Improve Workflow Scheduling Technique for Novel Particle Swarm Optimization in Cloud Environment

R. Pragaladan[1], R. Maheswari[2]

[1]Assistant Professor, Department of Computer Science, Sri Vasavi College, Erode, India

[2] Research Scholar (M.Tech), Department of Computer Science, Sri Vasavi College, Erode, India

E-mail- mahe.rk123@gmail.com

Abstract—Cloud computing is the latest distributed computing paradigm [1], [2] and it offers tremendous opportunities to solve large-scale scientific problems. However, it presents various challenges that need to be addressed in order to be efficiently utilized for workflow applications. Although the workflow scheduling problem has been widely studied, there are very few initiatives tailored for cloud environments. Furthermore, the existing works fail to either meet the user's quality of service (QoS) requirements or to incorporate some basic principles of cloud computing such as the elasticity and heterogeneity of the computing resources. In this paper proposes a resource provisioning and scheduling strategy for scientific workflows on Infrastructure as a Service (IaaS) clouds. The proposed system presents an algorithm based on the particle swarm optimization (PSO), which aims to minimize the overall workflow execution cost while meeting deadline constraints.

Keywords—Cloud Environments, resource allocation, scheduling, PSO, Multi Cloud Provider

I. INTRODUCTION

The Scientific workflows [3] include ever-growing data and computing resources requirements and demand a high-performance computing cloud environment in order to be executed in a logical amount of time. These workflows are commonly modeled as a set of tasks interconnected via data or computing dependencies. The distributed resources have been studied extensively over the years, focusing on environments like grids and clusters. However, with the emergence of new paradigms such as cloud computing, novel approaches that address the particular challenges and opportunities of these technologies need to be developed.

Distributed environments have evolved from shared community proposals to usage-based models; the present of these being cloud environments. This novel technology enables the delivery of cloud related resources over the Inter communication system [4], and follows a usage-as-you-go model where users are charged based on their consumption. There are various types of cloud providers [5], each of which has different product offerings. They are classified into a hierarchy of as-a-service terms: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS).

The existing characteristic of preceding works developed for group of resources and grids is their center of attention on meeting application deadlines (total amount of time taken receive application) of the workflow though ignoring the cost of the exploit infrastructure. Even suited for such environments, policies developed for clouds are obliged to consider the usage-per-use model of the infrastructure in order to avoid prohibitive and preventable costs.

Our proposed work is based on the meta-heuristic optimization technique, particle swarm optimization (PSO). PSO is based on a swarm of particles moving through hole and converse with each other in order to finding an optimal search direction. PSO have been better totaling performance than other evolutionary algorithms [6] and fewer parameters to tune, which makes it easier to implement. Many problems in different areas has been successfully addressed by adapting PSO to specific domains; for instance this technique has been used to solve problems in areas such as reactive voltage control[7] , pattern recognition[8] and data mining[9],among others. In this paper, proposed system develops a static cost-minimization.

The main contributions of this paper are:

- To define the problem of scheduling prioritized workflow ensembles under budget and deadline constraints.
- To analyze and develop several dynamic and static algorithms for task scheduling and resource provisioning that rely on workflow structure information (critical shortest paths and workflow levels) and estimates of task runtimes in multi cloud providers
- To evaluate these algorithms using infrastructure model and the application, taking into account reservations in task runtime estimates, provisioning delays, and failures.
- To discuss the performance of the algorithms on a set of synthetic workflow ensembles based on important, real scientific applications, using a broad range of different application scenarios and varying constraint values.

The rest of this paper is organized as follows. Section 2 presents the related work followed by the main contribution resource allocated and scheduling models as well as the problem definition in Section 3. Section 4 gives a brief introduction to NPSO while and explains the proposed approach. Finally, Section 5 presents the evaluation of the algorithm followed by the conclusions and future work described in Section 6 and Section 7.

II. RELATED WORK

Scientific workflows, usually represented as Graph, are an important class of applications that lead to challenging problems in resource management on grid and utility resources systems. Workflows for large computational problems are often composed of several interrelated workflows grouped into ensembles. Workflows in an ensemble typically have a similar structure, but they differ in their input data and number of tasks, individual task sizes. There are many applications that require scientific workflow in single cloud provider in cloud environments.

In general, scheduling multitask workflows on any distributed computing resources (including clouds) is an NP-hard problem [10]. The main challenge of dynamic workflow scheduling on virtual clusters lies in how to reduce the scheduling overhead to adapt to the workload dynamics with heavy fluctuations. In a cloud platform, resource profiling and stage simulation on thousands or millions of feasible schedules are often performed, if an optimal solution is demanded. An optimal workflow schedule on a cloud may take weeks to generate.

