Available online at www.elixirpublishers.com (Elixir International Journal)

Marketing Management

Elixir Marketing Mgmt. 74 (2014) 27156-27159

ISSN: 2229-712X

Cloud brokering mechanisms

Fatemehnamdarpoorbengar and Mehdi Javanmard
Payemenoor University, Tehran, Iran.

ARTICLE INFO

Article history:
Received: 5 August 2014;
Received in revised form:
15 September 2014;
Accepted: 23 September 2014;

Keywords
Interface, Instance type,
Cloud brokers.

ABSTRACT

To help the user to cope with such a variety of interfaces, instance types, and pricing models, cloud brokers have emerged as a powerful tool to serve as intermediary between end users and cloud providers. A cloud broker can provide a uniform interface independently of the particular cloud provider technology, can collect automatically information from providers (instance availability, prices, etc.), and can help cloud users to choose the right platforms when deploying their services across multiple clouds, also allowing them to switch between platforms to get the best conditions.

Introduction

The current cloud market is composed of several public cloud providers These cloud providers and platforms exhibit many differences regarding the functionality and usability of exposed cloud interfaces, the methods for packaging and managing images, the types of instances offered, the level of customization allowed for these instances, the price and charging time periods for different instance types, the pricing models offered (e.g. on-demand, reserved, or spot prices), etc. The use of cloud computing technology has gained popularity in recent years, and many companies are currently moving their business to the cloud, by deploying their services and executing their workloads in private or public clouds according to the application requirements or the particular business models.[1]

Important analysts such as Gartner have predicted an exciting opportunity for the figure of the Cloud Broker (CB) According to Gartner, a broker is any service company who, acting as an intermediary between users and providers of Cloud services, offers its expertise in the evaluation of the proposals that are best suited to user needs and in the subsequent adoption or development of new products based on them. Another business opportunity for Cloud operators derives from an actual limitation of most Cloud infrastructure and concerns the poor offer of Quality of Service (QoS) supplied.[2]

The main opportunities that a cloud broker provides to cloud users are the following:

- Intermediation: building services atop an existing cloud platform, such as additional security or management capabilities.
- Aggregation: deploying customer services over multiple cloud platforms.
- Arbitrage: brokers supply flexibility, opportunistic choices, and foster competition between clouds. However, most current cloud brokers do not provide advanced service management capabilities to make automatic decisions, based on optimization algorithms, about how to select the optimal cloud to deploy a service, how to distribute optimally the different components of a service among different clouds, or even when to move a given service component from one cloud to another to satisfy some optimization criteria. So, an open research line in cloud brokering is the integration of different placement algorithms and policies in the broker for optimal deploying of virtual services among multiple clouds, based on different optimization criteria, for example cost optimization, performance optimization, energy efficiency, etc.[1]

Several research works in the field have studied how to take advantage of cloud brokering features under static conditions, e.g. when provider and user conditions do not change. These works reveal optimal deployments in several use cases, scheduling the virtual infrastructure once and deploying it in the best providers. But when the virtual infrastructure life-time is long enough, cloud provider conditions can change (e.g. prices), so it is necessary to analyse how to optimally reconfigure the service to adapt it to new situations. In dynamic scenarios, e.g. if a new cloud provider appears, an instance type is retreated/added from/to the cloud market, the user needs change, or prices change along the time-line, it is possible to obtain a better placement of the resources by reallocating the current infrastructure to some different clouds. For instance, pricing schemes can differ by vendor, or even prices can vary dynamically based on current demand and supply (e.g. Amazon EC2 spot prices). These differences provide users the chance to compare providers and reduce their virtual infrastructure investment.[1]

Cloud brokering mechanisms in multi provider

In the past few years, we have witnessed the proliferation of a heterogeneous ecosystem of cloud providers, each one with a different infrastructure offer and pricing policy. We explore this heterogeneity in a novel cloud brokering approach that optimizes placement of virtual infrastructures across multiple clouds and also abstracts the deployment and management of infrastructure components in these clouds. Experimental results confirm that multi-cloud deployment provides better performance and lower costs compared to the usage of a single cloud only. [3]

As the cloud computing market grows and the number of IaaSproviders increases, the market complexity is also increased as users have to deal with many different Virtual Machine (VM) types, pricing schemes, and cloud interfaces. In this context, the use of efficient cloud brokering mechanisms are essential to transform the heterogeneous cloud market into a commodity like service. These cloud brokers have a two folded role. First, they provide the scheduling mechanisms required to optimize placement of VMs amongst multiple clouds. Second, they offer a uniform management interface with operations, e.g., to deploy, monitor, and terminate VMs, with independence of the

Tele:
E-mail addresses: fnamdar1390@gmail.com

particular cloud provider technology.

