Chapter 9
Brokering Cloud Computing:
Pricing Models and Simulation Approaches

Georgia Dede
Harokopio University, Greece

George Hatzithanasis
Harokopio University, Greece

Thomas Kamalakis
Harokopio University, Greece

Christos Michalakelis
Harokopio University, Greece

ABSTRACT

Cloud computing is a rapidly evolving computational model, which has succeeded in transforming the ICT industry and the economy's production techniques by making corresponding services even more accessible to businesses, offering cost-effective solutions. The cloud broker is a new business model, derived from the necessity of finding the best provider, or the best bundle for the end user. It is a third-party business that assists clients in making the best decision in choosing the most suitable cloud provider and the most effective service bundle for their needs, in terms of performance and price. This chapter analyzes the cloud broker business model and highlights the broker's vital role and the benefits that arise from the use of its services. In that context, it describes cloud brokering and a market analysis, together with the most popular pricing models, together with a comparison among them, concluding with future directions for the expansion of the brokerage model.

DOI: 10.4018/978-1-5225-6114-9.ch009

INTRODUCTION

The contemporary transformation of the ICT industry has been achieved, to a large extend, by the introduction of cloud computing. The combination of hardware and software offers clients a more accessible and easy experience. Organizations and companies can use a more flexible strategy concerning the pricing, the "pay as you go" pricing model without any additional costs (Marston et al., 2011). Cloud computing offers a cornucopia of features that can easily adapt in any business needs. Sometimes choosing the best bundle of cloud computing services can be a very demanding task, because knowledge and experience is needed. Enterprises willing to migrate their infrastructure to the cloud are mainly concerned about the services they need, rather than who provides them (Rajkumar et al., 2009). Despite the innovative and profitable veneer cloud computing has, it also incorporates difficulties and challenges. As a consequence, the necessity of cloud brokerage was realized and the business model of cloud broker was developed. In order to provide significant assistance to the cloud computing market the cloud brokering model has emerged. The cloud broker acts as intermediary between the clients and the providers and creates a bundle of cloud services the match the need of the client.

The aim of this work is to underline the significance of cloud brokering. Cloud brokering is also a major factor that affects the rate of cloud diffusion. Due to the broker's beneficial influence, users can harvest the benefits cloud computing offers easier and with greater confidence (Buyya et al., 2009). The development of a business model has as a final purpose not only to enhance productivity and efficiency, but also το increase the profit for all stakeholders (client, providers and broker). This paper describes some brokering algorithms that can be used. Section 2 provides a definition of cloud computing (characteristics, architecture, deployment models) while Section 3 describes the cloud brokering, its' categories and a small market analysis. The brokerage pricing models are presented in Section 4 with a comparison between them and Section 5 concludes, providing directions for future research.

CLOUD COMPUTING

According to (Mell & Grance, 2011) cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. In an easier and more comprehensive way we could define cloud as a very potent mean of optimization for business processes and minimization of costs.

At this time of its expansion cloud computing has become pretty common technology among businesses as well as government services. A short introduction to the cloud is included, for the sake of completeness. Briefly, according to (Hassan, 2011) these attributes characterize cloud computing:

