Received: 20 December 2018 | Revised: 21 May 2019 | Accepted: 22 May 2019
DOI: 10.1002/spe.2723

RESEARCH ARTICLE

WILEY

Cloud broker proposal based on multicriteria decision-making and virtual infrastructure migration

Leonardo R. Rodrigues[1] | **Euclides Cardoso Jr.**[1] | **Omir C. Alves Jr.**[2] | **Fernando F. Redígolo**[3] | **Maurício A. Pillon**[1] | **Charles C. Miers**[1] | **Guilherme P. Koslovski**[1]

[1]Graduate Program in Applied Computing, Santa Catarina State University, Joinville, Brazil
[2]Laboratory of Computer Networks and Architecture, University of São Paulo, São Paulo, Brazil
[3]Department of Computer Science, Santa Catarina State University, Joinville, Brazil

Correspondence
Guilherme P. Koslovski, Graduate Program in Applied Computing, Santa Catarina State University, 89219-710 Joinville-SC, Brazil.
Email: guilherme.koslovski@udesc.br

Present Address
Guilherme P. Koslovski, Universidade do Estado de Santa Catarina, Rua Paulo Malschitzki, 200 - Zona Industrial Norte, 89219-710 Joinville-SC, Brazil

Summary

The adoption of infrastructure as a Service (IaaS) is a reality for academic, industrial, and governmental institutions. Cloud tenants request dynamically provisioned virtual infrastructures (VIs) tailored to their application requirements, detailing not only the virtual compute/storage resources but also the network components, topology, and services. The creation of a large number of cloud providers came along with the widespread use of VIs. The selection of an appropriate provider is a challenging task due to the diversity of the IaaS market and formally is a multicriteria analysis (NP-hard). Notwithstanding the provider selection complexity, the mobility of VI-hosted applications is limited due to the optimization anchors introduced by providers. Although the existing IaaS cloud brokers can indicate a hosting provider, they lack on conceptual and technical skills to migrate a VI and all its internal components between providers. This work enhances the state-of-the-art on IaaS cloud brokerage by proposing virtual infrastructure multicriteria allocation and migration–based broker (VIMAM), which performs a multicriteria analysis of providers and VI migration. VIMAM is driven by an analytic hierarchy process (AHP) to select an IaaS provider, offering a set of predefined weighting schemas to represent distinct tenant perspectives. Moreover, to migrate a VI, VIMAM takes into account the virtual machines, containers, switches, and other topology elements. In addition to discussing the AHP ranking weights and frequency of providers selection, the experimental analysis details the implementation of an OpenStack and Docker–based prototype for VI migration.

KEYWORDS

broker, container, IaaS, migration, multicriteria, network, virtual infrastructure, virtualization

1 | INTRODUCTION

The cloud computing paradigm consolidated the on-demand provisioning and delivering services following a pay-as-you-use or pay-as-you-go cost model.[1] Nowadays, tenants can request elastic computing, storage, and network resources without human interaction. On the infrastructure-as-a-service (IaaS) providers' perspective, the virtualization technology is the driving force to better usage and share data center (DC) resources. Indeed, virtual machines (VMs) and containers are jointly allocated to support distributed applications. Furthermore, for adhering to applications specificities, virtual private clouds (or virtual infrastructures (VIs)) can be composed of VMs, containers, and virtual switches, which are interconnected by a virtual private network.[2,3]

In the last decade, the industrial and academic communities observed a wide adoption of IaaS cloud services and, consequently, the creation of large number of cloud providers.[4] Nowadays, a simple Internet search reveals more than 100 cloud providers, each one composed of its own cloud computing DCs, management framework, application programming interfaces (APIs), and data models. Although only four IaaS providers have approximately 53% of the world's market share, other alternatives are available. In fact, cloud computing environment became a heterogeneous and complex scenario. In most cases, the tenants are unaware of the existence of small-scale providers.

Thus, the selection (for initial allocation or reallocation) of which provider is more adequate to a specific tenant is a challenging task.[5,6] Several providers offer similar services changing only their charging policy,[7] performance flavors, and other management features.[8-10] In parallel, the cloud computing adoption is a reality for multiple enterprises. Production, research, and development infrastructures are consolidated and running for some years. Each provider is based on orchestration tools specifically defined to its DCs, optimizing the capabilities according to the services offered and classified objective functions. Indeed, IaaS providers have full knowledge on their infrastructures and apply optimizations to increase their revenues. Although the tenants believe their VIs are isolated provisioned, there are anchors and fixed thresholds linked to virtual resource allocation and/or management tools.[11] *In such a complex and challenging scenario, our work focus on cloud provider selection and VI migration to advance in this research field, defining a brokerage service based on tenants' perspective.*

Selecting the appropriated cloud provider

Related to IaaS cloud provider selection, the analysis and selection of IaaS providers is based on the multicriteria decision-making (MCDM) method to qualify IaaS providers is a promising alternative.[10,12-14] MCDM enabled a near-optimal mapping of virtual requests on cloud providers by performing simultaneous analysis of a set of relevant variables and candidates. In this work, the MCDM is guided by multiple criteria and trade-offs: energy versus performance and consolidation versus availability, among others. The analytic hierarchy process (AHP) method stands out to make a decision with trade-off and multicriteria,[15] allowing to set assortment aspects and component hierarchy. The specialized literature proposed the use of AHP-based models to select providers and servers, highlighting the efficiency and impact of AHP parameterization to align provider objectives and the allocation results.[10,12-14] However, the majority of proposals (discussed in Section 3) consider only the selection of VMs, disregarding the virtual private network details. The present work innovates by adding virtual network requirements to quality of service (QOS) and cost hierarchical weight schema used to guide the AHP model. Moreover, the proposed AHP model is based on Common Information Model (CIM) initiative.[16]

Migrating VIs

Traditionally, IaaS cloud providers have different billing policies and QoS even between their own regions and zones (eg, Amazon Web Services (AWS)). These aspects directly affect the choice of providers by tenants. Over time, tenant requirements (high availability and QoS) and economical funds certainly change. Eventually, these changes can lead tenants to select another zone, region, or provider to host theirs VIs. Providers typically have standardized management tools between their sites, which enable VIs migration between them.[17] However, providers are not interested in adopting standardized tools between different providers, and the lack of interoperability inhibits migration of tenants towards competitor providers.[11,18] Although portability and interoperability are important requirements described by National Institute of Standards and Technology (NIST),[2] to leave a provider, the tenants need to migrate all VI components involved in their applications.[19-21] In summary, to migrate a VI from provider A to provider B, a tenant must migrate all VMs and/or containers, which support his/her distributed application. Latter, all VI network connections must be recreated on provider B.[22-24] Hence, VI network migration should ensure the following:

1. Reestablish the private network connection, as soon as the VM or container is alive.
2. Keep active network link between migrated VI (from provider A) and external network.

Virtual infrastructure migration

In this context, we propose a cloud broker based on MCDM and virtual infrastructure migration. Our IaaS cloud broker, termed virtual infrastructure multicriteria allocation and migration–based broker (VIMAM), receives a provider selection request, and after defining, the appropriated new cloud provider (region or zone) performs the VI migration without human interaction, ie, automatically. Specifically, VIMAM relies on the AHP method to select a cloud provider: a tenant submits a VI request to the broker management, which identifies and qualifies all possibilities of IaaS providers for this request. The list of candidate providers is rated according to multicriteria to attend the tenant's perspective. It is worthwhile to mention that a previously performed allocation may not be the appropriate choice after a provider failure or application reconfiguration. An eventual failure or peak of use may trigger a VI migration. In this sense, VIMAM offers a migration module to move containers and virtual networks between providers. A proof-of-concept implementation was developed based on OpenStack cloud management project and Docker framework. Finally, VIMAM was evaluated in three key aspects covering the key modules developed:

1. Response time to make a decision.
2. Quality of a decision.
3. Applicability of migration module in real environment.

VIMAM results demonstrate it is possible to make an online multicriteria decision for selecting the cloud provider and, if necessary, migrate VIs between different real providers.

The remainder of this paper is organized as follows. Section 2 reviews the concepts and issues that motivate our research, while Section 3 discusses related work. The proposed cloud broker architecture is presented in Section 4. Experimental analysis is discussed in Section 5, while considerations and perspectives are presented in Section 6.

2 | MOTIVATION AND PROBLEM DEFINITION

2.1 | VIs and cloud brokers

A VI can be formally defined as a set of VMs, containers, and virtual networking resources provided during a given time frame.[1,25-27] It is worthwhile to mention that virtual network resources have the same importance level of processing and storage. Virtual links, switches, and routers can be configured to support peak load and to reserve the appropriated bandwidth to support the VI-hosted applications. Figure 1 depicts an example of a VI composed of 20 containers placed atop of two VMs.

In Figure 1, a private local area network (with tenant-defined Internet Protocol (IP) address) is deployed as a virtual switch. In addition (not represented in Figure 1), multiple services to help the internal management, such as the floating IP service can be contracted to deliver Internet-routable reachability for VMs and containers.

Cloud computing environment allows tenant to request on-demand access to processing, storage, and networking resources with pay-as-you-use and/or pay-as-you-go business model. One challenge for tenants is to know the correct provisioning of their resources when they use an IaaS model.[28,29] Moreover, the ever-increasing number of providers and the number of resource arrangements among different providers lead to a new challenge for tenants to choose the right provider and to perceive how long their choices were correct.

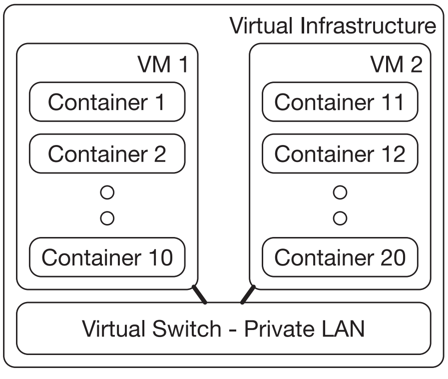


FIGURE 1 Example of a virtual infrastructure composed of 20 containers placed atop of two virtual machines (VMs). A private local area network (LAN) is deployed as a virtual switch

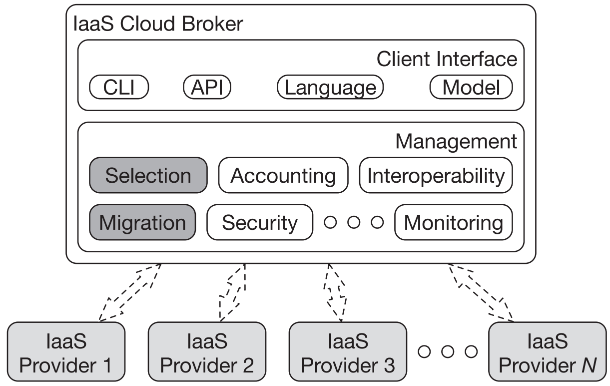


FIGURE 2 Main modules of an infrastructure-as-a-service (IaaS) cloud broker. We focus on the provider's selection and migration modules. API, application programming interface; CLI, command-line interface

A possible provider's selection is built on multicriteria analysis and depends on tenant's perspective, providers' billing policies, and existing monitoring tools. These criteria can change over the time, influenced by the administrative decision of providers to offer in a differentiated way their resources or due to tenant's application requirements (eg, peak of use and failure event, among others). With new criteria or new requirements, tenants have other options to select; the best cloud provider for their VI in the past may no longer be the same. Although service-level agreement (SLA) allows the termination of ongoing VI, tenants wish to minimize service downtime from their VI-hosted applications. In summary, it is evident that provider's selection and VI migration are technical challenges to tenants.

