ingenia

2

Cloud Brokering

New value-added services and pricing policies

Felipe Díaz-Sánchez

ediciones USTA

CLOUD BROKERING

New value-added services and pricing policies

ediciones USTA

CLOUD BROKERING

New value-added services and pricing policies

Felipe Díaz-Sánchez

ediciones USTA

Díaz-Sánchez, Felipe

Cloud brokering: New value-added services and pricing policies / Felipe Díaz-Sánchez - Bogotá: Universidad Santo Tomás, 2016.

vii, 115 páginas; ilustraciones, gráficas.

Incluye referencias bibliográficas (páginas 99-108)

ISBN 978-958-631-961-4

1. Almacenamiento virtual (computación) 2. Comercio electrónico

3. Computación en la nube I. Universidad Santo Tomás (Colombia).

CDD 004.67820688 CO-BoUST

© Felipe Díaz-Sánchez

© Universidad Santo Tomás

Directora editorial

Matilde Salazar Ospina

Coordinación de libros

Karen Grisales Velosa

Ediciones USTA

Carrera 9 No.51-11

Edificio Luis J. Torres, sótano 1

Bogotá, D. C., Colombia

Teléfono: (+571) 5878797 ext. 2991

editorial@usantotomas.edu.co

http://ediciones.usta.edu.co

Asistente editorial

Andrés Felipe Andrade

Diagramación

Julián E. Morales Ortega

Diseño de cubierta

Kilka Diseño Gráfico

Hecho el depósito que establece la ley

ISBN: 978-958-631-961-4

Primera edición: 2016

Todos los derechos reservados. Se prohíbe la reproducción total o parcial de esta obra, por cualquier medio, sin la autorización previa por escrito de los titulares.

To Leta and Days.

Contents

About this book

Service arbitrage enables advanced services in cloud brokering by taking advantage of two or more cloud provider offers. This allows cloud brokers to simplify the vast number of offers by categorizing the features and benefits of each cloud provider in order to match consumer needs with an ideal set of cloud providers. In the first part of this book entitled "Value-added services in cloud brokering", a comprehensive, state of the art study has been carried out on cloud performance evaluations and placement in cloud brokering (Chapter 1). Then, a method is proposed to calculate cloud performance through a single figure of merit based on the mapping of the physical features of a Virtual Machine (VM) to their respective performance capacities (Chapter 2). Finally, an exact placement approach for optimizing the distribution of cloud infrastructure across multiple providers is proposed (Chapter 3). Parameters such as price, VM configuration, VM performance, network latency and availability are considered for that purpose.

Nowadays, *pay-as-you-go* and *reserved* pricing dominate the way consumers acquire cloud resources from legacy cloud providers at the infrastructure level. However, the introduction of cloud brokers may induce the commoditization of cloud infrastructures. Facing such an evolution, new pricing models are necessary to capture potential consumers or untapped market segments. The second part of this book entitled "A new pricing model for cloud brokering" focuses on the design of a pricing model for cloud brokering, called *pay-as-you-book* (Chapter 4). *Pay-as-you-book* is based on two types of information. The first type consists of the forecast of users' job requests. The second one consists of the ability of cloud brokers to take advantage of such advanced reservations. With this aim in view, a study comparing three resource allocation policies under *pay-as-you-book* is carried out. The aim of this book is to contribute to the design of new value-added services and pricing models for cloud brokering. The majority of the investigations and original results presented in this manuscript have been achieved and obtained in the context of the CompatibleOne [1] research project supported by the French Ministry of Industry. Its objective was to demonstrate the feasibility of a cloud brokering intermediation platform integrating and adapting the various software solutions proposed by the industrial and academic partners of the project. This platform provides a single point for service consumption in order to avoid vendor lock-in. This book has three objectives:

- The first one is to present a single figure of merit for cloud VMs performance based on the application profile.
- The second one is to propose an exact approach for allocation of VMs across multiple cloud providers based on different optimization criteria.
- The third one is to describe a pricing model for cloud brokering, called *pay-as-you-book*.

The contribution of this book can be itemized as the following:

- A method to calculate a figure of merit for VM cloud performance. The originality of this figure of merit is to offer a single value to express VM cloud performance that is based on the type of application to be deployed. Thus, end-users may in a straightforward manner compare and select the best cloud provider in which to deploy an application.
- The formulation of Mixed-Integer Linear Programming for placement of VMs across multiple cloud providers. The originality of this approach is in associating the heterogeneity of cloud providers' offers with their respective performance. This approach may be applied to the optimization of cost, performance, cost-performance and disaster recovery scenarios.
- The description of pay-as-you-book, a pricing model between pay-as-you-go and subscription. Pay-as-you-book consists of paying and reserving time-slots of VMs in advance without a fixed fee

to subscribe to the service and without a long-term commitment, avoiding vendor lock-in, while obtaining lower prices than in pay-as-you-go. Pay-as-you-book may be applied in scenarios with predictable workloads. Through simulations, it has been shown why a model such as pay-as-you-book is not convenient for cloud providers. However, cloud brokers reselling cloud infrastructure may create attractive service offerings based on pay-as-you-book.

This book consists of an overview of the following conference publications:

1. F. Díaz-Sánchez, S. Al Zahr, M. Gagnaire, J-P. Laisné, J. Marshall. "CompatibleOne: Bringing cloud as a Commodity". *IEEE International Conference on cloud Engineering (IC2E)*, Boston, US, May. 2014.

2. F. Díaz-Sánchez, S. Al Zahr, and M. Gagnaire. "An Exact Placement Approach for Optimizing Cost and Recovery Time under Faulty Multi-cloud Environments". *IEEE cloudCom Conference*, Bristol, UK, Dec. 2013.

3. F. Díaz Sánchez, E. Doumith, S. Al Zahr and M. Gagnaire. "An Economic Agent Maximizing cloud Providers Revenues Under Pay-as-you-Book Pricing Model". *Conference on the Economics of Grids, clouds, Systems, and Services (GECON)*, Berlin, Germany, Nov. 2012.

4. F. Díaz, E. Doumith and M. Gagnaire. "Impact of Resource over-Reservation (ROR) and Dropping Policies on cloud Resource Allocation". *IEEE cloudCom Conference*, Athens, Greece, Nov. 2011.

5. F. Díaz-Sánchez, M. Gagnaire, J. Marshall and J-P. Laisné. "COSCHED: A Scheduling Agent Maximizing cloud Broker's Revenues under the CompatibleOne Architecture". *The 11th IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA-13)*, Melbourne, Australia, Jul. 2013.

Cloud brokering

The role of cloud Brokers in the near future of cloud computing has been identified by Gartner as a major market trend: "*By 2015, cloud Brokers will represent the single largest category of growth in cloud computing, moving from a sub-$1 billion market in 2010 to a composite market counted in the hundreds of billions of dollars.*" [2]. This prediction seems to be reinforced by the amount of funding raised by some cloud brokering companies: *Rightscale* US$47.3m in three rounds[1], *6fusion* US$10m in two rounds, *cloud Cruiser* US$7.6m in two rounds, *Zimory Systems* US$7.2m in two rounds and *Gravitant* US$3.7m in one round [3]. One of the main reasons, behind this high economic expectation, is the highly heterogeneous current cloud market constituted by many cloud providers. Each cloud provider exhibits different interfaces, pricing models and value-added services. To help the end-user cope with such a fragmented ecosystem, cloud brokers have emerged as an intermediary third-party that provides unified self-service access to multiple cloud providers. Thus, by being a single point for service consumption, cloud brokers provide interoperability and portability of applications across multiple cloud providers. Besides this inherent role, current cloud brokers provide other value-added services to cloud consumers, such as the following: Advanced management by using tools beyond the stacks offered by cloud providers (*e.g.* consolidated billing, infrastructure monitoring, disaster recovery, SLA enforcement), elasticity management in order to automatically scale up or down infrastructure resources based on the workload and service arbitrage with the aim of taking

[1] A funding round is a practice by which a company raises money to fund operations, expansion, an acquisition, or some other business purpose.

advantage of two or more cloud provider offerings (*e.g.* cost optimization). These services can be overlayed, enabling new cloud computing scenarios such as cloud bursting or cloud marketplaces (Figure 1). These new scenarios may be beneficial for both end-users and cloud providers. In the case of cloud bursting, end-users have the possibility of extending their computing facilities by moving the development of applications or the non-mission-critical applications to public clouds. In the case of a cloud marketplace scenario, end-users have access to multiple cloud providers through a single interface, while cloud providers may sell spare infrastructure capacity. Cloud brokers are expected to drive creation of

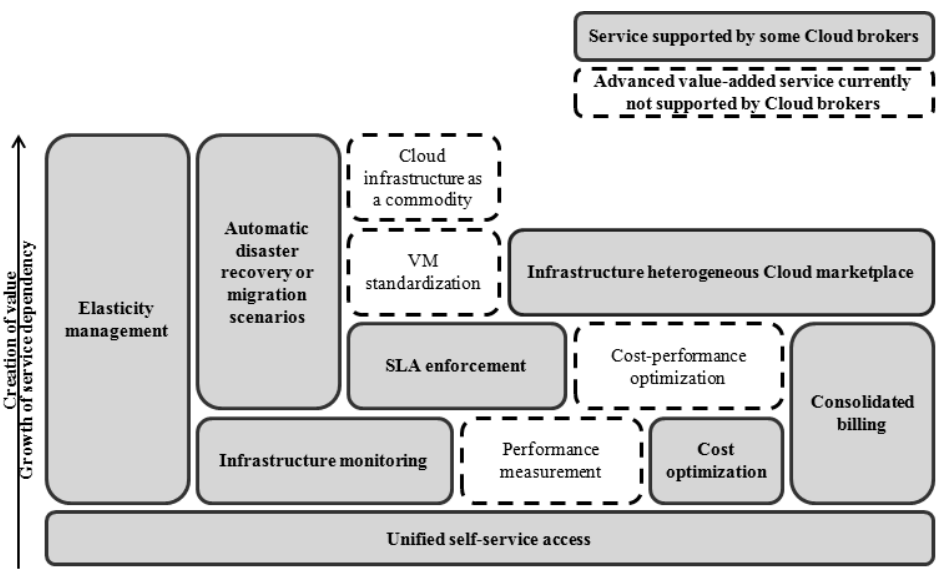

Figure 1: Evolution and dependency of value-added services in cloud brokering

value through advanced value-added services enabling new cloud computing scenarios. The price of cloud computing resources varies around 20% between cloud providers, while the difference in performance between cloud providers remains unknown or less studied [3]. Due to the fact that cloud brokers are able to deploy a workload in any cloud provider, the measurement of performance for cloud providers and the placement of cloud resources based on a cost-performance relationship may be supported by cloud brokers in future value-added services. Moreover, the commoditization of infrastructure resources will increase the cloud adoption rate by simplifying the purchase of cloud computing resources. As cloud computing resources are traded like any other commodity (*e.g.* wheat, oil, iron) the currently fragmented cloud market will flatten. This will open the door to new pricing models in which cloud brokers not only act as intermediaries but also as liquidity providers, negotiating volume discounts from cloud providers and guaranteeing resource availability to end-users.

Part I

Value-added services in cloud brokering

Chapter 1

Cloud performance and placement in cloud brokering

Introduction

As the number of cloud computing services increases, so does the interest of consumers to be able to compare these services in order to choose those best adapted to their needs. This chapter focuses on the performance issues related to cloud provider evaluation and on the role of cloud brokers in the automatic optimization of resource allocation across multiple cloud providers. This chapter is structured as follows. Section "cloud performance evaluation". presents a survey of the current studies related to cloud performance evaluation. The motivation for and challenges behind the evaluation of cloud provider performance is also detailed in this section. Section "Placement in cloud brokering" describes the current methods used to allocate resources in cloud brokering. The studies are classified into two categories: placement based on non-functional requirements and application-aware placement.

Cloud performance evaluation

Motivations and challenges

The current cloud computing landscape hinders a straightforward comparison of cloud provider service offerings. In the case of computing resources, this is mainly due to the heterogeneity of VM configurations and prices. On one hand, traditional cloud providers such as Amazon, Rackspace and WindowsAzure sell fixed-size VMs. These VM configurations often vary from one cloud provider to another, making it impossible to make a direct comparison. On the other hand, new cloud providers in an effort to attract consumers, look to differentiate their services through technology by allowing consumers to freely configure the size of the computing resources to be purchased.

VM performance evaluation adds another layer of complexity to the comparison of cloud providers. Firstly, consumers have little knowledge and control over the infrastructure hosting their applications. Due to the virtualization of hardware used in cloud computing, cloud providers may use resource sharing practices (*e.g.* processor sharing, memory overcommit, throttling or under-provisioned network [4]) that degrade the performance of a cloud application. Secondly, cloud provider's data centers are equipped with hundreds of thousands of servers with different qualities of hardware and software. The evaluation of performance cross all the data centers of multiple cloud providers, implies a trade-off between thoroughness, time and cost of the evaluation [5]. Thirdly, cloud providers may continually upgrade or extend their hardware and software infrastructures, and new commercial services and technologies may gradually enter the market [6]. Therefore, performance evaluations become quickly out of date and the tools for performance measurement must be continuously re-designed. Finally, there are no cloud-specific benchmarks to evaluate all VM features [7]. However, traditional benchmarks can partially satisfy the requirements for cloud performance evaluation.

Cloud performance evaluation would be beneficial for both consumers and cloud providers [5]. Consumers testing their applications across multiple cloud providers can choose the cloud provider that represents the best performance-cost trade-off. Also, performance evaluations can serve as a recommendation for the performance of a particular system [4] or can give technical arguments to consumers to put pressure on cloud providers to use better practices [7]. A provider may identify its market positioning in order to improve its services or to adjust its prices [5].

Studies related to cloud providers performance evaluation

An exhaustive study about the academic approach to commercial cloud services evaluation has been carried out by the Australian National University [6]. A Systematic Literature Review (SLR) was the methodology employed to collect the relevant data to investigate the evaluation of cloud services. As a result, 82 relevant cloud service evaluation studies were identified. The key findings of this study represent a state-of-practice when evaluating cloud services and are as follows:

- 50% of the relevant studies investigated applying cloud computing to scientific issues, while only 16% of the studies focused on the evaluation of business applications in the cloud.
- 21 cloud services over 9 cloud providers were identified. 70% of the relevant studies evaluated cloud services provided by Amazon Web Services (AWS).
- Three main aspects and their properties for cloud services evaluation have been investigated (performance, economics and security), performance being the most studied aspect (78 studies).
- There is no consensus regarding the definition and the usage context of metrics. Some metrics with the same name were used for different purposes, some metrics with different names were essentially the same. The study identified more than 500 metrics including duplications.
- There is a lack of effective metrics vis-à-vis elasticity and security aspects in cloud computing. Therefore, it is hard to quantify these aspects.
- There is not a single or a small set of benchmarks that provides a holistic evaluation of cloud services. The SLR identified around 90 different benchmarks in the selected studies of cloud services evaluation. These benchmarks can be grouped in three

main categories: application, synthetic and micro-benchmarks, as explained below.

- 25 basic setup scenarios for constructing complete cloud service evaluation experiments have been identified and classified.
- The cloud service evaluation is getting more and more attention from the research community. The number of relevant studies was 17 times larger in 2011 (34 studies) than in 2007 (2 studies).

Cloud performance evaluation is done by running application benchmarks, synthetic benchmarks or micro-benchmarks in single or multiple cloud providers. Application benchmarks correspond to real-world software that provides an overall view of the performance of a specific application. Synthetic benchmarks simulate application behavior by imposing a workload on the system. Similarly, micro-benchmarks impose a workload with the aim of measuring hardware-specific VM features. Since there are no cloud-specific benchmarks, cloud performance has been measured through widely used benchmarks such as TPC-W (a transactional web e-Commerce benchmark) [8], HPCC (a software suite consisting of 7 basic benchmarks) [4, 9, 10], NPB (set of parallel benchmarks to evaluate the performance of parallel supercomputers) [4, 11] or common measurement tools such as *ping* or *iperf* [12,13]. Also, specific benchmarks have been developed to measure cloud performance of CPU, memory, disk and network [14, 15] further the VM provisioning or deprovisioning time [10, 12, 16]. Details about the studies related to cloud providers' performance evaluation are presented in Table Appendix A.

Recent studies tend to clarify confusing concepts, inaccurate terms, as well as to unify the metrics used by previous cloud performance evaluation studies. Li *et al.* propose a taxonomy of performance for evaluating commercial cloud services [8] and potential approaches to bring about a holistic impression of cloud services performance through a single figure of merit [17].

Cloud Virtual Machine (VM) characterization

According to the studies of cloud provider performance evaluation presented in the previous section, a cloud VM can be represented by a set of criteria and a set of capacities (Figure 1.1). The criteria set is composed of the

VM physical properties (*i.e.* communication, computation, memory and storage) and of cloud service related features (*i.e.* availability, reliability, scalability and variability). The set of capacities corresponds to the metrics used to describe the performance of the criteria. Both criteria and capacities are described below.

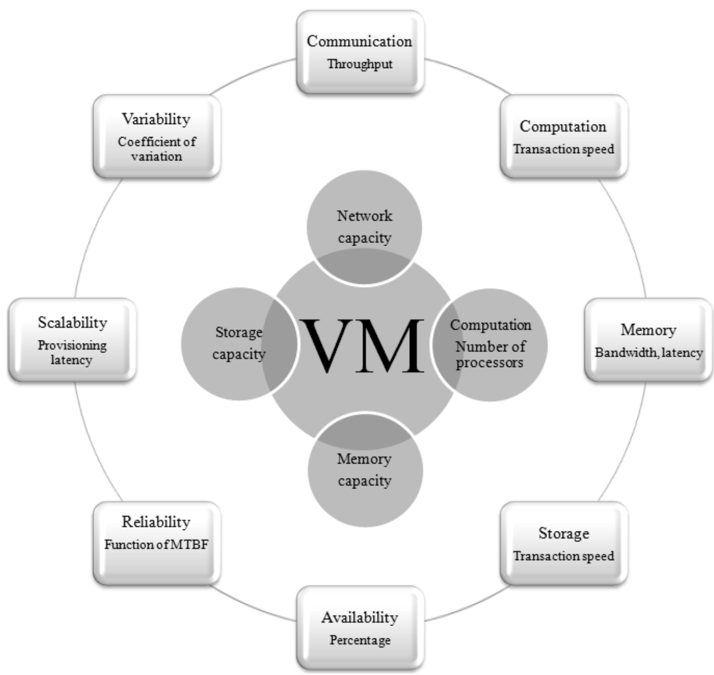

Figure 1.1: VM characterization. The inner circle represents the physical properties of a VM. The outer circle presents the performance criteria and examples of their capacities.

Criteria

- *Communication* is defined as the property of transferring data between two entities through a network. Three types of communications can be distinguished: intra- and inter-datacenter network, and wide-area network [5]. Intra-datacenter network refers to the communication of two VMs belonging to the same datacenter, while inter-datacenter network corresponds to the communication between two VMs located in different datacenters but belonging to the same cloud provider. Wide-area network refers to the communication between a VM located in a datacenter and an external host on the Internet.

- *Computation* refers to the physical property of processing data. In the case of a VM, computation corresponds to the evaluation of the virtual CPU.
- *Memory* corresponds to the physical property of storing data on a temporary basis. Both RAM memory and cache are considered in this category.
- *Storage* refers to the physical property of storing data on a permanent basis, until the data is removed or the service is suspended by the end-user.
- *Availability* is defined as the percentage of time an end-user can access the cloud service (Equation 1.1). For a given interval of time, it is calculated as a ratio of the uptime of the cloud service to the total time of the interval, usually on a yearly basis.

$$\text{Availability} = \frac{\text{total uptime}}{\text{total time of the interval}} \tag{1.1}$$

- *Reliability:* In the literature, the definition of reliability varies depending on different contexts or perspectives. Here, reliability refers to the ability of a cloud service to perform its function for a specified period of time (Equation 1.2). It is defined based on the previous failures experienced by users and the promised Mean Time Between Failures (MTBF) by the cloud provider [18].

$$\text{Reliability} = \left(1 - \frac{\text{number of users experiencing a failure}}{\text{number of users}}\right) \times \text{MTBF} \tag{1.2}$$

Thus, if a cloud provider promises a MTBF of 8760 hours (one failure per year) and 20% of his clients experienced a failure in an interval less than promised by the cloud provider. The reliability performed by the cloud provider, according to the definition presented in this study is 7008 hours (or 9 months and 22 days).

- *Scalability (also known as elasticity)* is the extent to which the application's capacity can be adapted to the demand of end-user [19]. Two types of scalability can be distinguished: *Horizontal* [20, 21] and *Vertical* [22, 23]. The former refers to the provisioning of multiple instances of the cloud service (*e.g.* deploy new VMs). The latter implies adding more resources to a current cloud service (*e.g.* adding more processors or storage to a VM).

- *Variability (also known as stability)* refers to the variation of performance of a cloud service. Unlike the availability and the reliability that are either provided by the cloud provider or that can be easily calculated, variability depends on the values of the capacities (as explained below). Therefore, variability can be considered as a derived capacity. Several metrics have been employed to evaluate variability [24]. Here, the Coefficient of Variation (CV) has been used, which is defined as the ratio of the standard deviation to the mean (Equation 1.3). The CV is useful for comparison between data sets with different units (as it is the case for most of the benchmarks), since it allows for the comparison of the degree of variation from one data set to another.

$$\mathrm{CV} = \frac{1}{\bar{x}} \cdot \sqrt{\frac{1}{N-1} \cdot \sum_{i=1}^{N} (x_i - \bar{x})^2} \tag{1.3}$$

Here, N is the number of measurements; $x_1, ..., x_N$ are the measured results; and $\bar{x}$ is the mean of those measurements.

Capacities

The presented capacities have been defined by Li *et al.* in [8].

