On the Performance-cost Tradeoff for Workflow Scheduling in Hybrid Clouds

Thiago A. L. Genez, Luiz F. Bittencourt, Edmundo R. M. Madeira
Institute of Computing – University of Campinas – UNICAMP
Av. Albert Einstein, 1252, 13083-852 – Campinas – São Paulo – Brazil
e-mail: {thiagogenez, bit, edmundo}@ic.unicamp.br

***Abstract*—The use of public clouds to extend the capacity of private resources has become a popular manner to achieve elasticity in the available in-house computational power to meet deadlines. Schedulers for such hybrid clouds have the role of deciding which types of instances should be leased in a pay-per-use basis to fulfill application demands. Often these schedulers assume a costless private cloud, which may not be a real scenario: aggregated costs can come from maintenance, energy consumption, administrative staff, and, more recently, from leasing a private datacenter from hardware manufacturers or a virtual private cloud from public providers. Based on these more realistic assumptions, we assess the behavior of a cloud scheduling algorithm in the face of different pricing relation between private and public resources. We show there exist a trade-off between using local resources to maximize the utilization of private cloud to minimize the monetary cost with outsourced resources and using leased resources from public cloud to satisfy deadlines, which can lead to idleness in the private cloud. Preliminary results indicate that application deadlines and the cost of private cloud can influence in the private cloud utilization, sometimes outsourcing most executions to public clouds, which leads to increased costs to run the application. We then argue that utilization-aware schedulers are also important to be developed when considering hybrid clouds.**

I. INTRODUCTION

Hybrid clouds are characterized by the combination of computational resources from public and private clouds [1]. In the context of infrastructure as a service (IaaS), a private cloud owner can take advantage of the elasticity provided by on-demand leasing of processing power and storage from an IaaS provider in a pay-per-use basis [2]. The resource leasing may be necessary on peak demand and also when the locally available resources are not sufficient to comply with application requirements. Currently, a *hybrid cloud* is also referred to as *cloud bursting*, where a single organization *bursts* workload to an external cloud in an on-demand basis.

In order to utilize public cloud resources to compose the so-called hybrid cloud, one of the challenges is how to make decisions over the cloud composition. When dealing with a pay-per-use leasing, resources composition must take into account not only quality of service (QoS), but also the monetary costs involved during the resources utilization [3]. The scheduler is the hybrid cloud component responsible for this decision-making process using as input the information about the private and public clouds, application demands, and monetary costs. Schedulers for clouds are concerned in selecting the best public cloud resources that, in conjunction with the private resources, will fulfill the application QoS keeping the monetary costs as low as possible. In general, these schedulers consider a private cloud free of cost [4]–[7], which is not necessarily true. Private clouds present operational and management costs, but, more than that, a private cloud can also be a set of leased resources for private use, as for example the container-hosted HP Performance Optimized Datacenter or the Amazon Virtual Private Cloud.

The goal of this paper is to bring to discussion some aspects involved in the hybrid cloud scheduling when considering that private cloud is not costless. In order to do that, we present and discuss results of workflow scheduling in a hybrid cloud with different relations between costs of private and public cloud resources. More specifically, we aim to show that there are cases where it is necessary to use public cloud resources and let the private idle to fulfill deadlines. Nevertheless, a private cloud idle is not free of cost, and system administrators must bear the cost of using outsourced resources and the cost of maintaining the private cloud idle (or undeutilized).

This paper is organized as follows. Section II presents the related work. Section III describes the conceptual background of the hybrid cloud model considered, while Section IV discusses some important points to be considered in the cost evaluation and performance of the hybrid cloud. Section V presents preliminary results obtained by scheduling workflows with different cost relations in the hybrid cloud resources, and Section VI concludes the paper with the final remarks.

