Towards to Virtual Infrastructure Allocation on Multiple IaaS Providers with Survivability and Reliability Requirements

Anderson Schwede Raugust

Graduate Program in Applied Computing, Santa Catarina State University
Joinville, SC, Brazil E-mail: anderson.raugust@edu.udesc.br

Wilton Jaciel Loch

Department of Computer Science, Santa Catarina State University
Joinville, SC, Brazil E-mail: wilton.loch@edu.udesc.br

Felipe Rodrigo de Souza

École Normale Supérieure de Lyon, University of Lyon, INRIA Avalon
Lyon, France E-mail: felipe-rodrigo.de-souza@ens-lyon.fr

Maurício Aronne Pillon

Graduate Program in Applied Computing, Santa Catarina State University
Joinville, SC, Brazil E-mail: mauricio.pillon@udesc.br

Charles Christian Miers

Graduate Program in Applied Computing, Santa Catarina State University
Joinville, SC, Brazil E-mail: charles.miers@udesc.br

Guilherme Piêgas Koslovski

Graduate Program in Applied Computing, Santa Catarina State University
Joinville, SC, Brazil E-mail: guilherme.koslovski@udesc.br

Abstract: The diversity of services, prices, and geographical footprints have turned the clouds into a complex and heterogeneous environment. Moreover, the survivability and reliability aspects are often disregarded by tenants, eventually resulting in heavy losses due to unavailability of services that are hosted on Virtual Infrastructures (VIs). We present an alternative to improve VIs survivability and reliability, which considers the use of replicas and the spreading of virtual resources atop providers, regions, and zones. We formulate the VI allocation with survivability and reliability requirements as a Mixed Integer Program, and three strategies to solve the formulation are proposed. First, the binary constraints are relaxed to obtain a Linear Program (LP), and the LP solution is given as input for the Simulated Annealing technique. Complementary, two GPU-accelerated algorithms are proposed to speedup the allocation of large-scale scenarios. Simulation with different reliability requests indicate an increasing in survivability without inflating costs.

Keywords: IaaS; Allocation; Reliability; Survivability; Availability; Virtual Infrastructure; Providers.

1 Introduction

The Infrastructure-as-a-Service (IaaS) cloud providers offer Virtual Infrastructures (VIs) following the pay-as-you-go model, in which tenants are charged for the Virtual Machines (VMs) flavors and network requirements [1]. Several services are hosted by VIs and an eventual unavailability can affect users on different parts of the world. Although providers comply with strict administrative actions in their Data Centers (DCs) and divulge Service Level Agreement (SLA) availability figures (*e.g.*, 99.95%), their efforts may not be enough for critical applications. Indeed, the outage of VIs can induce financial loss to several companies. For instance, when Amazon EC2 had a 20-hours outage, VIs hosting services like Netflix, Instagram, and Pinterest were impacted (*e.g.*, unavailable services, and slow access) affecting millions of users [2]. Recently, another 4-hours outage, affected services like Github, Trello, Giphy, Medium, and Slack [3]. In these cases, cloud tenants just receive credits to re-launch their VIs.

Thus, SLAs based only on up-time availability metrics are insufficient for VIs hosting critical applications. Although the VI-hosted service may be available, the delivered information may be inaccurate. Indeed, reliability and survivability are essential goals as unplanned DC outages are fairly common, specially in the network control and management plans [4]. Reliability accounts for the probability that a VI is operating properly, while VI survivability indicates the ability to remain operational in the occurrence of cloud provider outages. Both are more precise metrics when compared to up-time availability.

An intuitive technique to increase reliability is the use of VM replicas ready to take over the operation in case of failure [5]. However, replicating each VM of the VI twice, at least, the provisioning cost could be prohibitive for most tenants. One approach to decrease the cost is to reduce the spectrum of replicas to only critical instances, those vital for running the service [6]. Consequently, the temporary unavailability of non-critical VMs is tolerable. Complementary, an approach to increase VI survivability is the spreading of VMs atop multiple resources, decreasing the probability of total failure [7, 8]. Although intuitive, both approaches require the analysis of data from multiple providers, regions, and zones.

In this context, we present an alternative to improve VI reliability and survivability based on replicas (for critical components) and controlled virtual resources spreading atop providers. Both techniques are agnostic to hosted applications requirements, and to internal high availability mechanisms. Among the management tasks performed for provisioning a VI, our proposal acts as a cloud broker on allocation of IaaS providers, guided by the tenant's perspective. We make four main contributions in this paper:

1. We formulate the VI allocation in multi-cloud providers as a Mixed Integer Program (MIP). Our formulation considers the survivability, reliability, and cost aspects. Moreover, the MIP comprises regular and critical VMs as well as data transfer requirements.

2. MIP constraints are relaxed to obtain a Linear Program (LP), and the simulated annealing technique is applied to find an acceptable solution, composing a strategy termed Reliable and Survivable Virtual Infrastructure Allocation (RS-VIA) based on Simulated Annealing (SA).

3. Graphics Processing Unit (GPU)-tailored algorithms are described to speedup the allocation. The algorithms use clustering techniques and perform controlled spreading of VMs atop zones.

4. Simulation results based on SLA data from public cloud providers are analyzed demonstrating the proposal's applicability.

This paper is organized as follows. Section 2 outlines the motivation and challenges on VI allocation atop multiple IaaS providers and formulates the problem. Section 3 details the proposed MIP, while the proposed CPU- and GPU-based strategies are described in Section 4. Simulation results are presented in Section 5, and related work is reviewed in Section 6. Final considerations and future works are discussed in Section 7.

2 Motivation and Problem Formulation

A VI is a set of VMs and network resources provisioned following the tenant's requirements [9]. On a single cloud provider scenario, a tenant selects the target provider and submits a VI request indicating the VMs configuration and the SLA specification. The cloud provider relies on on-line algorithms to allocate physical servers and links for hosting the VI request [10, 11, 12, 13]. Usually, the provider aims to maximizing profit, decreasing cost, and increasing Quality-of-Service (QoS) [10, 14].

There are two technical barriers on tenant's perspective. First, and foremost, the tenants often lack of technical knowledge to select the appropriated provider and to manage the VI. Second, the cloud-internal allocation process is a provider-oriented algorithm [15]. In this sense, the present work plays the role of a cloud broker using public providers data to assist tenants on survivable and reliable VI provisioning.

Concerning to the tenant's perspective, QoS and cost-related goals are recurring aspects [16]. The former is addressed by selecting VMs and services based on previously defined flavors, while for the latter, the pay-as-you-go model avoid over-provisioning costs that commonly happens in private and dedicated DCs. Survivability and reliability are QoS requirements that can impact on management and operational costs.

Although cloud providers inform the up-time availability of IaaS services on the SLA establishment, nothing is accounted on reliability and correctness of the service hosted by VIs. There are cases where one hour of downtime can result in million-dollar losses [2]. However, tenants do not have direct access to the cloud DC and rely on providers services to minimize the impact of outages, or may have to implement application-level solutions [17, 8]. In this sense, we propose a cost-effective, survivable, and reliable allocation of a VI atop multiple IaaS providers. Table 1 summarizes the notation used along this paper.

2.1 VI Requests and IaaS Providers

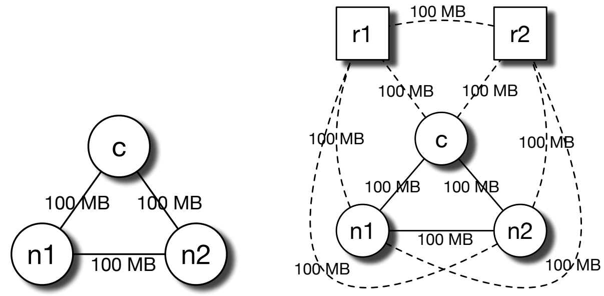
A tenant must identify the critical components, and the target reliability level on the SLA establishment,

Notation	Description
$P(R, Z)$	IaaS provider comprising a set of regions (R), and zones (Z).
$j \in R_i$	A region j from the provider i.
$k \in Z_{ij}$	A zone k from the region j and provider i.
$VI(N, D, V, c)$	A VI composed of N VMs, $D \subset N$ critical VMs, a traffic matrix (V), and the target reliability level (c).
$n \in N$	A regular VM n.
$m \in D \subset N$	A critical VM m.
$l_{nm} \in V$	A virtual link between VMs n and m. Each link requests a data volume to be transferred v_{nm}.
B	Set of replicas for the worst-case failure scenario.
$M(i, j, k, c, s)$	Number of replicas for supporting a reliability level c with s critical VMs on provider i, region j, and zone k.
$C(i, j, n)$	Cost for hosting VM n on provider i, region j.
$C_v(z, k)$	Cost for data transfers between zones z and k, accounted even for different providers.
x_{nijk}	VM n mapping on provider i, region j, and zone k (binary).
b_{nijk}	Replica b mapping on provider i, region j, and zone k (binary).
xl_{nmzk}	Virtual links (nm) to zones (z and k) mapping matrix (binary).
bl_{nmzk}	Backup links (nm) to zones (z and k) mapping matrix (binary).
y_i^p	Number of VMs hosted by provider i (integer).
y_{ij}^r	Number of VMs hosted by provider i, and region j (integer).
y_{ijk}^z	Number of VMs hosted by provider i, regions j, and zone k.

Table 1 Notation to represent VI requests, IaaS providers, and the fundamental system details.

