A methodological framework for cloud resource provisioning and scheduling of data parallel applications under uncertainty

Maria Carla Calzarossa

Dipartimento di Ingegneria Industriale e dell'Informazione
Università di Pavia
via Ferrata 5 – I-27100 Pavia, Italy
mcc@unipv.it

Marco L. Della Vedova

Dipartimento di Matematica e Fisica
Università Cattolica del Sacro Cuore
via Musei 41 – I-25121 Brescia, Italy
marco.dellavedova@unicatt.it

Daniele Tessera

Dipartimento di Matematica e Fisica
Università Cattolica del Sacro Cuore
via Musei 41 – I-25121 Brescia, Italy
daniele.tessera@unicatt.it

Abstract

Data parallel applications are being extensively deployed in cloud environments because of the possibility of dynamically provisioning storage and computation resources. To identify cost-effective solutions that satisfy the desired service levels, resource provisioning and scheduling play a critical role. Nevertheless, the unpredictable behavior of cloud performance makes the estimation of the resources actually needed quite complex. In this paper we propose a provisioning and scheduling framework that explicitly tackles uncertainties and performance variability of the cloud infrastructure and of the workload. This framework allows cloud users to estimate in advance, i.e., prior to the actual execution of the applications, the resource settings that cope with uncertainty. We formulate an optimization problem where the characteristics not perfectly known or affected by uncertain phenomena are represented as random variables modeled by the corresponding probability distributions. Provisioning and scheduling decisions – while optimizing various metrics, such as monetary leasing costs of cloud resources and application execution time – take fully account of uncertainties encountered in cloud environments. To test our framework, we consider data parallel applications characterized by a deadline constraint and we investigate the impact of their characteristics and of the variability of the cloud infrastructure. The experiments show that the resource provisioning and scheduling plans identified by our approach nicely cope with uncertainties and ensure that the application deadline is satisfied.

Keywords: Cloud computing; Resource provisioning; Scheduling; Data parallel workload; CloudSim; Genetic Algorithm.

1. Introduction

The analysis and management of the increased variety of massive complex data streams being produced nowadays by applications, such as multimedia, social media, Internet of Things and social dispersed computing, are challenging issues (Hashem et al. (2015); García-Valls et al. (2018)). To cope with these challenges and accelerate large scale data processing and data analytics, "big data" applications exploit programming paradigms that support parallelism, e.g., MapReduce. These applications are often deployed in cloud or multi-cloud environments because of virtually unlimited storage and computation resources that can be dynamically provisioned on demand.

In this framework, cloud users are willing to minimize monetary costs and delays by seamlessly combining their privately owned systems with external public infrastructure. On the contrary, cloud providers are mainly interested in maximizing resource utilization and throughput and more specifically their profit. Therefore, to identify cost-effective solutions and achieve the desired service levels, provisioning and scheduling of cloud resources play a critical role.

The estimation of the resources to be provisioned to an application is a complex task since their performance varies and depends on many factors, including, among the others, the mix of workloads being concurrently processed (see, e.g., Calzarossa et al. (2016a,b)). In particular, the use of virtualized resources and the co-location of heterogeneous applications on a given physical machine could lead to contentions and performance instability that are often not directly correlated to cloud characteristics, as suggested by Ristov et al. (2017). Moreover, in the case of data parallel applications that require massive data and computations, performance fluctuations could also result in a severe load imbalance across Virtual Machines (VM) (see Calzarossa et al. (2003); Della Vedova et al. (2016b). Since all these uncertainties have a direct impact on resource provisioning and scheduling decisions, it is important to identify the various types of uncertainty and their sources – see Jamshidi et al. (2016); Tchernykh et al. (2016) for their classification. In addition, these phenomena need to be taken into account before their effects appear, that is, prior to the application execution. This will also allow users to obtain an accurate estimation of the budget required to deploy their applications and take advantage of the reservation options that offer significantly cheaper cloud resources with respect to the on demand counterpart. In fact, neglecting uncertainties could lead to inefficient resource provisioning, i.e., over-provisioning or under-provisioning.

In this paper, we propose a resource provisioning and scheduling approach for data parallel applications that includes explicitly cloud uncertainties and performance variability in its definition. According to this approach provisioning and scheduling decisions are evaluated offline, i.e., prior to the actual execution of the application, thus reducing for cloud users the risk of over-provisioning and under-provisioning. In particular, to identify – among the large variety of resources offered nowadays by cloud providers – an "optimal" set of resources to be provisioned and the corresponding scheduling plan, our methodological framework relies on the formulation of an optimization problem where the characteristics of the cloud infrastructure not perfectly known or affected by uncertain phenomena are represented as random variables modeled by the corresponding probability distributions. Thus, the provisioning and scheduling decisions – while optimizing various parameters, such as monetary leasing costs of cloud resources and application execution time – take fully account of uncertainties encountered in cloud environments.

To test our methodological framework we develop various scenarios affected by uncertainty. Moreover, as a comparison baseline, we define two reference models that do not include any variability in the decision process, even though the variability is actually affecting the performance of the cloud infrastructure. The evaluation of these scenarios relies on an integrated environment implemented as extensions of the CloudSim toolkit. In particular, to validate the proposed framework the identified resource provisioning and scheduling plans have been tested under different uncertainty conditions.

We summarize our contributions as follows:

- Definition of a methodological framework for offline resource provisioning and scheduling that copes with performance uncertainty and variability of workload characteristics and cloud environments.

- Customization of well-known meta-heuristics for solving combinatorial optimization problems.

- Development of an integrated environment – implemented as extensions of the CloudSim toolkit – for the probabilistic evaluation of resource provisioning and scheduling plans prior to the actual execution of the applications.

The paper is organized as follows. Section 2 reviews the state of the art of resource provisioning and scheduling in cloud environments by focusing in particular on the issues related to performance uncertainty. Section 3 presents in detail the formal settings of the proposed methodological framework. The setup of the experiments carried out to test the proposed approach is described in Section 4, while their results are discussed in Section 5. Finally, Section 6 draws some concluding remarks and outlines future research directions.

2. Related work

Many research efforts have been directed towards the investigation of resource provisioning and task scheduling in cloud environments. Most of these works formulate optimization problems whose goal is to identify resource settings that minimize various parameters (e.g., monetary cost, execution time, energy consumption) subject to some workload constraints (e.g., budget, deadline). In what follows, we review the approaches applied in the literature to address these research challenges and we outline the novelty of our approach.

2.1. Resource provisioning and scheduling

An optimization problem with a binary integer program formulation is presented in Van den Bossche et al. (2010) to study resource provisioning and scheduling of a set of batch applications characterized by hard deadlines. An integer linear programming formulation is proposed in Ruiz-Alvarez et al. (2015) for the optimal resource provisioning of two specific classes of cloud applications, namely, MapReduce applications and Monte Carlo simulations. In particular, to devise an optimal scheduling plan based on cost and execution time, the problem formulations take into account the characteristics of the infrastructure as well as the peculiarities of the applications. Cost-effective VM provisioning approaches for deadline-constrained MapReduce applications are suggested in Hwang and Kim (2012). Resource provisioning of MapReduce applications is also addressed by Chen et al. (2014) who introduce various optimization problems. The core component for solving these problems is represented by a cost model that takes into account the relationship among amount of input data, available system resources and application-specific complexity. The problem of cost-efficiently scheduling deadline-constrained bag-of-tasks applications in hybrid cloud environments is studied by Van den Bossche et al. (2013). The proposed algorithms consider the task properties as well as the inaccuracies in their runtime estimates. Chen et al. (2017) present a heuristic algorithm for task scheduling of budget constrained parallel applications in heterogeneous cloud environments. Moschakis and Karatza (2015a,b) focus on meta-heuristic optimization algorithms to schedule bag-of-task applications in a dynamic multi-cloud environment characterized by heterogeneous performance and fluctuating costs even under sporadically arriving critical jobs.

A comparative review of workflow scheduling algorithms is offered by Wu et al. (2015) in the framework of a taxonomy defined according to workflow properties (e.g., structure, execution time) and resource characteristics.

Despite these studies, our modeling framework takes explicitly account of the presence of uncertainty in cloud performance as well as in the characteristics of the workload to be processed and it allows cloud users to assess the impact of the uncertainty prior the execution of their applications.

2.2. Provisioning and scheduling under uncertainty

As already pointed out, resource and workload uncertainties have a direct impact on provisioning and scheduling decisions. Nevertheless, few papers explicitly deal with the effects exercised by uncertainty on these decisions. Chaisiri et al. (2012) include the uncertainty associated with resource demand and pricing in multi-cloud environments in their resource provisioning algorithm. The goal of this algorithm – formulated as a stochastic programming problem – is to minimize the provisioning costs by adjusting the tradeoff between resource reservation and on demand allocation. The provisioning and scheduling strategy for scientific workflows presented by Rodriguez and Buyya (2014) model the variations of cloud performance and data transfer by simply inflating the processing capacity and bandwidth according to the estimations of the performance variance reported by Schad et al. (2010). On the contrary, Fard et al. (2016) address the uncertainty under the assumption of unknown processing time of a workflow activity by proposing a robust approach based on upper and lower bounds of processing times. Tang et al. (2017) formulate a linear programming model to address the problem of scheduling tasks – whose characteristics are assumed stochastic or unknown in advance – on heterogeneous clouds under budget constraints. A probabilistic approach that takes into account uncertainties is considered in Della Vedova et al. (2016a) to address the cloud resource provisioning and task scheduling of MapReduce applications.

