Probabilistic Provisioning and Scheduling in Uncertain Cloud Environments

Marco L. Della Vedova, Daniele Tessera
Dept. of Mathematics and Physics
Università Cattolica del Sacro Cuore
Via Musei 41, I-25121 Brescia, Italy
{marco.dellavedova,daniele.tessera}@unicatt.it

Maria Carla Calzarossa
Dept. of Electrical, Computer and Biomedical Eng.
Università degli Studi di Pavia
Via Ferrata 5, I-27100 Pavia, Italy
mcc@unipv.it

***Abstract*—Resource provisioning and task scheduling in Cloud environments are quite challenging because of the fluctuating workload patterns and of the unpredictable behaviors and unstable performance of the infrastructure. It is therefore important to properly master the uncertainties associated with Cloud workloads and infrastructures. In this paper, we propose a probabilistic approach for resource provisioning and task scheduling that allows users to estimate in advance, i.e., offline, the resources to be provisioned, thus reducing the risk and the impact of overprovisioning or underprovisioning. In particular, we formulate an optimization problem whose objective is to identify scheduling plans that minimize the overall monetary cost for leasing Cloud resources subject to some workload constraints. This cost-aware model ensures that the execution time of an application does not exceed with a given probability a specified deadline, even in presence of uncertainties. To evaluate the behavior and sensitivity to uncertainties of the proposed approach, we simulate a batch workload consisting of MapReduce jobs. The experimental results show that, despite the provisioning and scheduling approaches that do not take into account the uncertainties in their decision process, our probabilistic approach nicely adapts to workload and Cloud uncertainties.**

Keywords: Cloud computing; Resource provisioning; Task scheduling; Uncertainty; Probabilistic approach; MapReduce workload; CloudSim

I. Introduction

Cloud computing with its promise of lower cost and better efficiency – exploited through features, such as on demand, pay-per-use, elasticity – opens challenging issues related to resource management and provisioning. Cloud providers are interested in maximizing their profit, often achieved by consolidation policies that maximize their resource utilization. Conversely, Cloud users are primarily interested in identifying the most cost-effective infrastructure for successfully deploying their applications in Cloud or multi-cloud environments.

In these complex scenarios, the estimation of the resources actually needed by a given application is quite difficult because of the presence of fluctuating workload patterns as well as of heterogeneous virtualized Cloud infrastructures characterized by some unpredictable behaviors and unstable performance [1]. For example, multi-tenant resource sharing, failures and Virtual Machine (VM) migration and consolidation could lead to the creation of sudden performance bottlenecks that, in turn, are often responsible of significant performance degradation. It is therefore important to properly master the "uncertainties" associated with Cloud workloads and infrastructures.

This paper addresses these issues by proposing a probabilistic approach for resource provisioning and task scheduling. Our objective is to devise a cost-aware model that identifies for a given application a cost-effective setting, i.e., amount and types of Cloud resources and corresponding scheduling plan. The model will ensure that the execution time of the application does not exceed with a given probability a specified value, i.e., deadline, even in presence of variability and uncertainties.

More specifically, the model is formulated as an optimization problem whose objective is to minimize the overall monetary cost for leasing Cloud resources subject to constraints related, for example, to application structure and execution time.

This model will allow users to estimate in advance, i.e., offline, the resources to be provisioned, thus reducing the risk and the impact of overprovisioning or underprovisioning. In fact, overprovisioning results in unnecessary costs due to unused resources. Similarly, an increased cost is also the result of insufficient resource provision since on-demand Cloud instances are in general more expensive than their reserved counterparts (up to 75% for Amazon EC2). The evaluation of the proposed approach is performed on a case study based on a MapReduce workload.

The main contributions of this paper can be summarized as follows:

- Probabilistic approach for resource provisioning and task scheduling.
- Formulation of an optimization problem that adapts to workload and Cloud uncertainties.
- Extensions of the CloudSim toolkit to simulate uncertainties.

The paper is organized as follows. Section II presents the state of the art in the framework of provisioning and scheduling in Cloud environments. The cost-aware model is described in Section III, while its application to MapReduce workloads is presented in Section IV. The evaluation of the proposed model is discussed in Section V. Finally, Section VI presents some concluding remarks.