Maria Alejandra Rodriguez et al. proposed the particle swarm optimization (PSO) method for solving complex problems with a very large solution space. Subsequently, the authors demonstrated that the PSO method is effective to generate a soft or suboptimal solution for most of reduces the cost and communication NP-hard problems [10].

In this paper, a new novel particle swarm optimization (NPSO) algorithm is proposed. The NPSO applies the OO method iteratively, in search of adaptive schedules to execute scientific workflows on multi cloud provider in cloud compute nodes with dynamic workloads [11]. During each iteration, the NPSO is applied to search for a suboptimal or good-enough schedule with very low overhead. From a global point of view, NPSO can process more successive iterations fast enough to absorb the dynamism of the workload variations. The initial idea of this paper was presented at the Deadline Based Resource Provisioning and Scheduling Algorithm [12] with some preliminary results. This paper extends significantly from the conference paper with some theoretical proofs supported by an entirely new set of experimental results.

III. MAIN CONTRIBUTIONS

In this paper main contribution are combined resource provisioning and scheduling strategies for executing scientific workflows on IaaS clouds. The scenario was modeled as an optimization problem which aims to minimize the overall execution cost while meeting a user defined deadline and was solved using the meta-heuristic optimization algorithm, PSO. The proposed approach incorporates basic IaaS cloud principles such as a usage-as-you-go model, heterogeneity, multi cloud, and cloud provider of the resources. Furthermore, our solution considers other characteristics typical of IaaS platforms such as performance variation and VM dynamic booting time. The experiments conducted with four well known workflows show that our solution has an overall better performance than the state-of-the-art algorithms. Furthermore, our heuristic is as successful in meeting deadlines as SCS, which is a dynamic algorithm. Also, in the best scenarios, when our heuristic, SCS and IC-PCP meet the deadlines, they are retable to produce schedules with lower execution costs.

IV. PROPOSED SCHEME

The proposed system contains all the existing system implementation. In addition, it extends the resource model to consider the data transfer cost between data centers so that nodes can be deployed on different regions. Extending the algorithm to include heuristics that ensure a task is assigned to a node with sufficient memory to execute it will be included in the algorithm. Also, it assigns different options for the selection of the initial resource pool. For example, for the given task, the different set of initial resource requirements is assigned. In addition, data transfer cost between data centers are also calculated so as to minimize the cost of execution in multi-cloud service provider environment. The main contribution of proposed system, the following problem solve in the existing system, they contribution are

- Adaptable in situations where multiple initial set of resource availability.
- Suitable for multiple cloud service provider environments.
- Data transfer cost is reduced between different cloud data centers.

PROPOSED NPSO ALGORITHMS

Input: Set of workflow task T, Initial Resources R,
Set Dimensional Particle ***dp,*** Set Entropy **ϴ,**
Set Optimal Best ***opbest,*** Set Optimal Global Best ***ogbest***

Output: Multi cloud Provider Scheduling

1. Set the dimension of the particle to ***dp***
2. Initialized the population of particles with random position and velocities
3. For each particle, calculated its Entropy values **ϴ**
   a. Compare the particle's Entropy **ϴ** value with the particle's ***opbest***.
      - If the current **ϴ** values is better than ***opbest*** then set ***opbest*** to the current value and location
   b. Compare the particle's Entropy **ϴ** value with Global best ***ogbest.***
      - If the current **ϴ** values is better than ***ogbest*** then set ***ogbest*** to the current value and location
   c. Update the position and velocity of the particle

$$V X_i (t+1) = V X_i (t) + V V_i (t)$$

4. Repeat from Step 3 until the stopping criterion is met

The range in which the particle is allowed to move is determined in this case by the number of resources avail able to run the tasks. As a result, the value of a coordinate can range from 0 to the number of VMs in the initial resource pool. Based on this, the integer part of the value of each coordinate in a particle's position corresponds to are source index and represents the compute resource assigned to the task defined by that particular coordinate. In this way, the particle's position encodes a mapping of task to resources.