- *Transaction speed* defines how fast transactions (*e.g* job execution, read/write operations) can be processed.
- *Data throughput (Bandwidth)* is considered as the amount of data processed by any physical property in a given period of time.
- *Latency* includes all the time-related capacities of a cloud service.
- *Other* consists of dimensionless metrics (*i.e.* availability, CV) or single metrics such as the reliability.

Placement in cloud brokering

The placement or resource allocation in cloud brokering refers to the mechanisms used to distribute infrastructure resources across multiple cloud providers based on end-user' needs and constraints. The optimization goal in placement is to select a single or a set of cloud providers to optimally deploy a service based on optimization criteria, for example, cost optimization or performance optimization. Placement mechanisms can be classified into *non-functional requirements-based placement* and *application-aware placement.* The non-functional requirements-based placement corresponds to the allocation of cloud infrastructure based on matching both cloud provider resources and end-user requirements. The application-aware placement is based on the constraints that guarantee a Quality of Service (QoS) of the application running on top of the infrastructure.

Non-functional requirements-based placement

Placement studies based on non-functional requirements consider performance of cloud providers and/or dynamic pricing scenarios[1]. In the literature, two cloud brokering placement scenarios have been identified: static and dynamic. Static placement assumes that changes within the

[1] Another non-functional requirement, out of the scope of this book, covers end-users limiting the set of placement solutions due to political and legislative considerations. For example, end-users could avoid placing data either outside or inside a given region (*e.g.* the EU Data Protection Directive which regulates the processing and free movement of personal data within the European Union).

cloud environment never happen. Dynamic placement addresses the issue of how to reconfigure cloud resources optimally, adapting them to new situations when conditions change (*e.g.* cloud provider outage, new VM prices, and *etc.*). The approaches described below are based on exact models (*e.g.* binary integer programming formulation).

Static placement

Tordsson *et al.* [25] propose an architecture for cloud brokering and a placement algorithm based on the performance of the GridNPB/ED benchmark and the price of resources. An end-user may constrain resource deployment by specifying the type and number of VMs to be deployed and the percentage of VMs located within each cloud provider. Chaisiri *et al.* [26] propose an optimal VM placement across multiple cloud providers that considers both reserved and on-demand provisioning plans. However, including the reservation plan implies not only a long-term commitment in exchange for lower prices regarding on-demand service provisioning but also raises new issues in case of underprovisioning or overprovisioning of IaaS resources. On one hand, in the underprovisioning scenario, the demand can be fully met through on-demand resources at a higher cost. On the other hand, in the overprovisioning scenario, questions arise such as: Who (the end-user or the cloud broker) is going to pay for the unutilized IaaS resources?

The usefulness of cloud brokering placement for fully-decoupled or loosely-coupled applications is studied by Van den Bossche *et al.* [27] and Moreno-Vozmediano *et al.* [11]. Both approaches improve the cost-effectiveness of the deployment and consider an on-demand provisioning plan and a hybrid IaaS cloud architecture. Van den Bossche *et al.* [27] propose a cost-optimal placement for preemptible but non-provider-migratable batch workloads with a strict completion deadline. The workloads are characterized by memory, CPU and data transmission requirements. The problem is tackled by Linear programming. Moreno-Vozmediano *et al.* [11] evaluate the scenario of deploying a computing cluster on top of a multi-cloud infrastructure for solving loosely-coupled Many-Task Computing (MTC) applications. The goal is to improve the cost-effectiveness of the deployment, or to implement high-availability strategies. This approach is evaluated through a low scale testbed including a local data-center and three different public cloud sites.

This testbed is complemented with simulations that include a larger number of resources.

Dynamic placement

Lucas-Simarro *et al.* [28] propose a VM placement algorithm with the goal of minimizing the costs for end-users in a dynamic pricing environment. The cloud broker transfers a client's infrastructure from one cloud provider to another based on price fluctuations. The algorithm calculates possible future prices based on the average cloud provider's price and its price trend. In order to guarantee the performance of the applications running on top of the IaaS resources, the placement decisions are constrained by: The maximum and minimum number of VMs to reallocate in each placement and a load balancing requirement that indicates the percentage of resources to maintain within each cloud provider. In this approach, the placement problem is limited to one VM configuration. Lucas-Simarro [29] extends this work to multiple VM configurations and addresses the problem of performance optimization. Performance optimization consists in maximizing the performance of the deployed resources by choosing the VMs with the best performance in terms of hardware resources (hard disk, memory, CPU). A drawback of this approach is that VM performance measurements can only be provided by end-users after testing all VM configurations within each cloud provider.

A more complex model that not only involves cost-optimization but also copes with changes in the cloud environment through VM migration is proposed by Tordsson *et al.* [30]. In this model, the time for VM migration is estimated as the time required to shut down a VM within one cloud provider and start a new VM with the same configurations within another.

Chaisiri *et al.* [31] propose an optimal cloud resource provisioning algorithm minimizing the cost of resource provisioning for a certain period given the uncertainty for demand and price. The optimal decision calculated by the cloud broker is based on the end-users' demands and cloud providers' prices. This allows the cloud broker to adjust the number of resources acquired in advance under reservation and the number of resources to be acquired under on-demand provisioning, taking into account that reserved VMs are generally cheaper than on-demand ones. This approach tackles the underprovisioning and overprovisioning problem.

Chaisiri addresses this problem through stochastic integer programming.

Application aware placement

The application-aware placement dynamically scales resources up or down across multiple cloud providers' infrastructures under QoS constraints specific to the application. In the case of tightly-coupled applications with low delay or strong communication requirements, the placement process should guarantee a single-cloud deployment [32]. However, in the case of fully-decoupled[2] or loosely-coupled applications, the placement process may take advantage of the heterogeneity of cloud providers' offers to deliver a cost-effective solution that guarantees the performance of the application [27, 33]. In the case of interactive applications (*e.g.* on-line gaming), user experience relies on network bandwidth and on the latency caused by geographical distances [32]. Therefore, these kinds of applications should be treated near the geographical location of their origin to achieve lower latency and higher throughput.

The importance of cloud brokering for telecommunication services is highlighted by Carella G. *et al.* [34]. In this approach, the cloud broker enhances his placement mechanisms based on: real-time data on network performance, QoS requirements and cloud providers' prices. The goal is to provide to telecommunication service operators a minimum QoS to satisfy customer requirements by monitoring the deployed services. This approach is evaluated in a testbed composed of a cloud broker and an IP Multimedia Subsystem (IMS) deployment. The cost-effective placement of Web 2.0 applications with high-availability and fault-tolerance requirements across multiple cloud providers is proposed by Frincu *et al.* [35]. In this approach, authors consider applications consisting of several components and connectors (C/Cs). C/Cs are reallocated by making a snapshot, stopping the execution of each C/C, moving the snapshot to a new VM and starting the C/C from the snapshot. A cloud broker architecture with the intelligence to react to changes in business processes by changing the cloud configuration across multiple cloud providers is described by Grivas *et al.* [36].

The placement of services with different QoS and service provisioning

[2] Applications are fully-decoupled when the jobs that form the application have no precedence constraints, and can be executed in parallel.

requirements for risk assessment services and e-learning education applications is tackled by Quarati *et al.* [37]. The goal is to maximize user satisfaction and broker revenues by reducing energy costs, through energy saving mechanisms. For this, the cloud broker allocates IaaS resources to the public or private cloud, based on end-user's QoS expectations and the workload of the private resources. This approach was evaluated through a discrete event simulator.

Conclusion

In this chapter, research work tackling the problem of cloud performance evaluation and placement in cloud brokering has been surveyed. A shortcoming in the current approaches to cloud performance evaluation is the absence of a single figure of merit that provides a straightforward comparison of cloud providers. Regarding the problem of placement in cloud brokering, the surveyed studies assume that cloud providers offer the same type of VM configurations. This assumption is not true for all the cases; VM configurations may vary from one cloud provider to another. For some cases, even a VM offered by a cloud provider in one location, may not exist in another location belonging to the same cloud provider. These issues are tackled in the next two chapters.

Chapter 2

Towards a figure of merit for cloud performance

Introduction

Let's imagine creating a figure of merit for automobiles and that the most expensive Mercedes-Benz has the highest figure of merit. Does this mean that everyone should buy that particular car? What if you want to tow a trailer? In cloud computing, although we may be able to find a single value to represent cloud performance, it does not mean that this value will be useful for all types of applications. According to the physical property that limits performance, applications can be classified as CPU-bound, memory bound or I/O bound (Figure 2.1). Therefore, the application profile must be taken into account in the calculation of a single figure of merit.

Currently, the information given by cloud providers allows for a simple but inaccurate comparison between providers. End-users can choose a cloud provider by comparing quantitatively different VM offers (*e.g.* number of cores per dollar, memory or storage capacity per dollar). Using this simple approach, end-users can select the cloud provider that offers the largest quantity of resources at the lowest price. However, this only makes sense in the unreal scenario in which cloud providers have qualitatively homogeneous resources.

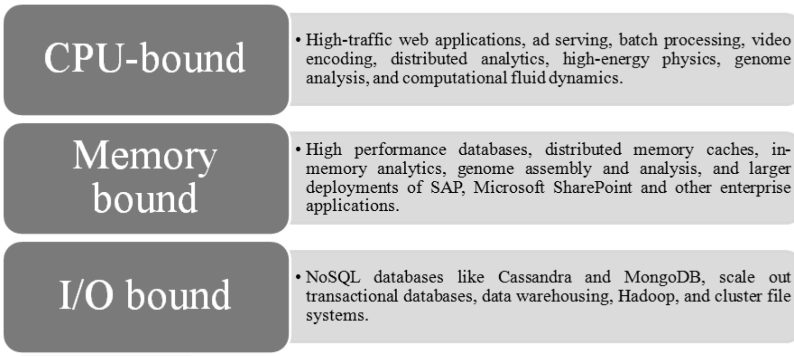

Figure 2.1: Examples of types of applications[a]

[a]Source: http://aws.amazon.com/ec2/instance-types/

Another more precise alternative for comparing cloud providers, is to evaluate the performance of an application across multiple cloud providers and to choose the set of cloud providers with the best performance-cost ratio [29]. This approach is technically feasible through a cloud broker but requires a time-consuming and highly expensive task: The application must be evaluated in each VM configuration offered by each cloud provider. Moreover, the performance estimation becomes increasingly inaccurate as the cloud providers upgrade their infrastructures (Section "Motivations and challenges"). In contrast to the methods described above, a figure of merit for cloud performance based on the application profile, can serve as a general guide for the performance of a particular system configuration and provide a straightforward comparison of cloud providers for a given type of application. In this chapter, a figure of merit for cloud performance is calculated, by running benchmarks in advance across multiple cloud providers and by obtaining a composed metric based on the benchmarks' results.

The motivation behind a figure of merit in cloud brokering is twofold. Firstly, the performance of the cloud providers can be measured in advance with consistent results for a period of time. This is due to the fact that the time to run benchmarks is much shorter than the upgrade cycle of cloud infrastructures. Secondly, a cloud broker, granting access to multiple cloud infrastructures as a trusted third-party, may provide up-to-date evaluations of cloud provider performance. Cloud brokers may automatically deploy benchmarks and process the results. Thereby, a cloud broker can easily automate the process for calculating a figure of merit. This chapter is organized as follows. Section "Performance evaluation" describes, the methodology and the experimental setup used in this cloud evaluation. This is followed by the evaluation of the provisioning time and the evaluation of criteria such as computation, memory, storage and variability for different types of VMs and cloud providers. In Section "Figure of merit of VM cloud performance", two approaches to calculate a single figure of merit for cloud performance are presented. Finally, Section "Case study: CPU-Intensive application" presents a case study for a CPU-intensive application. In this case study, three different methods have been used to calculate a figure of merit. Real performance result have been used in the study.

Performance evaluation

Evaluation methodology

There is a lack of standardized methodology for cloud performance evaluation through benchmarks (*cf.* Section "Studies related to cloud providers performance evaluation"). The methodology employed in this work to measure cloud performance. Is composed of five main steps (Figure 2.2):

1. *Define scenarios:* the stakeholders (*i.e.* cloud providers to be evaluated) are identified, as well as the features related to the cloud services such as VM configurations and datacenter locations.

2. *Identify Benchmarks:* selection of suitable benchmarks according to the scenario formulated in the previous step. If the evaluation of a specific application is going to be performed, this step is omitted. Evaluation-related issues such as the number of benchmark repetitions and the type of workload are defined in this step [8].

3. *Run tests:* the resources are acquired on the selected cloud providers' locations. Then, benchmarks are deployed into the chosen VM configurations. At the end of this step, the results are collected and the resources are released.

4. *Process results:* the results are treated and synthesized. For example, by calculating a figure of merit for cloud performance or by generating a graphical representation that summarizes the main results, aspects and trends.

5. *Analyze results:* In this final step, comments or recommendations are formulated based on the results.

Figure 2.2: Evaluation methodology

Experimental setup

The initial idea was to create an Operative System (OS) image consisting of all the scripts and benchmarks necessary to measure cloud performance. This image would be uploaded and deployed in every cloud. Thus, the same workload conditions for every cloud provider is guaranteed. However, the cloud providers, covered in this study, present issues that hinder or completely prevent VM import. In some cases, cloud providers only support the import of VMs generated via licensed software (*e.g.* Amazon only support the import of images generated with VMware vSphere Client). In other cases, the import of VM images is only supported for some of the image formats (*e.g.* Cloudsigma only supports import of VMs in *RAW* format). Finally, particularly in recently emerged cloud providers, the import of VM images is not supported at all. For these reasons, the choice was made to build a VM image from the images already offered by cloud providers. The experimental setup presented here consists of three phases: image setup, running benchmarks and processing benchmark results. These phases occur once the accounts have been created in every cloud provider to be evaluated and the payment details have been registered. More issues related with this evaluation of performance are presented in Appendix B.

The image setup is as follows. First, a VM via web interface or command line is created. During the VM creation, the OS system is chosen. In this setup, Linux CentOS 6.X for a 64 bits processor architecture, an OS supported by the majority of cloud providers, has been chosen. Once the VM has been created, the OS is updated and a *ssh* server is installed to enable a secure remote control of the VM. Then, the scripts, benchmarks and tools necessary to evaluate cloud performance are installed and configured. The scripts have been developed in Python. The execution permissions of the */etc/rc.local* file have been modified and changed, in order to automatically trigger the benchmarks once the VM is turned-on. The *phoronix-test-suite* [38] has been selected as the framework to deploy benchmarks due to its wide set of supported benchmarks (more than 350). For the transmission of benchmark results, *s3cmd*, a command line tool for using the Amazon S3 service, has been used. The image containing all the scripts, benchmarks and tools necessary to evaluate cloud performance has been called *ceilo* (Figure 2.3). As explained above, the benchmarks are triggered automatically and sequentially once the VM is turned-on. Once all the benchmarks have been executed, the results are sent to an Amazon S3 bucket and the VMs are automatically turned-off. Benchmark results correspond to XML files. Thus, result files are parsed with *xsltproc* [39], a command line tool for applying XSLT stylesheets to XML documents; and values such as the average, the variance, the standard deviation and the coefficient of variation are calculated. Finally, for some of the results, a script to automatically generate graphical representations of the results with *google charts* is run.

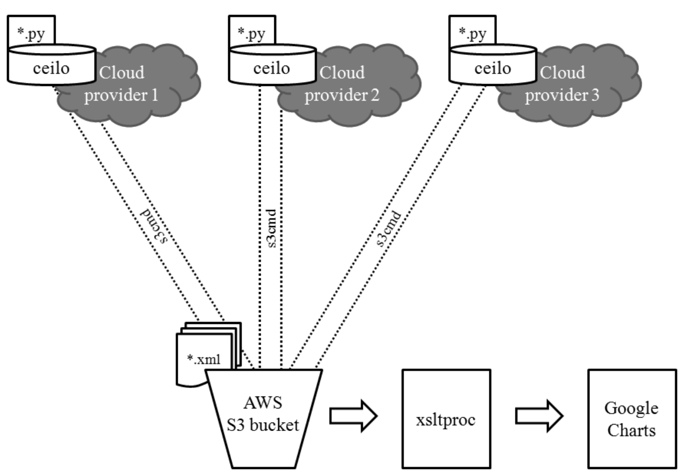

Figure 2.3: Experimental setup

In this study, the performance of 37 different VM types across 8 cloud providers with data centers in Europe has been measured (Table 2.1). According to the pre-configured VMs offered by some cloud providers and in order to compare VMs with similar capacities, five different sizes have been defined: VM sizes (*xs*, *s*, *l*, *m* and *xl*) (Table B.1). This classification has been based on the number of virtual CPUs (vCPUs) and the RAM memory size.

Id	Cloud provider
ARU	ArubaCloud
AWS	Amazon Web Services
CLO	Cloudsigma
JOY	Joyent
LUN	Lunacloud
PRO	Profitbricks
RAC	Rackspace
WIN	WindowsAzure

Table 2.1: Evaluated cloud providers

Criteria	**Capacity**	**Benchmark**	**Metric**	**Type**
Computation	Transaction speed	7-zip [40]	MIPS	HB
		C-Ray [41]	seconds	LB
Memory	Data throughput	Stream [42]	MB/s	HB
		CacheBench [43]	MB/s	HB
Storage	Transaction speed	Threaded I/O Tester [44]	MB/s	HB
		Iozone [45]	MB/s	HB

Table 2.2: Benchmarks

The performance evaluation presented here is based on 6 benchmarks. These benchmarks measure the computation, memory and storage capacities (Table 2.2). Depending on the operation implemented by the benchmark, the magnitude of the results can be classified as: Lower is Better (LB) or Higher is Better (HB). LB means that the lower the value, the better the system to execute a given benchmark. Inversely, HB means that the higher the value, the better the system to execute a given benchmark.The provisioning time has been measured with the scripts developed for this research. In this section, the mean values and the standard deviation of the obtained results have been plotted. An analysis related to the benchmark duration is presented in Appendix B.

Provisioning time

The provisioning time (or scaling latency) is defined as the time taken by a cloud provider to allocate a new VM once the end-user requests it [5]. The provisioning time corresponds to the amount of time it takes for a cloud provider to power-on a VM (VM provisioning time) and the boot time of the OS, defined as the time between when the VM has been powered-on and the VM is ready to be used. The provisioning time has a direct impact in the scalability of a cloud application, particularly in peak load scenarios, where the deployment of cloud infrastructure must follow the workload variations and the VMs must be ready to be used as soon as possible.

The VM provisioning time for every VM size of WindowsAzure and Amazon has been measured (Figure 2.4). In general, WindowsAzure has a higher provisioning time and a larger standard deviation than Amazon. The VM provisioning time for Amazon stays under 25s while for WindowsAzure it is consistently over 25s.

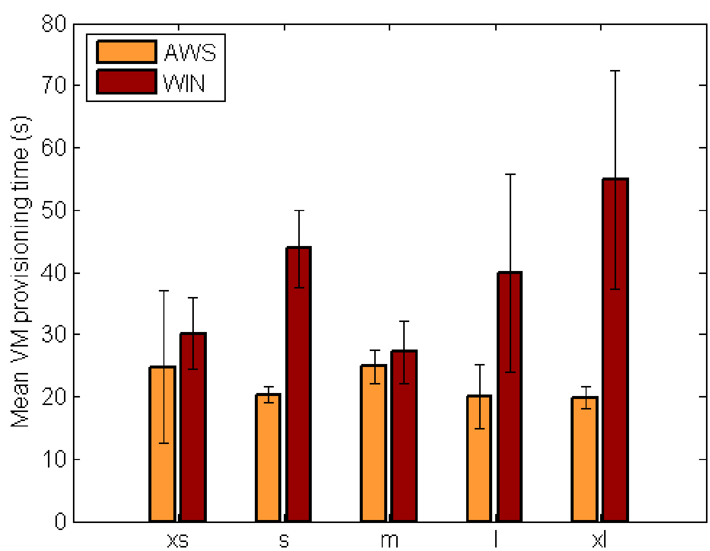

Figure 2.4: Average VM provisioning time for Windowsazure and Amazon

The boot time for every VM size of every cloud provider has also been measured (Figure 2.5). In 5 out of 8 evaluated cloud providers, the VM boot time varies from 10 to 25 seconds and is independent of the VM size. In the case of Joyent, the boot time is inversely proportional to the VM size. Arubacloud presents a VM boot time that varies from 40 to 50 seconds*s* and it is independent of the VM size. Lunacloud presents the lowest boot time values for *xs*-, *s*-, *m*- and *xl*-VM sizes. Lunacloud's *l*-size VMs present the highest boot time among all the evaluated cloud providers. There is not a logical explanation to this fact, from the collected data. It has been inferred: Lunacloud's *l*-size VMs shared the same processors family (Intel Xeon E5-2620) with the other Lunacloud's VM sizes. Thus, the processor brand is unlikely to be the reason for these boot time differences. Unfortunately, there are no additional results or information about the Lunacloud's underlying infrastructure to determine the reasons behind this high boot time.

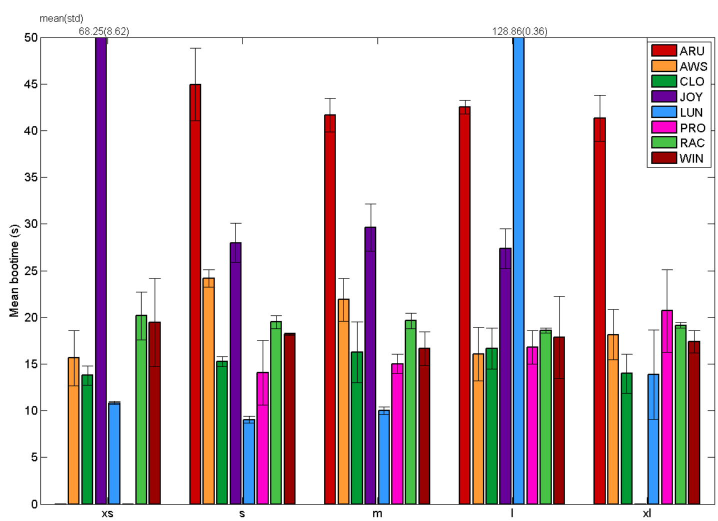

Figure 2.5: Average boot time

Computation benchmarks performance

The transaction speed has been measured with *7-Zip* and *C-Ray* benchmarks (Figure 2.6). 7-Zip is an application to compress files. The 7-Zip benchmark consists of compressing a file with random data and measuring the number of CPU instructions executed during the compression. C-Ray measures floating point CPU performance. By default, the benchmark uses only a small amount of data, such that on most systems the CPU does not have to access the RAM to run the benchmark. In this performance evaluation, C-Ray was set up to measure the time to render an image with a resolution of 800x600 pixels. Therefore unlike for 7-Zip in C-Ray, lower results are better.