II. RELATED WORK

Cloud resource provisioning and scheduling have been addressed in the literature under different perspectives by considering different workload types. In particular, MapReduce workloads have been the target of several studies. In [2] the resource allocation is formulated as an optimization problem based on a cost model that takes into account the relationships among amount of input data, available system resources and complexity of the application components. Two VM provisioning approaches aimed at minimizing the cost for running MapReduce applications under deadline constraints are proposed in [3]. Resource provisioning and task scheduling in heterogeneous Cloud environments are addressed in [4] in the framework of big data analytics. The proposed algorithms aim at minimizing the cost as a function of the budget and deadline constraints associated with MapReduce applications. Resource provisioning and scheduling for batch workloads with hard deadlines are addressed in [5] as an optimization problem with a linear programming formulation. The paper shows that cost-optimization techniques are particularly suitable for multi-provider hybrid Cloud settings.

A recent study [6] offers a comprehensive survey of workflow scheduling in Cloud environments in the framework of a taxonomy based on the knowledge of workflow properties (e.g., structure, execution time) and resource characteristics.

Even though many studies acknowledge the variability and performance instability affecting Cloud environments (see [7] for a classification of these uncertainties), very few studies explicitly consider these effects in the formulation of the provisioning and scheduling problems. The combined provisioning and scheduling strategy for scientific workflows presented in [8] models Cloud performance and data transfer variations by adjusting processing capacity and bandwidth according to a degradation percentage reported in [9]. The uncertainty associated with resource demand and pricing in multi-cloud environments is taken into account in the resource provisioning algorithm proposed by Chaisiri et al. [10]. The algorithm – formulated as a stochastic programming problem – minimizes the provisioning costs by adjusting the trade-off between resource reservation and on-demand allocation.

Despite previous studies, the approach proposed in this paper models the variability of the workload characteristics as well as the instability of Cloud performance in terms of probabilistic distributions. Therefore, these uncertainties are considered as an integral part of the decisions about provisioning and scheduling.

III. COST-AWARE MODEL

As already pointed out, the estimation of the resources needed by the applications deployed in the Clouds has to take into account the performance instability that might affect the infrastructure as well as the variability in the workload behavior. The cost-aware model presented in this paper tries to cope with these issues by adopting a probabilistic approach to the provisioning and scheduling problem. The goal of the model is to minimize the deployment cost and satisfy at the same time possible workload constraints. In what follows, we define in detail the formulation of this optimization problem and present the methods devised for obtaining an efficient solution.

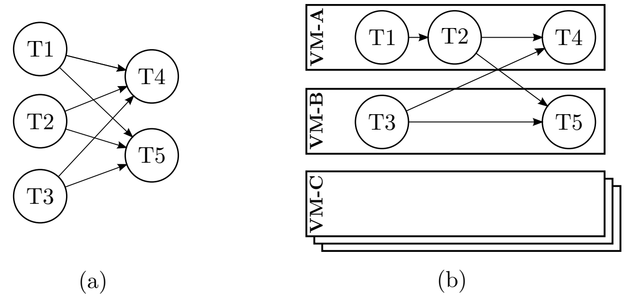


Fig. 1. Representation of a job consisting of tasks with precedence constraints (a) and mapping of the tasks on two VMs (b).

A. Problem definition

Given a workload (i.e., jobs consisting of tasks with precedence constraints, as depicted in Figure 1 (a)) to be deployed on a Cloud (or multi-cloud) infrastructure (i.e., multiple instances of different VM types interconnected together), our goal is to provide Cloud users with a model that guides them in their decisions on provisioning and scheduling, namely:

- Selection of the amount and types of resources (e.g., VMs) to be allocated to the jobs (*provisioning* phase);
- Mapping between the individual tasks and the selected VMs (*scheduling* phase).

In this scenario, the decision process is particularly challenging because of the potentially large number of choices. Hence, we address this process as an optimization problem whose objective is to minimize the total cost for leasing Cloud resources and satisfy the probabilistic constraint on the deadline associated with the job execution time. The formulation of the problem is as follows:

$$\begin{aligned} \text{minimize} \quad & \mathbb{E}\,[\text{Cost}] \\ \text{subject to} \quad & \Pr(\mathcal{T} \leq d) \geq (1-p) \\ & \text{task precedence constraints} \end{aligned} \tag{1}$$

where $\mathbb{E}\,[\text{Cost}]$ denotes the expected total cost for leasing Cloud resources, d and p denote the deadline associated with the job execution time $\mathcal{T}$ and the probability of deadline violation, respectively.

The solution space of this combinatorial problem grows exponentially with the number of tasks the job consists of and the number and types of VMs the Cloud infrastructure consists of. In particular, the expected cost depends on the pricing of the Cloud resources as well as on job execution time $\mathcal{T}$ and the overall resource usage. The execution time in turn depends on the job and Cloud characteristics and on the identified resource provisioning and task scheduling plan.