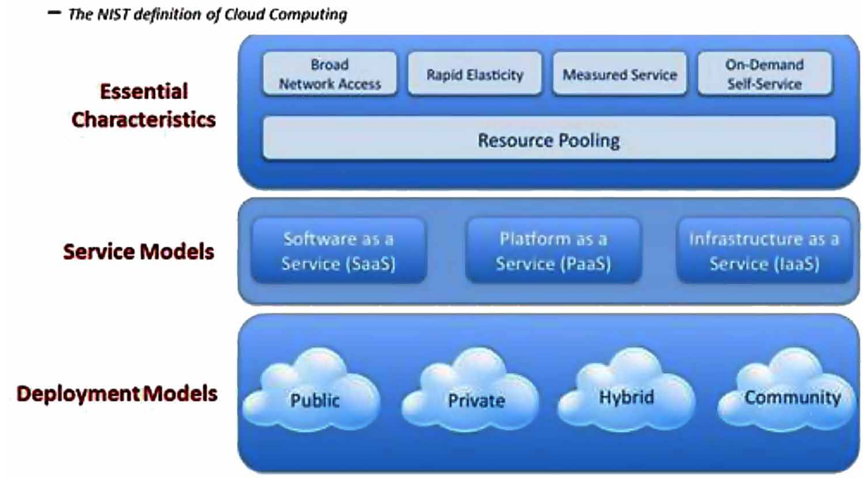
- **On-Demand Computing Model:** Organizations are able to escape from complex and expensive in-house infrastructure and choose the amount of resources they require for their operation.
- **Autonomous:** Clients are separated from the technical details of the cloud services they use.
- **Predefined Quality of Service:** Cloud providers state QoS terms in their service level agreements to inform clients about expected level of service.
- **Internet-Based:** All cloud services are hosted beyond organizations and delivered over the Internet.
- **Easy-to-Use:** Cloud providers offer easy-to-use interfaces that enable clients to make use of their services.
- **Scalable:** Clients are not limited with fixed amounts of resources. They can scale up and down at free will.
- **Inexpensive:** Cloud computing offers small-and-medium-sized enterprises (SMEs) a significantly lower-cost option than building an in-house infrastructure.
- **Subscription-Based Model:** Clients subscribe to services they are interested in, and they are charged accordingly.

Cloud Computing Services and Architecture

The architecture of cloud computing starts from IaaS as a foundation and on top, SaaS (Varia, 2010). The main logic behind the hierarchy, is that on the road to the top, the user is not required to know any detail about how things work in the cloud and everything is concealed like in a Black Box. The architecture of cloud computing is graphically illustrated in Figure 1

- **Infrastructure-as-a-Service (IaaS):** IaaS provides capability to provision hardware such as CPUs, memory, storage, networks, and load-balancers. The client does not have any control of the components but can manage over operating systems, deployed applications, networking and security. According to (Marinescu, 2012) services offered by this delivery model include: server hosting, web servers, storage, computing hardware, operating systems, virtual instances, load balancing, Internet access, and bandwidth provisioning. The next architectures are based on IaaS in order to work.

Figure 1. Cloud computing architecture



- **Platform-as-a-Service (PaaS):** Supplies users with development and administration platforms that provide on-demand access to available hardware resources. Many PaaS platforms are available to enable access to IaaS resources. PaaS is not particularly useful for portable applications, or when proprietary programming languages are used, or when the underlying hardware and software must be customized to improve the performance of the application. (Marinescu, 2012)
- **Data-as-a-Service (DaaS):** Frees organizations from buying high-cost database engines and mass storage. This service offers database capabilities for storing client information. DaaS is useful for big data analytics from enterprises and research.
- **Software-as-a-Service (SaaS):** The ultimate form of cloud resources that delivers software applications to clients in terms of accessible services. With SaaS, clients subscribe to applications offered by providers rather than building or buying them. The applications are accessible from various client devices through a thin client interface such as a web browser. If the first pillar of this paper is the cloud computing, the second is the enterprises. The cloud services offer great amount of options, so every organization can enjoy the aspects of cloud it needs.

Deployment Models

Cloud computing has different deployment models. The need for different cloud approaches comes from the need for variation between pricing, use and security demands of each separate company or individual client. (Mell & Grance, 2011)

- **Private Cloud:** The cloud infrastructure is used by a single organization. Usually private cloud is owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises.
- **Community Cloud:** A community cloud usually serves multiple organization that have same vision, exchange data and use same resources, or just need a very specific security policy. It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises.
- **Public Cloud:** The cloud infrastructure is provisioned for open use by the general public. It is hosted maintained operated and located on the cloud provider's premises.
- **Hybrid Cloud:** The cloud infrastructure is a composition of two or more distinct cloud infrastructures that have been described above.