To intermediate the tenant-provider relationship, there is an important actor in the cloud computing context: the cloud broker. Renowned institutions such as Gartner[30] and NIST[2] defined cloud brokers as actors of relevant importance in cloud computing. In fact, a broker simplifies the delivery of aggregated services to the tenants,[31] softening the lack of interoperability and application portability between several services providers.[32] Finally, a broker avoids potential vendor lock-in[33] and becomes essential to transform the complex market of the cloud computing into a commodity.[19,34,35] Figure 2 describes two layers of an IaaS cloud broker: client interface and management.

Analyzing the Figure 2 from left to right, the Client Interface offers solutions to interact with the brokerage management mechanism. specifically, command-line interface (CLI) and API are interfaces to submit commands and receive data, while Language and Model deals with data representation. The second layer, Management, represents all technical service modules used to select a provider, to migrate (when need), to change or increase virtual resources, to monitor the resources usage, and to manage the security services. As observed, a cloud broker is a complex combination of management tools. This paper describes our solution specifically for selection and migration modules located in the management layer.

Since providers have different classifications for the same type of service that must be intermediated by a cloud broker, Table 1 resumes the nomenclature used by the providers with the highest level of market-share adoption. Some criteria (eg, switch, router, and latency) are not explicitly provided being part of greater services. Table 1 summarizes the services that can be individually used (or combined) to provide the domain name system, load balancing, QoS, virtual networking, security management, firewall, network address translation (NAT), VMs, and containers.[36-39] Table 1 also corroborates that different providers can offer similar services varying the fee approaches, levels of performance, and QoS indicator, as well as supporting technologies.[7-10] Internally on IaaS cloud providers, the resource orchestration is performed by mechanisms optimized to manage computing, storage, and networking resources according to the services offered and objectives of the provider. Thus, a tenant makes a VI request theoretically independent of the physical architecture of providers, but the DC optimizations can generate *anchors* setting a VI to the provider. Exemplifying, the interfaces and rules for defining load balancing, security rules, and IP address resolution are considerably different between providers.

2.2 | Virtual resources migration between IaaS providers

The migration of virtual resources is a challenging task. In the IaaS cloud scenario, the migration is the process of moving a running virtual resource (VM, container, switch, router, or link) between different physical hosts, preferable without disrupting the network. Memory, storage, and network connectivity of virtual resources are transferred from the original host to the destination. In the scope of the present work, the original and destination hosts can be placed at distinct IaaS providers. Three main categories of virtual resources migration are discussed (pros and cons) as follows.

TABLE 1 Nomenclature adopted by the four providers with the highest level of market share

Services / Providers	Rackspace (OpenStack)	Microsoft Azure	AWS	Google Cloud
DNS	Designate/DNS as a service (DNSaaS)	Azure DNS	Amazon Route 53	Cloud DNS
Load balancing	Neutron/load balancer as a service (LBaaS)	LBaaS	Elastic Load Balancing	Google Cloud Load Balancing
QoS	Neutron/QoS	Express Route	SLA QoS	SLA QoS
Virtual networking	Neutron	Azure Virtual Network	Amazon Virtual Private Cloud /AWS Direct Connect	Cloud Virtual Network
Security	Keystone	Azure Security Central	Amazon Inspector	Cloud Identity & Access Management
Firewall	Neutron/Firewall as a service (FWAAS)	Firewall	AWS Web Application Firewall	Firewall
NAT	Neutron	NAT gateways	NAT gateways	NAT gateways
VMs and containers	Nova/Mangnum	VMs/Containers	VMs/Containers EC2	Compute Engine

Abbreviations: AWS, Amazon Web Services; DNS, domain name system; NAT, network address translation; QoS, quality of service; SLA, service-level agreement; VM, virtual machine.

2.2.1 | VM migration

Among the main benefits of migrating VMs, the load balancing in IaaS cloud DCs is highlighted.[40] In contrast, the performance of hosted applications is impaired during the VM migration process.[41,42] On the technical details, one important factor that should be considered is the management of virtual storage, which is necessary to carry out the process without interrupting the hosted application execution.[43,44] To perform migration between small-scale clouds, shared storage can be used as a solution. However, when it comes to large-scale clouds, this approach may be inefficient due to the high volume of data. The specialized literature comprehends proposals to speed up the migration[43,45,46] by applying memory precopy[43] and on-demand memory migration.[45] These mechanisms are focused on migrating VMs on private DCs[46] being ineffective for migrations between IaaS cloud providers conducted atop of the Internet. In addition, to perform a VM migration, administrative privileges are required to access the hypervisor, thereby restricting the migration decision only to the cloud manager. In other words, the tenants are susceptible to vendor lock-in.

2.2.2 | Container migration

The container technology is based on sharing a single kernel among multiple processes.[47] In this sense, the migration of a container consists of unplugging a process from the original operating system (OS) comprehending memory state CPU and register values and then attaching it on a new host. Two main pros of container migration are worthwhile to mention:

1. All components (files and libraries) required for executing a container-hosted application are linked to it.
2. It is not necessary to have access to the hypervisor for performing a migration, thus enabling the tenant to migrate without need to contact the IaaS service provider.

In this sense, the container's technology prevents the vendor lock-in. In contrast, container-based virtualization is still a new and evolving technology in cloud providers, and not all container management tools have migration support.

2.2.3 | VI migration

The VI migration consists of moving an application along with the virtualization layer, which hosts it as well as all related services. Thus, in addition to the computational resources (VMs or containers), the virtualized network (links, routers, and switches) are jointly migrated. It is important to note that, in the IaaS cloud scenario, the virtualization layer maybe provisioned as VMs, containers on hardware, containers on shared OS, or containers inside VMs. Moreover, the key advantage in migrating a VI is that the network and hosted application settings are maintained unchanged. In this sense, the virtual network must be preconfigured in advance in the destination service provider (using the same private IP configuration).[22-24] The correctness of application's communication is ensured by replicating the topology model from the source provider to the destination provider. On the other hand, after migrating the VI to the destination provider, the IP of the public network may not be maintained as it is associated with the source provider configuration. To maintain the Internet visibility and availability of the VI-hosted services, it is necessary to reestablish or redirect the existing connections. Although not trivial, research work in the area reports some solutions to the problem of traffic redirection and Border Gateway Protocol dynamic reconfiguration.[48,49] Thus, this broad subject is outside the scope of this work. It is worthwhile to mention that brokerage in IaaS clouds can be based on the migration of VIs composed of containers allocated in VMs as two advantages are proposed by this approach:

1. The freedom of management given to tenants by this technology (provider independence).
2. The use of containers atop VM-related services (eg, elastic provisioning and load balance) currently offered and consolidated by IaaS cloud providers.

In this work, we focus on migration, the exclusion of container provisioning over hardware is justified by the management dependency. The same limitation discussed in VM migration is observed: the tenant must require support from the IaaS provider to perform a migration (composing an anchor and vendor lock-in). It is important to highlight that if the tenant has access to the management of their containers (independent of the IaaS provider), the benefits of the brokerage provided by this work (discussed in Section 4) can be applied.

A specific discussion must address the services used to support the VI-hosted services. Cloud services as firewall, load balancing, and NAT (as detailed by Table 1) are candidates for composing vendor lock-ins. Each service is configured by a specific API, which composes a technological barrier on automating the migration. In fact, to perform a VI migration, each service must be preconfigured in advance on target provider.

Finally, the VI migration by a cloud broker is transparent to the tenant. The brokerage tool can help to evaluate and decide if a migration must be performed based on predefined thresholds. When authorized, the broker executes the migration actions, both on the source provider and on the destination provider, as discussed in Section 4.

3 | RELATED WORK

According to the literature review performed considering the cloud broker modules investigated in the present work, two main categories of related work are identified. The first one aims to use multicriteria methods in the cloud providers, while the second one comprehends papers focusing on migration of virtual resources.

3.1 | Multicriteria methods to allocate virtual resources

Initially, an analysis of several multicriteria methods used in the cloud computing and a decision-making taxonomy were proposed.[14] In addition, the authors analyzed hybrid computing platforms (ie, cloud computing and private clusters) discussing the differences between global and local algorithms. Simulations based on discrete events were performed to evaluate a set of algorithms, ie, allocated cluster algorithm, cloud cluster algorithm, and dedicated cluster algorithm, pointing out the applicability of multicriteria analysis on hybrid scenarios.

Yazir et al decomposed the management of cloud computing resources in different tasks executed by autonomous agents.[50] Such agents realized settings in parallel with the PROMETHEE method. In addition to the scalability achieved by parallel and decentralized processing, this approach ensures for the provider flexibility, by varying criteria weights and by adding or removing criteria rather than changing the cloud provider settings. In turn, Ergu et al proposed a model for cloud resource allocation based on the AHP method, the criteria considered in the parity comparison are time of each task, costs, reliability, and bandwidth of the performed tasks.[51] Although both works proposed the application of multicriteria methods, they focused only in the provider's perspective, while the present work addresses an inherent issue to cloud tenants.

In addition to applying AHP, Martens et al presented a method that selects a service provider based on risk analysis.[52] The risk metrics were build atop the confidentiality, integrity, and availability triad of security. While they seek for factors that impact on cost and risks selection of the providers, the present work consider a wide range of criteria, applying the AHP as focus on VI provisioning.

Garg et al developed a framework to select an IaaS service provider based on a system composed of three modules[9]: SMICloudBroker, in charge of interacting with tenants; Monitoring, based in finding all services that fulfill the tenant's requirements; and Service Catalog, a module to store the characteristics of services and providers. The architecture proposed in Section 4 can be integrated to the framework by adding a specialized multicriteria analysis and a VI migration module.

Dastjerdi et al developed CloudPick, a QoS architecture for deploying services across clouds.[53] The proposal uses multiple providers to host a VI to reduce costs and improve the quality-of-experience of end users. VIMAM can be combined

TABLE 2 Percentage of choice impact according to the weight schema. Bold-written values indicate greater impact on provider choice

Services/schema	Flat	Security	Network elasticity	QoS	Web hosting	Network-intensive	Market research
Switch	12.5%	1.9%	2%	2.1%	2.9%	4%	1.1%
Router	12.5%	3.9%	4.3%	6.2%	6.2%	7.1%	1.8%
SDN equipment	12.5%	6.9%	**8.7%**	8.8%	8.8%	**18.5%**	2.9%
Container	6.25%	0.75%	0.9%	1.45%	1.05%	1.85%	4.2%
Virtual machine	6.25%	0.75%	0.9%	1.45%	1.05%	1.85%	4.2%
Load balancing	10%	9.6%	7.9%	9.4%	**23.6%**	5.3%	10.3%
DNS	10%	6%	5.3%	8.6%	7.1%	4%	3.4%
Latency	5%	12.7%	**8.1%**	**23.9%**	6.3%	**16.3%**	6.2%
Bandwidth	5%	6.4%	4%	**11.9%**	3.1%	**8.2%**	1%
NAT	3.3%	**4.8%**	1.6%	3.7%	4.3%	0.6%	4%
Firewall	3.3%	**29.7%**	9.5%	9.4%	**22.7%**	3.9%	**26.6%**
IDS	3.3%	**11.9%**	5.5%	5.9%	8%	1.6%	**17.8%**
L1	2.525%	0.3%	**3.2%**	0.5%	0.4%	2.4%	1.4%
L2	2.525%	0.5%	**5.2%**	0.8%	0.8%	**4.7%**	2.2%
L3	2.525%	1.3%	**12.6%**	2.1%	1.2%	**7.9%**	3.3%
L4	2.525%	2.6%	**20.3%**	3.8%	2.5%	**11.8%**	9.6%

Abbreviations: DNS, domain name system; IDS, intrusion detection system; QoS, quality of service; NAT, network address translation; SDN, software-defined networking.

with CloudPick to analyze networking resources as well as to eventually migrate services to improve their metrics. In addition, a framework for selecting cloud services was proposed by Menzel et al.[54] The proposed framework and methodology demonstrate the efficiency of MCDM for selecting services based on high-level requirements. In Section 4.1.2 (see Table 2), a similar method based on market-research schema is presented to guide the selection of cloud providers.