II. RELATED WORK

Deelman et al. show in [8] a cost-effective solution to execute data-intensive applications on public clouds, more specifically to execute entire astronomy application Montage on Amazon EC2[1]. Several plans to execute Montage in the Amazon cloud were analyzed by the authors. Through simulations, they concluded that provisioning the right amount of storage and compute resources, monetary cost can be reduced with no significant impact on application performance. However, an analysis in hybrid cloud has not been studied by the authors. Assunção et al. investigate in [4] the benefits of cloud bursting to improve the performance of requests' response times. They evaluate seven scheduling strategies to utilise paid resources from the public cloud to enhance the capacity of

---

[1] http://aws.amazon.com/ec2/

in-house cluster. The authors show that the naïve scheduling strategy can result in a higher monetary cost under heavy peak demands. In other words, increasing the performance of application scheduling when local cluster is under-utilised can be very expensive. On the other hand, selective backfilling strategy showed a good ratio of slowdown improvement to the money spent for using public cloud. Although these works present several scheduling strategies in hybrid cloud, none of them considers the monetary cost comparison between the local and outsourced resource in order to check up when it is worth using local cluster instead of public paid resources.

Bittencourt and Madeira provide in [2] a brief survey of scheduling algorithms for hybrid clouds and also the impact of communication networks channels between private and public clouds on scheduling decisions. They state that, due to location of such channels in the Internet backbone, the available bandwidth can fluctuates widely. For example, a bandwidth highly inaccurate can increase delays, and hence increasing (i) the execution time (*makespan*) of applications and (ii) the monetary execution cost on public clouds. In other words, an uncertainly bandwidth can result in non-compliance with the application's service level agreements. They concluded that the development of communication-aware or even communication-driven scheduling algorithms is important to meet application's QoS in hybrid clouds. Although this work presents advances on scheduling problems in hybrid cloud, it considers only a costless private cloud.

In this paper we aim to fill a discussion gap where current workflow schedulers are only concerned with QoS guarantees, neglecting the resource utilization. In the case of hybrid clouds, this underutilization can turn into higher monetary costs, since the private cloud owner would pay twice for resources to run an application: once for the underutilized resource in the private cloud, and once for the utilization of the public cloud.

III. HYBRID CLOUD MODEL

Composing a hybrid cloud is one way of taking advantage of the elasticity provided by the cloud computing paradigm. In this paper we consider the user owns an IaaS private cloud and leases resources from public IaaS providers. The private cloud can be a set of resources in the user premises. These resources can be put together as a grid computing [9] or a cluster. The private cloud can also be in the form of leased resources in the public cloud providers premises, such as the Amazon Virtual Private Cloud[2].

When a workflow is submitted, the hybrid cloud management system needs to determine the resources needed to fulfill the QoS needs. In this work, we consider the deadline of the application as one of the main QoS parameters specified during the workflow submission [4]. Based on information about the application CPU and communication demands and hybrid cloud resources performance, the scheduler is responsible for selecting where each workflow component will run, as well as to detect if more (and which) resources are needed to obey the application deadline.

Computational resources are often made available as *virtual machines* (VMs). Thus, virtualization is one of the key technologies involved in the leasing of resources from infrastructure providers. In the IaaS context, virtual machines offer flexibility and server consolidation by splitting physical resources into virtual resources that can be offered to the users as isolated machines. Virtual machines can be utilized in the on-premises private cloud to manage resources, however the private cloud owner may also decide not to use VMs since he/she has total access and control over the infrastructure. On the other hand, if the private cloud is not on-premises, the user must follow the cloud provider policies on the resource usage and management.

In order to lease a virtual machine, a service level agreement (SLA) is established between the client and the provider. The SLA describes the QoS levels that are expected to be observed during the leasing. Common QoS parameters in SLAs for IaaS providers (e.g. Amazon EC2 and Google Compute Engine) are CPU cores, CPU performance, storage space, RAM amount, I/O performance, and network bandwidth. The scheduler takes information about resources from these SLAs and, combined with application demands, runs an optimization approach following pre-determined objectives, such as obey the deadline and minimize monetary costs.

In public clouds, costs involved in the VM leasing are determined by the pricing models adopted by the IaaS provider. VMs are usually charged in a discrete-time basis, therefore a VM, for example, leased for exactly 3 hours will end up costing the same as the same VM leased for 2 and a half hours. Based on the Amazon EC2 cloud, we classify the pricing in three different categories:

- On-demand: VMs are leased on-demand for the price listed by the provider according to its configuration (instance type), and the user pays for the time utilized (rounded up).
- Reserved: The user pays a single fee in order to reserve instances for possible use during a time frame (1 year, for instance). When the reserved instance is utilized, the client also pays for the time utilized but with a considerable discount over the on-demand instance of the same type.
- Spot: The user stipulates a maximum value he/she is willing to pay for a certain VM instance type. Instance types are driven by the market based on offer-demand for that type. If the price rises over the stipulated price, the instance is interrupted and the user pays for the time utilized (rounded down).

