Bat Algorithm for Scheduling Workflow Applications in Cloud

S Raghavan
National Institute of Technology Karnataka, Surathkal
Mangalore, India
raghavan.sm2005@gmail.com

Marimuthu C
National Institute of Technology Karnataka, Surathkal
Mangalore, India
muthucwc.seopro@gmail.com

Sarwesh P
National Institute of Technology Karnataka, Surathkal
Mangalore, India
sarweshpj@gmail.com

K Chandrasekaran
National Institute of Technology Karnataka, Surathkal
Mangalore, India
kchnitk@ieee.org

***Abstract*— Workflow is one of the important aspects of cloud computing today. Cloud computing is one of the fastest growing technologies in the world. Workflows can be used in cloud as we use them in grid. Many operations in the cloud are based on workflow execution. Workflow systems are now becoming more complex and for such kind of systems efficient workflow management is important. Workflow scheduling is an important part of workflow management. Scheduling in general is NP-hard problem. To solve such kind of problems exhaustive methods cannot be used. Only non-exhaustive techniques can be used. In this paper we have used a metaheuristic approach called bat algorithm. Bat algorithm is specifically designed for optimizing hard problems. Here, bat algorithm with the help of binary bat algorithm is used for scheduling workflow in a cloud. Specifically the mapping of tasks and resources is done using this method. The optimal resources are selected such that the overall cost of the workflow is minimal.**

***Index Terms*—Workflow Scheduling, Cloud Workflow Scheduling, Bat Algorithm, Binary Bat Algorithm (BBA)**

I. INTRODUCTION

Today Cloud computing has become an integral part of the computing technology. The use of cloud is inevitable. From small jobs to big computational jobs the back end server is cloud server in more than half of the cases. There are two popular definitions of cloud; one classical definition is given by NIST [24] and another by Rajkumar Buyya which is a market oriented view of cloud. According to the Buyya et al. Cloud can be defined as a type of parallel and distributed system consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resource(s) based on service-level agreements established through negotiation between the service provider and consumers [8]. In the back end it involves a collection of interdependent tasks. These can be called as workflows. Workflows can be defined as automation of a process which involves many tasks; these tasks pass information to each other and follow certain set of rules to accomplish a single overall goal.

These workflows are closely related to grid technology. By 1980's the computational requirement started increasing at an exponential rate. This was because several scientific experiments in areas such as astronomy, physics, and biology involved a lot of data transfer and intense computation. These high computational tasks were big and hence will not be considered as single task. These are usually considered as set of tasks which pass data among themselves and work according to a set of rules to accomplish single goal, in other words these were called as workflows. To execute these workflows grids were used in the backend. Hence workflows and grid got related to each other. There were many tools and applications called as workflow managers which were developed namely DAGMan [13], Condor [15] etc.

Cloud computing is similar to grid computing. Both cloud and grid share certain properties and also there are properties which are not common to cloud and grid. But according tolin et al. [1] there are certain properties of cloud [in general] which make it one of the top choices for using it and at the same time one of the most difficult systems to be managed. There are several features of cloud which distinguishes itself from others [1]. They are as follows:

Service Oriented: Everything in cloud is given as a service. These services can be accessed over internet using a user interface (usually a web interface). This is one of the features which make cloud as one of the best suited technology for the market. This service oriented nature of cloud eventually affects the properties of a workflow [1].

Dynamic and on demand nature of cloud (Elasticity): A cloud is dynamic in nature. Services can be dynamically added or removed and there isn't any condition to predefine the number of resources or services to be used. Here services can be added on the go and while working with workflows a number of resources can be defined in runtime [1]. These are the major issues which makes cloud workflow different from other

technology. Due to these features of the cloud workflows it is difficult to design a workflow manager.

Workflow management revolves around scheduling. A good workflow scheduling algorithm has a considerable impact on the performance of the workflow manager [2]. Thus it is very important to have a workflow scheduling algorithm which is efficient.