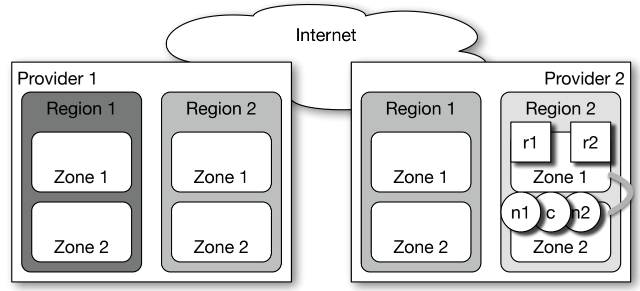
to request a reliable and survivable VI [5, 6]. In this sense, a VI request comprises two set of VMs, termed: regular, and critical. The failure of a regular VM is not severe for the VI-hosted service performance, while the failure of a critical one can fully interrupt the service. Formally, a VI request is represented by $VI(N, D, V, c)$, where N is the set of VMs, $D \subset N$ represents the critical VMs, V denotes the virtual links between VMs (each link has a data transfer request, v_{nm}). The target reliability is given by c (*i.e.*, 99.995%). The VI request must be allocated atop a single or multiple cloud providers. Each IaaS provider is represented by $P(R, Z)$, in which servers are organized in regions (R), and zones (Z).

Figures 1 and 2 resume the examples scenario we used along this paper. A request for a reliability level $c =$ 99.995%, one critical VM, and two regular VMs ($n1$ and $n2$) is submitted (Figure 1(a)), with the data transfer request for each VMs pair (100MB).

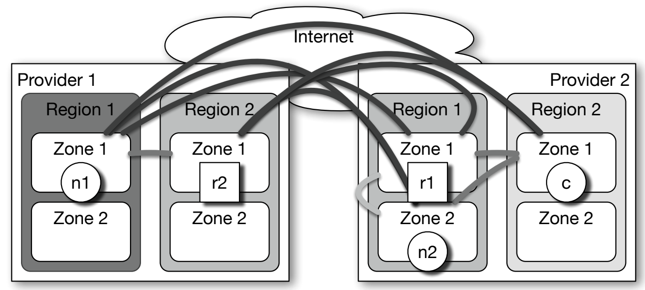


(a) VI request. (b) VI and replicas.

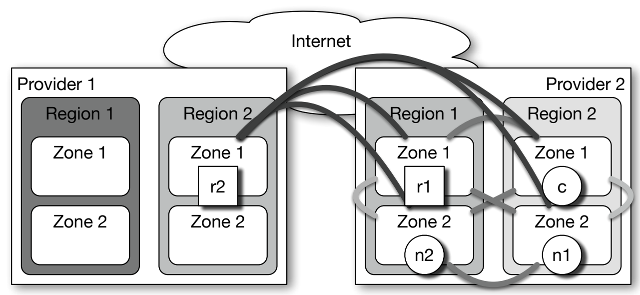
Figure 1 VI allocation with target reliability $c = 99.995\%$.



(a) Minimization of VI allocation cost.



(b) Spreading VI resources atop providers.



(c) Cost-effective, reliable, and survivable allocation.

Figure 2 VI allocation with target reliability $c = 99.995\%$ and 3 groups of failure (providers, regions, and zones). The lighter is the color, then lowest is the allocation cost.

2.2 *Probability of Failure*

There is usually a sequence of events which may result on failures. Initially, a fault activation causes an error that is propagated to a failure [18]. The failure of a subsystem can cause a fault in other system that interact with it, following the propagation chain. Such failures may happen in servers and network resources (*e.g.*, switches, and routers). Logs and data on DC failures are useful to understand and reduce the probability of future events [4]. However, raw data is privately accounted and confidentially kept. Cloud tenants are just aware of availability figures specified during the

SLA establishment. Precise information on Mean Time Between Failures (MTBF), Mean Time To Repair (MTTR), and Mean Time Between Outages (MTBO) are usually not shared.

Limited by the confidentially barrier on MTBF, MTTR, and MTBO figures, a tenant must rely on approximations to improve the VI configuration, specifically for identifying the number of VM replicas. When not available, the probability of failures and the reliability numbers can be inferred based on previous outages. For instance, the MTBF can be roughly deduced for the last 30 days as $\frac{720-\sum \text{duration of outage}}{\#\text{outages}}$, for a one-hour window. The probability of failures (p) is given by $\frac{1}{\text{MTBF}}$. Finally, the reliability is given as $1-p$.

Widespread measures of IaaS providers availability and outages (30 days period) are accounted by external services, such as CloudHarmony (`https://cloudharmony.com/`). For instance, on April 2017, CloudHarmony identified an availability of 99.997% for *ap-northeast-2* region of Amazon EC2 provider, and 99.809% for *ams-e* region of ElasticHosts provider. The latter had a higher number of outages over the analyzed period. In this way, the reliability is approximately defined as 97.495% and 99.861% for ElasticHosts and Amazon, respectively. It is worthwhile to highlight that p is an approximation. Any mechanism capable of offering a more precise probability can be applied. Moreover, the probability represents an independent failure (crash) which may affect a single resource (*e.g.*, a server) or a group of resources (*e.g.*, zone, and region). In this sense, it is evidenced that the spreading of VMs and replicas across different failure groups is beneficial to decrease the probability of total failure, consequently increasing the VI survivability [7, 8].

2.3 Defining Replicas for Critical VMs

The use of replicas is a promising approach for fulfilling the reliability gap between the providers and the VI requirement [17, 6]. Initially, the critical VMs ($D \subset N$) and the target reliability (c) for VI are requested. Afterwards, this information is combined with the providers probabilities of failure (zones) to apply the Opportunistic Redundancy Pooling (ORP) technique [5]. ORP uses an incomplete regularized beta function, $I_{1-p} = (n, k+1)$, where n is the number of critical VMs ($|D|$), $k+1$ is the number of required replicas, and $1-p$ is the zone reliability level. Therefore, the number of replicas is the smallest number which guarantees c. Formally, the number of replicas required for achieving c with D critical VMs is per zone calculated and represented by M.

Figure 3 exemplifies the composition of M. Using ORP, a range of critical nodes supported by k replicas is identified. Given the number of critical VMs (x-axis), the number of replicas (y-axis) is computed for providers using distinct failure figures. Based on April/2017 data collected by CloudHarmony, the VPS.NET (Atlanta) region has a low probability of failure ($p = 0.001$), and consequently, with a target reliability c is 99.995%, only 4 replicas are need to support between 74 and 241 critical VMs. M is indexed by zone, target reliability c, and the number of critical VMs.

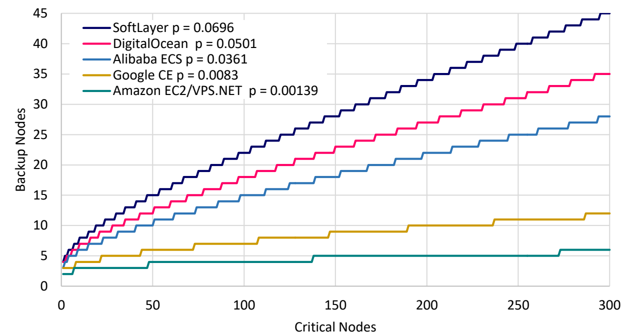


Figure 3 Number of replicas required to support $c = 99.995\%$ for providers with different p.

Figure 1(b) exemplifies the extension of a VI request (from Figure 1(a)) adding replicas and links. For this example, 2 replicas are arbitrarily added ($r1$ and $r2$) to achieve the target reliability c atop both providers. The dashed lines represent the new virtual links required to delivery connectivity in the occurrence of a failure.

2.4 Allocating IaaS providers to host VIs

f

The VI requests are individually analyzed by the mechanism characterizing an on-line allocation problem [10]. The mapping of VMs to zones is given by $\mathcal{M} : N \mapsto Z$. The internal provider allocation policy is out of scope on this paper. We argue providers selection is a tenant's choice, consequently performed considering the tenant's perspective, while intra DC allocation algorithms [10, 11, 5, 12, 13, 15] are arbitrarily defined by the provider. In addition, the broker service can be executed any time for accomplishing with new probability of failures. However, reallocation and migration mechanisms are not discussed and indicated as future work.

Regarding to the tenant's perspective, the allocation goal is the cost-effective selection of providers guided by survivability and reliability requirements. The first dimension aims at minimizing the VMs (regular, critical and replicas) and networking (data transfer between VMs) provisioning costs, while the second aims at minimizing the impact of providers failures on the VI [8, 7].

Three examples of VI allocation are presented on Figure 2. For differentiating prices between providers and regions (as commonly performed by public clouds), a color scale is used. The lighter is the color, the low-cost is the allocation. The same approach is applied for lines on virtual links. Initially, Figure 2(a) exemplifies an allocation decreasing the VI provisioning cost. All resources are placed on two zones from a single region. Thus, besides of decreasing the VMs provisioning cost, the allocation softened the communication costs as data transfers inside a zone are not charged (white color).

A survivable-only solution is presented by Figure 2(b). VMs are spread atop 5 zones, 4 regions, and 2 providers, ignoring the provisioning cost. Indeed, data transfer between providers must be performed. Based on this allocation, the probability of a total VI failure is minimized for all failure groups (providers, regions, and zones).

The focus of our work is to identify an intermediate approach, as given by Figure 2(c). The allocation still used 2 providers and 5 zones, but reduced the number of regions to 3, motivated by the allocation cost: region 1 from provider 1 was ignored due to the high price. In addition, the number of virtual links communicating over the Internet was reduced. Finally, it is noted the reliability level c was achieved for all scenarios by adding the replicas.