CLOUD BROKERAGE

The growing number of Cloud computing services increases the interest of consumers in comparing these services in order to choose those best adapted to their needs (Felipe, Sanchez, Felipe, Sanchez, & Diaz-sanchez, 2016). According to Gartner a cloud service broker is company or other entity that adds value to one or more (public or private) cloud services on behalf of one or more consumers of that service via three primary roles including aggregation, integration and customization brokerage. A CSB enabler provides technology to implement CSB, and a CSB provider offers combined technology, people and methodologies to implement and manage CSB-related projects. Because of the role cloud computing plays in the technology market the needs for brokering is essential as the client and the provider must communicate in a lingua franca. The broker understands the needs of the client and tries to find the best suited cloud bundle for him.

Categories of Cloud Brokering

According to (Khanna & Jain, 2015) the categorization of the brokering services splits into three:

- **Service Intermediation:** An intermediation broker provides a service that directly enhances a given service delivered to one or more service consumers, essentially adding value on top of a given service to enhance some specific capability. CSBs will offer intermediation for multiple services. Intermediation brokers also supervise pricing and billing.
- **Service Aggregation:** An aggregation brokerage service combines multiple services into one. It will ensure that all data is modeled across all components and integrated as well as ensuring the movement and security of data between the service consumer and multiple providers. Aggregation brokers usually are cloud service providers. In aggregation-style brokerages, the services brokered are bundled and do not change frequently.
- **Service Arbitrage:** Cloud service arbitrage is similar to cloud service aggregation but with one difference. The services that are being aggregated are not standard. Indeed the goal of arbitrage is to provide flexibility and opportunistic choices for the service aggregator, providing multiple e-mail services through one service provider or providing a credit-scoring service that checks multiple scoring agencies and selects the best score.

Cloud Brokering Market

"Markets and Markets Research Private Limited" predicts that the Cloud Service Brokerage market size is expected to grow from USD 4.50 Billion in 2016 to USD 9.52 Billion by 2021 and an annual growth rate up to 16.2% during the forecast period. The major drivers of this market include the proliferation of hybrid & multi-cloud environments and the enterprise need of achieving cost savings.

"Global Industry Analysts, Inc" stated that new market reports on Cloud Services Brokerage, Asia-Pacific represents the largest market worldwide, supported by the region's growing clout as the IT outsourcing hub worldwide, the emergence of cloud ready Asian countries and the growing adoption of cloud IT architecture in the enterprise sector. The United States is projected to grow at the fastest CAGR of 33.8% over the analysis period, led by the well-developed cloud ecosystem, strong early adopters' trust in the cloud, and robust sales of cloud brokerage enablement solutions as a result of the growing focus on internally handling cloud brokerage functions. Finally, Cloud Brokerage Services represents the largest market segment accounting for a majority share in total revenue, while Cloud Brokerage Enablement

Solutions represent the fastest growing market segment with revenue growing at a CAGR of 33.6% over the analysis period.

BROKERAGE PRICING MODELS

In this section, we briefly review some of the prevalent cloud brokering pricing schemes.

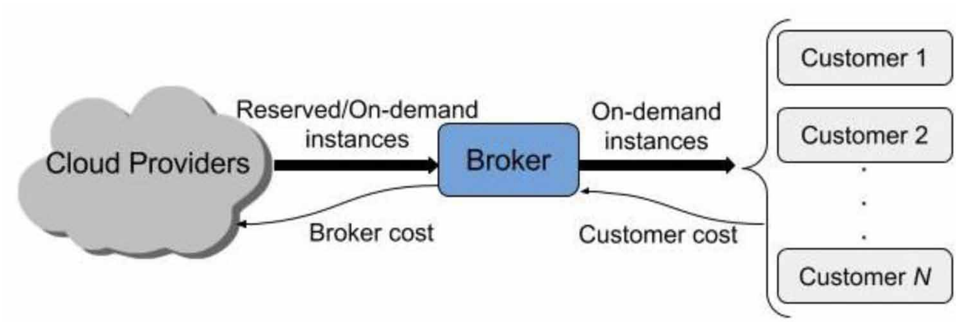
Description of Methodologies

Dynamic Cloud Resource Reservation

According to the Dynamic Cloud Resource Reservation (DCRR) model the cloud brokerage service reserves a great amount of resource instances (RI) - Virtual Machines (VMs) from different cloud providers and the customers take advantage of price discounts (Wang et al., 2013) The broker exploits the financial benefits of long-term instances reservations as well as the multiplexing gains. Therefore, instead of directly buying the instances from cloud providers, the customer will buy them from the cloud broker, which serves a great amount of customers demand providing on-demand instances. The DCRR model architecture is presented in Figure 2.

