Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/13S137

Improved Hyper Elliptic Curve Cryptography with Hybrid Bat Algorithm for Tasks Replication to Meet Deadlines in Clouds

A. Ramathilagam[1,*] *and S. Maheswari*[2]

[1]Department of Computer Science and Engineering, P.S.R.Engineering College, Sivakasi, India
[2]Department of Computer Science and Engineering, National Engineering College, Kovilpatti, India

Received: 2 Mar. 2019, Revised: 2 May 2019, Accepted: 11 May 2019
Published online: 1 Aug. 2019

Abstract: This current work proposes a Hybrid Bat and Differential Evolution Algorithm (HBDEA) based on IaaS Cloud Partial Critical Path (IC-PCP) Replication and Improved Hyper Elliptic Curve Cryptography (HBDEAIPR with IHECC). The proposed HBDEAIPR algorithm applies a deadline constraint and a variable budget for replication to achieve its goals. Proposed Enhanced IC-PCP with Replication (EIPR) algorithm is increasing the likelihood of completing the execution of a scientific workflow application within a user-defined deadline in a public Cloud environment. The proposed HBDEA algorithm determines the parameters of tasks are the early start time and latest finish time to replicate workflow tasks to mitigate effects of performance variation of resources so that soft deadlines can be met. The proposed HBDEAIPR approach is verified by developing a IHECC system which authenticates the user not only based on task deadlines. Also, it uses secret-key and encrypted value for genuine users authentication in CC environment. Simulation experiments with well-known scientific workflow show that the proposed HBDEAIPR approach increases the likelihood of deadlines being met and reduces the total execution time of applications as the budget available for replication increases.

Keywords: Cloud computing, Scientific workflows, task Replication, soft Deadline, bat algorithm, hybrid hyper elliptic curve cryptography

1 Introduction

Scientific workflows are described as Direct Acyclic Graphs (DAG) whose nodes represent tasks and vertices represent dependencies among tasks. Because a single workflow can contain hundreds or thousands of tasks [1], this type of application can benefit large-scale infrastructures. Among such infrastructures, over the past decade, advances in commodity computing and virtualization technologies have enabled the cost-effective realization of large-scale data centers that run large portion of today's Internet applications and backend processing. The economies of scale that arose allowed data center infrastructure to be leased profitably to third parties. Cloud Computing (CC) [2]-[5], quantum computing [6]-[12], neural computing [10,11] and DNA computing [13,14] are different ways of computations. Thus emerged the Cloud Computing (CC) paradigm, where in a pool of computing resources is shared between applications that are accessed over the Internet. CC has become a broad and popular term and applications, used not only by the technology community but the general public as well. It refers to applications delivered over the Internet, as well to hardware and system software residing in data centers that host those applications.

Research related to different aspects of CC has accelerated steadily over the last three to five years; workshops and conferences have been held and delivered and an increasing number of publications are being produced. General introductions into the field and its research challenges have been studied in [15]-[16]. In the recent work [17] discusses resource management challenges and techniques with the perspective of VMwar offerings. A topic that is receiving considerable attention is data center networking, which has been surveyed in the recent works [18]-[19]. This is because these infrastructures are available in a pay-per-use system and can provide dynamic scaling in response to the needs of the application (a propriety known as elasticity).

* Corresponding author e-mail: nrsd20001@gmail.com

© 2019 NSP
Natural Sciences Publishing Cor.

Therefore, resources for execution of the workflow can be provisioned on demand, and their number can be increased if there are enough budgets to support it. This Cloud utilization model, where users obtain hardware resources such as Virtual Machines (VMs), where they deploy their own applications, it is called Infrastructure as a Service IaaS.

These capabilities of clouds make them a suitable platform to host deadline-constrained scientific workflows. In this class of workflows, a specific soft deadline for completion of the workflow is assigned, along with the workflow specification, during the application submission. A soft deadline is a deadline that, when unmet, does not render the computation useless [16]. Thus, although the maximal value of the computation is achieved when the deadline is met, investment is not lost if the deadline is missed by small margins.

As the execution of the workflow in the cloud incurs financial cost, the workflow may also be subject to a budget constraint. Although the budget constraint of a workflow may be a limiting factor. Its structure also imposes a significant limitation. This is because dependencies among task and the number of tasks ready for execution in a given time may limit the amount of resources being used in parallel for executing the workflow. Because cloud resource providers charge resource utilization by integer time intervals, such a limitation in the workflow scalability causes situations where cloud resources are available (i.e., their allocation time interval is paid and there are still time before it expires) but no task is ready to be executed.

Workflow scheduling is one of the task in the CC. It tries to map the workflow tasks to the VMs based on different functional and non-functional requirements [20]. A workflow consists of a series of interdependent tasks, which are bounded together through data or functional dependencies. These dependencies should be considered in the scheduling [21]. However, workflow scheduling in the cloud computing is a NP-hard optimization problem and it is difficult to achieve an optimal schedule. Because there are numerous VMs in a cloud and many user tasks should be scheduled by considering various scheduling objectives and factors.

The common objective of the workflow scheduling technique is to minimize the makespan by the proper allocation of the tasks to the virtual resources [22]-[23]. For example, a scheduling scheme may try to support the promised Service Level Agreements (SLAs), the user specified deadlines and cost constraints. Also, scheduling solutions may consider factors such as resource utilization, load balancing and availability of the cloud resources and services in the scheduling decisions [24]-[25].

A critical challenge in integrating workflow systems with resource provisioning technologies is to determine the right amount of resources required for the execution of workflows. The resource capacity affects the total execution time of application workflow and determines the financial cost. Then, an efficient scheduling algorithm is required to optimally dispatch tasks to the cloud resources. It consists of taking the decision for mapping tasks to computing resources by optimizing performance metrics such as time and cost execution. These resources are in units of Virtual Machines (VMs) instances.

Task scheduling is a NP-complete problem in the general form. So, various approaches based on heuristics have been proposed for scheduling workflows [26]-[27]. Nevertheless, this problem is still an open research challenge and requires some efforts to reach an optimum solution. Each approach takes into account some metrics to ensure resources provisioning. However, they implement limited contingency strategies to correct delays caused by underestimation of tasks execution time and inefficient management of cloud resources with less-quality cloud applications. In order to overcome the above mentioned problems, this research work presents a new cloud resource management approach which is not only automate the selection of an appropriate cloud but also implements dynamic resource allocation. This work proposes a Hybrid Bat and Differential Evolution Algorithm (HBDEA) based on IaaS Cloud Partial Critical Path (IC-PCP) Replication and Improved Hyper Elliptic Curve Cryptography (HBDEAIPR with IHECC).