Unlike most of the previous works, we formulate an optimization problem that minimizes the expected monetary cost under a probabilistic constraint on the application deadline. This problem represents workload and cloud uncertainty by means of random variables characterized by arbitrary probability distributions, whose algebraic composition describes the application cost and performance.

2.3. Simulation of uncertainty

Performance instability of cloud environments has also been addressed within simulation tools. For example, DynamicCloudSim – developed by Bux and Leser (2015) as an extension of the CloudSim simulation toolkit (Calheiros et al. (2011)) – introduces several capabilities to model the instability, including heterogeneity and dynamic changes of performance at runtime as well as failures during task execution. In particular, this tool provides the functionality to randomize the individual performance characteristics of a VM (e.g., CPU, I/O, bandwidth) by sampling a normal distribution. An extension of DynamicCloudSim is proposed by Mathá et al. (2017), who developed a model that injects some noise in the workload to mimic cloud performance uncertainty and simplify the setup of the simulator.

Similarly to this tool, we developed customized extensions of the CloudSim toolkit that allow us to analytically model and simulate both the workload and cloud characteristics. In details, we describe the uncertainty with any type of distribution without being limited to normal distributions. In fact, normal distributions do not always comply with real workload execution as stated by Ristov et al. (2017). Our integrated environment includes the Stochastic Simulation in Java (SSJ) software library (L'Ecuyer et al. (2002)) and the environment for statistical computing R (R Core Team (2018)).

2.4. Optimization algorithms

An additional issue faced by provisioning and scheduling decisions in cloud environments deals with finding global optimal solutions in very large search spaces. These problems are usually intractable with exact methods. Therefore, heuristic or meta-heuristic methods are often applied to obtain feasible solutions of optimization problems even though the solution is not guaranteed to be optimal. Simple methods as well as more sophisticated approaches are devised for this purpose. Zhan et al. (2015) examine the problem and solutions of resource scheduling and offer a comprehensive survey of the evolutionary approaches adopted in this framework. In particular, various combinations of Genetic Algorithms have been proposed to schedule independent tasks as well as tasks characterized by precedence constraints.

The solutions of the optimization problem proposed in this paper rely on customizations of the Branch-and-Bound and Genetic Algorithms whose cost and fitness functions take directly into account the probabilistic nature of the problem.

3. Methodological framework

This section presents the methodological approach proposed for resource provisioning and scheduling of data parallel applications being deployed in cloud environments. Our methodology allows cloud users to identify the best resource settings that meet their optimization targets. In particular, to cope with performance uncertainty, we define a general modeling framework whose overall architecture is sketched in Figure 1. According to this framework, the application and cloud characteristics that could be affected by uncertainty are described by independent random variables, whose probability distributions explain their variability. Furthermore, these distributions are evaluated in the optimization problems formulated to identify the resource settings, (i.e., resources to be provisioned and corresponding scheduling plans) that satisfy the selected performance and cost metrics. Due to the combinatorial nature of these problems, their solution requires the application of optimization methods based on heuristic or meta-heuristic approaches.

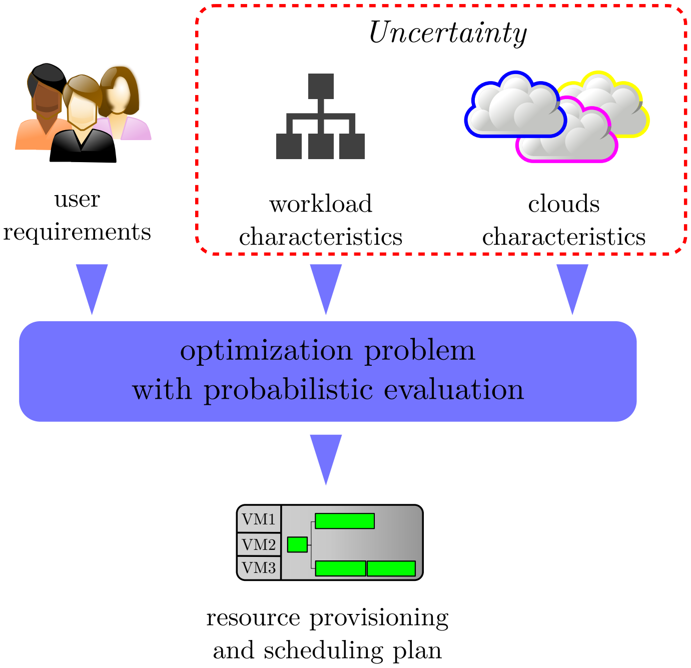
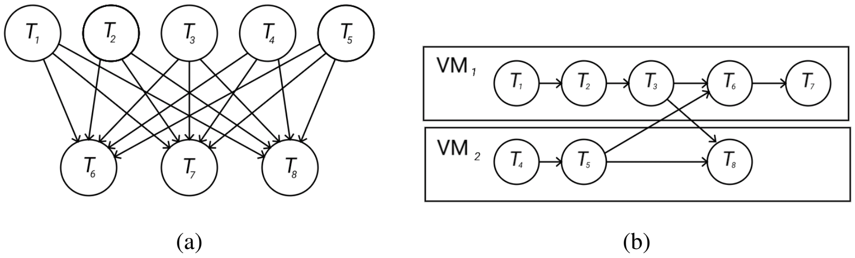

Uncertainty
user requirements
workload characteristics
clouds characteristics
optimization problem with probabilistic evaluation
VM1
VM2
VM3
resource provisioning and scheduling plan

Figure 1: Architecture of the modeling framework.

3.1. Modeling framework

Let us recall that data parallel applications are usually represented by Directed Acyclic Graphs whose nodes correspond to the tasks the application consists of and whose edges describe the dependencies among tasks, that is, their control and data precedence constraints and more specifically their execution order (see Figure 2(a)).



(a)

(b)

Figure 2: Directed Acyclic Graph of an application consisting of eight tasks (a) and corresponding scheduling plan on two VMs (b).

Therefore, according to our formalism, a data parallel application $\mathcal{A}$ is described by the set of its n tasks T_i, namely, $\mathcal{A} = \{T_i; i = 1, \dots, n\}$, and by the set $E \subseteq \mathcal{A} \times \mathcal{A}$ of the edges connecting pairs of tasks.

The resource requirements of the application, that is, its demands, are described in terms of the demands of the individual tasks. In particular, we define the following random variables to represent the demands of task T_i:

- $\mathrm{D}_i^{\mathrm{comp}}$: computation demand, that is, amount of processing;
- $\mathrm{D}_{i,j}^{\mathrm{comm}}$: communication demand, that is, data volume to be exchanged with task T_j;
- $\mathrm{D}_i^{\mathrm{xfer}}$: transfer demand, that is, data volume to be transferred to/from I/O devices.

The computation demand of each task can be expressed in million of instructions, while the communication and transfer demands in megabytes.

Similarly, the characteristics of the cloud infrastructure are described in terms of the performance of the VMs that can be provisioned to the application. More precisely, we define the following random variables associated with VM_i:

- $\mathrm{V}_i^{\mathrm{proc}}$: processing capacity;
- $\mathrm{V}_{i,j}^{\mathrm{bw}}$: network bandwidth between VM_i and VM_j;
- $\mathrm{V}_i^{\mathrm{xfer}}$: data transfer rate to/from I/O devices (e.g., local storage, distributed storage).

The processing capacity can be expressed in million instructions per second (MIPS), while the bandwidth and the transfer rate in megabits per seconds (Mbps) and megabytes per second, respectively.

While task demands and VM performance are expressed by random variables that account for uncertainty, the monetary costs of cloud infrastructure are not affected by any type of uncertainty. Therefore, we define the deterministic costs as follows:

- c_i^{VM}: leasing cost of VM_i;
- $c_{i,j}^{\mathrm{bw}}$: cost of network bandwidth between VM_i and VM_j;
- c_i^{xfer}: data transfer cost to/from I/O devices.

Finally, we introduce a function $match(i, j)$ that describes whether the characteristics of VM_i match the functional and non functional requirements of task T_j (e.g., number of cores, memory size, availability zone).

Note that multi-cloud environments usually consist of multiple instances of either private and public VM types (e.g., large, medium, small). In what follows, we model each of the characteristics of the instances of a given VM type with independent and identically distributed random variables.