In summary, by analyzing the multicriteria methods applied to cloud computing, it is possible to identify a research gap considering the tenant's perspective. Moreover, the existing solutions disregard the existence of virtualized networking interconnecting the VI components. The present proposal differs from the literature since it proposes that, given one or more tenant requests, the selection and analysis must be carried out according to the criteria established on the hierarchical structures towards service providers set (the details are discussed in Section 4).

3.2 | Migration of virtual resources

The specialized literature comprehends proposals from different categories, as listed in Section 2.2. Initially, considering the migration of applications between cloud providers, an algorithm to orchestrate the live migration was proposed by Carrasco et al.[55] Although the proposal is agnostic to source and target cloud providers, the algorithm does not take into account the migration of network components. A description of P-TOSCA, a model proposed to move applications and data, is applied to migrate virtual resources.[56] Ristove et al indicated the migration is performed by moving a Cloud Service Archive file, described in Topology and Orchestration Specification for Cloud Applications (TOSCA), having all information and metadata need to deploy an application.[57] A limitation is noted by the fact of the cloud provider must be able to replicate the application environment and allow deployment of the application from another cloud provider, ie, the destination cloud provider must be able to process files in the format defined by TOSCA.

Following the migration of applications between providers, Ali et al proposed Cloud Interoperability Broker, a software-as-a-service (SaaS) cloud broker.[58] To perform the migration, the providers must follow standard APIs to enable the data management and interoperability. Although promising, this approach is specific to SaaS providers and disregard the communication between cloud-hosted applications.

The live migration of VMs between IaaS cloud providers is investigated by Duggan et al.[59] This approach proposes an autonomous agent based on machine learning that has the ability to schedule and perform the migration of VMs. In other words, it aims to balance the use of cloud resources migrating the VM based on, for example, the seasonality of usage. As the network configuration is not migrated by the management agent, both source and target providers must be configured in advance to enable the VMs transfer without interrupting the hosted service.

A different approach to migrate VMs between IaaS providers is proposed by Tsakalozos et al.[60] The goal is to migrate a queue of tasks defined by the cloud administrator, as well as manage all the resources so that the migration happens without violating the SLA nor affecting the QoS requirements. In addition to sharing the limitations mentioned for other

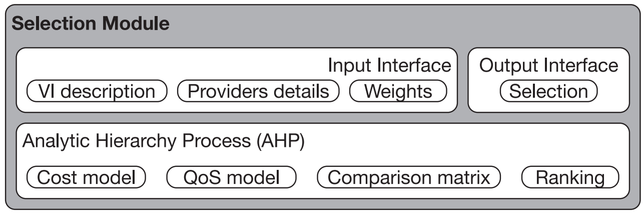


FIGURE 3 Architectural components of the selection module of the virtual infrastructure (VI) multicriteria allocation and migration–based infrastructure-as-a-service broker. QoS, quality of service

works regarding the network, the proposal targets the provider's perspective, which does not help in the prevention of vendor lock-in.

A framework to migrate VMs between IaaS cloud DCs was proposed by Zhang et al.[46] For speeding up the live migration process, the framework relies on a shared repository of VM images, accessible by source and destination DC servers. However, as discussed in Section 2.2, the migration management requires administrative privileges being totally realized by the provider. In turn, Zhao et al[61] addressed the joint migration of virtual computing and networking resources between providers. However, the VI must be completely managed by software-defined networking (SDN) technology. In a scenario with reduced interoperability tools and standards, as IaaS cloud providers, the requirement for a SDN-enabled network administration on source and target providers is a strong assumption.

Regarding the contemporaneous technical solutions to manage containers, Kubernetes[†] and Docker[‡], the migration of a container set is in the early stages of development. A de facto solution to manage migration between providers has not yet been proposed. Moreover, due to the multiple possibilities for configuring networks in containers, the migration of virtual connections between distinct administrative domains is a technological challenge.

4 | CLOUD BROKER ARCHITECTURE

The present work focus on selection and migration modules of the generic cloud broker architecture depicted in Figure 2,[2] composing the VIMAM mechanism. Indeed, both modules are agnostic to implementation details of other components (eg, accounting, interoperability, security, and monitoring) as they are based on well-defined APIs. Thus, this section details both modules implementation following the traditional execution ordering: first, the provider's selection phase, and latter, the VI migration.

4.1 | First phase: selecting IaaS providers

The architecture for the selection module is depicted by Figure 3. Following a top-down discussion, the input and output interfaces define the model and data need to perform a selection and to represent the solution, respectively. As input, the module receives a VI description, the provider details, and a set of weights to configure the multicriteria mechanism. After processing, a possible selection map (or an empty map) is returned.

The internal mechanism of the selection module is based on MCDM, composed of cost and QoS models, a comparison matrix, and a ranking method. In summary, to find a suitable solution, the mechanism analyzes the tenant's VI request configuring the AHP mechanism with the weights previously informed. An overview of the steps executed by the selection module is enumerated in Figure 4.

A brief explanation of the steps numbered in Figure 4:

- The tenant submits a VI request to VIMAM.
- Given a tenant's VI request, the weights are distributed based on distinct preconfigured schemas (summarized in Table 2). The tenant can select multiple schemas to be analyzed by the selection module.
- Each selected schema is individually processed by the AHP mechanism.
- The AHP mechanism is driven by QoS and cost models. The QoS represents the expected quality when selecting a given provider to host the request, while the cost is the canonical representation for provisioning costs. In summary, two executions of AHP are performed, one for each hierarchical model.
- As each AHP hierarchical returns an independent solution, this step performs a joint ranking.

[†]Available at https://kubernetes.io
[‡]Available at https://www.docker.com

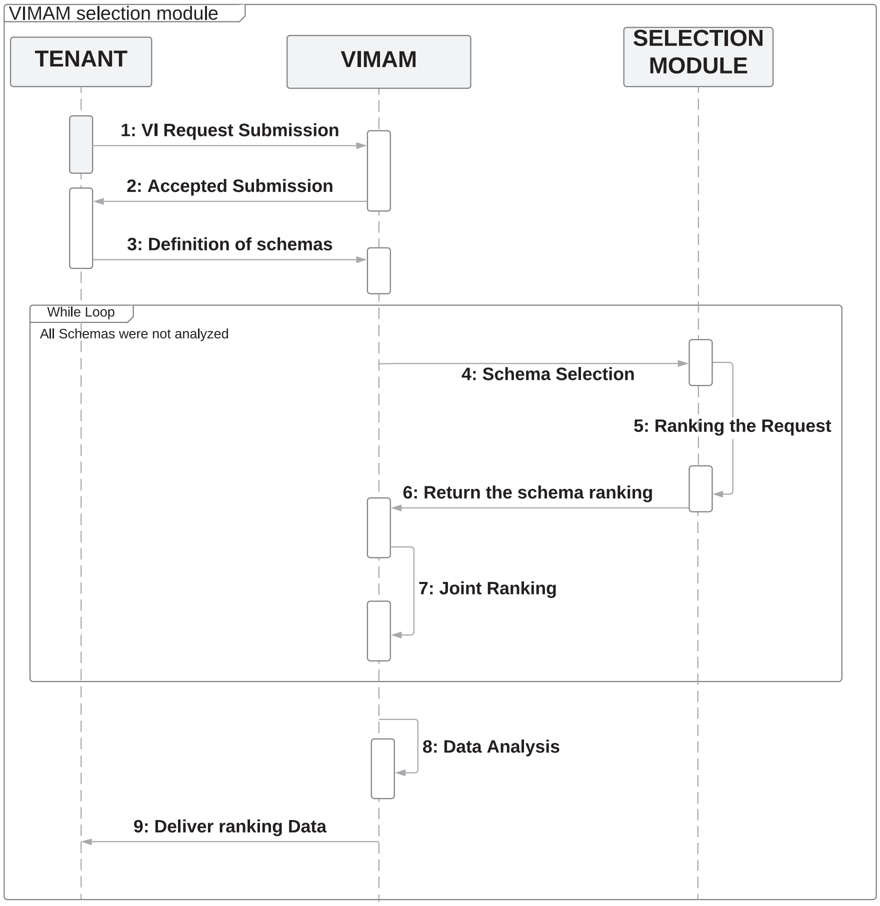


FIGURE 4 Summarized sequence diagram for decision-making on infrastructure-as-a-service provider. VI, virtual infrastructure; VIMAM, virtual infrastructure multicriteria allocation and migration

- A verification is performed before advancing to the final step. If there are schemas to be analyzed, the loop is resumed; otherwise, all data are grouped and delivered to the tenant.

The core tasks of VIMAM multicriteria selection module are the composition of the hierarchical structures, the definition of schema to weight and compare candidates, and the ranking of candidates.

4.1.1 | Hierarchical AHP structures

Two hierarchical models were composed to represent the objectives when selecting an IaaS cloud provider, describing the main characteristics and/or services addressed by them. The first one focus in the analysis of QOS, while the second one on the provisioning costs. These hierarchies are depicted in Figure 5; both models were based on CIM initiative[16] and allowed the specification of several elements that compose the VI (eg, VM, router, switch, SDN equipment, and the definition of aggregated services). Moreover, the services can be individually detailed, for example, informing the TCP/IP layer model (L1 to L4) used for composing the virtual private network of a VI.

It is worthwhile to highlight the cost hierarchy model may take into account the cost for performing a VI migration to a target provider. The motivation behind this choice is to abstract and account eventual data movement costs and QoS. Finally, a clear relationship between the components of the hierarchical models and the service currently provisioned by IaaS cloud providers is observed (services summarized in Table 1). The hierarchical representation of services and components is essential to specific the importance level of each one, performed by the definition of weighting schema.

4.1.2 | Schemas to weight and compare candidates

The identification of AHP hierarchical weighing distribution is a fundamental step since the ranking of service providers is directly related to weight policies presented in each model. Natively, VIMAM offers seven schemas (flat, security, network elasticity, QoS, web hosting, network intensive, and market research), defined based on the most common services in the IaaS cloud.[16] The broker can be easily configured to extend these options. The schemas are summarized in Table 2. The first six schemas attend to distinct tenant profiles (eg, the weights are distributed to provide higher importance on