Scheduling in general is a NP-hard problem. And thus for these kinds of problems it is not advisable to have an exhaustive solution as it becomes mostly impossible to get the perfect solution within a given time frame. Thus here we have chosen a metaheuristic approach to solve the problem. Bat Algorithm is a metaheuristic approach developed by Xin-She Yang [19]. This approach is based on echolocation of bats. This technique is similar to other metaheuristic techniques, in a way that it isn't exhaustive in nature. Hence, it can be extensively used for optimization. Bat algorithm is completely a probabilistic approach and is based on randomization.

Further the paper is divided into several sections: The second section gives an introduction about workflow scheduling; the third section explains the details regarding the related works that have been carried out previously for scheduling workflows in cloud. The fourth section describes Bat algorithm and it's working. The fifth section describes workflow scheduling using bat algorithm in cloud. The final section concludes the work with few future works.

II. WORKFLOW SCHEDULING

According to workflow management coalition [24], Workflow is concerned with the automation of procedures where information or tasks are passed between participants according to a defined set of rules to achieve, or contribute to, an overall business goal. This definition of workflow is business oriented and it need not be the same for cloud. Here in the case of cloud workflow the process need not be business process but a simple process that can be divided into several tasks. Workflow has several tasks and managing the overall workflow is considered to be a complex issue. Hence to efficiently manage workflow some of the important operations related to workflow need to be considered, one of such operations is workflow scheduling. It is a component of Workflow Management and can have a significant impact on the performance of workflow [2] .A scheduling algorithm is NP-hard problem. Thus, exhaustive approaches are not preferable for designing scheduling algorithms. Here workflow scheduling is considered. The workflow also requires scheduling and this is considered to be far more complex than a usual scheduling algorithm.

Workflow scheduling has several processes involved in it. Scheduling can be divided into two parts, one is task scheduling and another one is mapping the resources and tasks. Both the components are considered to be a part of scheduling. In the first part all the tasks are considered and they are scheduled according to the time. Several parameters such as arrival time, burst time and completion time are considered. Some of the algorithms of these types are Round Robin Scheduling, First Come First Serve (FCFS) etc. In the second part the resources and tasks are mapped together. The aim of efficiently mapping the tasks and resources is to reduce the total cost of workflow execution. A cloud consists of several resources and each resource has a cost associated with it. Based on certain parameters the cost of the resources may vary. Thus there needs to be a choice of resource such that the overall cost incurred due to mapping of resources and task to the cloud provider is less as far as possible and this mechanism should assure that all the Service Level Agreements (SLA) does not get affected. The makespan of the workflow may reduce or may not reduce but in many cases makespan is also considered.

III. RELATED WORK

There are several algorithms designed for scheduling workflow in cloud. The algorithms are discussed below.

A market oriented hierarchical scheduling strategy in cloud workflow system proposed by Wu, Zhangjun, et al. [7] is a new approach which takes into account both the aspects of workflow scheduling namely service level scheduling and task level scheduling. Here a hierarchical methodology is followed in which at the first level i.e. service level scheduling is done by using a package based random scheduling algorithm and the task level scheduling which maps the tasks to VMs is done using three prominent meta heuristic approaches namely Ant colony optimization (ACO), Genetic algorithm (GA) and Particle Swarm Optimization (PSO). All the three approaches are compared and Ant Colony optimization is found to be best according to the results. The comparison was done considering three basic parameters at the task level which eventually affects the Quality of Service (QoS) namely Number of CPU cycles, Makespan and Cost. The algorithm has been implemented using SwinDev-C [11] which is specifically designed for cloud.

Suraj Pandey et al [2] have proposed a particle swarm optimization based algorithm for scheduling workflow applications. The particle swarm optimization [23] technique was proposed by Eberhart and Kennedy in 1995 and is considered to be one of the best swarm intelligence techniques. It gives a better convergence with a few runs. This is because it adjusts individual value based on local best and the global best which is the key for optimization. The proposed algorithm mainly compares the costs required for mapping tasks in workflow by the proposed algorithm (PSO) and to total cost occurred when using Best resource Selection. According to their results the PSO based algorithm is three times more cost effective than Best Resource Selection algorithm. The implementation is done using Amazon EC2.