Considering the VM instance types available for leasing, their prices and pricing models, and the resources available in the private cloud, the hybrid cloud management system is able to compose a hybrid cloud that has sufficient computational capacity to furnish the required QoS. Typically, the private cloud is assumed to be costless in the literature, but in this

---

[2] http://aws.amazon.com/vpc/

paper we adopt the more realistic assumption where a private cloud generates costs to its owner.

IV. OBJECTIVES AND SETUP

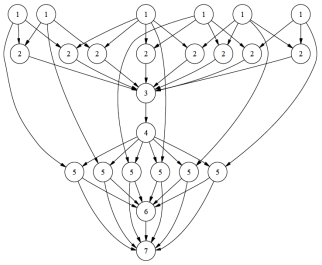
In this paper we are concerned in discussing how a scheduling algorithm behaves when splitting workflows between private and public clouds in the face of different price relations among those resources.

A. Monetary Costs of The Private Cloud

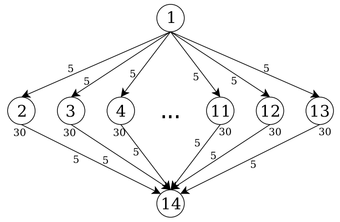
The majority of papers in the literature assume a costless private cloud. However, a monetary cost for maintaining a private cloud in operation can be non-negligible. Even in off-peak demand times, where local resources can be underutilized, or idle times the monetary cost to keep them in operation exists and can not be ignored by the scheduler. The cost of maintaining a private cloud online includes investments in: racks, cables, air-conditioning, room safe, uninterruptible power supply (UPS), power generator, energy consumption, resource redundancy, backups, and the computational resources themselves. In addition to investments in computer equipments, there are investments in human resources, such as administrative and maintenance staff, to maintain local resources online (and perhaps idle) in order to compare whether it is worth to migrate workflow application executions to public cloud. In other words, for the service provider of workflow execution (a privately owned cloud or an SaaS provider, for instance), the total monetary cost is given by the sum of the cost of using the public cloud plus the cost of maintaining the private cloud online and potentially idle. On the other hand, if the public cloud is not considered to be used, application's deadlines can not be met and SLA can be broken, which can result in financial losses. Therefore, there is a clear trade-off between using in-house resources to maximize the utilization of the private cloud (and minimize outsourced resources) and using leased resources to satisfy deadlines (and perhaps let the private cloud idle).

B. Applications

A workflow is generally represented by a Directed Acyclic Graph (DAG) $\mathcal{G} = \{\mathcal{U}, \mathcal{E}\}$, where each node $u_i \in \mathcal{U}$ represents a service and each edge $e_{i,j} \in \mathcal{E}$ represents a data dependency between services i and j, that is, $e_{i,j}$ is the data produced by u_i and consumed by u_j. Figure 1 shows two workflows of real world applications such as Montage and Fork-join. Montage application creates science-grade astronomical image mosaics using data collected from telescopes. Fork-join DAG represents a median filter image processing application, where (i) a large image is *fork* into small images; (ii) each small image is processed separately; and (iii) all processed small images are *joined* to form the original large image. We assume, without loss of generality, that all DAGs have only one entry node (first node) and one exit node (last node). In addition, we assume that each node $u \in \mathcal{U}$ is indivisible and will execute in only one virtualized processing core.



(a) Montage DAG



(b) Fork-join DAG

Fig. 1: DAGs.