To identify the optimal set of resources to be provisioned to a given application and the corresponding scheduling plan – prior to its execution – we define the resource provisioning $\mathcal{R}$ as the set of the VMs to be allocated, that is, $\mathcal{R} = \{\mathrm{VM}_i, i = 1, \ldots, m\}$. Moreover, the scheduling plan is defined by:

- a function $map : [1, \ldots, n] \rightarrow [1, \ldots, m]$ that assigns each of the n tasks to one of the m provisioned VMs matching its functional and non functional requirements;
- a preorder scheduling relation $\mathcal{S}$ that includes the task precedence constraints, that is, $\mathcal{S} = \{(T_i, T_j); T_i, T_j \in \mathcal{A}\}$, where (T_i, T_j) implies that T_j can be processed only after T_i has completed its execution.

In particular, in this paper we refer to static non preemptive scheduling approaches where decisions are taken offline, that is, prior to the application execution. Moreover, we assume that tasks are executed as soon as their scheduling constraints are fulfilled, that is, at their earliest start time.

Figure 2(b) shows an example of a scheduling plan on two VMs, i.e., $\mathcal{R} = \{\mathrm{VM}_1, \mathrm{VM}_2\}$, of the eight tasks of the application of Fig. 2(a). As can be seen, some of precedence constraints between tasks (e.g., between T_4 and T_8) are indirectly taken into account by the scheduling relationships (e.g., (T_4, T_5) and

(T_5, T_8)) under the assumption of strictly positive task execution times. Moreover, we notice that the scheduling has introduced additional precedences between tasks that result in their sequential processing on the allocated VMs.

As already pointed out, we model the application and cloud characteristics affected by uncertainty in terms of random variables. Hence, to predict the execution time of a task ahead of its actual execution, we perform algebraic computations on these random variables. In detail, the total execution time t_i of task T_i includes three components, namely:

- t_i^{comp} i.e., processing time;
- t_i^{comm} i.e., time spent to communicate with other tasks by exchanging data;
- t_i^{xfer} i.e., time spent to transfer data to/from I/O devices.

Hence, under the assumption that task T_i is allocated to VM_k, that is, $k = map(i)$, and sequential communications among VMs, we can express these three components as follows:

$$t_i^{\text{comp}} = \frac{\text{D}_i^{\text{comp}}}{V_k^{\text{proc}}}$$

$$t_i^{\text{comm}} = \sum_{\substack{j=1; j \neq i \\ l = map(j)}}^{n} \frac{\text{D}_{i,j}^{\text{comm}}}{V_{k,l}^{\text{bw}}}$$

$$t_i^{\text{xfer}} = \frac{\text{D}_i^{\text{xfer}}}{V_k^{\text{xfer}}}$$

Note that in the case of parallel communications t_i^{comm} is given by the maximum of all communication times instead of their sum.

The task execution time t_i is the random variable computed as the sum of the random variables previously defined, that is, $t_i = t_i^{\text{comp}} + t_i^{\text{comm}} + t_i^{\text{xfer}}$. By combining these t_i's according to the preorder scheduling relation $\mathcal{S}$, we obtain the random variable $T_{\mathcal{A}}$ corresponding to the overall execution time of the application.

The monetary cost c_i for leasing the cloud resources allocated to task T_i is obtained as follows:

$$c_i = t_i \times c_i^{\text{VM}} + \text{D}_i^{\text{xfer}} \times c_i^{\text{xfer}} + \sum_{\substack{j=1; j \neq i \\ l = map(j)}}^{n} \text{D}_{i,j}^{\text{comm}} \times c_{k,l}^{\text{bw}}$$

The overall cost C for processing the application is given by the sum of the costs of the individual tasks, namely, $C = \sum_{i=1}^{n} c_i$. Note that the c_i's and C are random variables as they are the result of algebraic computations on random variables.

The probability distributions of the application execution time and monetary cost describe the uncertainty of workload and cloud infrastructure and allow the resource provisioning and scheduling plans to take explicitly into account the effects of the uncertainty.

3.2. Probabilistic evaluation

According to the modeling framework introduced in the previous section, the execution time t_i of task T_i and the monetary cost c_i of the cloud resources provisioned to the task are random variables. Similarly, as the overall execution time $T_{\mathcal{A}}$ of the application and its cost C are obtained from these random variables, they are themselves random variables. Therefore, it is necessary to perform some algebraic computations over non negative independent random variables. The types of computation depend on the specific relationships being created by the resource provisioning and scheduling plan identified for the application.

We recall that a random variable is described in terms of its cumulative distribution and probability density functions. More specifically, let us denote with X, Y and Z three non negative independent random variables, where Z is obtained from X and Y after some algebra (see, e.g., Grimmett and Stirzaker (2001); Petrov (2012)). Moreover, let us denote with F_X, F_Y, F_Z and f_X, f_Y, f_Z their probability distribution and density functions, respectively. The probability distribution function F_Z of $Z = X + Y$ is given by:

$$F_Z(z) = \int_{-\infty}^{+\infty} F_Y(z-x) f_X(x) dx = (F_Y * f_X)(z)$$

where $*$ denotes the convolution product. Similarly, F_Z of $Z = X/Y$ is obtained as follows:

$$F_Z(z) = \int_{-\infty}^{z} \int_{-\infty}^{+\infty} |y| f_Y(x \times z) f_Y(z) \, dy \, dz$$

On the other hand, F_Z of $Z = \max\{X, Y\}$ is given by:

$$F_Z(z) = F_X(z) \times F_Y(z)$$

As discussed in Sect. 3.1, the random variable t_i is derived from the random variables associated with task demands and cloud infrastructure characteristics. More specifically, the task processing time t_i^{comp} and the time t_i^{xfer} spent in I/O transfers are computed as the ratio of two random variables, while the time for data exchanges t_i^{comm} is computed as the sum or the maximum of ratios of random variables.

The random variable $T_{\mathcal{A}}$ is then computed by properly applying addition and maximum operators to the t_i's. In particular, the addition of random variables takes into account the sequential execution of tasks on a given VM, whereas the maximum takes into account the precedence constraints introduced by the scheduling plan. For example, for the scheduling plan shown in Fig. 2(b), the overall execution time of the application $T_{\mathcal{A}}$ is given by:

$$T_{\mathcal{A}} = \max\{(t_1 + t_2 + t_3), (t_4 + t_5)\} + \max\{(t_6 + t_7), t_8\}$$

Note that the probability distribution functions of random variables obtained from algebraic computations of other random variables – as it is the case of $T_{\mathcal{A}}$ – are seldom analytically tractable. Hence, it is necessary to resort to their numerical evaluation. These evaluations are the basis for assessing the impact of the uncertainties typical of cloud environments on the resource provisioning and scheduling decisions.

3.3. Provisioning and scheduling

In cloud and multi-cloud environments, the decisions associated with resource provisioning and scheduling strategies are always very challenging because of the large number of possible choices. These strategies aim at finding “optimal” resource settings that cope with the application requirements and cloud uncertainty. Therefore, the identification of a resource provisioning set $\mathcal{R}$ and a scheduling plan $\{map(), \mathcal{S}\}$ is formulated as an offline optimization problem whose goals include various performance and cost metrics.

The total monetary cost required to deploy an application in the clouds is a typical metrics that needs to be minimized, while at the same time some probabilistic guarantees on the obtained performance are to be satisfied. For any pair of resource provisioning $\mathcal{R}$ and scheduling plan $\{map(), \mathcal{S}\}$, such a problem can be formulated as follows:

$$\begin{aligned} \text{minimize} \quad & \mathbb{E}\left[C\right] \\ \text{subject to} \quad & \Pr(T_{\mathcal{A}} \leq d) \geq p \end{aligned} \tag{1}$$

where $\mathbb{E}\left[C\right]$ and d denote the expected overall cost and the deadline associated with the application execution time, while p is a measure of the acceptable risk, namely, the probability that the deadline is satisfied.

To identify the cheapest scheduling plan that satisfies the deadline constraint, a straightforward approach consists of evaluating all possible mappings between the tasks and the VMs to be provisioned. Nevertheless, this approach is seldom feasible especially in the case of data parallel applications to be deployed in multi-cloud environments. In fact, the solution space of this combinatorial problem grows exponentially with the number of tasks the application consists of and the number and types of VMs to be provisioned. Hence, heuristic or meta-heuristic approaches are applied to identify suitable – even though non optimal – resource settings.

3.4. Optimization methods

The methods applied to solve the optimization problems have to cope with the nature and characteristics of the application to be deployed. Therefore, popular methods, such as Branch-and-Bound and Genetic Algorithms, require customized extensions that take also into account the probabilistic nature of the problem. In what follows, we present the extensions developed to cope with our modeling framework. In particular, without loss of generality, we will focus on data parallel applications consisting of n tasks distributed across two sequential phases, where all tasks of the first phase distribute their output across the tasks of the second phase as depicted in Fig. 2(a).

3.4.1. Branch-and-Bound

The Branch-and-Bound (BB) is a well known state space search technique for exploring in a rooted tree structure all solutions of discrete optimization problems (see, e.g., Brusco and Stahl (2006)). Our extensions focus on an n level solution tree whose levels correspond to application tasks and whose nodes to all possible VMs that can be provisioned and satisfy at the same time the $match$ function.