The scheduling mechanisms required to optimize selection of virtual resources, which can be independent or belong to a multi-component service, amongst different clouds must take into account requirements such as configuration of individual resources, aggregated service performance, total cost, etc. In addition, the user can specify constraints regarding load balancing, service configuration, etc., e.g. to avoid a given set of resources to be allocated in the same cloud (or in different clouds). The cloud scheduler finds an allocation of virtual resources among the different cloud providers (a deployment plan) that optimizes the user criteria and adheres to the placement constraints. It is important to note that we are implicitly considering the possibility of a hybrid deployment, i.e., the resources can be placed in different clouds. This multi-cloud setup can be suitable for deployment of independent virtual resources or for loosely-coupled multi-component services with no or weak communication requirements. In the case of tightly-coupled services with strong communication requirements or latency sensitive ones, the service configuration constraints should be used to guarantee single-cloud deployment.

Notably, another important task of a cloud broker is to provide a uniform management interface to deploy, pause, resume, shutdown, monitor, etc. VMs in any cloud, with independence of the particular provider technology. There is, despite multiple on going efforts, currently no agreed-upon mechanism to interface with a cloud to perform these actions, but rather, each provider exposes its specific API. Thus, a cloud broker must, in order to interface multiple providers, use a software adapter layer to translate between generic management operations and provider-specific APIs.

In summary, our contributions are the following. We propose an architecture for cloud brokering and multi-cloud VM management. We also describe algorithms for optimized placement of applications in multi-cloud environments. Our placement model incorporates price and performance, as well as constraints in terms of hardware configuration, load balancing, etc. An evaluation against commercial clouds demonstrates that compared to single cloud deployment, our multi-cloud placement algorithms improve performance, lower costs, or provide a combination thereof.

In this paper we present a modular broker architecture that can work with different scheduling strategies for optimal deployment of virtual services across multiple clouds, based on different optimization criteria (e.g. cost optimization or performance optimization), different user constraints(e.g. budget, performance, instance types, placement, reallocation or load balancing constraints), and different environmental conditions (i.e., static vs. dynamic conditions, regarding instance prices, instance types, service workload, etc.).[2]

Cloud brokering architecture

Fig. 1 outlines the cloud brokering architecture used in this work and also illustrates the three roles in the herein studied cloud brokering scenario: the user, the cloud providers, and the cloud broker. A user of the cloud broker requests a virtual infrastructure using a service description template. This template consists of a set of virtual resources that may include compute, network and storage; an optimization criteria, e.g. the total infrastructure capacity, and a set of constraints, e.g. the maximum number of VMs of a certain type.

Each cloud provider offers several VM configurations, often referred to as instance types. An instance type is defined in terms of hardware metrics such as main memory, CPU (number of cores and clock frequency), available storage space, and price per hour.

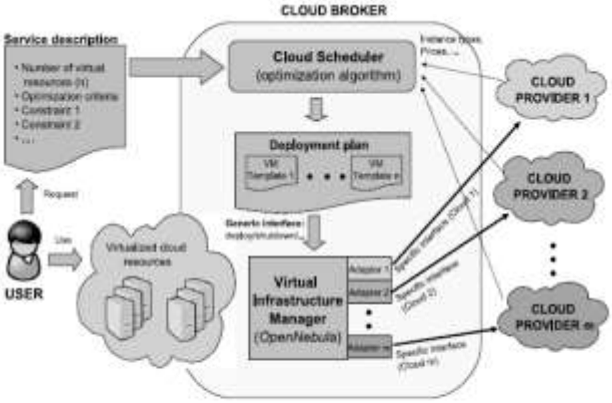


Fig. 1.Architecture overview that illustrates the roles in a cloud brokering scenario and outlines the operation of the cloud broker. [3]

From a cloud broker architecture perspective it does not matter whether prices are static or determined dynamically, i.e., on demand lookup of current prices upon each brokering request. In the evaluation against contemporary cloud providers described inSection4, flat hourly prices are used as this is the only model offered under comparable terms by all providers

The cloud broker performs two main actions: (i) the optimal placement of the virtual resources of a virtual infrastructure across a set of cloud providers, and (ii) management and monitoring of these virtual resources. Based on the infrastructure criteria and constraints provided by the user, the offerings of the available cloud providers, and the used scheduling algorithm, the scheduler component of the cloud broker generates an optimal deployment plan. The deployment plan embodies an explicit implementation of the user’s abstract infrastructure request and it contains a list of VM templates. Each template includes the target cloud provider to deploy the VM as well as attributes specific for the selected provider, e.g., the identifier of the VM image in that cloud.