Similar to the above, Wu, Zhangjun, et al have used revised discrete particle swarm optimization for scheduling workflow in cloud [3]. This algorithm is similar to Particle Swarm Optimization but with a set based concept embedded in it. Here authors use GRASP (Greedy randomized adaptive search procedure) to initialize the swarm. The proposed algorithms

results are compared with Particle Swarm Optimization and Best Resource Select algorithms results based on makespan and cost optimization ratio. These results and statistics are obtained by varying computational cost and communication cost. According to the results RDPSO proves to be better. These algorithms have been implemented using Amazon elastic cloud compute.

The algorithms discussed till now consider only single workflow execution but there is another algorithm which considers scientific workflow ensemble [4]. Workflow ensemble is a collection of several interrelated workflows grouped together. Malawski, Maciej, et al. have proposed a method for cost and deadline constrained provisioning for scientific workflow ensemble in IaaS. Based on static and dynamic strategies the authors have considered three algorithms. The algorithms are namely Dynamic Provisioning and Dynamic Scheduling (DPDS), Workflow aware DPDS and Static Provisioning Static Scheduling. The first two algorithms are dynamic in nature i.e. these algorithms do resource provisioning and scheduling at run time but SPSS creates the resource provisioning and schedule plan before runtime. The implementation is done by using CloudSim [20].

Similarly, Abrishami, Saeid, Mahmoud Naghibzadeh, and Dick HJ Epema [5] have used same Partial critical path for scheduling workflow in IaaS. The properties of SaaS and IaaS are different and same approach cannot be followed. So based on the properties of IaaS the authors have proposed two approaches for solving the problem of workflow scheduling in IaaS. Maintaining QoS is still the main goal. This first approach named IC-PCP is one phase algorithm which uses two methods namely parentSchedule and pathSchedule repetitively used to schedule all the nodes. Here parentSchedule schedules all the unscheduled predecessors given the scheduled node as input and pathSchedule schedules all the tasks into a single instance of computational service with minimum price and considering the latest finish time. Similarly another approach called IaaS Cloud Partial Critical Paths with Deadline Distribution (IC-PCPD2).It is two phase approach with two phases namely Deadline Distribution and planning phase. Both of these phases uses parentSchedule and pathSchedule algorithm with some minor changes in it. These two approaches are compared for their efficiency and it is found that the IC-PCP performs better in several cases [5].

There is another algorithm that is proposed by Lin et al [1] which proposes solution for scheduling workflow in elastic cloud. There are two phases involved in workflow scheduling these are: task prioritizing phase and resource allocation phase. In the task prioritizing phase tasks are selected by using the proposed priority based algorithm. In the second phase for resource allocation a new algorithm by name Scalable Heterogeneous Earliest Finish Time is proposed. This algorithm uses several parameters such as Earliest Start Time, Earliest Finish Time, Earliest Ready time and Task Finish time. This algorithm is extension of Heterogeneous Earliest Finish time (HEFT) and is specifically proposed for elastic cloud.

There is another algorithm proposed by Genez, Thiago AL, Luiz F. Bittencourt, and Edmundo RM Madeira [22] for Scheduling Workflow cloud specifically SaaS workflows into Components of IaaS i.e. mapping the SaaS workflows which are used by Customer and the IaaS components of multiple IaaS which are present in the datacenter. This method involves two levels of SLA's. Integer Linear Program is formulated for solving the scheduling problem and they have also proposed two heuristics to obtain integer solution from the relaxed version of ILP. Both the algorithms are implemented and it is found that ILP gives a low cost solution in very less time and thereby achieving the goal of proposing an optimized scheduling process.

Above all there is another paper by which checks the efficiency with which scientific workflows runs in cloud. This paper in short checks the suitability of cloud to run scientific workflows. All the experiments are done based on the astronomical tool named montage [25]. The montage workflows are tested in three cloud environments namely local environment, virtual environment and wide area environment. According to their results it is found that the local environment is better for small workflows. But there is a scalability issue which is solved by using virtual environment. But on the whole, the wide area network (public) is found to be the best option for scheduling scientific workflow in cloud. In this paper the authors have proposed a new algorithm by name iterative ordinal optimization. This method was designed to reduce the scheduling overhead and to reduce the time taken by the algorithm to schedule a complex task. The authors have compared this methodology with Monte Carlo method and blind pick methods. The algorithm proves to be better than them. The authors have done a detailed study and have demonstrated the working of scientific workflows in different clouds. According to the results the different types of clouds doesn't have any specific significant differences. They also test scientific workflow manager pegasus [16] over different cloud infrastructure and come to the conclusion that sky computing supports scientific workflow scheduling.