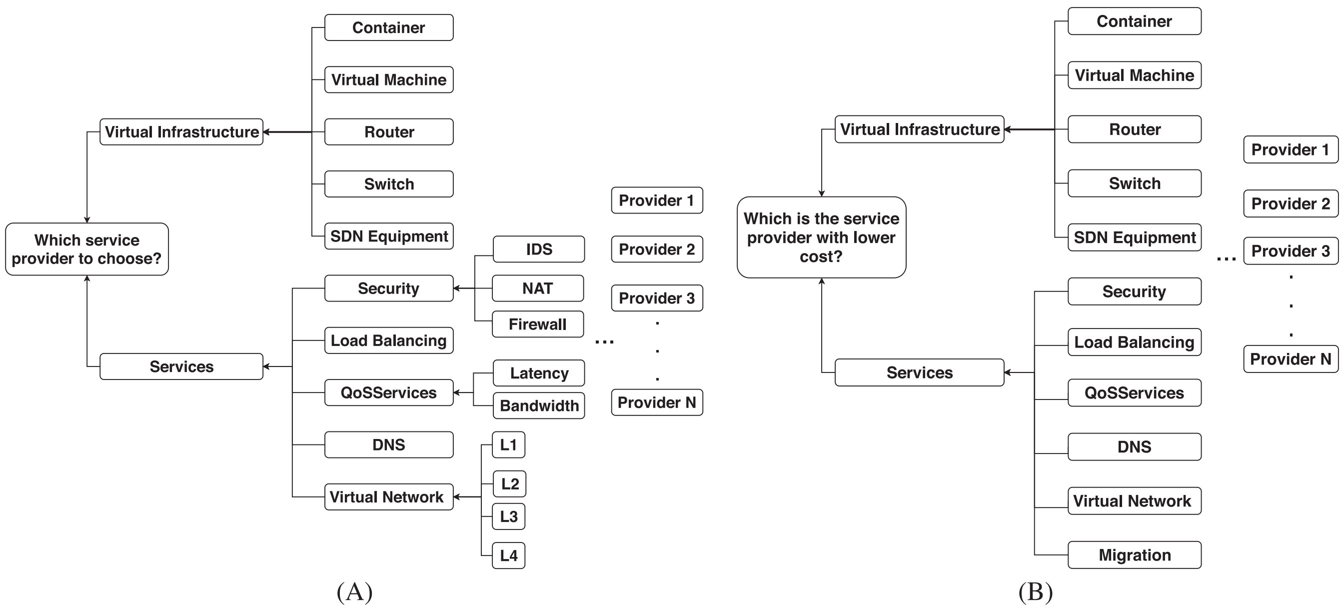


FIGURE 5 Virtual infrastructure multicriteria allocation and migration (VIMAM) quality of service (QoS) and cost hierarchy models. A, QoS hierarchy model; B, Cost hierarchy model. DNS, domain name system; IDS, intrusion detection system; NAT, network address translation; SDN, software-defined network

a determined variable), while the seventh schema was defined based on market research using a questionnaire with specialized enterprise administrators. For all scenarios depicted in Table 2, a tenant is requiring a new VI, so the migration cost (from Figure 5B) is not analyzed. Finally, for each weight schema, bold-written values indicate greater impact on provider choice.

The Flat schema represents the weight distribution in an equality form, attending to the tenants without depth knowledge about their needs (nonspecialists). The balanced distribution is recursively applied on QoS hierarchy model layers. By composing the weighting table with equal values, the AHP method processes the request given total importance to cost hierarchy model. It is worthwhile to note that even the flat schema considers network elements that are crucial to select an appropriated provider. In turn, the strategy prioritizing security proposes weights appropriated to select an IaaS provider that attend the security-related attributes. Indeed, the weighting distribution gives priority to the security node from the hierarchical AHP organization (see Figure 5A), helping tenants dealing with confidential data.

The network elasticity schema emphasis in the VI node from the hierarchical model to serve tenants with malleability requirements for network resources. In this sense, services based on SDN, latency sensitivity, and virtual networks have higher priority when compared with other attributes. It is expected that applications hosted by VI provisioned with network elasticity can absorb peaks of loads, internally implementing SDN-guided management.[62,63] Following this line of thought, the QoS schema benefits services with bandwidth reservation and latency control, focusing in applications manipulating a large volume of data or needing to keep QoS even in the presence of network variations (eg, video streaming, and stream processing). Stream process applications hosted atop VI based on QoS schema benefit from the internal service stability to offer a continuous and reliable service, fundamental requirements to improve end-user experience.[64] The configuration for web hosting represents the weighted distribution turned in security and load balancing nodes of the hierarchical organization attending to customers that work with confidential data and need network guarantee (eg, large-scale stores, financial banks, and insurer agencies). The network-intensive schema to attend distributed applications has priority on QoS, SDN, and VI nodes, representing tenants with advanced knowledge and management skills. Finally, a market research schema is proposed representing the configuration developed by a subset of administrators. A questionnaire was used and the compilation of all anonymous responses guided the weight distribution.

4.1.3 | Ranking IaaS providers

After composing the hierarchical organization and selecting one or more weighting schemas to be processed, the AHP method performs the ranking of candidates. Formally, let P represents a set of IaaS providers and S the set of schemas. The ranking vector $V_i = [p_1, p_2, \ldots, p_j]$ contains the ranking values for all providers $p_j \in P$ for a schema $i \in S$. Each

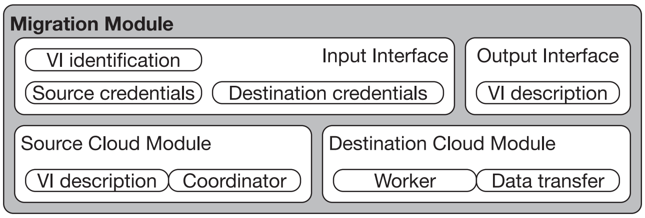


FIGURE 6 Architectural components of the migration module of the virtual infrastructure (VI) multicriteria allocation and migration–based infrastructure-as-a-service broker

position of the ranking vector V_i is calculated by three terms ($Q(p_j)$, $C(p_j)$, and $R(p_j)$) which represent the partial ranking values. In other words, $U(p_j) = \alpha_1 * Q(p_j) + \alpha_2 * C(p_j) + \alpha_3 * R(p_j)$. The first term, $Q(p_j)$, represents the ranking vector obtained from the QoS hierarchical model (see Figure 5A). This term focuses on ranking the provider $p_j \in P$ based on the quality of offered services. The second term, $C(p_j)$, represents the ranking vector from the cost structure (see Figure 5B), which ranks the providers $p_j \in P$ based in the provisioning costs. In turn, $R(p_j) = \sum_{k \in N} T(k, p_j)$ represents the ranking vector obtained from the tenant request cost for the provider $p_j \in P$, wherein $T(k, p_j)$ expresses the provisioning cost of k on provider p_j, and N denotes the set of elements in the customer requisition. All terms are in the range of values between 0 and 1, and $\alpha_1 + \alpha_2 + \alpha_3 = 1$. Finally, a matrix composed of all ranking vectors and IaaS providers is used to order the options and select the appropriated provider.

4.2 | Second phase: migrating VIs

The AHP multicriteria mechanism applied by VIMAM takes a decision based on the weighting template informed by the tenant. It is worthwhile to note that the requirements may evolve/change during the lifetime of a VI-hosted application. In this sense, the initial provider selected to host the resources may not attend the new dynamically changed expectations. Eventually, the migration of all VI resources will be necessary to move for a new provider. The components and interfaces implemented by VIMAM to achieve this goal are depicted in Figure 6.

The VIMAM migration module receives from the input interface the credentials for source and destination providers, as well as the VI identification on the source provider. Internally, the module has two separated subsets of tasks used to manage the source and destination providers.

To conduct a VI migration, a management coordinator is launched at the source service provider, while a worker is instantiated on target resources. It is worthwhile to mention that both tools are temporary launched by VIMAM to perform the migration and immediately deactivated after the VI reactivation. To access and manage the source and target providers, VIMAM requires the credentials and private keys delegation, a common trust-chain used by existing brokerage solutions.[4,19,20,65] Finally, the broker does not create or manage the client accounts in the source and destination cloud providers, and SLA establishment and negotiation are not discussed in the present work.

4.2.1 | Migration scenario

To exemplify the execution scenario of the VIMAM migration module, the VI detailed in Figure 1 will be migrated from a IaaS cloud provider A to B. As previously presented in Section 2.2, the VI is composed of a set of containers hosted by VMs to increase the management independence from provider's optimizations and tools. All resources composing the VI are interconnected by a private and isolated virtual network, which must be jointly migrated to the destination provider. Figure 7 shows VIMAM access to the source and target providers.

Initially (see Figure 7), the coordinator is instantiated at IaaS provider A to retrieve the VI specification detailing the VMs, containers, and the private network topology (router, switches, and IPs). In parallel, the worker is initialized at IaaS provider B to communicate with the coordinator and prepare the migration scenario. At this point, the VMs are recreated at IaaS provider B to prepare the container migration.

After instantiation the target scenario (depicted in Figure 8), the worker starts the virtual network configuration recreating the virtual routers, switches, and topology. Latter, the virtual network is prepared with the IPs configuration retrieved from IaaS provider A. Finally, the worker performs the migration of all containers and volumes attached following the VM-container mapping originally defined and informed by the coordinator.

Since the container and volume migration are finished, they are restored at the destination VM, as indicated in Figure 9. Here, the VI migration is concluded, and the resources allocated on IaaS provider A can be released.

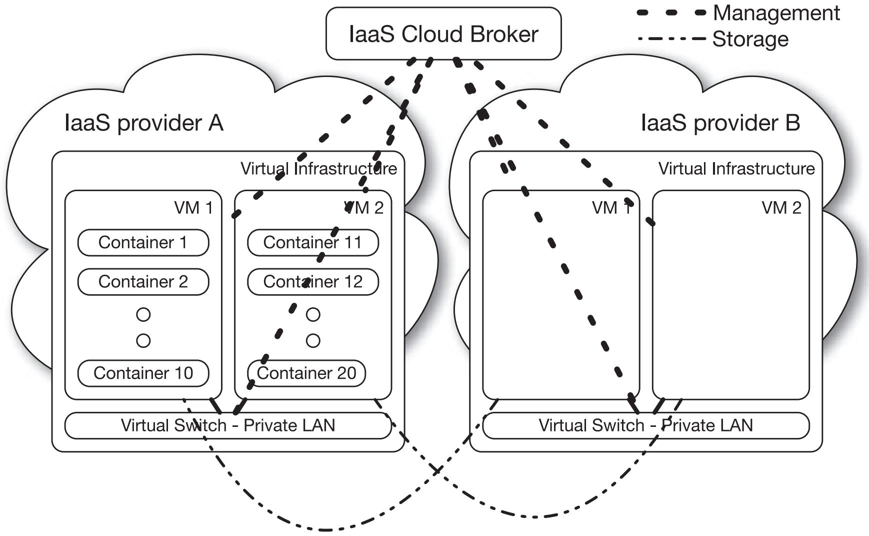


FIGURE 7 Virtual infrastructure (VI) multicriteria allocation and migration–based broker migration module: preparing a VI migration. IaaS, infrastructure as a service; LAN, local area network; VM, virtual machine

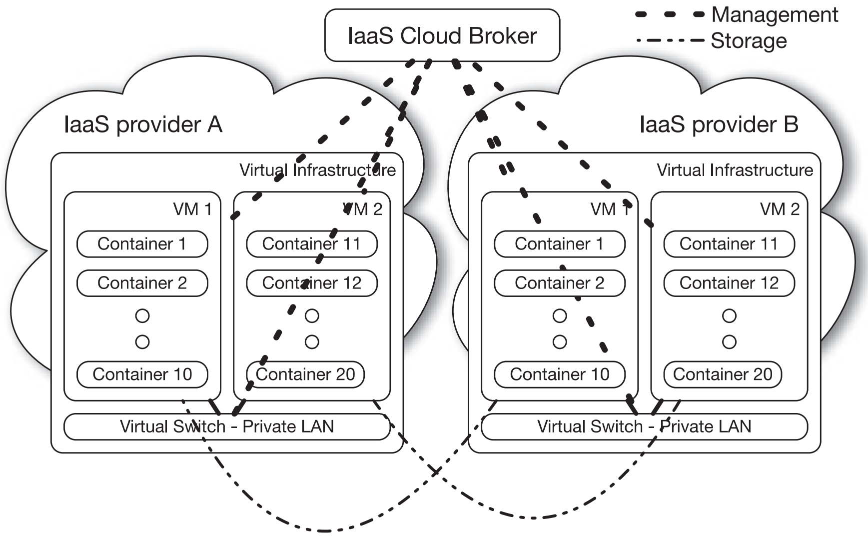


FIGURE 8 Virtual infrastructure (VI) multicriteria allocation and migration–based broker migration module: migrating containers and volumes. IaaS, infrastructure as a service; LAN, local area network; VM, virtual machine

