This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2469650, IEEE Transactions on Cloud Computing

Estimation of the Available Bandwidth in Inter-Cloud Links for Task Scheduling in Hybrid Clouds

Thiago A. L. Genez, Luiz F. Bittencourt, Nelson L. S. da Fonseca, and Edmundo R. M. Madeira
Institute of Computing, University of Campinas (UNICAMP)
Av. Albert Einstein, 1251, 13083-852, Campinas, São Paulo, Brazil
{thiagogenez, bit, nfonseca, edmundo}@ic.unicamp.br

Abstract—In hybrid clouds, inter-cloud links play a key role in the execution of jobs with data dependencies. Insufficient available bandwidth in inter-cloud links can increase the makespan and the monetary cost to execute the application on public clouds. Imprecise information about the available bandwidth can lead to inefficient scheduling decisions. This paper attempts to evaluate the impact of imprecise information about the available bandwidth in inter-cloud links on workflow schedules, and it proposes a mechanism to cope with imprecise information about the available bandwidth and its impact on the makespan and cost estimates. The proposed mechanism applies a deflating factor on the available bandwidth value furnished as input to the scheduler. Simulation results showed that the mechanism is able to increase the number of solutions with makespans that are shorter than the defined deadline and reduce the underestimations of the makespan and cost provided by workflow schedulers.

Index Terms—Cloud computing, hybrid clouds, workflow, scheduling

1 Introduction

In hybrid clouds, when computational demands exceed the capacity of the private pool of computational resources, users can lease resources from public cloud providers. Such capacity of leasing on demand resources from public providers as needed is called “elasticity” [1].

A workflow scheduler is an agent that is responsible for determining on which resource the jobs of a workflow should run, distributing workflow jobs to different processing elements so that they can run in parallel, speeding up the execution of the application (the term job refers to any computational work to be executed, therefore comprising services or standard tasks) [2]. The scheduler is a central element in a cloud architecture that promotes elasticity.

When processing workflows with data dependencies, jobs can be placed on different clouds which require data transfer on Internet links. The available bandwidth on the Internet links fluctuates because it is a shared resource; it can decrease, which increases data transfer times and consequently the execution time (makespan) of workflows. Moreover, the increase of the makespan can cause the need of longer rental periods, which increases costs. Such variability of data transfer periods needs to be considered by a scheduler when producing a schedule. Moreover, the problem of workflow scheduling in hybrid clouds encompasses several variables, such as a bound to the execution time (deadline). A previous study [3] suggested that data transfer on the Internet links when processing workflows with data dependencies has a significant impact on the execution times and costs when using public clouds.

Most studies on cloud networking focus on datacenter networks [4], [5], including those aimed at guaranteeing bandwidth [6]. These studies do not address the problem of having imprecise information about the available bandwidth at scheduling time. In a previous work [3], we proposed a procedure for adjusting the estimates of the available bandwidth that are used as inputs to schedulers that are not designed to address with inaccurate information. The contribution of the present paper is twofold. The first is to provide an answer to the fundamental question about the impact of uncertainty in relation to the value of the available inter-cloud bandwidth on the schedules produced by a hybrid cloud scheduler not originally designed to address such uncertainty. The second is the introduction of a mechanism to reduce the impact of such uncertainties on the schedule generated by scheduling algorithms proposed in the literature.

In an attempt to answer this question, the performance of three different scheduling algorithms, that were not designed to use inaccurate information, are evaluated to show their inability to address with the variability of the available bandwidth in inter-cloud links. A procedure is then proposed for adjusting the estimate of the available bandwidth based on the expected uncertainty so that underestimations of the makespan and costs can be reduced.

The proposed procedure deflates the estimated available bandwidth that is used as an input to a hybrid cloud scheduler. The deflating factor is computed using

2168-7161 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2469650, IEEE Transactions on Cloud Computing

multiple linear regression, and the execution history of a specific workflow is employed in the computation. The effectiveness of the procedure in handling imprecise is compared the effectiveness the HCOC (Hybrid Cloud Optimized Cost) [7] and PSO (Particle Swarm Optimization) [8] schedulers. The procedure was evaluated using several scenarios involving different clouds and their occupancy as well as a wide range of deadline constraints. Three types of scientific workflows available from the workflow generator gallery[1] were employed in the evaluation. This gallery contains workflows that use structures and parameters taken from real applications.

Simulation results showed that the use of the proposed procedures can reduce the cost estimation error as well as increase the number of qualified solutions. The proposed procedures were able to reduce the number of disqualified solutions by up to 70% with costs reduced by up to 10%.

This paper is organized as follows. Section 2 reviews the workflow scheduling problem and some cloud schedulers that were not designed to handle inaccurate information about the available bandwidth. The proposed mechanism is introduced in Section 3 and evaluated in Section 4, and the results are discussed in Section 5. Section 6 presents related work, and Section 7 provides final remarks and conclusions.

2 Workflow Scheduling Problem

In this section, we describe workflow scheduling in hybrid clouds and present two scheduling algorithms; one is cost-aware, and the other is cost- and deadline-aware.

2.1 Workflow Representation

Workflow applications vary in size and computational demands, and both can depend on the input data. This process makes this type of workflow suitable for processing in a hybrid cloud because the computational demands can exceed the available capacity of private clouds. Data dependencies in non-cyclic workflows are commonly modeled as a directed acyclic graph (DAG), in which nodes represent jobs, and arcs represent job dependencies. A topological ordering of a DAG is an easy way of visualizing the workflow. A job can only start after all of its data dependencies have been solved; i.e., after all of its parent jobs have finished and have sent the necessary data to the job. Labels on the nodes and arcs of a workflow DAG designate the computational and communication demands, respectively. The computational demands depict how much computing power a job requires to be executed, and the communication demands depict how much data must to be transferred between two jobs.

An arc e_{v_a,v_b} of $\mathcal{G}$ represents a data file produced by v_a that will be consumed by v_b. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a DAG with n nodes (or jobs), in which each $v \in \mathcal{V}$ is a workflow job with a non-negative computational demand $w_v \in \mathcal{P}$, and $e_{v_a,v_b} \in \mathcal{E}$, $v_a \neq v_b$, is an arc that represents the data dependency between jobs j_a and j_b with a non-negative communication demand $c_{a,b} \in \mathcal{C}$. The 4-tuple $\mathcal{W} = \{\mathcal{V}, \mathcal{E}, \mathcal{P}, \mathcal{C}\}$ represents the workflow and job demands.

Several scientific workflows generate petabytes or even exabytes of data, which are usually distributed for processing at different sites. Communication between jobs occurs via file transfer, usually on Internet links. Examples of scientific workflows include Montage[2] [9], LIGO[3] [10], AIRSN [11], and Chimera [12]. Moreover, a scheduler must produce schedules that support the QoS requirements of an application while considering the communication demands due to data dependencies. Therefore, the bandwidth availability in communication channels that connect private and public clouds plays an important role in achieving a schedule objective. Neglecting the variabilities of the available bandwidth in the production of a schedule can hamper the execution of the workflow [13] and compromise the provisioning of quality of service (QoS) [14]. *Uncertainties* in the value of the input data to the scheduler requires elaborate scheduling schemes.

2.2 Scheduling Workflows in Hybrid Clouds

A hybrid cloud (Figure 1) is composed of the resources of a private cloud, which are free of charge, as well as those of one or more public cloud providers, with the public cloud provider furnishing on-demand or reserved resources via long-term contracts. When public resources are leased, clients are charged on a pay-per-use basis, usually for hours of use with reserved resource typically leased at lower prices than on-demand ones. The hybrid cloud scheduler must decide which resources should be leased from public cloud providers in the composition of the hybrid cloud so that the runtime of the application does not exceed the deadline and satisfies budget constraints. In this paper, we assume that the organization has a broker in its premises which is responsible for connecting resources from the public clouds to those of the private cloud resources. This broker receives requests for workflow execution and rents resources from the public cloud based on the decisions of schedulers.

The schedulers for hybrid clouds considered in this paper are not aware of data center network topology. This paper considers only application schedulers which responsibility is to allocate tasks onto virtual machines already provisioned by the infrastructure provider. The allocation of virtual machines onto physical machines is done by the infrastructure provider, which uses mechanisms aware of the data center network topology. Examples of network-aware VM placement scheduler can be found in [15], [16]. The application scheduler in this

1. https://confluence.pegasus.isi.edu/display/pegasus

2. http://montage.ipac.caltech.edu/
3. http://www.ligo.caltech.edu/

2168-7161 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2469650, IEEE Transactions on Cloud Computing

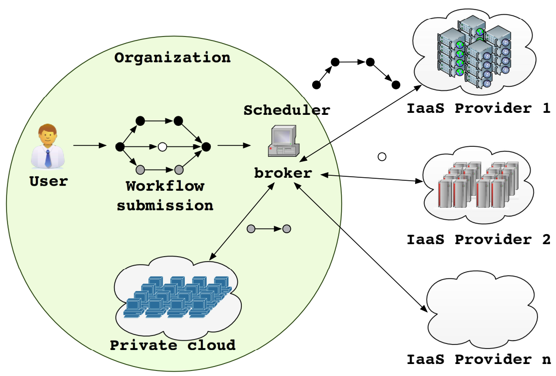


Fig. 1. Hybrid cloud infrastructure and workflow submission.

paper does not receive as input information about the topology of the data center network, but considers this information indirectly since it uses historical data of the performance of the execution of workflows as well estimations of the available bandwidth

Except for a few simple cases [2], the scheduling problem is NP-Complete, which calls for heuristics to produce a near-optimum schedule. Scheduling is a decision-making process that involves characteristics from the application being scheduled, such as memory and disk space consumption, and from the target system, such as processing power, storage and bandwidth availability. Intra- and inter-cloud communication is important in hybrid clouds because large bandwidth availability decreases communication delays, which contributes to the minimization of the makespan. Information about the application and on cloud resources is commonly assumed to be precise; however, in a real environment, it is difficult to obtain precise estimates due to the limitations of the measurement tools [17]. Information about the cost of resource usage is also required by the schedulers. By considering all this information, a scheduler for hybrid clouds should be able to assign the jobs of a workflow to the resources of a cloud.