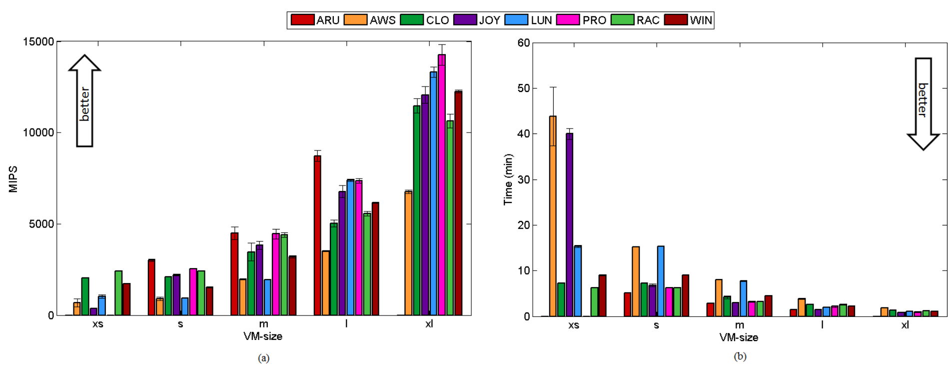

Figure 2.6: Performance of computation benchmarks (a) 7-Zip results (HB) (b) C-Ray results (LB)

Memory benchmarks performance

RAM and cache memory bandwidth have been measured with the *Stream* and *Cachebench* benchmarks, respectively (Figure 2.7). Stream is a simple synthetic benchmark program that measures sustainable memory bandwidth and the corresponding computation rate for simple vector kernels. In this performance evaluation, Stream was set up to measure the memory bandwidth through the *copy* and *add* operations. The copy operation consists of fetching two values from memory and updating the value of one of these fetched values with the other. The add operation fetches three values from memory and updates one of the fetched values with the sum of the other two fetched values. Cachebench is a benchmark designed to evaluate the performance of the cache memory present on a system. In this performance evaluation, CacheBench was set up to measure the cache memory bandwidth through *read* and *write* operations. In general, CacheBench results show the writing speed is around 60%-80% faster than the reading speed (Figure 2.7e).

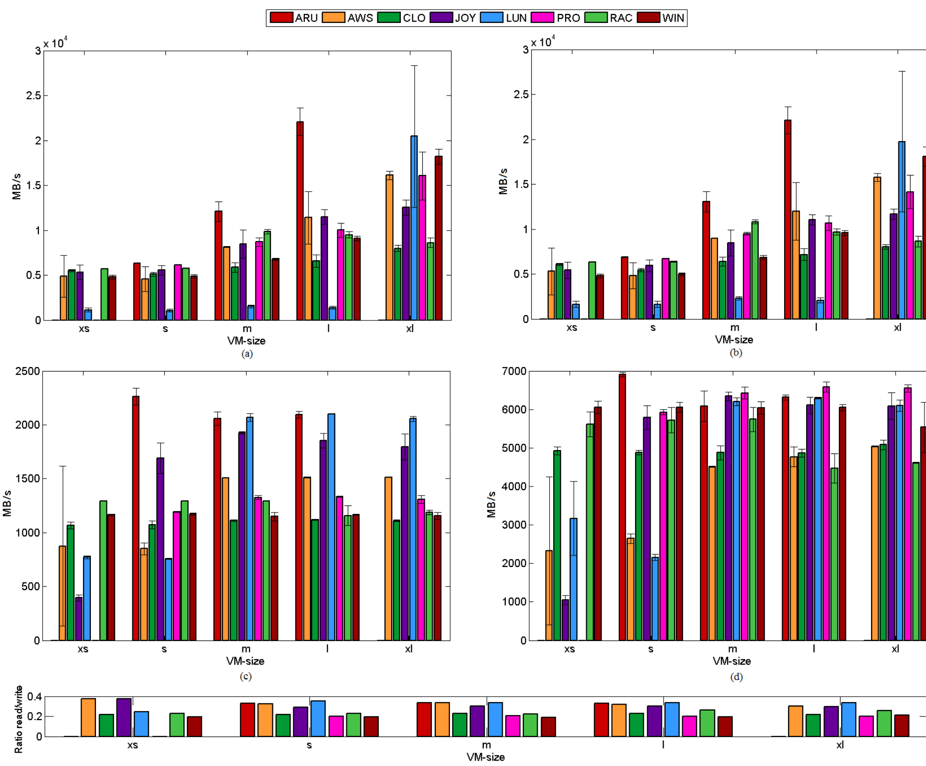

Figure 2.7: Performance of memory benchmarks. (a) Stream results for copy operation (b) Stream results for add operation (c) CacheBench results for read operation (d) CacheBench results for write operation (e) Ratio CacheBench read/write speed.

Storage benchmarks performance

The storage bandwidth has been measured with the *Iozone* and *Threaded I/O Tester (TIO)* benchmarks (Figure 2.8). Iozone is a filesystem benchmark tool. The benchmark generates and measures a variety of file operations. In this performance evaluation, Iozone was used to measure the transaction speed for reading and writing a file of 2GB. Similarly, the read and write speeds have been measured with TIO for a 64MB file by using 16 threads. For the small VM sizes (*xs* and *s*), the read and write speeds are comparable for both benchmarks. For the *m*-, *l*- and *xl*-VM sizes, the read speed is at least ten times faster than the write speed (Figure 2.8c). Iozone's read/write ratio is bigger than 150 for Amazon (Figure 2.8f).

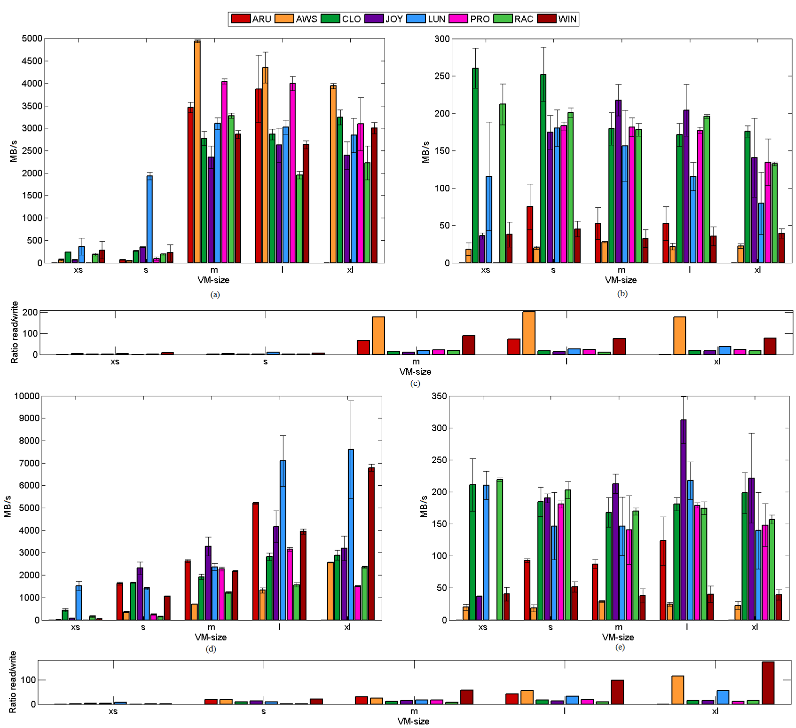

Figure 2.8: Performance of storage benchmarks. (a) Iozone read speed (b) Iozone write speed (c) Ratio Iozone read/write speed (d) TIO read speed (e) TIO write speed (f) Ratio TIO read/write speed.

Variability

The variability of VMs has also been studied. For this, a single value of variability has been calculated by averaging the Coefficient of Variation (CV) of all the benchmark results. The distribution of variability (Figure 2.9) shows that 70.3% of the evaluated VMs have a variability less or equal to 10%. Since physical servers host many VMs at the same time, one should expect that the bigger the VM size, the lower the variability, and vice versa. However, results show that even small VM sizes present low variability values. The percentage of VMs with a variability between 40% and 45% corresponds to the *xs*-VM size of AWS. One possible explanation to this fact is that the number of processor of the AWS's *xs*-VMs is not constant, providing spiky CPU resources [46].

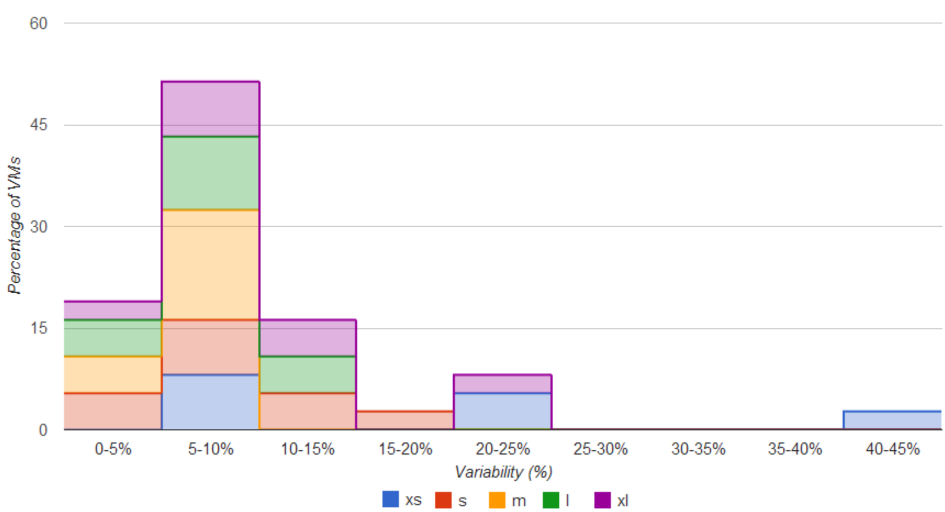

Figure 2.9: Distribution of variability for the measured VMs

Figure of merit of VM cloud performance

No benchmark offers a holistic view through a single score of cloud performance. Instead, benchmarks have their own specific metrics and magnitudes to express results. This heterogeneity in results prevents a straightforward, simple calculation of an absolute figure of merit for VM cloud performance. Moreover, even in the case of benchmarks sharing the same units to express results, it is incorrect to directly add values from different benchmarks. The reason is that the magnitudes of values can differ significantly, for example, read and write cache results differ by a magnitude of three. Therefore, it is not an easy task to choose a cloud provider based on individual benchmark results. In this section, some methods to calculate a figure of merit for VM cloud performance to allow a simple cloud provider selection are presented.

Mean and radar plot as figures of merit

Most of the performance evaluation studies report individual benchmarking results (Table A.1). In an attempt to express the holistic performance of a cloud service through a single score, Li *et al.* [17] propose *Boosting*[1] *Metrics* such as the *mean* (*eg.* arithmetic, geometric, harmonic) and the *radar plot*. The *geometric mean*, by definition, is the n^{th} root of the product of the n units in a data set (Equation 2.1). There is a defect when employing means as the method of boosting metrics: The results from different benchmarks must use the same units. This shortcoming is overcome by using a radar plot.

$$M = \sqrt[n]{\prod_{i=1}^{n} \text{Benchmark}_i} \tag{2.1}$$

$$\text{HB Standardized}_i = \frac{\text{Benchmark}_i}{\text{MAX}(\text{Benchmark}_{1,\ldots,n})} \tag{2.2}$$

[1] The boosting concept comes from the machine learning field. In cloud service evaluation, boosting refers to the creation of a measurement based on primary metrics that measure individual cloud service features.

A radar plot is a simple graphical tool that can depict three or more quantitative values relative to a central point (Figure 2.10). When benchmark results are expressed in different metrics, Li *et al.* propose two standardization methods to express results over a predefined baseline: Higher is Better (HB) (Equation 2.2) and Lower is Better (LB) (Equation 2.3). HB (LB) means the higher (the lower) the benchmark result, the better.

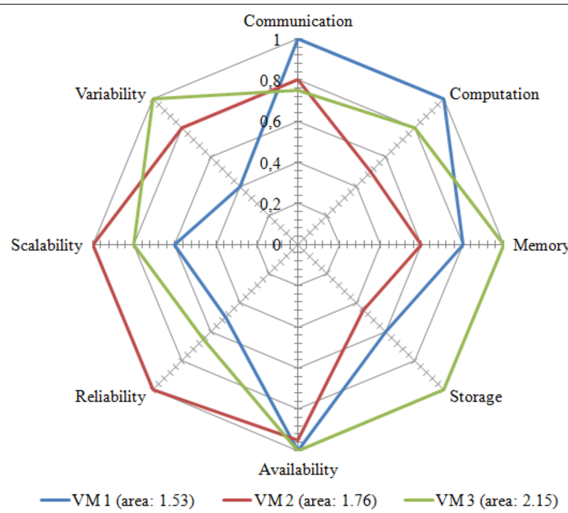

Figure 2.10: Radar plot as a figure of merit

$$\text{LB Standardized}_i = \frac{\frac{1}{\text{Benchmark}_i}}{\text{MAX}(\frac{1}{\text{Benchmark}_{1,\ldots,n}})} \tag{2.3}$$

Here HB Standardized$_i$ and LB Standardized$_i$ refer to the standardized i^{th} benchmark result. Thus, the area of the polygon representing n standardized benchmarking results can be considered as a figure of merit for cloud performance (Equation 2.4) [17].

$$\text{Figure of merit}_{(\text{radar plot})} = \sum_{i=1}^{n} \frac{\sin\left(\frac{2\pi}{n}\right) \times \text{Standardized}_i \times \text{Standardized}_{mod(i+1,n)}}{2} \tag{2.4}$$

Although these metrics result in a figure of merit, they present huge flaws such as lack of weighting and categorical scores[2].

Simple figure of merit

The simple figure of merit for cloud performance is a method similar to the one employed by companies reporting cloud performance such as CloudSpectator or Cloudharmony. It is called simple since it does not take into account the trade-off among the different criteria. In this method, each benchmark result is scaled between two fixed values, A and B. Where A is the lower bound corresponding to the worst performance result (*wpr*) and B is the upper bound corresponding to the best performance result (*bpr*). The intermediate values (x_i) are calculated with Equation 2.5. Then, all the scaled values are averaged and a single figure of merit is obtained for each VM configuration. This method has been applied to the data previously reported (*c.f.* Section 2) to obtain a figure of merit for cloud performance (Figure 2.11). More results based on the simple figure of merit method can be found in Appendix B.

$$\text{Performance score} = \begin{cases} A + \frac{B-A}{\text{bpr}-\text{wpr}}(x_i - \text{wpr}) & \text{if HB benchmark} \\ B - \frac{B-A}{\text{bpr}-\text{wpr}}(x_i - \text{bpr}) & \text{if LB benchmark} \end{cases} \tag{2.5}$$

[2]Score with a limited and usually fixed, number of possible values.

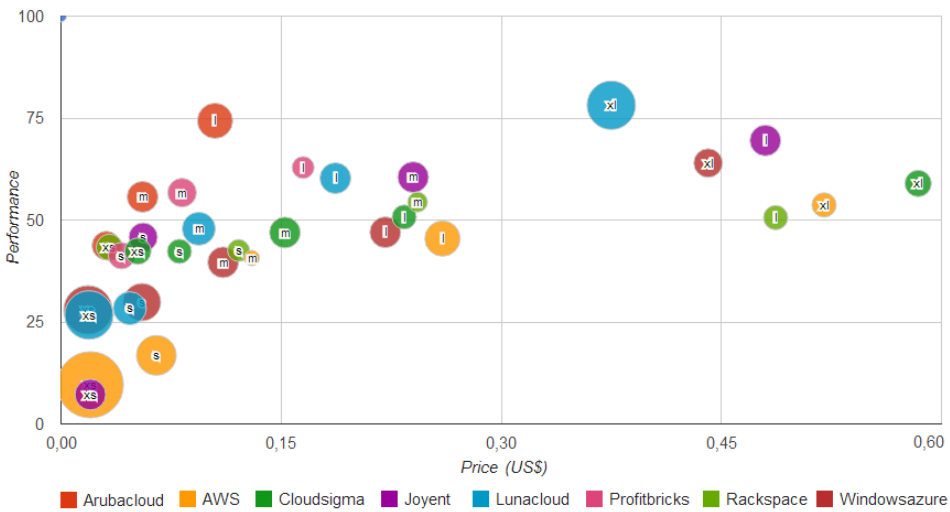

Figure 2.11: Correlation between performance and price for different VM sizes. The variability is represented by the size of the spot. A=1 and B=100.

Figure of merit based on Analytic Hierarchy Process

Analytic Hierarchy Process (AHP) is a structured technique for analyzing, organizing and solving problems related to Multiple Criteria Decision Making (MCDM) [47]. In AHP, complex problems are simplified and structured by arranging the decision factors in a hierarchical structure. The trade-offs among criteria are determined by a pairwise comparison. Unlike the traditional weighted sum-based methods, AHP is based on both subjective and objective evaluation measures. In cloud computing, AHP has been used to rank cloud services [18, 48]. In this section, AHP is used to determine the relative merit of members of a set of alternatives. This process consists of three phases: hierarchy structure modeling, judgement of priorities and hierarchical synthesis.

Phase 1: Hierarchical structure modeling

In this phase, the problem is defined and the goal is determined. Also, all the criteria that have an influence in resolving the issue are identified, as well as the alternatives that offer an answer to the problem. Both criteria and alternatives are organized in a hierarchical structure. The hierarchical structure used here (Figure 2.12) is based on the performance criteria described previously (*c.f.* Section 1). The alternatives correspond to the different cloud providers supported by a cloud broker. Each alternative represents a set of benchmark results that contains a figure of merit for each criterion. In this case, the goal is to find a figure of merit for a cloud infrastructure based on performance.

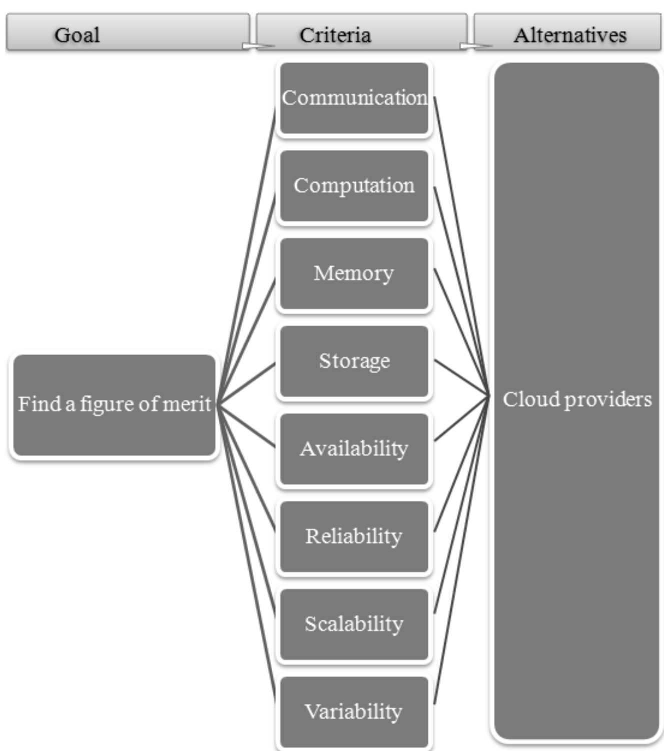

Figure 2.12: Hierarchy of the problem

Phase 2: Judgement of priorities

Pairwise comparisons are used to determine the relative importance of each alternative and each criterion. Saaty [47] proposes a relative rating scale (Table 2.3) by which the decision-maker expresses his opinion about the relative importance of one criterion over another. This scale allows for the quantification of the pairwise comparisons. This phase leads to the construction of P pairwise comparison matrices of size N-*by*-N, where N is the number of alternatives and P is the total number of criteria. One additional matrix C, the pairwise comparison criteria matrix, is constructed to express the relative weights between each one of the criteria to be evaluated.

Intensity of importance	**Definition**	**Explanation**
1	Equal importance	Two activities contribute equally to the objective
3	Weak importance of one over another	Experience and judgement slightly favor one activity over another
5	Essential or strong importance	Experience and judgement strongly favor one activity over another
7	Demonstrated importance	An activity is strongly favored and its dominance demonstrated in practice
9	Absolute importance	The evidence favoring one activity over another is of the highest possible order of affirmation
2,4,6,8	Intermediate values between the two adjacent judgements	When compromise is needed
Reciprocals of above nonzero	If activity i has one of the above nonzero numbers assigned to it when compared with activity j, then j has the reciprocal value when compared with i	

Table 2.3: Relative rating scale [47]

Phase 3: Hierarchical synthesis

Once all comparisons have been made in Phase 2, the numerical probability of each alternative is calculated. This probability determines the likelihood that the alternative has to fulfill the expected goal. This process is also applied to the matrix C that expresses the relative weights between each one of the criteria. The hierarchical synthesis phase is applied to the pairwise comparison matrices as follows:

1. Synthesize the pairwise comparison matrix. Given that the pairwise matrix S is of size N-*by*-N, the synthesized pairwise comparison matrix (A) is obtained by dividing each value of A by the total of its column, as follows:

$$\forall n \in [1...N], \forall i \in [1...N] : a_{ij} = \frac{s_{ij}}{\sum_{i=1}^{N} s_{in}} \tag{2.6}$$

   where a_{ij} is an element of matrix A in row i and column j.