2 Literature Review

The resources allocation problem for workflow applications is one of the most difficult challenges in the cloud. In fact, the type and the number of allocated resources affect the execution time of workflow and determine the financial cost. Many researchers have come up with new ways to face this challenge:

Caron et al.(2012) [27], presented two original allocation strategies for non-deterministic workflows in the cloud computing under budget constraints. The proposed approach consists in transforming the scheduling problem into a set of smaller and well studied sub-problems. Concretely, it decomposes the non-deterministic workflow in input into a set of deterministic sub-workflows.

Jianfang et. al [28] proposed to solve the problems of security threats on workflow scheduling in the cloud. It consists in quantizing the security of tasks and VMs in the cloud workflow scheduling and quantizing the users satisfaction degree to tasks assigned to VMs. Then, it establishes a scheduling model considering security, completion time and cost in the cloud workflow scheduling.

Varalakshmi et al. [29] proposed an Optimal Workflow based Scheduling (OWS) algorithm for scheduling workflows in the cloud. First, the Resource discovery algorithm, indexes all the resources and this helps in locating the free resources. Second, the scheduling algorithm that takes user specified QoS

© 2019 NSP
Natural Sciences Publishing Cor.

parameters (execution time, reliability, monetary cost etc.) as a key factor used for scheduling workflows. Using a special metric called the QoS heuristic, the sub-task cluster is assigned to its optimal resource. Third, in case that resources are not available for allocating to a task, compaction is performed. By using this, a significant improvement in CPU utilization is achieved in CC environment.The approaches already stated are not able to exploit the elasticity of resources offered by the cloud. In fact, the selection of resources is determined before applications start. Although, there may be variations in the cloud environment during the workflow execution such as the unavailability of resources, the change of communication time between resources and so on. Thus, a dynamic mode is needed for scheduling workflow activities in the cloud.

Nagavaram et al. [30] used a time constraint to drive the resources allocation. The main idea in this module is as follows. First parallelize the search method used in this algorithm. Next, create a flexible workflow using the Pegasus Workflow Management System. Finally, they add a new dynamic resource allocation module, which can use a fewer or a larger number of resources based on a time constraint specified by the user. Evaluate proposed implementation using several different datasets, and show that the application scales quite well, and that the dynamic framework is effective in meeting time constraints.

Rahman et al. [31] presented an Adaptive Hybrid Heuristic (AHH) for workflow scheduling in hybrid cloud environment. AHH is not only capable of adapting to changes in the cloud but also able to meet user's budget and deadline. It is designed to first generate a task-to-resource mapping with minimum execution cost using Genetic Algorithm (GA) within user?s budget and deadline. This initial schedule is then utilized to distribute the workflow-level budget and deadline to task levels. Finally, the Dynamic Critical Path (DCP) heuristic is employed to dynamically schedule the ready tasks level-by-level based on the initial schedule, budget and deadline constraints, as well as changed status of resources.

Pandey et al. [32] presented a Particle Swarm Optimization (PSO) based heuristic to schedule applications to cloud resources that takes into account both computation cost and data transmission cost. Experiment with a workflow application by varying its computation and communication costs. Compare the cost savings when using PSO and existing Best Resource Selection (BRS) algorithm. The results show that PSO can achieve: a) as much as three times cost savings as compared to BRS, and b) a good distribution of workload onto resources. Additionally, when the remote resource management systems are not able to allocate a task to the resources due to the resource unavailability, the recomputation of the PSO makes the heuristic dynamically balance other tasks' mappings.

Verma and Kaushal [33] presented Budget Constrained Priority based Genetic Algorithm (BCPGA) to optimize the execution cost and data transfer cost with a budget constraint. Each workflow's task is the assigned priority using bottom level (b-level) and top level (t-level). BCPGA, at first, calculates the b-level and t-level of all workflow tasks and creates the initial population which for all individuals the priority of each task is set equal to the total of its b-level. Then, all the tasks are assigned to the available VMs with their priority. While termination criteria are not met, BCPGA evaluates the fitness of the individual in the population, afterwards it applies the selection operator to select the parent. Then, it applies the crossover operator on the selected parent using crossover probability to create the children and applies the mutation operator on the newly created children. Then, it validates each child according to the fitness function. Finally, it adds the valid child to create the new population.

Liu et al. [34] presented a novel compromised-time-cost scheduling algorithm which considers the characteristics of cloud computing to accommodate instance-intensive cost-constrained workflows by compromising execution time and cost with user input enabled on the fly. The simulation performed demonstrates that the algorithm can cut down the mean execution cost by over 15% whilst meeting the user-designated deadline or shorten the mean execution time by over 20% within the user-designated execution cost.

Yassa et al. [35] combined the Ant colony and Max-min algorithm and focuses on total processing time and cost. They tried to balance the total system load and minimize the total makespan. In the Max-min algorithm, large tasks have higher priority than smaller ones. This scheme reduces the waiting time of the short jobs by assigning large tasks to the slower resources. Thus, small tasks are executed concurrently on the fastest resource to finish large number of tasks during finalizing at least one large task on the slower resource. In max min, if it cannot execute the tasks concurrently, makespan becomes larger. To overcome such limitations, a new modification is applied for the Max-min scheduling algorithm. This approach improves the total cost and time factor and is only concerned with the number of the resources and tasks. Improved max min provides an optimal solution during the preliminary stage. Furthermore, during the starting stage of ant algorithm, the searching speed is very slow for the lacking of pheromones, but after the pheromones reach a certain degree, the speed of the optimal solution improves quickly.

Byun et al. [36] proposed the Partitioned Balanced Time Scheduling (PBTS) algorithm for cost-optimized and deadline-constrained execution of the workflow applications on Clouds. The PBTS algorithm considers only one type of cloud resource, chosen a priori, for its provisioning and scheduling decision. Abrishami et al. [37] proposed two algorithms for cost-optimized, deadline-constrained execution of workflows in clouds.

© 2019 NSP
Natural Sciences Publishing Cor.

These algorithms do not consider all the data transfer times during provisioning and scheduling, increasing the execution budget. Proposed algorithm is based on one of such algorithms (called IC-PCP), but also accounts for data transfer times and Cloud resources boot time during the provisioning and scheduling process. Furthermore, it explores possibility of tasks replication to increase the probability of meeting application's deadlines.

3 Applications and System model

Consider workflow applications composed of dependent tasks and modelled as Directed Acyclic Graphs (DAGs), where each task can only start its execution after all its predecessors have finished and the data have been transferred to the machine where it is scheduled to execute. Each workflow task has a computational demand associated, which is translated into how long it takes to run according to the Central Processing Unit (CPU) capacity of each VM. Also, the data transmission between two tasks occurs in the network that connects the VMs where these two tasks are scheduled. If they are scheduled to the same VM, the data transmission time between them is zero. During the workflow execution, the necessary data to each task can be kept into the storage available in the VM where the task is running, or it can be stored in extra storage from rented storage facilities. While the VM-storage cost is included in the VM price, the extra storage is charged independently. Moreover, this extra storage is deployed over the network, and therefore there exists a maximum transfer speed from this storage to the VM where the task is being run.