2.3 Cost-aware Scheduler

One class of schedulers that can be employed in hybrid clouds focuses only on cost minimization. One such algorithm employs Particle Swarm Optimization (PSO) [8]. A particle in the swarm is taken as a potential schedule and represents a set of workflow jobs to be scheduled and evaluated by a fitness function that computes the cost of execution. A heuristic is employed to iteratively utilize the results from the particle swarm optimization. This heuristic first computes the average computational and communication demands, which are expressed by labels of nodes and arcs, respectively. It then computes an initial schedule for each "ready job"; i.e., jobs that have received the files that are necessary for its execution. PSO is then run over the next set of ready jobs, and these steps are repeated until all the jobs have been scheduled. Algorithm 1 shows a scheduling heuristic that is based on PSO.

Algorithm 1 Scheduling Heuristic Algorithm Overview

```
1: Compute the average computing cost of all jobs in all resources
2: Compute the average cost of communication between resources
3: R is a set of all jobs in the DAG G
4: Call PSO(R)
5: repeat
6:    for all ready jobs j_i ∈ R do
7:       Assign job j_i to resource p_j according to the solution provided by PSO
8:    end for
9:    Dispatch the mapped jobs
10:   R is a set of ready jobs in the DAG G
11:   Call PSO(R)
12: until all jobs are scheduled
```

Even though the heuristics that are based on PSO do not consider deadlines, they do not completely neglect the makespan. The cost function that is optimized by the PSO ensures that all of the jobs are not mapped onto the same resource, which promotes parallelism and consequently reduces the makespan by avoiding sequential execution of the workflow on the least expensive virtual machine available. The PSO algorithm has a time complexity of $\mathcal{O}(n^2 \times r)$ where n is the number of workflow tasks, corresponding to the number of dimensions of a particle swarm, and r the number of resources in the hybrid cloud. The number of particles in the swarm and the number of iterations (stopping criteria) are constant values.

To make this scheduler deadline-aware, the PSO function was modified to calculate the monetary costs of only the particles that satisfy the deadline. If a particle does not satisfy the deadline, the solution is not "qualified", and the cost is given a value of infinity. The objective function consists of minimizing the monetary cost by considering only particles on the swarm that represent qualified solutions. In this scheduling algorithm, which is called adapted-PSO, the equation

$$Cost(M) = \max_{\forall j \in P}(C_{total}(M_j)) \tag{1}$$

is replaced by

$$Cost(M) = \begin{cases} \sum\limits_{\forall j \in P} C_{total}(M_j) \text{ if } & makespan(M) \leq \mathcal{D}_M \\ +\infty & \text{otherwise} \end{cases} \tag{2}$$

where M is a schedule that is represented by a particle of the swarm, $C_{total}(M_j)$ is the function that calculates the monetary cost of using the resource j, and $\mathcal{D}_M$ is the deadline required by the user. The cost $C_{total}(M_j)$ of a resource j is the sum of the costs of execution and data transfer. The time complexity of the adapted-PSO algorithm is equal to that of the PSO algorithm.

2.4 Deadline- and Cost-Aware Scheduler

The Hybrid Cloud Optimized Cost (HCOC) [7] is an algorithm for workflow scheduling in hybrid clouds that

2168-7161 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2469650, IEEE Transactions on Cloud Computing

attempts to minimize the execution cost while meeting specified deadlines. HCOC starts by scheduling the workflow in the private cloud because those resources are free of charge. If the predicted makespan exceeds the specified deadline, jobs are selected to be rescheduled in the public cloud.

HCOC accepts different heuristics to perform the initial schedule in the private cloud. In this paper, the first schedule is obtained by employing the path clustering heuristic (PCH) [18], which clusters jobs in the same path of the DAG. In HCOC, the selection of the job that will be rescheduled and the selection of the resource from the public cloud are directly influenced by the available bandwidth in the inter-cloud links. An incorrect estimate of the available bandwidth can lead to deadline violations and increased costs. Algorithm 2 details these steps. The heuristic PCH in the fist step can be replaced by any other heuristic, such as HEFT [19]. Both PCH and HCOC algorithms have $\mathcal{O}(n^2 \times r)$ time complexity when considering n tasks and r resources.

Algorithm 2 HCOC Algorithm Overview

$\mathcal{R}$ is the set of all resources in the private cloud
Schedule $\mathcal{G}$ in the private cloud using PCH
while $makespan(\mathcal{G}) > \mathcal{D}$ AND $iteration < size(\mathcal{G})$ **do**
    $iteration = iteration + 1$
    select node n_i such that its priority is max AND $n_i \ni \mathcal{T}$
    $\mathcal{H} = \mathcal{R}$; $\mathcal{T} = \mathcal{T} \cup t$
    $num\_clusters =$ the number of clusters of nodes in $\mathcal{T}$
    while $num\_clusters > 0$ **do**
        Select resource r_i from the public clouds such that $((price_i) \div (num\_cores_i \times p_i))$ is minimum AND $num\_cores_i <= num\_clusters$
        $\mathcal{H} = \mathcal{H} \cup \{r_i\}$
        $num\_clusters = num\_clusters - num\_cores_i$
    end while
    for all $n_i \in \mathcal{T}$ **do**
        Schedule n_i in $h_j \in \mathcal{H}$ such that EFT_i is minimum
        Recalculate the jobs attributes for each job
    end for
end while

3 Handling Uncertainty in Available Inter-cloud Bandwidth Values

One of the problems faced in scheduling is the quality of the information that is provided to the scheduler [20]. Imprecision in the input data may lead to an underestimation of the makespan and as service is offered on a pay-per-use basis, the customer will then potentially pay more to run the application. Communication between computational resources in the private cloud and in the public cloud is a major issue in hybrid clouds. Algorithms that are robust for low quality information are of paramount importance to minimize the negative consequences of uncertainty in the available bandwidth information and to avoid underestimating the makespan and budget.

3.1 Procedure for Deflating Estimates of the Available Bandwidth in Inter-cloud Links

The quality of information model used in this paper is based on the model presented in [20]. The estimate of the available bandwidth that is utilized as an input to the scheduler at the scheduling time may deviate from that experienced at runtime by a percent error p. For example, in a scenario in which the expected uncertainty of the available bandwidth in the inter-cloud links is $p = 50\%$, a data transfer that is initially estimated by the scheduler to take 30 seconds could take between 15 and 45 seconds during the execution of the workflow.

Overestimating the available bandwidth yields longer data transfer times than expected, which increases the makespan and can potentially exceed the established deadline. This can lead to increased costs due to the need for longer rental periods, which can surpass the defined budget. On the other hand, underestimating the available bandwidth can also lead to unnecessary leasing of processing power; however, this does not call for a budget increase. Indeed, the precision of the input data that are provided to the scheduler is essential to the effectiveness of the schedule that is generated by the scheduling algorithm.

Traditional prediction methods, such as time series, can be used to predict the available bandwidth. However, although such predictions may be a good representation of the variability of the available bandwidth for a specific period, they would not represent the impact of the bandwidth estimates on the predicted makespan and costs. Thus, a procedure that correlates the uncertainty in the available inter-cloud bandwidth with the produced schedule is needed and has not been proposed.

Thus, a proactive procedure is proposed here to minimize the negative impact of imprecise information about the available bandwidth on the workflow execution in hybrid clouds. The proposed procedure produces a deflating multiplier value that is applied to the estimated available bandwidth. In other words, the proposed mechanism applies a reduction factor $\mathcal{U}$ to the measured inter-cloud bandwidth at the scheduling time to prevent misleading information from affecting the performance of the workflow execution. Information about the degree of uncertainty is provided to the scheduling algorithm, and the schedule is generated based on the expected uncertainty in the data transfer time. The proposed procedure calculates a $\mathcal{U}$ value for a workflow using a required deadline value, an estimate of the available bandwidth and the bandwidth variability, which indicates the uncertainty of the available bandwidth. The computation of $\mathcal{U}$ value employs information from previous executions of the target workflow. Thus, for each type of workflow, the proposed method uses previously calculated $\mathcal{U}$ values, the available bandwidth and the error that occurred in previous estimates of the makespan and cost. The uncertainty value can be derived from benchmarks or historical data of bandwidth availability in the inter-cloud links obtained from monitors [21], [22], [23].

Figure 2 illustrates the proposed approach. Information about the deadline violation and cost of a workflow as well as on the bandwidth variabilities during the execution of a workflow is stored in a repository and

2168-7161 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2469650, IEEE Transactions on Cloud Computing

contributes to the history of the execution of a specific workflow. This data is retrieved to compute the $\mathcal{U}$ value.

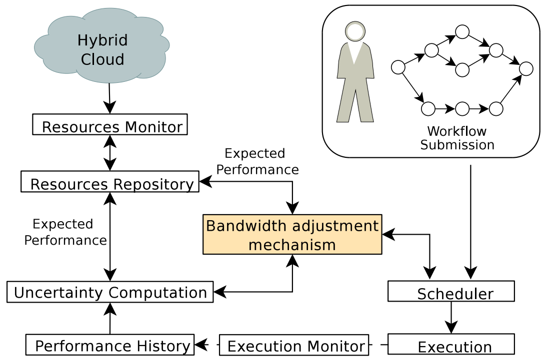


Fig. 2. Scheduler bandwidth adjustment mechanism.