2. Calculate the priority vector (V). The priority vector corresponds to the eigenvector of matrix A. The priority vector can be approximated to the average value of each row of matrix A (Equation 2.7), in order to avoid the mathematical effort required to calculate an eigenvector [49].

$$\forall v \in [1...N] : v_i = \frac{\sum_{i=1}^{N} a_i}{N} \tag{2.7}$$

3. Calculate the maximum eigenvalue (λ_{max}). λ_{max} is calculated by adding the product of each element of vector V by the sum of its corresponding column of matrix S (Equation 2.8).

$$\lambda_{max} = \sum_{j=1}^{N} v_j \cdot \sum_{i=1}^{N} s_{ij} \tag{2.8}$$

4. Calculate the Consistency Index (CI):

$$\text{CI} = \frac{\lambda_{max} - N}{N - 1} \tag{2.9}$$

5. Check the consistency of the pairwise comparison matrix (S). Saaty [50] suggests the Consistency Ratio (CR) in order to determine if the pairwise comparisons made by the decision maker are consistent. For example, consider three criteria x, y and z. If the decision maker has considered $x>y$ and $y>z$, then it would be inconsistent to consider that $x<z$. The CR is calculated as follows:

$$\text{CR} = \frac{\text{CI}}{\text{RI}} \tag{2.10}$$

where the Random Consistency Index (RI) is a fixed value provided by Saaty [50] (Table 2.4). The decisions are considered as consistent when $CR < 0.1$

N	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
RI	0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49

Table 2.4: Random Consistency Index (RI) [50]

Finally, the overall performance for each alternative is calculated. For this, a matrix L of size P-*by*-H is formed. In matrix L, each row corresponds to one of the H priority vectors found for each one of the P criteria. The overall performance (OP) corresponds to the product of the priority vector (G) of matrix C and the matrix L (Equations 2.11 and 2.12). Each element of vector OP corresponds to the performance of one of the H alternatives.

$$\text{OP} = GL \tag{2.11}$$

$$\mathrm{OP} = \begin{pmatrix} g_1 & g_2 & \cdots & g_P \end{pmatrix} \begin{pmatrix} l_{1,1} & l_{1,2} & \cdots & l_{1,H} \\ l_{2,1} & l_{2,2} & \cdots & l_{2,H} \\ \vdots & \vdots & \ddots & \vdots \\ l_{P,1} & l_{P,2} & \cdots & l_{P,H} \end{pmatrix} \tag{2.12}$$

Case study: CPU-intensive application

The objective of this case study is to find a single figure of merit for cloud performance for a CPU-intensive application (*e.g.* file encryption, encoding, scientific computing). In order to compare the different approaches previously presented, a figure of merit with the radar plot has been computed (*c.f.* Section "Figure of merit of VM cloud performance"), using the simple and the AHP techniques. The criteria considered were: computation, memory, storage, availability, scalability and variability. In order to calculate a figure of merit based on real data, the performance values previously obtained have been used and the availability values have been obtained from cloud providers' websites. The availability used is the one for which the cloud provider will not reimburse the end-user in case of service unavailability. This study has been limited to *s*-size VMs. The implementation details per technique are the following:

1. *Radar plot*: Benchmark results are standardized with Equations 2.2 and 2.3. A single figure of merit is calculated with Equation 2.4 (Figure 2.13a).

2. *Simple figure of merit*: Benchmark results are scored with Equation 2.5. Then, the single figure of merit corresponds to the mean value of the scored values for performance (Figure 2.13b).

3. *Figure of merit based on AHP*: The benchmark results are standardized with Equations 2.2 and 2.3. Mean values of benchmarks evaluating cloud performance for the same criterion have been found (*e.g.* in the case of the computation criterion, the mean value of standardized results of 7-Zip and C-Ray was calculated). The pairwise comparison criteria matrix (Table 2.5) considers computation and variability criteria with an equal importance. The

computation criterion is also far more important than the storage and scalability criterion. The memory criterion is slightly more important than the storage, availability and scalability criterion. The availability criterion is considered more important than the storage and the scalability criterion. The priority vector represents the relative importance of each criterion in the single figure of merit.

Criteria	Computation	Memory	Storage	Availability	Scalability	Variability	Priority vector
Computation	1	6	9	3	9	1	0.3753
Memory	1/6	1	3	3	3	1/6	0.1250
Storage	1/9	1/3	1	1/3	1	1/6	0.0408
Availability	1/3	1/3	3	1	3	1/3	0.1053
Scalability	1/9	1/3	1	1/3	1	1/3	0.0501
Variability	1	6	6	3	3	1	0.3036

Table 2.5: Pairwise comparison criteria. CR = 0.0702 and RI = 1.24.

The priority vectors are calculated for each criterion based on the these mean standardized values. The priority matrix for assessment of the overall cloud performance (Table 2.6) presents the overall performance for each cloud provider (Figure 2.13c).

Rank	Criteria/ Provider	Computation	Memory	Storage	Availability	Scalability	Variability	Priority vector
1	**ARU**	0.2507	0.2304	0.3016	0.1788	0.2726	0.2564	0.2455
6	**AWS**	0.0978	0.1152	0.0912	0.1128	0.1151	0.0995	0.1027
2	**CLO**	0.1319	0.1230	0.1334	0.1233	0.1649	0.1501	0.1371
3	**JOY**	0.1319	0.1334	0.1334	0.1233	0.1457	0.1323	0.1320
7	**LUN**	0.0765	0.0720	0.1061	0.1128	0.0641	0.0853	0.0830
4	**PRO**	0.1284	0.1310	0.1042	0.1128	0.0832	0.1040	0.1164
5	**RAC**	0.1074	0.1100	0.0882	0.1233	0.0890	0.1294	0.1144
8	**WIN**	0.0753	0.0850	0.0419	0.1128	0.0653	0.0431	0.0688

Table 2.6: Overall cloud performance matrix

Besides finding a figure of merit for cloud performance, the cost plays an important role in cloud service selection. For this reason, the performance-price ratio has been considered (Figure 2.13d). The results show that for the three methods to compute a figure of merit, ArubaCloud s-size VMs present the best performance among all the evaluated cloud providers. In the case, of the radar plot and the simple figure of merit approaches, the performance results for Joyent are close to those of ArubaCloud (Figure 2.13a-b). However, with the AHP approach, it can be clearly seen that in the case of a CPU-intensive application, ArubaCloud doubles in performance when compared to Joyent (Figure 2.13c).

Summary

In this chapter, cloud performance has been evaluated by using micro-benchmarks. The results obtained from benchmarks have been used to calculate a single figure of merit for cloud performance with the radar plot, the simple and the Analytic Hierarchy Process (AHP) techniques. The advantage of the AHP technique, in comparison to other techniques when calculating figures of merit, is that it is based on pairwise comparisons in which users can express the importance of one feature over another. Thus end-users can take into account the requirements an application has in terms of performance criteria.

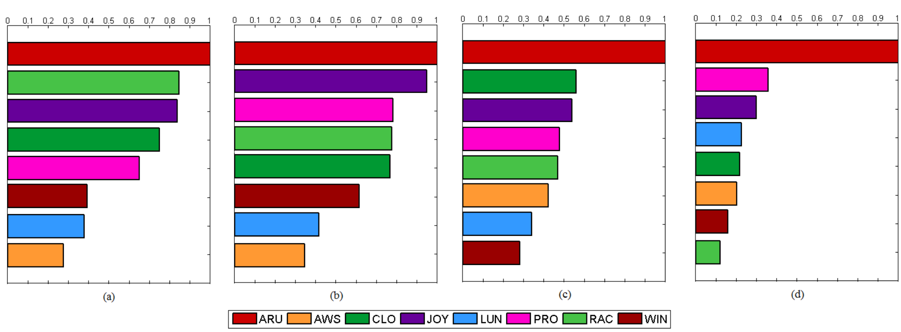

Figure 2.13: Comparison of figures of merit techniques for s-VM size. (a) Radar plot (b) Simple figure of merit, A = 0 and B = 1 (c) Figure of merit based on AHP (d) Performance-price ratio based on AHP performance values. The values of the four figures have been normalized for ease of comparison.

Chapter 3

An exact approach for optimizing placement in cloud brokering

Introduction

In the near future, cloud brokers may become the online travel planning companies for cloud computing. Several years ago, the travel industry was in the same situation as the current cloud computing industry. Travelers left planning in the hands of travel agencies who had agreements with airline companies and hotels in order to obtain reduced overall prices. This situation changed with the advent of travel planning websites that provide instant online comparisons for millions of flights on thousands of airlines. Similarly, the current service typically provided by consulting companies in helping end-users with the placement of cloud infrastructure may, in the near future, be provided by cloud brokers. Cloud brokers could provide the service of discovery and comparison of cloud provider offers and even the automatic and optimal placement of cloud resources.

Placement in cloud brokering refers to the techniques used to efficiently distribute infrastructure resources across multiple cloud providers (*c.f.*

Section "Placement in cloud brokering"). Cloud brokers may react to new situations when conditions change, dynamically repositioning or deploying new cloud infrastructure in order to maintain the performance of end-users' applications. Examples of scenarios in which a cloud broker may trigger placement algorithms in order to compute a new infrastructure topology include:

- *Changes in cloud market conditions*: For example, introduction of new VM configurations, change in prices, apparition of a new cloud provider or implementation of a new pricing model. In this scenario, a cloud broker could determine the impact of the changes of market conditions on the economies or performance of end-users applications. In the case of a positive impact, the end-user can be advised to migrate its cloud infrastructure.
- *Unexpected changes in cloud infrastructure*: Outages may strongly impact economies of end-users' running cloud applications. Although cloud providers offer economic compensation to end-users having experienced an outage, in most cases, this compensation is negligible in comparison to the impact of having a service unavailable (*e.g.* an e-commerce website down). Thus, cloud brokers may not only automatically redeploy infrastructure in recovery scenarios but also minimize the time an application is inaccessible.

This chapter is organized as follows. Goal programming, a technique to solve Multiple Criteria Decision Making (MCDM) problems is briefly described in Section "Goal programming". An exact approach for optimizing placement in cloud brokering is presented in Section "An exact approach for the placement problem". A case study considering an online trading platform is presented in Section "Case study: Online trading plataform".

Goal programming

The optimization goal in MCDM problems is to find an efficient (but not necessarily an optimum) solution by considering multiple objectives (or goals) that can possibly conflict with each other. Thus, MCDM problems contrast with Linear Programming (LP) problems which optimizes a single linear objective. Here, goal programming as a technique for solving MCDM problems has been considered. Goal programming is usually carried out using either the *weighted* or the *preemptive* method.

The weighted method transforms a MCDM problem into a standard LP. A weighted objective function corresponds to a weighted sum of functions representing the multiple objectives of the problem. The weight determines the priority of each objective. Although the computation of the weighted method is easy, there are drawbacks:

- Weighting is subjective and may result in under- or over-rating the contribution of objectives.
- The objectives may be expressed in different metrics or different orders of magnitude that prevent a straightforward calculation of the objective's function.

The preemptive method considers a MCDM problem as a set of multiple LPs with different priorities assigned by the end-user. Thus, each LP is optimized one at a time from the highest to the lowest priority (Figure 3.1). Between LP executions, the optimum value is added as a constraint to the successive LP model. This guarantees that the optimum value of a higher priority objective is not degraded by a lower priority objective. This process continues until the lowest priority is optimized. In cases where a limited amount of degradation is acceptable, a constraint is added as an inequality that allows higher priority solutions to be in the near region for the optimal solution.

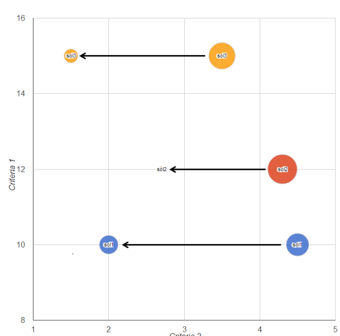

Figure 3.1: Preemptive method. Optimization priority: Criterion 1, Criterion 2, Criterion 3 (represented by the size of the spot).

An exact approach for the Placement problem

The cloud VM placement problem in cloud brokering can be represented as a constrained Knapsack problem: Given a set of VMs, each with a configuration, price and performance, determine the number of each VM configuration to provide so that the provisioning infrastructure is more than or equal to the end-user request (*i.e.* request is satisfied) whilst providing the most cost effective solution. The cloud placement problem is formulated as a Mixed-Integer Linear Programming (MILP) problem and the preemptive goal programming analysis is performed to solve this problem.

Parameters

- *End-user request parameters:*
  - $reqCPU$: number of vCPUs.
  - $reqMEM$: memory capacity.
  - $reqSTO$: storage capacity.
  - $reqNET$: network capacity.
  - $reqRTT$: average latency between the cloud providers and the application customers.
  - $reqAVA$: average availability.
  - $reqREL$: average reliability.
  - $reqSCA$: average VM provisioning time.
  - $reqVAR$: average variability.
  - $reqPER$: performance required by the end-user.
  - $LOCmax_k$: maximum percentage of resources that can be allocated to cloud provider k.
  - $VMmax$: maximum number of VMs.
  - *Pricing model*: on-demand, reserved or spot.

- *Cloud provider parameters:*
  - V: j-by-k matrix composed of v_{jk} elements. Where $v_{jk} = 1$ if and only if the VM configuration j exists at the cloud provider k, otherwise $v_{jk} = 0$.
  - CPU_{jk}: the number of vCPUs of the VM configuration v_{jk}.
  - MEM_{jk}: the memory capacity of the VM configuration v_{jk}.
  - STO_{jk}: the storage capacity of the VM configuration v_{jk}.
  - NET_{jk}: the bandwidth capacity of the VM configuration v_{jk}.
  - $Price_{jk}$: the price per unity of time for running a VM configuration of type v_{jk}.
- *Cloud broker parameters:* Parameters measured or calculated by the cloud broker.
  - p: index of the smallest VM configuration (in terms of computing, memory and storage).
  - η_k: the number of VMs of type p that guarantees the fulfillment of the request for the cloud provider k.

$$\forall k \in [1, K] :$$
$$\eta_k = max\left(\left\lceil \frac{reqCPU}{CPU_{pk}} \right\rceil, \left\lceil \frac{reqMEM}{MEM_{pk}} \right\rceil, \left\lceil \frac{reqSTO}{STO_{pk}} \right\rceil\right) \tag{3.1}$$

  - N: Taking into consideration the possibility of cloud providers having unlimited resources, N is an upper bound that limits the set of solutions. Thus, N represents the maximum number of any kind of VM configuration used to fulfill the request. This parameter is set in case of the $VMmax$ is not specified by the end-user.

$$N = max(\eta_k) \tag{3.2}$$

- RTT_k: the latency of the cloud provider k.
- α_k : average availability of a cloud provider k.
- β_k : average reliability of a cloud provider k.
- γ_{jk}: average time to provision a VM of type j at cloud provider k.
- cv_{jk}: average variability of a VM of type j at cloud provider k.
- $Performance_{jk}$: the performance of a VM configuration of type v_{jk}.

Variables

- *Binary variables:*
  - $x^n_{jk} = 1$: If and only if the VM n of type j is used and belongs to the cloud provider k, otherwise $x^n_{jk} = 0$.
- *Real variables:*
  - TCC: Total Computing Capacity. Amount of computing capacity for a particular solution.

$$TCC = \sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{n=1}^{N} CPU_{jk}x^n_{jk} \tag{3.3}$$

– *TMC*: Total Memory Capacity. Amount of memory capacity for a particular solution.

$$TMC = \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{n=1}^{N} MEM_{jk} x_{jk}^{n} \tag{3.4}$$

– *TSC*: Total Storage Capacity. Amount of storage capacity for a particular solution.

$$TSC = \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{n=1}^{N} STO_{jk} x_{jk}^{n} \tag{3.5}$$

– *TVM*: Total VMs with a particular solution.

$$TVM = \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{n=1}^{N} x_{jk}^{n} \tag{3.6}$$

– *TDC*: Total Deployment Cost. Total cost for deploying an infrastructure across multiple cloud providers.

$$TDC = \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{n=1}^{N} Price_{jk} x_{jk}^{n} \tag{3.7}$$

– *TP*: Total performance of a particular solution.

$$TP = \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{n=1}^{N} Performance_{jk} x_{jk}^{n} \tag{3.8}$$

- *TDT*: Total Deployment Time. Total time for deploying an infrastructure across multiple cloud providers.

$$TDT = \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{n=1}^{N} \gamma_{jk} x_{jk}^{n} \tag{3.9}$$

- *TV*: Total Variability. Total variability for a particular solution.

$$TV = \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{n=1}^{N} cv_{jk} x_{jk}^{n} \tag{3.10}$$

Goal

Preemptive goal programming analysis is performed to solve the MILP depending on the scenario. For instance, in a disaster recovery scenario the criteria are prioritized as follows:

1. Minimize the real variable *TDT*.
2. Minimize the real variable *TDC* constrained by the minimal deployment time previously obtained.
3. Maximize the real variable *TP* constrained by the minimal deployment time and the total deployment cost previously obtained.

This method guarantees a minimal deployment time and suboptimal cost and performance values for the infrastructure to be provisioned.

Constraints

The constraints associated are the following:

- *Physical constraint*: This constraint guarantees VMs are allocated to an existing VM configuration.

$$x_{jk}^{n} \leq v_{jk} \tag{3.11}$$

- *VM configuration constraints*:

$$TCC \geq reqCPU \tag{3.12}$$

$$TMC \geq reqMEM \tag{3.13}$$

$$TSC \geq reqHD \tag{3.14}$$

$$VMmax \geq TVM \tag{3.15}$$

- *Load Balancing constraints*:

$$\begin{gathered} \forall k \in [1, K]: \\ \sum_{j=1}^{J} \sum_{n=1}^{N} CPU_{jk} x_{jk}^{n} \leq LOCmax_k \cdot TCC \end{gathered} \tag{3.16}$$

$$\sum_{j=1}^{J}\sum_{n=1}^{N} MEM_{jk}x_{jk}^{n} \leq LOCmax_k \cdot TMC \tag{3.17}$$

$$\sum_{j=1}^{J}\sum_{n=1}^{N} HD_{jk}x_{jk}^{n} \leq LOCmax_k \cdot TSC \tag{3.18}$$

- *Availability and reliability constraint*:

$$\begin{gathered}\forall k \in [1, K] : y_k = \sum_{j=1}^{J}\sum_{n=1}^{N} x_{jk}^{n} \Rightarrow \\ \sum_{k=1}^{K} \alpha_k \cdot y_k \leq reqAVA \cdot \sum_{k=1}^{K} y_k \\ \sum_{k=1}^{K} \beta_k \cdot y_k \leq reqREL \cdot \sum_{k=1}^{K} y_k\end{gathered} \tag{3.19}$$

- *Latency constraint*:

$$\sum_{k=1}^{K} RTT_k \cdot \sum_{j=1}^{J}\sum_{n=1}^{N} x_{jk}^{n} \leq reqRTT \cdot TVM \tag{3.20}$$

- *Scalability constraint*:

$$\sum_{j=1}^{J}\sum_{k=1}^{K}\gamma_{jk}\cdot\sum_{n=1}^{N}x_{jk}^{n}\leq reqSCA\cdot TVM \tag{3.21}$$

- *Variability constraint*:

$$\sum_{j=1}^{J}\sum_{k=1}^{K}cv_{jk}\cdot\sum_{n=1}^{N}x_{jk}^{n}\leq reqVAR\cdot TVM \tag{3.22}$$

Case study: Online trading platform

Bezimie is a London based company that provides an online trading platform. Using Bezimie's application, traders can purchase and sell stocks and currencies easily through its web interface. The main competitive advantage of Bezimie regarding other online trading platforms is its low latency[1] when placing market orders or reporting quote prices for stocks and currencies to traders. Bezimie manages its cloud infrastructure with the help of the CompatibleOne cloud broker. The current cloud infrastructure topology of Bezimie consists of multiple VMs deployed across two providers (Amazon and Rackspace) with datacenters in Ireland and England. The cloud providers have been chosen due to their proximity to Bezimie's clients, resulting in a low average latency ideal for online trading (Table 3.1).

[1]The network latency is particularly important in financial instruments with a high price variation (volatility). Even a price variation of just a few cents may represent large amounts of money when trading in high volume. Moreover, higher latency connections are more prone to packet delivery delays and loss.

Bezimie is planning to expand its portfolio of clients to France. After some tests, Bezimie's IT department notes that the latency of French traders to its UK-based cloud infrastructure is acceptable ($\leq$ 125 ms) for online trading but greater than the latency of Zimie ($\leq$ 110 ms) its French counterpart and direct competitor. Therefore, in order to be competitive in the French market, Bezimie's IT department compares different solutions to serve its French traders with help of the CompatibleOne broker. The first optimization priority is to minimize the latency between cloud providers and traders; the second priority is to minimize the cost of the requested infrastructure; the third is to maximize the performance of the VMs acquired in the future. Bezimie's IT department chooses the best performing solution (19.6) with the lowest latency in France ($\leq$ 82 ms) at the lowest cost (2.5 US$ per hour) for serving its French traders (Figure 3.2). However, this solution places all the infrastructure serving French traders into one cloud provider (ArubaCloud). This represents a serious risk in terms of of cloud service outages.

Cloud provider	RTT to England (ms)	RTT to France (ms)
ARU	127	82
AWS	95	112
CLO	135	105
JOY	105	101
LUN	110	91
PRO	130	110
RAC	85	97
WIN	122	123

Table 3.1: RTT from cloud providers to current and future Bezimie's client portfolio

Bezimie's IT department also simulates disaster recovery scenarios through the CompatibleOne cloud broker. In disaster scenarios, the main priority for Bezimie is to minimize the time the service is offline while keeping an acceptable latency. The second priority is to minimize the cost of the required infrastructure. The solutions for provisioning time optimization in case of an ArubaCloud outage are presented in Figure 3.3. The figure shows the two optimization stages for different LOCmax values. Solutions resulting from the first optimization stage (minimization of the provisioning time) are shown on the right and solutions resulting from the cost optimization stage are shown on the left. Note that cost optimization brings cheaper but higher latency and more variable solutions.