Therefore, it is necessary to derive an accurate estimation of the execution time prior the job execution, that is, offline.

This estimation has to take into account the variability and uncertainties in the job characteristics and Cloud performance. Hence, we describe the job and Cloud attributes in terms of random variables with their probability distribution functions. In particular, we assume that these distributions are obtained by previous characterization studies of the workloads and of the infrastructure.

Since we model the task execution times and the network bandwidths as random variables, each described by its own probability distribution, we derive the probability distribution of the job execution time as a composition of these random variables. Therefore, the probabilistic formulation of the optimization problem requires the knowledge of the distributions of these random variables.

As an illustrative example, we consider the job depicted in Figure 1 (a) whose five tasks have been allocated to two of the three available VMs, that is, three tasks (i.e., T1, T2 and T4) to one VM and two tasks (i.e., T3 and T5) to another VM (see Figure 1 (b)). To evaluate this scheduling plan, we derive the random variable $\mathcal{T}$ describing the job execution time as:

$$\mathcal{T} = \max\{T_1 + T_2, T_3\} + \max\{T_4, T_5\}$$

where T_i's denote the random variables referring to the execution times of tasks i.

B. Provisioning and scheduling

To identify the cheapest scheduling plan that satisfies the given deadline constraint, a straightforward approach consists of evaluating all possible mappings between tasks and VMs. However, this approach becomes unfeasible for jobs with large number of tasks especially whenever they are deployed in multi-cloud environments. Therefore, as we will discuss in Section IV, heuristic approaches are often applied.

C. Probabilistic evaluation

In order to solve the optimization problem formulated in (1) and evaluate whether the scheduling plan satisfies the constraint, we need to derive the cumulative distribution function associated with $\mathcal{T}$. This function is the result of the combination of the probability distributions of the random variables describing the task execution times. The combination of these distributions is seldom analytically tractable. Even for simple distributions, the analytic solution might be very complex. For example, the cumulative distribution of the execution time of two sequential tasks allocated on one VM is obtained as the sum of the random variables describing the execution times of the individual tasks, namely, T_1 and T_2. The sum of these two random variables is computed as:

$$\begin{aligned} F_{T_1+T_2}(t) &= \Pr(T_1 + T_2 \le t) \\ &= \int_{-\infty}^{+\infty} F_{T_1}(x) f_{T_2}(t-x)dx \\ &= (F_{T_1} * f_{T_2})(t) \end{aligned}$$

where F_{T_1} and f_{T_2} denote the cumulative distribution function and probability density function of T_1 and T_2, respectively, while $*$ denotes the convolution product. For deriving numerical solutions we resort to statistical computing techniques based on spectral methods. This approach can be applied to random variables described by either discrete or continuous probability density functions.

TABLE I
MAPREDUCE JOB MODEL NOTATIONS

Job model	
Description	**Notation**
Number of map tasks	m
Number of reduce tasks	r
Map task i = 1, …, m	
Input data size (from the data source)	δ_i
Number of instructions	L_i
Intermediate data size to reduce task j	$\gamma_{i,j}$
Reduce task i = m+1, …, m+r	
Number of instructions	L_i

IV. CASE STUDY

This section presents the case study implemented to evaluate the proposed probabilistic approach to resource provisioning and task scheduling. In particular, we focus on a batch workload consisting of MapReduce jobs [11], each characterized by m map tasks and r reduce tasks whose precedence constraints are such that no reduce task can start before all map tasks have completed their execution.

A. MapReduce model

In our case study we model the entire execution of a MapReduce job as consisting of two phases: a *map phase* and a *reduce phase*. Note that the intermediate data shuffle is considered as part of the map phase because a completed map task can start transferring its intermediate data to reduce tasks even before the completion of all the other map tasks.

In what follows, we assume a map task $i, i = 1, 2, ..., m$, defined in terms of its input data size δ_i, length L_i (i.e., number of instructions to be executed), and intermediate data size $\gamma_{i,j}$ to be transferred to each reduce task $j, j = m + 1, ..., m + r$. A reduce task j is defined in terms of its length L_j. Table I summarizes the notations used throughout the rest of the paper.

B. Cloud model

In our case study, we consider MapReduce jobs deployed in a multi-cloud environment consisting multiple resources (e.g., data sources, VMs, networks) with different performance and pricing models.