The root-to-leaf branches encode the candidate scheduling plans. These branches are built according to the task precedence constraints and evaluated by means of a cost function defined as follows:

$$f(\mathcal{R}, map) = \begin{cases} \mathbb{E}\left[C\right] & \Pr(T_{\mathcal{A}} \leq d) \geq p \\ +\infty & \Pr(T_{\mathcal{A}} \leq d) < p \end{cases} \tag{2}$$

More precisely, we discard solutions (i.e., assign them an infinite cost) that satisfy the deadline with a probability less than p and rank the other solutions according to their expected cost. In addition, to identify the cheapest resource provisioning and scheduling plan, we discard the partial branch explorations whose cost is higher than the cost of previously explored root-to-leaf branches.

Various optimization techniques based on branch pruning are usually applied to reduce the solution space exploration by discarding redundant equivalent solutions. For example, at each tree level when an additional VM has to be provisioned to a given task, our approach considers only one VM per type (see Figure 3).

Figure 3 depicts a portion of the BB exploration tree for the application of Fig. 2 (a) to be scheduled on two VM types. The highlighted branch corresponds to the identified scheduling plan on two VMs, namely, an instance of a small VM and an instance of a large VM. Each root-to-leaf branch describes the provisioned resource set $\mathcal{R}$ and the map scheduling function.

Let us remark that because of the exponential growth of the number of branches, the application of the BB algorithm is not always feasible.

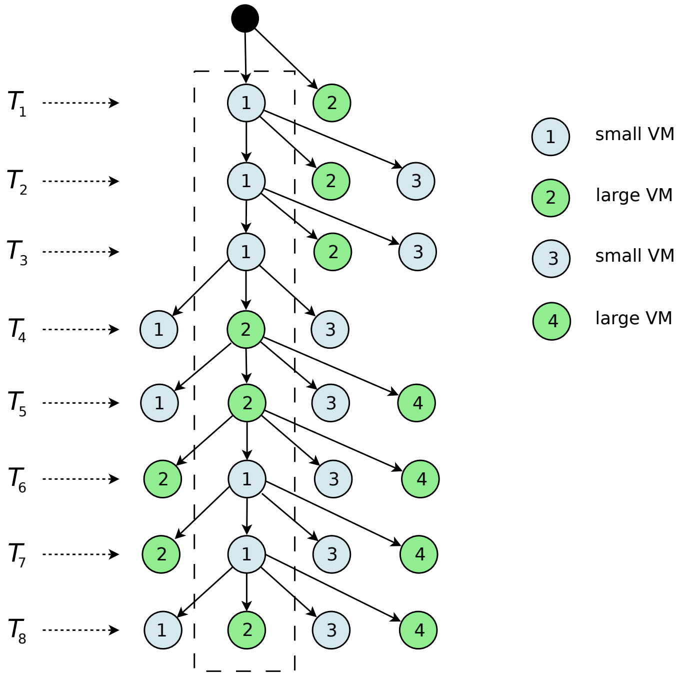


Figure 3: Example of pruned tree derived with the Branch-and-Bound algorithm. The branch in the box corresponds to the scheduling plan.

3.4.2. Genetic Algorithm

A Genetic Algorithm (GA) is an adaptive heuristic search method for finding a global suboptimal solution within a predefined processing time. The underlying idea is to study the evolution of a population of possible candidate solutions, i.e., individuals or chromosomes in the GA terminology (see, e.g., Affenzeller et al. (2009)).

To apply the GA approach to our optimization problem, we formulate this problem as an integer problem. In detail, each individual – encoded by a vector M of n integers – describes the mapping between the tasks and the VMs corresponding to the identified resource provisioning and scheduling plans. Figure 4 shows an example of the individual representing the scheduling plan of eight tasks on the two provisioned VMs. Each element $M[i]$ encodes the VM – satisfying the $match$ function – where task T_i is scheduled. Therefore, M is the tabular representation of the map function and the unique VM instances represent the provisioned resource set $\mathcal{R}$.

The choice of the initial population is very important as it affects the processing

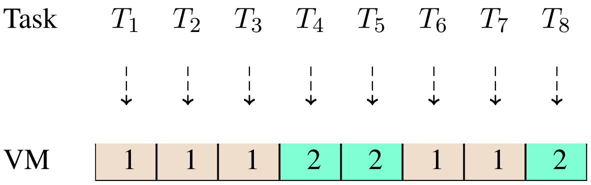


Figure 4: Individual representing the scheduling plan of Fig. 2 (b).

time of the Genetic Algorithm. Therefore, instead of randomly selecting the individuals, we generate the initial population using two simple bin packing heuristics (see Hwang and Kim (2012)) that take advantage of the structure of data parallel applications, namely:

- *List and First Fit*. According to this heuristic, each task of the first sequential phase is allocated to one VM as a function of its expected execution time and monetary cost. The tasks of the second phase are then allocated to the VMs already provisioned. The allocation process is repeated until the proposed solution satisfies the desired deadline.

- *Deadline-aware Tasks Packing*. According to this heuristics, the application deadline is broken down into sub-deadlines associated with each phase of the application. A first-fit strategy is applied to allocate tasks to the VMs – sorted according to their cost/performance ratio. In particular, starting from the tasks of the first phase, each VM is allocated as many tasks as the sub-deadlines allow.

In addition, the initial population includes the individuals associated with a fully parallel schedule of the tasks on the cheapest VMs as well as with fully parallel and sequential schedule on the fastest VMs.

At each generation, we rank the population according to a fitness function inversely proportional to the cost function (2) introduced in Section 3.4.1. Therefore, top ranked individuals – identified according to tournament selections – are preserved in the next generation, while individuals with lower fitness scores are replaced by offsprings randomly generated according to simulated binary crossover and polynomial mutation operators.

We note that the parameters associated with the population size, the probabilities of crossover and mutation, and the number of tournaments are used to fine tune the trade off between the exploitation and the exploration of the search space.

4. Experimental setup

In this section we describe the setup of the experiments carried out to test our methodological framework and probabilistically evaluate the corresponding resource provisioning and scheduling plans. The experiments rely on an integrated environment developed to cope with the probabilistic nature of the problem. In particular, we consider the deployment of data parallel applications in a simple model of a hybrid multi-cloud environment. Moreover, we define some scenarios to be used as a reference for our probabilistic framework.

4.1. Experimental environment

The experimental environment developed for testing our methodological framework allows the identification of optimal resource provisioning and scheduling plans as well as their simulation. As shown in Figure 5, this integrated environment consists of a core component that includes the optimization and probabilistic evaluation modules. In detail, the *optimization module* – responsible of the exploration of the solution space – includes the Branch-and-Bound algorithm and the Genetic Algorithm heuristic. In particular, the implementation of BB is based on existing CloudSim extensions (Alrokayan et al. (2014)), while GA relies on `jMetal` framework (Durillo and Nebro (2011)).

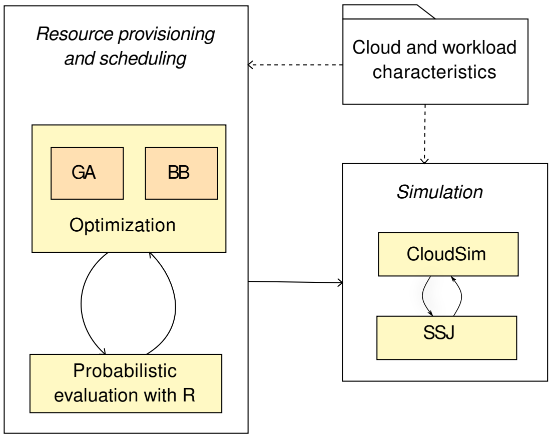

Resource provisioning and scheduling
GA
BB
Optimization
Probabilistic evaluation with R
Cloud and workload characteristics
Simulation
CloudSim
SSJ

Figure 5: Architecture of experimental environment.

The optimization module is strictly coupled with the *probabilistic evaluation module* for ranking the solutions that satisfy the deadline constraint according to their cost. More precisely, the evaluation of their probability distribution functions relies on numerical techniques based on spectral methods. For this purpose, our evaluation module integrates `distr` package of the environment for statistical computing R (Ruckdeschel and Kohl (2014)).

An add-on of our experimental environment is a simulation component for reproducing the execution of the workload deployed in uncertain cloud environments. In detail, the simulation component consists of customized extensions of the *CloudSim* toolkit that integrates the Stochastic Simulation Library for tackling the uncertainty.

Various probability distributions are implemented in the core and simulation components for describing the cloud and workload characteristics.

4.2. Cloud and workload characteristics

As already pointed out, to test our methodological framework we model a batch workload consisting of data parallel applications to be executed in a multi-cloud environment. More precisely, we model two providers – each offering multiple instances of several types of VMs – and a private cloud infrastructure deploying multiple instances of two VM types.