One of the main components of the broker architecture is the cloud scheduler, which is responsible for making autonomously scheduling decisions based on dynamic pricing schemes, dynamic user demands, and different instance type performance. The scheduler can be configured to work with different scheduling policies based on different optimization criteria, such as service cost, service performance, etc. According to these policies, the scheduler performs an optimal deployment of the service components among different cloud providers trying to optimize a particular cost function.

The cloud scheduling problem can be addressed using either a static or a dynamic approach. The static approach is suitable for situations where the number of required virtual resources is constant (for example, a fixed-size service, a virtual classroom, etc.), and the cloud provider conditions (resource prices, resource availability, etc.) do not change throughout the service life-cycle. In this scenario, the resource selection can be done off-line, once only, and in advance to service deployment. Conversely, the dynamic approach is more suitable for variable size services (e.g., a web server cluster with a fluctuating resource requirements), or in the case of changing cloud provider conditions (variable prices, dynamic resource availability, etc.). In this case, the optimization algorithm should run periodically to adapt the resource selection to the variable service resource requirements and cloud conditions. As current cloud provider conditions are static (prices change very rarely, resource availability is unknown but assumed to be high, etc.)

the practical evaluation in this paper is focused on the static approach and a fixed-size service use case, whereas dynamic cloud scheduling mechanisms will be investigated in future work.

To handle the second task of the cloud broker, the Virtual Infrastructure Manager (VIM) provides an abstraction layer on top of the heterogeneous set of clouds, where each cloud has a different interface. This component is responsible for the deployment of each VM in the selected cloud as specified by the VM template, as well as for the management of the VM life-cycle. The VIM caters for user interaction with the virtual infrastructure by making the respective IP addresses of the infrastructure components available to the user once it has deployed all VMs. In this way, the user has a uniform view of the resources and is unaware of their distribution across the clouds. Furthermore, the resources can be accessed using standard applications, e.g., a secure shell or batch queue system. [3]

The current cloud market, constituted by many different public cloud providers, is highly fragmented in terms of interfaces, pricing schemes, virtual machine offers and value added features. In this context, a cloud broker can provide intermediation and aggregation capabilities to enable users to deploy their virtual infrastructures across multiple clouds. However, most current cloud brokers do not provide advanced service management capabilities to make automatic decisions, based on optimization algorithms, about how to select the optimal cloud to deploy a service, how to distribute optimally the different components of a service among different clouds, or even when to move a given service component from a cloud to another to satisfy some optimization criteria. [1]

The architecture is supported by a central database and has three main components: the Cloud manager, which collects information from cloud providers; the Scheduler, which reads the user description file, invokes the selected scheduling strategy, and makes the placement decision; and the VM manager, which performs the deployment action. The architecture has two main actors: the administrator and the user of the cloud broker. The former adjusts the broker configuration options (available clouds, instances types from each cloud, pricing information, etc.) before the execution's beginning; and the latter receives information from the broker and specifies a new service to deploy among available clouds, describing it through a service description file. A service is a set of components each one composed by a number of virtual machines, a scheduling strategy, an optimization criteria, and some particular restrictions. The service description file contains detailed information about the service to deploy by the broker, such as the components of the service, optimization criteria, scheduling policies to use, scheduling constraints or type of instances to use. For example: Component 1: web server front–end; Component 2: data-base servers; Component 3: application servers (back-ends); Component 4: file server, with a list of images (e.g. AMI in Amazon Fig. 1. Cloud brokering architecture overview.

EC2) associated with each component in each cloud to use, a list of post-configuration files for each service component (if necessary),and timing information (e.g. service start and end times).The architecture components' functionality is the following: the Cloud manager periodically collects information about instances availability and instances price for each instance in the database. It obtains this information from each particular cloud provider and acts as a pricing interface for users, updating the database when new information is available. This is specially useful in dynamic price case, in which it is necessary to have these prices updated. The Scheduler is responsible for making the placement decision.