C. Scheduler

We have developed an integer linear programming to schedule workflows in clouds [10]. In this work, we consider a scenario where the cloud customer submits his/her scientific workflow to be executed and managed by a workflow execution service (SaaS or PaaS, for instance) along with a response time (deadline) to be obeyed. Due to the NP-Completeness of the scheduling problem, we embedded the λ-*granularity* technique in the ILP scheduler, in order to modify the granularity of the schedule timeline and yield feasible (acceptable) scheduling solutions quickly. In other words, by using different levels of discrete-time intervals in the timeline of the ILP, the input size of the schedule problem can be reduced, and the solver can be able to achieve solutions faster. However, this scheduler was evaluated only in environments composed of virtual machines leased from multiple IaaS providers, and this is the first time we use this ILP scheduler to make an evaluation in a hybrid cloud environment. This scheduler selects the resources in such a way that the monetary cost of scheduling is minimized and the required deadline required is met, i.e., it does not give priority to a specific cloud, whether private or public. Thus, it does not try to maximize the utilization, which is a common behavior in hybrid cloud schedulers [2]. Therefore, in this work we aim to show a problem yet to be tackled where, depending on the deadline required for the workflow execution, the system administrator will need to lease virtual machines from public cloud and he/she will have to bear the monetary costs of both public cloud and an underutilized private cloud.

D. An overview of the Integer Linear Program

The ILP solves the scheduling problem through the following variables binary variables: (i) $x_{u,t,v}$ that assumes the value 1 if the node u finishes at time t in the VM v; otherwise this variable assumes the value 0; and (ii) $y_{t,v}$ that assumes the

TABLE I: Public and Private Clouds

Cloud	Type	Core	Core Performance	On-demand Price $\mathcal{P}$
Private	Small	1	1.0	$\alpha \times \mathcal{P}_{\mathcal{R}_{public}}$
	Medium	2	2.0	$\alpha \times \mathcal{P}_{\mathcal{R}_{public}}$
Public	Small	1	1.0	\$0.06
	Medium	2	2.0	\$0.12
	Large	4	2.0	\$0.24
	XLarge	8	3.5	\$0.50

value 1 if the VM v is being used at time t; otherwise assumes the value 0. The ILP also uses the constant $\mathcal{C}_v$ that assumes the cost per time unit for using the VM v. The ILP objective function has been developed to minimize the overall monetary cost of the workflow execution. In other words, this function minimizes the following sum: $\sum_{t \in \mathcal{T}} \sum_{v \in \mathcal{V}} y_{t,v} \times \mathcal{C}_v$, where $\mathcal{T}$ is the time set that represents the schedule discrete-timeline and $\mathcal{V}$ is the virtual machine set. The objective function is subject to various constraints, which are detailed in [10].

V. EVALUATION

We implemented the presented integer linear program in JAVA and conducted simulations using the IBM ILOG CPLEX Optimizer[3] with default configuration. Our evaluation comprises simulations with DAGs of real world applications such as Montage DAG and 30-nodes fork-join DAG. Simulations were run on an Intel® Core™ Xeon $X5650$ CPU 2.67GHz and 16GB of RAM.

A. Simulation Setup

We have run 30 simulations for each DAG with the public and private cloud configurations shown in Table I. In our hybrid cloud scenario, we assume only the availability of on-demand prices on the public cloud, but other types of pricing could be used, such as reserved and spot. In each simulation, the computation cost for each DAG node and the communication cost for each DAG node dependency were randomly taken from the $[1, 3]$ real interval. Public clouds usually do not provide information about the quality of service for communication among internal nodes or external links. We assume that the bandwidth of links between VMs within the same cloud is larger than the external links (between public and private clouds), which is a reasonable assumption in real environments. This is reflected in our simulation by randomly generating an $\mathcal{L}$ in the $[2, 3]$ interval for links between two different clouds, while for links between VMs inside the same cloud $\mathcal{L}$ is taken from the $[0.1, 0.2]$ interval.

To represent different deadlines, we have run simulations with deadlines $\mathcal{D}_\mathcal{G}$ varying from $T_{max} \times 2/7$ to $T_{max} \times 6/7$ in $1/7$ steps, where T_{max} is the makespan of the cheapest sequential execution of all DAG nodes on a single resource. Deadlines of $T_{max} \times 1/7$ showed only unfeasible solutions for all solver executions, while $\mathcal{D}_\mathcal{G}$ of $T_{max} \times 7/7$ can be achieved by putting all tasks in the cheapest resource. The divisor 7 was chosen only to assess the evolution of the utilization of private cloud increasing $\mathcal{D}_\mathcal{G}$, so other dividers could be used. In order to obtain a feasible solution, the solver had its running time limited to 10 minutes for each simulation. After that time, the solver returned the best solution so far, if any.