IV. BAT ALGORITHM

In this paper we develop a workflow scheduling algorithm based on a heuristic approach called Bat algorithm. Bat algorithm is designed by Xin-She Yang [19]. This algorithm is a metaheuristic algorithm which is designed for optimization purposes. This is one of the recently developed technologies which was primarily aimed at solving engineering problems. Metaheuristic algorithms are known for optimization. There are several other metaheuristic algorithms that are designed for optimization. The most basic problem of optimization is maximization or minimization of a function. Many metaheuristic algorithms concentrate on this aspect of optimization. Usually a good algorithm is characterized by the capability of maximizing or minimizing more accurately. Basically optimization is used for the problems which do not have exhaustive solution. Thus the best optimality involves finding out the best solution (Optimal solution), usually by means of randomization.

Similarly bat algorithm uses randomization for calculating the optimal value. To prove the superiority of an algorithm, an algorithm is tested with several standard functions available such as Rastrigin's function etc. If an algorithm is able to minimize these standard functions with utmost accuracy, then these algorithms are considered to be good algorithms. In this context the Bat algorithm, according to the authors is the best algorithm as it is one of the few algorithms which gives accurate results. This is one of the prime reasons for choosing this algorithm for workflow scheduling. Mapping of tasks and process units in a workflow is an important step. The mapping process is an NP-hard problem. These NP-hard problems cannot be solved in exhaustive manner i.e. it is not possible by any algorithm to solve it exhaustively within polynomial time. Hence a non exhaustive approach is suitable for these kinds of problem and Bat algorithm is the obvious choice for this as it is the best among all the metaheuristic approaches.

Bat algorithms a type of Swarm intelligence algorithm. Bat algorithm was designed to incorporate properties of several other metaheuristic algorithms so that it could perform better than other algorithms by considering the advantages and disadvantages of all the other algorithms. These algorithms include the simulated annealing, Harmony search and Particle swarm optimization. The bat algorithm achieves this and is considered to be a good algorithm which has the positives of other metaheuristic algorithms.

The whole bat algorithm described below was developed by Xin-She Yang and is described as mentioned in his paper [19]. Bat algorithms are based on echolocation techniques of bat (specifically Microbats). The bats usually rely on tiny prey and thus to find the prey by using echo which is a high frequency ultrasonic sound. The sound is very loud. This sound is used intermittently at different time intervals to catch the prey. This process is used as the base for designing this algorithm. There are several parameters involved in this algorithm such as frequency, wavelength, loudness and velocity which is constant at 340 m/s. The sound is usually very loud and it varies according to the distance between the bat and the prey. There are three main assumptions or rules according to the algorithm are [19]:

The bats are considered to know the difference between the prey and food automatically and they use echolocation to estimate the distance.

The bats have a velocity v_i and their position is x_i, They fly randomly with a fixed frequency and varying wavelength and with a loudness A_0 to search for prey.

They have capability of automatically adjusting the wavelength or frequency of the emitted pulses and can control the rate of emission of pulses automatically based on the distance between them and they prey.

The loudness is considered only between the limits A_0 which is the upper limit and A_{min} which the lower limit. The addition to above mentioned assumptions the frequency is directly converted into whole number wavelengths to reduce the complexity (by considering velocity of sound as 340m/s).

Bat motion [19] in bat algorithm is defined as follows; this is primarily based on the velocity v_i, position x_i and frequency of the bat f_i.

$$f_i = f_{min} + (f_{max} - f_{min})\beta \quad (1)$$

$$v_i^t = v_i^{t-1} + (x_i^{t-1} - x_*)f_i \quad (2)$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (3)$$

Where beta (β) is a random vector drawn from a uniform distribution between [0,1]. Here x_* is the current global best. The whole algorithm is based global and local searches.