In detail, each VM type – usually consisting of multiple instances – is described by its speed s_{VM} and leasing cost $c^{\mathrm{exe}}_{\mathrm{VM}}$. Moreover, the characteristics of the networks between VMs are specified in terms of their bandwidth $B_{\mathrm{VM-VM}}$ and transfer cost $c^{\mathrm{net}}_{\mathrm{VM}}$. Similarly, the networks between data sources and VMs are described in terms of bandwidth $B_{\mathrm{DS-VM}}$ and the transfer cost c_{DS}. Note that this simplified model captures the most relevant characteristics of the Cloud infrastructures and as such it allows the assessment of the proposed probabilistic approach.

C. Variability

As already pointed out, variability and uncertainties are due to workload and Cloud infrastructure behaviors. In our study we choose to model the task length, i.e., a workload characteristic, and the communication bandwidth, i.e., a characteristic of the Cloud infrastructure, as random variables. We assume these random variables to be independent. In what follows, uppercase letters refer to random variables.

D. Execution time and cost evaluation

Once defined the MapReduce and Cloud models and given a scheduling plan, it is possible to derive the random variables associated with the job execution time and the corresponding monetary cost. As already discussed, these evaluations depend on the mapping between the tasks and the VMs which in turn determine the execution times of the individual tasks.

1) Task execution time: To evaluate the execution time of a task, we need to consider whether it is a map or a reduce task. In particular, the execution time of a map task includes the processing time – determined by the task length and the speed of the allocated VM – as well as the data transfers from the datasource and to the reduce tasks. On the contrary, the execution time of a reduce task only includes its processing. In details, the execution time T_i of task i is obtained as follows:

$$T_i = \begin{cases} \underbrace{\frac{\delta_i}{B_{\mathrm{DS-VM}i}}}_{\text{data input}} + \underbrace{\frac{L_i}{s_{\mathrm{VM}i}}}_{\text{processing}} + \underbrace{\sum_{j=m+1}^{m+r} \frac{\gamma_{i,j}}{B_{\mathrm{VM}i-\mathrm{VM}j}}}_{\text{data shuffle}} & i \leq m \\ \underbrace{\frac{L_i}{s_{\mathrm{VM}_i}}}_{\text{processing}} & i > m \end{cases}$$

where $\mathrm{VM}i$ denotes the VM instance allocated to task i.

We recall that L_i, $B_{\mathrm{DS-VM}i}$ and $B_{\mathrm{VM}i-\mathrm{VM}j}$ are random variables. Therefore, T_i's are themselves random variables.

2) Job execution time: Predicting the job execution time $\mathcal{T}$ is fundamental for evaluating the probability of deadline violation, which is a constraint of our optimization problem. However, the offline evaluation of the job execution time is not straightforward because it relies on the combination of several probability distributions as discussed in Section III-C. In particular, it is necessary to perform some algebraic operations on the random variables describing the task execution times, e.g., sum of the times spent in the map and reduce phases. In detail, these values are obtained by computing the maximum times spent in each phase across the allocated VM instances. In addition, whenever multiple tasks are executed sequentially on a single VM, it is necessary to take the sum of the execution times of the individual task.

3) Job execution cost: The objective of the optimization problem (1) is to minimize the expected job execution cost. The cost for deploying a MapReduce job on a Cloud infrastructure is determined by the execution times of the individual tasks as well as by the data transfer and the Cloud pricing models, namely:

$$\mathrm{Cost} = \underbrace{\sum_{i=1}^{m+r} T_i \cdot c_{VMi}^{\mathrm{exe}}}_{\text{VM leasing}} + \underbrace{c_{DS} \cdot \sum_{i=1}^{m} \delta_i}_{\text{data transfer DS-VM}} + \underbrace{\sum_{i=1}^{m} \sum_{j=m+1}^{m+r} c_{VMi}^{\mathrm{net}} \cdot \gamma_{i,j}}_{\text{data transfer VM-VM}}$$

E. Provisioning and scheduling algorithms

To solve the optimization problem, we consider two minimization algorithms: a pruned-tree version of the *Branch-and-Bound* (BB) algorithm [4] and an extension of the *Deadline-aware Tasks Packing* (DTP) heuristic [3]. The BB algorithm is an exact method, therefore, it is only feasible for "small" problems in terms of number of tasks and VM types.

On the contrary, the DTP heuristic takes advantage of the precedence constraints of MapReduce jobs when devising the scheduling plans. In details, this heuristic splits the job deadline into map phase and reduce phase deadlines whose lengths are proportional to the lengths (i.e., number of instructions) of the longest map and the longest reduce tasks. The algorithm considers the VMs starting from the cheapest to the most expensive and schedules on a given VM the maximum number of sequential tasks that satisfies the probabilistic constraint on the corresponding phase deadline. This process is iterated across all VMs until all tasks have been scheduled. The first scheduling plan that satisfies the overall job deadline is then selected as the solution. Note that while the DTP algorithm can be successfully applied to large MapReduce jobs, it does not guarantee to identify the optimal solution, i.e. the cheapest scheduling plan.