B. Results

The averages shown in the following graphs are over 30 DAG-simulations for each deadline $\mathcal{D}_G$. In order to analyze possible prices of private cloud resources, we use the following linear relation:

$$\mathcal{P}_{\mathcal{R}_{private}} = \alpha \times \mathcal{P}_{\mathcal{R}_{public}} \quad (1)$$

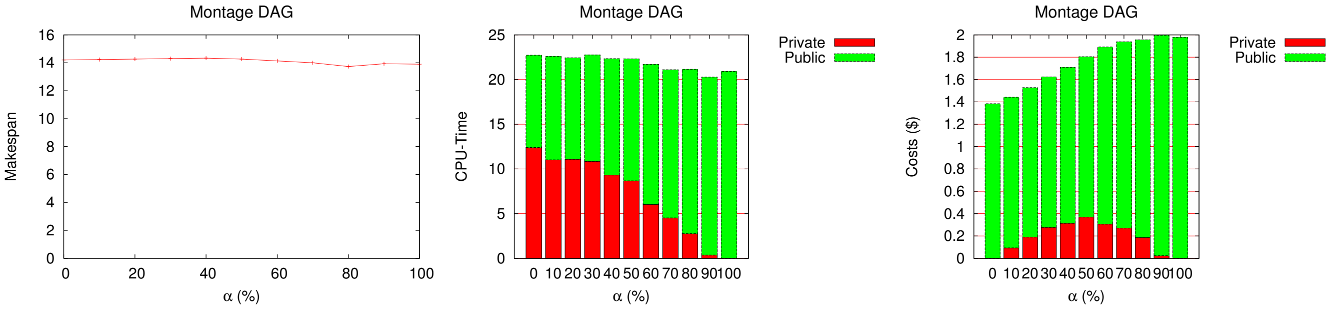
where $\mathcal{R}_{private}$ and $\mathcal{R}_{public}$ are resources of the private and public cloud, respectively, with similar computational power; α is the linear correlation, with $0\% \leq \alpha \leq 200\%$; and finally, $\mathcal{P}_{\mathcal{R}_{private}}$ and $\mathcal{P}_{\mathcal{R}_{public}}$ are prices to use private and public resources per unit of time. For example, $\alpha = 0\%$ means that the private cloud is costless, and $\alpha = 100\%$ means that the price of private resource is equal to the price of outsourced resources. For all simulations with $\alpha > 100\%$, the ILP solver used only public cloud and private cloud remained free of use. This occurred because the goal of the scheduler is to minimize workflow execution cost while satisfying the deadline. Thus, results for $\alpha > 100\%$ are not plotted on the graphs below. It is important to note that we do not show in the results the idleness cost, but only the cost of the workflow execution when it uses those private resources. Therefore, the actual execution cost would depend from the system administrator to establish how much the private cloud infrastructure costs to run, which we consider out of the scope of this paper.

1) Montage DAG: Figures 2, 3 and 4 show results for simulations with Montage DAGs and deadlines equal to $\mathcal{D} = T_{max} \times 2/7$, $\mathcal{D} = T_{max} \times 3/7$ and $\mathcal{D} = T_{max} \times 4/7$, respectively. The DAG Montage simulations were performed as follows: for $\mathcal{D} = T_{max} \times 2/7$ we use $\lambda = 1$; for $\mathcal{D} = T_{max} \times 3/7$ and $\mathcal{D} = T_{max} \times 4/7$ we use $\lambda = 2$; and others deadlines we use $\lambda = 3$. For more details about λ-granularity technique, please, see [10]. Regardless of the deadline $\mathcal{D}_\mathcal{G}$ and the α cost relation, the ILP solver had to use resources outsourced from public cloud because the available local resource from private cloud was not sufficient to meet the execution deadline. Generally, the higher the deadline, the lesser processing was outsourced, thus with a higher utilization of the private cloud.