3 Exact MIP for Allocating Survivable and Reliable VIs

To study and analyze the IaaS allocation for hosting survivable and reliable VIs, a formalization based on MIP is detailed. In order to elucidate the MIP, the Figures 1 and 2 are used to exemplify the equations.

3.1 Variables and Objective

Four variables are used to identify which providers must host a given VI request (*e.g.*, Figure 1(b)). Initially, x_{nijk}, a binary variable, indicates the mapping of regular and critical VMs ($n \in N$) on provider i, region j, and zone k. For applying the same rationale to replicas, the set B must be defined. However, the exact number of replicas depends on which providers, regions and zones will be selected to host the critical VMs, and such information is unknown in advance. On the survivability perspective, B represents the worst-case scenario where the zone selected for hosting critical VMs has the highest probability of failure. However, the model aims at minimizing the number of replicas need for guaranteeing the requested reliability level. The allocation of a replica is indicated by the binary variable b_{nijk} ($n \in B$).

Thus, for data transfer between VMs, two variables are used to define the allocation of virtual links, xl and bl. The former represents the allocation of a virtual link l_{mn} between VMs n and m, while the second follows the rationale to replicas. The source n of a l_{nm} link is mapped to the corresponding zone that is hosting n, while the target m is mapped to destination zone. For regular and critical VMs, l_{nm} are known in advance, while for connectivity to replicas, they are quantified on-the-fly. In this sense, all possible connections between N (regular and critical VMs) and B (replicas) are analyzed by bl. Moreover, the connectivity between replicas ($B \times B$) is also accounted. However, just those need (according to b) are effectively allocated.

3.1.1 VI Allocation Cost

IaaS providers apply different cost models for VMs, usually differentiated by regions. In this sense, function $C(i,j,n)$ returns the cost for hosting a VM n on provider i, region j, and Equations (1) and (2) account the costs for hosting all VMs and the dynamically defined replicas, respectively.

$$C_{vm}(VI) = \sum_{n \in N} \sum_{i \in P} \sum_{j \in R_i} \sum_{k \in Z_{ij}} x_{nijk} \times C(i,j,n) \quad (1)$$

$$C_{vmb}(VI) = \sum_{w \in B} \sum_{i \in P} \sum_{j \in R_i} \sum_{k \in Z_{ij}} b_{wijk} \times C(i,j,w) \quad (2)$$

The costs for data transfer between VMs are given by Equation (3) (regular and critical) and Equation (4) (replicas). As commonly applied by public cloud providers, the data transfer cost is differentiated for zones, regions, and providers. This information is abstracted by $C_v(z,k)$, with informs the per MB price for transferring data between zones z and k (even between different providers).

$$C_{net}(VI) = \sum_{l_{nm} \in V} \sum_{i_s \in P} \sum_{j_s \in R_{i_s}} \sum_{z \in Z_{i_s j_s}} \sum_{i_t \in P} \sum_{j_t \in R_{i_t}} \sum_{k \in Z_{i_t j_t}} xl_{nmzk} \times v_{nm} \times C_v(z,k) \quad (3)$$

$$C_{netb}(VI) = \sum_{l_{nm} \in N \times B} \sum_{i_s \in P} \sum_{j_s \in R_{i_s}} \sum_{z \in Z_{i_s j_s}} \sum_{i_t \in P} \sum_{j_t \in R_{i_t}} \sum_{k \in Z_{i_t j_t}} (bl_{nmzk} \times v_{nm} \times C_v(z,k)) + \sum_{l_{nm} \in B \times B} \sum_{i_s \in P} \sum_{j_s \in R_{i_s}} \sum_{z \in Z_{i_s j_s}} \sum_{i_t \in P} \sum_{j_t \in R_{i_t}} \sum_{k \in Z_{i_t j_t}} (bl_{nmzk} \times v_{nm} \times C_v(z,k)) \quad (4)$$

Finally, the total cost for allocating a VI is given by Equation (5). The weight's vector α is used to denote the importance level of each component.

$$C_{total}(VI) = \alpha_{vm} C_{vm}(VI) + \alpha_{vmb} C_{vmb}(VI) + \alpha_{net} C_{net}(VI) + \alpha_{netb} C_{netb}(VI) \quad (5)$$

3.1.2 Impact of IaaS Providers Failures

An intuitive approach to decrease the impact of failures on VI-hosted applications is the spreading of virtual resources atop multiple domains of failures [7, 17, 8]. In our context, a domain of failure is a provider, region, or zone. A zone represents the smallest unit, consequently with the highest probability of failure. The remaining domains aggregate zones (or regions) and soften the probability. In short, in tenant's perspective, the larger the spreading of virtual resources, the lower the probability that a failure can cause an outage on VI-hosted service. Formally, three integer variables are used to represent the use of providers, regions and zones, y_i^p, y_{ij}^r e y_{ijk}^z, respectively. Equations (6)-(8) account the number of VMs hosted by providers, regions, and zones.

$$y_i^p = \sum_{j \in R_i} \sum_{k \in Z_{ij}} \left(\sum_{n \in N} x_{nijk} + \sum_{w \in B} b_{wijk} \right) \quad \forall i \in P \tag{6}$$

$$y_{ij}^r = \sum_{k \in Z_{ij}} \left(\sum_{n \in N} x_{nijk} + \sum_{w \in B} b_{wijk} \right) \quad \forall i \in P; \forall j \in R_i \tag{7}$$

$$y_{ijk}^z = \sum_{n \in N} x_{nijk} + \sum_{w \in B} b_{wijk} \quad \forall i \in P; \forall j \in R_i; \forall k \in Z_{ij} \tag{8}$$

For spreading the VMs and replicas atop failure groups, three integer (positive) variables (Equations (9)-(11)) are applied for compositing the minimization (min. $I(VI)$, Equation (15)). All three variables maximize the distribution atop failure groups (providers, regions, and zones) respecting the number of VMs and replicas (Equations (12)-(14)). The weight's vector β differentiates the importance of each component.

$$I^p \geq y_i^p; \forall i \in P \tag{9}$$

$$I^r \geq y_{ij}^r; \forall i \in P; \forall j \in R_i \tag{10}$$

$$I^z \geq y_{ijk}^z; \forall i \in P; \forall j \in R_i; \forall k \in Z_{ij} \tag{11}$$

$$I^p \leq |N| + |B|; \forall i \in P \tag{12}$$

$$I^r \leq |N| + |B|; \forall i \in P; \forall j \in R_i \tag{13}$$

$$I^z \leq |N| + |B|; \forall i \in P; \forall j \in R_i; \forall k \in Z_{ij} \tag{14}$$

$$I(VI) = \beta_p I^p + \beta_r I^r + \beta_z I^z \tag{15}$$

3.1.3 Objective Function

The minimization of Equation (16) results on lowest allocation cost and decreases the impact caused by a failure. The first term is normalized by the cost for hosting on the costly zone ($C_{max}(VI)$), while the second term is normalized by the number of VMs and replicas.

$$min : \frac{C_{total}(VI)}{C_{max}(VI)} + \frac{I(VI)}{|N| + |B|} \tag{16}$$

3.2 Constraints

For guaranteeing the SLA QoS, a set of capacity, data transfer, meta and binary constraints must be satisfied.

$$\sum_{i \in P} \sum_{j \in R_i} \sum_{k \in Z_{ij}} x_{nijk} = 1; \forall n \in N \tag{17}$$

$$\sum_{i \in P} \sum_{j \in R_i} \sum_{k \in Z_{ij}} b_{nijk} \leq 1; \forall n \in B \tag{18}$$

$$\sum_{w \in B} \sum_{i \in P} \sum_{j \in R_i} \sum_{k \in Z_{ij}} b_{wijk} \geq min(M) \tag{19}$$

$$\sum_{w \in B} \sum_{i \in P} \sum_{j \in R_i} \sum_{k \in Z_{ij}} b_{wijk} \leq |B| \tag{20}$$

$$\sum_{q \in Z_{st}} xl_{nmkq} + \sum_{q \in Z_{st}} xl_{nmzq} = x_{nijk} + x_{mijk} \quad i \in P, j \in R_i, k \in Z_{ij}, s \in P, t \in R_s, l_{nm} \in V \tag{21}$$

$$\sum_{q \in Z_{st}} bl_{nmkq} + \sum_{q \in Z_{st}} bl_{nmzq} = x_{nijk} + b_{mijk} \quad i \in P, j \in R_i, k \in Z_{ij}, s \in P, t \in R_s, l_{nm} \in N \times B \tag{22}$$

$$\sum_{q \in Z_{st}} bl_{nmkq} + \sum_{q \in Z_{st}} bl_{nmzq} = b_{nijk} + b_{mijk} \quad i \in P, j \in R_i, k \in Z_{ij}, s \in P, t \in R_s, l_{nm} \in B \times B \tag{23}$$

$$\sum_{k \in Z_{ij}} \sum_{q \in Z_{st}} xl_{nmkq} = 1 \quad i \in P, j \in R_i, s \in P, t \in R_s, l_{nm} \in V \tag{24}$$

$$\sum_{k \in Z_{ij}} \sum_{q \in Z_{st}} bl_{nmkq} \leq 1 \quad i \in P, j \in R_i, s \in P, t \in R_s, l_{nm} \in N \times B \tag{25}$$

$$\sum_{k \in Z_{ij}} \sum_{q \in Z_{st}} bl_{nmkq} \leq 1 \quad i \in P, j \in R_i, s \in P, t \in R_s, l_{nm} \in B \times B \tag{26}$$

Constraints (17) and (18) indicate VMs and replicas, respectively, must be allocated at most one time. The minimum number of replicas indicated by ORP is guaranteed by Equation (19), while the upper-bound limit is the allocation on the zone with highest failure (Equation (20)). Constraints (21)-(23) ensure virtual links V are hosted by zones hosting source and destination [19]. Finally, Equations (24)-(26) guarantee that virtual links are hosted at most one time.