V. Experimental results

To evaluate the probabilistic approach proposed in this paper, we focus on the case study described in Section IV. In particular, after identifying an optimal scheduling plan for the given workload and Cloud infrastructure, we run several simulation experiments aimed at assessing the behavior of the identified plan in terms of two metrics, namely, the number of deadline violations and the average job execution cost.

A. Experimental setup

The case study relies on the MapReduce and Cloud models presented in Section IV. In a first set of experiments, we consider jobs consisting of five map tasks and two reduce tasks. The characteristics of these tasks are as follows. Each map task receives 1 GB of data from an external datasource and transfers 50 MB and 100 MB intermediate results to the two reduce tasks. In addition, the length of a map task is 270,000 millions of instructions (MI), whereas the lengths of the reduce tasks are equal to 40,000 MI and 27,000 MI, respectively. Each job is also characterized by a deadline. In our experiments, we consider two different deadlines, equal to

TABLE II
MAIN CHARACTERISTICS OF THE CLOUD INFRASTRUCTURE CONSIDERED IN THE EXPERIMENTS.

	VM Type	Cost [$/h]	Speed [MIPS]
Cloud A	1	0.84	800
	2	0.10	500
	3	0.07	400
Cloud B	1	1.70	1,200
	2	2.50	1,500

TABLE III
MAIN CHARACTERISTICS OF THE NETWORKS CONNECTING THE DATASOURCE TO THE VMS AND BETWEEN VMS.

	Bandwidth	Cost
VM-VM	200Mbit/s	0.010¢/GB
DS-VM	100Mbit/s	0.005¢/GB

800 and 1,400 seconds, respectively. Note that the deadlines are chosen to lie between slowest and fastest scheduling plan. The probability of deadline violation is set to p=0.05.

Table II summarizes the main characteristics of the VMs considered in the experiments. Although the pricing model is expressed on a per hour basis, we assume a per minute billing policy.

The main characteristics of the networks connecting the datasource to the VMs and of the network between VMs are presented in Table III.

To take into account the variability and uncertainties in the task length as well as in the communication bandwidth between datasource and VMs and between VMs, we describe each of these attributes in terms of their Lognormal distribution. We recall that the corresponding probability density function is given by:

$$f(x) = \frac{1}{\sqrt{2\pi}\mu x} e^{-\frac{(\ln(x/\mu))^2}{2\sigma^2}} \qquad \mu, \sigma > 0$$

where μ and σ denote the scale and shape parameters, respectively.

B. Simulation environment

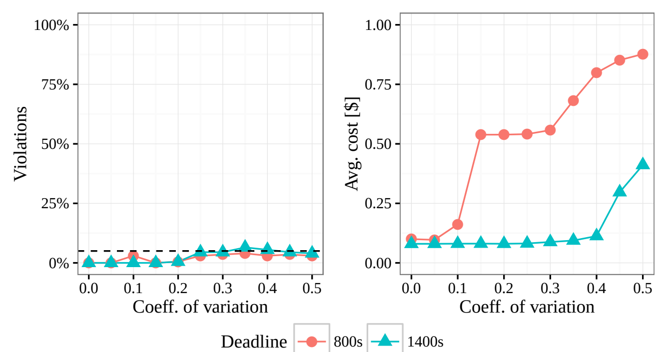
Our experiments rely on the CloudSim toolkit, a framework for modeling and simulation of Cloud computing infrastructure and services [12]. Since in our approach the characteristics of the workload and of the Cloud infrastructure are described in terms of random variables, we extend the toolkit to accommodate these requirements. In detail, we integrate the SSJ library [13] to generate random variables according to the given probability distribution. In addition, to efficiently solve the optimization problem, we integrated CloudSim with the R Project for Statistical Computing[1]. In particular, we exploit the `distr` package [14] for the computation of the algebraic operations required to evaluate the probabilistic constraints associated with the workload. Moreover, for each simulation experiment, we perform 200 independent replications.