A scientific workflow application is modelled as a DAG $G=(T,E_T)$, where T is the set of tasks that compose the workflow and E_T is the set of dependencies between tasks. Dependencies are in the form of edges $e_{(i,j)}=(t_i,t_j)$, $t_i,t_j \varepsilon T$, that establish a task t_j that depends on the data generated by t_i for its execution, and therefore t_j cannot start before the execution of t_i completes and data generated by the latter is transferred to the location where t_j will be executed. Task t_i is a parent task of t_j and t_j is a child task of t_i. Tasks without parents are called entry tasks and tasks without children are called exit tasks. For the correct operation of the proposed algorithm, assume that a workflow can have only one entry task and one exit task. This can be achieved with the insertion of dummy tasks t_{entry} and t_{exit} that have execution time equals to 0. All the actual entry tasks are children of t_{entry} and all the actual exit tasks are parents of t_{exit}. The sets of parents and children of a task t_j are given respectively by functions $parents(t_j)$ and $children(t_j)$. Each workflow G has a soft deadline dl(G) associated to it. It determines the time to complete its execution, counted from the moment it is submitted to the workflow scheduler. The latter manages the execution of the workflow, makes decision on allocation of Virtual Machines (VMs), and schedules and dispatches tasks for execution in the Cloud. Cloud provider offers a set of n VM types denoted by $\overrightarrow{VM}=vm_1,...,vm_n$. Each VM type offers different amount of resources, and incurs a different cost per use. Let $\overrightarrow{C}=c_1,c_2,...,c_n$ be the cost vector associated with the use of each VM. VMs are charged per integer amount of time units, and partial utilization of a time period incurs charge for the whole period. Therefore, if the time period is one hour, utilization of a VM per 61 minutes incurs in the payment of two hours. There is no limit imposed on the number of VMs of each type that can be running in any moment for execution of the workflow.

Runtime of each task is defined in the runtime matrix R. An element r_{jk} of R specifies the estimated runtime of task t_j in A VM of $typevm_k$. The minimum runtime $R_{min}(t_i)$ of a task t_i is the smallest runtime for such a task in the matrixR. Notice that $r_{entryk}= r_{exitk}= 0$ for all k. Tasks cannot be pre-empted or check pointed. Therefore, if the execution of a task fails or if a task is cancelled by the scheduler, it has to be restarted. Each edge $e_{(i,j)}$ of G has an associated data transfer time D(i,j). This is the amount of time required to transfer the data required by the non-entry and non-exit task t_j from the VM where t_i is running to the VM where t_j is running. Notice that, if both t_i and t_j are running on the same VM, D(i,j)=0. The existence of data transfer time among different VMs implies that, for each task t_j to be executed in a given VM, vm_k is deployed before the data transfer from parents of t_j start, and is decommissioned after all the data transfers to its children are completed.

Important parameters of tasks are the early start time (est) and latest finish time (lft). The former represents the earliest time a task is able to start, which happens when all its parent tasks finish as early as possible and the latter represents the latest time a task can finish without missing the deadline, which happens when all the children of a task are executed as late as possible. Formally, est and lft are defined as:

$$est(t_j)=\begin{cases}0, \text{if } t_j=t_{entry} \\ \max_{t_a \varepsilon parents(t_j)}(est(t_a)+R_{min}(t_a)+D(e_{a,j})) \\ \text{otherwise}\end{cases} \quad (1)$$

$$ift(t_j)=\begin{cases}dl(G), \text{if } t_j=t_{exit} \\ \max_{t_s \varepsilon children(t_j)}(lft(t_s)-R_{min}(t_s)+D(e_{j,s})) \\ \text{otherwise}\end{cases} \quad (2)$$

The schedule time $st(t_j)$ of a task t_j is the time on which the task has been scheduled for execution. This parameter is defined during the scheduling process, and can assume any value between $est(t_j)$ and $lft(t_j)$. For this problem to be solved, three sub-problems have to be solved, namely provisioning, scheduling and authentication. The provisioning problem consists in the determination of the optimal number and type of VMs that can complete the workflow within its deadline. The scheduling problem consists in the determination of the placement and order of execution of the different tasks that compose the workflow in the VMs selected during the provisioning stage. The provisioning, scheduling and authentication problems are interconnected, as a different decision in types and number of machines may result in a different scheduling of tasks. Analyzing the security models are used for the interfaces of the cloud provider.

4 Methodology

Utilization of the runtime matrix R in the model implies a known a stable execution time for each task that

© 2019 NSP
Natural Sciences Publishing Cor.

composes the workflow. Similarly, the data transfer time function D assumes a stable data transfer time between VMs considers the workflow and cloud VM model. In this research, a more efficient scheduling and provisioning can be achieved if both problems are solved as one rather than independently. Among the existing approaches for combined provisioning and scheduling of workflow applications in public cloud environments, the IaaS Cloud Partial Critical Path (IC-PCP) algorithm [38] works with the closest assumptions to the system and application models. In the IC-PCP algorithm the estimation of the time to complete the data transfer time for each task in the VM becomes difficult if the number of tasks become high and still computational complexity is a main issue. The Enhanced IC-PCP with Replication (EIPR) algorithm also enables the provisioning of more VMs than the required for the execution of the workflow within the deadline to further increase the likelihood of deadline meeting. This is achieved with the enhancement of the model described in the previous section with the maximum number of replicas allowed for a single task and the replication budget rb(G), which defines the amount, in relation to the estimated cost of executing the workflow determined during the original provisioning and scheduling, that can be used to provision extra VMs to enable further tasks replication. Notice that, if rb(G)= 0, only opportunistic replication caused by idle time slots is applied, and therefore the algorithm operates like existing cost minimization approaches [39] with the advantage of opportunistic tasks replication.

