorithms

ABSTRACT

Cloud computing is emerging as a high performance computing environment with a large scale, heterogeneous collection of autonomous systems and flexible computational architecture. Many resource management methods may enhance the efficiency of the whole cloud computing system. The key part of cloud computing resource management is resource scheduling. Optimized scheduling of tasks on the cloud virtual machines is an NP-hard problem and many algorithms have been presented to solve it. The variations among these schedulers are due to the fact that the scheduling strategies of the schedulers are adapted to the changing environment and the types of tasks. The focus of this paper is on workflows scheduling in cloud computing, which is gaining a lot of attention recently because workflows have emerged as a paradigm to represent complex computing problems. We proposed a novel algorithm extending the natural-based Intelligent Water Drops (IWD) algorithm that optimizes the scheduling of workflows on the cloud. The proposed algorithm is implemented and embedded within the *workflows* simulation toolkit and tested in different simulated cloud environments with different cost models. Our algorithm showed noticeable enhancements over the classical workflow scheduling algorithms. We made a comparison between the proposed IWD-based algorithm with other well-known scheduling algorithms, including MIN-MIN, MAX-MIN, Round Robin, FCFS, and MCT, PSO and C-PSO, where the proposed algorithm presented noticeable enhancements in the performance and cost in most situations.

1. Introduction

Cloud computing is emerging as a hot topic that gains a lot of attention in both academia and industry as it enables data and services to be stored remotely but can be accessed from everywhere. Every day many companies and organizations transfer their data and applications to the cloud, due to its flexible and dynamic infrastructures, which offers QoS guaranteed computing environments in addition to configurable software services that can be rapidly provisioned and released with minimal management effort. Hence, large number of research~~ers~~ focuses on how to improve the performance and reduce the obstacles that may lead to degradation in the performance or unsafe use. Therefore, active research fields in cloud computing includes security, scheduling, privacy and policy, cloud storage, cloud performance, deployment, energy management, etc.)

Task scheduling is an important issue which greatly influences the performance of cloud computing environment. Scheduling is the process of mapping tasks to available resources on the basis of tasks' characteristics and requirements. Efficient scheduling is essential requirement for the cloud, as it can optimize the use of the cloud resources to provide highly efficient services with high quality. As more users begin to use clouds for deploying complex applications and store remote data, there is importunate need to have strong scheduling algorithms that can allocate tasks to data centers.

The focus in this paper is on the problem of workflow scheduling in the cloud computing environment. The problem of workflow scheduling in the cloud is a special case of the general problem of

* Corresponding author.
E-mail address: mrahmawy@gmail.com (M. Alrahmawy).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

ELSEVIER Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.eij.2017.07.001
1110-8665/

Please cite this article in press as: Elsherbiny S et al. An extended Intelligent Water Drops algorithm for workflow scheduling in cloud computing environment. Egyptian Informatics J (2017), http://dx.doi.org/10.1016/j.eij.2017.07.001

task scheduling in the cloud, where some precedence execution dependencies exist among some of the schedulable tasks; hence, workflow scheduling has, in addition to its own dependences characteristics, all the characteristics and features of the task scheduling problem.

There are many existing workflow scheduling algorithms for heterogeneous computing environments, these algorithms implement suitable compromise, or apply combination of scheduling algorithms according to different applications requirements. However, these algorithms have difficulties in being directly applied to the Cloud environments; as the Cloud, unlike traditional heterogeneous environment and the Grid, offers extremely large-size resource pools; however, using these pools is restricted by the complex processing cost scheme offered by the cloud provider. This adds an extra objective to be satisfied by the cloud scheduler; this objective is to optimize resource utilization of the rented cloud resources in order to minimize the processing cost of executing the workflows [1]. In this paper, we propose a cloud workflow scheduling algorithm based on the meta-heuristic IWD algorithm in order to satisfy the required objectives of minimizing both makespan and the cost of executing workflows on the cloud. In order to evaluate the proposed scheduler, we used workflowSim simulation environment to use the proposed algorithm to execute a set of common workflow in different simulated data centers with different configurations and cost models.

The rest of the paper is organized as follows: In Section 2, background and concepts of workflow scheduling in cloud computing are presented. In Section 3, the basic Intelligent Water Drops (IWD) algorithm is overviewed. In Section 4, some of the related work is presented. In Section 5, we present our proposed workflow scheduling algorithm and explain its different phases; then, in section 6, we give a simple detailed case study to explain the operation of the proposed algorithm. Then a set of performance and cost evaluation experiments and comparisons of the proposed algorithm with other algorithms are presented in Section 7, using various datacenters configurations and cost models. Finally conclusion and future work is presented in section 8.