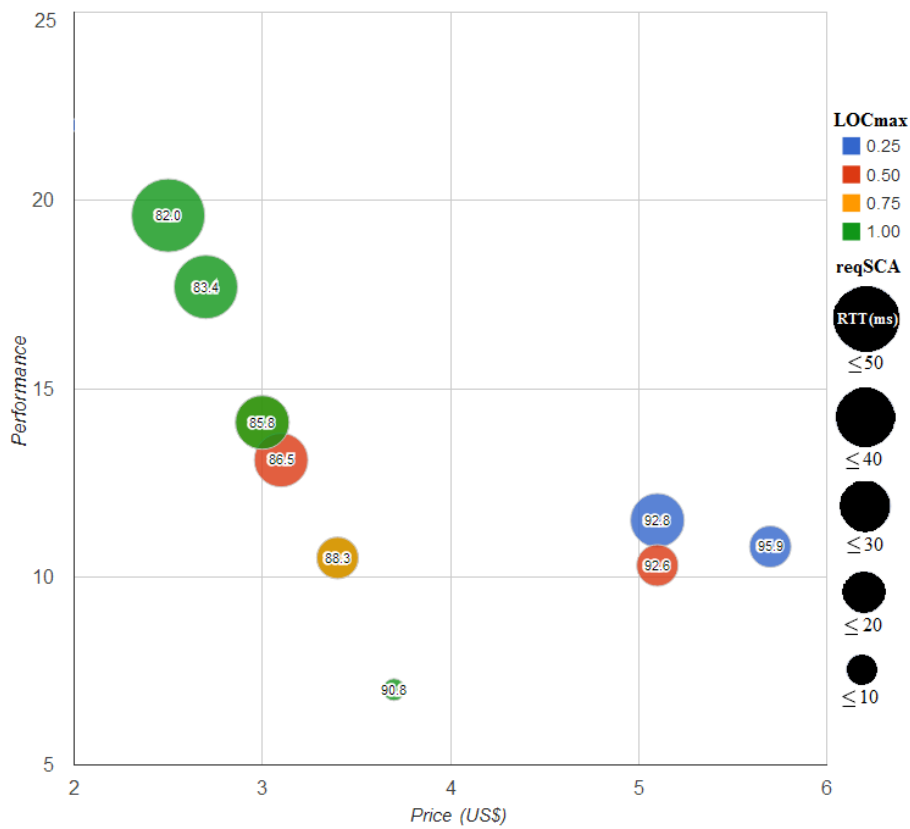

Figure 3.2: Preemptive optimization solutions for latency. The size of the spot represents the average provisioning time in seconds (reqSCA). The figure of merit used here is the same as the one obtained in the previous case study (*c.f.* Section 2). Parameters: reqCPU = 80, reqMEM = 60, reqSTO = 300, reqRTT $\leq$ 110ms.

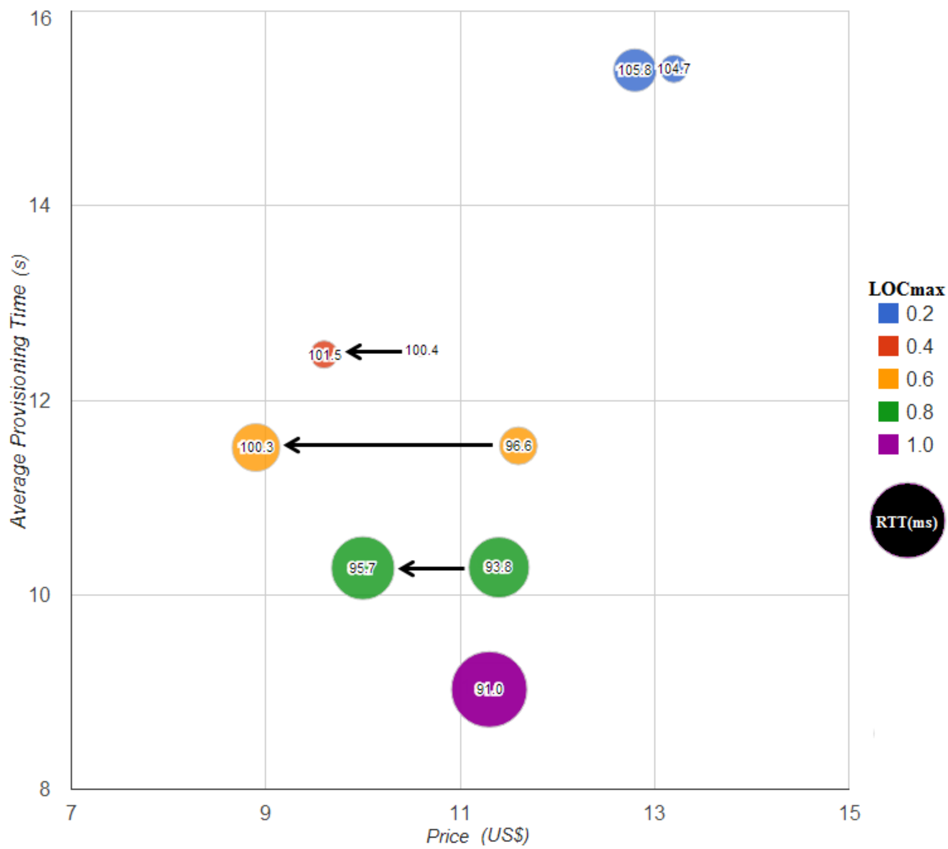

Figure 3.3: Preemptive optimization solutions for provisioning time. The size of the spots represents the solution variability. Parameters: reqPER = 20, reqRTT $\leq$ 110ms.

Part II

A new pricing model in cloud brokering

Chapter 4

The pay-as-you-book pricing model

Introduction

The most popular pricing models used by the current cloud providers are: *pay-as-you-go* and *subscription-based*. *Pay-as-you-go* involves a high price per unit hour but does not require long-term commitment. The subscription-based pricing models are cheaper than pay-as-you-go in the long-term but normally require a long-term commitment and associated vendor lock-in. In this chapter, the currently employed pricing models in cloud computing are briefly described (Section 4). Advance Reservations (ARs), an efficient way to guarantee the availability of a given amount of resources for use at a specific time in the future, are studied (Section 4). Then, the concept of *pay-as-you-book* (Section 4), a novel manner of acquiring cloud resources in advance for future use based on ARs, is presented. Pay-as-you-book combines the main advantages presented in pay-as-you-go and subscription-based pricing models: no long-term commitment and low cost, respectively. At the end of this chapter, a case study comparing the impact of different resource allocation policies on the economies of a Virtual Cloud Provider (VCP) is developed (Section 4).

Pricing models in cloud computing

Several economic models from other fields of study have been proposed for Grid Computing [51]. The commodity market, posted price, tender, bargaining and auction models are among the commonly studied economic models employed for managing the resources in the cloud [52]. However, most of them have not been implemented by current cloud providers. *Pay-as-you-go* and *subscription-based* pricing models are among the most popular cloud pricing models applied by current cloud providers [53]. In the pay-as-you-go model, users pay a value proportional to their resource consumption, while in subscription-based pricing models, users must commit to use the service for a given period of time in exchange for a lower price per hour than in pay-as-you-go. Generally, purchased resources through subscription-based pricing models have priority in terms of availability over resources acquired through pay-as-you-go. The following cloud pricing models are currently deployed by IaaS cloud providers:

1. *Freemium:* A product or a service is free of charge, but users must to pay for advanced features. The product or service may be restricted by time, capacity, customer class, features, and so on (*e.g.* Amazon EC2 *Free Tier Instances*).

2. *Usage duration or pay-as-you-go:* Users pay a value proportional to their resource consumption (*e.g.* Amazon EC2 *On-Demand Instances*).

3. *Subscription-based:* Users must commit to use the service for a given period of time, in exchange they pay a lower price in the long term than in pay-as-you-go. This pricing model allows cloud providers to foresee the utilization of their cloud infrastructure in advance and to speed up their Return On Investment (ROI). The resource allocation, in this pricing model, is based on ARs; cloud providers lock resources and guarantee their future availability to end-users [54]. Subscription-based pricing models may be divided into three categories:

   - Flat-fee or flat-rate: Users are charged a fixed fee for a given period of time, regardless of the resource utilization (*e.g.* Amazon EC2 *Heavy Utilization Reserved Instances*).

   - Subscription with quota: Users are charged a fixed fee to subscribe the service and are given a usage quota. If the quota is exceeded, there is an additional charge.

- Subscription without quota: Users are charged a fixed fee to subscribe the service plus an additional extra charge depending on usage (*e.g.* Amazon EC2 *Light* and *Medium Utilization Reserved Instances*).

4. *Market-based:* Users bid for computing power, resources are allocated if the bid exceeds the price fixed by the cloud provider (*e.g.* Amazon EC2 *Spot Instances*). This pricing model is used by cloud providers to sell spare cloud computing capacity.

Users select a pricing model based on their needs (such as computation power, memory and storage capacities, QoS, execution time, budget and so on). Thus, users with time-constrained tasks would be more interested in purchasing *flat-rate* VMs, in order to assure computing power at anytime. On the contrary, users without time-constrained tasks would be willing to acquire VMs through the market-based pricing model. In case of fluctuating and unpredictable loads, VMs are purchased through the pay-as-you-go model.

Advance Reservations

Advance Reservations (ARs) have been introduced as an efficient way to guarantee the availability of a given amount of resources for use at a specific time in the future. Hotel room bookings are a very well known example of ARs. In hotel room bookings, an AR is described by at least three parameters: numbers of rooms to be booked, check-in dates and check-out dates. AR mechanisms have been applied to several problems of resource sharing in computer science such as bandwidth reservation, job scheduling and VM scheduling. A classification of some studies dealing with ARs applied to computer science is presented in Section 4.3.1 below.

Advance Reservation specified by cloud providers

This type of AR is very closely related to the subscription-based pricing model, widely proposed by cloud providers (Section "Pricing models in cloud computing"). This type of reservation operates on a time-interval basis. At the beginning of each time-interval, the end-user may adjust the amount of resources to be reserved by the cloud provider for the next time-interval. Published research studies can be classified into short-term reservation plans [55, 56] (*e.g.* fine granularity of 10-minute/1-hour time-intervals) and long-term reservation plans (*e.g.* multi-year time-intervals) [31, 57]. Niu, D. *et al.* [55] investigated pricing policies for guaranteed bandwidth reservation in a cloud on a short-term basis such as hours or tens of minutes. Requests are characterized by an estimated average bandwidth requirement, its variability and the percentage of the traffic flow to be satisfied with guaranteed bandwidth. As for the cloud provider, it computes the current bandwidth reservation in order to guarantee the required performance in a probabilistic way. It also decides on the reservation fee taking into account the burstiness and the time correlation of the various requests. A similar problem, where a broker is introduced between the cloud providers and the end-users, is also investigated by Niu, D. *et al.* [56]. While the broker sells guarantees to end-users individually, it jointly reserves bandwidth from multiple cloud providers for the mixed demand, exploiting statistical multiplexing to save reservation cost. The problem has been solved using a game theory approach where the equilibrium bandwidth price depends on the demand expectation, its burstiness as well as its correlation to the market.

The long-term reservation plan was first studied by San-Aniceto, I. *et al.* [57]. This approach considered a single cloud provider and proposed an algorithm that selects the number of VMs to be reserved by an end-user while deploying a service in the cloud. In order to cope with request fluctuations and unpredictability, additional resources may be dynamically provisioned with an on-demand plan. The proposed algorithm minimizes the global cost of using a mixture of reserved and on-demand VMs by taking advantage of the different pricing models within the same provider. Chaisiri, S. *et al.* [31] generalized the problem using the context of multiple cloud providers taking into account the uncertainty on end-users future requests and providers' resource prices. They formulated the problem as an integer stochastic program and solved it numerically using various approaches.

Advance Reservation specified by end-users

In this type of AR, end-users have a higher flexibility as they can specify, in addition to their capacity requirements, various time constraints associated with the execution of their tasks. Time constraints can be expressed in terms of various parameters such as start-time, completion time, duration and task deadline. Thus end-users have the opportunity to reserve the estimated required resources for the completion of their tasks without any further commitment. The AR window is defined as the time-interval delimited by the start-time and the deadline of a given AR request. ARs specified by end-users can be classified into the following three categories.

Strict start and completion time

This type of AR is characterized by a duration equal to its AR window. In other words, end-users require the resources at a specified exact time in the future and for a specified duration (Figure 4.1). This type of AR does not leave any flexibility to the cloud provider to reschedule the AR. Several studies have shown that ARs with strict start and completion times lead to high fragmentation of resource availability by increasing the number of time intervals that are left unused [58, 59]. In the case of cloud computing VMs, these time intervals can be used by other types of requests such as spot or on-demand VMs.

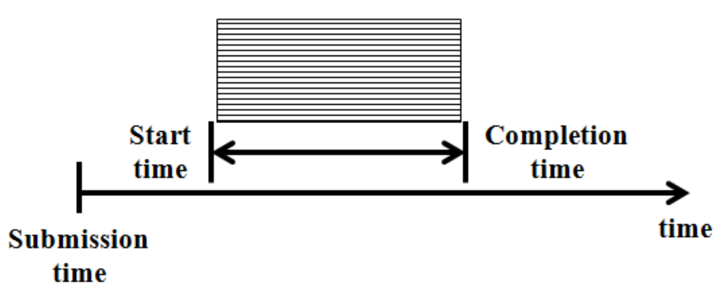

Figure 4.1: Strict start and completion time Advance Reservation

Aoun, R. *et al.* [60] investigated the provisioning of computing, storage, and networking resources in order to satisfy AR requests. They considered several basic services and highlighted how distributed data storage and multicast data transfer can satisfy a larger number of end-users and improve resource utilization of cloud providers. In further studies, the business model of the aforementioned problem has been investigated [61]. The authors proposed and compared three pricing strategies assessing the expectations of both end-users and cloud providers.

Flexible start but strict completion time

This type of AR is characterized by a higher flexibility than the former as the AR window is larger than its execution time. However, these ARs are time-critical and, if accepted, the cloud provider must ensure that they will complete prior to their firm deadline (Figure 4.2). Thus, cloud providers may use various mechanisms to efficiently arrange, manage, and monitor their resources. For instance, Lu, K. *et al.* [54] introduced a model based on computational geometry that allows cloud providers to record and efficiently verify the availability of their resources during the SLA negotiation and planning phase. According to this model, when the cloud provider lacks resources, a flexible alternative solution, referred to as counter-offer, can be generated in order to satisfy the end-user. Hence, the cloud provider's reputation can be enhanced by improving its ability to satisfy as many end-users as possible leading to higher resource utilization and consequently higher profits. Venugopal, S. *et al.* [62] investigated a negotiation mechanism that allows both parties (cloud providers and end-users) to modify the SLA or to make counter proposals in order to converge on a mutually acceptable agreement. In the investigated scenarios, once the SLA has been agreed upon, the cloud provider has to execute the task at the specified time. Numerical simulations have been carried out to highlight the benefit of using time-flexible AR requests. Kaushik, N. *et al.* [63] investigated the impact of the AR window size on the blocking probability and the resource utilization for various models of inter-arrival and service times under the first-come-first-served scheduling policy.

Aoun, R. *et al.* [64] investigated the resource provisioning problem in a market-oriented cloud considering ARs with flexible windows, the size of the AR window being a function of the requirements and the budgets of end-users. The aim of this study is to propose a fair management algorithm that guarantees the QoS requirements of end-users while increasing the expected benefit of cloud providers. For this purpose, the authors introduced a weighted cost function that enables service differentiation, based on disparity in time constraints of the requests. An exact linear formulation, [64] as well as an heuristic approach [65], has been considered for the numerical performance evaluation. Instead of charging fixed prices, Yeo, C.S. *et al.* [66] offer to automatically adjust the price for accessing the resources, whenever necessary, in order to increase the cloud provider revenues. By charging variable prices, cloud providers can give incentives to end-users with less urgent requirements to shift their use to off-peak periods to benefit from lower prices. As the prices are adjusted based on the expected workload and the resource availability, ARs submitted a long time in advance are privileged with cheaper prices compared to late ARs.

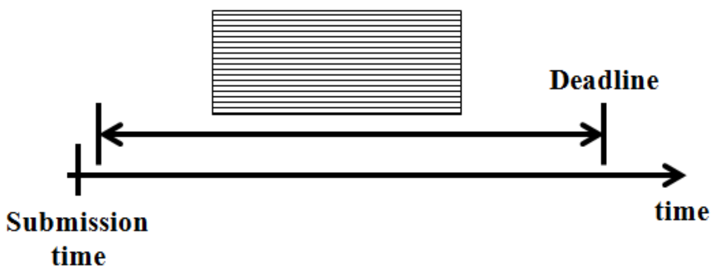

Figure 4.2: Flexible start but strict completion time Advance Reservation

Similar investigations have been carried out in a slightly different environment. The new environment allows the cloud provider to modify the execution schedule of already accepted ARs in order to accommodate new requests right up until each execution starts [67]. Such rescheduling of existing ARs is carried out while respecting the deadline constraints specified in the SLA. The authors have shown that this mechanism can mitigate the negative effects of ARs and improve the performance of reservation-based schedulers as it tends to reduce the amount of time intervals where resources remain free. Another solution to improve resource utilization is to make use of comprehensive overbooking which is particularly efficient in scenarios with no-show policy, AR cancelation [68], and over-estimated execution time of ARs [69]. In this context, rescheduling existing ARs may allow overbooked ARs to get access to the resources during their full execution period if previous ARs do not show or finish earlier. The Earliest Deadline First scheduler has been shown to provide probabilistic real-time guarantees for ARs over time-shared machines [70]. With this scheduling strategy, an admission control policy is developed where new AR requests are accepted if they do not break the QoS constraints of previously accepted reservations. This can be achieved, for instance, by changing the priority of the running ARs to ensure that the execution completes prior to its deadline.

Flexible start and completion time

This type of AR is also characterized by a high flexibility. However, the AR window is not clearly defined. Instead of defining a start-time and a fixed deadline for the execution of each AR, the end-user provides a set of time-intervals along with its preferences represented by a utility function (Figure 4.3). The utility function represents the level of satisfaction that the end-user will experience as a result of the negotiation outcome. This satisfaction may depend on several parameters such as the time of execution, the price of the resources, the delays and the QoS requirements. Not being able to reach an agreement is the worst possible outcome as the end-user receives a null utility from the rejected request. Dynamic pricing based on resource utilization and end-users classification was introduced by Püschel, T. *et al.* [71]. Such dynamic pricing strategies allow adapting the price to set incentives for using the resources during off-peak periods. Two different approaches, which are already well established in other areas, are compared by Meinl, T. *et al.* [72] namely, reservation performed by derivative markets in a perfect competition environment and by yield management techniques assuming an imperfect competition environment. The authors analyze the different requirements in order to apply the proposed approaches in the cloud and provide models to derive the suitable reservation price. Son, S. *et al.* [73] introduced a bilateral negotiation mechanism for cloud service reservation that simultaneously considers price and execution time. Numerical simulations have been used to compare the proposed mechanism for traditional pricing models used by current cloud providers, namely fixed-prices for on-demand and reserved VMs, and variable prices for spot VMs. The Time-of-Use pricing policy has been investigated by Saure, D. *et al.* [74]. According to this policy, the price of accessing resources is totally independent from the utilization ratio of the requested resources but varies within a day. The optimal pricing strategy that maximizes the end-user satisfaction is derived.

In under-estimated ARs, ARs will run for a longer period than expected. Yeo, C.S. *et al.* [66] deal with the problem of under-estimated ARs with *flexible start but strict completion time.* However, in order to enforce future scheduled ARs, ARs are killed once the time period of reservation expires. In this approach, decisions on whether to kill or keep ARs are made by evaluating ARs' SLA constraints.

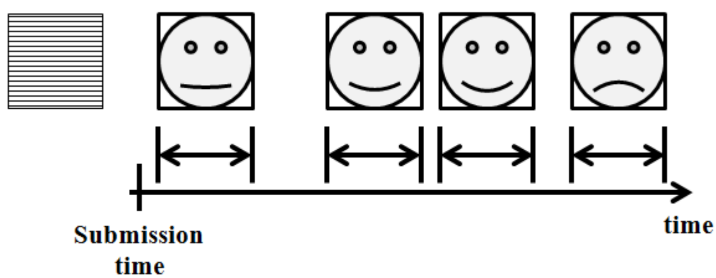

Figure 4.3: Flexible start and completion time Advance Reservation. The mood represents the level of end-user satisfaction for a specific interval.

Pay-as-you-book

Pay-as-you-book is a pricing model between pay-as-you-go and subscription-based. Pay-as-you-book consists of paying and reserving time-slots of VMs in advance without a fixed fee to subscribe to the service and without a long-term commitment, avoiding vendor lock-in, while obtaining lower prices than in pay-as-you-go. Thus, combining the advantages of pay-as-you-go and subscription-based pricing models. Another advantage of pay-as-you-book is a fixed user cost provision, due to the fact that users pay what they have reserved. This also represents an advantage for cloud providers, that could substantially reduce or avoid the use of predictive analytical techniques (*e.g.* modeling, game theory, machine learning, and data mining) when determining utilization patterns. Table 4.1 presents a comparison between the current popular pricing models of cloud providers and pay-as-you-book.

Pay-as-you-book may be applied in scenarios with predictable workloads [75] such as:

- **Time-of-day patterns:** Scenarios with recurring cycles in user resource consumption based on people's behavior, *e.g.* the consumption of IT resources by users of a company can be easily predicted and described by R resources between 8AM to 5PM from Monday to Friday.

- **Industry-specific variability:** Scenarios with predictable variability based on recurrent events, such as tax season, FIFA World Cup and gift purchases for Christmas.