This paper is oriented to the development of various scheduling strategies based on different criteria that will be integrated with this component. Its way of working is the following:

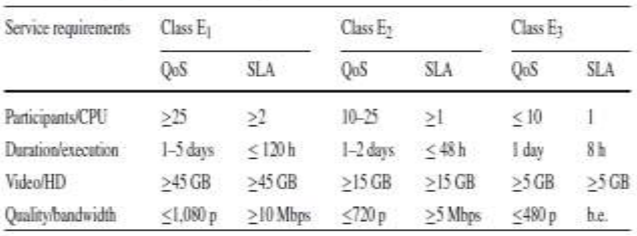
• It receives each new service from the database and reads its service description file to make an optimal placement decision.
• Before each decision, the scheduler obtains information about clouds, instances, prices, and others from the database, and invokes to the particular scheduling strategy specified in the service description and its features (static or dynamic scheduling, optimization criteria or restrictions).
• Then, it decides which set of VM has to be deployed in which cloud, and updates the database preparing it for a reading from the VM manager module.

The VM manager performs two main actions: the deployment of the virtual resources of a service across a set of cloud providers, and management and monitoring of these virtual resources. Both actions are addressed using a VM managing interface based on Deltacloud. Nowadays there are lot of cloud managers, such as Open Nebula among others, which interoperate perfectly with

Delta cloud. As an example, the VM manager periodically reads the database, uses the accounting information available to access to each cloud, and submits the VM in pending state or shutdowns the VM in cancelled state. It also monitors the deployed VMs collecting data about CPU, memory and network usage of each one, which is continuously updated in the database.[1]

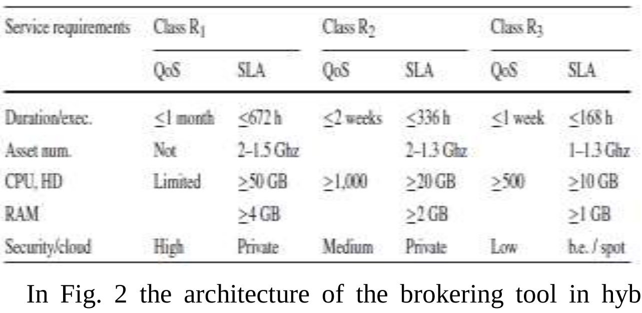
Hybrid clouds

Table 1 User requests classes of e-Learning services



Service requirements	Class E_1		Class E_2		Class E_3	
	QoS	SLA	QoS	SLA	QoS	SLA
Participants/CPU	≥25	≥2	10–25	≥1	≤ 10	1
Duration/execution	1–5 days	≤ 120 h	1–2 days	≤ 48 h	1 day	8 h
Video/HD	≥45 GB	≥45 GB	≥15 GB	≥15 GB	≥5 GB	≥5 GB
Quality/bandwidth	≤1,080 p	≥10 Mbps	≤720 p	≥5 Mbps	≤480 p	b.e.

Table 2 User requests classes of risk assessment services



Service requirements	Class R_1		Class R_2		Class R_3	
	QoS	SLA	QoS	SLA	QoS	SLA
Duration/exec.	≤1 month	≤672 h	≤2 weeks	≤336 h	≤1 week	≤168 h
Asset num.	Not	2–1.5 Ghz		2–1.3 Ghz		1–1.3 Ghz
CPU, HD	Limited	≥50 GB	≥1,000	≥20 GB	≥500	≥10 GB
RAM		≥4 GB		≥2 GB		≥1 GB
Security/cloud	High	Private	Medium	Private	Low	b.e. / spot

In Fig. 2 the architecture of the brokering tool in hybrid clouds is shown. The user interacts with the tool via a Web interface that specifies the service he/she needs and the desired level of QoS (1), as defined in Tables 1 and 2. These parameters are sent to the Broker's scheduler which translates them in the proper SLA specification, corresponding to one of the six pre-defined VM templates, and interacts with the private Cloud to obtain the resource status (2). This interaction is not possible, in general, with the public Clouds, which only provide the ability to launch pre-defined VM templates and monitor them. It is also to be considered that such information would also be not necessary, because usually public Clouds have enough resources to satisfy most of the requests. The broker scheduler then decides where to allocate the E and R3 requests on the basis of the policies described in Sect. 5 that consider the status information, the type of service, and the required QoS, and issues the launch command on the selected Cloud using the proper API . [4]