4 Allocation Strategies

Solving a MIP is known to be computationally infeasible. Thus, we propose a set of techniques to overcome the dimensionality barrier and to approximate the allocation from the exact MIP mapping. The strategies are depicted by Figure 4. In addition to the exact MIP allocation (from Section 3), three strategies are proposed.

- RS-VIA based on SA: The strategy is tailored for CPU execution. In short, we relax the binary constraints obtaining a LP. Latter, the approximated result is interpreted and used as input for a simulated annealing technique.

- RS-VIA-GPU-Global: The approach is GPU-accelerated and performs an exhaustive search guided by allocation cost. In parallel, a controlled spreading of virtual resources atop providers, regions, and zones is performed.

- RS-VIA-GPU-FW: The strategy is also GPU-accelerated and executes a set of worst-fit allocation atop groups of resources. In short, providers are grouped by the probability of failure, reducing the search space.

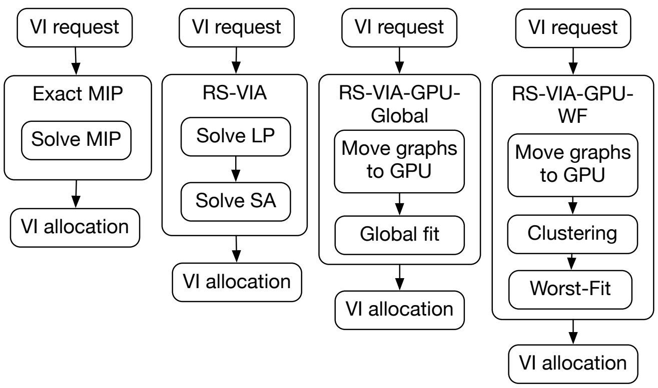


Figure 4 The allocation strategies discussed along the paper.

The strategies are detailed in Sections 4.1 and 4.2.

4.1 RS-VIA based on SA

Initially, for obtaining an LP, the binary constraints of variables x, b, xl, and bl are relaxed (≥ 0, ≤ 1, $\in \mathbb{R}$). Previous work applied deterministic and random rounding techniques to interpret the LP results [20]. Although efficient for physical resources allocation for hosting virtual networks, the techniques are not suitable for multiple providers selection. In this work, we propose the use of SA for interpreting the LP (Algorithm 1).

The same set of constraints and the objective function were implemented as a SA algorithm. The SA algorithm (Algorithm 1) receives as input the VI request, two parameters (T and α) for controlling the SA execution, the LP results given by relaxed variables x and b, and the set of providers. While the annealing criteria holds (T, lines 4 and 29), the SA shuffles the VMs sets N and B for composing an initial solution (lines 5 and 6). For each VM a candidate is chosen based on LP values. The set of candidates, termed *cand*, is composed of all possible candidates previously identified by the LP ($x > 0, b > 0$). Rather of composing *cand* only based on LP [20], RS-VIA accounts the networking impact analyzing the previous mapping ($\mathcal{M}$) on lines 10 and 12. Preference is given to candidates with high p_k as the network cost may be reduced (line 17). After placing all VMs, the objective function is accounted (line 25) and stored if improves the previous one (lines 26 to 28). Latter, a suitable solution or an empty mapping ($\mathcal{M}$) is returned.

```
   Input: VI, x, b, T, α, P
   Output: M; VI to zones mapping
1  T = C_max(VI)
2  obj_best = T
3  sol = ∅; M = ∅
4  while T ≥ 1 do
5      shuffle(N)
6      shuffle(B)
7      for n ∈ N ∪ B do
8          for i ∈ P, j ∈ R_i, k ∈ Z_ij do
9              if n ∈ N then
10                 p_k = x_nijk × Σ_{l_nm ∈ V} v_nm; m ∈ M
11             else
12                 p_k = b_nijk × Σ_{bl_nm ∈ V} v_nm; m ∈ M
13             end
14         end
15         cand = ∅
16         for i ∈ P, j ∈ R_i, k ∈ Z_ij do
17             s = ⌈ p_k / min(p) ⌉
18             for range(1, s) do
19                 cand.add(w)
20             end
21         end
22         c = rand(cand)
23         sol ← [c, z]
24     end
25     obj = Eq. 16
26     if obj ≤ obj_best then
27         obj_best = obj
28         M = sol
29     T = T × (1 − α)
30 end
31 return M
```

Algorithm 1: RS-VIA based on simulated annealing.

4.2 GPU-Accelerated Algorithms

The optimal embedding of a VI is NP-hard. Although RS-VIA based on SA founds efficient solutions, the applicability with large-scale data (providers, regions, zones and VI requests) remains a challenge due the number of comparisons that must be performed on CPU. Moreover, for cloud computing environments the majority of VI allocation requests are time critical as delay in processing an allocation or reallocation can compromise the performance of hosted applications. In this context, the high-performance of GPUs make them potential candidates to overcome CPU limitations and support the allocation of survivable and reliable VIs. GPUs offer a high-degree of parallelism, however, although GPUs are often faster than CPUs, the selection of suitable algorithms to speedup the allocation remained an open challenge [21].

In this sense, we propose two strategies for speeding up the survivable and reliable allocation of VIs based on GPU. The first one, termed RS-VIA-GPU-WF, combines the K-Means clustering algorithm [22] with worst-fit allocation. For decreasing the number of comparisons between servers and virtual resources, the K-Means is used for creating groups of zone based on probability of failure. Each group represents the aggregated capacity of its composing resources. In turn, the worst-fit approach tends to spread virtual resources atop the physical graph [21] following the survivability objective.

The RS-VIA-GPU-WF is detailed in Algorithm 2. The algorithm receives as input the VI request and the set of providers. Initially, for each pair of zones, the absolute difference between the probability of failures is accounted, and this information is used as input for composing the distance matrix MD (line 1). The rationale for this approach is to induce the clustering of zones with similar values. Clustering is performed with K-Means (line 2). As the number of replicas is unknown, RS-VIA-GPU-WF follows the same approach used for composing the set B for MIP formulation, initially identifying the set of replicas for the worst-case failure scenario (lines 3 and 4). The solution is iteratively refined for each group (lines 6 to 14). RS-VIA-GPU-WF performs a worst-fit allocation (line 7) considering only the resources from one group and accounts the number of replicas and allocation cost (lines 8 and 9). Whenever a cost-efficient solution is found it is selected by the algorithm. Finally, the VMs to zones mapping is returned (line 15).

Input: VI, P
Output: $\mathcal{M}$; VI to zones mapping
1 MD=Dist(M)
2 $groups$ = K-Means(P, MD)
3 $replicas_{best}$ = criticalZone(P, MD)
4 $cost_{best}$ = Eq. 5 considering the critical zone
5 $\mathcal{M} = \emptyset$
6 **for** $\underline{group \in groups}$ **do**
7     $\mathcal{M}_{aux}$ = allocateWorstFit(VI, $group$)
8     $replicas$ = criticalZone($P \in \mathcal{M}_{aux}$, MD)
9     $cost$ = Eq. 5 considering $P \in \mathcal{M}_{aux}$
10   **if** $\underline{cost \leq cost_{best}}$ **then**
11         $cost_{best} = cost$
12         $replicas_{best} = replicas$
13         $\mathcal{M} = \mathcal{M}_{aux}$
14 **end**
15 **return** $\underline{\mathcal{M}}$

Algorithm 2: RS-VIA-GPU-WF: GPU-accelerated RS-VIA with multiple worst-fit allocations.

The second GPU-accelerated strategy, termed RS-VIA-GPU-Global, is cost-oriented, as described by the Algorithm 3. The strategy performs an exhaustive search identifying a cost-efficient allocation for each VM. In addition, to adhere the survivability requirement, a upper-bound limit of VMs allocation per zone is defined, forcing a controlled spreading of virtual resources (the limit is represented by the $maxVMsPerZone$ parameter). In this sense, the algorithm receives a VI request, the set of providers, and the parameter to limit the number of VMs per zone. For each VM, critical or regular (line 2), the algorithm identifies a zone (line 4) that decreases the allocation cost (line 5). The limit of VMs per zone is accounted (line 6), where $|\mathcal{M}(k)|$ denotes the number of VMs allocated by zone k. When a map is identified, it is appended to VMs to zones mapping ($\mathcal{M}$, line 10) Finally, the map is returned by the algorithm.

It is worthwhile to mentation that both algorithms explore the parallelism provided by GPUs. RS-VIA-GPU-Global and RS-VIA-GPU-WF are compared with RS-VIA based on SA and exact MIP allocation on Section 5.

Input: VI, P, $maxVMsPerZone$
Output: $\mathcal{M}$; VI to zones mapping
1 $\mathcal{M} = \emptyset$
2 **for** $\underline{n \in N}$ **do**
3     $cost_{best} = 0$
4     $host = \emptyset$ **for** $\underline{i \in P, j \in R_i, k \in Z_{ij}}$ **do**
5         $cost = cost + Eq.$ 5 considering only Z_{ij} and VM n
6   **if** $\underline{cost \leq cost_{best} \wedge |\mathcal{M}(k)| \leq maxVMsPerZone}$ **then**
7             $cost_{best} = cost$
8             $host = k$
9   **end**
10     $\mathcal{M}$ append $n \mapsto k$
11 **end**
12 **return** $\underline{\mathcal{M}}$

Algorithm 3: RS-VIA-GPU-Global: GPU-accelerated RS-VIA with global allocation.