The main characteristics of the cloud infrastructures are summarized in Table 1. In particular, the processing capacity $\mathrm{V}_i^{\mathrm{VM}}$ of each VM type is modeled in

Table 1: Main characteristics of the cloud infrastructures considered in the experiments.

Provider	VM Type	ECU	Leasing cost [USD/h]	Proc. capacity [MIPS $\times 10^3$]	Bandwidth [Mbps]
Public cloud A	*micro*	0.5	0.040	1.95	300
	small	1	0.080	3.91	300
	medium	4	0.320	15.63	600
	large	6.5	0.520	25.38	800
	xlarge	8	0.640	31.25	800
	x2large	13	1.040	51.02	1,100
	x3large	26	2.080	101.63	1,100
Public cloud B	*micro*	0.5	0.045	1.95	300
	small	1	0.090	3.91	300
	medium	2	0.180	7.81	600
	large	4.1	0.369	16.03	800
	xlarge	8.6	0.774	33.67	1,100
Private cloud	*small*	0.5	0.001	1.95	800
	medium	2	0.001	7.81	800

terms of MIPS rate. We note that different VM types are characterized by different processing capacities independently of the type of hardware. On the contrary, the nominal bandwidth $\mathrm{V}_{i,j}^{\mathrm{bw}}$ between VMs depends on their type. For example, the bandwidth for *x3large* instances is equal to 1,100 Mbps, while for *micro* type VM instances it is equal to 300 Mbps. The table also lists for each VM type the corresponding EC2 Compute Unit (ECU) – a measure of the integer processing power introduced by Amazon – and the corresponding cost – expressed in USD/hour. In the experiments we will use a per-minute billing.

Additional specifications of the cloud infrastructures refer to the bandwidth between different VM types, that is assumed to be equal to the smaller of the two nominal bandwidths, e.g., 300 Mbps for a *micro* VM instance and any VM instance. Similarly, the bandwidth across the public clouds is assumed equal to 800 Mbps, while between private and public clouds to 150 Mbps. Moreover, since the storage devices are associated with a public cloud infrastructure, the corresponding data transfer rate $\mathrm{V}_{i}^{\mathrm{xfer}}$ varies with the bandwidth.

The workload considered in our experiments consists of two data parallel applications characterized by a different degree of parallelism. In addition, without loss of generality, we assume the tasks of each application organized in two sequential phases – of u and v tasks – and characterized in each phase by the same transfer and processing demands.

One of our target applications – referred to as $5/3$ in what follows – consists of five parallel tasks in the first phase and three parallel tasks in the second phase, while the other – referred to as $64/16$ – includes 64 and 16 parallel tasks in the two phases, respectively. The tasks of the first phase read their input data from a data storage and – after some local processing – communicate with every other task of the second phase for transferring their output. The tasks of the second phase, in turn, process this data and transfer their output back to the data storage. The overall characteristics of the target applications are as follows:

- total transfer demand – input: $\sum_{i=1}^{u} \mathrm{D}_{i}^{\mathrm{xfer}} = 1.5$ TB;
- total transfer demand – output: $\sum_{i=u+1}^{u+v} \mathrm{D}_{i}^{\mathrm{xfer}} = 750$ GB;
- total communication demand: $\sum_{i,j} \mathrm{D}_{i,j}^{\mathrm{comm}} = 750$ GB;
- total computation demand – first phase: $\sum_{i=1}^{u} \mathrm{D}_{i}^{\mathrm{proc}} = 1.7 \times 10^{7}$ million of instructions;
- total computation demand – second phase: $\sum_{i=u+1}^{u+v} \mathrm{D}_{i}^{\mathrm{proc}} = 3.75 \times 10^{6}$ million of instructions.

Note that by varying these characteristics we can model compute intensive or data intensive applications. In addition, we consider a fixed application deadline $d = 36$ hours to be satisfied with a probability p greater or equal to 0.8.

In what follows, for the sake of simplicity, we assume task demands perfectly known and not affected by any variability. On the contrary, we model uncertainty and variability of cloud performance in terms of well-know probability distributions, such as Uniform, Half-Normal and Weibull. More precisely, to take account of the upper bounds associated with VM processing capacity, bandwidth and transfer rate – bounds that correspond to their maximum (i.e., nominal) performance – we model the reciprocal of the positive random variables describing these characteristics. In fact, many well-known distributions are not upper bounded.

Moreover, we will describe the uncertainty associated with the probability distributions in terms of a variability factor *VF* that represents the relative deviation with respect to the nominal performance of the VMs. In the experiments, we will vary this factor to investigate the impact of different degrees of uncertainty on the behavior of our approach.

4.3. Baseline models

As part of the experimental setup, we define two baseline models that differ from the proposed approach in the estimation of the execution time of the application under the various resource provisioning and scheduling plans. These estimations are based on simple algebraic computations performed on the constant values describing the cloud performance.

In particular, we define the "Nominal" and "Average" (Avg) baseline models. The Nominal model assumes the performance not affected by any variability or uncertainty. Hence, cloud characteristics are simply specified by their nominal values, i.e., maximum performance (see Table 1).

On the contrary, the Avg model tries to take account of the degradation affecting cloud performance by assuming some – although limited – knowledge of the actual performance, that is, the average values. More precisely, from the expected value of a random variable X we derive the corresponding estimation (or underestimation), namely: $x = (1 - \alpha)\mathbb{E}[X]$, with $0 \leq \alpha < 1$.

In what follows, we consider the Nominal and the Avg models as well as the Avg-25% model, that is, an underestimation of cloud performance with $\alpha = 0.25$. These models will be used as a reference for our approach, denoted to as Probabilistic.

5. Experimental results

In this section we describe the results of the experiments carried out to test our methodological framework and probabilistically evaluate the corresponding resource provisioning and scheduling plans. In particular, the experiments investigate the impact of the characteristics of the applications (i.e., input transfer demand, degree of parallelism) and of the variability of the cloud infrastructure (i.e., probability distribution, variability factor). Moreover, we compare the behavior of the proposed approach with the baseline models.

5.1. Validation

As an introductory proof of concept, we focus on estimations of the execution time of the $5/3$ application with the resource provisioning and scheduling plan depicted in Fig. 2. Two VMs are provisioned, namely, one *xlarge* (i.e., VM_1) and one *medium* (i.e., VM_2) instance of Public cloud B (see Table 1), whose variabilities are modeled with Half-Normal distributions and *VF*$= 0.3$.

In particular, we compare the execution time $T_{\mathcal{A}}$ computed according to our probabilistic approach with the times obtained by simulating the behavior of the application using our integrated environment. Figure 6 shows the two corresponding probability density functions. As can be seen, the distribution of the execution time provided by our framework nicely copes with the distribution obtained with 1,000 simulation replications. This is also confirmed by statistical tests, such

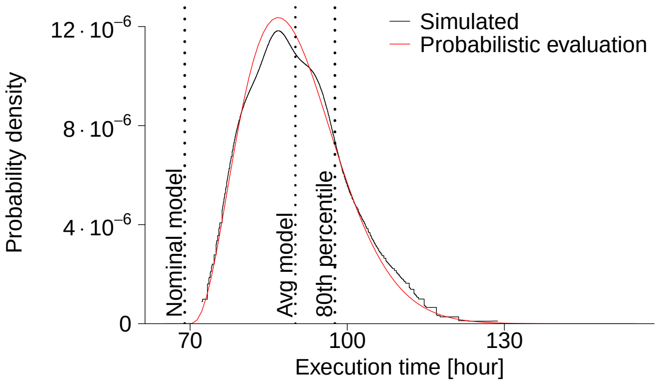


Figure 6: Probability density functions of the execution time of the $5/3$ application obtained by probabilistic evaluation and by simulation. Vertical lines refer to two baseline models and to the 80th percentile of the distribution of the execution time obtained with the probabilistic evaluation.

as the quantile-quantile plots and the Kolmogorov-Smirnov test (i.e., $D = 0.025$, $p-$value $= 0.9135$).

Moreover, to investigate the behavior of the baseline models previously defined, we plot in the figure the execution times derived by these models, namely, Nominal and Avg. Note that the lower bound of $T_{\mathcal{A}}$ – about 70 hours – corresponds to a cloud infrastructure without any variability, i.e., to the Nominal model. On the contrary, the execution time for the Avg model – that is about 90 hours – is computed using the expected values of the random variables describing the cloud variability. We finally observe that the 80th percentile of the distribution of the execution time corresponds to the deadline constraint of the optimization problem considered in the following experiments.

5.2. Impact of cloud variability

To assess the impact and the sensitivity of the proposed approach to cloud variability, we perform several experiments where we vary the probability distributions describing the VM characteristics as well as the corresponding variability factors *VF*. In detail, these tests – that focus on the 5/3 application – consider three different probability distributions (i.e., Uniform, Half-Normal, Weibull). Moreover, the value of *VF* is set to 0.3 for the network bandwidth $\mathrm{V}_{i,j}^{\mathrm{bw}}$ and data transfer rate $\mathrm{V}_{i}^{\mathrm{xfer}}$, while for the VM processing capacity $\mathrm{V}_{i}^{\mathrm{VM}}$ it varies from 0.1 to 0.7.