To handle resource provisioning problem, the objective function (1) (2), the early start time (est) and latest finish time (lft) is computed via the use of the optimization algorithm. The major aim of the proposed Hybrid Bat and Differential Evolution Algorithm (HBDEA) with IHECC is not only solving resource provisioning problem in scientific workflow application within a user-defined deadline but also offering higher level of security for each cloud users with an optimal performance analysis, with the use of task replication. The proposed HBDEAIPR with IHECC algorithm applies a deadline constraint, higher level of security and variable budget for task replication to achieve its goals. The proposed HBDEAIPR is focused on producing the effective resources and allocates the tasks optimally by generating the fitness values. The resource management has been done optimally by means of superior QoS parameters. Figure 1 shows the overall working procedure of the proposed HBDEAIPR system. In a high level, the proposed algorithm performs three distinct steps:

Step 1: Combined provisioned of cloud resources
Step 2: Cloud resources management using the HBDEA and task scheduling
Step 3: Data transfer-aware provisioning adjusts
Step 4: Task replication
Step 4.1: Authentication for each nodes
Step 4.2: Grouping of tasks

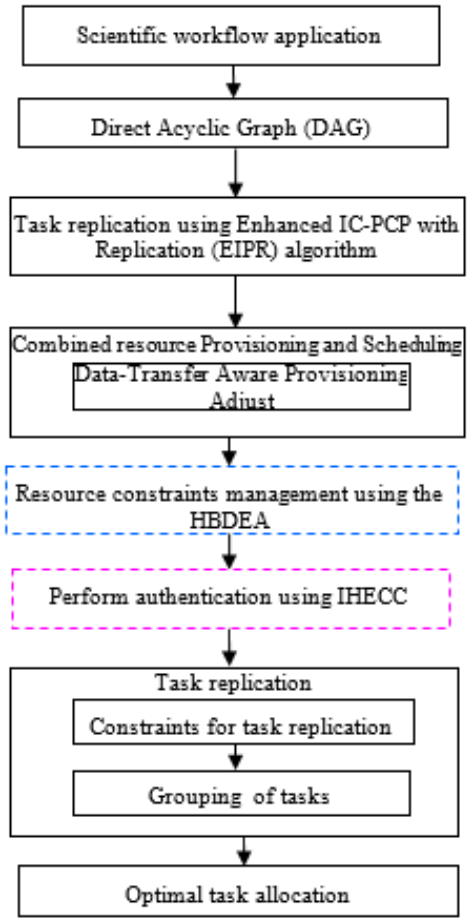

Fig. 1: Overall architecture of the proposed HBDEAIPR system.

Step 4.3: Filter genuine nodes
Step 5: Optimal task replication

4.1 Combined Provisioning and Scheduling

The first step of the EIPR algorithm consists in the determination of the number and type of VMs to be used for workflow execution as well as start and finish time of each VM (provisioning) and the determination of ordering and placement of tasks on such allocated resources (scheduling). The provisioning and scheduling problems are closely related, because the availability of VMs affects the scheduling, and the scheduling affects finish time of virtual VMs. Therefore, a more efficient scheduling and provisioning can be achieved if both problems are solved as one rather than independently. Among the existing approaches for combined provisioned and scheduling of workflow applications in public Cloud environments, the IC-PCP (IaaS Cloud Partial Critical Path) algorithm [40] works with the closest assumptions to the system and application models described in the

© 2019 NSP
Natural Sciences Publishing Cor.

recent work.

The IC-PCP algorithm disregards deployment and boot time of virtual machines, by assuming that the earliest start time of the entry task is 0. However, virtual machines provisioned from public Cloud providers are not immediately available for task execution; VMs need to be properly initialized and this time is not negligible. To better model effects of such non-negligible deployment and boot times of virtual machines in the workflow scheduling process, the EIPR algorithm assigns the average boot time of virtual machines, rather than 0, to $est(t_{entry})$, and $st(t_{entry})$ before calculating est and lft of each task.

4.2 Hybrid Bat and Differential Evolution Algorithm (HBDEA)

In the resource provisioning problem in PCP, the objective function (1) (2), the early start time (est) and latest finish time (lft) are calculated using the HBDEA algorithm. The Bat algorithm [41] exploits the so-called echolocation of the bats. Each bat is considered as the number of tasks in a workflow G. The bats use sonar echoes to search early start time (est) and latest finish time (lft) for the execution of a task in a workflow G in the Cloud on or before dl(G) and avoid obstacles. There happens a special dancing behavior of the bees [42]-[44] during the search for early start time (est) and latest finish time (lft) for the execution of a task in a workflow G in the Cloud on or before dl(G). It is generally known that sound pulses of the executed tasks are transformed into a frequency which reflects from obstacles. The bats navigate from one VM to another VM to search early start time (est) and latest finish time (lft) by using the time delay from emission to reflection. The pulse rate is usually defined as 10 to 20 times per second. The bats are using wavelengths that vary in the range from 0.7 to 17 mm. To implement the HBDEA based resource provisioning algorithm, the pulse frequency and the rate have to be defined based on the est and lft time of the task. The pulse rate of each task can be simply determined in the range from 0 to 1, where 0 means that there is no emission and 1 means that the bats emitting is their maximum [45]-[47]. The bat algorithm used three generalized rules for solving resource provisioning problem in CC are described as follows:

1.All the bats use an echolocation to sense the distance which finds the est and lft time of the task, they also guess the difference between the food/prey and background barriers in a somewhat magical way.

2.When searching for their prey, the tasks (bats) fly randomly with velocity v_i at position x_i with fixed frequency f_{min}, varying wavelength λ and loudness A_0. They can automatically adjust the wavelength (or frequency) of their emitted pulses and adjust the rate of pulse emission rε [0,1], depending on the proximity of their target.

3.Although the loudness can vary in many ways, we assume that it varies from a large (positive) A_0 to a minimum constant value A_{min}. Step 1: Objective function f(x), x=$(x_1,...,x_d)^T$
Step 2:Initialize the number of task in the bat population x_i and v_i for i = 1 : n
Step 3: Define pulse frequency f_i at x_i
Step 4:Initialize pulse rates r_i and the loudness A_i
Step 5: $t < T_{max}$ // number of iterations
Step 6:Produce new solutions by adjusting frequency and updating velocities and locations/solutions
Step 7: Generate a new solution by flying randomly
if $rand(0,1) < A_i$ and $f(x_i) < f(x))$
Accept the new solutions
Increase r_i and reduce A_i
endif
Rank the est, lft, and st of all tasks affected by the scheduling of the PCP and find the current best est, lft, and st results for all tasks
end
Step 8:Post-process results and visualization

The original BA is illustrated in Algorithm 1. In this algorithm, the bat behaviour is captured into the fitness function from equation (1-2) of the problem to be solved. It consists of the following components:

- –Initialization of number of tasks running in the workflow initialization (lines 2-4),
- –Generation of new est, lft, and st solutions for all tasks (lines 6-7),
- –Local search (lines 8-11),
- –Generation of a new est, lft, and st solution by flying randomly (lines 12-16)
- –Find the current best results of est, lft, and st for all tasks

Initialization of the task running in the workflow application as bat population is performed randomly. Generating new solutions is performed by moving virtual bats from one VM to another VM according to the following equations:

$$Q_i^t = Q_{min} + (Q_{max} - Q_{min})U(0,1) \tag{3}$$

$$v_i^{t+1} = v_i^t + (x_i^t - best)Q_i^{(t)} \tag{4}$$

$$x_i^{(t+1)} = x_i^{(t)} + v_i^{(t)} \tag{5}$$

where U(0, 1) is a uniform distribution. A random walk with direct exploitation is used for the local search that modifies the current best est, lft, and st solution according to equation:

$$x^{(t)} = best + \varepsilon A_i^{(t)}(2U(0,1) - 1) \tag{6}$$

where ε is the scaling factor, and $A_i^{(t)}$ the loudness. The local search is launched with the proximity depending on