5 Evaluation and Analysis

As proof-of-concept, four versions of a cloud broker were developed. The exact MIP and RS-VIA based on SA were implemented in Java v1.8 using the IBM CPLEX optimizer (v12.6.1.0), and the simulation was executed on a desktop using processor AMD Phenom II X4 (4 cores), 4GB RAM, running GNU/Linux Ubuntu 14.04. The GPU-accelerated versions were implemented in C++ with CUDA 9.0.176, NVIDIA driver 384.81, and GCC 5, hosted by a machine Intel i7 2600K / 32GB RAM, NVIDIA Titan XP /12GB, running GNU/Linux Ubuntu 17.04 Server. The GPU code is based on a graph embedding framework [21].

5.1 Metrics

For representing the tenant's perspective, seven metrics were selected:

(i) Regular and critical VMs costs;

(ii) Cost of replicas;

(iii) Network cost – (Eq. (3)); and

(iv) Network cost between replicas – (Eq. (4)).

(v) – (vii) Number of zones, regions and providers; used for hosting VMs regular, critical, and replicas.

The cost metrics are normalized by the maximum cost, while the failure groups (zones, regions, and providers) are represented as the ratio related to the total group size.

5.2 Simulation Parameters

As discussed in Section 3, a set of parameters must be defined for guiding the MIP execution. In this sense, the probability of failures for each zone was extracted from the CloudHarmony platform (August/2017). In addition, for composing the data transfer cost function, prices were uniformly selected at three ranges:

(i) Data transfer between zones of the same region: between \$0.01 and \$0.05;

(ii) Data transfer between zones of the same provider, but different regions: between \$0.1 e \$0.5; and

(iii) Data transfer between different providers the price is selected between \$1.5 e \$2.0.

Moreover, when two VMs are communicating in the same zone no cost is charged. It is worthwhile to mention that any more accurate pricing scheme can be applied.

Regarding the VI requests, the Amazon EC2 popular instance type $m3.large$ were selected [23]. For composing the cost function $C(i, r, n)$, a similar configuration was selected for each IaaS cloud provider. All VMs were interconnected by a full mesh network, representing the worst-case communication scenario in terms of provisioning cost. Thus, transfer requests on all virtual links were defined as 500MB per month.

Finally, to configure the objective function, each element of β was set to 0.25, while 0.33 was used for γ. We analyze different weights for α due it impacts on cost, reliability and spreading of virtual resources.

5.3 Simulation Scenarios

For analyzing the efficiency of the proposed mechanisms, a set of simulation scenarios were defined representing the client's perspective. The scenarios are individually detailed in Subsections 5.3.1, 5.3.2 and 5.3.3, focusing on exact MIP, accuracy of RS-VIA based on SA, and RS-VIA scalability (CPU and GPU), respectively.

5.3.1 MIP

The main objective of this scenario is to analyze the performance of solving an exact MIP (described in Section 3) for allocating VIs with survivability, reliability and cost constraints. Due the combinatorial explosion, this simulation performed the allocation of a single VI request atop a restricted subset of providers. The well-know IaaS cloud providers Amazon EC2 and Google Computing Engine were selected, accounting 17 regions and 24 zones. Each VI request is composed by 5 regular and 5 critical VMs.

The simulation is further divided into two reliability requirement (the c parameter from a VI request - Table 1): 99.95%, and 99.995%. The number of replicas need to achieve the reliability target differs between zones, regions and providers (represented by M - described in Section 2.3).

In this context, we tackled five different approaches. The first one aims an efficient selection considering the minimization of total allocation cost (termed Cost-Only (CO), $\alpha = 1$) as depicted by Figure 2(a). With Survivable-Only (SO) ($\alpha = 0$), spreading is benefited, therefore this approach maximizes the spreading of virtual resources without a fixed allocation cost constraint (the example from Figure 2(b)). Finally, the exact allocation considers both objectives together (cost and survivability) as exemplified by Figure 2(c). Moreover, to carry out an depth analysis of the exact MIP, three values of α (0.25, 0.5, and 0.75) are discussed, pointing out distinct cost-survivability trade-offs.

5.3.2 Exact Allocation and RS-VIA based on SA

For the second scenario, the main goal is to verify the applicability of RS-VIA based on SA against the baseline results from the exact MIP allocation. In this sense, the same parameters from Subsection 5.3.1 are applied (subset of providers and VI request). In short, this simulation scenario compares four approaches:

(i) CO ($\alpha = 1$);

(ii) SO ($\alpha = 0$);

(iii) Exact Allocation (EA) ($\alpha = 0.5$); and

(iv) RS-VIA based on SA.

5.3.3 Scalability of RS-VIA based on SA and GPU-Accelerated Scenarios

The objective of the last scenario is to verify the scalability of RS-VIA based on SA with real data from multiple providers. Our simulation comprises 31 public IaaS cloud providers totaling 133 regions and 153 zones geographically distributed with a world-size footprint. Moreover, for representing clients with different requirements, 3 VI configurations were prepared varying the ratio of regular-critical VMs: $40 - 10$, $25 - 25$ and $10 - 40$, where the first term represents the regular VMs and the second one denotes the number of critical VMs.

As discussed in Section 6, the literature lacks on definitive solutions for allocation VIs with survivability and reliability requirements. Thus, the algorithm selected as baseline for comparison performs a naive random selection of potential candidates.

For each CPU-based algorithm (RS-VIA with SA and Random (RND) allocation), 10 rounds were executed and results are plotted as means with standard deviation. However, for GPU-tailored, a single execution was performed as the allocations performed by both approaches are deterministic (discussed in Section 4.2).

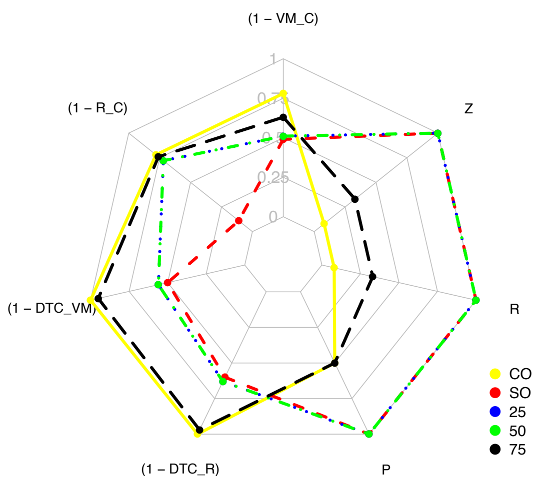
5.4 Results

The values for all metrics (from Section 5.1) are plotted with two kind of graphics; radar and bar plots. Bar plots results are side-by-side placed and analyzed. Radar graphics were adopted to observe and compare the area formed by all metrics. In this sense, to facilitate the analysis, the radar graphs plot $1-$ costs (critical / regular VMs, and links), and the survivability metrics are represented by number of providers, regions, and zones.

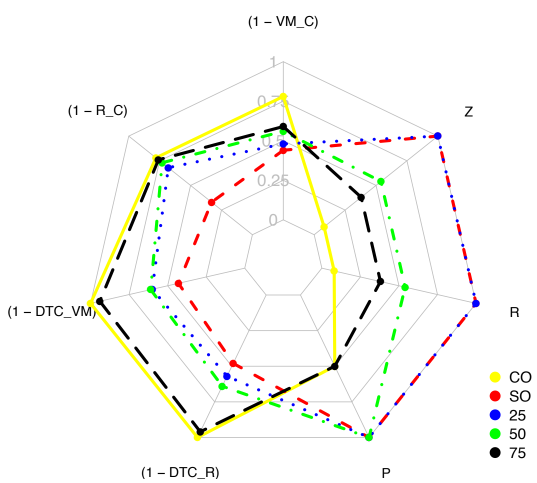
Costs for VMs, replicas, data transfer between VMs, and data transfer between replicas are represented by VM_C, R_C, DTC_VM, and DTC_R, respectively.

5.4.1 MIP Analysis

All MIP results are presented in Figure 5. Specifically, results for the reliability target $c = 99,95\%$ are plotted in Figure 5(a), while Figure 5(b) summarizes the results for $c = 99,995\%$. As expected, the CO approach filled the smallest area on radar plot, concentrating VMs in regions with lower cost. However, the consolidation negatively impacts on spreading atop providers, regions, and zones. Looking for lower cost, the CO approach keeps critical VMs and their corresponding replicas in the same zone. In the opposite sense, the SO approach filled the largest area, allocation more providers, regions and zones. In this case, the virtual resources are spread atop multiple domains of failure, however, this approach maximized all costs (VMs, replicas and communication).



(a) 99.95%



(b) 99.995%

Figure 5 Results for the exact MIP varying the α parameter.

After analyzing two contradictory approaches, the exact allocation uses the advantage of both, in one sense, the cost, and another, the spreading. This fact can be observed in Figure 5(a), where the best-case spreading is reached by the exact allocation configured with $\alpha = 0.25$ or $\alpha = 0.5$, with lower cost than SO. If the target is cost, the exact allocation with $\alpha = 0.75$ can get more spreading (24% more for regions and zones) than CO with almost the same allocation cost. Results with reliability of $99,995\%$ are depicted in Figure 5(b). The exact allocation with $\alpha = 0.25$ obtained the same spreading as the SO approach, however, with lower cost due to the reduction of replicas. In addition, the exact allocation with $\alpha = 0.75$ found the same cost allocation as CO, nevertheless, with more survivability due the spreading atop providers, regions, and zones. With $\alpha = 0.5$, the exact allocation keeps the same number of providers, but the cost is smoothed decreasing the number of region and zones. Finally, our exact MIP offers to tenants more flexibility (cost, survivability or reliability) by tuning the α parameter.