Feature	**Pay-as-you-go**	**Subscription-based**	**Pay-as-you-book**
Cost	High	Low	Medium
User Cost Provision	Variable	Variable/fixed	Fixed if no under-estimated ARs
Reimbursement in case of service unavailability	None	Percentage of the user fees	X times reservation value
Payment terms	In-arrears	Up-front	Up-front or in-arrears
Term commitments	None	Long (From months to years)	Short (Duration of the reservation)
Availability during periods of very high demand	Low	High	Depends on cloud provider's policies
Use of predictive analytics	Unpredictable usage patterns	Necessary and done by cloud provider	Unnecessary, prediction done by the end-user
Type of applications	Unpredictable workloads, spiky	Predictable and continuous usage	Very predictable usage

Table 4.1: Current popular pricing models compared with pay-as-you-book

In pay-as-you-book, an AR Ω^i can be modeled by a set of VMs ω^i_j. Each VM ω^i_j is meant to be used during a specific period of time (*"strict start and completion times"*) and is represented by the tuple $(\alpha^i_j, \beta^i_j, \gamma^i_j)$, where α^i_j denotes the start-time of the VM, β^i_j its stop-time estimated by the end-user, and γ^i_j its real stop-time. An AR is accepted if the set of VMs described in it can be provisioned.

Initial scheduling of Advance Reservations

Since ARs are made prior to VM utilization, the cloud provider can use various scheduling approaches in order to optimize the resource utilization of its infrastructure and consequently increases its revenue. At this stage, the cloud provider only has knowledge of the execution time estimated by end-users. Even though these estimations may be imprecise, the cloud provider has to decide whether to accept ($\varpi^i = 1$) or reject ($\varpi^i = 0$) each AR Ω^i depending on its resource availability. The initial scheduling problem can be formulated as follows. Given that there is a number $\mathcal{N}$ of available VMs and the set of $\mathcal{M}$ ARs, the cloud provider has to determine, for each accepted AR, the physical machine that will host it. This should be carried out while respecting the limited resources of the cloud provider and the fixed start and completion times estimated by the end-users. The main objective of the cloud provider is to maximize the utilization of its resources which can be expressed mathematically as:

$$\mathcal{G} = \sum_{i=1}^{\mathcal{M}} \varpi i \times \sum_{\forall j} \left(\beta_j^i - \alpha_j^i\right) \tag{4.1}$$

The choice of the type of the initial scheduling algorithm and its setup depends on the provider goals. For example, in the case of a cloud provider, the resource allocation goal may aim to minimize the number of physical machines used to host VMs in order to reduce the power consumption, thus reducing the operational expenditures, whilst in the case of a cloud broker reselling VM time from different cloud providers, the resource allocation goal may aim to minimize the cost of resold VMs.

This problem turns out to be similar to the 2-dimensional bin packing problem with rejection. In order to solve this problem, a very straightforward sequential algorithm commonly known as " The Decreasing First Fit" (DFF) algorithm has been used. DFF is a simple offline heuristic algorithm that achieves a near-optimal solution for the classical 1-dimensional bin packing problem [76]. The DFF strategy operates in two phases. First, it sorts the ARs in decreasing order based on their duration $\sum_{\forall j}(\beta_j^i - \alpha_j^i)$. Then, it processes the ARs according to the previous order, and schedules each VM in the first physical host with sufficient remaining capacity during its execution time. If none of the physical hosts can fully accommodate the incoming VM, the AR will be rejected, as previously stated.

Pricing and rewarding end-users

The cloud provider is responsible for guaranteeing the QoS required by the reservations. In return, the cloud provider expects a payment in the form of reward or fee for the successful completion of a reservation. If α_j^i denotes the start-time of a VM and β_j^i its expected stop-time estimated by the end-user, the end-user will be charged a fee $\mathcal{F}_j^i$ equal to $(\beta_j^i - \alpha_j^i) \times \Delta^R$, where Δ^R is the hourly rate of a reserved VM. However, it may occur that a VM is needed for more time than initially estimated ($\gamma_j^i > \beta_j^i$). In this case, the cloud provider can allocate the required resources for a longer period for a higher hourly rate Δ^O on a best-effort basis ($\Delta^O > \Delta^R$). In other words, the cloud provider cannot guarantee the VM availability until the real stop-time γ_j^i θ_j^i denotes the time when the VM is stopped ($\theta_j^i = \gamma_j^i$) or it is forced to terminate by the cloud provider if the VM is reserved for executing another end-user ($\theta_j^i < \gamma_j^i$). In the case of under-estimated reservations, the end-user will be charged a fee $\mathcal{F}_j^i$ equal to $(\beta_j^i - \alpha_j^i) \times \Delta^R + (\theta_j^i - \beta_j^i) \times \Delta^O$.

When the cloud provider accepts an AR, the end-user expects to be able to access the reserved VMs at the specified starting time. However, changes may occur between the time when the end-user submits the reservation and this specified starting time. This can happen for various reasons such as end-users canceling or modifying requests, resource failures, and errors in the estimation of the execution time. Since an AR is a commitment by the cloud provider, failing to meet this commitment may result in the provider having to pay a penalty $\mathcal{P}_j^i$ to the end-user equal to $(\beta_j^i - \alpha_j^i) \times \Delta^P$.

Resource allocation policies

From the previous discussion, three scenarios have been identified: over-estimated ARs (Figure 4.4a), under-estimated ARs without any conflict (Figure 4.4b), and under-estimated ARs resulting in a conflict with other ARs (Figure 4.4c). The first two scenarios are trivial since the cloud provider does not have to intervene and the AR will end normally. However, for the third scenario, a cloud provider motivated by profit has to decide at the arrival of a new AR $\alpha_{j'}^{i+n}$ whether to keep running the under-estimated AR or abort it. In order to tackle this conflictive scenario, three different resource allocation policies have been defined: highest priority to running ARs, highest priority to future ARs and an economic agent.

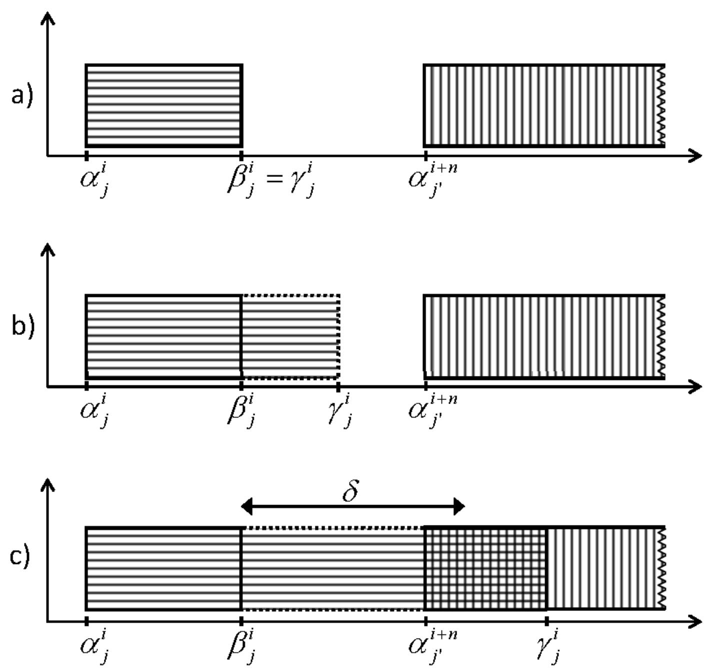

Figure 4.4: Possible scenarios of running Advance Reservations

Strategy 1: Highest priority to running ARs

Under this strategy, the cloud provider will never stop a running VM and always try to postpone the incoming AR that causes the conflict to a later period through negotiations. The only incentive for the end-user to accurately estimate the time of VM utilization is motivated by the lower price of ARs ($\Delta^O > \Delta^R$). This strategy is characterized by a null percentage of dropped ARs during their execution.

Strategy 2: Highest priority to future ARs

Under this strategy, under-estimated VMs are penalized as they are aborted after they have been started if there is a conflict with a future AR. In order to protect their application from forced termination, end-users with critical applications must ensure that the ARs times are sufficient for their applications to be completed. This strategy is characterized by a null percentage of rejected ARs prior to their execution since all accepted ARs are honored by the cloud provider.

An economic agent for maximizing revenues under pay-as-you-book

Under this approach, an agent to manage the conflict between currently running under-estimated ARs and future ARs is proposed. The cloud provider has to first estimate the average extra-time δ required by the currently running under-estimated AR. This can easily be obtained by analyzing the past history of AR executions and hence adjusting δ accordingly. Based on this, the cloud provider predicts that if the under-estimated AR is kept running, it will get an additional fee of $(\delta + \beta_j^i - \alpha_{j'}^{i+n}) \times \Delta^O$ but will have to pay a penalty $\mathcal{P}_{j'}^{i+n}$ equal to $(\beta_{j'}^{i+n} - \alpha_{j'}^{i+n}) \times \Delta^P$. If the under-estimated AR is aborted and the new AR is executed, the cloud provider estimates its gain to be equal to $\mathcal{F}_{j'}^{i+n} = (\beta_{j'}^{i+n} - \alpha_{j'}^{i+n}) \times \Delta^R$. By comparing these two values, the cloud provider will decide on the way to resolve this conflict. If the cloud provider decides to keep the under-estimated AR, it should negotiate with the owner of the incoming AR if it accepts to delay its current execution and get, in exchange, a penalty and a new time slot for executing its AR. In this study, it is assumed that the end-user can accept such a proposal with a probability ρ.

Case study: A Virtual Cloud Provider maximizing revenues through the pay-as-you-book pricing model

In the mobile business, Mobile Virtual Network Operators (MVNOs) offer attractive mobile communication services without having their own infrastructure or spectrum. As in the mobile business, cloud brokers may operate in the near future, like Virtual Cloud Providers (VCPs) by assuming credit risk and by creating new pricing models addressing specific market segments. This case study considers a VCP that resells VMs, reserved in advance, at a flat rate. For this, the VCP reserves a large number of VMs across multiple platforms (quota) at a lower price that announced by the cloud providers and resells them under the pay-as-you-book pricing model.

Experimental setup

For these simulations, a VCP with a fixed number $\mathcal{N}$ of same size VMs reserved across multiple cloud providers is considered. A simulation period of 4 days (or equivalently 96 hours) has been defined. The VCP collects the set of ARs prior to their execution. It is assumed that each AR is composed of a single VM. The start-time α^i_j of a VM is uniformly chosen in the interval $[0, 96]$ while its estimated utilization μ^i_j follows a negative exponential law of mean $\hat{\mu} = 5$ hours bounded by a maximum utilization of 8 hours ($\beta^i_j = \alpha^i_j + \mu^i_j$). The percentage ψ of ARs that are under-estimated varies in the set $\{20\%, 30\%, 40\%, 50\%\}$ and the extra-time required by these reservations λ^i_j also follows a negative exponential law of mean $\hat{\lambda}$ equal to 1 or 2 hours ($\gamma^i_j = \beta^i_j + \lambda^i_j$). Without loss of generality, the value of Δ^R has been fixed to 1. Consequently, the parameters Δ^O and Δ^P can take their values in the sets $\{1, 2, 3, 4, 5\}$ and $\{0.5, 1, 1.5\}$, respectively. Finally, the probability ρ of a successful negotiation between the VCP and the end-users was fixed to 100%.

For each simulation, the percentage is reported:

- $\mathcal{R}_i$ of ARs that were rejected at the end of the offline initial scheduling;
- $\mathcal{R}_d$ of initially accepted ARs that were dropped during their execution because their execution times were under-estimated;

- $\mathcal{R}_r$ of initially accepted ARs that were rejected prior to their execution because the VCP decided to keep running an under-estimated request;
- $\mathcal{R}_a$ of advanced reservations that are accepted and executed during their complete activity period. It is obvious that the following equation holds true:

$$\mathcal{R}_i + \mathcal{R}_d + \mathcal{R}_r + \mathcal{R}_a = 100\% \tag{4.2}$$

as well as:

- the average utilization ratio χ of the VCP resources during the simulation period.
- the revenue Ξ of the VCP computed as a function of Δ^R, Δ^O, and Δ^P;

All the experiments have been repeated 1000 times. The average and the standard deviation computed over these different runs are recorded. In these simulations, the three resource allocation policies previously described (*c.f.* Section "Resource allocation policies") as well as the on-demand approach, have been considered. In the on-demand approach, no ARs are made at all and the resource allocation is performed online. Upon the arrival of a new request, the VCP evaluates its instantaneous resource utilization. If enough free resources are available, the new request is accepted; otherwise, it is rejected. In return, the end-user is expected to pay a higher price Δ^O for accessing the resources as they are not reserved in advance. This approach does not ensure end-user satisfaction with a request for multiple VMs as there is no guarantee that all the VMs will be provisioned.

Results and analysis

Impact of the number of submitted ARs

In the first scenario, the parameters of the simulation have been fixed as follows: $\mathcal{N} = 10$, $\Delta^R = 1$, $\Delta^P = 1$, $\Delta^O = 3$, $\psi = 20\%$, $\hat{\lambda} = 1$, $\rho = 100\%$ (Table 4.2).

	$\mathcal{M}$ = 100 ARs					$\mathcal{M}$ = 200 ARs				
	$\%\mathcal{R}_i$	$\%\mathcal{R}_d$	$\%\mathcal{R}_r$	$\%\chi$	Ξ	$\%\mathcal{R}_i$	$\%\mathcal{R}_d$	$\%\mathcal{R}_r$	$\%\chi$	Ξ
DFF	0.10	0	0	35.45	30 500	7.64	0	0	67.51	58 000
On-Demand	0.20	*	*	38.54	99 750	9.68	*	*	67.01	173 250
Strategy 1	0.10	*	8.17	35.09	33 750	7.64	*	10.01	64.77	61 000
Strategy 2	0.10	8.97	*	37.15	32 750	7.64	11.29	*	69.64	61 000
Economic Agent	0.10	6.42	2.32	37.03	35 250	7.64	8.00	2.91	69.34	64 750

Table 4.2: Impact of the number of submitted Advance Reservations (ARs). A * denotes 0% by default.

As expected, the on-demand approach ensures the highest VCP revenue as the end-users are paying a higher price for the execution of all their tasks ($\Delta^O = 3\times\Delta^R$). It also achieves a high overall acceptance ratio $\mathcal{R}_a$ as it does not have to deal with estimation uncertainties. Notice that both strategies 1 and 2 achieve similar revenue Ξ for the VCP. However, Strategy 1 achieves the highest acceptance ratio $\mathcal{R}_a$ for ARs, while Strategy 2 has a better performance in terms of resource utilization χ. The proposed economic agent achieves slightly lower resource utilization compared to Strategy 2 and keeps the percentage of rejected ARs prior to their execution $\mathcal{R}_d$ at an acceptable value. In summary, the proposed economic agent is a trade-off in terms of resource utilization and acceptance ratio between the intuitive strategies 1 and 2, but outperforms both of them in terms of VCP revenue. These conclusions hold true independently of the number of submitted ARs.

Impact of the percentage under-estimated ARs and their execution extra-time

In the second scenario, the parameters of the simulation have been fixed as follows: $N = 10$, $\Delta^R = 1$, $\Delta^P = 1$, $\Delta^O = 3$, $\mathcal{M} = 200$, $\rho = 100\%$ (Table 4.3).

As the initial scheduling does not have any knowledge about the error in estimating the execution time, it achieves the same performance independently of the values of ψ and $\hat{\lambda}$. As the percentage of under-estimated ARs increases, the percentage of ARs that are rejected prior to their execution in Strategy 1 also increases. However, this increase is less pronounced than the increase observed in the Strategy 2 for the percentage of dropped ARs during their execution. Overall, the proposed economic agent still keeps its superiority whilst achieving a trade-off in terms of resource utilization and acceptance ratio between the strategies 1 and 2, whilst also outperforming both of them in terms of VCP revenue.

In general, the results show that the on-demand approach is better in terms of revenue than the proposed economic agent, and the strategies 1 and 2. Since the main interest for a cloud provider is to maximize its revenues, the obtained results explain why a pricing policy such as pay-as-you-book has not been implemented by cloud providers. Thus, pay-as-you-book may be implemented by a cloud broker taking advantage of multiple cloud providers' service offerings, acting as a VCP.

		$\hat{\lambda} = 1$					$\hat{\lambda} = 2$				
		$\%\mathcal{R}_i$	$\%\mathcal{R}_d$	$\%\mathcal{R}_r$	$\%\chi$	Ξ	$\%\mathcal{R}_i$	$\%\mathcal{R}_d$	$\%\mathcal{R}_r$	$\%\chi$	Ξ
$\psi = 20\%$	DFF	7.64	0	0	67.51	58 000	7.43	0	0	67.58	58 000
	On-Demand	9.68	*	*	67.01	173 250	11.12	*	*	68.31	176 250
	Strategy 1	7.64	*	10.01	64.77	61 000	7.43	*	11.21	65.89	65 000
	Strategy 2	7.64	11.29	*	69.64	61 000	7.43	12.81	*	70.15	61 500
	Economic Agent	7.64	8.00	2.91	69.34	64 750	7.43	5.53	6.51	69.40	67 500
$\psi = 30\%$	DFF	7.44	0	0	67.45	58 000	7.45	0	0	67.49	58 000
	On-Demand	11.10	*	*	68.24	176 250	13.48	*	*	70.66	180 750
	Strategy 1	7.44	*	14.34	63.60	62 000	7.45	*	15.91	65.22	67 750
	Strategy 2	7.44	17.07	*	70.61	62 250	7.45	19.34	*	71.38	63 250
	Economic Agent	7.44	11.89	4.33	70.15	67 750	7.45	9.52	8.09	69.98	71 750
$\psi = 40\%$	DFF	7.51	0	0	67.54	58 000	7.48	0	0	67.48	58 000
	On-Demand	12.68	*	*	69.52	179 250	15.97	*	*	71.27	183 750
	Strategy 1	7.51	*	18.10	62.66	62 750	7.48	*	19.89	64.71	69 750
	Strategy 2	7.51	22.71	*	71.78	63 750	7.48	25.50	*	72.63	62 500
	Economic Agent	7.51	15.67	5.59	71.11	71 000	7.48	12.36	10.27	70.72	75 250

Table 4.3: Impact of the percentage of under-estimated Advance Reservations (ARs) and their execution extra-time. A* denotes 0% by default.

Summary

In this chapter, the problem of resource provisioning while using ARs under the *pay-as-you-book* pricing model has been investigated. The proposed model handles the extra-time required by running ARs at a higher price, on a best-effort basis. Indeed, an extra-time of an AR plan may lead to resource conflicts with other AR plans. In order to resolve such resource conflicts, an economic agent responsible for managing the under-provisioning problem has been proposed. The economic agent aims to achieve provider satisfaction by maximizing its revenues through intelligent resource management. In order to assess the performance of the proposed agent, the proposed economic agent has been compared with two intuitive approaches that systematically prioritize reserved ARs or currently running ARs. The economic agent achieves a trade-off between the two intuitive strategies in terms of resource utilization and acceptance ratio, whilst outperforming both in terms of provider's revenue. These conclusions remain true regardless of the number of submitted ARs, the percentage of under-estimated ARs, and the average duration of the extra-time required.

Chapter 5

Conclusion and future works

The aim of this book has been to propose new value-added services and pricing models in cloud brokering at the infrastructure level. With this in mind, a comprehensive overview of the current and future value-added services in cloud brokering has been provided. After surveying the research related to cloud performance evaluation and placement in cloud brokering (Chapter 1), needs and shortcomings in the current cloud computing service offerings have been identified. In particular, in the first part of this book (Chapters 2 and 3), the problem of a single figure of merit for cloud performance and the problem of VM placement in cloud brokering have been addressed. In the second part of this book, a new pricing model for cloud computing known as pay-as-you-book has been proposed (Chapters 4).

The computation of a single figure of merit for VM cloud performance has been described as a multi-criterion problem (Chapter 2). This problem relies on eight criteria: Communication, Computation, Memory, Storage, Availability, Reliability, Scalability and Variability (Section 1). The weight of these criteria in the computation of a figure of merit for cloud performance depends on the application profile foreseen to run on top of the cloud infrastructure. The Analytic Hierarchy Process (AHP) has been used to analyze and to solve the Multiple Criteria Decision Making (MCDM) problem of finding a single figure of merit for cloud performance. In this case, AHP enables an objective determination of the relative merit of the VM performance criteria for a given set of cloud providers.

Similarly to the problem of finding a figure of merit for cloud performance, the problem of placement in cloud brokering has been described as a multi-criterion problem (Chapter 3). This problem refers to the efficient distribution of cloud infrastructure across multiple and non-interoperable cloud providers. Preemptive goal programming has been used to tackle this problem by defining a set of multiple LPs with different priorities assigned by the end-user. A pricing model between pay-as-you-go and subscription-based known as pay-as-you-book has been proposed (Chapter 4). Contrary to subscription-based pricing models, pay-as-you-book allows for the reservations of cloud resources for future use without long-term commitment. Three resource allocation policies to manage the extra-time required by running reservations under pay-as-you-book have been described and evaluated. Among the evaluated policies, the economic agent maximizes cloud provider's revenue while keeping an acceptable ratio of resource utilization.

Cloud brokers have emerged in the cloud computing landscape as a technical solution to bringing unified self-service access to multiple non-interoperable cloud providers. By bringing interoperability and portability of end-user's applications across multiple cloud providers, cloud brokers act as an ideal doorway to fill the current technical gaps in cloud computing and to introduce new pricing and business models. Technically, cloud brokers already complement or enhance some of the cloud provider service offerings such as infrastructure monitoring, cost optimization, elasticity management and consolidated billing. In this manner, cloud brokers act as a single point of access for consumption of cloud services. With the introduction of new value-added services, such as those presented in this book, cloud brokers may become trusted third-parties, providing un-biased information that benefits end-users.