2) Fork-Join DAG with 30-nodes: Figures 5, 6 and 7 show results for simulations with fork-join DAGs and deadlines equal to $\mathcal{D} = T_{max} \times 2/7$, $\mathcal{D} = T_{max} \times 3/7$ and $\mathcal{D} = T_{max} \times 4/7$, respectively. For the fork-join DAG, no λ-granularity was utilized, thus the ILP was run over the original scheduling problem. Compared to the Montage DAG, we observe a lower utilization by the fork-join DAG with 30 nodes for $\mathcal{D} = T_{max} \times 2/7$, a similar utilization for $\mathcal{D} = T_{max} \times 3/7$, and a higher utilization for $\mathcal{D} = T_{max} \times 4/7$. However, many cases resulted in the underutilization of the private cloud, leading the private cloud owner to bear with

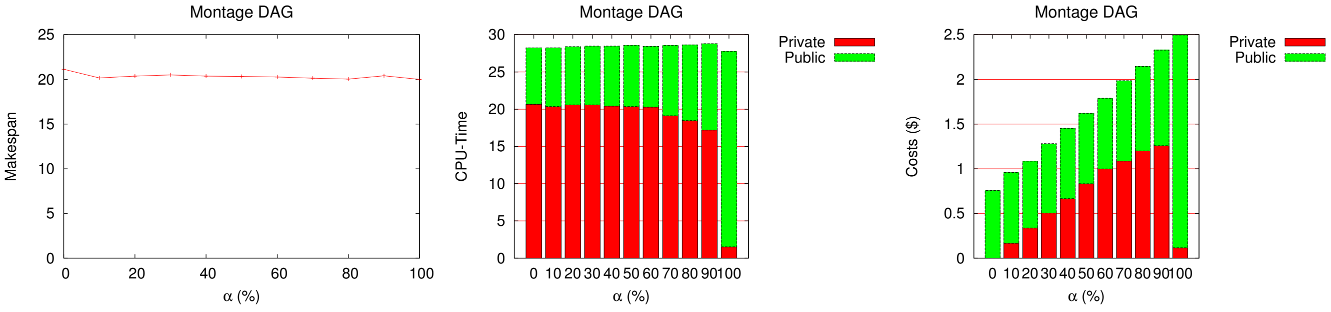
---

[3] http://www.ibm.com/software/integration/optimization/cplex-optimizer/



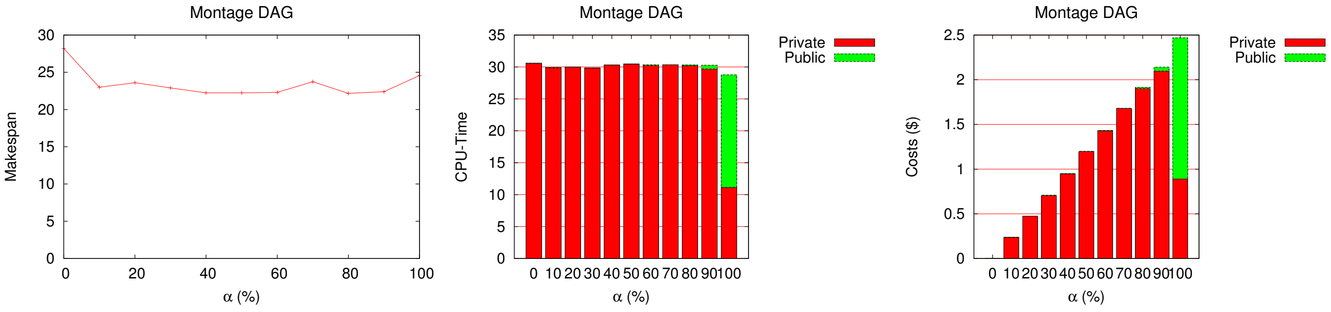
(a) Makespan (b) CPU Time (c) Cost

Fig. 2: Results for Montage with deadline $\mathcal{D} = T_{max} \times 2/7$



(a) Makespan (b) CPU Time (c) Cost

Fig. 3: Results for Montage with deadline $\mathcal{D} = T_{max} \times 3/7$



(a) Makespan (b) CPU Time (c) Cost

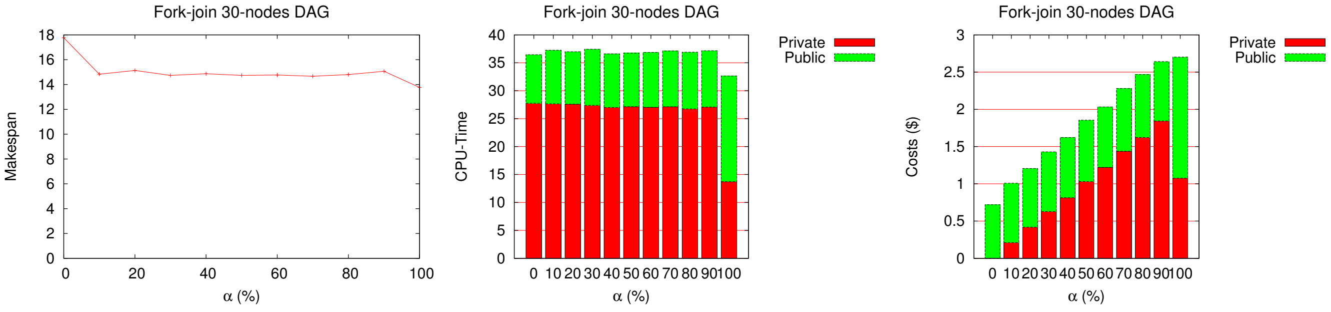
Fig. 4: Results for Montage with deadline $\mathcal{D} = T_{max} \times 4/7$

both the costs of the private and the public clouds, even though computational capacity was yet available in the private cloud. This event occurs mainly for short deadlines, as we can see the results of both DAGs.