The optimal resource provisioning and scheduling plans identified by the Branch-and-Bound algorithm are evaluated in terms of the probability that the deadline is satisfied and of the expected monetary cost required to deploy the application in uncertain cloud conditions (see Figure 7) As expected, our approach guarantees the deadline constraint regardless of the type of distribution, resulting in probabilities always greater or equal to the desired value, that is, 0.8. Moreover, the figure suggests that the selection of the resource provisioning and scheduling plans is influenced by both the distribution type and *VF*. In particular, the greater the variability, the more expensive the set of resources required to ensure the deadline constraint. For example, the minimum cost estimated by the probabilistic evaluation, that is, 236.20 USD, corresponds to Uniform distribution with *VF* equal to 0.1. Eight VMs have been allocated, namely, five *x2large* of Public cloud A and three *xlarge* of Public cloud B. On the contrary, the maximum cost, i.e., 316.33 USD, corresponds to Half-Normal distribution with *VF* equal to 0.7 where five *x3large* VMs of Public cloud A have been allocated.

It is worth to point out that the provisioning and scheduling problem (see Eqn. (1)) may not allow any solution when the variability increases above a certain value. For example, in the case of Half-Normal distribution with *VF*$= 0.9$, even the fully parallel schedule of the application on the fastest VMs leads to a probability of satisfying the deadline equal to 0.62, well below the 0.8 constraint.

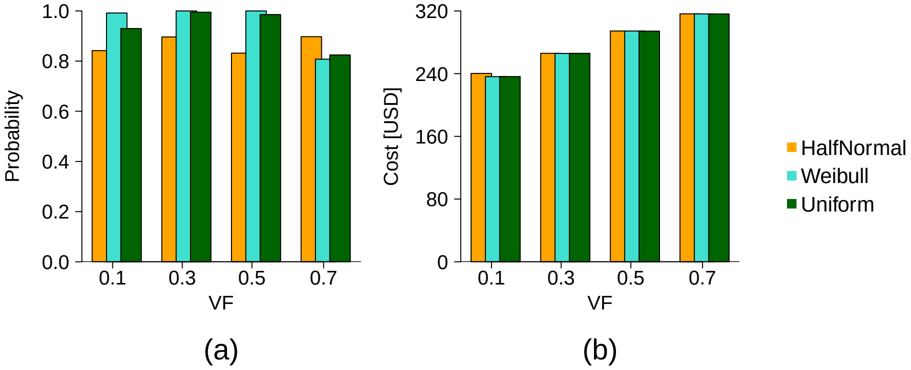


Figure 7: Probability of satisfying the deadline (a) and expected monetary cost (b) for the proposed approach under different variability conditions.

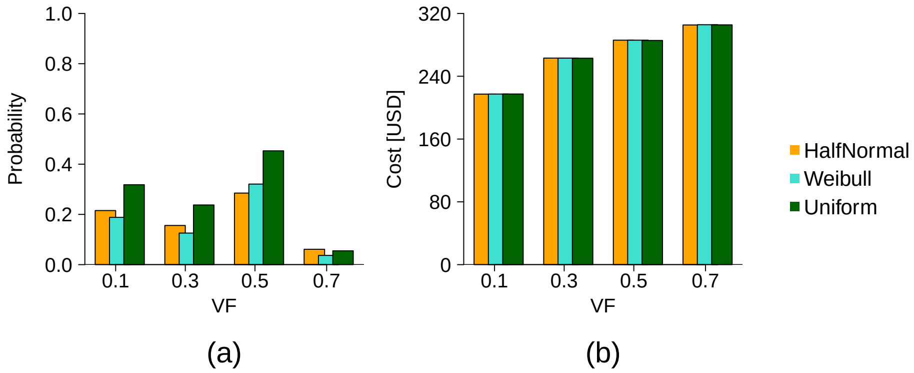
Finally, we compare the behavior of our approach with one of the baseline models defined in Sect. 4.3, namely, the Avg model. As shown by Figure 8 (a), this model leads to plans that always violate the deadline constraint. In fact, for identifying the optimal resource provisioning and scheduling plans the Avg model takes into account only the expected values of the VM characteristics and ignores their distributions. Nevertheless, the type and *VF* of these distributions affect the distribution of the execution time $T_{\mathcal{A}}$ and the probability to satisfy the deadline.

Despite the differences in the probabilities, the provisioned resources do not vary with the distribution type for a given *VF*. As can be seen from Figure 8 (b), the expected monetary costs depend on the *VF* value, while they are slightly affected by the distribution type. For example, in the three tests with *VF*$=0.7$, eight VMs have been provisioned, namely, five *x3large* of Public cloud A and three *xlarge* of Public cloud B and the corresponding monetary costs differ by less than one USD (i.e., 0.05%).

In summary, these tests have shown that our approach guarantees to satisfy the deadline constraint. The monetary costs are only about 4% higher than the cost associated with resource provisioning and scheduling plans identified by the Avg model.

5.3. *Impact of transfer demands*

These experiments are aimed at studying and comparing the behavior of the proposed approach with some baseline models as a function of the input transfer demand of the $5/3$ application. In particular, we vary the transfer demand from 1



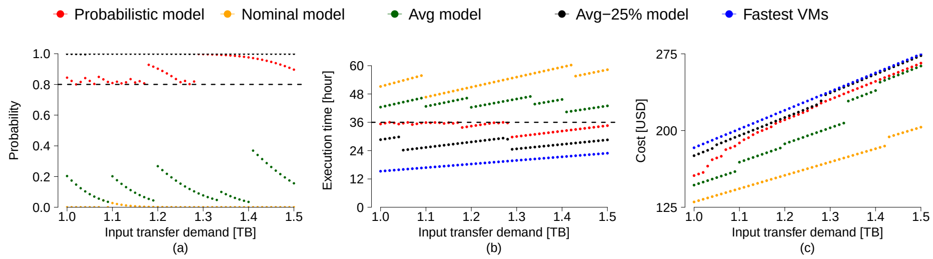
(a) (b)

Figure 8: Probability of satisfying the deadline (a) and expected monetary cost (b) for the Avg model under different variability conditions.

to 1.5 TB and scale proportionally the remaining application demands. Moreover, we model the cloud variability with Half-Normal distributions and $VF = 0.3$.

The optimal resource provisioning and scheduling plans – derived by the Branch-and-Bound algorithm – are evaluated in terms of the probability of satisfying the deadline, the application execution time and the expected monetary cost. As baseline models we consider Nominal, Avg and Avg-25%. Moreover, we also consider the Fastest VMs model, that corresponds to the fully parallel schedule of the application on the fastest available VMs.

Figure 9 summarizes the main results of these tests. In details, Figure 9 (a) confirms that our approach satisfies the deadline constraint regardless the transfer de-



(a) (b) (c)

Figure 9: Probability of satisfying the deadline (a), 80th percentile of the execution time (b) and expected monetary cost (c) for the $5/3$ application as a function of the input transfer demand.

mand. The Avg-25% and the Fastest VMs models exhibit the same behavior, while the Nominal and the Avg models violate the deadline constraint. These behaviors are also visible in the patterns of the application execution time (see Figure 9 (b)). Note that the figure plots the 80th percentile of the execution time since it corresponds to the deadline constraint of the optimization problem. For example, for the Probabilistic model these times range from 29.74 to 36 hours. Figure 9 (c) shows that the expected monetary costs increase with the transfer demand because of the increased computation and communication requirements. In particular, the Probabilistic model minimizes the expected monetary cost under the deadline constraint. On the contrary, the Avg-25% and the Fastest VMs models, while satisfying this constraint, do not minimize the monetary cost.

In the figures we notice piecewise patterns due to the discrete nature of the solution space of the optimization problem: each step corresponds to a change of the provisioning and scheduling plan to satisfy the deadline constraint. For example, for the Avg model the optimal solutions for transfer demands – ranging from 1.0 to 1.09 TB – correspond to the same plan, whereas for transfer demand equal to 1.10 TB a more expensive plan is identified leading to a shorter execution time. These changes are due to an estimation of the execution time (based on the expected value only) that would exceed the deadline. Figure 10 explains in detail this behavior by plotting the execution times estimated by the Avg model and the 80th percentile of the distribution of the random variable $T_{\mathcal{A}}$. Note that for the transfer demand equal to 1.09 TB the estimation is very close to the deadline, i.e., 35.98 hours, while the new resource provisioning and scheduling plan for 1.10 TB leads to an estimation of 33.02 hours. As a consequence, the expected monetary cost exhibits a sharp increase (i.e., 5.8%) as shown in Figure 9 (c)).