In this section, we demonstrate that the exact allocation provides an efficient survivability-cost trade-off. However, it is worthwhile to mention that its scalability limit (the NP-hard nature), and approximately two hundred minutes were need to perform a single exact allocation.

5.4.2 Exact MIP and RS-VIA based on SA

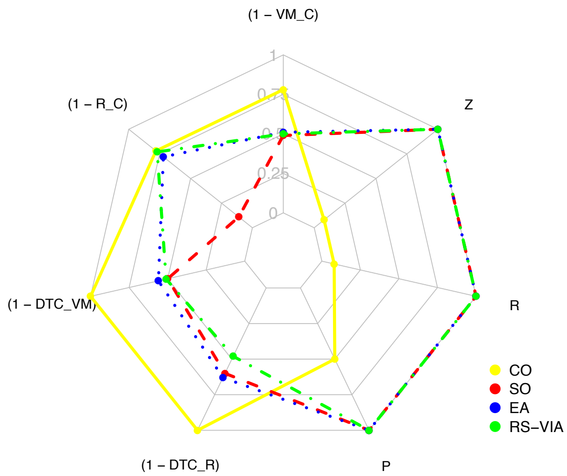
This section compares the performance of RS-VIA based on SA with the baseline provided by the exact MIP. Initially, it is important to observe that for the reliability requirement $c = 99.95\%$, RS-VIA based on SA obtained results close to the exact allocation (Figure 6(a)). Considering the spreading of virtual resources, both approaches have the same results, while analyzing the cost efficiency, the cost for replicas with RS-VIA based on SA is 21% less than the exact allocation. However, in others three metrics (VMs costs, communication cost between VMs, and communication cost between replicas), the exact allocation is more efficient.

In turn, results with target reliability $c = 99.995\%$ are depicted in Figure 6(b). In this case, RS-VIA with SA reaches reliability target using less providers than the exact allocation, reducing the network communication costs. In short, for the RS-VIA based on SA, the VMs and replicas communication costs are 23 and 25% lower than the exact MIP, respectively. Indeed, the data transfer costs for RS-VIA with SA (0.91 in both cases) are close to the Cost-Only approach.

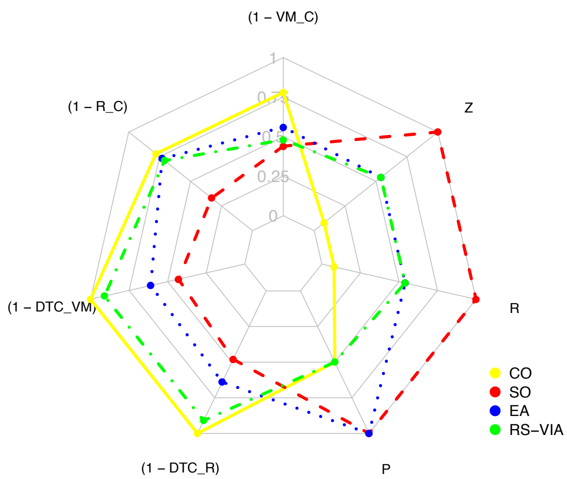
Unfortunately, although obtaining competitive results when compared to the exact MIP, the RS-VIA based on SA requires almost hundred minutes to solve the LP and to find an approximate solution with SA.

5.4.3 Scalability and GPU Acceleration

For discussing the application on large-scale scenarios, RS-VIA based on SA, RS-VIA-GPU-Global, and RS-VIA-GPU-WF are analyzed, investigating the allocation efficient when compared to naive random allocation. The simulation scenario consists of 31 providers, allocating requests higher than those discussed in Sections 5.4.1

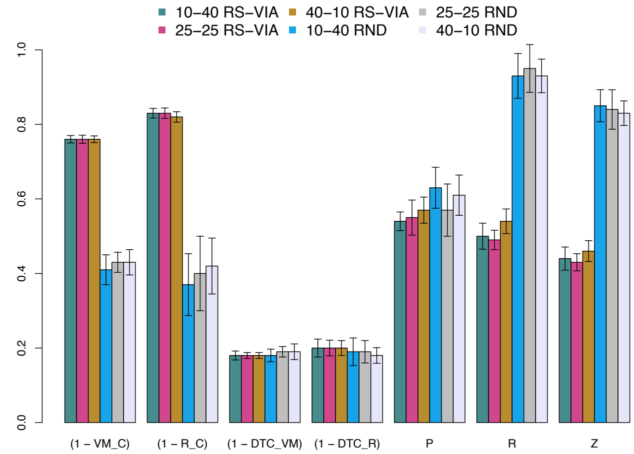


(a) 99.95%

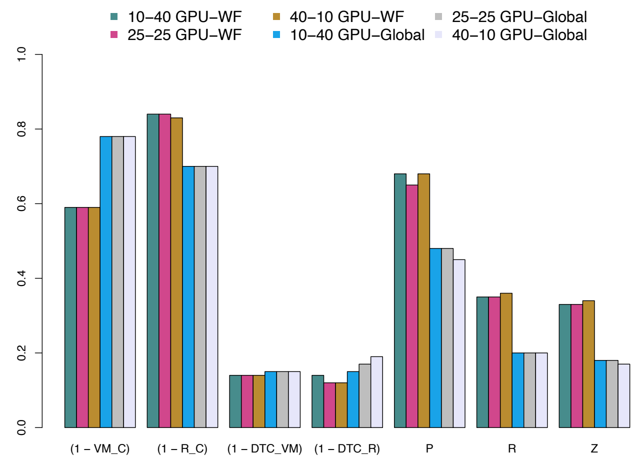


(b) 99.995%

Figure 6 Results with MIP and RS-VIA based on SA.



(a) RS-VIA without GPU.



(b) RS-VIA with GPU.

Figure 7 RS-VIA results to reliability target $c = 99.95\%$

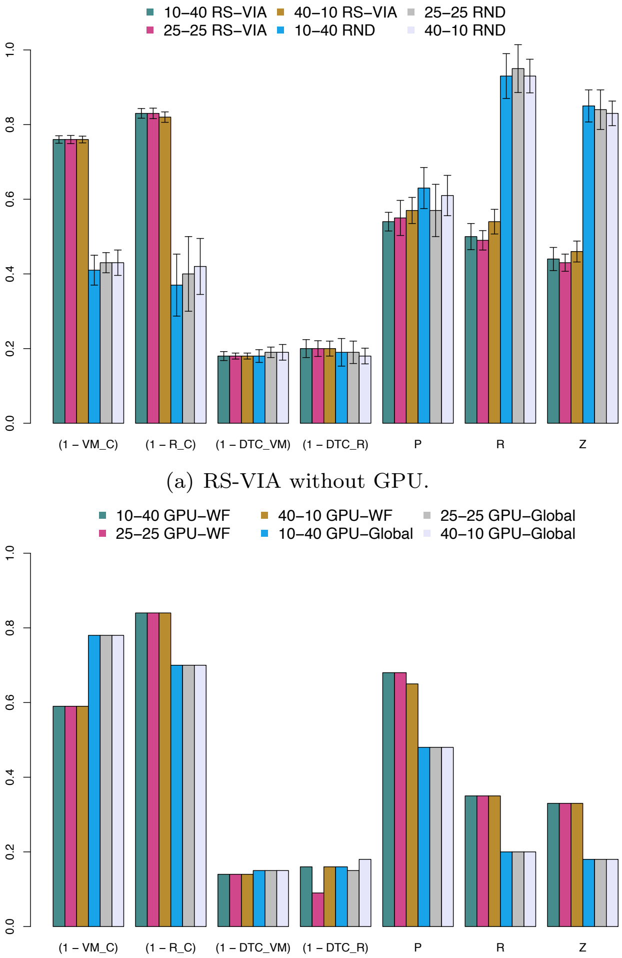
and 5.4.2. Figures 7(a) and 8(a) show the results for all algorithms. The random approach spreads resources atop multiple providers, regions, and zones, representing a client that performs the control and spreading manually, without support of the RS-VIA. Consequently, the allocation cost is greatly increased, however, the communication one remains statistically equivalent to RS-VIA with SA.

It is worthwhile to mention the allocations conducted by RS-VIA with SA drastically decreased the costs associated with VMs and replicas, achieving a cost-efficient allocation for both reliability levels (99.95% and 99.995%). In addition, RS-VIA with SA maintained a controlled spread of virtual resources atop zones, regions, and providers, which increases the survivability capacity of VIs in the eventual occurrence of a failure.

Although efficient, the execution time of RS-VIA based on SA remains a real obstacle. For allocating a single request, 256 minutes are need to solve the LP, on average, followed by approximately 61 minutes to execute the SA algorithm. Thus, an allocation strategy accelerated by GPU can potentially improve the execution speedup (Section 4.2). This hypothesis is confirmed by simulations: the values for all metrics remain competitive with RS-VIA based on SA and higher than the random method. This behavior can be observed with the results in Figures 7(b) and 8(b). Regarding to the execution time, both GPU-accelerated methods process a VI request on less than 1 second. Specifically, the average execution time is:

- RS-VIA-GPU-WF: 241.08 ms, ± 23.56 ms; and
- RS-VIA-GPU-Global: 224.21 ms, ± 18.81 ms.