3) Discussion: The preliminary results showed that the concern of minimizing costs in hybrid cloud scheduling can lead to underutilization of the private cloud, actually masquerading private cloud costs provenient from its idleness. Neglecting the private cloud utilization when scheduling workflows in hybrid clouds can anticipate the need for public cloud resources, as well as turn the once thought cheaper execution in the public cloud actually more expensive. The development of utilization-aware workflow schedulers are needed to cope with this problem, where cost models for the private cloud are also of paramount importance to be utilized by the scheduler. Another aspect to be considered is to tackle with multiple workflows simultaneously to maximize the private cloud utilization.

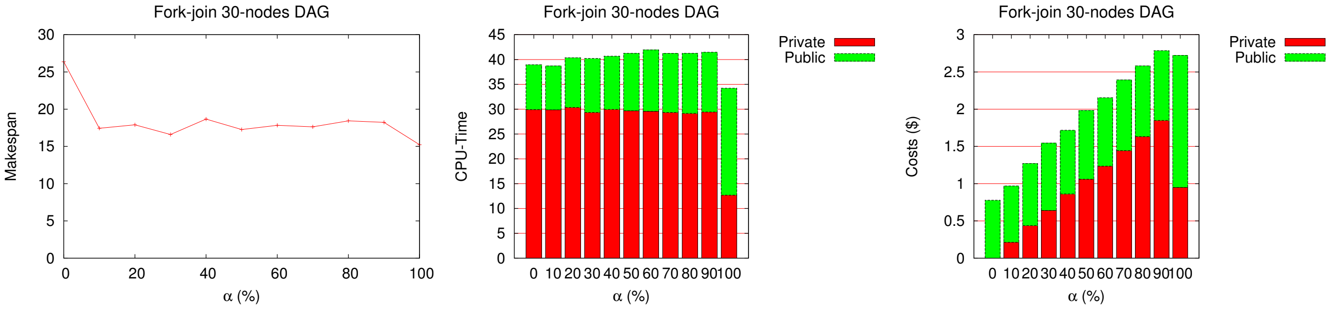
VI. CONCLUSION

In this paper we present a discussion about the behavior of workflow scheduling algorithms when considering a private cloud free of cost, which is not a real scenario. By using outsourced paid resources, the workflow scheduling algorithm can let the private cloud idle to fulfill deadlines, and system administrators must bear the cost of using public cloud and the cost of maintaining the private cloud idle (or underutilized). Therefore, scheduling algorithms must take into account the cost of the private cloud before scheduling workflows in order to (i) minimize the overall monetary cost of the workflow execution; (ii) maximize the use of the private cloud and (iii) meet the workflow's deadlines. In-development works include the development of multiple workflow scheduling to use the private cloud when it is idle. Analysis of monetary costs involved with transferring data between private and public cloud is also considered as future work.