Algorithm 3 $\mathcal{H}_k$ computation for a DAG G_i

$\mathcal{H}$ = history of past executions of G_i
$\mathcal{H}_k = \emptyset$
for all (bw_i, p_i) pair **do**
  $minimumError^m = +\infty$; $minimumError^{\$} = +\infty$
  for all 5-tuples $\left\langle bw_j, p_j, \mathcal{U}_j, error^m_{G_i}, error^{\$}_{G_i} \right\rangle \in \mathcal{H}$ **do**
    if $bw_j = bw_i$ **and** $p_j = p_i$ **then**
      if $|error^m_{G_i}| < minimumError^m$ **then**
        $minimumError^m = |error^m_{G_i}|$
        $best\_tuple^m = \langle bw_j, p_j, \mathcal{U}_j \rangle$
      end if
      if $|error^{\$}_{G_i}| < minimumError^{\$}$ **then**
        $minimumError^{\$} = |error^{\$}_{G_i}|$
        $best\_tuple^{\$} = \langle bw_j, p_j, \mathcal{U}_j \rangle$
      end if
    end if
  end for
  $\mathcal{H}_k = \mathcal{H}_k \cup \{best\_tuple^m, best\_tuple^{\$}\}$
end for

3.2 Computation of the $\mathcal{U}$ value

The proposed procedure works between the available bandwidth estimation tool and the scheduler. It takes the available bandwidth value provided by the estimation tool and applies the deflating $\mathcal{U}$ value to it before providing the deflated bandwidth value as an input to the scheduler.

Historical information about previously computed $\mathcal{U}$ values and estimates of the available bandwidth are stored in the database as well as the differences between previously predicted and measured makespan values and costs. A multiple linear regression (MLR) procedure is employed to compute the $\mathcal{U}$ value; these data are used to compute the values of the coefficients a, b and c in the equation $f(x, y) = ax + by + c$, where x is a variable that represents the current predicted uncertainty in the available bandwidth, y is an independent variable that represents the currently available bandwidth, and $f(x, y) = \mathcal{U}$.

The data set $\mathcal{H}_G$ is composed of the 5-tuple $h_i = \left\langle bw, p, \mathcal{U}, error^m_G, error^{\$}_G \right\rangle$ that is used for the future production of schedules, where bw is the estimated available bandwidth in the inter-cloud channel for the scheduling of a workflow G, which is represented by a directed acyclic graph (DAG), $error^m_G$ is the difference between the estimated makespan at the scheduling time and the real observed runtime value of G, $error^{\$}_G$ is the difference between the estimated monetary execution cost at the scheduling time and the real monetary cost of the execution of G, p is the maximum error between bw and the observed available bandwidth in the inter-cloud channel during the execution of the workflow on the hybrid cloud, and $\mathcal{U}$ is the reduction factor used at the scheduling time.

The subset $\mathcal{H}_k \subseteq \mathcal{H}_G$ is used by the multiple linear regression procedure. For each (bw, p) pair in $\mathcal{H}_G$, the subset $\mathcal{H}_k$ is constructed by selecting two 3-tuples: $(bw, p, \mathcal{U}^m)$, which gives the minimum absolute value of $error^m_G$, and $(bw, p, \mathcal{U}^{\$})$, which gives the minimum absolute value of $error^{\$}_G$. The construction of the $\mathcal{H}_k$ subset is illustrated in Algorithm 3. This algorithm has a time complexity of $\mathcal{O}(n^2)$ with n being the number of past executions of $\mathcal{G}$ (size of the history of past execution for a DAG $\mathcal{G}$), and each execution n use different values of $bw_j, p_j, \mathcal{U}_j, error^m_{G_i}$ and $error_{G_i}$.

When an application workflow G is about to be scheduled, the value of the reduction factor $\mathcal{U}$ is computed using the linear equation $f(x, y) = ax + by + c$, where the coefficients a, b, and c are given by a multiple linear regression that employs $\mathcal{H}_k$ as an input. The dependent variable $\mathcal{U}$ is computed from the explanatory (independent) variables p and bw, which are obtained from the repository. Making $x = p$ the current predicted uncertainty and $y = bw$ the current available bandwidth, a percentage reduction factor $\mathcal{U} = f(x, y)$ is computed for the next schedule computation. The proposed procedure requires a set of historical data for training before the MLR approach can be applied.

4 Evaluation Procedure

This section describes the evaluation of the efficacy of the proposed procedure. The performance of the scheduling algorithms used in this paper are evaluated in the publications in which they were originally presented [7], [8]. Comparing results derived using the proposed scheme to optimal values is infeasible, due to the NP-Completeness of the scheduling problem. The objective of this section is to assess the proposed mechanism when attached to existing schedulers.

4.1 Workflows

The proposed procedure is evaluated using simulations with synthetic workflows from the workflow generator gallery[1]. The gallery contains synthetic workflows that were modeled using patterns taken from real applications, such as Montage, LIGO and CyberShake. Workflows with different sizes such as 50, 100, 200 and 300 tasks (DAG nodes) are employed. For each workflow application and for each workflow size, 100 different

2168-7161 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2469650, IEEE Transactions on Cloud Computing

TABLE 1
Hybrid cloud environment

Provider	Type	Core	Core Performance	Price per time unit	Data transfer out price
A_1 (private)	S	1	1.0	\$0.00	\$0.00
A_2 (private)	S	1	1.0	\$0.00	\$0.00
	M	1	1.5	\$0.00	\$0.00
A_3 (private)	S	1	1.0	\$0.00	\$0.00
	M	1	1.5	\$0.00	\$0.00
	G	2	2.0	\$0.00	\$0.00
B (public)	S	1	2.75	\$0.104	\$0.01
	M	2	2.75	\$0.207	\$0.01
	L	4	2.75	\$0.415	\$0.01
	XL	8	2.75	\$0.829	\$0.01
C (public)	S	1	1.0	\$0.06	\$0.12
	M	1	2.0	\$0.12	\$0.12
	L	2	2.0	\$0.24	\$0.12
	XL	4	2.0	\$0.48	\$0.12
	XXL	8	3.25	\$0.9	\$0.12

workflows were generated differing in the computational demands (weights of the DAG nodes) uniformly distributed in the interval $[1, 10]$, and the communication demands (weights of the DAG edges) uniformly distributed in the interval $[50, 60]$. The set of synthetic workflows contains 3 types of workflow applications, 4 different workflow sizes, and 100 workflows per application and size, resulting in a total of $1,200$ synthetic workflows. The simulator reads the workflow description files in a DAX format that is used by the Pegasus Workflow Management System[4].

4.2 Hybrid Cloud Scenario

Hybrid cloud scenarios are shown in Table 1. They employ resources and prices based on the configurations of the Amazon EC2[5] and Google Compute Engine[6] cloud providers. To simulate the availability of private resources, we use three different sizes of the private cloud to simulate situations in which private resources are not sufficient to handle the computational demands of the workflow execution and the leasing of public resources becomes necessary to meet with the application deadline.

4.3 Simulator

To evaluate the use of the proposed procedure, a cloud simulator was developed to calculate the estimated values of the makespan and monetary cost considering an uncertainty value $p\%$. Figure 3a shows an overview of how the simulations were performed. The HCOC and PSO schedulers were implemented in Java according to their descriptions in [7] and [8], respectively, and the adapted-PSO according to the adjustments described in Section 2.3. The data inputs for the scheduler are a DAX file, a VM file and an estimated value of the available inter-cloud bandwidth bw. The DAX file contains the description of the workflow in XML format, while the VMs file contains information about the hybrid cloud (Table 1). For the HCOC and adapted-PSO schedulers, a deadline value for the workflow execution is also provided as input data. The estimated inter-cloud available bandwidth value bw is deflated by the factor $\mathcal{U}$, and the result is used by the scheduler; $\mathcal{U} = 0$ means that no reduction was applied, and $\mathcal{U} = 5$ means that the deflated bandwidth is 95% of the value provided at the scheduling time.

After the schedule is produced by the scheduler, it is used as an input to the simulator, which estimates the values of the makespan and cost of the execution of the workflow. The simulator uses the uncertainty value p of the inter-cloud bandwidth. Every time the simulator needs to use the value of the inter-cloud available bandwidth, it takes a new value that is uniformly distributed in the interval $[b - p\% ; b + p\%]$. This uncertainty p is only applied to the inter-cloud available bandwidth.

4.4 Experimental Parameters

A broad space of parameter values, including deadlines, was used cases ranging from tight deadlines, which imply that a small number of workflows can be completed with private resources, to more loose deadlines, which imply that almost all of the workflows can be completed using only private resources. In addiction, three levels of cloud utilization were employed, ranging from an idle private cloud, in which most of the executions could be performed without violating the deadline, to a busy private cloud, which calls for the use of a public cloud to accomplish most of the executions before the deadline.

The deadline values, $\mathcal{D}$, varied from $T_{max} \times 2/7$ to $T_{max} \times 6/7$ in $1/7$ steps, where T_{max} is the makespan of the least expensive sequential execution of all the DAG nodes on a single resource. Deadlines of $T_{max} \times 1/7$ resulted in only disqualified solutions (makespans greater than the deadline value) for all executions. Schedules for deadlines of $T_{max} \times 7/7$ can be trivially achieved by scheduling all of the tasks sequentially in the least expensive resource and were not generated in this paper.

Inter-cloud bandwidths of 10 to 60 Mbps and intra-cloud bandwidths of 1 Gbps were considered. We assume that the intra-cloud bandwidth is greater than the inter-cloud bandwidth, which is a reasonable assumption in real hybrid cloud environments. The available bandwidth values in the inter-cloud links were based on measurements of TCP connections for the Amazon EC2 cloud [21] and the Rackspace.com cloud [22].