V. PERFORMANCE EVALUATION

The experiments were conducted using different deadlines. These deadlines were calculated so that their values lie between the slowest and the fastest runtimes. To calculate these runtimes, two additional policies were implemented. The first one calculates the schedule with the slowest execution time; a single VM of the cheapest type is leased and all the workflow tasks are executed on it. The second one calculates the schedule with the fastest execution time; one VM of the fastest type is leased for each workflow task. Although these policies ignore data transfer times, they are still a good approximation to what the slowest and fastest runtimes would be. To estimate each of the difference between the fastest and the slowest times is divided by five to get an interval size. To calculate the first deadline interval we add one interval size to the fastest deadline, to calculate the second one we add two interval sizes and so on. In this way we analyze the behavior of the algorithms as the deadlines increase from stricter values to more relaxed ones.

The results **Fig 1.1** obtained for the PSO workflow and the NPSO algorithms are very similar to those obtained for the workflow. Its performance improves considerably for the 4th interval where it achieves a 100 percent hit rate.

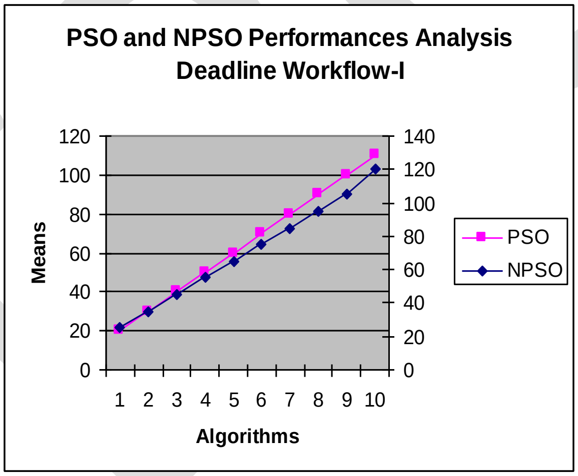

Fig 1.1 PSO Vs NPSO Performances-I

The results **Fig 1.2** obtained for the PSO workflow and the NPSO algorithms are very similar to those obtained for the workflow. Its performance improves considerably for the 3rd interval where it achieves a 100 percent hit rate.

The average execution costs obtained for each workflow are shown in **Fig. 1.3** We also show the mean PSO as the algorithms should be able to generate a cost-efficient schedule but not at the expense of a long execution time. The reference line on each panel displaying the average NPSO is the deadline corresponding to the given deadline interval. We present this as there is no use in an algorithm generating very cheap schedules but not meeting the deadlines; the cost comparison is made therefore, amongst those heuristics which managed to meet the particular deadline in a give case.

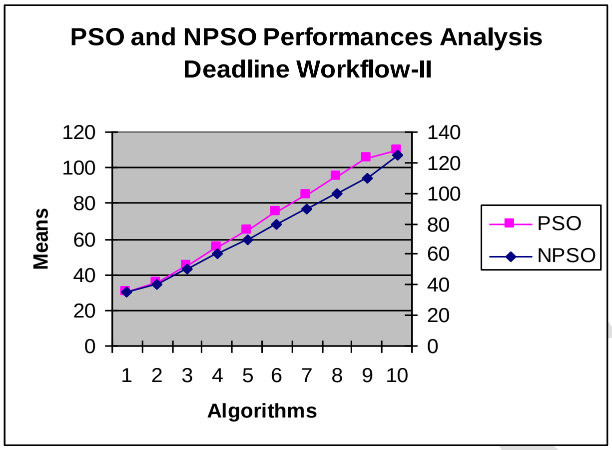

Fig 1.2 PSO Vs NPSO Performances-II

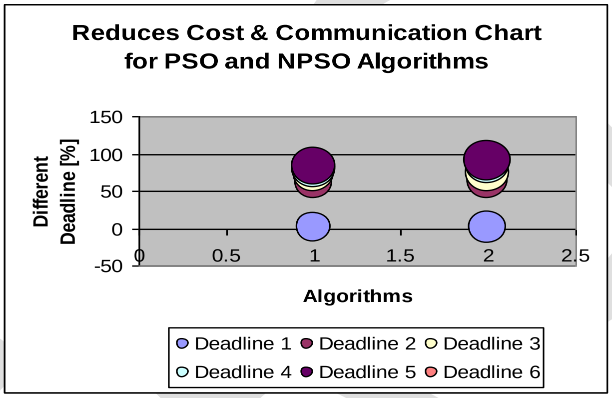

Fig 1.3 Reduces Cost and Communication for PSO and NPSO

ACKNOWLEDGMENT

My abundant thanks to **Dr.JAYASANKAR**, **Principal**, **Sri Vasavi College, Erode** who gave this opportunity to do this research work. I am deeply indebted to **Prof.B.MAHALINGAM M.Sc., M.Phil., (CS), M.Phil., (Maths), P.G.D.C.A., Head, Department of Computer Science at Sri Vasavi College, Erode** for this timely help during the paper. I express my deep gratitude and sincere thanks to my supervision **R.PRAGALADAN M.SC., M.Phil., Assistant Professor, Department of Computer Science** at **Sri Vasavi College , Erode** for her valuable, suggestion, innovative ideas, constructive, criticisms and inspiring guidance had enabled me to complete the work successfully.