Figures 9 and 10 summarize RS-VIA results with three different requests composition in terms of regular-critical VMs ($\alpha = 10 - 40$, $25 - 25$, and $40 - 10$). Indeed, the efficiency of CPU and GPU implementations are compared.



(a) RS-VIA without GPU.

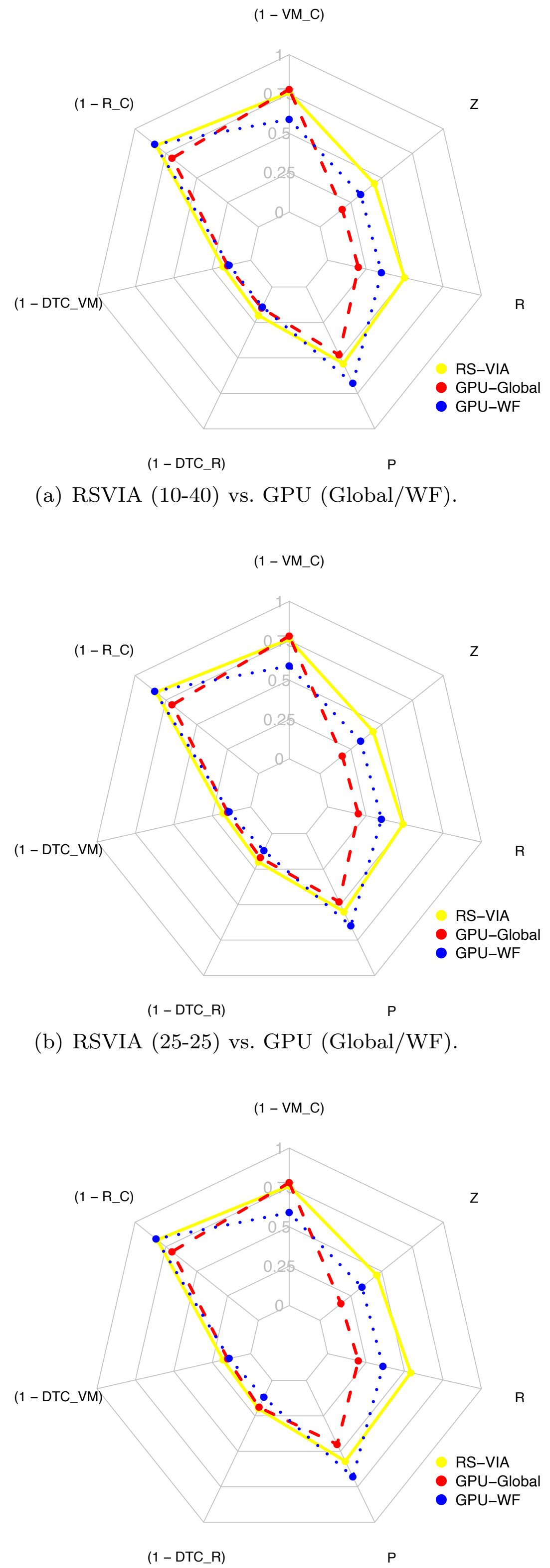
(b) RS-VIA with GPU.

Figure 8 RS-VIA results to reliability target $c = 99.995\%$.

In all cases, RS-VIA based on SA has the biggest area on radar plot, while RS-VIA-GPU-WF is worse than RS-VIA based on SA in only one metric (the use of providers). Moreover, the number of providers used by RS-VIA-GPU-WF to achieve the reliability target is greater than RS-VIA based on SA in all cases, except for requests with $40 - 10$ regular-critical VMs and $c = 99.995\%$(Figure 10(c)). In addition, RS-VIA-GPU-Global has the opposite behaviour being better than RS-VIA based on SA in almost all cases (except in Figure 10(c)). Finally, RS-VIA-GPU-Global has the smallest area in radar plots, so, the best results.

5.5 Discussion and Key Observations

Our work contributes to enhance the state of the art on survivable and reliable VIs allocation atop multiple IaaS providers. We proposed an exact MIP, a SA-based heuristic which receives as input the results obtained from an LP (relaxed version of the MIP), and two GPU-accelerated algorithms to find efficient solutions on suitable allocation time. In order to evaluate the efficiency of all mechanisms, simulations with real data were performed, analyzing small and large-scale scenarios.



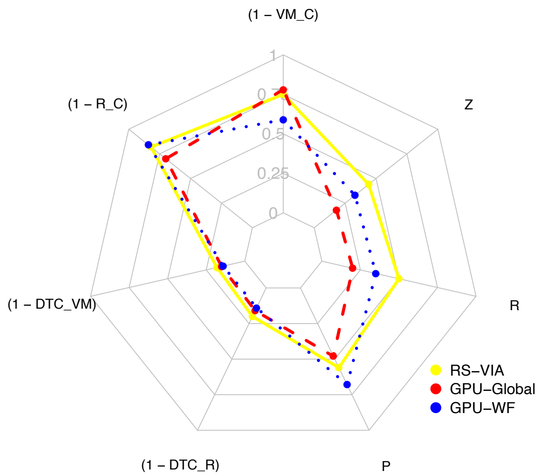
(a) RSVIA (10-40) vs. GPU (Global/WF).

(b) RSVIA (25-25) vs. GPU (Global/WF).

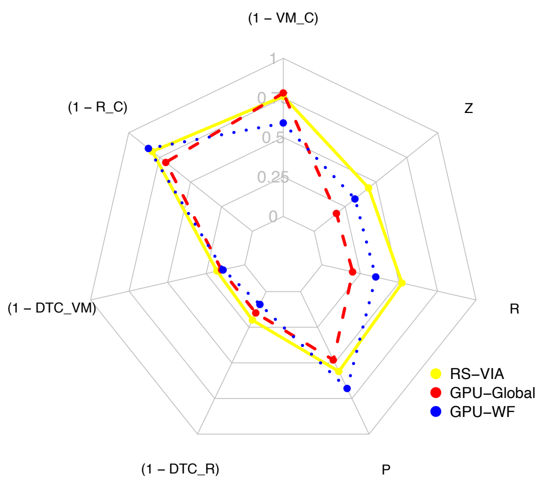
(c) RSVIA (40-10) vs. GPU (Global/WF).

Figure 9 RS-VIA vs. GPU $c = 99.95\%$.

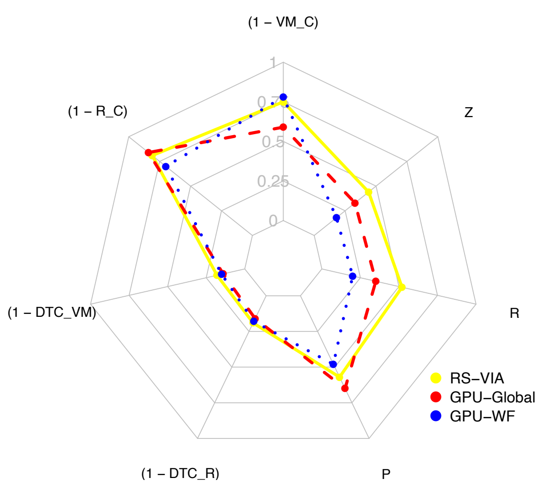
The first set of simulations (Subsection 5.3.1) analyzed the cost-survivability trade-off with a reduced set of providers (2 providers with large geographical footprint were selected, Amazon EC2 and Google Computing Engine). CO, SO, and EA approaches were



(a) RSVIA (10-40) vs. GPU (Global/WF).



(b) RSVIA (25-25) vs. GPU (Global/WF).



(c) RSVIA (40-10) vs. GPU (Global/WF).

Figure 10 RS-VIA vs. GPU $c = 99.995\%$.

compared with different configurations for α parameter. In turn, the second scenario (Subsection 5.4.2) compared the allocations performed by the exact MIP with results from RS-VIA based on Simulated Annealing. The third simulation scenario (Subsection 5.4.3) discussed the efficient of RS-VIA executed on CPU and GPU when applied to large-scale scenarios. In short, 31 IaaS cloud providers, 133 regions and 153 zones were analyzed as candidates for hosting VIs requests composed of 50 VMs with 3 possible compositions in terms of critical and regular virtual resources.

The results from first scenario indicates the exact allocation model can perform a VI allocation with controlled spread of virtual resources atop zones, regions, and providers, without inflating the provisioning cost for VMs, replicas and network data transfer. Based on the allocation with balanced configuration between cost and survivability terms of objective function ($\alpha = 0.5$), the results from the second scenario demonstrated that RS-VIA obtains a close performance to EA regarding all metrics. The third scenario demonstrated the same tendency previously observed in the second scenario. However, the problem dimension penalizes the CPU-tailored simulated annealing. Following this line, the GPU-accelerated algorithms found efficient solutions in a few seconds. Finally, it is possible to conclude that the proposed mechanisms meet the initial objectives, increasing survivability and reliability without drastically increasing cost.

6 Related Work

The specialized literature comprises the allocation of physical resources to host VIs, and techniques to improve virtual resources survivability and reliability.

Allocating physical resources for hosting VIs. Houidi *et al.* [13] proposed a MIP and a set of heuristics to solve the Virtual Network Embedding (VNE) problem focusing on cost reduction and acceptance ratio increase. They propose the allocation atop multiple providers. We share a similar approach considering IaaS clouds, however, providers details are not required neither interoperability mechanisms.