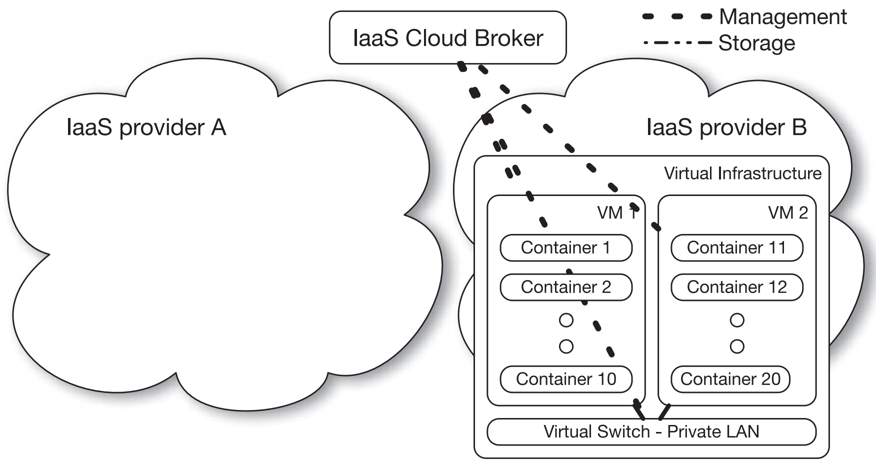
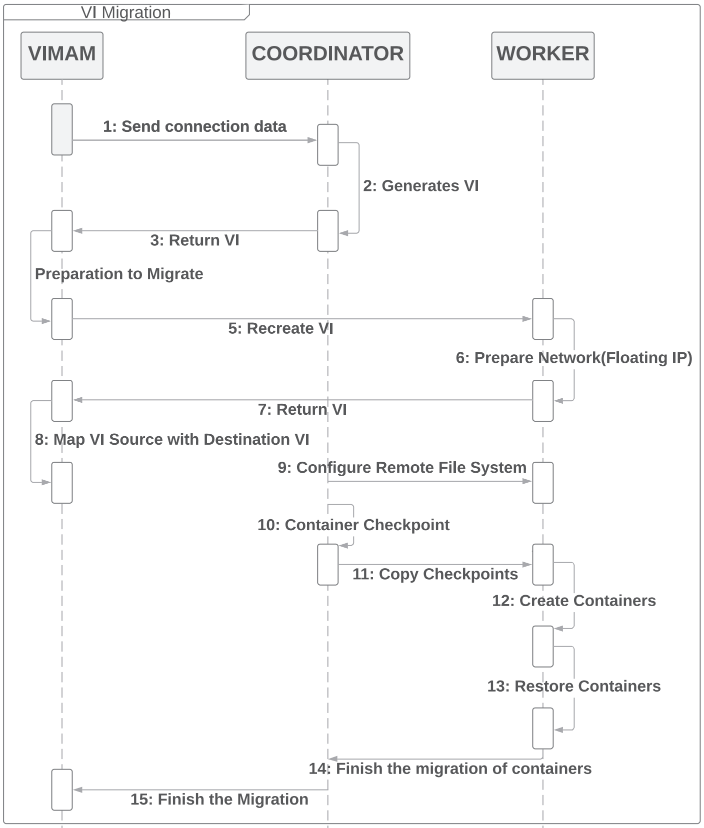


FIGURE 9 Virtual infrastructure (VI) multicriteria allocation and migration–based broker migration module: conclusion of the VI migration. IaaS, infrastructure as a service; LAN, local area network; VM, virtual machine

4.2.2 | Sequence diagram and algorithms to migrate a VI

Formally, the VI migration follows the description of VIMAM, coordinators, and workers, as depicted in Figure 10. The main steps of sequence diagram are listed in Figure 10A, while the algorithms of the VIMAM, coordinator, and worker are depicted in Figure 10B.



(A)

Algorithm 1: VIMAM pseudocode.

```
Data: P_src; P_dst; VI_id
template = get_vi_specification(P_src, VI_id);
target_vms = create_hosts(VI_id, template, P_dst);
network = create_network(VI_id, template, P_dst);
for ∀ dst_hosts ∈ target_hosts do
    dst_host_ip = get_host_ip(dst_hosts);
    src_host = get_src_host(template, dst_hosts);
    src_host_ip = get_host_ip(src_host);
    launch_coordinator(P_src, src_host_ip, dst_host_ip);
    launch_worker(P_dst, dst_host_ip);
for ∀ dst_hosts ∈ target_hosts do
    join_algorithm(P_dst, dst_host);
if migration_accomplished(P_dst) then
    release_vi(P_src, VI_id)
else
    release_vi(P_dst, VI_id)
```

Algorithm 2: Coordinator pseudocode.

```
Data: P_src, dst_host_ip
nfs_path = mount_remote_file_system(dst_host_ip);
for ∀ container ∈ get_containers() do in parallel
    volumes = get_volumes_checkpoints (container, file_server_path);
    checkpoint = container_checkpoint(container, file_server_path);
```

Algorithm 3: Worker pseudocode.

```
Data: P_dst, src_host_ip
nfs_path = export_remote_file_system_server_to(src_host_ip);
wait_containers_migration(src_host_ip);
for ∀ container ∈ nfs_path do in parallel
    volumes = get_volumes_checkpoints (container, nfs_path);
    create_container(container, volumes);
```

(B)

FIGURE 10 Virtual infrastructure (VI) multicriteria allocation and migration (VIMAM) main steps to migrate a VI. A, Sequence diagram of VIMAM actions to migrate a VI; B, Algorithms used on VIMAM, coordinator, and worker. IP, Internet Protocol

The main steps of sequence diagram (see Figure 10A) are enumerated from 1 to 15. VIMAM starts the migration process (step 1) connecting to the source cloud and starting the coordination module (step 2). The VM specification, network topology, private IP configuration, instance and containers images, and other metadata are retrieved from the source provider. The VIMAM migration module concentrates and verifies this data (steps 3 and 4). At this point, all information regarding the target provider (network and credentials) are validated and if a failure is identified, the process can be aborted. Latter, VIMAM launches all workers (step 5), in charge of creating VMs and private network configuration, associating floating and Internet-routable IPs (step 6). On step 7, the worker returns the status of remotely performed operations.

VIMAM constructs a source-destination IPs mapping to create a remote file-system (steps 8 and 9). Once the remote file system is composed and synchronized, the containers checkpointing starts (step 10), and the copy is performed to target provider (step 11). After concluding the images, volumes, and checkpoints to target VMs the containers are restored (steps 12 and 13). At this point, the service is totally enabled on the target IaaS provider. It is worthwhile to note that the time between the last checkpointing at the source cloud, and the containers restoration denotes the downtime for a VI migration. Finally, when the VI migration is concluded (steps 14 and 15), the original resources are released.

5 | EXPERIMENTAL ANALYSIS

The experimental evaluation of the selection and migration modules of VIMAM is discussed in this section. Specifically, Section 5.1 shows the efficiency of the AHP-based selection mechanism, while Section 5.2 demonstrate how a decision of reallocation is performed using an OpenStack and Docker–based prototype.

TABLE 3 Monthly unit costs of resources composing a virtual infrastructure for four public infrastructure-as-a-service providers

Alternative	VM/Container (Unit)	CPU (Unit)	RAM (GB)	Storage (GB)
Rackspace	$23.8	$3.75	$3.75	$0.032
Microsoft Azure	$23.4	$5.1	$4.87	$0.06
Amazon Web Services	$24.43	$7.9	$8.64	$0.042
Google Cloud	$24.27	$9.1	$3.35	$0.054

Abbreviations: VM, virtual machine.

5.1 | Efficiency of AHP-based selection module for selecting IaaS providers

The selection module of VIMAM was implemented on C++11, and executed on a computer configured with 16-GB RAM DDR4, Intel i7 processor, running the GNU/Linux ArchLinux OS. To measure the efficiency of the selection module, the experimental analysis was developed in two steps. The first step consists of an analysis on provider selection frequency, and AHP ranking values are presented. The second step is a comparison based on public market share data. The AHP weighting tables proposed by fuzzy AHP (FAHP)[13] were selected for composing the baseline for comparison. The six criteria originally identified by FAHP (cost, availability, storage capacity, CPU capacity, performance indicator, and security) were slightly adapted to fit the VIMAM model. However, it is worthwhile to mention that the weights for FAHP remained as originally defined.

5.1.1 | Frequency of selection and ranking values

Initially, 1000 VI requests were submitted to VIMAM. Each request is composed of a set of containers (represented by CPU, RAM, and storage configuration). For sake of simplicity, each container is allocated on a single VM, and the number of containers composing a VI is uniformly selected between 1 and 10. Regarding the containers requirements, the parameters are uniformly distributed following predefined ranges: CPU is defined between 1 and 20, RAM is selected from the $[1, 128]$ interval, and finally the volume size is selected in $[1, 102\,400]$. A subset of four well-known public IaaS cloud providers composed the alternatives for hosting the VI requests. Each provider is analyzed regarding the predefined weighting schemas (detailed in Table 2). The cost of each request is calculate considering the monthly values present in Table 3 and ranking values obtained from the QoS hierarchical model (detailed in Section 4.1.1) applying $\alpha_1 = 0.3$, $\alpha_2 = 0.2$, and $\alpha_3 = 0.5$ (as detailed in Section 4.1.3). The α vector configuration represent tenants concerned to the provisioning cost, ie, a provisioning on public cloud providers.

Figures 11 and 12 summarize the results for the frequency of provider selection and ranking values, respectively.

Initially, Figure 11 indicates that the provider's selections is direct impact from the weighting schema, highlighting the importance of detailing the VI specification and simultaneously apply multicriteria mechanisms. A simple and intuitive flat schema results on high frequency of selection for two providers. However, when a weighting schema is selected the tenant, VIMAM can indicate appropriated options. For example, a VI-hosted application requiring a network-intensive schema that will select an IaaS provider mostly ignored by the flat schema. In addition, the weights used by FAHP are

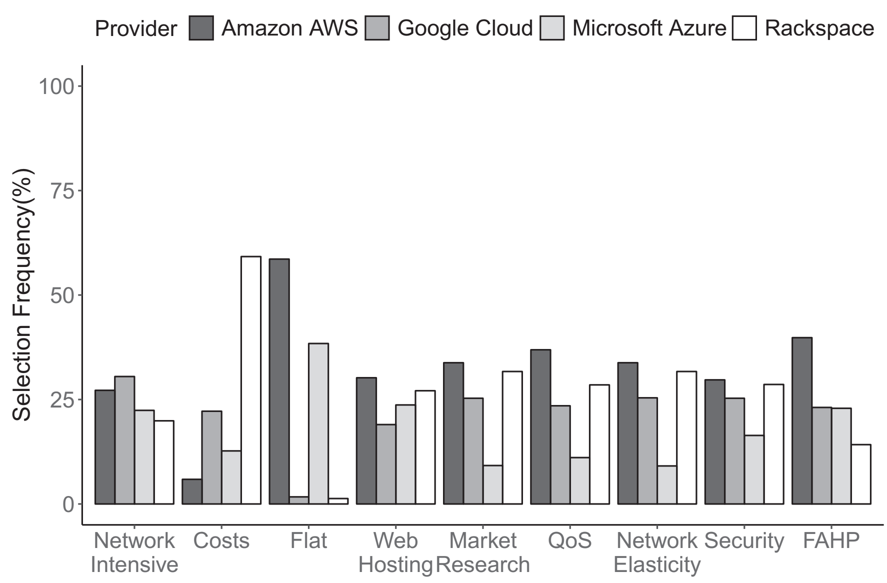


FIGURE 11 Frequency of provider selection according to a predefined weighting schema. QoS, quality of service; FAHP, fuzzy analytic hierarchy process