© 2019 NSP
Natural Sciences Publishing Cor.

pulse rate r_i . The rate of pulse emission r_i increases and the loudness A_i decreases depending on the *est*, lft, and *st* solution. Mathematically, these characteristics are captured with the following equations:

$$A_i^{(t+1)} = \alpha A_i^{(t)}, r_i^{(t)} = r_i^{(0)}[1 - exp(-\gamma\varepsilon)] \quad (7)$$

where α and γ are constants. Differential Evaluation (DE) optimizes a resource provisioning local optima problem by maintaining a task of candidate results and creates new candidate results by combining the existing task time results according to its simple formulae, and then keeping whichever candidate solution has the fitness on the optimization problem. DE supports a differential mutation, a differential crossover and a differential selection. In particular, the differential mutation randomly selects two tasks resource provisioning results and adds a scaled difference between these to the third solution. This mutation can be expressed as follows:

$$u_i^{(t)} = w_{r_0} + F.(w_{r_1}^{(t)} - w_{r_2}^{(t)}, fori = 1, ..., NP \quad (8)$$

where F ε [0.1, 1.0] denotes the scaling factor as a positive real number that scales the rate of modification, while r_0, r_1, r_2 are randomly selected vectors in the interval 1 . . . NP. The trial vector is built out of parameter values copied from two different solutions. Mathematically, this crossover can be expressed as follows

$$Z_{i,j} = \begin{cases} u_{i,j}^{(t)} rand_j(0,1) \leq CRV\, j = j_{rand} \\ w_{i,j}^{(t)} \text{ otherwise} \end{cases} \quad (9)$$

where CR ε [0.0,1.0] controls the fraction of parameters that are copied to the trial solution. Note that, the relation $j = j_{rand}$ assures that the trial vector is different from the original solution Y(t). Mathematically, differential selection can be expressed as follows:

$$w_i^{(t+1)} = \begin{cases} Z_i^{(t)} \quad if f(Z^{(t)}) \leq f(Y_i^t) \\ w_{i,j}^{(t)} \text{ otherwise} \end{cases} \quad (10)$$

In the technical sense, the crossover and mutation can be performed in many ways in DE.

Step1: Objective function f(x), x=$(x_1, ..., x_d)^T$
Step2: Initialize the number of tasks in the bat population x_i and v_i for $i = 1 : n$
Step3: Define pulse frequency f_i at x_i
Step4: Initialize pulse rates r_i and the loudness A_i
While $t < T_{max}$ // number of iterations
Produce new est, lft, and st for all tasks by adjusting frequency and updating velocities and locations/solutions
if $rand(0,1) > r_i$
Alter the est, lft, and st for all tasks using differential evaluators operations in Eqs. (6) and (8)
endif Generate a new solution by flying randomly
if $(rand(0,1) < A_i \; and \; f(x_i) < f(x))$
Accept the new solutions
Increase r_i and reduce A_i
endif
Rank the est, lft, and st of all tasks affected by the scheduling of the PCP and find the current best est, lft, and st results for all tasks
end
Step5: Post-process results and visualization.

4.3 Data-Transfer Aware Provisioning Adjust

The combined provisioning and scheduling detailed in the previous section does not dictate the start and stop times of VMs. To determine both values, the algorithm has to consider not only start and end time of scheduled tasks, but also the data transfers to the first scheduled task and from the last scheduled task. If the first scheduled task in a VM is not an entry task, data from parent tasks have to be moved to the virtual machine before the task can run, and thus, the VM needs to be provisioned before the start time of its first task. This affects the start time of tasks and the total provisioning time of the VM, and may cause workflow execution delay and execution of VMs for an extra billing period. For each non-entry task scheduled as first task of a virtual machine, and for each non-exit task scheduled as the last task of a virtual machine, the algorithm meets the required communication time by setting the start time of the machine D(i,j) earlier than st of the first task, and/or setting the end time of the machine D(i,j) later than the finish time of the last task, depending on where the extra time is required. Finally, the beginning of the first allocation slot of each virtual machine is anticipated by the estimated deployment and boot time for virtual machines, which was accounted for during the scheduling process.

4.4 Authentication of task and resources using Improved Hyper Elliptic Curve Cryptography (IHECC)

In this research, IHECC method is applied for ensuring higher security. The hyper elliptic curves [48] achieve the better security level with a smaller key length as compared to cryptosystems using elliptic curves. Hyper elliptic curves are the basis for a relative new class of public-key schemes. It is thus of great interest to develop algorithms, which allow efficient implementations of elliptic and hyper elliptic curve cryptosystem. In this research, multiple clouds ensure the security to protect the task replication. Each node in the tasks is authenticated and ensuring that appropriate policies are enforced for data sharing. The HECC have q_g points on it, where q denotes the number of elements in the field of definition of the Jacobian. By choosing HECC this research can achieve the same order of magnitude of the group order with a smaller value of q when compared with elliptic

© 2019 NSP
Natural Sciences Publishing Cor.

curves. On the other hand, the group operation is much more cumbersome in HECC than in normal elliptic curves. A hyper elliptic curve C of genus g defined over a field F_q of characteristic p is given by an equation of form

$$Y^2 + h(x)y = f(x) \tag{11}$$

where f(x) is a monic polynomial of $Fq[x]$ of degree $2g+1$ and $h(x)$ is a polynomial over $Fq[x]$ of deg h≤g. A point on curve C is denoted by $P = (x,y)$, and its inverse is defined as $P = (x,y,h(x))$. Call a point P that satisfies P = P a ramification point. In contrast to ECC, points on a hyper elliptic curve do not form a group. Rather than points, divisors are deployed. A divisor D is a formal sum of points $\sum m_i P_i$, where $m_i \varepsilon Z$ .The degree of a divisor D is defined as $\sum m_i$.The jacobian variety $J_c(F_q)$ is defined by the quotient group D_0/P ,where D_0 is the divisor of the degree 0 and P is the principal divisor. The principal divisor is the divisor of a rational function on C which is a finite formal sum of the zeros and poles. A semi-reduced divisor can be expressed by two polynomials (u,v) of $F_q[x]$.

$$u(x) = \pi_i(x-x_i)^{m_i}, v(x_i) = y_i, degv < degu, v^2 + hv - f \equiv 0 modu \tag{12}$$

If $degu \leq g$ then the semi-reduced divisor is referred to as a reduced divisor. Elements in $J_c(F_q)$ are uniquely identified as reduced divisors. In order to improve the security of the task and resources management in the cloud computing model, new divisors are introduced to this work. Let $D1 = (u1, v1)$, $D2 = (u2, v2)$ be reduced divisors of the Jacobian $J_c(F_{2^n})$. Denote by D3 the addition of $D_1 + D_2$.