The Bat Algorithm is iterative in nature, hence for each iteration the parameters needs to be updated. After the position and velocity are updated the loudness and pulse emission rate needs to be updated. As the distance between the bat and the prey changes the bat has to automatically change the loudness and rate of pulse emission. The algorithm is described in detail by Xin-She Yang in his paper [19].

Bat Algorithm

Objective function $f(x), X = (x_1, \ldots . x_d)^T$
Initialize the bat population $x_i (i = 1,2, \ldots, n) and\ v_i$
Define pulse frequency $f_i\ at\ x_i$
Initialize pulse rates r_i *and the loudness* A_i
while *(t < Max number of iterations)*
Generate new solutions by adjusting frequency, and updating velocities and locations/solutions
    if *(rand>* r_i*)*
    Select a solution among the best solutions
    Generate a local solution around the selected best solution
    end if
    Generate a new solution by flying randomly
    if *(rand <* A_i *&* $f(x_i) < f(x_*)$ *)*
    Accept the new solutions
    Increase r_i *and reduce* A_i
    end if
Rank the bats and find the current best x_*
end while
Results / Visualizations

Figure 1 Bat Algorithm adapted from [19]

The algorithm above is described for continuous optimization but discrete set of inputs are given here, so as to suit the application, changes are made with reference to the Binary Bat Algorithm (BBA) [18]. Binary Bat algorithm is based on bat algorithm with a slight difference, bat algorithm is for continuous values but Binary Bat Algorithm can be used to select from Binary values. BBA algorithm is proposed for the feature selection. For feature extraction continuous values are not used for optimization. Here, few modifications in binary

bat algorithm are made and with bat algorithm it is used here for selecting the optimal cost.

V. SCHEDULING WORKFLOWS AND RESULTS

Workflow is an important part of cloud. These workflows are extensively used in cloud. Whenever it comes to workflows its scheduling becomes an important part. There are several aspects of workflow scheduling in cloud. Here we specifically concentrate on task to resource mapping. The main aim of the proposed algorithm is to map the tasks to resources efficiently and that the total overall cost incurred is reduced. Efficient workflow scheduling can improve the workflow management significantly [2]. Workflow Scheduling in cloud is different from other scheduling because of several reasons as mentioned by Lin et al. [1].

Scheduling Algorithm

Initialize or determine cost for different compute resources.
Calculate the cost of each task (T_i) on each compute resource R_j.
All the possible mappings $T_i X\ R_j$ need to be made.
Determine the minimal cost value for the tasks on different resources according to Bat algorithm with a condition that a resource doesn't have more than one tasks.
Based on the above results assign the tasks.
Get the final overall minimal cost.

Figure 2 Scheduling Algorithm for Workflow Applications

Workflow scheduling depicted in figure 2 involves one important operation of mapping the resources and the tasks. The resources and the tasks are mapped based on the total cost of computation i.e. out of collection of resources; the resources should be assigned to the tasks based upon the total overall cost for executing the whole workflow. Thus eventually each workflow will use the resources optimally to have least cost. This process of mapping the resources and tasks is based on the bat algorithm; specifically using the bat algorithm the resources are selected and assigned to the corresponding tasks. The main aim of algorithm is to minimize the cost. Here only cost of processing is considered. In this particular example a workflow consisting of 4 tasks is considered each task is assigned resources. The resources are of different capacity (MIPS). Based on the MIPS each resource has different cost ($/hour).

In this paper bat algorithm is used for selecting the minimal cost among all the total sort combinations. This selection of minimal value is done by using bat algorithm. The bat algorithm is described in the previous section. The Bat algorithm can be readily used for minimization of continuous values but for discrete values some additions and changes need to be made. These additions are similar additions that are done for feature extraction in paper titled Binary bat algorithm by Nakamura, Rodrigo YM, et al [18]. The main aim is to obtain a minimal cost value for the whole workflow.