Figure 2 shows the architecture of the DCRR cloud brokerage service, where solid arrows show the direction of instance provisioning and dashed arrows show the direction of money flow. According to DCRR the customers purchase instances from the cloud broker in lower prices than those of cloud provider, since the broker leverages the wholesale price model and the price gap between the on-demand and the reserved instances to reduce costs for all users. Even more importantly, it can organize user requests to achieve additional cost savings with some benefits.

Figure 2. The DCRR cloud brokerage architecture



For example, better exploitation of reservation instances, which is achieved by concentrating service requirements from a large number of customers, thus limiting individual request bursts, which is considered more appropriate for reserved instances. Otherwise, users would often make individual requests, which is not economically advantageous for reserved instances. Wasted costs are also reduced due to the partial use of instances. When a customer uses a service, he is charged for the whole hour even if he uses the hour partially. On the contrary, the broker can take advantage of the single-hour charge and serves two users at the same time reducing the total cost of service in half. Finally, the broker takes the advantage of the fact that many providers offer significant discounts to customers who purchase a large number of instances. In this way, the broker reduces the costs of serving the users and increases its own revenues.

The key problem of the DCRR model for the broker is the decision on how many instances should buy in advance, how many it should launch on-demand and when it should reserve instances as the requests change dynamically over time. To meet this challenge, the problem of reserving resources is being addressed in the light of user requirements and the optimal solution comes via dynamic programming. The main types of cloud purchasing instances are on-demand instances and reserved instances.

For the case of on-demand instances, the users pay a fixed amount in each billing cycle without any commitment, while for the case of reserved instances they pay a subscription once to rent an instance / VM for a certain time period. The key issue is an optimization problem as it attempts to reduce the total cost of user requirements through the following equations (Wang et al., 2013):

$$\min_{\{r_1,\dots,r_T\}} \cos t = \sum\nolimits_{t=1}^{T} r_t \gamma + \sum\nolimits_{t=1}^{T} \left(d_t - n_t\right)^{+} p\,,\ n_t = \sum\nolimits_{i=t-r+1}^{t} r_i, \forall t = 1,\dots,T$$

where refers to the total cost of reservations and n_t is the cost for on-demand requests. The problem for the broker is to dynamically decides for the reservations r_1, ..., r_t in order to reduce the total cost, where r_t denotes the number of reserved instances at time $t \geq 0$. Each r_t is efficient from time t to $t + r$-1, where r is the instance reservation period, d_t denotes total instances and n_t the number of reserved instances that are efficient at time t = 1,2, .. T, where t is associated with each billing cycle. The symbol γ denotes the subscription fee for each reserved instance and p refers to the value of the current on-demand instance for each cycle. Finally, the term $(d_t - n_t)^+$ refers to the additional on-demand instances that need to be released.

As mentioned above, the goal of the broker is to minimize the total costs (according to) as it meets all customer requests. However, this equation is described by the Curse of Dimensionality (Bellman, 2013) as the results have exponential complexity as there are a large number of possible combinations and situations for the solutions. Such problems are solved using Approximate Dynamic Programming (ADP) (Powell, 2007).

The simulation process for the performance evaluation of the above pricing model implemented in (Wang et al. 2013) was based on Google cluster usage traces. The dataset contained 180GB over a month's resource demand/usage information of 933 users on a cluster of 12,583 physical machines. Based on the results, the broker brings an aggregate cost saving at 15%, for all user demands. The benefit of the broker is different in different user groups and more specifically 40% cost saving for users with medium demand fluctuation and almost 5% for users with low demand fluctuation. Considering the individual price discount of each user, almost 70% of them can save more than 30%, while the broker can bring more than 25% price discounts to 70% of aggregated users.