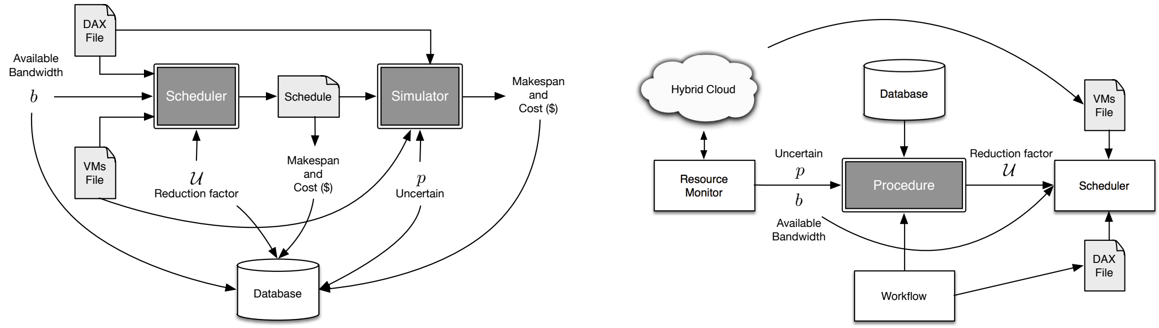
4.5 Experimental Setup

The procedure was evaluated in three steps. In the first step, a database that contains historical information about the workflow execution is produced for each type of workflow, deadline and scheduler (Figure 3a) using a fixed deflating bandwidth $\mathcal{U}$. In the second step, the

4. https://pegasus.isi.edu/
5. http://aws.amazon.com/ec2/
6. https://cloud.google.com/compute/

2168-7161 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2469650, IEEE Transactions on Cloud Computing



(a) First step

(b) Second and third steps

Fig. 3. Overview of the steps of the simulations

coefficient values of the MLR equation are generated using information from the database. In the last step, the MLR procedure is applied to deflate the bandwidth value, which is used as an input to the scheduler (Figure 3b) when a new workflow is set to be scheduled.

Initially, we assessed how using the procedure with a fixed $\mathcal{U}$ factor ameliorates the negative impact of imprecise information about the inter-cloud available bandwidth on the execution of the workflow. To investigate the results of the produced schedules, fixed bandwidth deflating factors $\mathcal{U} \in \{0, 5, 15, 25, 35, 45, 50\}$ were used. The choices of the fixed $\mathcal{U}$ values was based on experiments performed for this purpose. Results indicated that when the measured available bandwidth values were deflated by more than 50%, the scheduler did not find qualified solutions within the established deadline in most of the experiments conducted. For this reason, $\mathcal{U}$ values less or equal than 50% were used in this paper. For each value of $\mathcal{U}$, the percent error p ranged from 10% to 99%. For example, given a $\mathcal{U}$ value, a percentage error p was introduced into the simulation to derive the makespan and costs. The scheduling and simulation results were stored in the database. Therefore, for each 3-tuple $\langle bw, p, \mathcal{U}\rangle$, the cost and makespan estimates given before (by the original values from scheduler) and after the simulations were stored.

In the second step, based on the type of workflow and the scheduler, the values of the coefficients of the multiple linear regression (MLR) equation are computed using Algorithm 3. The MLR derives the values of the coefficients a, b and c in the equation $f(x, y) = ax+by+c$. By checking the resource monitor and making $x = p$ the currently predicted uncertainty and $y = bw$ the currently available bandwidth, a deflating factor $\mathcal{U} = f(x, y)$ can be computed and used as an input to the scheduler.

In the last step, new simulations were run for each type of workflow to evaluate the behaviour of the proposed mechanism. For all $\mathcal{U} = f(x, y)$ and p values, average values were computed considering only the schedules that results in makespan values lower than the deadline.

5 RESULTS

This section presents numerical results that were produced to assess the impact of imprecise information on the available bandwidth in inter-cloud links as well as the effectiveness of using a constant value and a value derived by the MLR procedure for the deflating factor that is applied to the available bandwidth.

5.1 Impact of Imprecise Information about the Available Bandwidth on Inter-Cloud Links

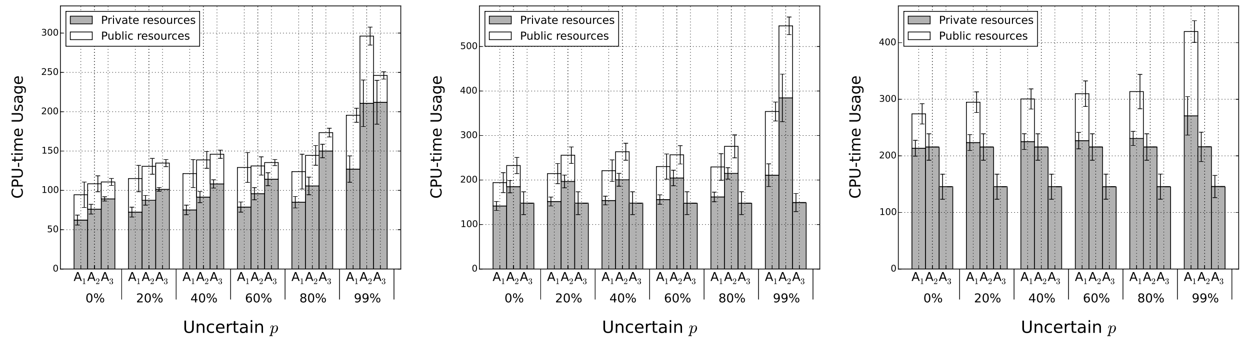
To address the question posed in the introduction, uniformly-distributed errors were introduced into the estimates of the available bandwidth in inter-cloud links, and the increase of CPU usage and the achievement of the target objectives were assessed. The scheduling algorithms were *not* changed to cope with inaccurate inaccurate estimates of the available bandwidth in this experiment. For each solution that satisfies the deadline (qualified solution) defined by the scheduler, the estimates of cost and makespan were obtained by applying a random error p to the estimated available bandwidth bw, which is considered to be 100% accurate at the scheduling time.

5.1.1 *Resource demand*

The goal of this evaluation is to assess for different bandwidth availability in the inter-cloud link the needs of leasing public resources to meet execution deadline of workflows. The CPU-time usage for the execution of workflows was evaluated considering three private clouds: A_1, A_2 and A_3 (Table) 1). Workflow templates with 50, 100, 200 and 300 tasks for three different applications were employed in the evaluation. For each workflow template, twenty workflows were created with different values of the weight of nodes and edges, for a total of 240 workflows. Each workflow was simulated using 3 different scheduling algorithms, 5 different deadline values, 6 inter-cloud estimates of available bandwidth and 11 different values of p. Therefore, results were derived from an extensive set of $712, 800$ simulations.

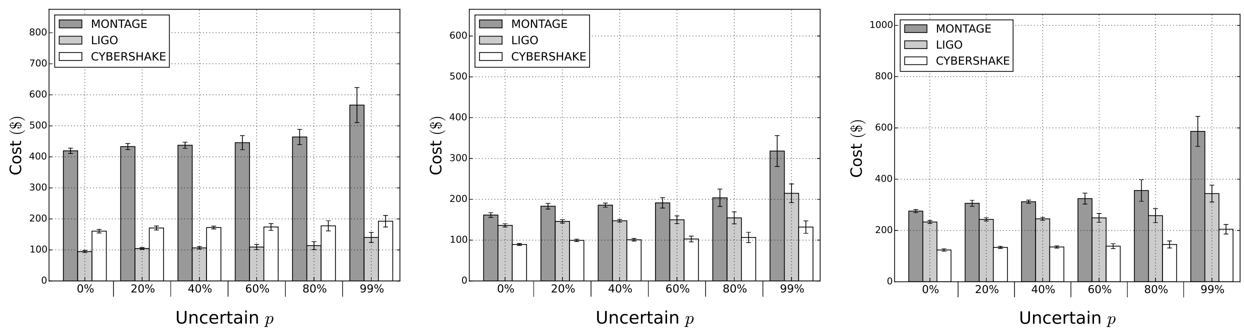
2168-7161 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2469650, IEEE Transactions on Cloud Computing



(a) $\mathcal{D}_{\mathcal{G}} = \mathcal{T}_{max} \times 2/7$ (b) $\mathcal{D}_{\mathcal{G}} = \mathcal{T}_{max} \times 4/7$ (c) $\mathcal{D}_{\mathcal{G}} = \mathcal{T}_{max} \times 6/7$

Fig. 4. Averages of CPU usage of private and public resources of the HCOC scheduler for scheduling the CyberShake application of 50 tasks when the measured inter-cloud available bandwidth of 30 Mbps was varied from 0% to 99% for all private cloud sizes A_1, A_2 and A_3 of the Table 1.



(a) Cost – HCOC scheduler (b) Cost – adapted-PSO scheduler (c) Cost – PSO scheduler

Fig. 5. Barplots of the cost estimates given by the HCOC, adapted-PSO and PSO schedulers to schedule 50-task workflow application when the error in the estimated available bandwidth of 40 Mbps varies from 0% to 99%. For HCOC and adapted-PSO schedulers, the deadline value was $\mathcal{D}_{\mathcal{G}} = \mathcal{T}_{max} \times 4/7$.

Figure 4 shows the averages CPU usage in the private, as well as the public cloud for the HCOC scheduler to schedule a workflow with 50 tasks of the CyberShake application when the measured inter-cloud available bandwidth of a 30 Mbps link was varied from 0% to 99%, with this usage shown as a function of the size of the private cloud and the p value. Figure 4 evinces the increase in CPU-time usage with increasing uncertainty in the available bandwidth estimates (for a better visualization, the results for p values of 10%, 30%, 50%, 70% and 90% are not plotted). The increase in data transfer times implies longer periods of CPU usage because the jobs need to wait longer for the data they depend on. This increase in CPU-time usage increases the idle periods during which the CPU is allocated and becomes unavailable for processing other dependent jobs, which can lead to the need to lease resources from the public cloud. Figure 4 shows the increase in CPU-time usage triggered by the uncertainty in the available bandwidth estimates.