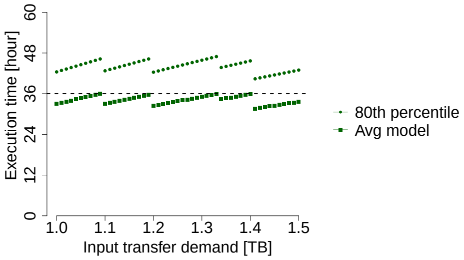


Figure 10: Estimated execution time for the Avg model vs. the 80th percentile of the random variable $T_{\mathcal{A}}$.

To analyze in more details the behavior of the Probabilistic model, we present in Table 2 the metrics previously defined together with the number of provisioned

VMs, the expected value and the 95th percentile of the execution time. These results refer to transfer demands ranging from 1.2 to 1.3 TB. The table highlights the four different provisioning and scheduling plans – whose characteristics are presented in Table 3 – identified as optimal solutions.

Table 2: Behavior of the Probabilistic model applied to the $5/3$ application for selected values of transfer demands. $T_{\mathcal{A}}^{(80)}$ and $T_{\mathcal{A}}^{(95)}$ denote the 80th and the 95th percentiles of the execution time distribution, respectively.

Transfer [TB]	**# VMs**	$\Pr(T_{\mathcal{A}} \leq d)$	$\mathbb{E}[C]$ [USD]	$\mathbb{E}[T_{\mathcal{A}}]$ [h]	$T_{\mathcal{A}}^{(80)}$ [h]	$T_{\mathcal{A}}^{(95)}$ [h]
1.20	5	0.902	210.52	31.63	34.37	37.32
1.21	5	0.888	212.24	31.90	34.66	37.63
1.22	5	0.872	214.04	32.16	34.95	37.95
1.23	5	0.854	215.76	32.42	35.23	38.26
1.24	5	0.835	217.56	32.69	35.52	38.57
1.25	5	0.815	219.28	32.95	35.80	38.88
1.26	6	0.823	222.30	32.80	35.69	38.81
1.27	6	0.802	224.10	33.06	35.97	39.11
1.28	5	0.819	226.83	32.72	35.74	38.98
1.29	8	0.998	228.67	27.49	29.74	32.13
1.30	8	0.997	230.42	27.71	29.97	32.37

We notice that these plan changes happen in accordance with the deadline constraint. More precisely, for transfer demand up to 1.25 TB, five *x2large* VM instances of Public cloud A are able to satisfy the deadline with probability greater or equal to 0.815. The 80th percentile of the execution time does not exceed the 36 hours deadline, even though it is very close, i.e., 35.80 hours. However, these VMs cannot cope with an increased transfer demand, hence, an extra *x3large* VM instance is provisioned. A further increase (i.e., 1.28 TB) leads to another change of the resource settings, where two tasks (i.e., T_5 and T_8) are scheduled on the same *x3large* instance. A new plan is also identified for transfer demand equal to 1.29 TB where the tasks are scheduled on eight VM instances from the two public cloud providers. Despite the larger number of VMs and the probability of satisfying the deadline close to one, this plan is only 0.8% more expensive than the previous plan.

In conclusion, these tests have shown that models other than the Probabilistic derive resource provisioning and scheduling plans that either do not satisfy the deadline or experience extra costs.

Table 3: Details of the four resource provisioning and scheduling plans derived by the Probabilistic model for the $5/3$ application for transfer demand from 1.20 to 1.30 TB (see Table 2 for their behavior).

	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
Provider	Public A	Public A	Public A	Public A	Public A	Public A	Public A	Public A
VM type	*x2large*	*x2large*	*x2large*	*x2large*	*x2large*	*x2large*	*x2large*	*x2large*
Instance	1	2	3	4	5	1	2	3
Provider	Public A	Public A	Public A	Public A	Public A	Public A	Public A	Public A
VM type	*x2large*	*x2large*	*x2large*	*x2large*	*x2large*	*x2large*	*x2large*	*x3large*
Instance	1	2	3	4	5	1	2	6
Provider	Public A	Public A	Public A	Public A	Public A	Public A	Public A	Public A
VM type	*x2large*	*x2large*	*x2large*	*x2large*	*x3large*	*x2large*	*x2large*	*x3large*
Instance	1	2	3	4	5	1	2	5
Provider	Public A	Public A	Public A	Public A	Public A	Public B	Public B	Public B
VM type	*x3large*	*x3large*	*x3large*	*x3large*	*x3large*	*xlarge*	*xlarge*	*xlarge*
Instance	1	2	3	4	5	6	7	8

5.4. Impact of parallelism

The last experiments focus on the impact of the increased parallelism of the application on the proposed approach. For this purpose, a first set of tests considers the $64/16$ application with transfer demands varying from 1 to 1.5 TB and cloud variability modeled with Half-Normal distributions and *VF*=0.3. In the second set of tests we vary *VF*, while keeping the transfer demand equal to 1.5 TB.

Because the larger number of tasks, the solution space of the optimization problem is vast and cannot be completely searched by an exact algorithm such as BB. In fact, with 14 types of VMs (see Table 1) the number of possible resource provisioning and scheduling plans is equal to $(14 \times (64 + 16))^{64+16} \approx 8.7 \times 10^{243}$. Therefore, we apply the Genetic Algorithm (see Section 3.4.2) to find a suboptimal solution of our optimization problem. In particular, our implementation is based on a standard single-objective generational algorithm in which we use a population of size ten and we set as termination criterion the maximum number of evaluations equal to 10,000. Moreover, we set the parameters of the crossover, mutation and selection operators as follows: crossover probability 0.9, distribution index for crossover and mutation 0.5, and number of tournaments five.

The probability of satisfying the deadline, the 80th percentile of the execution times and the expected monetary costs are plotted in Figure 11 against the transfer demand. As expected, the deadline constraint is always satisfied with probability greater or equal to 0.8. Similarly, the 80th percentile of the execution time very closely approaches the deadline. For example, for about three fourths of the 51 tests these times are within 30 minutes from the deadline. The monetary costs increase almost proportionally with the transfer demand from 137.68 to 212.49 USD.

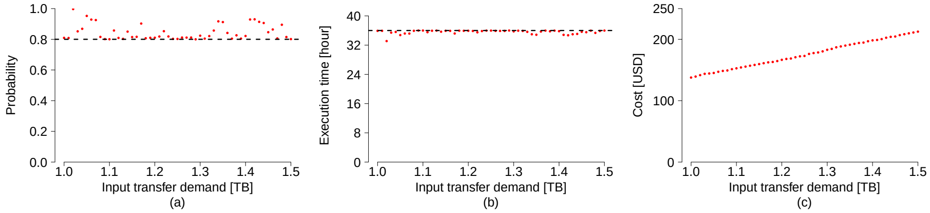


Figure 11: Probability of satisfying the deadline (a), 80th percentile of the execution time (b) and expected monetary cost (c) of the $64/16$ application computed by the Probabilistic model.

Let us remark that, despite the piecewise patterns exhibited by BB, GA does not result in regular patterns. This is due to the increased degree of application parallelism that leads to a finer-grained solution space of the optimization problem. Moreover, the randomness of the evolution of the GA population produces some jitters around the optimal (unknown) solution.

Finally, we investigate – with the second set of tests – the impact of the suboptimal solutions on the resource provisioning and scheduling plans as a function of the cloud variability. In the tests with *VF* equal to 0.1 and 0.7, the suboptimal solutions correspond to a provision of 15 and 21 VM instances, respectively. More precisely, four out of 15 and three out of 21 VMs belong to the Private cloud, whereas the remaining VMs are *xlarge* instances of Public cloud B. The probability of satisfying the deadline and the expected monetary costs are shown in Figure 12. Note that for *VF*=0.1 the probability is close to one, even though the existence

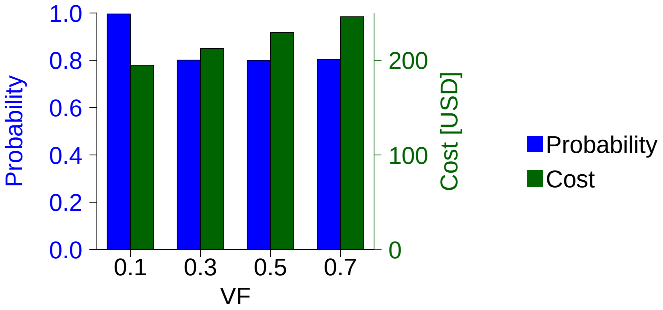


Figure 12: Probability of satisfying the deadline and expected monetary costs for the $64/16$ application under different variability conditions of the cloud infrastructure.

of cheaper resource settings that match the deadline constraint is very likely. As already stated, this is one of the side effects of the limited search of the solution space of GA.

As a conclusion, it is worth mentioning that – thanks to the customized extensions of GA – our approach is able to identify a solution – if it exists – that copes with the deadline constraint since it includes in the initial resource setting the fully parallel schedule of the application on the fastest VMs.