Current cloud providers look for differentiation through the addition of new value-added services to their portfolios. Similarly to supermarkets, cloud providers have become a place to find aggregated but non-interoperable services. For example, cloud providers do not provide infrastructure monitoring services that monitor the infrastructure of their competitors. In the near future, cloud brokers can take part of the cloud economies by actively changing the value chain of cloud computing. Similarly to the Mobile Virtual Network Operator (MVNO) business model, cloud brokers, without hardware infrastructure, may develop new and appealing pricing models addressing untapped market sectors. By acting as a single point of access for the consumption of cloud services, cloud brokers could set the bar for how much end-users should pay for a given cloud offer depending on the SLA of the cloud provider and its respective performance. Thereby, cloud brokers will increase competition between cloud providers.

The practical implication of this book is threefold. First, the proposed figure of merit can be used to objectively compare cloud providers based on their performance and on the application profile to be deployed. Second, the computation of this figure of merit linked with the proposed knowledge of cloud brokering placement optimizes costs of the distributed resources depending on the end-user constraints. This knowledge may enrich the service portfolio of not only cloud brokers, who could automatically respond to unforeseen scenarios, but also consultancy firms and IT departments who may take data-driven decisions when migrating into the cloud. Third, the proposed pricing model is a first step to the study of mechanisms enabling new and appealing ways of purchasing cloud infrastructure.

This work has identified two areas for possible further study. These include the identification of standard sizes and the establishment of standard SLAs for cloud VMs. The definition of standard VM sizes solves the problem of product differentiation created by current heterogeneous VM service offerings from cloud providers. Thus, the challenge is to identify a measurement for VM configurations that satisfies the largest demand from end-users by taking into account the different application profiles. Cloud SLAs vary from one cloud provider to another. In order to enable the comparison of service offerings, SLAs terms and definitions need to be standardized across cloud providers. In summary, standard SLAs and standard VM sizes along with our proposed figure of merit, contribute to the commoditization of cloud VMs.

Appendix A

Studies related to cloud providers performance evaluation

Table A.1: Studies related to cloud providers performance evaluation

Study	Type of benchmark	Applications or Suite/Benchmarks	Property	Metric
Stanchev [77]	Synthetic	WSTest	Overall performance	Transactions per second
Yigitbasi et al. [78]	Application	Modified Grenchmark	Overall performance	Queue waiting time (s)
		Modified Grenchmark	Overall performance	Response time (s)
	Micro	Benchmark developed by authors	Elasticity	VM adquisition and release (s)
Dejun et *al.* [79]	Application	CPU-intensive web	CPU	Duration of operation (ms)
		Database read-intensive	Disk	Duration of operation (ms)
		Database write-intensive	Disk	Duration of operation (ms)
Baun and Kunze [13]	Application	Compilation Linux Kernel	CPU	Duration (s)
	Micro	Bonnie++	Disk	KB/s
		Bonnie++	Disk	Number of file operations/s
		iperf	Network	Transfer rate (KB/s)

Continued on next page

Table A.1: (Continued)

Study	**Type of benchmark**	**Applications or Suite/Benchmarks**	**Property**	**Metric**
		ping	Network	RTT (ms)
Alhamad *et al.* [14]	Application	Java application	Network	Response time (ms)
El-Khamra *et al.* [80]	Application	EnKF-based matching	CPU	Total execution time (s) per number of processor cores
Hill *et al.* [15]	Synthetic	TPC-E	Overall performance	Average transaction time (s)
	Micro	Not specified	Elasticity	VM adquisition and release (s)
		Not specified	Elasticity	Time per web role action (s)
		Not specified	Network	RTT (ms)
		Not specified	Network	Bandwidth (MB/s)
Schad *et al.* [12]	Micro	Ubench	CPU	Ubench score
		Ubench	Memory	Ubench score
		Bonnie++	Disk	Total execution time (KB/s)
		iperf	Network	TCP throughput (MB/s) intra-datacenter network

Continued on next page

Table A.1: (Continued)

Study	Type of benchmark	Applications or Suite/Benchmarks	Property	Metric
		Application developed by authors	Elasticity	VM adquisition (s)
He *et al.* [4]	Application	CSFV	CPU	Total execution time (s)
	Synthetic	NPB	CPU	Total execution time (s)
		HPCC/HPL	CPU	GFLOPS
	Micro	iperf	Network	Message Latency (s)
		iperf	Network	TCP throughput (bytes/s) intra-datacenter network
Moreno-Vozmediano *et al.* [11]	Synthetic	GridNPB/ED	Overall performance	Throughput (jobs/s)
		NAS/NGB	Overall performance	Throughput (jobs/s)
VÃűckler *et al.* [81]	Application	Scientific workflow application	Overall performance	Jobs/s
Ostermann *et al.* [82] extended to 4 providers in [10]	Synthetic	HPCC/HPL	CPU	GFLOPS
		HPCC/RandomAccess	Network	MB/s
	Micro	lmbech/all	Many	Many
		HPCC/DGEMM	CPU	GFLOPS
		CacheBench	Memory	MB/s

Continued on next page

Table A.1: (Continued)

Study	**Type of benchmark**	**Applications or Suite/Benchmarks**	**Property**	**Metric**
		HPCC/STREAM	Memory	GB/s
		HPCC/b_{eff}	Memory	GB/s
		Bonnie	Disk	MB/s
		Benchmark developed by authors	Elasticity	Duration (s)
Phillips *et al.* [83]	Application	Gromacs, FFmpeg, Blender	Many	Total execution time (s)
	Synthetic	TORCH/Dhrystone	CPU	Total execution time (s)
		TORCH/Spectral	CPU	Total execution time (s)
		TORCH/Particle	CPU	Total execution time (s)
Salah *et al.* [9]	Micro	Simplex	CPU	Total execution time (ms)
		HPCC/STREAM	Memory	MB/s
		FIO	Disk	KB/s
Lenk *et al.* [84]	Micro	Phoronix/crafty,dcraw	CPU	Test duration (s), MFLOPS
Li *et al.* [8]	Application	TPC-W	Overall performance	Page generation time(s)
	Synthetic	Modified SPECjvm2008	CPU	Finishing time of a CPU-intensive task (s)

Continued on next page

Table A.1: (Continued)

Study	Type of benchmark	Applications or Suite/Benchmarks	Property	Metric
		Modified SPECjvm2008	CPU	Finishing time of a memory intensive task (s)
		Modified SPECjvm2008	Memory	Finishing time of a disk I/O intensive task (s)
		Modified SPECjvm2008	Disk	Finishing time of a CPU-intensive task (s)
	Micro	iperf	Network	TCP throughput (MB/s) in intra- and inter-datacenter network
		ping	Network	RTT (ms)
Mao et Humphrey [16]	Micro	Benchmark developed by authors	Elasticity	VM acquisition and release (s)

Appendix B

Cloud performance evaluation: details and extended results

Related issues to the performance evaluation

Issues faced during the development of this study:

- Not all cloud providers provide an API to manage the VMs. This fact obliged us to start and stop VMs via the web interface which prevents the exact measurement of the provisioning time.
- Some cloud providers (particularly the recently emerged) do not support the import of VM images.
- The image provided by one cloud provider had the root user account deactivated (for security reasons as expressed by the technical support). As *ceilo* was conceived for being used under the root account, some troubles were encountered during the installation and configuration of the benchmark tools.
- Acquisition of cloud resources is not automatic for all cloud providers. For instance, for some cloud providers, the creation of the account needed to be validated by human-intervention before the resources could be used. Sometimes the confirmation took more than one

working day.

- One cloud provider had a security policy that considered the benchmarks used in this study a risk for its cloud infrastructure. The VMs were immediately stopped and the account got blocked till further explanation was provided for the reasons behind the tests.
- In some cases, the online documentation is extensive and well-explained, in other cases, the documentation is insufficient to solve technical issues but the technical support assisted us in deploying the applications.

VM configurations

The table B.1 presents the evaluated VM configurations. All prices have been converted to US$ (1US$ = 1.23€).

Table B.1: VM configurations

Cloud provider	VM size	vCPU (number)	RAM (GB)	Disk (GB)	Price(US$)/hr
Arubacloud	s	1	2	10	0.0309
	m	2	4	20	0.0556
	l	4	8	40	0.1050
Amazon	xs	1	0.615	8	0.0200
	s	1	1.7	160	0.0650
	m	1	3.75	410	0.1300
	l	2	7.5	840	0.2600
	xl	4	15	1680	0.5200
Cloudsigma	xs	1	0.512	10	0.0524
	s	1	2	10	0.0807
	m	2	4	10	0.1526
	l	4	8	10	0.2339
	xl	8	16	10	0.5841
Joyent	xs	0.15	0.625	20	0.0200
	s	1	1.75	56	0.0560
	m	2	7.5	738	0.2400

Continued on next page

Table B.1: (Continued)

Cloud provider	VM size	vCPU (number)	RAM (GB)	Disk (GB)	Price(U$)/hr
	l	4	15	1467	0.4800
	xl	8	30	1683	0.9600
Lunacloud	xs	1	0.512	10	0.0191
	s	1	2	10	0.0469
	m	2	4	20	0.0939
	l	4	8	40	0.1870
	xl	8	16	80	0.3750
Profitbricks	s	1	2	10	0.0413
	m	2	4	20	0.0825
	l	4	8	40	0.1650
	xl	8	16	80	0.3300
Rackspace	xs	1	0.512	20	0.0330
	s	1	1	40	0.1210
	m	2	4	160	0.2430
	l	4	8	320	0.4870
	xl	8	30	1200	1.5240
WindowsAzure	xs	2-shared	0.768	20	0.0184
	s	1	1.75	70	0.0552
	m	2	3.5	135	0.1105
	l	4	7	285	0.2210
	xl	8	14	605	0.4410

Benchmark duration

The benchmark duration is important when measuring cloud performance, since the costs related with the evaluation are directly proportional to its duration. Regarding the benchmark duration (Figure B.1), half of the cloud providers (Arubacloud, Cloudsigma, Profitbricks and Rackspace) have a benchmark duration of under one hour for all VM sizes. For the others, the benchmark duration and VM size are inversely proportional. This proportional relationship is mainly due to three facts. First, a constant workload in the computation benchmarks (*7-Zip* and *C-Ray*) across all VM sizes is kept. Thus, the lower the number of processors in a VM, the longer the duration of the computation benchmarks. Second, the practice of processor-sharing by cloud providers increases the benchmark duration. For example, Amazon and Joyent share the processor time between VMs for *xs*-VMs. Third, the differences in processor brands and qualities make some cloud providers more powerful than others in computing terms (Table B.2).

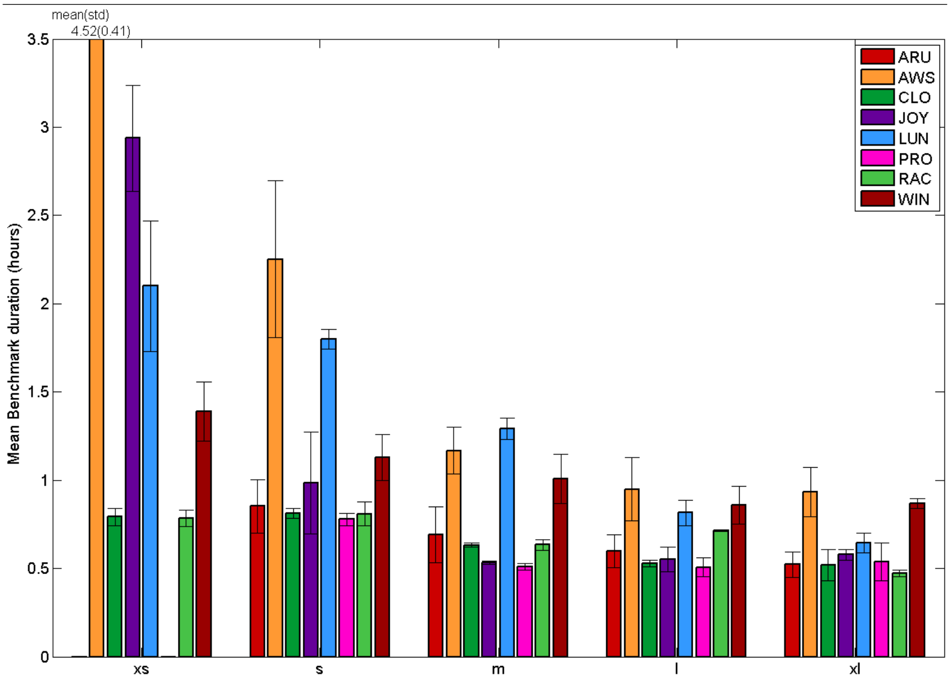

Figure B.1: Benchmark duration

Performance-price correlation with a simple figure of merit of Cloud performance

The results presented here were calculated with the simple figure of merit for cloud performance method (c.f. Section 2).

Provider	Processor
AWS	Intel Xeon E5-2650 @ 1.80GHz
CLO	AMD Opteron 6380 @ 2.50GHz
JOY	Intel Xeon E5645 @ 2.40GHz
LUN	Intel Xeon E5-2620 @ 1.50GHz
RAC	AMD Opteron 4332 HE @ 3.00GHz
WIN	AMD Opteron 4171 HE @ 2.09GHz

Table B.2: Type of processor for *xs*-VM size

Correlation among VM sizes from different cloud providers

The performance-price relationship for the same VM size from different cloud providers has been studied. The performance values have been calculated as previously described for each VM size (*i.e.* for every graph presented in Figure B.2). The Average Values (AVE) have been used to split each graph into four quadrants: High Performance (HP) and Low Cost (LC), High Performance (HP) and High Cost (HC), Low Performance (LP) and Low Cost (LC), and, Low Performance (LP) and High Cost (HC). The highlights of the findings are as follows:

- Arubacloud presents the best performance-price relationship among the evaluated cloud providers for the three VM sizes (*s*,*m* and *l*) evaluated with a low variability in the case of *s* and *m* sizes.
- AWS is placed on the HC-LP and LC-LP quadrants for all the VMs sizes but for the *m* VM size. AWS VMs present a low variability (0-10%) for the *m* and *xl* sizes.
- Cloudsigma presents a HP and a small variability at a HC for the small sizes (*xs* and *s*). For the *m*, *l* and *xl* sizes the performance is close to the AVE. VMs have a low variability for all the sizes but the *m* size.
- Joyent has a HP at a HC for VMs sizes *m*, *l* and *xl* sizes but not the *xs* size. VMs have a low variability for all the sizes but the *xl* size.
- Lunacloud VMs are on the HP-LC and LP-LC quadrants. VMs have a performance over the AVE for the *l* and *xl* sizes, with a low variability for the *m* and *l* sizes.
- Profitbricks VMs are on the HP-LC quadrant. VMs have a low variability for the *s*, *m* and *l* sizes.
- Rackspace presents a low variability and is placed on the HP-HC and

HP-LC quadrants for all the VMs sizes.

- WindowsAzure VMs are on the HP-LC and LP-LC quadrants. Performance results are 1 point over the AVE values for xs and xl sizes and 1 point under for s, l and m sizes. In terms of cost, WindowsAzure is under the AVE for all VMs sizes. Low variability is presented in m, l and xl VMs.

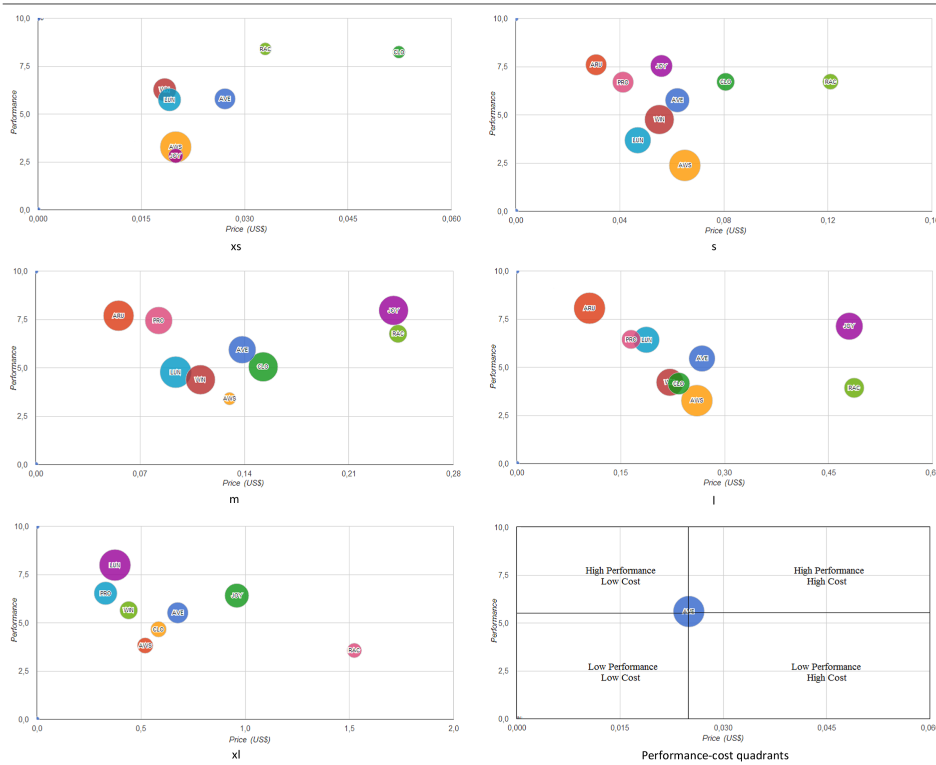

Figure B.2: Correlation between performance and price for VM sizes. The variability of a VM is represented by the size of the spot. Lower bound equal to 1 and upper bound equal to 10.

Correlation among different VM sizes from a single cloud provider

The performance-price relationship for different VM sizes from the same cloud provider has been studied here. The performance values have been calculated as previously described. The motivation behind this is to check the correspondence among size, price and performance of VMs. In general, prices are proportional to the size and performance of the VMs (Figure B.3). The highlights of the findings are as follows:

Economic advantage

A VM-pair comparison for every cloud provider in order to find VMs with similar performance values has been made. Users may reduce costs by using cheaper VMs with equivalent performance. For each cloud provider, pairs of VMs ($\mathrm{VM}_x, \mathrm{VM}_y$) have been found as follows. For two VMs, VM_x and VM_y, where:

$$Size(\mathrm{VM}_x) < Size(\mathrm{VM}_y) \tag{B.1}$$

The pair ($\mathrm{VM}_x, \mathrm{VM}_y$) is selected if:

$$Performance(\mathrm{VM}_x) \geq Performance(\mathrm{VM}_y) \tag{B.2}$$

or if:

$$0 < Performance(\mathrm{VM}_y) - Performance(\mathrm{VM}_x) < 0.5 \tag{B.3}$$

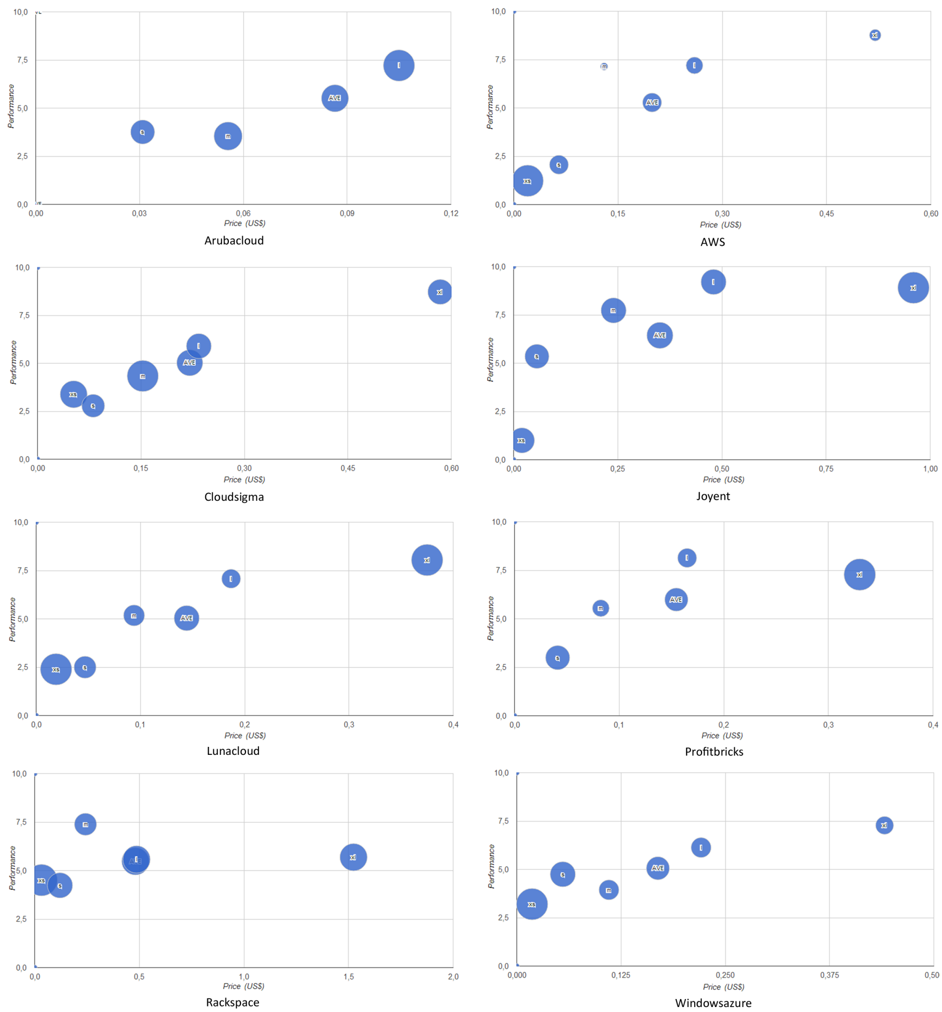

Figure B.3: Correlation between performance and price for cloud providers. The variability of a VM is represented by the size of the spot. Lower bound equal to 1 and upper bound equal to 10.