VI.CONCLUSION

As research conclusion of proposed system work, they are exploring to different options for the selection in multi cloud environment in cloud provider. The cloud provider select the initial resource pool have been significant impact on the performance of the algorithm. We would also like to experiment with different optimization techniques and compare their performance with PSO. Finally, we aim to implement our approach in a workflow engine so that it can be utilized for deploying applications in real life environments.

VII. FUTURE ENHANCEMENTS

Cloud computing is broad field research problem; this paper describes the allocating resources workflow scheduling process from cloud provider in cloud environments. The proposed system is mainly contribution for allocated resources scheduling using novel particle swarm optimization algorithm. In this algorithm solve optimization problem and allocated the resources job workflow from multiple cloud provider in single cloud environments. In future, implementing new algorithm such as new AI application applied solve workflow scheduling problem in cloud environment.

REFERENCES:

[1] Velte, A., Velte, T., Elsenpeter, R., (2010), Cloud Computing: A Practical Approach, McGraw-Hill Osborne (Primary book to be used).

[2] Reese, G., (2009), Cloud Application Architectures: Building Applications and Infrastructure in the Cloud, O'Reilly, USA (Secondary book from which 3-4 chapters are likely to be used)

[3] Maria Alejandra Rodriguez and Rajkumar Buyya, "Deadline Based Resource Provisioning and Scheduling Algorithm for Scientific Workflows on Clouds", ieee transactions on cloud computing, vol. 2, no. 2, april-june 2014

[4] G. Juve, A. Chervenak, E. Deelman, S. Bharathi, G. Mehta, and K. Vahi, "Characterizing and profiling scientific workflows," FutureGeneration Comput. Syst., vol. 29, no. 3, pp. 682–692, 2012.

[5] P. Mell, T. Grance, "The NIST definition of cloud computing recommendations of the National Institute of Standards and Technology" Special Publication 800-145, NIST, Gaithersburg, 2011.

[6] R. Buyya, J. Broberg, and A. M. Goscinski, Eds., Cloud Computing: Principles and Paradigms, vol. 87, Hoboken, NJ, USA: Wiley, 2010.

[7] Y. Fukuyama and Y. Nakanishi, "A particle swarm optimization for reactive power and voltage control considering voltage stability," in Proc. 11th IEEE Int. Conf. Intell. Syst. Appl. Power Syst., 1999, pp. 117–121.

[8] C. O. Ourique, E. C. Biscaia Jr., and J. C. Pinto, "The use of particle swarm optimization for dynamical analysis in chemical processes," Comput. Chem. Eng., vol. 26, no. 12, pp. 1783–1793, 2002.

[9] T. Sousa, A. Silva, and A. Neves, "Particle swarm based data mining algorithms for classification tasks," Parallel Comput., vol. 30,no. 5, pp. 767–783, 2004.

[10] M. R. Garey and D. S. Johnson, Computer and Intractability: A Guide to the NP-Completeness, vol. 238, New York, NY, USA: Freeman, 1979.

[11] M. Rahman, S. Venugopal, and R. Buyya, "A dynamic critical path algorithm for scheduling scientific workflow applications on global grids," in Proc. 3rd IEEE Int. Conf. e-Sci. Grid Comput., 2007, pp. 35–42.

[12] L. Golubchik and J. Lui, "Bounding of Performance Measures for Threshold-baded Systems: Theory and Application to Dynamic Resource Management in Video-on-Demand Servers", IEEE Transactions of Computers, 51(4), pp 353-372, April, 2002.

[13] F. Zhang, J. Cao, K. Hwang, and C. Wu, "Ordinal Optimized Scheduling of Scientific Workflows in Elastic Compute Clouds", Third IEEE Int'l Conf. on Cloud Computing Technology and Science (CloudCom'11), Athens, Greece, Nov.29-Dec.1, 2011, pp. 9-17.

[14] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proc. 6th IEEE Int. Conf. Neural Netw., 1995, pp. 1942–1948