2. Workflow scheduling in cloud computing

A Workflow is an attractive paradigm that helps scientists orchestrate complex, multistep simulations and analyses. Workflows are commonly used in distributed computing environments, e.g. grids and clouds for their powerful capabilities in modeling a wide range of applications, including scientific computing, multiprocessors system [2], multi-tier Web [3], and big data processing applications [4]. The problem of workflow scheduling in cloud has become an important research topic due to the development of cloud technology.

A Workflow is represented graphically using *Directed Acyclic Graph* (DAG) to reflect the interdependency among the workflow's tasks, where the nodes represent computation tasks to be executed on the available resources and directed edges represent data flow or control flow dependencies among the tasks. In general, the execution time of a task on a specific resource is inversely proportional to the speed of that resource; therefore, the optimized mapping of individual workflow tasks to the available computational cloud resources is an essential requirement in the cloud scheduler. In addition to the extensive workflows execution time requirements, due to the processing of significant amounts of data, workflow applications often involve dependencies that exist amongst tasks, which enforce the scheduler to respect these precedence requirements. The emergence of cloud computing has introduced a utility-type market model, where computational resources with varying capacities can be procured on demand, in a pay-per-use fashion. In general, the two most important objectives of workflow schedulers are the minimization of both cost and makespan. The cost involves not only the computational costs incurred from processing individual tasks, but also data transmission costs, where potentially large amounts of data must be transferred between compute and storage sites.

The problem of mapping various tasks of services to a set of resources is categorized as an NP-hard problem [5]. Finding solutions to NP-hard problems, using known algorithms is impractical, which means that it is difficult to build an optimum scheduler that works with a reasonable computation speed. In general, NP-hard problems can be solved by enumeration method, heuristic method or approximation method. However, building an optimum workflow scheduler using enumeration method requires first building all the possible schedulers and comparing them one by one to select the best one, which is not feasible for cloud workflow scheduling, as there are exhaustive number of possible schedulers due to the large number of workflows to be processed. Hence, schedulers built based on heuristics can be a suboptimal method to find reasonably good schedulers that are reasonably fast. For that, we propose a scheduler that extends the meta-heuristic IWD algorithm, to schedule tasks connected to each other in workflows and we evaluate this scheduler using the WorkflowSim simulator, which has been chosen as it provides a framework that takes into consideration heterogeneous system overheads and failures and it supports widely used workflow optimization techniques such as task clustering.

3. Intelligent Water Drops (IWD) algorithm

In recent years, IWD had been used in many fields to solve optimization and complex scientific problems such as Travelling salesman problem (TSP [6]), Job shop scheduling [7], Steiner tree problem [8], Code coverage [9], Graph Coloring [10], Optimizing routing protocol [11], Multidimensional Knapsack problem (MKP) [12], Vehicle routing problem [13], Air Robot Path Planning [14], Automatic multilevel three sholding using a modified Otsu's criterion [15], Optimal data aggregation tree in wireless sensor networks [16], Economic Load Dispatch [17], etc.

IWD algorithm *was first introduced by Shah-Hosseini, H. in 2007* [6]. The idea of IWD algorithm is inspired by the flow of natural water drops in lakes, rivers, and seas, where a water stream goes through an optimum path to reach its final destination, which is often a lake or a sea. In their way to the destination, the water drops react with the surrounding environment (river beds) and influence it. The river beds can be changed by the water drops and they can influence directions of the water drops. The gravity forces the water drops to move toward the destination. If there are no barriers or obstacles, the moving of water drops goes in a straight path to the destination. But actually, in real scenarios, there are different types of obstacles like twists, rocks and turns. In the original IWD algorithm, the Natural water drops have two properties: the velocity of the water drops and the amount of soil in their path. The water drops' soil collection and movement is negatively affected by the high amount of soil in their path, while they move faster along the path with less soil and can attain a higher speed and erode more soil from that path. The velocity of the water drops enables them to transfer soil from one place to another. More soil from the river beds can be gathered and transferred by faster water drops, where the velocity of the water drops is affected by the path properties.

In the IWD algorithm, the water drops move from the source to the destination in finite-length time steps. The water drops' velocity proportionally increases nonlinearly with the inverse of the soil of the path it goes through. Also, soils of the water drops increase