A different perspective was analyzed by Caron *et al.* [24]: instead of considering multiple providers, the proposal aimed the simultaneously allocation atop a private cluster and a public cloud. An optimal allocation concerning the multiple allocation criteria was proposed. Ficco *et al.* [25] proposed a meta-heuristic scheme for managing elastic resources reallocation in cloud infrastructures. They aim to maintain a balance between the different interests of clients SLAs and the provider during the allocation, resizing, replication and migration processes. Both proposals can be jointly applied with our approach for improving the selection of a candidate cloud provider.

Regarding to the allocation of VIs into DCs, techniques to minimize the bandwidth consumption combined with privacy support were proposed in [19]. We are aligned with this proposal considering the virtual link modeling. In [12], a tree-based heuristic was proposed to speedup the VI allocation. The heuristic tends to group virtual resources increasing the impact of an eventual failure. Although not aiming a survivable allocation, a controlled spreading of virtual resources atop a cloud DC was applied in [11], however full knowledge and control on cloud DC is required.

Summing up, the literature on VI embedding into DC, or similar scenarios (VNE), comprises multiple proposals with distinct goals [10, 15]. Concerning to the multiple provider approaches, the previously proposed techniques rely on interoperability data and/or sharing of private provider's data [26, 27], while the present proposal is based on public information and can be applied for any IaaS provider. Moreover, the present proposal is agnostic to private allocation mechanism.

Techniques for provisioning survivable VIs. The survivable provisioning of VIs was proposed in [6]. Similarly to the present work, the mechanism relied on ORP for defining the number of replicas. However, the allocation was conducted on a controlled DC with where the mechanism has full knowledge on probability of failures and MTBF. The allocation was performed in two steps, first defining the number of replicas and later applying an allocation heuristic, which can lead to a suboptimal solution. Groups of failures and cost-effective allocation were not considered. Our approach advance the field by jointly defining the replicas and spreading VMs on multiple providers. In short, on a single step, the exact survivable, reliable and cost-effective allocation is accounted, as discussed in Sec. 5.4.

The ORP technique was also applied for VNE [5]. A small set of replicas was defined for backing up multiple tenants. We share a different view on the present work considering a non-cooperative scenario as usually observed on public providers. Indeed, the SLA is individually performed with each tenant defining the target reliability. In addition, Bodik *et al.* [7] improved the fault tolerance on DC without increasing the bandwidth load, while Cavalcanti *et al.* [8] investigated the trade-off between DC fragmentation and survivable provisioning. It is worthwhile to highlight that the present proposal combined cost-effective with survivable and reliable VI allocation on multiple cloud providers, filling a research gap with concerns to the tenant's perspective.

7 Considerations & Future Work

We presented an alternative to increase the VI survivability, ensuring the request reliability through replicas, without increase the cost of the VI allocation. In order to achieve that, we formulate a MIP to define the exact allocation of the VI atop multiples providers. Latter, a set of variables were relaxed obtaining a LP. The approximated results are used as input for decreasing the number of candidates on a SA algorithm, composing RS-VIA. The results shows our solution is effective in terms of reliability and survivability, without inflating the provisioning cost. The total cost remains as close as possible to the minimum for the requested VI, respecting the target reliability. In addition, the GPU implementation shows it is feasible in large-scale scenarios and takes the state-of-the-art in this area to another level. Further work aims to perform the implementation as an open cloud service, using OpenStack clouds as testbed.

Acknowledgments

This research was supported by the UDESC, FAPESC, PROMOP / CAPES, and developed at LabP2D.

References

[1] Peter M Mell and Timothy Grance. SP 800-145. the NIST definition of cloud computing. Technical report, NIST, 2011.

[2] Abel Avram. Amazon EC2 Outage Explained and Lessons Learned. Technical report, InfoQ, 2011.

[3] Amazon EC2. Amazon Web Services summary of the amazon s3 service disruption in the northern virginia (us-east-1) region. Technical report, Amazon, 2017.

[4] Ramesh Govindan, Ina Minei, Mahesh Kallahalla, Bikash Koley, and Amin Vahdat. Evolve or die: High-availability design principles drawn from googles network infrastructure. In Proc. of the SIGCOMM conference, pages 58–72, Florianopolis, Brazil, 2016. ACM, ACM.

[5] Wai-Leong Yeow, Cédric Westphal, and Ulas C Kozat. Designing and embedding reliable virtual infrastructures. SIGCOMM Computer Communication Review, 41(2):57–64, 2011.

[6] G. Koslovski, W. L. Yeow, C. Westphal, T. T. Huu, J. Montagnat, and P. Vicat-Blanc. Reliability support in virtual infrastructures. In 2nd Int. Conf. on Cloud Computing Technology and Science, pages 49–58, Indianapolis, USA, Nov 2010. IEEE.

[7] Peter Bodík, Ishai Menache, Mosharaf Chowdhury, Pradeepkumar Mani, David A. Maltz, and Ion Stoica. Surviving failures in bandwidth-constrained datacenters. In SIGCOMM Conf. on Applications, Technologies, Architectures, and Protocols for Computer Comm., pages 431–442. ACM, 2012.

[8] Gustavo Cavalcanti, Rafael R Obelheiro, and Guilherme Koslovski. Optimal resource allocation for survivable virtual infrastructures. In Design of Reliable Communication Networks, pages 1–8, Ghent, Belgium, 2014. IEEE.

[9] Borja Sotomayor, Rubén S Montero, Ignacio M Llorente, and Ian Foster. Virtual infrastructure management in private and hybrid clouds. IEEE Internet computing, 13(5):14–22, 2009.

[10] Andreas Fischer, Juan Felipe Botero, Michael Till Beck, Hermann De Meer, and Xavier Hesselbach. Virtual network embedding: A survey. IEEE Communications Surveys & Tutorials, 15(4):1888–1906, 2013.

[11] Felipe Rodrigo de Souza, Charles Christian Miers, Adriano Fiorese, and Guilherme Piegas Koslovski. QoS-Aware Virtual Infrastructures Allocation on SDN-based Clouds. In Proc. of the Int. Symp. on Cluster, Cloud and Grid Computing, CCGRID, Madrid, Spain, 2017. IEEE.

[12] Ramon de Oliveira and Guilherme Piegas Koslovski. A tree-based algorithm for virtual infrastructure allocation with joint virtual machine and network requirements. International Journal of Network Management, 27(1):e1958, 2017.

[13] Ines Houidi, Wajdi Louati, Walid Ben Ameur, and Djamal Zeghlache. Virtual network provisioning across multiple substrate networks. Computer Networks, 55(4):1011–1023, 2011.

[14] NM Mosharaf Kabir Chowdhury and Raouf Boutaba. A survey of network virtualization. Computer Networks, 54(5):862–876, 2010.

[15] Guofu Feng and Rajkumar Buyya. Maximum revenue-oriented resource allocation in cloud. Int. J. Grid Util. Comput., 7(1):12–21, December 2016.

[16] Jothy Rosenberg and Arthur Mateos. The Cloud at Your Service. Manning Publications Co., 1st edition, 2010.

[17] Shriram Rajagopalan, Brendan Cully, Ryan O'Connor, and Andrew Warfield. Secondsite: Disaster tolerance as a service. SIGPLAN Not., 47(7):97–108, March 2012.

[18] Algirdas Avizienis Ucla, Algirdas Avizienis, Jean claude Laprie, and Brian Randell. Fundamental concepts of dependability. University of Newcastle upon Tyne, Computing Science, Newcastle, U.K., 2001.

[19] Leonardo Richter Bays, Rodrigo Ruas Oliveira, Luciana S Buriol, Marinho Barcellos, and Luciano Paschoal Gaspary. A toolset for efficient privacy-oriented virtual network embedding and its instantiation on SDN/OpenFlow-based substrates. Computer Communications, 82:13–27, 2016.

[20] M. Chowdhury, M.R. Rahman, and R. Boutaba. Vineyard: Virtual network embedding algorithms with coordinated node and link mapping. IEEE/ACM Transactions on Networking, 20(1):206–219, Feb 2012.

[21] Lucas Leandro Nesi, Mauricio Aronne Pillon, Marcos Dias de Assunção, and Guilherme Piegas Koslovski. GPU-accelerated algorithms for allocating virtual infrastructure in cloud data centers. In 18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2018), May 2018.

[22] Stephen Marsland. Machine learning: an algorithmic perspective. Chapman and Hall/CRC Press, Taylor and Francis Group, Boca Raton, FL, USA, 2 edition, 2015.

[23] Valerio Persico, Pietro Marchetta, Alessio Botta, and Antonio Pescapè. Measuring network throughput in the cloud. Comput. Netw., 93(P3):408–422, December 2015.

[24] Eddy Caron and Marcos Dias de Assunção. Multi-criteria malleable task management for hybrid-cloud platforms. In 2016 2nd CloudTech, pages 326–333, Marrakech, Morocco, May 2016. IEEE.

[25] Massimo Ficco, Christian Esposito, Francesco Palmieri, and Aniello Castiglione. A coral-reefs and game theory-based approach for optimizing elastic cloud resource allocation. Future Gen. Comp. Sys., 78:343–352, 2018.

[26] Fabrizio Messina, Giuseppe Pappalardo, Antonello Comi, Lidia Fotia, Domenico Rosaci, and Giuseppe ML Sarné. Combining reputation and qos measures to improve cloud service composition. International Journal of Grid and Utility Computing, 8(2):142–151, 2017.

[27] Mohamed Raouf Habes and Habiba Belleili-Souici. Towards a fairer negotiation for dynamic resource allocation in cloud by relying on trustworthiness. International Journal of Grid and Utility Computing, 8(3):185–200, 2017.