[1]https://www.R-project.org/

C. Impact of the variability

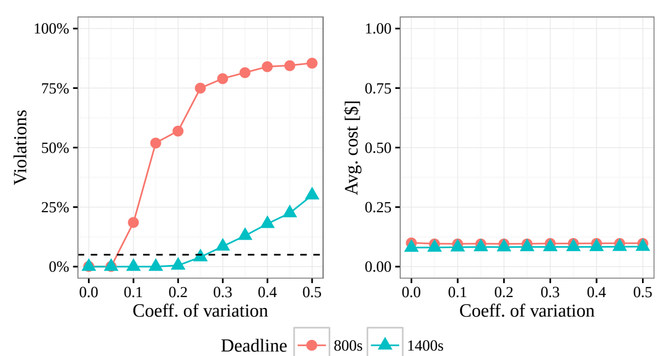
The first set of experiments focuses on the impact of the variability of the task lengths and network bandwidths on provisioning and scheduling. As already pointed out, we describe these attributes by means of Lognormal distributions. Moreover, we vary the coefficient of variation (CV) of each distribution from 0 up to 0.5. More precisely, we apply the Branch-and-Bound algorithm to identify the optimal scheduling plan corresponding to each value of the CV and then we simulate the various plans. In what follows, we present the results of these simulation experiments.

Figure 2 shows the percentage of deadline violations and average job execution cost as a function of the CV of the distributions. As expected, the fraction of violations is mostly under the given 5% threshold (represented by the dashed line, in Figure 2 (a)). We can also observe (see Figure 2 (b)) that the costs of the scheduling plans tend to increase with the variability especially in the case of the tighter deadline. This is because the identified scheduling plans require faster and more expensive resources to satisfy the deadline constraint when the variability increases.



(a) (b)

Fig. 2. Percentage of deadline violations (a) and cost (b) of the probabilistic scheduling plans as a function of the coefficients of variation of the Lognormal distributions.



(a) (b)

Fig. 3. Percentage of deadline violations (a) and cost (b) of the deterministic scheduling plans as a function of the coefficients of variation of the Lognormal distributions.

To underscore the role played by the job and Cloud variability in the identification of the scheduling plans, we identify plans – by means of the BB algorithm – that only take into account the expected values of task lengths and network bandwidths. We then compare these optimal plans with the optimal plans identified by the probabilistic approach. Let us recall that in the simulation of these deterministic scheduling plans we take into account the actual job and Cloud variability. As shown in Figure 3 (a), the percentage of deadline violations increases with the variability up to 85% for the tighter deadline, i.e., 800 seconds. On the other hand, the corresponding cost is not significantly affected by the variability (Figure 3 (b)).

As a general result, we can observe that shorter deadlines result in more expensive probabilistic scheduling plans or in deterministic plans with increased number of deadline violations.

To further assess the role of uncertainties and variability, we analyze additional deterministic scheduling plans identified by considering some sort of "over-estimations" of the workload and Cloud attributes. In particular, we identify a first set of scheduling plans (hereafter denoted as "Over-Est1") where the decisions are based on the expected values of these attributes inflated by 20%. In a second set of scheduling plans (hereafter denoted as "Over-Est2"), we assume to some additional knowledge on the variability; more precisely, both the expected values and the standard deviations of the workload and Cloud attributes are known. Therefore, the identification relies on the expected values plus the values of their corresponding standard deviations. As Figure 4 shows, the additional knowledge of the variability does not provide any substantial benefit in terms of number of deadline violations and cost. The observed behavior is quite unpredictable and the deadline is often violated. Note

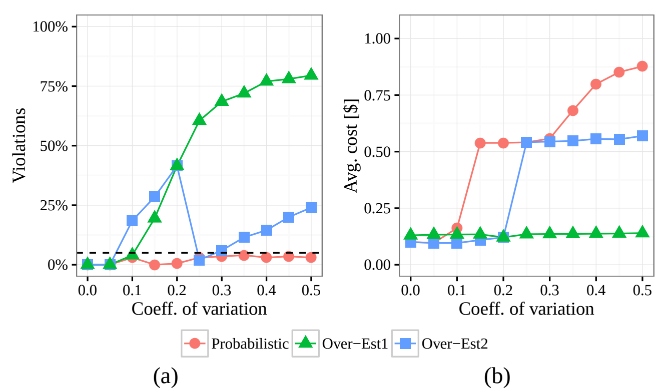


Fig. 4. Percentage of deadline violations (a) and cost (b) as for the two over-estimations.

that these "over-estimations" do not result in overprovisioning of Cloud resources since they are only considered for the selection the number and types of VMs to be allocated.

To further analyze the impact of the variability on provisioning and scheduling, we apply the DTP heuristic to identify the scheduling plans and we compare these plans with the optimal plans identified by the Branch-and-Bound algorithm. (Figure 5). As can be seen, both algorithms satisfy the deadline constraint, although the DTP provides slightly more expensive scheduling plans because it does not fully explore the entire solution space. However, as already pointed out, the applicability of Branch-and-Bound algorithm is limited because of the combinatorial nature of solution space.