Here pair of compute resource and task is mapped together; the scheduling algorithm depicts the actual algorithm for scheduling which involves mapping. The first step is to determine the cost of resources (cost/hr) based on the capacity. Then calculate the cost of each task T_i over each resource R_j and based on the values $T_i X R_j$ compute the optimal minimal value using bat algorithm. Bat algorithm will calculate the optimal value by considering all the resources with no more than one task for each resource. Based on the result obtained the tasks are assigned.

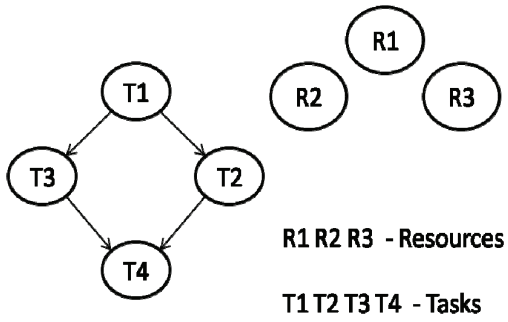

Figure 3 Task and Resource Mapping

The experiment consists of four tasks and three resources as described in figure 3. Each resource has different capacity and so has different cost. A sample set of resources with different cost is considered and the tasks are considered. After mapping the following task to the resource a list is obtained. The tasks are selected based on minimal value using the bat algorithm. In this algorithm each task is assigned to a resource such that not more than one task uses a resource. Here overall minimal cost is considered rather than considering the cost for executing each task separately. For comparison and to ascertain the superiority of bat algorithm, Best Resource Selection algorithm is used. The best resource selection algorithm maps the best resource to the task i.e. it assigns tasks with least time to execute [2] . This doesn't consider all the task in the workflow, it will consider the best resource only for the current task i.e. one task at a time. Thus overall efficiency is not considered.

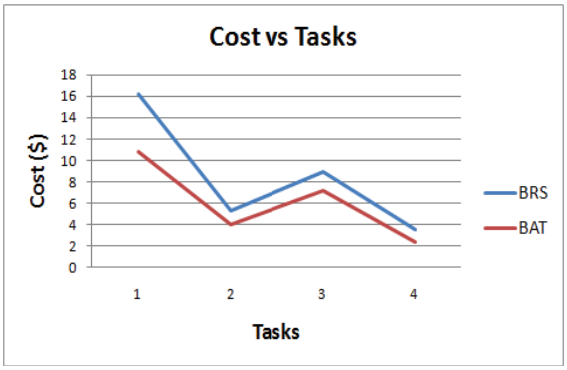

Figure 4 Cost comparisons of BRS and Bat scheduling algorithm

For comparison, preassigned unit cost is considered. Here, each task is different with different complexity. Thus based on

its complexity the required time of execution varies. On each resource the tasks will have different costs of execution. Eventually the overall cost required for execution of workflow is calculated. The figure 4 depicts the difference between the cost ($) in BRS and in Bat algorithm.

From the graph it is evident that bat algorithm gives a minimal cost when compared to the BRS algorithm. The algorithm on an average gives an overall improvement of fifty percent over the BRS algorithm.

VI. Conclusion and Future Work

Workflow scheduling is a NP-hard problem. To solve these kinds of problems exhaustive techniques cannot be used. Hence a non exhaustive optimization technique is better choice for solving such problems. Thus in this paper we have used metaheuristic algorithm called Bat algorithm and some aspects Binary Bat Algorithm for scheduling workflow applications in cloud. Bat algorithm is better than other metaheuristic algorithms. The Bat algorithm is compared with BRS algorithm and the results show that the bat algorithm is fifty percent better than BRS algorithm considered here. Further as a future work transportation cost can be considered and further realtime analysis of this algorithm can be done.

References

[1] Lin, Cui, and Shiyong Lu. "Scheduling scientific workflows elastically for cloud computing." *Cloud Computing (CLOUD)*, 2011 IEEE International Conference on. IEEE, 2011.

[2] Pandey, Suraj, et al. "A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments." *Advanced Information Networking and Applications* (AINA), 2010 24th IEEE International Conference on. IEEE, 2010.

[3] Wu, Zhangjun, et al. "A revised discrete particle swarm optimization for cloud workflow scheduling." *Computational Intelligence and Security (CIS),* 2010 International Conference on. IEEE, 2010.