Figure 4 shows results for three different deadline values: a low, a median and a high value, each expressed as a proportion of the $\mathcal{T}_{max}$ value. In most cases, the low deadline can not be met when using the private cloud A_3, and resources from the public cloud must consequently be leased (Figure 4a). However, for a median deadline, resources in the private cloud A_3 were generally sufficient, which did not happen when the workflows are executed in private cloud A_1 and A_2 (Figure 4b). For a high deadline value, execution only on the private cloud A_1 was not satisfatory (Figure 4c). In general, private clouds A_2 and A_3 have either sufficient or nearly sufficient resources to process the required workload. Because the aim of this study is to analyze the impact of the uncertainty of the available bandwidth in

TABLE 2
$\%$ of disqualified solutions of Figure 5

		Uncertain $p\%$					
		0	20	40	60	80	99
HCOC	Montage	12%	58%	64%	82%	91%	92%
	LIGO	42%	47%	49%	56%	62%	79%
	CyberShake	1%	22%	24%	26%	28%	63%
Adapted-PSO	Montage	0%	16%	17%	18%	39%	95%
	LIGO	1%	20%	20%	25%	24%	74%
	CyberShake	3%	19%	21%	19%	23%	49%

2168-7161 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2469650, IEEE Transactions on Cloud Computing

the inter-cloud links, results will be shown only for the private cloud A_1 since its resources were not sufficient to meet the application deadline, independent of the workflow application, scheduling algorithm, deadline value, and available bandwidth in the inter-cloud links.

5.1.2 Failure to Achieve the Established Objective

Various simulations were performed to assess the impact on the achievement of the objectives expressed in the objective function of the algorithms. This evaluation involved $1,200$ workflows (3 types of application workflow templates $\times$ 4 different sizes per template $\times$ 100 workflows per application and size). These workflows were simulated by using 6 different available bandwidth values, 11 p values and 5 deadline values. The evaluation summarizes the outcome of an extensive set of $79,200$ simulations for the PSO scheduler and $396,000$ simulations for the HCOC and adapted-PSO schedulers. The number of simulations using the PSO scheduler was five times lower since the deadline value is not used as part of the input data. Figure 5 shows the cost estimates given by the HCOC, adapted-PSO and PSO schedulers in scheduling a 50-task workflow application when the measured inter-cloud available bandwidth of a 40 Mbps link was between 0% and 99%.

The Montage and CyberShake applications have a similar structural pattern in which the DAG jobs send the same intermediate file to two or more successor jobs. Moreover, depending on how the schedule was produced, sending multiple copies of the intermediate file can increase the traffic in the inter-cloud links [24]. Because the PSO algorithm does not consider the available bandwidth between resources but only the cost of data transmission from/to the public cloud, unnecessary flooding of the inter-cloud links can occur. This implies an increase of the execution cost of the Montage and CyberShake applications (Figure 5c). For example, when applying an uncertainty value of 20% to the inter-cloud available bandwidth of a 40 Mbps that was used to produce the original schedule, the average cost of running the Montage application in hybrid clouds increased by 12%, while for an uncertainty value of 99%, the cost doubled. For CyberShake, the cost increased by 9% for a p value of 20%, while for a p value of 99% the cost doubled as well. Although the LIGO application sends fewer duplicate intermediate files to other dependent jobs, which contributes with less traffic in the inter-cloud links, the uncertainty in the available bandwidth estimates cooperated with the increase in the cost estimates. The cost increased 4% for $p = 20\%$ and only 47% for $p = 99\%$.

For each qualified schedule, a variability $p > 0$ in the available bandwidth estimates was introduced by the simulator that checks if the schedule remained qualified. Table 2 shows the percentage of solutions that missed the deadline (disqualified solutions) for the HCOC and adapted-PSO schedulers. Schedules produced by the PSO scheduler were not classified as either qualified and disqualified since it is a cost-aware scheduler.

Because both HCOC and the adapted-PSO are deadline- and cost-aware schedulers, they are expected to produce schedules that meet the deadline with low costs. However, the effect of errors in the inter-cloud available bandwidth estimates resulted in increased makespans, which caused the deadline to be missed. For example, of the 88% of the solutions for Montage produced by HCOC that were eligible when $p = 0\%$, 58% were disqualified for $p = 20\%$, and 91% were disqualified for $p = 80\%$. The schedules produced by adapted-PSO were less impacted. For example, of the 100% of the Montage solutions that were eligible when $p = 0\%$, 16% were disqualified when $p = 20\%$, and 39% were disqualified when $p = 80\%$. The cost estimates provided by HCOC, increased by 5% when the p value varied from 0% to 20%, and 30% when p value varied from 0% to 80%. The cost estimates provided by the adapted-PSO also increased with increasing values of p. The cost estimate increased by 14% when the p value varied from 0% to 20%, and 28% when p value varied from 0% to 80%. Similar trends were obtained for the LIGO and CyberShake workflows; i.e, increasing the error in the estimated available bandwidth, increases the number of disqualified schedules as well as underestimates costs.

This section showed results for schedulers that were not designed to address with inaccurate information about the available bandwidth and produced schedules with misleading cost and makespan estimates. The next section will show that the use of a deflating factor for the available bandwidth that is used as an input to the scheduler can reduce the disqualification of schedules, by avoiding underestimates of the makespan and cost. Because the original PSO scheduler does not consider the available bandwidth to produce schedules, it was not evaluated hereafter.

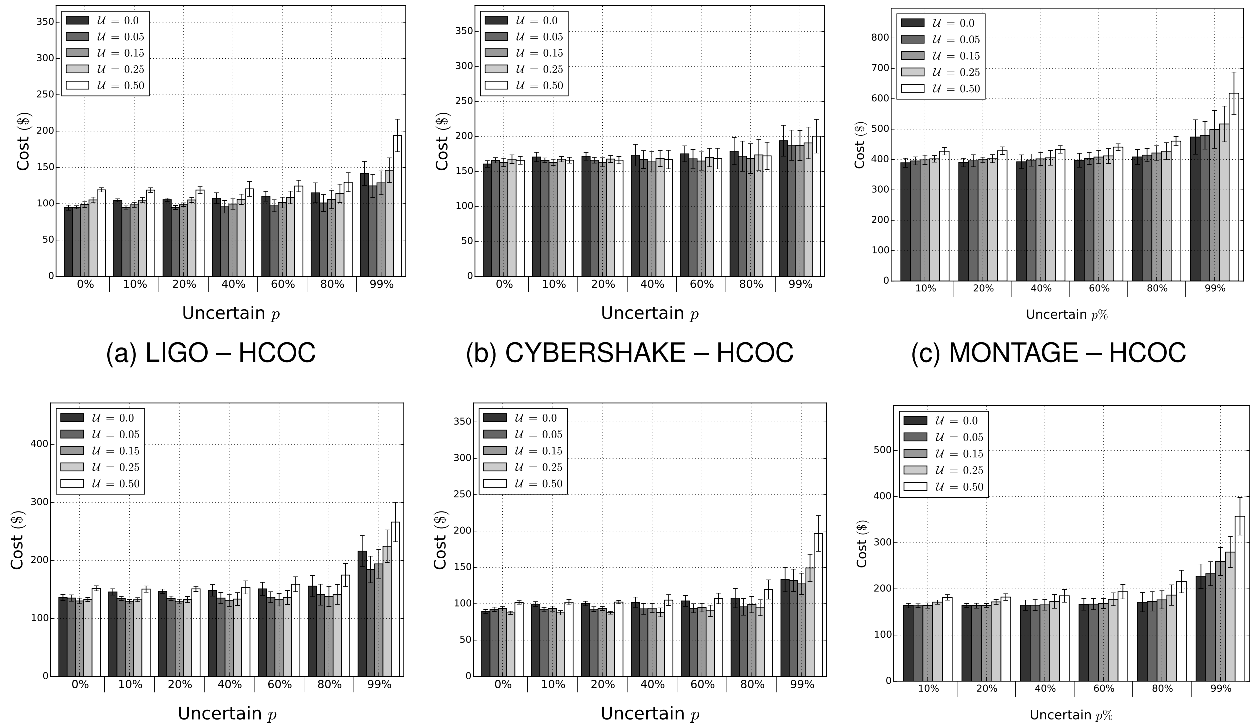
5.2 Using a Fixed Deflating $\mathcal{U}$ value

Deflating the measured available bandwidth value using the $\mathcal{U}$ value and providing the deflated bandwidth estimate to the scheduler at the scheduling time will hopefully reduce the negative impact of imprecise bandwidth information on the production of schedules. The results from the next experiments assess the extent to which a deflating factor $\mathcal{U}$ mitigates the negative impact of imprecise information about the available bandwidth on the produced schedules.

One thousand two hundred workflows were employed in this evaluation (3 types of application workflow template $\times$ 4 different workflow size per template $\times$ 100 workflows per application and size). These workflows were used in simulations with 6 different available bandwidth values, 11 p values, 5 deadline values and 6 deflating $\mathcal{U}$ values. The experiment used a fixed $\mathcal{U}$ value, $\mathcal{U} \in \{0, 5, 15, 25, 35, 45, 50\}$, but for better visualization, results for $\mathcal{U} = 35$ and $\mathcal{U} = 45$ were not plotted.