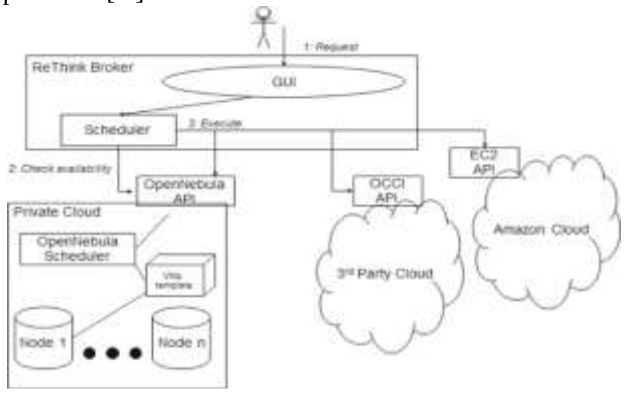


Fig 2. The architecture of the brokering tool and its interaction with users in hybrid clouds [4 مقاله]

Related works

We focus this section in two ways: the efforts made in the development of cloud brokers; and current research works in the field of cloud brokering which exhibit the improvement potential obtained when using cloud brokering middleware. Some private companies offer brokering solutions in the current cloud market, such as Right Scaleor Spot Cloud among others. For instance: Right Scale offers a private cloud middleware that provides a cloud management platform for control, administration, and life-cycle support of cloud deployments across multiple clouds. It includes an adaptable automation engine that automatically adapts the deployment to certain events in a pre-established way. In addition, it includes a multi-cloud engine that interacts with cloud infrastructure APIs and manages the unique requirements of each cloud. Customers can select, migrate and monitor clouds of their choosing from a single management environment. RightScale supports clouds from Amazon Web Services, Eucalyptus Systems, Go Grid, and VMware.

Another example is Spot Cloud, which provides a structured cloud capacity marketplace where service providers sell the extra capacity they have and the buyers can take advantage of cheaprates selecting the best service provider at each moment. The broker we propose also provides this feature but in an automatized way, without checking manually the prices of each cloud provider at each moment. Thus, optimization algorithms can be used to select the best way to place the VM according to the actual rates of all the cloud service providers.

On the other hand, there are also open source brokering middleware available in the market, such as Aeolus [20], an open source, Ruby-written cloud management software sponsored by Red Hat which runs on Linux systems. As a management software, Aeolus allows users to choose between private, public or hybrid clouds, using Delta Cloud cross-cloud abstraction library for making it possible. It has four different components: Conductor, which provides cloud resources to users, manage users' access to and use of those resources, and control users' instances in clouds. This lets users make intelligent choices about which cloud to use; Composer, which allows users to build cloud-specific images from generic templates, so that they can choose clouds freely using compatible images; Orchestrator, which provides a way to manage clumps of instances in an organized way. [1]

Hybrid Clouds couple the scalability offered by public Clouds with the greater control supplied by private ones. A (hybrid) Cloud broker acting as an intermediary between users and providers of public Cloud services, may support customers in the selection of the most suitable offers, optionally adding the provisioning of dedicated services with higher levels of quality. [2]

Conclusion

To help the user to cope with such a variety of interfaces, instance types, and pricing models, cloud brokers have emerged as a powerful tool to serve as intermediary between end users and cloud providers. A cloud broker can provide a uniform interface independently of the particular cloud provider technology, can collect automatically information from providers (instance availability, prices, etc.), and can help cloud users to choose the right platforms when deploying their services across multiple clouds, also allowing them to switch between platforms to get the best conditions. [1]

Reference

[1] Jose Luis Lucas-Simarro∗, Rafael Moreno-Vozmediano, Ruben S. Montero, Ignacio M. Llorente. Scheduling strategies for optimal service deployment across multiple clouds. Future Generation Computer Systems (2012)

[2] Alfonso Quarati, Andrea Clematis, AntonellaGalizia, Daniele D'Agostino. Hybrid Clouds brokering: Business opportunities, QoS and energy-saving issues. Simulation Modelling Practice and Theory (2013)

[3] Johan Tordsson , Rubén S. Montero, Rafael Moreno-Vozmediano, Ignacio M. Llorente. Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers. Future Generation Computer Systems 28 (2012) 358–367

[4] Daniele DAgostino · Antonella Galizia · Andrea Clematis · Matteo Mangini · Ivan Porro · Alfonso Quarati.AQoS-aware broker for hybrid clouds. Computing DOI 10.1007/s00607-012-0254-4. © Springer-Verlag Wien 2012