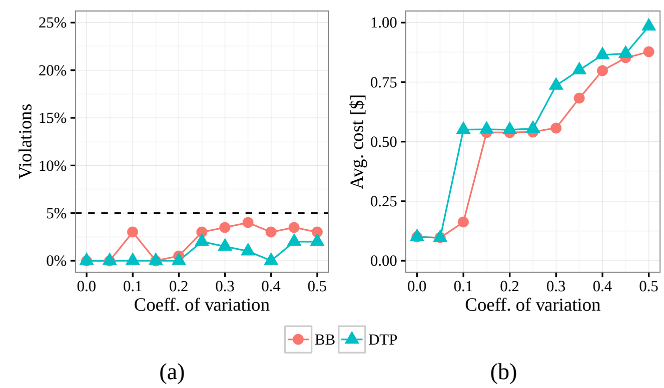


Fig. 5. Deadline violations (a) and cost (b) of the scheduling plans identified by DTP heuristic and by the Branch-and-Bound algorithm as a function of the coefficients of variation of the Lognormal distributions.

D. *Impact of job size*

The second set of experiments focuses on analyzing the behavior of our probabilistic approach with respect to the job size. Hence, we focus on jobs consisting of $m = 5 \cdot 2^{n-1}$ map and $r = 2^n$ reduce tasks, with n ranging from one up to six. We consider a deadline of 5,000 seconds to allow for sequential execution of multiple tasks on the individidual VMs. Moreover, we set the coefficients of variation of the Lognormal distributions to 0.25. Figure 6 shows a comparison, as a function of the job size, of various DTP based approaches. In detail, the figure plots the percentage of deadline violations and the cost of the scheduling plans identified by the probabilistic approach and by three "over-estimations" based on:

1) Expected values of the workload and Cloud attributes (Over-Est0).
2) Expected values of the workload and Cloud attributes inflated by.50% (Over-Est1);
3) Expected values of the workload and Cloud attributes plus half of their standard deviations (Over-Est2).

As expected, the probabilistic approach satisfies the deadline even in the case of jobs with a large number of tasks. Conversely, as shown in Figure 6 (a), for the scheduling plans corresponding to the three over-estimation approaches the number of violations increases with the job size and exceeds the constraint. The Over-Est0 and Over-Est2 scheduling plans begin violating the 5% constraint when the number of map tasks is equal to 20 and 40, respectively. On the other hand, Over-Est1 performs better and exceeds the 5% limit only for jobs consisting of 160 map tasks.

Interestingly, all approaches provide scheduling plans with very similar costs, as depicted in Figure 6 (b). This is due to

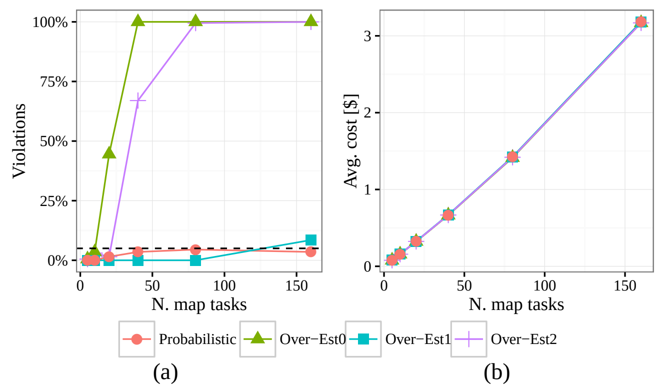


Fig. 6. Comparison of probabilistic approach with various over-estimations as a function of the job size.

the fact that the DTP heuristic always starts selecting instances of the cheapest VM type regardless the estimation approach. The various scheduling plans differ only in the number of provisioned VMs, hence in their degree of parallelism, which depends on the approach used by the heuristic. Therefore, since tasks are allocated to VMs with the same characteristics, the execution time of all the tasks – and the corresponding cost – does not significantly vary.

In summary, the application of the DTP heuristic to the probabilistic approach identifies scheduling plans, which cope with the constraint on the deadline violation probability and are as expensive as the solutions based on over-estimations.