2168-7161 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2469650, IEEE Transactions on Cloud Computing



(a) LIGO – HCOC (b) CYBERSHAKE – HCOC (c) MONTAGE – HCOC

(d) LIGO – Adapted-PSO (e) CYBERSHAKE – Adapted-PSO (f) MONTAGE – Adapted-PSO

Fig. 6. Barplots of the costs of disqualified solutions when the measured available bandwidth of 40 Mbps was deflated by a fixed $\mathcal{U}$ value and provided as input to the HCOC (top) and PSO-adpted (bottom) schedulers to schedule the applications workflows of 50 tasks with the deadline value equal to $\mathcal{D}_{\mathcal{G}} = \mathcal{T}_{max} \times 4/7$. The x-axis represents the estimated error in the available bandwidth which varies from 10% to 99%.

This experiment summarizes the outcome of a set of $2,376,000$ simulations for each deadline- and cost-aware scheduler. Figure 6 shows the cost estimates given by the HCOC and adapted-PSO schedulers to schedule a 50-task workflow application when the uncertainty of the measured inter-cloud available bandwidth of 40 Mbps and 60 Mbps varied from 0% to 99%. Table 3 shows the percentage of disqualified solutions shown in Figure 6 when the available bandwidth estimates were deflated by a fixed $\mathcal{U}$ value.

The results show a reduction in the percentage of disqualified schedules when the deflating $\mathcal{U}$ value was employed. This trend was present for most cases with the three types of applications. For example, using HCOC with $\mathcal{U} = 0.05$ and $p = 60\%$, there was an increase of 10%, 21% and 22% in the number of qualified solutions for the LIGO, CyberShake and Montage applications, respectively; while for the adapted-PSO, the number of qualified solutions increased 21%, 20% and 8% for LIGO, CyberShake and Montage application, respectively.

However, the use of a large deflating $\mathcal{U}$ value increased the number of disqualified solutions. For HCOC, when $\mathcal{U} = 0.5$ and $p = 60\%$, the number of disqualified solutions for LIGO, CyberShake and Montage increased by 15%, 5% and 13%, respectively; while using the adapted-PSO caused increases of 15%, 20% and 24% for LIGO, CyberShake and Montage, respectively. This experiment also suggests that the number of disqualified solutions can be reduced in the presence of high variability of the available bandwidth. For example, for an uncertainty value of 99% and $\mathcal{U} = 0.05$, the numbers of qualified solutions produced by HCOC increased by 23% for CyberShake and 3% for LIGO. For the adapted-PSO, the increases were 2% for Montage, 21% for LIGO, and 7% for CyberShake. In general, regardless of the scheduler and application, deflating the available bandwidth estimates increases the number of schedules that satisfy the deadline in the presence of variabilities of the available bandwidth of inter-cloud links.

The use of a non-null deflating factor $\mathcal{U}$ did not imply an increase in estimated costs, as shown in Figure 6. For example, using HCOC to schedule the LIGO application when $p = 40\%$ led to costs 13% lower for $\mathcal{U} = 0.05$ instead of $\mathcal{U} = 0$. In other cases, the use of a large deflating $\mathcal{U}$ value resulted in a further reduction in costs instead of using a short $\mathcal{U}$ value. For example, when the adapted-PSO algorithm was used to schedule CyberShake application for $p = 10\%$, the cost was 6% lower when using $\mathcal{U} = 0.25$ instead of $\mathcal{U} = 0.05$. The decrease in cost was possible since the set of virtual machines selected by the algorithm to perform the same task scheduling is different when the available bandwidth estimates deflated by different $\mathcal{U}$ values.

These experiments suggest that the decrease in the

2168-7161 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2469650, IEEE Transactions on Cloud Computing

TABLE 3
% of disqualified solutions of the evaluation for the HCOC scheduler (left) and the Adapted-PSO (right) of Figure 6

HCOC		Uncertain p%					
	$\mathcal{U}$ value	10	20	40	60	80	99
LIGO	0.0	52%	55%	61%	67%	81%	85%
	0.05	43%	53%	53%	57%	56%	66%
	0.15	45%	46%	47%	59%	62%	70%
	0.25	46%	47%	47%	60%	58%	75%
	0.50	79%	80%	80%	82%	86%	88%
CYBER-SHAKE	0.0	16%	17%	22%	23%	31%	80%
	0.05	2%	2%	2%	2%	5%	13%
	0.15	1%	1%	1%	1%	6%	24%
	0.25	6%	6%	6%	6%	10%	22%
	0.50	22%	24%	26%	28%	37%	57%
MONTAGE	0.0	13%	58%	64%	82%	91%	92%
	0.05	1%	49%	52%	60%	79%	85%
	0.15	1%	74%	73%	75%	81%	87%
	0.25	2%	77%	79%	83%	87%	89%
	0.50	74%	90%	92%	95%	95%	95%

Adapted-PSO		Uncertain p%					
	$\mathcal{U}$ value	10	20	40	60	80	99
LIGO	0.0	18%	19%	17%	22%	22%	61%
	0.05	3%	3%	3%	3%	3%	6%
	0.15	11%	11%	11%	11%	12%	13%
	0.25	14%	14%	14%	14%	16%	19%
	0.50	20%	20%	20%	32%	35%	40%
CYBER-SHAKE	0.0	18%	17%	18%	25%	24%	52%
	0.05	5%	5%	5%	5%	7%	8%
	0.15	10%	10%	10%	11%	11%	13%
	0.25	14%	14%	14%	14%	15%	19%
	0.50	22%	23%	24%	40%	40%	45%
MONTAGE	0.0	14%	15%	17%	13%	14%	67%
	0.05	2%	4%	4%	5%	7%	12%
	0.15	12%	13%	13%	15%	17%	19%
	0.25	14%	19%	19%	22%	26%	23%
	0.50	21%	33%	33%	37%	53%	65%

number of disqualified solutions depends on the $\mathcal{U}$ value and that there is no clear trend for choosing an optimum value. This motivated the adoption of a procedure that considers the execution history to determine the most appropriate $\mathcal{U}$ value.

5.3 Using the Proposed Mechanism to Calculate the $\mathcal{U}$ value

When the $\mathcal{U}$ value is derived using the proposed procedure, qualified schedules were produced for almost all p values. The analysis of the execution history is the key to determining a specific deflating factor for the $\mathcal{U}$ value. The results show the extent to which using the $\mathcal{U}$ value that is computed considering the execution history can mitigate the negative impact of imprecise information about the available bandwidth. This section shows the results of simulations when the employment of the proposed procedure to calculate a deflating factor $\mathcal{U}$ to the measured available bandwidth value was used as an input to the scheduler. This experiment summarizes a set of $396,000$ simulations for each deadline- and cost-aware scheduler (3 application workflow templates $\times$ 4 different sizes $\times$ 100 workflows per size $\times$ 6 available bandwidth values $\times$ 11 p values $\times$ 5 deadline values).

A comparison of the results obtained when the proposed MLR procedure was not employed (Section 5.1.2) showed a decrease in the percentage of disqualified schedules. This is true independent of the p value for the three types of applications and for both schedulers. For example, by comparing Table 2 with 4 using the employment of the HCOC scheduler for a measured available bandwidth of 40 Mbps was deflated by a specific $\mathcal{U}$ value with $p = 20\%$, the number of qualified solutions increased by 10%, 22% and 51% for the LIGO, CyberShake and Montage applications, respectively. When the adapted-PSO was employed, the number of qualified solutions increased by 17%, 16% and 13% for the LIGO, CyberShake and Montage applications, respectively. Even in scenarios with high uncertainties in the available bandwidth estimates, a noticeable increase in the number of qualified solutions was found. For example, using HCOC and $p = 99\%$, the number of qualified solutions for the LIGO, CyberShake and Montage applications increased by 35%, 35% and 28%, respectively; while for the adapted-PSO this increased by 31%, 19% and 12%, respectively.

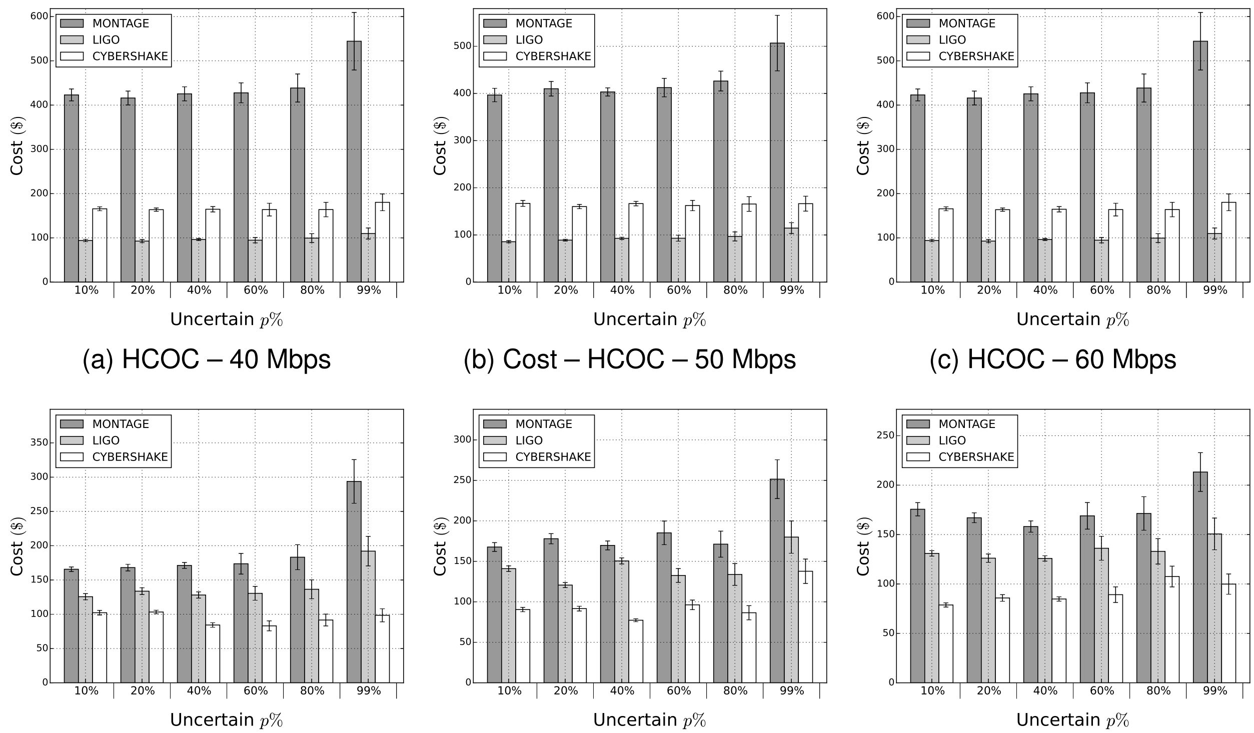
The use of a specific $\mathcal{U}$ value not only increases the number of qualified solutions, but also achieves slightly less expensive schedules, as shown in Figure 7 . For example, using HCOC and $p = 40\%$, the costs of the LIGO, CyberShake and Montage applications were respectively 9%, 5% and 4% lower than the original cost estimates (Figures 6 with $\mathcal{U} = 0$). The adapted-PSO scheduler reduced the costs of these LIGO, CyberShake and Montage applications by 5%, 4% and 2%, respectively. The results show that when the deflated values for the measured available bandwidths are used, the scheduler produces schedules with higher utilisation of resources, thus ensuring that the schedules meet the deadlines as well as reducing misleading cost estimates.