The data centers also have the responsibility of security of data. The IHECC is used for security which decides the access right for other people. In this technique, the service provider site is only responsible for the search operation, all other responsibilities are taken by the data owner. The responsibility of cloud provider is provides a security to client. And the tasks are authenticated before its replication then the optimal task is allocated.

4.5 Task Replication with Security

The aforementioned corrections enable virtual machines to be ready to receive data and tasks in the moment that they are required to meet times estimated during the scheduling process. However, it does not account for delays in the tasks execution caused by poor performance of public cloud resources. IPR tries to mitigate such effects with the utilization of task replication in idle slots of provisioned VMs or on new VMs allocated for enabling extra replication (if the replication budget allows). Notice that, because the goal of this replication is increasing the performance rather than fault tolerance, space replication is the target of IPR. Therefore, tasks are only replicated on different VMs, oppositely to a time replication approach where the same task could be scheduled multiple times in a single VM to increase fault-tolerance.

5 Performance Evaluation

In this section, describe the experiments conducted to evaluate the EIPR, Enhanced Artificial Bee Colony (ABC) based IaaS Cloud Partial Critical Path (IC-PCP) with Replication algorithm known as EAIPR , Enhanced Artificial Fish Swarm Algorithm (AFSA) based IaaS Cloud Partial Critical Path (IC-PCP) Replication (EAFSAIPR) and proposed HBDEAIPR algorithm . Experiments were conducted with the CloudSim toolkit [36]. The simulation testbed consists of a data center containing 500 hosts. Each host has 256 GB of RAM and 8 cores. The data center models Amazon AWS EC2 standard instance types, and the parameters relevant for the experiments are presented in Table 1. The billing period is 60 minutes. Table 1 and 2 shows the average

Table 1: VM Types Used in the Experiments

Type	Memory (GB)	Core Speed (ECU)	Cores	Cost ($)
m1.small	1.7	1	1	0.06
m1.medium	3.75	2	1	0.12
m1.large	7.5	2	2	0.24
m1.xlarge	15	2	4	0.48

cost (U$) of Workflows Execution. Standard Deviation for Each Case Is Presented in Parenthesis. Utilization of the resources under different algorithms with their Policy is also discussed in this table. The Budget on EIPR represents the Extra Budget Available for Replication, in Relation to the Amount Spent before the Replication. Four workflow applications are used in these tests. They are Montage (generation of sky mosaics), CyberShake (earthquake risk characterization), LIGO (detection of gravitational waves), and SIPHT (bioinformatics). These applications were characterized in [49]. Figure 2 presents the normalized execution time.

The average values observed for 50 executions are presented along with the standard deviation (in parenthesis). In most scenarios, HBDEAIPR significantly reduces the execution time of applications compared to EAIPR, EIPR and IC-PCP are illustrated in Figure 2. The proposed HBDEAIPR method provides optimal task replication with higher level of security. Utilization of opportunistic replication (i.e., utilization of the available gaps in the allocated VMs, without deployment of extra VMs) introduces, in most cases, performance improvements in the application execution times.

© 2019 NSP
Natural Sciences Publishing Cor.

Table 1: Average Cost (U$) Utilization of the EAFSAIPR and HBDEAIPR Policy without the Replication Stage

Workflow	Size	ICPCP	EAFSAIPR				HBDEAIPR			
			Budget =0	Budget =0.5X	Budget =1X	Budget =2X	Budget =0	Budget =0.5X	Budget =1X	Budget =2X
Montage	Medium	0.06 (0)	1.29 (0.25)	1.48 (0.35)	2.15 (0.43)	3.58 (0.66)	1.24 (0.29)	1.41 (0.38)	1.96 (0.46)	3.45 (0.69)
	Large	0.06 (0)	2.63 (0.45)	3.05 (0.48)	3.68 (0.72)	4.96 (0.48)	2.51 (0.51)	2.86 (0.52)	3.45 (0.78)	4.63 (0.54)
	Very large	20.09 (18.81)	18.45 (21.56)	52.58 (15.84)	54.48 (20.15)	95.72 (18.41)	16.58 (22.41)	48.36 (16.41)	52.41 (21.41)	86.51 (19.47)
Cybershake	Medium	0.47 (0.46)	0.29 (0.087)	0.42 (0.092)	0.56 (0.145)	0.89 (0.189)	0.27 (0.091)	0.39 (0.12)	0.52 (0.09)	0.82 (0.136)
	Large	1.13 (0.14)	0.54 (0.145)	0.96 (0.145)	1.45 (0.195)	2.03 (0.32)	0.52 (0.158)	0.87 (0.152)	1.38 (0.21)	1.95 (0.35)
	Very Large	46.22 (7.2)	50.25 (4.95)	80.58 (6.35)	102.36 (9.21)	156.41 (14.51)	48.51 (5.12)	75.72 (6.53)	98.28 (9.86)	142.57 (15.24)
LIGO	Medium	0.83 (0.13)	0.85 (0.55)	1.42 (1.12)	2.15 (1.24)	2.56 (2.12)	0.72 (0.61)	1.35 (1.25)	1.89 (1.32)	2.41 (2.27)
	Large	2.1 (0.24)	2.86 (0.88)	4.23 (0.81)	5.24 (0.98)	6.72 (1.24)	2.81 (0.838)	3.92 (0.682)	4.961 (1.18)	5.63 (1.38)
	Very Large	41.26 (17.69)	18.53 (22.58)	30.51 (29.87)	52.14 (27.41)	53.75 (32.87)	15.87 (23.71)	28.15 (30.51)	48.96 (34.18)	50.17 (33.45)
SIPHT	Medium	0.76 (0.06)	0.85 (0.108)	0.98 (0.145)	1.25 (0.184)	2.16 (0.29)	0.82 (0.124)	0.89 (0.154)	1.15 (0.191)	2.08 (0.31)
	Large	1.22 (0.09)	1.05 (0.25)	1.59 (0.28)	2.01 (0.345)	2.86 (0.401)	0.963 (0.28)	1.42 (0.31)	1.859 (0.358)	2.614 (0.425)
	Very Large	14.51 (0.96)	14.12 (2.32)	20.25 (2.41)	22.58 (2.68)	32.48 (3.41)	12.53 (2.45)	18.47 (2.48)	21.40 (2.72)	30.17 (3.57)

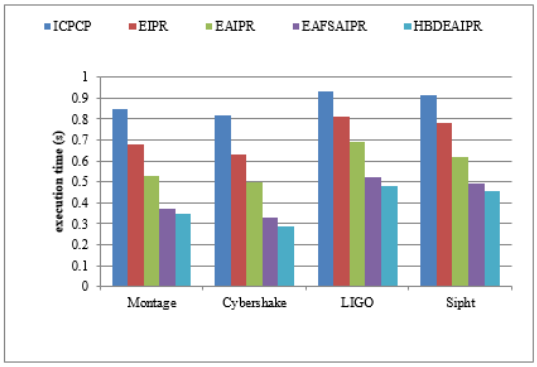

Fig. 2: Execution time vs. different application sizes.