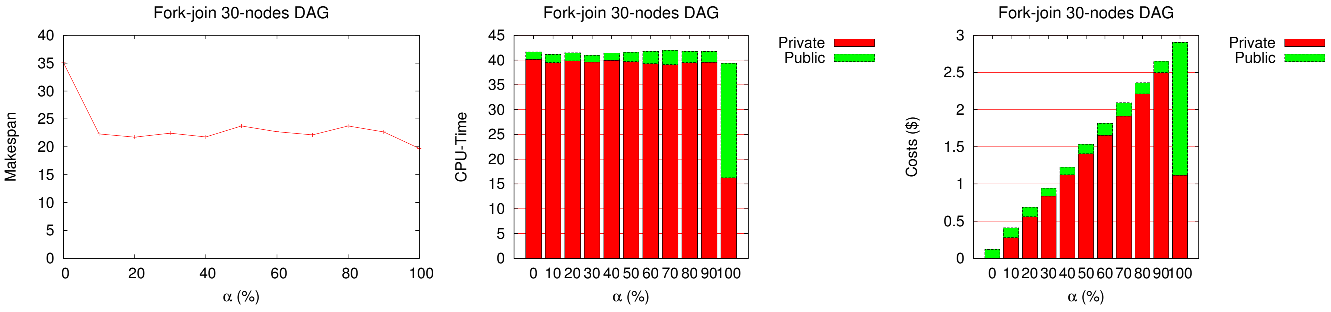
(a) Makespan (b) CPU Time (c) Cost

Fig. 5: Results for fork-join with deadline $\mathcal{D} = T_{max} \times 2/7$



(a) Makespan (b) CPU Time (c) Cost

Fig. 6: Results for fork-join with deadline $\mathcal{D} = T_{max} \times 3/7$



(a) Makespan (b) CPU Time (c) Cost

Fig. 7: Results for fork-join with deadline $\mathcal{D} = T_{max} \times 4/7$

ACKNOWLEDGMENT

The authors would like to thank CNPq and State of São Paulo Research Foundation (FAPESP – grant 2012/02778-6) 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] 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.

[3] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica *et al.*, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[4] M. D. Assunção, A. Costanzo, and R. Buyya, "A cost-benefit analysis of using cloud computing to extend the capacity of clusters," *Cluster Computing*, vol. 13, pp. 335–347, September 2010.

[5] L. F. Bittencourt and E. R. M. Madeira, "HCOC: A cost optimization algorithm for workflow scheduling in hybrid clouds," *Journal of Internet Services and Applications*, vol. 2, no. 3, pp. 207–227, 2011.

[6] M. Shifrin, R. Atar, and I. Cidon, "Optimal scheduling in the hybrid-cloud," in *2013 IFIP/IEEE International Symposium on Integrated Network Management*. IEEE, 2013, pp. 51–59.

[7] C. Vecchiola, R. N. Calheiros, D. Karunamoorthy, and R. Buyya, "Deadline-driven provisioning of resources for scientific applications in hybrid clouds with aneka," *Future Generation Computer Systems*, vol. 28, no. 1, pp. 58–65, January 2012.

[8] E. Deelman, G. Singh, M. Livny, B. Berriman, and J. Good, "The cost of doing science on the cloud: the montage example," in *Proceedings of the 2008 ACM/IEEE conference on Supercomputing*, ser. SC '08. Piscataway, NJ, USA: IEEE Press, 2008, pp. 50:1–50:12.

[9] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud computing and grid computing 360-degree compared," in *Grid Computing Environments Workshop, 2008. GCE '08*, 12-16 2008, pp. 1 –10.

[10] T. A. Genez, L. F. Bittencourt, and E. R. Madeira, "Using time discretization to schedule scientific workflows in multiple cloud providers," in *IEEE 6th International Conference on Cloud Computing*. IEEE, 2013.