A $\mathcal{U}$ value computed by the proposed procedure is dependent on several parameters such, as the type of workflow, the deadline value, the value of the measured available inter-cloud bandwidth, the error occurred in previous estimates of the makespan and the cost due to the presence of an uncertainty value on the bandwidth value. Because several parameters can influence the $\mathcal{U}$ value, applying a fixed value as in the previous subsection does not produce a single best deflating factor value, because it can either underestimate or overestimate the available bandwidth, which leads to a deadline violation or unnecessary rental of processing power on public clouds. Thus, for a particular workflow application, a deadline value, an available bandwidth value and an uncertainty value, the execution history assists the proposed procedure to determinate a specific $\mathcal{U}$ value that tries to avoid the disqualification of the schedule.

The computation of the $\mathcal{U}$ value depends on the size of the execution history. The greater the number of tuples $\langle bm, p, \mathcal{U}\rangle$ in the $\mathcal{H}_k$ subset, the greater the amount of information used by the multiple linear regression to yield the linear equation $\mathcal{U} = f(x, y)$. For example, in our previous work [14], the multiple linear regression were obtained by using 25%, 50%, 75% and 100% of

2168-7161 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2469650, IEEE Transactions on Cloud Computing



(a) HCOC – 40 Mbps (b) Cost – HCOC – 50 Mbps (c) HCOC – 60 Mbps

(d) PSO-adapted – 40 Mbps (e) Cost – PSO-adapted – 50 Mbps (f) PSO-adapted – 60 Mbps

Fig. 7. Barplots of the costs when the measured available bandwidth 40 Mbps, 50 Mbps and 60 Mbps was deflated by a specific $\mathcal{U}$ value and provided as input to the HCOC (top) and PSO-Adapted (bottom) schedulers to schedule the applications workflows of 50 tasks with the deadline value equal to $\mathcal{D}_{\mathcal{G}} = \mathcal{T}_{max} \times 4/7$.

the historical data. When using half of the execution history, better results were obtained when a quarter of this historical data was used. A simulation pattern of improvement was also found when three quarters of the historical data were used in the regression instead of half of it. However, we noticed that there was little improvement in the results when the amount of historical data considered increased from 75% to 100%. Therefore, since the amount of execution history in this paper is the same as the previous paper, we consider only the employment of 100% of the historical data in the evaluation of the proposed procedure. Due to space limitations in [14], we only presented the evaluations results when 50% and 100% of the historical were considered.

The results shown in this section indicate that the employment of the proposed mechanism was able to reduce the impact of misleading estimates of the available bandwidth values and produced a large number of qualified solutions. When the proposed procedure is used with schedulers that were not designed to cope with inaccurate information about the available bandwidth, the costs were not higher than when the $\mathcal{U}$ value was not used, which implies that a higher utilization of resources in the private cloud was achieved.

6 Related Work

Schedulers for cloud computing determine which jobs should use local resources and which resources should be leased from the public cloud to minimize the workflow execution time (makespan) and overall costs. Resource allocation and workflow scheduling in heterogeneous systems, such as hybrid clouds, have been intensively studied [7], [8], [13], [25]. Nevertheless, scheduling workflows is an NP-Complete problem [2]; as a result, alternatives are crucial for finding approximate optimal solutions. Heuristics [7], [13], meta-heuristics [8] and linear programming [25] are tools that are commonly used to solve scheduling problems.

Imprecision in the input information to the scheduler has been approached in the literature using several reactive techniques, such as dynamic scheduling [26], rescheduling [27], and self-adjusting [28]. Reactive techniques are best suited of fluctuations in resource availability during the execution of the workflow. Because reactive mechanisms are based on monitoring of resource utilization and on the performance of the execution of applications, imprecise information can be generated by the intrusion effects of monitoring tools. As a result, unnecessary job migration can occur in addition to monitoring overhead [29].

An algorithm based on fuzzy optimization to schedule workflows in grids under uncertainty in both application demands and resource availability was presented in [29]. Although the authors compared the proposed algorithm to static schedulers for heterogeneous systems, the impact of imprecise estimates of the available

2168-7161 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2469650, IEEE Transactions on Cloud Computing

TABLE 4
% of disqualified solutions of the evaluation for the HCOC scheduler (left) and the Adapted-PSO (right) of Figure 7

HCOC		Uncertain $p\%$					
		0	20	40	60	80	99
40 Mbps	LIGO	33%	37%	40%	47%	60%	70%
	CyberShake	0%	0%	3%	7%	7%	50%
	Montage	3%	7%	23%	27%	77%	90%
50 Mbps	LIGO	30%	20%	23%	37%	43%	70%
	CyberShake	0%	0%	3%	0%	7%	40%
	Montage	7%	0%	3%	7%	33%	87%
60 Mbps	LIGO	3%	3%	3%	0%	7%	43%
	CyberShake	3%	10%	0%	3%	3%	3%
	MONTAGE	7%	3%	3%	13%	20%	83%

Adapted-PSO		Uncertain $p\%$					
		0	20	40	60	80	99
40 Mbps	LIGO	3%	3%	3%	0%	7%	43%
	CyberShake	3%	3%	0%	3%	3%	30%
	Montage	7%	3%	3%	13%	20%	83%
50 Mbps	LIGO	30%	20%	23%	37%	43%	70%
	CyberShake	0%	3%	0%	0%	0%	20%
	Montage	7%	0%	3%	7%	33%	87%
60 Mbps	LIGO	0%	0%	0%	0%	3%	13%
	CyberShake	0%	0%	0%	3%	3%	7%
	Montage	7%	7%	3%	7%	10%	53%

bandwidth were not evaluated for utility grids and clouds. Rahman et al. [5] characterized a measurement study to characterize the impact of virtualization on the performance of the network of the Amazon EC2. Through analysis of packet delay, TCP/UDP throughput and packet loss at Amazon EC2 virtual machines, they concluded that unstable network throughput and large delay variations can negatively impact the performance of scientific applications running on clouds. Indeed, the authors argued that strategies for task-mapping needs customization to handle inaccurate information on the available bandwidth.

Although these papers advance the workflow scheduling problem in cloud computing environments, none of them considers the impact of bandwidth uncertainty on the schedule. In other words, the scheduling algorithms assume that the estimated available bandwidth at the scheduling time is 100% precise. However, this assumption is not always true, and the schedules can be negatively affected during the execution of the workflow. Tools for estimating available bandwidth produce estimates with large variability [30], and schedules can be negatively affected by misleading bandwidth estimates during the execution of workflows, especially in hybrid clouds. Few works in the literature consider the impact of the bandwidth uncertainty in communication channels at the scheduling time.

7 Conclusions

The performance of workflow execution in hybrid clouds depends largely on the available bandwidth in the inter-cloud links since this bandwidth determines the data transfer time between tasks residing in different clouds. The available bandwidth in inter-cloud links fluctuates considerably since Internet links are shared resources, this fluctuation contributes to imprecision in estimates of the available bandwidth, and the use of these estimates as an input data to a scheduler usually results in increase in the workflow execution time, as well as costs. The negative impact of imprecise estimation of available bandwidth can be reduced by the adoption of scheduling schemes which take into consideration uncertainties in input values.

The mechanism proposed in this paper applies a deflating factor to the available bandwidth value that is provided as input to the scheduler. The deflating factor is computed using multiple linear regression, and the execution history of a specific workflow is used in the computation. The procedure is employed before the information about the currently available bandwidth is provided to the scheduler.

We evaluated the proposed multiple linear regression mechanism using simulations and three different scheduling algorithms to schedule three different real-life scientific workflow application. The results provide evidence that, in general, the proposal is capable of producing schedules with less misleading cost and makespan estimates, as well as increasing the number of qualified solutions (solutions within the application deadline). More specifically the proposed procedures reduce the number of disqualified solution by up to 70%, with costs reduced by up to 10%.