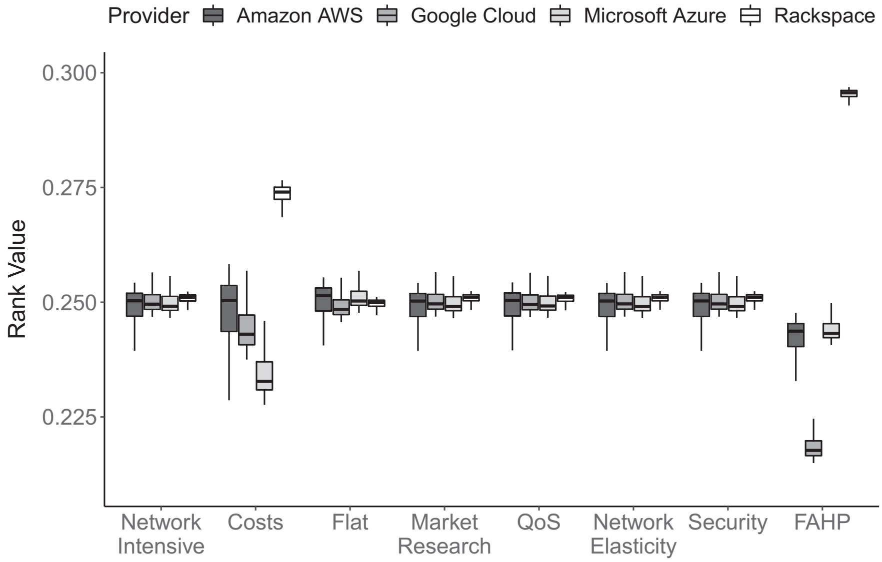


FIGURE 12 Ranking values for each provider considering the predefined weighting schemas. QoS, quality of service; FAHP, fuzzy analytic hierarchy process

mainly driven by the provisioning cost, given preference to the AWS provider. The variation of ranking values used to compute the frequency of provider selection is presented in Figure 12.

It is possible to note that the AWS provider has the bigger variation (see Figure 12), which is independent of the applied scenario, while the other providers show a lower variation in the values. The variation between the providers ranking is minimal among different schemas defined in Table 2 or proposed by FAHP. In addition, the reduced number of attributes analyzed by the FAHP schema resulted on distinct rank values for all providers, a characteristic which is softened when a large number of attributes are applied (from Table 2).

5.1.2 | Comparison with market-share data

To show the applicability of VIMAM predefined weighting schemas facing the real market share, a bibliographic research was carried out to identify which are the effective market participation of each listed service provider.[66-68] Figure 13 presents our research outcome.

Regarding the Microsoft Azure provider (see Figure 13), the VIMAM value are consistent with the real market share. However, there is a significant difference between the results obtained for the AWS, Google Cloud, and Rackspace providers. For Google Cloud and Rackspace providers, VIMAM increased the selection frequency. Indeed, the variation is justified as the market-share indicators ignore most of the variables that impact the decision-making of selecting a provider, providing almost intuitive (flat) results. Finally, it is worthwhile to mention in the VIMAM selection module analysis that the AWS provider corresponds to a total of 32.9% of the selections, followed by Rackspace with 30.5%, Google Cloud with 18.9%, and Microsoft Azure with 17.7% (see Figure 13).

The results demonstrate that the IaaS provider selection can evolve with the management knowledge of VI-hosted application. A provider initially selected by a flat approach may not be effective in provisioning the VI. In other words, VIMAM can identify and offer to tenants other options for hosting the VI infrastructure. If the tenant accepts the proposal,

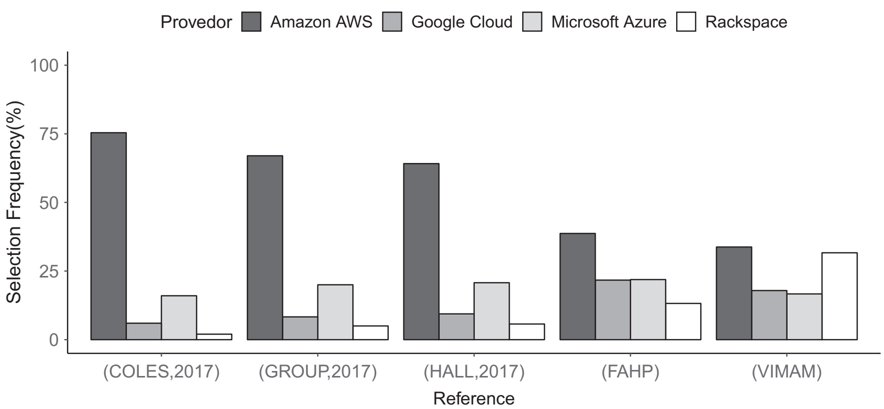


FIGURE 13 Comparison between the analytic hierarchy process of the virtual infrastructure multicriteria allocation and migration–broker results and the market share

a VI migration is performed. In this sense, Section 5.2 details the implementation of a VIMAM migration prototype, and Section 5.3 discusses the evaluation results.

5.2 | OpenStack and Docker–based prototype for VI migration

The tenant's perspective regarding the VI-hosted application requirements or providers' billing policies may change over time. Eventually, the VIMAM selection module will inform that a better option is available to host the VI. Thus, a VI migration must be performed to move computing, storage, and networking resources to a new provider (region or zone). We present an OpenStack and Docker–based prototype for the VI migration module of VIMAM. The OpenStack Cloud Management Plataform (CMP) manages and controls pools of compute, storage, and networking resources applied to public or private clouds.[69] Among the projects composing the OpenStack, the Heat is a project responsible for VI orchestration and is essential for recreating a VI obtained by the Flame[§] tool.

Technically describing the sequence diagram discussed in Subsection 4.2.2, the first challenge is to obtain the current VI composition including its network connections and dependencies. The OpenStack Heat project describes a VI through a Heat orchestration template (HOT) model, and enables the dynamic reload of a specification. Therefore, the VIMAM migration prototype extracts all VI specification from current cloud provider (using Flame) and recreates the VI on a target provider (using OpenStack Heat/HOT). After recreating the VI on the target provider, the container migration is started.

In turn, Docker is a framework to deliver and manage containers in standalone or cloud environments.[70] Conceptually, containers are ephemeral environments which can be created, destroyed, and recreated several times, without offering a persistent storage system. However, Docker works with volume as an alternate use for persistent data. A volume can be associated to a new container allowing an application to be restarted. Thus, all active volumes must be migrated along containers. Thus, to perform this operation, the VIMAM migration prototype relies on the Convoy[¶] tool set to make volumes snapshots of a migrating container hosted on the source cloud provider. These volume snapshots are available to the destination cloud provider by a classical network file system (NFS) configuration.

The VIMAM migration prototype is based on the following: Docker version 18.03.0, Community Edition; Convoy version 0.5.0; Python version 2.7.6; VMs running GNU/Linux Ubuntu 14.04 Trusty with one vCPU and 2 GB of RAM; Secure Shell clients and servers installed into VMs; and OpenStack CMP version Pike. It is worthwhile to mention that the OpenStack and Docker–based implementation is a proof-of-concept prototype. The VIMAM broker architecture described in Section 4.2 can be further implemented to address others cloud providers and technologies.

5.3 | Evaluation of the VIMAM migration prototype

We defined two scenarios to evaluate the VIMAM migration prototype on real cloud providers:

1. Baseline scenario quantifies the minimum time for migrating a VI composed of a single container.
2. Migration of VI composed of one switch, two VMs, and 20 containers (ten containers per VM).

The CloudLab infrastructure[71] was selected to support the experimental analysis. CloudLab is a large-scale testbed in which researchers can deploy their own isolated clouds, with administrative privileges, to install multiple versions of OpenStack. In our scenario, the source and destination cloud providers were interconnected by a wide-area network with round-trip time of 51.853 ms. Specifically, the source cloud provider is hosted at Utah site while the destination is at South Carolina. Regarding the metrics to analyze VIMAM migration module, we collected the total migration time and the network throughput. Total time is stratified by each migration step performed by the VIMAM module (worker and coordinator tasks): *retrieve VI specification*, *migrate VI components*, *wait services*, *check containers*, *check volumes*, and *create checkpoint*. The results are depicted in Figures 14, 15, and 16, and the cumulative distribution function graphs present the average of ten samples. Initially, the total migration time for the baseline scenario (VI composed of a single container) is 3.41 s.

Based on Figure 14, we can note the six steps performed by the coordinator (executing at the source cloud provider). The x-axis (see Figure 14) represents the time in seconds, while the y-axis demonstrates the probability of each step being completed. Analyzing the coordinator results, the total time is about 34 s, and it is worthwhile to note that four of the six steps end in less than 3 s. The *wait services* and *migrate VI components* steps are the most time-consuming, spending about 4.5 and 14 s, respectively. In common, both steps interact with the work tool (from the destination provider).

[§]Available at https://github.com/openstack/flame
[¶]Available at https://github.com/rancher/convoy

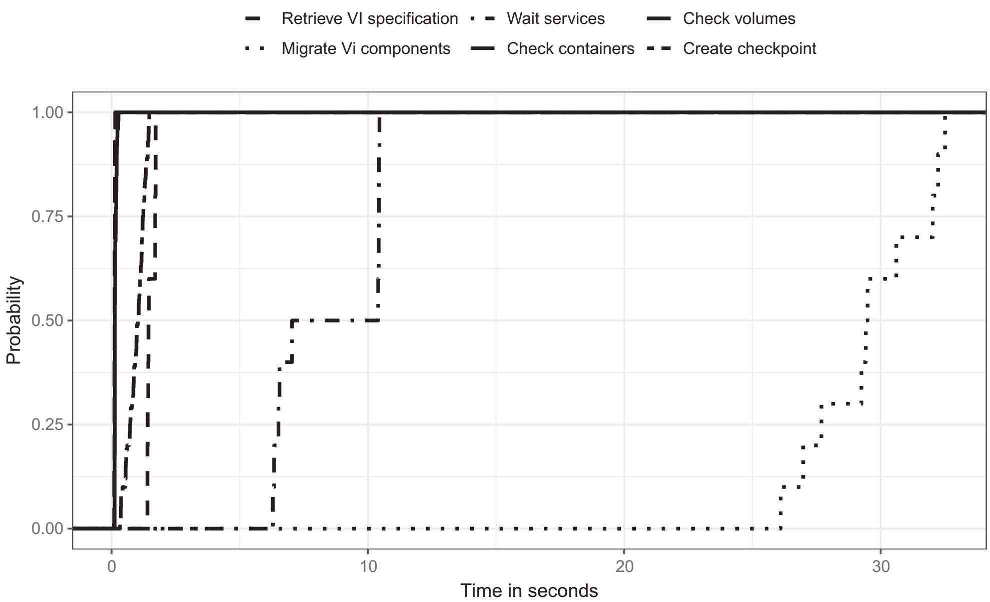


FIGURE 14 Execution time of virtual infrastructure migration steps executed on source provider (coordinator)

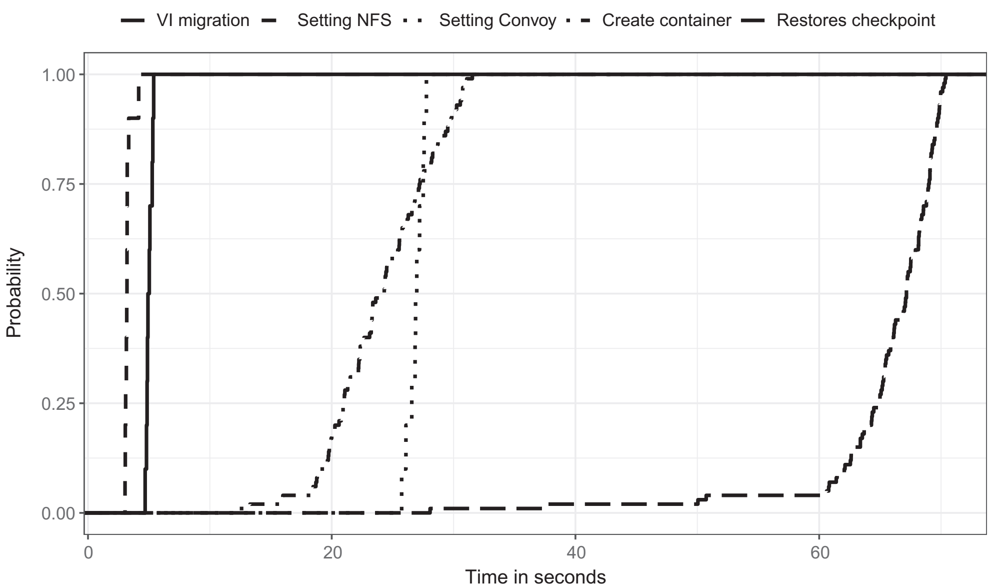


FIGURE 15 Execution time of virtual infrastructure migration steps executed by the worker (destination cloud)

The worker execution on destination provider, depicted by Figure 15, is composed of five steps: VI *migration*, *setting* NFS, *setting convoy*, *create container*, and *restore container*.