6 Perspective

Previous research in workflow scheduling in the context of Clusters and Grids usually ignore costs related to utilization of the infrastructure, and also have limitations in the capacity of taking advantage of elastic infrastructures. Existing research in execution of scientific workflows in Clouds either tries to minimize the workflow execution time ignoring deadlines and budgets or focus on the minimization of cost while trying to meet the application's deadline. To address limitations of previous research, propose a Hybrid Bat and Differential Evolution Algorithm (HBDEA) based IaaS Cloud Partial Critical Path (IC-PCP) algorithm that uses idle time of provisioned resources to replicate workflow tasks to mitigate effects of performance variation of resources so that soft deadlines can be met. Hybrid Bat and Differential Evolution Algorithm (HBDEA) based IaaS Cloud Partial Critical Path (IC-PCP) Replication and Improved Hyper Elliptic Curve Cryptography (HBDEAIPR with IHECC). In this research, IHECC method is applied for ensuring higher security. Curves are the basis for a relative new class of public-key schemes. It is thus of great interest to develop algorithms which allow efficient implementations of elliptic and hyper elliptic curve cryptosystem. In this research, multiple clouds ensure the security to protect the task replication. To reduce the impact of performance variation of public Cloud resources in the deadlines of workflows, proposed a new algorithm, called EIPR, which takes into consideration the behavior of Cloud resources during the scheduling process and also applies replication of tasks to increase the chance of meeting application's deadlines. Future work will increase the capabilities of the EIPR algorithm by enabling replication of tasks across multiple Clouds. Another point that can be further explored is a

© 2019 NSP
Natural Sciences Publishing Cor.

new criteria for ranking candidate tasks for replication and also workflow structure-aware scheduling of replicas, where the structure of the workflow application is considered not only during the selection of candidates for replication but also during the replica′s scheduling. Also investigate how the replication-based approach can be used when the provisioning and scheduling process is performed for multiple workflows whose requests arrive at different rates.

References

[1] G. Juve, A. Chervenak, E. Deelman, S. Bharathi, G. Mehta, and K. Vahi,Characterizing and Profiling Scientific Workflows **29**, 682-692 (2013).

[2] Y.Duan, G. Fu, N. Zhou, X. Sun, N. Narendra, B. Hu, Everything as a Service (XaaS) on the Cloud: Origins, Current and Future Trends, 2015 IEEE 8th International Conference on Cloud Computing. IEEE, ISBN 978-1-4673-7287-9, pp. 621628, 2015 doi:10.1109/CLOUD.2015.88.

[3] G. von Laszewski, J. Diaz, F. Wang and G. C. Fox, Comparison of Multiple Cloud Frameworks, 2012 IEEE Fifth International Conference on Cloud Computing, Honolulu, HI, pp. 734-741, 2012, doi: 10.1109/CLOUD.2012.104

[4] S.He, L. Guo, Y. Guo, M. Ghanem, Improving Resource Utilisation in the Cloud Environment Using Multivariate Probabilistic Models, 2012 2012 IEEE 5th International Conference on Cloud Computing (CLOUD). pp. 574581, 2012

[5] M. Mao, M. Humphrey, A Performance Study on the VM Startup Time in the Cloud, Proceedings of 2012 IEEE 5th International Conference on Cloud Computing (Cloud2012), pp. 423, ISBN 978-1-4673-2892-0, 2012, doi:10.1109/CLOUD.2012.103.

[6] M. Abdel-Aty, Quantum information entropy and multi-qubit entanglement, Progress in Quantum Electronics, 31(1), pp. 1-49, 2007

[7] M. Abdel-Aty, An investigation of entanglement and quasiprobability distribution in a generalized JaynesCummings model, Journal of Mathematical Physics 44(4), pp. 1457-1471, 2003

[8] N. Metwally, M. Abdelaty, A.-S.F Obada, Entangled states and information induced by the atom-field interaction, Optics Communications 250(1-3), pp. 148-156, 2005

[9] M. Zidan, A.-H. Abdel-Aty, M. El-shafei, M. Feraig, Y. El-Abou, H. Eleuch and M. Abdel-Aty, Quantum Classification Algorithm Based on Competitive Learning Neural Network and Entanglement Measure, Appl. Sci., 9, 1277, 2019.

[10] A. Sagheer, M. Zidan and M. M. Abdelsamea, A Novel Autonomous Perceptron Model for Pattern Classification Applications, Entropy, 21(8), 763, 2019.

[11] M. Zidan, A. Sagheer and N. Metwally, An Autonomous Competitive Learning Algorithm using Quantum Hamming Neural Networks, In Proceedings of the 2015 International Joint Conference on Neural Networks (IJCNN), Killarney, Ireland, pp. 1–7, 2015.

[12] M. Zidan, A.-H. Abdel-Aty, D. M. Nguyen, A. S. A. Mohamed, Y. El-Abou, H. Eleuch H and M. Abdel-Aty, A quantum algorithm based on entanglement measure for classifying Boolean multivariate function into novel hidden classes, Results in Physics, 15, 102549 (2019), doi.org/10.1016/j.rinp.2019.102549

[13] M. Ogihara and A. Ray, Simulating Boolean circuits on a DNA computer, Algorithmica 25:239250, 1999.

[14] Y. Benenson, B. Gil, U. Ben-Dor, R. Adar, E. Shapiro, An autonomous molecular computer for logical control of gene expression, Nature, 429 (6990): 423429, 2004.

[15] Zhang, Q., Cheng, L., Boutaba, R., Cloud computing: state-of-the-art and research challenges, Journal of Internet Services and Applications, **1**, 7-18 (2010).

[16] Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M., A view of cloud computing, Communications of the ACM, textbf53, 50-58 (2010).

[17] Gulati, A., Shanmuganathan, G., Holler, A., Irfan, A.,Cloud scale resource management: challenges and techniques, In: Proc. 3rd USENIX Workshop on Hot Topics in Cloud Computing (2011).

[18] Abts, D., Felderman, A guided tour of data-center networking, Communications of the ACM **55**,44 (2012).

[19] Bari, M.F., Boutaba, R., Esteves, R., Zambenedetti Granville, L., Podlesny, M., Rabbani, M.G., Zhang, Q., Zhani, M.F., Data center network virtualization: A survey, IEEE Communications Surveys, **15**, 909-928 (2013).

[20] Jayadivya S, Bhanu SMS., Qos based scheduling of workflows in cloud computing,Int J Comput Sci Electr Eng 2315-4209 ISSN (2012).