As Future work, we intend to implement the proposed procedure on a real testbed in order to verify the efficacy of the mechanism in a hybrid cloud environment. Besides that, we also intend to improve the performance of the proposed mechanism by updating the historical information as well as by deriving criteria for selecting obsolete information to be deleted from the database in order to maintain the scalability of the procedure.

Acknowledgment

The authors would like to thank CNPq, CAPES and State of São Paulo Research Foundation (FAPESP 2012/02778-6 and 2014/08607-4) for the financial support.

References

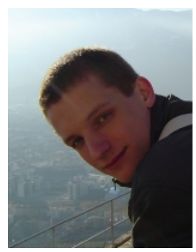
[1] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 7–18, 2010.
[2] M. L. Pinedo, *Scheduling: Theory, Algorithms, and Systems*. Springer Publishing Company, 2008.
[3] T. A. L. Genez, L. F. Bittencourt, N. L. S. Da Fonseca, and E. R. M. Madeira, "Refining the estimation of the available bandwidth in inter-cloud links for task scheduling," in *IEEE Global Communications Conference (GLOBECOM)*, Dec 2014.
[4] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: Research problems in data center networks," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 68–73, Dec. 2008.
[5] G. Wang and T. S. E. Ng, "The impact of virtualization on network performance of Amazon EC2 data center," in *IEEE Conf. on Information Communications (INFOCOM)*, 2010, pp. 1163–1171.

2168-7161 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2469650, IEEE Transactions on Cloud Computing

[6] C. Guo, G. Lu, H. J. Wang, S. Yang, C. Kong, P. Sun, W. Wu, and Y. Zhang, "SecondNet: A data center network virtualization architecture with bandwidth guarantees," in *6th International Conference*. ACM, 2010, pp. 15:1–15:12.
[7] L. F. Bittencourt and E. R. M. Madeira, "HCOC: a cost optimization algorithm for workflow scheduling in hybrid clouds," *Internet Services and Applications*, vol. 2, no. 3, pp. 207–227, 2011.
[8] S. Pandey, L. Wu, S. Guru, and R. Buyya, "A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments," in *IEEE International Conference on Advanced Information Networking and Applications*, april 2010, pp. 400 –407.
[9] E. Deelman, G. Singh, M.-H. Su, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, G. B. Berriman, J. Good, A. Laity, J. C. Jacob, and D. S. Katz, "Pegasus: A framework for mapping complex scientific workflows onto distributed systems," *Scientific Programming Journal*, vol. 13, no. 3, pp. 219–237, 2005.
[10] A. Ramakrishnan, G. Singh, H. Zhao, E. Deelman, R. Sakellariou, K. Vahi, K. Blackburn, D. Meyers, and M. Samidi, "Scheduling data-intensive workflows onto storage-constrained distributed resources," in *IEEE International Symposium on Cluster Computing and the Grid (CCGRID)*, May 2007, pp. 401–409.
[11] Y. Zhao, J. Dobson, I. Foster, L. Moreau, and M. Wilde, "A notation and system for expressing and executing cleanly typed workflows on messy scientific data," *ACM SIGMOD Record*, vol. 34, no. 3, pp. 37–43, Sep. 2005.
[12] J. Annis, Y. Zhao, J. Voeckler, M. Wilde, S. Kent, and I. Foster, "Applying chimera virtual data concepts to cluster finding in the sloan sky survey," in *ACM/IEEE Conference Supercomputing*, Nov 2002, pp. 56–56.
[13] L. F. Bittencourt, E. R. M. Madeira, and N. L. S. da Fonseca, "Scheduling in hybrid clouds," *IEEE Communications Magazine*, vol. 50, no. 9, pp. 42–47, 2012.
[14] L. Bittencourt, E. Madeira, and N. Da Fonseca, "Impact of communication uncertainties on workflow scheduling in hybrid clouds," in *IEEE Global Communications Conference (GLOBECOM)*, Dec 2012, pp. 1623–1628.
[15] O. Biran, A. Corradi, M. Fanelli, L. Foschini, A. Nus, D. Raz, and E. Silvera, "A stable network-aware vm placement for cloud systems," in *2th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, May 2012, pp. 498–506.
[16] K. LaCurts, S. Deng, A. Goyal, and H. Balakrishnan, "Choreo: Network-aware task placement for cloud applications," in *Conference on Internet Measurement Conference*, 2013, pp. 191–204.
[17] R. F. da Silva, W. Chen, G. Juve, K. Vahi, and E. Deelman, "Community resources for enabling research in distributed scientific workflows," in *10th IEEE int'l Conference on e-Science*, Nov 2014.
[18] L. F. Bittencourt and E. R. M. Madeira, "A performance-oriented adaptive scheduler for dependent tasks on grids," *Concurrency and Computation: Prac. and Exp.*, vol. 20, no. 9, pp. 1029–1049, 2008.
[19] H. Topcuoglu, S. Hariri, and M.-Y. Wu, "Performance-effective and low-complexity task scheduling for heterogeneous computing." *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 3, pp. 260–274, 2002.
[20] H. Casanova, A. Legrand, D. Zagorodnov, and F. Berman, "Heuristics for scheduling parameter sweep applications in grid environments," in *9th Heterogeneous Computing Wksh.*, 2000, pp. 349–363.
[21] S. Sanghrajka, N. Mahajan, and R. Sion, "Cloud performance benchmark series – network performance: Amazon EC2," Stony Brook University, Tech. Rep., 2011.
[22] S. Sanghrajka and R. Sion, "Cloud performance benchmark series – network performance: Rackspace.com," Stony Brook University, Tech. Rep., 2011.
[23] R. Wolski, N. T. Spring, and J. Hayes, "The network weather service: A distributed resource performance forecasting service for metacomputing," *Future Gener. Comput. Syst.*, vol. 15, no. 5-6, pp. 757–768, Oct. 1999.
[24] G. Juve, A. Chervenak, E. Deelman, S. Bharathi, G. Mehta, and K. Vahi, "Characterizing and profiling scientific workflows," *Future Gener. Comput. Syst.*, vol. 29, no. 3, pp. 682–692, Mar. 2013.
[25] T. A. L. Genez, L. F. Bittencourt, and E. R. M. Madeira, "Workflow Scheduling for SaaS / PaaS Cloud Providers Considering Two SLA Levels," in *IEEE/IFIP Network Operations and Management Symposium (NOMS 2012)*. IEEE, 2012.
[26] G. Allen, D. Angulo, I. Foster, G. Lanfermann, C. Liu, T. Radke, E. Seidel, and J. Shalf, "The cactus worm: Experiments with dynamic resource discovery and allocation in a grid environment," *Journal of High Performance Computing App.*, vol. 15, p. 2001, 2001.
[27] R. Sakellariou and H. Zhao, "A low-cost rescheduling policy for efficient mapping of workflows on grid systems," *Scientific Programming*, vol. 12, no. 4, pp. 253–262, Dec. 2004.
[28] D. M. Batista, N. L. S. da Fonseca, F. K. Miyazawa, and F. Granelli, "Self-adjustment of resource allocation for grid applications," *Computer Networks*, vol. 52, no. 9, pp. 1762–1781, Jun. 2008.
[29] D. M. Batista and N. L. S. da Fonseca, "Robust scheduler for grid networks under uncertainties of both application demands and resource availability," *Computer Networks*, vol. 55, no. 1, pp. 3–19, 2011.
[30] D. M. Batista, L. J. Chaves, N. L. Fonseca, and A. Ziviani, "Performance analysis of available bandwidth estimation tools for grid networks," *Journal of Supercomputing*, vol. 53, no. 1, pp. 103–121, Jul. 2010.

Thiago Augusto Lopes Genez received his bachelor's degree in Computer Science from the State University of Londrina, Brazil, in 2009, and his master's degree in Computer Science from the University of Campinas, Brazil, in 2012. He is currently a Ph.D. student in the University of Campinas, Brazil, and his research interests involve aspects of cloud computing. He has been researching algorithms for the problem of workflows scheduling in hybrid clouds and public cloud.



Luiz Fernando Bittencourt is an Assistant Professor of Computer Science at the University of Campinas, Brazil. He received his Ph.D. in 2010 from the University of Campinas, and his research interests involve aspects of resource allocation and management in heterogeneous distributed systems, focusing on algorithms for scheduling in hybrid clouds. He is the recipient of the 2013 IEEE ComSoc Latin America Young Professional Award.

Nelson Luis Saldanha da Fonseca received his Ph.D. degree in Computer Engineering from The University of Southern California in 1994. He is Full Professor at Institute of Computing of The University of Campinas, Campinas, Brazil. He has published $350+$ papers and supervised $50+$ graduate students. He is Director for Conference Development of the IEEE Communications Society (ComSoc). He served as Vice President Member Relations of the ComSoc, Director of Latin America Region and Director of on-line Services. He is the recipient of the 2012 IEEE Communications Society (ComSoc) Joseph LoCicero Award for Exemplary Service to Publications, the Medal of the Chancelor of the University of Pisa (2007) and the Elsevier Computer Network Journal Editor of Year 2001 award. He is past EiC of the IEEE Communications Surveys and Tutorials. He is Senior Editor for the IEEE Communications Surveys and Tutorials and Senior Editor for the IEEE Communications Magazine, a member of the editorial board of Computer Networks, Peer-to-Peer Networking and Applications, Journal of Internet Services and Applications and International Journal of Communication Systems. He founded the IEEE LATINCOM and was technical chair of over 10 IEEE conferences

Edmundo Roberto Mauro Madeira is a Full Professor of Computer Science at the University of Campinas, Brazil, and member of the editorial board of the Journal of Network and Systems Management (JNSM), Springer. He received his Ph.D. in Electrical Engineering from the University of Campinas in 1991, and his research interests involve aspects of computer networks, distributed system management, grid and cloud computing, mobility aware technologies and wireless networks.

2168-7161 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.