VI. Conclusions

Resource provisioning and task scheduling in Cloud environments face multiple challenges. On the one hand, they need to cope with the fluctuating workload patterns and the unpredictable behaviors and unstable performance of the infrastructure. On the other hand, they have to meet user expectations and identify the most cost-effective infrastructure for successfully deploying the applications. In this paper, we addressed the resource provisioning and task scheduling from a probabilistic perspective that takes into account the variability and uncertainties typical of Cloud environments. We formulated an optimization problem whose objective is to identify scheduling plans that minimize the overall monetary cost for leasing Cloud resources subject to some workload constraints. This cost-aware model ensures to meet with a given probability the deadline associated with the application. In addition, the model provides users with offline predictions of the resources to be provisioned, thus reducing the risk and the impact of overprovisioning or underprovisioning. A case study based on a MapReduce workload was implemented to evaluate the proposed approach. The simulation results show the importance of taking into account the workload and Cloud variability in the provisioning and scheduling strategies. Moreover, our model is robust in that the identified scheduling plans are able to adapt, with the specified probability, to the variability and at the same time to cope with the deadline constraint.

As a future work, we plan to extend the probabilistic provisioning and scheduling to different types of workloads, such as, bag of tasks and interactive applications. In addition, we plan to investigate novel heuristics targeted to the workload characteristics.

References

[1] M. Calzarossa, M. L. Della Vedova, L. Massari, D. Petcu, M. I. M. Tabash, and D. Tessera, "Workloads in the Clouds," in *Principles of Performance and Reliability Modeling and Evaluation*, ser. Springer Series in Reliability Engineering, L. Fiondella and A. Puliafito, Eds. Springer, 2016.

[2] F. Tian and K. Chen, "Towards Optimal Resource Provisioning for Running MapReduce Programs in Public Clouds," in *Proc. 2011 IEEE Int. Conf. on Cloud Computing (CLOUD)*. IEEE, 2011, pp. 155–162.

[3] E. Hwang and K. Kim, "Minimizing Cost of Virtual Machines for Deadline-Constrained MapReduce Applications in the Cloud," in *Proc. of the 2012 ACM/IEEE 13th Int. Conf. on Grid Computing (GRID'12)*. IEEE Computer Society, 2012, pp. 130–138.

[4] M. Alrokayan, A. Vahid Dastjerdi, and R. Buyya, "SLA-Aware Provisioning and Scheduling of Cloud Resources for Big Data Analytics," in *Proc. 2014 IEEE Int. Conf. on Cloud Computing in Emerging Markets (CCEM)*, 2014, pp. 1–8.

[5] R. Van den Bossche, K. Vanmechelen, and J. Broeckhove, "Cost-Optimal Scheduling in Hybrid IaaS Clouds for Deadline Constrained Workloads," in *Proc. 2010 IEEE Int. Conf. on Cloud Computing (CLOUD)*. IEEE, 2010, pp. 228–235.

[6] F. Wu, Q. Wu, and Y. Tan, "Workflow scheduling in cloud: a survey," *The Journal of Supercomputing*, vol. 71, no. 9, pp. 3373–3418, 2015.

[7] A. Tchernykh, U. Schwiegelsohn, V. Alexandrov, and E. Talbi, "Towards Understanding Uncertainty in Cloud Computing Resource Provisioning," *Procedia Computer Science*, vol. 51, pp. 1772–1781, 2015.

[8] M. Rodriguez and R. Buyya, "Deadline Based Resource Provisioning and Scheduling Algorithm for Scientific Workflows on Clouds," *IEEE Transactions on Cloud Computing*, vol. 2, no. 2, pp. 222–235, 2014.

[9] J. Schad, J. Dittrich, and J.-A. Quiané-Ruiz, "Runtime Measurements in the Cloud: Observing, Analyzing, and Reducing Variance," *Proc. VLDB Endow.*, vol. 3, no. 1-2, pp. 460–471, 2010.

[10] S. Chaisiri, B.-S. Lee, and D. Niyato, "Optimization of Resource Provisioning Cost in Cloud Computing," *IEEE Transactions on Services Computing*, vol. 5, no. 2, pp. 164–177, 2012.

[11] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[12] R. Calheiros, R. Ranjan, A. Beloglazov, C. De Rose, and R. Buyya, "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms," *Software Practice & Experience*, vol. 41, no. 1, pp. 23–50, 2011.

[13] P. L'Ecuyer, L. Meliani, and J. Vaucher, "SSJ: A Framework for Stochastic Simulation in Java," in *Proceedings of the 2002 Winter Simulation Conference*, E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, Eds. IEEE Press, 2002, pp. 234–242.

[14] P. Ruckdeschel and M. Kohl, "General Purpose Convolution Algorithm in S4 Classes by Means of FFT," *Journal of Statistical Software*, vol. 59, no. 4, pp. 1–25, 2014.