[21] Kumar PaSA.,Priority Based Workflow Task Scheduling In Cloud Computing Environments, Aust J Basic Appl Sci, **8**,(2014).

[22] Rahman M, et al.,Adaptive workflow scheduling for dynamic grid and cloud computing environment,Concurr Comput: Pract Exp, **25**, (2013).

[23] Bala A, Chana I,A survey of various workflow scheduling algorithms in cloud environment, In: Proceedings of the 2nd national conference on information and communication technology (NCICT); (2011).

[24] Motahari-Nezhad HR, Stephenson B, Singhal S.,Outsourcing business to cloud computing services: opportunities and challenges, IEEE Internet Comput, **10**,(2009).

[25] Barrett E, Howley E, Duggan J., A learning architecture for scheduling workflow applications in the cloud, In: Proceedings of the 2011 ninth IEEE European conference on web services (ECOWS) (2011).

[26] K. Bessai, S. Youcef, A. Oulamara, C. Godart, and S. Nurcan, Workflow Tasks Allocation and Scheduling in Cloud Computing Environments, In IEEE Cloud, 638-645 (2012).

[27] E. Caron, F. Desprez, A. Muresan, and F. Suter, Budget constrained resource allocation for non-deterministic workflows on an iaas cloud, Proceedings of the 12th international conference on Algorithms and Architectures for Parallel Processing - Volume Part I, ser. ICA3PP-12.

[28] C. Jianfang, C. Junjie, and Z. Qingshan, An optimized scheduling algorithm on a cloud workflow using a discrete particle swarm, Cybernetics and Information Technologies, **14**, 25-39 (2014).

[29] Varalakshmi, P., Ramaswamy, A., Balasubramanian, A. and Vijaykumar, P., An optimal workflow based scheduling and resource allocation in cloud, In International Conference

© 2019 NSP
Natural Sciences Publishing Cor.

on Advances in Computing and Communications ,411-420, (2011)
[30] A. Nagavaram, G. Agrawal, M. Freitas, K. Telu, G. Mehta, R. Mayani, and E. Deelman, A Cloud-based Dynamic Workflow for Mass Spectrometry Data Analysis, in eScience, 47-54 (2011).
[31] M. Rahman, X. Li, and H. N. Palit, Hybrid Heuristic for Scheduling Data Analytics Workflow Applications in Hybrid Cloud Environment, Proceedings of the 25th IEEE International Symposium on Parallel and Distributed, ser. IPDPS Workshops. Anchorage (Alaska) USA: IEEE, 966-974 (2011).
[32] Pandey S, et al., A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments, In: Proceedings of the 2010 24th IEEE International Conference on advanced information networking and applications (AINA) (2010).
[33] Verma A, Kaushal S., Budget constrained priority based genetic algorithm for workflow scheduling in cloud, In: Proceedings of the Fifth International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom 2013) (2013).
[34] Liu, K., Jin, H., Chen, J., Liu, X., Yuan, D. and Yang, Y., A compromised-time-cost scheduling algorithm in SwinDeW-C for instance-intensive cost-constrained workflows on cloud computing platform, International Journal of High Performance Computing Applications (2010).
[35] Yassa S., Multi-objective approach for energy-aware workflow scheduling in cloud computing environments, Sci World J. (2013).
[36] E.-K. Byun, Y.-S. Kee, J.-S. Kim, and S. Maeng, Cost Optimized Provisioning of Elastic Resources for Application Workflows, Future Gener. Comput. Syst., **27**, 1011-1026 (2011).
[37] S. Abrishami, M. Naghibzadeh, and D. Epema, Deadline-Constrained Workflow Scheduling Algorithms for IaaS Clouds, Future Gener. Comput. Syst., **29**,158-169 (2013).
[38] E.-K. Byun, Y.-S. Kee, J.-S. Kim, and S. Maeng, Cost Optimized Provisioning of Elastic Resources for Application Workflow, Future Gener. Comput. Syst., **27**, 1011-1026 (2011).
[39] Calheiros, R.N. and Buyya, R., Meeting deadlines of scientific workflows in public clouds with tasks replication, IEEE Transactions on Parallel and Distributed Systems, **25**,1787-1796 (2014).
[40] Suguna N.andK.G.Thanushkodi,An Independent Rough Set Approach Hybrid with Artificial Bee Colony Algorithm for Dimensionality Reduction,American Journal of Applied Sciences, **8**, 261-266 (2011).
[41] Li Bao and Jian-chaoZeng, Comparison and Analysis of the Selection Mechanism in the Artificial Bee Colony Algorithm,In Proc. IEEE Ninth International Conference on Hybrid Intelligent Systems, 411-416 (2009).
[42] R.N. Calheiros, R. Ranjan, A. Beloglazov, C.A.F.D. Rose, and R. Buyya,CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms, Softw.,Pract. Exper., **41**, 23-50 (2011).
[43] X.S. Yang.,A new metaheuristic bat-inspired algorithm, Nature Inspired Cooperative Strategies for Optimization, (NICSO 2010), 65-74 (2010).
[44] A.H. Gandomi, X.S. Yang, A.H. Alavi, and S. Talatahari, Bat algorithm for constrained optimization tasks, Neural Computing & Applications,1-17 (2012).
[45] P.W. Tsai, J.S. Pan, B.Y. Liao, M.J. Tsai, and V. Istanda,Bat algorithm inspired algorithm for solving numerical optimization problems,Applied Mechanics and Materials, 134-137 (2012).
[46] X.S. Yang.,Review of meta-heuristics and generalised evolutionary walk algorithm, International Journal of Bio-Inspired Computation, **3**,77-84 (2011).
[47] Pelzl, J., Wollinger, T. and Paar, C., Special hyperelliptic curve cryptosystems of genus two: Efficient arithmetic and fast implementation,Embedded Cryptographic Hardware: Design and Security (2004).
[48] R.N. Calheiros, R. Ranjan, A. Beloglazov, C.A.F.D. Rose, and R. Buyya, CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms,Softw., Pract. Exper., **41**, 23-50.
[49] G. Juve, A. Chervenak, E. Deelman, S. Bharathi, G. Mehta, and K. Vahi,Characterizing and Profiling Scientific Workflows, Future Gener. Comput. Syst., **29**, 682-692 (2011).

A. Ramathilagam is obtained her Ph.D degree from Anna University, Chennai in 2018. She is a member of ISTE. She is a reviewer for various reputed journals. Her research interest includes Computer Networks, Distributed System, Grid and Cloud Computing. Currently, She is working as a Associate Professor in the Department of Computer Science and Engineering, at P.S.R.Engineering College. She has published 15 papers in journals and conferences.

S. Maheswari is working as an Associate Professor at National Engineering College,India. She received PhD from Anna University. Her current research interests include Semantic web service selection,Computer Networks and SoA.

© 2019 NSP
Natural Sciences Publishing Cor.