Analyzing the results, the total time is about 67 s, and it is worthwhile to observe that *setting* NFS and VI *migration* tasks have the shortest time, with 7 s, in the worst case. The *setting convoy* reached values between 25 and 30 s, while the *create container* between 12 and 35 s. However, the *create checkpoint* has unstable values between 30 and 60 s. The highest degree of probability indicates a running time close by 60 s. This difference is explained by the status of the last checkpoint. If all data of the last checkpoint is already in the destination cloud provider, this step spends about 30 s. Otherwise, the worker must download the checkpoint data and may spend up to 60 s.

Furthermore, in Table 4, we present the proportion of migration time according to coordinator and worker steps. The steps executed by the worker are time consuming when compared with the steps performed by the coordinator, and comprises 75.3% of all the steps. Specifically for the worker, the highlights are *setting convoy*, *create container*, and *restore container*, all with more than 14%. However, in the coordinator, we only identified one step with more than 14%, the *migrate* VI *components*.

The steps with less than 1% are *retrieve* VI *specification*, *create checkpoint*, *check container*, and *check volumes*, all executed by the coordinator.

TABLE 4 Virtual infrastructure (VI) migration steps executed by coordinator and workers

Coordinator		Worker	
Create model	0.9%	VI's migration	3.0%
Wait for VI's migration	17.9%	Setting NFS	2.0%
Wait services	5.1%	Setting Convoy	16.2%
Create checkpoint	0.6%	Create container	14.5%
Check container & volumes	< 0.2%	Restore container	39.6%
Total	24.7%	**Total**	75.3%

Abbreviations: NFS, network file system.

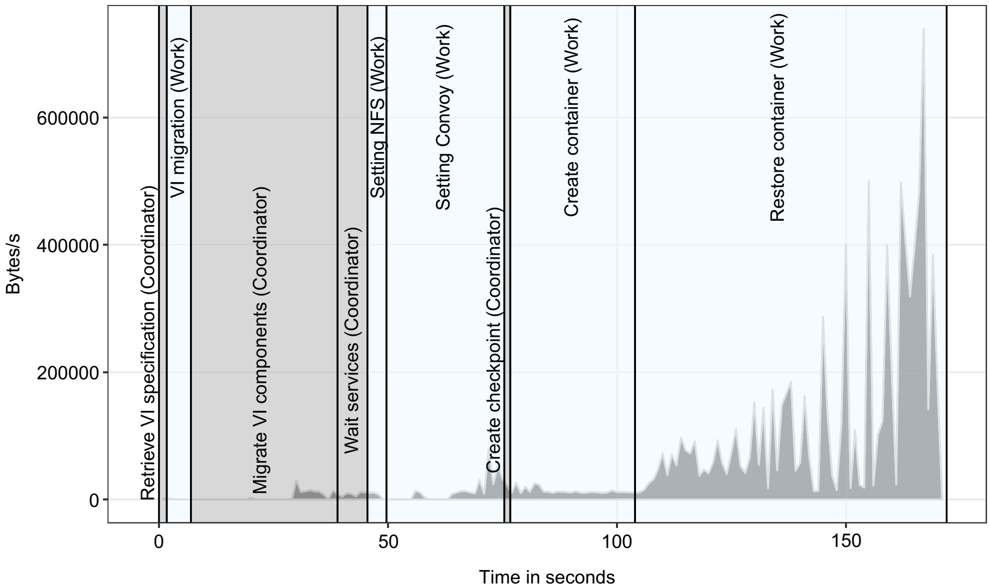


FIGURE 16 Network throughput (bytes/s) during a virtual infrastructure (VI) migration. NFS, network file system [Colour figure can be viewed at wileyonlinelibrary.com]

Finally, we chose a random VI migration to discuss the network throughput for each step, as detailed in Figure 16. On the *x*-axis, the graph presents the time in seconds, and describe the steps executed by the coordinator and worker in chronological order. On the *y*-axis, the network throughput (Bytes/s) is summarized. The results showed *restore container* is the largest network consuming step (peak of 594 500 bytes/s). This step downloads the container checkpoints (by NFS) from the source cloud provider. All the other steps have negligible network consumption (224 to 76 780 bytes/s).

The last aspect analyzed is the VI migration cost. During the VI migration, VIMAM maintains a temporary VI replica in the destination provider. The replica is populated with data from VMs and containers during the migration process. This replica will become the official VI after the migration is finished. In this sense, the migration cost must take into account the temporary resources provisioned as well as the data transfer amount between providers.

Based on the monthly unit cost of resources composing a VI for four public IaaS providers, depicted in Table 3, the migration cost can be obtained by the Equation (1) in which $replica_{cost}$ and *migration_time* represent the cost of temporary VI resources. In addition, another migration cost component is the data transfer from source to destination provider (given by *data_trans*).

$$migration_{cost} = (replica_{cost} * migration\_time) + data\_trans. \quad (1)$$

Since the CloudLab infrastructure is an open-source platform, we arbitrary used the Amazon EC2 price list[#] to estimate the migration cost for our experimental scenario. Specifically, the $replica_{cost}$ is defined by two VM *t2.2xlarge* (8 vCPU, 32GB of memory, moderate network performance (< 5 Gbps) and GNU/Linux OS). The *migration_time* is a few seconds, but on Amazon EC2, the minimum on-demand contract for VM is one hour. For this instance, the network payment is per available bandwidth, so *data_trans* cost is zero on this scenario. The VM cost to *t2.2xlarge* is US$ 0.3712 per hour, and we needed two for hosting our containers. Therefore, $migration_{cost}$ based on Amazon EC2 prices is US$ 0.7424, for 20 containers hosted atop two VM. It is worthwhile to highlight that this scenario migration cost represents less than 1% monthly cost of VI. Finally, the VIMAM migration prototype is operational and allows the automatic VI migration

[#] https://aws.amazon.com/ec2/pricing/

in real-world environment. The VI migration time is acceptable for sporadic migrations, motivated by administrative decisions.

6 | CONSIDERATIONS AND FUTURE WORK

Several relevant aspects were revealed in this article through the discussion of a cloud broker based on MCDM and VI migration. The first aspect is related with the selection of IaaS providers to host virtualized environments. In addition to the computational challenge of the problem NP-hardness, the selection is guided by a tenant-specific policy, which must be considered along the process. The VIMAM selection module demonstrates how different weighting scheme can be applied to represent the tenants' perspectives. Indeed, the experimental analysis pointed out the variability on IaaS frequency selection, demonstrating tenants with deep knowledge on VI requirements can select service-tailored providers. Another relevant aspect is that demonstrating the VIMAM migration module is effective to move VIs composed of VMs, containers, switches, and specific network configuration. Our prototype based on OpenStack and Docker was able to move the virtualized resources between two IaaS cloud providers.

Conceptually, a weak point of VIMAM broker is the management of failures during the VI migration process. VIMAM needs to complete the migration of all resources to identify if a problem occurred. A future research work will be to develop fault-tolerant mechanisms in the selection and migration modules. However, despite the aforementioned limitations, we believe VIMAM advanced the state of the art by jointly discussing the IaaS provider selection and the migration of VIs, considering the challenges introduced by containers and virtual network management.

ACKNOWLEDGEMENTS

The authors would like to thank the support provided by LabP2D (http://labp2d.joinville.udesc.br), FAPESC, CAPES, and UDESC.

ORCID

Guilherme P. Koslovski https://orcid.org/0000-0003-4936-1619

REFERENCES

1. Mell PM, Grance T. *SP 800-145: The NIST Definition of Cloud Computing*. Technical report. Gaithersburg, MD: National Institute of Standards and Technology; 2011.
2. Liu F, Tong J, Mao J, Messina J, Bohn R. *NIST Cloud Computing Reference Architecture: Recommendations of the National Institute of Standards and Technology (Special Publication 500-292)*. Scotts Valley, CA: CreateSpace Independent Publishing; 2012.
3. Rodriguez MA, Buyya R. Container-based cluster orchestration systems: a taxonomy and future directions. *Softw: Pract Exper*. 2018;49(5):698-719. https://doi.org/10.1002/spe.2660
4. Zhang Z, Wu C, Cheung DWL. A survey on cloud interoperability: taxonomies, standards, and practice. *ACM SIGMETRICS Perform Eval Rev*. 2013;40(4):13-22. https://doi.org10.1145/2479942.2479945
5. CloudSquare. Research and compare cloud providers and services: CloudHarmony. 2017. https://cloudharmony.com/status-group-by-regions-and-provider
6. Yousafzai A, Gani A, Noor RM, et al. Cloud resource allocation schemes: review, taxonomy, and opportunities. *Knowl Inf Syst*. 2017;50(2):347-381. https://doi.org/10.1007/s10115-016-0951-y
7. Hinz M, Koslovski GP, Miers CC, Pilla LL, Pillon MA. A cost model for IaaS clouds based on virtual machine energy consumption. *J Grid Comput*. 2018;16(3):493-512 . https://doi.org/10.1007/s10723-018-9440-8
8. Fischer A, Botero JF, Beck MT, de Meer H, Hesselbach X. Virtual network embedding: a survey. *IEEE Commun Surv Tutor*. 2013;15(4):1888-1906.
9. Garg SK, Versteeg S, Buyya R. A framework for ranking of cloud computing services. *Future Gener Comput Syst*. 2013;29(4):1012-1023.
10. Saaty TL, Vargas LG. *Models, Methods, Concepts & Applications of the Analytic Hierarchy Process*. New York, NY: Springer Science & Business Media; 2012. *International Series in Operations Research & Management Science*; vol. 175.
11. Kaur K, Sharma S, Kahlon KS. Interoperability and portability approaches in inter-connected clouds: a review. *ACM Comput Surv*. 2017;50(4). Article No. 49. https://doi.org/10.1145/3092698
12. Razavi M, Aliee FS, Badie K. An AHP-based approach toward enterprise architecture analysis based on enterprise architecture quality attributes. *Knowl Inf Syst*. 2011;28(2):449-472. https://doi.org/10.1007/s10115-010-0312-1
13. Tanoumand N, Ozdemir DY, Kilic K, Ahmed F. Selecting cloud computing service provider with fuzzy AHP. In: Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE); 2017; Naples, Italy.