The Economic Advantage Economic Advantage (EA) has been defined as the amount of money a user saves when choosing the smallest VM between two VMs with similar performance. EA is denoted as follows:

$$\mathrm{EA} = \left[1 - \frac{Price(\mathrm{VM}_x)}{Price(\mathrm{VM}_y)}\right] \times 100\% \tag{B.4}$$

Table B.3 presents the pairs of VMs with their corresponding EA that satisfy equation (B.2) or (B.3).

Provider	**($\mathbf{VM}_x$, $\mathbf{VM}_y$)**	**EA**
ARU	(s,m)	50%
AWS	(m,l)	50%
CLO	(xs,s)	37.5%
JOY	(l,xl)	50%
LUN	(xs,s)	60%
PRO	(l,xl)	49.5%
RAC	(xs,s)	25%
	(m,l)	51%
	(l,xl)	67.8%
WIN	(s,m)	45.5%

Table B.3: VM-pair and EA

Bibliography

[1] "Compatibleone: The opensource cloud broker," http://www.systematic-paris-region.org/fr/projets/compatible-one, Accessed: June 2014.

[2] M. Cantara, "Gartner: Hype cycle for cloud services brokerage," July 2012.

[3] W. Fellows, "Cloud brokers: Now seeking ready-to-pay customers," January 2013.

[4] Q. He, S. Zhou, B. Kobler, and T. Duffy, D. ad McGlynn, "Performance analysis of cloud computing services for many-tasks scientific computing," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 22, no. 6, pp. 931–945, 2011.

[5] A. Li, X. Yang, S. Kandula, and M. Zhang, "Cloudcmp: comparing public cloud providers," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, ser. IMC '10. New York, NY, USA: ACM, 2010, pp. 1–14. [Online]. Available: http://doi.acm.org/10.1145/1879141.1879143

[6] Z. Li, H. Zhang, L. O'brien, R. Cai, and S. Flint, "On evaluating commercial cloud services: A systematic review," *J. Syst. Softw.*, vol. 86, no. 9, pp. 2371–2393, Sep. 2013. [Online]. Available: http://dx.doi.org/10.1016/j.jss.2013.04.021

[7] A. Iosup, R. Prodan, and D. Epema, "Iaas cloud benchmarking: Approaches, challenges, and experience," in *proceedings of the ACM/IEEE Conference on High Performance Networking and Computing (SC), MTAGS.* IEEE/ACM, 2012, pp.

1–8. [Online]. Available: http://www.pds.ewi.tudelft.nl/~iosup/cloud-benchmarking12mtags.pdf

[8] Z. Li, L. OBrien, R. Cai, and H. Zhang, "Towards a taxonomy of performance evaluation of commercial cloud services," *2013 IEEE Sixth International Conference on Cloud Computing*, vol. 0, pp. 344–351, 2012.

[9] K. Salah, M. Al-Saba, M. Akhdhor, O. Shaaban, and M. I. Buhari, "Performance evaluation of popular cloud iaas providers," in *Internet Technology and Secured Transactions (ICITST), 2011 International Conference for*, 2011, pp. 345–349.

[10] A. Iosup, S. Ostermann, M. Yigitbasi, R. Prodan, T. Fahringer, and D. H. J. Epema, "Performance analysis of cloud computing services for many-tasks scientific computing," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 22, no. 6, pp. 931–945, 2011.

[11] R. Moreno-Vozmediano, R. S. Montero, and I. M. Llorente, "Multicloud deployment of computing clusters for loosely coupled mtc applications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 6, pp. 924–930, Jun. 2011. [Online]. Available: http://dx.doi.org/10.1109/TPDS.2010.186

[12] J. Schad, J. Dittrich, and J.-A. Quiané-Ruiz, "Runtime measurements in the cloud: Observing, analyzing, and reducing variance," *Proc. VLDB Endow.*, vol. 3, no. 1-2, pp. 460–471, Sep. 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1920841.1920902

[13] C. Baun and M. Kunze, "Performance measurement of a private cloud in the opencirrus™ testbed," in *Proceedings of the 2009 International Conference on Parallel Processing*, ser. Euro-Par'09. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 434–443. [Online]. Available: http://dl.acm.org/citation.cfm?id=1884795.1884845

[14] M. Alhamad, T. Dillon, C. Wu, and E. Chang, "Response time for cloud computing providers," in *Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services*, ser. iiWAS '10. New York, NY, USA: ACM, 2010, pp. 603–606. [Online]. Available: http://doi.acm.org/10.1145/1967486.1967579

[15] Z. Hill, J. Li, M. Mao, A. Ruiz-Alvarez, and M. Humphrey, "Early observations on the performance of windows azure," in *Proceedings of the 19th ACM International Symposium on*

High Performance Distributed Computing, ser. HPDC '10. New York, NY, USA: ACM, 2010, pp. 367–376. [Online]. Available: http://doi.acm.org/10.1145/1851476.1851532

[16] M. Mao and M. Humphrey, "A performance study on the vm startup time in the cloud," in *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, 2012, pp. 423–430.

[17] Z. Li, L. O'Brien, H. Zhang, and R. Cai, "Boosting metrics for cloud services evaluation – the last mile of using benchmark suites," in *2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*, 2013, pp. 381–388.

[18] S. K. Garg, S. Versteeg, and R. Buyya, "A framework for ranking of cloud computing services," *Future Gener. Comput. Syst.*, vol. 29, no. 4, pp. 1012–1023, Jun. 2013. [Online]. Available: http://dx.doi.org/10.1016/j.future.2012.06.006

[19] S. Islam, K. Lee, A. Fekete, and A. Liu, "How a consumer can measure elasticity for cloud platforms," in *Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering*, ser. ICPE '12. New York, NY, USA: ACM, 2012, pp. 85–96. [Online]. Available: http://doi.acm.org/10.1145/2188286.2188301

[20] L. M. Vaquero, L. Rodero-Merino, and R. Buyya, "Dynamically scaling applications in the cloud," *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 1, pp. 45–52, Jan. 2011. [Online]. Available: http://doi.acm.org/10.1145/1925861.1925869

[21] W. Wang, H. Chen, and X. Chen, "An availability-aware virtual machine placement approach for dynamic scaling of cloud applications," in *Ubiquitous Intelligence Computing and 9th International Conference on Autonomic Trusted Computing (UIC/ATC), 2012 9th International Conference on*, Sept 2012, pp. 509–516.

[22] W. Dawoud, I. Takouna, and C. Meinel, "Elastic vm for rapid and optimum virtualized resources' allocation," in *Systems and Virtualization Management (SVM), 2011 5th International DMTF Academic Alliance Workshop on*, Oct 2011, pp. 1–4.

[23] L. Yazdanov and C. Fetzer, "Vertical scaling for prioritized vms provisioning," in *Cloud and Green Computing (CGC), 2012 Second International Conference on*, Nov 2012, pp. 118–125.

[24] Z. Li, L. O'Brien, H. Zhang, and R. Cai, "On a catalogue of metrics

for evaluating commercial cloud services," in *Proceedings of the 2012 ACM/IEEE 13th International Conference on Grid Computing*, ser. GRID '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 164–173. [Online]. Available: http://dx.doi.org/10.1109/Grid.2012.15

[25] J. Tordsson, R. S. Montero, R. Moreno-Vozmediano, and I. M. Llorente, "Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers," *Future Generation Comp. Syst.*, vol. 28, no. 2, pp. 358–367, 2012.

[26] S. Chaisiri, B.-S. Lee, and D. Niyato, "Optimal virtual machine placement across multiple cloud providers," in *Services Computing Conference, 2009. APSCC 2009. IEEE Asia-Pacific*, 2009, pp. 103–110.

[27] R. Van den Bossche, K. Vanmechelen, and J. Broeckhove, "Cost-optimal scheduling in hybrid iaas clouds for deadline constrained workloads," in *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, 2010, pp. 228–235.

[28] J. Lucas Simarro, R. Moreno-Vozmediano, R. Montero, and I. Llorente, "Dynamic placement of virtual machines for cost optimization in multi-cloud environments," in *High Performance Computing and Simulation (HPCS), 2011 International Conference on*, 2011, pp. 1–7.

[29] J. Lucas-Simarro, R. Moreno-Vozmediano, R. Montero, and I. Llorente, "Scheduling strategies for optimal service deployment across multiple clouds," *Future Generation Computer Systems*, vol. in press, 2012.

[30] W. Li, J. Tordsson, and E. Elmroth, "Modeling for dynamic cloud scheduling via migration of virtual machines," in *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on*, 2011, pp. 163–171.

[31] S. Chaisiri, B.-S. Lee, and D. Niyato, "Optimization of resource provisioning cost in cloud computing," *Services Computing, IEEE Transactions on*, vol. 5, no. 2, pp. 164–177, 2012.

[32] N. Grozev and R. Buyya, "Inter-cloud architectures and application brokering: taxonomy and survey," *Software: Practice and Experience*, pp. n/a–n/a, 2012. [Online]. Available: http://dx.doi.org/10.1002/spe.2168

[33] B. Rimal, E. Choi, and I. Lumb, "A taxonomy and survey of cloud computing systems," in *INC, IMS and IDC, 2009. NCM '09. Fifth*

International Joint Conference on, 2009, pp. 44–51.

[34] G. Carella, T. Magedanz, K. Campowsky, and F. Schreiner, "Network-aware cloud brokerage for telecommunication services," in *Cloud Networking (CLOUDNET), 2012 IEEE 1st International Conference on*, 2012, pp. 131–136.

[35] M. Frincu and C. Craciun, "Multi-objective meta-heuristics for scheduling applications with high availability requirements and cost constraints in multi-cloud environments," in *Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on*, 2011, pp. 267–274.

[36] S. Grivas, T. Kumar, and H. Wache, "Cloud broker: Bringing intelligence into the cloud," in *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, 2010, pp. 544–545.

[37] A. Quarati, A. Clematis, A. Galizia, and D. DâĂŹAgostino, "Hybrid clouds brokering: Business opportunities, qos and energy-saving issues," *Simulation Modelling Practice and Theory*, vol. 1, no. 0, pp. –, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1569190X13000075

[38] "Phoronix test suite: The leading software for automated, open-source testing & benchmarking," http://www.phoronix-test-suite.com/, Accessed: June 2014,.

[39] "xsltproc: command line xslt processor," http://xmlsoft.org/XSLT/xsltproc.html, Accessed: June 2014,.

[40] "7zip: application to compress files," http://mattmahoney.net/dc/text.html#1789, Accessed: June 2014,.

[41] "c-ray: benchmark for measuring floating point cpu performance," http://www.futuretech.blinkenlights.nl/c-ray.html, Accessed: June 2014,.

[42] "Stream: synthetic benchmark that measures sustainable memory bandwidth," http://www.cs.virginia.edu/stream/ref.html, Accessed: June 2014,.

[43] "Cachebench: benchmark to evaluate the performance of the memory hierarchy of computer systems," http://icl.cs.utk.edu/projects/llcbench/cachebench.html, Accessed: June 2014,.

[44] "Threaded i/o tester: benchmark to evaluate the performance of the hard disk drive and the file-system." http://sourceforge.net/projects/tiobench/, Accessed: June 2014,.

[45] "Iozone: filesystem benchmark tool," http://www.iozone.org/, Accessed: June 2014,.

[46] "Amazon elastic compute cloud: Micro instances (t1.micro)," http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts_micro_instances.html, Accessed: June 2014,.

[47] T. Saaty, *The Analytic Hierarchy Process, Planning, Piority Setting, Resource Allocation*. New york: McGraw-Hill, 1980.

[48] M. Sun, T. Zang, X. Xu, and R. Wang, "Consumer-centered cloud services selection using ahp," *Service Sciences, International Conference on*, vol. 0, pp. 1–6, 2013.

[49] E. Kostlan, "Statistical complexity of dominant eigenvector calculation," *Journal of Complexity*, vol. 7, no. 4, pp. 371 – 379, 1991. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0885064X9190025S

[50] T. Saaty, *Theory and Applications of the Analytic Network Process. Pittsburgh*. 4922 Ellsworth Avenue, Pittsburgh, PA 15213: RWS Publications, 2005.

[51] R. Buyya, D. Abramson, J. Giddy, and H. Stockinger, "Economic models for resource management and scheduling in grid computing," in *The Journal of Concurrency and Computation: Practice and Experience (CCPE)*. Wiley Press, 2002, pp. 1507–1542.

[52] ——, "Economic Models for Resource Management and Scheduling in Grid Computing," *Concurrency and Computation: Practice and Experience*, vol. 7, no. 13-15, pp. 1507–1542, 2002.

[53] C. Weinhardt, A. Anandasivam, B. Blau, and J. Stösser, "Business models in the service world," in *IEEE IT Pro*, vol. 11, 2009, pp. 28–33, iSSN: 1520-9202. [Online]. Available: http://dx.doi.org/10.1109/MITP.2009.21

[54] K. Lu, T. Roblitz, R. Yahyapour, E. Yaqub, and C. Kotsokalis, "QoS-aware SLA-based Advanced Reservation of Infrastructure as a Service," in *IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, Nov.-Dec. 2011, pp. 288–295.

[55] D. Niu, C. Feng, and B. Li, "Pricing Cloud Bandwidth Reservations Under Demand Uncertainty," in *ACM SIGMETRICS/PERFORMANCE joint international conference on Measurement and Modeling of Computer Systems*, 2012, pp. 151–162.

[56] ——, "A Theory of Cloud Bandwidth Pricing for Video-on-Demand Providers," in *IEEE INFOCOM*, Mar. 2012, pp. 711–719.

[57] I. San-Aniceto, R. Moreno-Vozmediano, R. Montero, and I. Llorente, "Cloud Capacity Reservation for Optimal Service Deployment," in *IARIA International Conference on Cloud Computing, GRIDs, and Virtualization*, Sep. 2011, pp. 52–59.

[58] W. Smith, I. Foster, and V. Taylor, "Scheduling with Advanced Reservations," in *International Parallel and Distributed Processing Symposium (IPDPS)*, 2000, pp. 127–132.

[59] H. Topcuoglu, S. Hariri, and M.-Y. Wu, "Performance-Effective and Low-Complexity Task Scheduling for Heterogeneous Computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 3, pp. 260–274, Mar. 2002.

[60] R. Aoun, E. Doumith, and M. Gagnaire, "Resource Provisioning for Enriched Services in Cloud Environment," in *IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, Nov.-Dec. 2010, pp. 296–303.

[61] R. Aoun and M. Gagnaire, "Towards a Fairer Benefit Distribution in Grid Environments," in *IEEE/ACS International Conference on Computer Systems and Applications (AICCSA)*, May 2009, pp. 21–26.

[62] S. Venugopal, X. Chu, and R. Buyya, "A Negotiation Mechanism for Advance Resource Reservations Using the Alternate Offers Protocol," in *International Workshop on Quality of Service (IWQoS)*, Jun. 2008, pp. 40–49.

[63] N. Kaushik, S. Figueira, and S. Chiappari, "Flexible Time-Windows for Advance Reservation Scheduling," in *IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, Sep. 2006, pp. 218–225.

[64] R. Aoun and M. Gagnaire, "An Exact Optimization Tool for Market-Oriented Grid Middleware," in *IEEE International Workshop on Communications Quality and Reliability (CQR)*, May 2009, pp. 1–4.

[65] ——, "Service Differentiation Based on Flexible Time Constraints in Market-Oriented Grids," in *IEEE Global Telecommunications Conference (GLOBECOM)*, Nov.-Dec. 2009, pp. 1–8.

[66] C. S. Yeo, S. Venugopal, X. Chu, and R. Buyya, "Autonomic Metered Pricing for a Utility Computing Service," *Future Generation Computer*

Systems, vol. 26, no. 8, pp. 1368–1380, Oct. 2010.

[67] M. A. Netto, K. Bubendorfer, and R. Buyya, "SLA-Based Advance Reservations with Flexible and Adaptive Time QoS Parameters," in *International Conference on Service-Oriented Computing (ICSOC)*, 2007, pp. 119–131.

[68] A. Sulistio, K. H. Kim, and R. Buyya, "Managing Cancellations and No-Shows of Reservations with Overbooking to Increase Resource Revenue," in *IEEE International Symposium on Cluster Computing and the Grid (CCGRID)*, May 2008, pp. 267–276.

[69] G. Birkenheuer and A. Brinkmann, "Reservation-Based Overbooking for HPC Clusters," in *IEEE International Conference on Cluster Computing (CLUSTER)*, Sep. 2011, pp. 537–541.

[70] K. Konstanteli, D. Kyriazis, T. Varvarigou, T. Cucinotta, and G. Anastasi, "Real-Time Guarantees in Flexible Advance Reservations," in *IEEE International Computer Software and Applications Conference (COMPSAC)*, vol. 2, Jul. 2009, pp. 67–72.

[71] T. Püschel and D. Neumann, "Management of Cloud Infrastructures: Policy-Based Revenue Optimization," in *International Conference on Information Systems (ICIS)*, Dec. 2009.

[72] T. Meinl, A. Anandasivam, and M. Tatsubori, "Enabling Cloud Service Reservation with Derivatives and Yield Management," in *IEEE Conference on Commerce and Enterprise Computing (CEC)*, Nov. 2010, pp. 150–155.

[73] S. Son and K. M. Sim, "A Price- and-Time-Slot-Negotiation Mechanism for Cloud Service Reservations," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 42, no. 3, pp. 713–728, Jun. 2012.

[74] D. Saure, A. Sheopuri, H. Qu, H. Jamjoom, and A. Zeevi, "Time-of-Use Pricing Policies for Offering Cloud Computing as a Service," in *IEEE International Conference on Service Operations and Logistics and Informatics (SOLI)*, Jul. 2010, pp. 300–305.

[75] R. Harms and M. Yamartino, "The economics of the cloud," Published Online http://www.microsoft.com/presspass/presskits/cloud/docs/The-Economics-of-the-Cloud.pdf, Microsoft Corporation, Redmond, WA, USA, Microsoft Whitepaper, November 2010. [Online]. Available: http://www.microsoft.com/presspass/presskits/cloud/docs/The-Economics-of-the-Cloud.pdf

[76] M. Yue, "A Simple Proof of the Inequality FFD$(L) \leq 11/9$ OPT(L)

+ 1, $\forall L$ for the FFD bin-packing algorithm," *Acta Mathematicae Applicatae Sinica*, vol. 7, no. 4, pp. 321–331, 1991. [Online]. Available: http://dx.doi.org/10.1007/BF02009683

[77] V. Stantchev, "Performance evaluation of cloud computing offerings," in *Advanced Engineering Computing and Applications in Sciences, 2009. ADVCOMP '09. Third International Conference on*, 2009, pp. 187–192.

[78] N. Yigitbasi, A. Iosup, D. Epema, and S. Ostermann, "C-meter: A framework for performance analysis of computing clouds," in *Cluster Computing and the Grid, 2009. CCGRID '09. 9th IEEE/ACM International Symposium on*, 2009, pp. 472–477.

[79] J. Dejun, G. Pierre, and C.-H. Chi, "Ec2 performance analysis for resource provisioning of service-oriented applications," in *Proceedings of the 2009 International Conference on Service-oriented Computing*, ser. ICSOC/ServiceWave'09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 197–207. [Online]. Available: http://dl.acm.org/citation.cfm?id=1926618.1926641

[80] Y. El-Khamra, H. Kim, S. Jha, and M. Parashar, "Exploring the performance fluctuations of hpc workloads on clouds," in *Proceedings of the 2010 IEEE Second International Conference on Cloud Computing Technology and Science*, ser. CLOUDCOM '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 383–387. [Online]. Available: http://dx.doi.org/10.1109/CloudCom.2010.84

[81] J.-S. Vöckler, G. Juve, E. Deelman, M. Rynge, and B. Berriman, "Experiences using cloud computing for a scientific workflow application," in *Proceedings of the 2Nd International Workshop on Scientific Cloud Computing*, ser. ScienceCloud '11. New York, NY, USA: ACM, 2011, pp. 15–24. [Online]. Available: http://doi.acm.org/10.1145/1996109.1996114

[82] A. Iosup, S. Ostermann, M. Yigitbasi, R. Prodan, T. Fahringer, and D. H. J. Epema, "Performance analysis of cloud computing services for many-tasks scientific computing," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 22, no. 6, pp. 931–945, 2011.

[83] S. Phillips, V. Engen, and J. Papay, "Snow white clouds and the seven dwarfs," in *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on*, 2011, pp. 738–745.

[84] A. Lenk, M. Menzel, J. Lipsky, S. Tai, and P. Offermann,

"What are you paying for? performance benchmarking for infrastructure-as-a-service offerings," in *Cloud Computing (CLOUD), 2011 IEEE International Conference on*, 2011, pp. 484–491.

List of Figures

List of Tables

Esta obra se editó en Ediciones USTA,
Departamento Editorial de la Universidad Santo Tomás.
Tipografía: LM Roman.
2016

Cloud brokering is a service paradigm that provides interoperability and portability of applications across multiple cloud providers. The attractiveness of cloud brokering relies on the new services and extended computing facilities that enhance or complement those already offered by isolated cloud providers. These services provide new value to Small and Medium-sized Businesses (SMBs) and large enterprises and make cloud providers more competitive. Nowadays, at the infrastructure level, cloud brokers act as an intermediary between the end-users and the cloud providers. A cloud broker provides a single point for service consumption in order to avoid vendor lock-in, increase application resilience, provide a unified billing, and simplify governance, procurement and settlement processes across multiple cloud providers. In the future, cloud brokers will provide advanced value-added services and will use attractive pricing models to capture potential cloud consumers. The aim of this book is to propose advanced value-added services and a pricing model for cloud brokers.

This book has three objectives:

- The first one is to present a single figure of merit of cloud VMs performance based on the application profile.
- The second one is to propose an exact approach for allocation of VMs across multiple cloud providers based on different optimization criteria.
- The third one is to describe a pricing model for cloud brokering, called pay-as-you-book.

UNIVERSIDAD SANTO TOMAS
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA
DIVISIÓN DE INGENIERÍAS