[4] Malawski, Maciej, et al. "Cost-and deadline-constrained provisioning for scientific workflow ensembles in IaaS clouds." *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. IEEE Computer Society Press, 2012.

[5] Abrishami, Saeid, Mahmoud Naghibzadeh, and Dick HJ Epema. "Deadline-constrained workflow scheduling algorithms for Infrastructure as a Service Clouds." *Future Generation Computer Systems 29.1 (2013):* 158-169.

[6] Wang, Xiaofeng, et al. "Optimizing the makespan and reliability for workflow applications with reputation and a look-ahead genetic algorithm." *Future Generation Computer Systems 27.8 (2011*): 1124-1134.

[7] Wu, Zhangjun, et al. "A market-oriented hierarchical scheduling strategy in cloud workflow systems." *The Journal of Supercomputing* 63.1 (2013): 256-293.

[8] Buyya, Rajkumar, et al. "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility." *Future Generation computer systems* 25.6 (2009): 599-616.

[9] Xu, Meng, et al. "A multiple QoS constrained scheduling strategy of multiple workflows for cloud computing." *Parallel and Distributed Processing with Applications,* 2009 IEEE International Symposium on. IEEE, 2009.

[10] Moens, Hendrik, Koen Handekyn, and Filip De Turck. "Cost-aware scheduling of deadline-constrained task workflows in public cloud environments." *Integrated Network Management (IM 2013),* 2013 IFIP/IEEE International Symposium on. IEEE, 2013.

[11] Liu, Xiao, et al. "SwinDeW-C: a peer-to-peer based cloud workflow system." *Handbook of Cloud Computing. Springer US, 2010.* 309-332.

[12] Abrishami, S., and M. Naghibzadeh. "Deadline-constrained workflow scheduling in software as a service Cloud." *Scientia Iranica* 19.3 (2012): 680-689.

[13] DAGMan [Online] Available: http://research.cs.wisc.edu/htcondor/dagman/dagman.html

[14] Workflow management Coalition [Online] Available: http://www.wfmc.org/

[15] Couvares, Peter, et al. "Workflow management in condor." Workflows for e-Science. Springer London, 2007. 357-375.

[16] Altintas, Ilkay, et al. "Kepler: an extensible system for design and execution of scientific workflows." *Scientific and Statistical Database Management,* 2004. Proceedings. 16th International Conference on. IEEE, 2004.

[17] Pegasus [Online] Available: http://pegasus.isi.edu/wms/docs/latest/execution_environments.php

[18] Nakamura, Rodrigo YM, et al. "BBA: A binary bat algorithm for feature selection." *Graphics, Patterns and Images (SIBGRAPI)*, 2012 25th SIBGRAPI Conference on. IEEE, 2012.

[19] Yang, Xin-She. "A new metaheuristic bat-inspired algorithm." *Nature inspired cooperative strategies for optimization (NICSO 2010)*. Springer Berlin Heidelberg, 2010. 65-74.

[20] Calheiros, Rodrigo N., et al. "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms." *Software: Practice and Experience* 41.1 (2011): 23-50.

[21] Chen, Weiwei, and Ewa Deelman. "Workflowsim: A toolkit for simulating scientific workflows in distributed environments." E-Science (e-Science), 2012 *IEEE 8th International Conference on. IEEE, 2012.*

[22] Genez, Thiago AL, Luiz F. Bittencourt, and Edmundo RM Madeira. "Workflow scheduling for SaaS/PaaS cloud providers considering two SLA levels." *Network Operations and Management Symposium (NOMS)*, 2012 IEEE. IEEE,

[23] Kennedy, James, and Russell Eberhart. "Particle swarm optimization." *Proceedings of IEEE international conference on neural networks*. Vol. 4. No. 2. 1995.

[24] Mell, Peter, and Timothy Grance. "The NIST definition of cloud computing (draft*)." NIST special publication* 800.145 (2011): 7.

[25] Montage : An Astronomical Image Mosaic Engine [Online] Available: http://montage.ipac.caltech.edu/docs/gridtools.html