14. Whaiduzzaman M, Gani A, Anuar NB, Shiraz M, Haque MN, Haque IT. Cloud service selection using multicriteria decision analysis. *Sci World J*. 2014;2014.
15. Saaty TL. How to make a decision: the analytic hierarchy process. *Eur J Oper Res*. 1990;48(1):9-26.
16. Force DMT. Common information model. 2017. http://www.dmtf.org/standards/cim
17. Hong C-Y, Mandal S, Al-Fares M, et al. B4 and after: managing hierarchy, partitioning, and asymmetry for availability and scale in Google's software-defined WAN. In: Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication (SIGCOMM'18); 2018; Budapest, Hungary.
18. Hofmann P, Woods D. Cloud computing: the limits of public clouds for business applications. *IEEE Internet Comput*. 2010;14(6):90-93. http://doi.org/10.1109/MIC.2010.136
19. Grozev N, Buyya R. Inter-cloud architectures and application brokering: taxonomy and survey. *Softw: Pract Exper*. 2014;44(3):369-390. http://doi.org/10.1002/spe.2168
20. Toosi AN, Calheiros RN, Buyya R. Interconnected cloud computing environments: challenges, taxonomy, and survey. *ACM Comput Surv*. 2014;47(1). Article No. 7. http://doi.org/10.1145/2593512
21. García AL, del Castillo EF, Fernández PO. Standards for enabling heterogeneous IaaS cloud federations. *Comput Stand Interfaces*. 2016;47:19-23. http://doi.org/10.1016/j.csi.2016.02.002
22. Zhao J-F, Zhou J-T. Strategies and methods for cloud migration. *Int J Autom Comput*. 2014;11(2):143-152. https://doi.org/10.1007/s11633-014-0776-7
23. Jamshidi P, Ahmad A, Pahl C. Cloud migration research: a systematic review. *IEEE Trans Cloud Comput*. 2013;1(2):142-157.
24. Medina V, García JM. A survey of migration mechanisms of virtual machines. *ACM Comput Surv*. 2014;46(3). Article No. 30. https://doi.org/10.1145/2492705
25. Anhalt F, Koslovski G, Primet PV-B. Specifying and provisioning virtual infrastructures with HIPerNET. *Int J Netw Manag*. 2010;20(3):129-148. https://doi.org/10.1002/nem.732
26. Bhardwaj S, Jain L, Jain S. Cloud computing: a study of infrastructure as a service (IAAS). *Int J Eng Inf Technol*. 2010;2(1):60-63.
27. de Oliveira R, Koslovski GP. A tree-based algorithm for virtual infrastructure allocation with joint virtual machine and network requirements. *Int J Netw Manag*. 2016;27(1):e1958. https://doi.org/10.1002/nem.1958
28. Pfitscher RJ, Pillon MA, Obelheiro RR. Customer-oriented diagnosis of memory provisioning for IaaS clouds. *ACM SIGOPS Oper Syst Rev*. 2014;48(1):2-10. http://doi.acm.org/10.1145/2626401.2626403
29. Raugust AS, de Souza FR, Pillon MA, Miers CC, Koslovski GP. Allocation of virtual infrastructures on multiple IaaS providers with survivability and reliability requirements. In: Proceedings of the 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA); 2018; Krakow, Poland.
30. Pettey C, van der Meulen R. *Gartner Says Cloud Consumers Need Brokerages to Unlock the Potential of Cloud Services*. Technical report. Stamford, CT: Gartner; 2009.
31. Nair SK, Porwal S, Dimitrakos T, et al. Towards secure cloud bursting, brokerage and aggregation. In: Proceedings of the 2010 8th IEEE European Conference on Web Services; 2010; Ayia Napa, Cyprus.
32. Kang D, Son S, Kim J. Design of cloud service brokerage system intermediating integrated services in multiple cloud environment. *Int J Comput Electr Autom Control Inf Eng*. 2014;8(12):2134-2139.
33. Li W, Tordsson J, Elmroth E. Modeling for dynamic cloud scheduling via migration of virtual machines. In: Proceedings of the 2011 IEEE 3rd International Conference on Cloud Computing Technology and Science; 2011; Athens, Greece.
34. Tordsson J, Montero RS, Moreno-Vozmediano R, Llorente IM. Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers. *Future Gener Comput Syst*. 2012;28(2):358-367. https://doi.org/10.1016/j.future.2011.07.003
35. Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I. Cloud computing and emerging it platforms: vision, hype, and reality for delivering computing as the 5th utility. *Future Gener Comput Syst*. 2009;25(6):599-616.
36. Rackspace. Rackspace. 2017. https://www.rackspace.com/
37. Microsoft. Microsoft Azure. 2017. https://azure.microsoft.com/pt-br/
38. AWS. Amazon Web Services. 2018. https://aws.amazon.com
39. Inc. G. Products & services | Google Cloud. 2018. https://cloud.google.com
40. Ruck DB, Pillon MA, Miers CC, Koslovski GP. EAVIRA: energy-aware virtual infrastructure reallocation algorithm. In: Proceedings of the 2017 VII Brazilian Symposium on Computing Systems Engineering (SBESC); 2017; Curitiba, Brazil.
41. Akoush S, Sohan R, Rice A, Moore AW, Hopper A. Predicting the performance of virtual machine migration. In: Proceedings of the 2010 IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems; 2010; Miami Beach, FL.
42. Deshpande U, Keahey K. Traffic-sensitive live migration of virtual machines. *Future Gener Comput Syst*. 2017;72:118-128. https://doi.org/10.1016/j.future.2016.05.003
43. Clark C, Fraser K, Hand S, et al. Live migration of virtual machines. In: Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation - Volume 2 (NSDI'05); 2005; Boston, MA.
44. Mashtizadeh AJ, Cai M, Tarasuk-Levin G, Koller R, Garfinkel T, Setty S. XvMotion: unified virtual machine migration over long distance. In: Proceedings of the 2014 USENIX Annual Technical Conference (USENIX ATC 14); 2014; Philadelphia, PA.
45. Hines MR, Deshpande U, Gopalan K. Post-copy live migration of virtual machines. *ACM SIGOPS Oper Syst Rev*. 2009;43(3):14-26. https://doi.org/10.1145/1618525.1618528

46. Zhang F, Fu X, Yahyapour R. CBase: a new paradigm for fast virtual machine migration across data centers. In: Proceedings of the 2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing; 2017; Madrid, Spain.
47. Karmel A, Chandramouli R, Iorga M. *SP 800-180 (Draft): NIST Definition of Microservices, Application Containers and System Virtual Machines*. Technical report. Gaithersburg, MD: National Institute of Standards and Technology; 2016.
48. Rajagopalan S, Cully B, O'Connor R, Warfield A. SecondSite: disaster tolerance as a service. *ACM SIGPLAN Notices*. 2012;47(7):97-108.
49. Wood T, Ramakrishnan KK, Shenoy P, van der Merwe J. CloudNet: dynamic pooling of cloud resources by live WAN migration of virtual machines. *ACM SIGPLAN Notices*. 2011;46(7):121-132.
50. Yazir YO, Matthews C, Farahbod R, et al. Dynamic resource allocation in computing clouds using distributed multiple criteria decision analysis. In: Proceedings of the IEEE 3rd International Conference on Cloud Computing (Cloud 2010); 2010; Miami, FL.
51. Ergu D, Kou G, Peng Y, Shi Y, Shi Y. The analytic hierarchy process: task scheduling and resource allocation in cloud computing environment. *J Supercomput*. 2013;64(3):835-848 .
52. Martens B, Teuteberg F. Decision-making in cloud computing environments: a cost and risk based approach. *Inf Syst Front*. 2012;14(4):871-893.
53. Dastjerdi AV, Garg SK, Rana OF, Buyya R. CloudPick: a framework for QoS-aware and ontology-based service deployment across clouds. *Softw: Pract Exper*. 2015;45(2):197-231. https://doi.org/10.1002/spe.2288
54. Menzel M, Schonherr M, Tai S. $(MC^2)^2$: criteria, requirements and a software prototype for cloud infrastructure decisions. *Softw: Pract Exper*. 2013;43(11):1283-1297. https://doi.org/10.1002/spe.1110
55. Carrasco J, Durán F, Pimentel E. Component-wise application migration in bidimensional cross-cloud environments. In: Proceedings of the 7th International Conference on Cloud Computing and Services Science - Volume 1: CLOSER 2017; 2017; Porto, Portugal.
56. Ristov S, Kostoska M, Gusev M. P-TOSCA portability demo case. In: Proceedings of the 2014 2014 IEEE 3rd International Conference on Cloud Networking (CloudNet); 2014; Luxembourg City, Luxembourg.
57. OASIS Standard. Topology and orchestration specification for cloud applications version 1.0. 2013. http://docs.oasis-open.org
58. Ali H, Moawad R, Hosni AAF. A cloud interoperability broker (CIB) for data migration in SaaS. *Future Comput Inform J*. 2017;1(1-2):27-34.
59. Duggan M, Duggan J, Howley E, Barrett E. A network aware approach for the scheduling of virtual machine migration during peak loads. *Cluster Computing*. 2017;20(3):2083-2094.
60. Tsakalozos K, Verroios V, Roussopoulos M, Delis A. Live VM migration under time-constraints in share-nothing IaaS-clouds. *IEEE Trans Parallel Distributed Syst*. 2017;28(8):2285-2298.
61. Zhao Y, Lo S, Zegura E, Ammar M, Riga N. Virtual network migration on the geni wide-area sdn-enabled infrastructure. In: Proceedings of the 2017 IEEE conference on computer communications workshops (INFOCOM WKSHPS); 2017; Atlanta, GA.
62. de Souza FR, Miers CC, Fiorese A, de Assunção MD, Koslovski GP. QVIA-SDN: towards qos-aware virtual infrastructure allocation on SDN-based clouds. *J Grid Comput*. 2019. https://doi.org/10.1007/s10723-019-09479-x
63. de Souza FR, Miers CC, Fiorese A, Koslovski GP. QoS-aware virtual infrastructures allocation on SDN-based clouds. In: Proceedings of the 2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID); 2017; Madrid, Spain.
64. de Assunção MD, da Silva Veith A, Buyya R. Distributed data stream processing and edge computing: a survey on resource elasticity and future directions. *J Netw Comput Appl*. 2018;103:1-17. https://doi.org/10.1016/j.jnca.2017.12.001
65. Mansour I, Sahandi R, Cooper K, Warman A. Interoperability in the heterogeneous cloud environment: a survey of recent user-centric approaches. In: Proceedings of the International Conference on Internet of Things and Cloud Computing (ICC '16); 2016; Cambridge, UK.
66. Hall C. Microsoft's cloud market share grew more than anyone else's last quarter – analysts. 2017. https://www.datacenterknowledge.com/business/microsofts-cloud-market-share-grew-more-anyone-elses-last-quarter-analysts
67. Synergy Research Group. Microsoft, Google and IBM public cloud surge is at expense of smaller providers. 2017. https://www.srgresearch.com/articles/microsoft-google-and-ibm-charge-public-cloud-expense-smaller-providers
68. Coles C. AWS vs Azure vs Google Cloud market share 2017. 2017. https://www.skyhighnetworks.com/cloud-security-blog/microsoft-azure-closes-iaas-adoption-gap-with-amazon-aws/
69. OpenStack. OpenStack. 2017. https://www.openstack.org/
70. Docker. Docker - build, ship, and run any app, anywhere. 2017. Docker:https://www.docker.com/
71. CloudLab. 2018. https://www.cloudlab.us/index.php

How to cite this article: Rodrigues LR, Cardoso E Jr, Alves OC Jr, et al. Cloud broker proposal based on multicriteria decision-making and virtual infrastructure migration. *Softw: Pract Exper*. 2019;1–21. https://doi.org/10.1002/spe.2723