6. Conclusions

Resource provisioning and scheduling in cloud environments are particularly challenging because of the performance uncertainty and variability of the cloud infrastructure as well as of the workload being deployed. In this paper we defined a general modeling framework that tackles these issues. The application of this approach allows cloud users to estimate in advance, i.e., prior to the actual execution of the applications, the resource settings that cope with uncertainty, thus avoiding over-provisioning and under-provisioning.

More precisely, we represented the application and cloud characteristics subject to uncertainty by means of independent random variables, whose probability distributions explain their variability. These distributions – that can be empirically derived from measurements of real cloud environments – are an integral component of the optimization problem formulated to identify the resources to be provisioned and the scheduling plan that satisfy the desired performance and cost metrics.

To test our methodological framework, we developed an integrated environment – implemented as extensions of the CloudSim toolkit – and we defined various experimental scenarios affected by uncertainty as well as baseline models that do not take account of the variability in the decision process. In particular, the focus of our experiments was on the minimization of the expected monetary cost of the cloud resources to be provisioned to data parallel applications characterized by a deadline constraint.

The experiments have shown that our approach ensures to satisfy the deadline constraint independently of the cloud variability and the degree of parallelism of the application. On the contrary, the baseline models either fail to satisfy the deadline or require extra costs.

Future research directions could explore the feasibility of including in the decision process historical data as well as online measurements about the behavior of applications deployed in cloud environments. This process will require the development of novel customized extensions of well know optimization heuristics to cope with different workloads, such as complex workflows and interactive applications.

Acknowledgements

The authors would like to thank the anonymous referees for their valuable comments and suggestions that improved the overall quality and clarity of the manuscript.

References

M. Affenzeller, S. Wagner, S. Winkler, and A. Beham. *Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications*. CRC Press, 2009.

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)*, pages 1–8, 2014.

M. J. Brusco and S. Stahl. *Branch-and-bound applications in combinatorial data analysis*. Springer Science & Business Media, 2006.

M. Bux and U. Leser. DynamicCloudSim: Simulating heterogeneity in computational clouds. *Future Generation Computer Systems*, 46:85–99, 2015.

R.N. Calheiros, R. Ranjan, A. Beloglazov, C.A.F. 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*, 41(1):23–50, 2011.

M. Calzarossa, L. Massari, and D. Tessera. Load Imbalance in Parallel Programs. In V. Malyshkin, editor, *Parallel Computing Technologies*, volume 2763 of *Lecture Notes in Computer Science*, pages 197–206. Springer, 2003.

M. Calzarossa, M.L. Della Vedova, L. Massari, D. Petcu, M.I.M. Tabash, and D. Tessera. Workloads in the Clouds. In L. Fiondella and A. Puliafito, editors, *Principles of Performance and Reliability Modeling and Evaluation*, Springer Series in Reliability Engineering, pages 525–550. Springer, 2016a.

M.C. Calzarossa, L. Massari, and D. Tessera. Workload Characterization: A Survey Revisited. *ACM Computing Surveys*, 48(3):48:1–48:43, 2016b.

S. Chaisiri, Bu-Sung Lee, and D. Niyato. Optimization of Resource Provisioning Cost in Cloud Computing. *IEEE Transactions on Services Computing*, 5(2): 164–177, 2012.

K. Chen, J. Powers, S. Guo, and F. Tian. CRESP: Towards Optimal Resource Provisioning for MapReduce Computing in Public Clouds. *IEEE Transactions on Parallel and Distributed Systems*, 25(6):1403–1412, 2014.

W. Chen, G. Xie, R. Li, Y. Bai, C. Fan, and K. Li. Efficient task scheduling for budget constrained parallel applications on heterogeneous cloud computing systems. *Future Generation Computer Systems*, 74:1– 11, 2017.

M.L. Della Vedova, D. Tessera, and M.C. Calzarossa. Probabilistic Provisioning and Scheduling in Uncertain Cloud Environments. In *Proc. of the 2016 IEEE Symposium on Computers and Communication (ISCC)*, pages 797–803, 2016a.

M.L. Della Vedova, D. Tessera, M.C. Calzarossa, and J. Weinman. The Economics of Cloud Parallelism under Uncertainty. *IEEE Cloud Computing*, 3(6):16–22, 2016b.

J. J Durillo and A. J Nebro. jMetal: A Java framework for multi-objective optimization. *Advances in Engineering Software*, 42(10):760–771, 2011.

H. M. Fard, S. Ristov, and R. Prodan. Handling the Uncertainty in Resource Performance for Executing Workflow Applications in Clouds. In *Proc. of 2016 IEEE/ACM 9th Int. Conf. on Utility and Cloud Computing - UCC*, pages 89–98, 2016.

M. García-Valls, A. Dubey, and V. Botti. Introducing the new paradigm of Social Dispersed Computing: Applications, Technologies and Challenges. *Journal of Systems Architecture*, 2018.

G. Grimmett and D. Stirzaker. *Probability and random processes*. Oxford University Press, 2001.

I.A.T. Hashem, I. Yaqoob, N.B. Anuar, S. Mokhtar, A. Gani, and S.U. Khan. The rise of "big data" on cloud computing: Review and open research issues. *Information Systems*, 47:98–115, 2015.

E. Hwang and K.H. 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)*, pages 130–138. IEEE Computer Society, 2012.

P. Jamshidi, C. Pahl, and N. C. Mendona. Managing uncertainty in autonomic cloud elasticity controllers. *IEEE Cloud Computing*, 3(3):50–60, 2016.

P. L'Ecuyer, L. Meliani, and J. Vaucher. SSJ: A framework for stochastic simulation in Java. In E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, editors, *Proc. of the 2002 Winter Simulation Conference*, pages 234–242. IEEE Press, 2002.

R. Mathá, S. Ristov, and R. Prodan. A Simplified Model for Simulating the Execution of a Workflow in Cloud. In F. F. Rivera, T. F. Pena, and J. C. Cabaleiro, editors, *Euro-Par 2017: Parallel Processing*, volume 10417 of *Lecture Notes in Computer Science*, pages 319–331. Springer, 2017.

I. A. Moschakis and H. D. Karatza. A meta-heuristic optimization approach to the scheduling of bag-of-tasks applications on heterogeneous clouds with multi-level arrivals and critical jobs. *Simulation Modelling Practice and Theory*, 57 (Supplement C):1 – 25, 2015a.

I. A. Moschakis and H. D. Karatza. Multi-criteria scheduling of bag-of-tasks applications on heterogeneous interlinked clouds with simulated annealing. *Journal of Systems and Software*, 101(Supplement C):1 – 14, 2015b.

V. Petrov. *Sums of independent random variables*, volume 82. Springer Science & Business Media, 2012.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2018. URL `https://www.R-project.org/`.

S. Ristov, R. Mathá, and R. Prodan. Analysing the Performance Instability Correlation with Various Workflow and Cloud Parameters. In *Proc. of 25th Euromicro Int. Conf. on Parallel, Distributed and Network-based Processing - PDP*, pages 446–453, 2017.

M.A. Rodriguez and R. Buyya. Deadline Based Resource Provisioning and Scheduling Algorithm for Scientific Workflows on Clouds. *IEEE Transactions on Cloud Computing*, 2(2):222–235, 2014.

P Ruckdeschel and M. Kohl. General Purpose Convolution Algorithm in S4 Classes by Means of FFT. *Journal of Statistical Software*, 59(4):1–25, 2014.

A. Ruiz-Alvarez, I. K. Kim, and M. Humphrey. Toward Optimal Resource Provisioning for Cloud MapReduce and Hybrid Cloud Applications. In *Proc. of the IEEE 8th Int. Conf. on Cloud Computing*, pages 669–677, 2015.

J. Schad, J. Dittrich, and J.-A. Quiané-Ruiz. Runtime Measurements in the Cloud: Observing, Analyzing, and Reducing Variance. *Proc. VLDB Endow.*, 3(1-2): 460–471, 2010.

X. Tang, X. Li, and Z. Fu. Budget-constraint stochastic task scheduling on heterogeneous cloud systems. *Concurrency and Computation: Practice and Experience*, 29(19), 2017.

A. Tchernykh, U. Schwiegelsohn, E. Talbi, and M. Babenko. Towards understanding uncertainty in cloud computing with risks of confidentiality, integrity, and availability. *Journal of Computational Science*, 2016.

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)*, pages 228–235. IEEE, 2010.

R. Van den Bossche, K. Vanmechelen, and J. Broeckhove. Online cost-efficient scheduling of deadline-constrained workloads on hybrid clouds. *Future Generation Computer Systems*, 29(4):973–985, 2013.

F. Wu, Q. Wu, and Y. Tan. Workflow scheduling in cloud: a survey. *The Journal of Supercomputing*, 71(9):3373–3418, 2015.

Z.-H. Zhan, X.-F. Liu, Y.-J. Gong, J. Zhang, H. S.-H. Chung, and Y. Li. Cloud Computing Resource Scheduling and a Survey of Its Evolutionary Approaches. *ACM Computing Surveys*, 47(4):63:1–63:33, 2015.