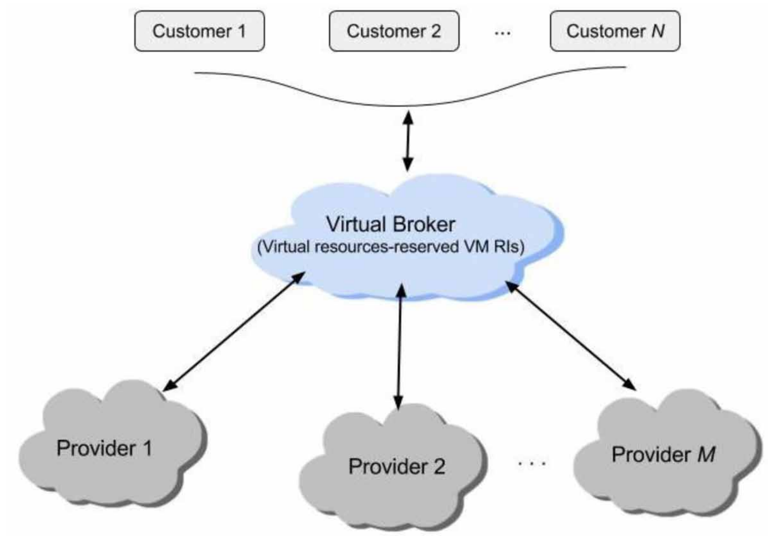
Heuristics for Virtual Machine Mapping

The model of Virtual Machine Mapping Problem (VMMP) seeks to anticipate the demand for the services so that the broker can buy them in sufficient quantity but also at the right time to serve customers gaining at the same time the maximum possible profit (Nesmachnow et al., 2017).. Therefore, all the requests of the customers for VMs should be mapped to the available reserved resource instances of the broker in order to maximize its profit. In contrast to traditional brokers, the virtual broker owns a number of instances, as shown in Figure 3, and provides a variety of services to its customers at lower prices than those offered by the cloud providers (Nesmachnow et al., 2015). The large gap of price between the on-demand and the reserved instances seems to be profitable for the broker.

The broker has to distribute all the available instances based on the requirements of the customers for VMs. In the absence of other available instances, the idle on-demand resources need to be closed in order to serve the customer and thus maintaining a high level of service. However, this will result in a reduction of the profit. This problem can be formulated by defining a mapping function *f*: *VM* → *RI* that maximizes the total profit of the broker, solved according to the following optimization problem (Nesmachnow et al., 2015):

$$\max \sum_{j=1}^{m} \left[\sum_{i:f(v_i)=r_j} \left(p\left(BF\left(v_i\right)\right) - C\left(r_j\right)\right) \times T\left(v_i\right) \right] + \sum_{h:ST(v_h)>D_j(v_h)} \left(p\left(BF\left(v_h\right)\right) - COD\left(BF\left(v_h\right)\right)\right) \times T\left(v_h\right)$$

Figure 3. The VMMP architecture



$BF(v_h)$ function considers every instance which meets or exceeds the requirements in hardware of the v_i request and selects the instance having the lowest cost on-demand. $ST(v_i)$ denotes the starting time of the v_h request based on the scheduling function f. VM={v_1, …v_n } is the set of customers' requests for VMs and $T(v_i)$ is the time length of v_i, which have to start before the deadline $D(v_i)$, defined from the user. RI={r_1, ..,r_m}, $m<<n$ is the set of resource instances that are reserved from the broker. C is the cost function for *RIs* and *COD* the cost function for on-demand instances with $C(r_j)<<COD(r_j)$. Both functions are related to charges per hour. Moreover, $p(r_j)$ is the pricing function that defines the price the broker charges the customer per hour for the instance r_j. The broker should charge an instance r_j with lower price than the on-demand pricing of provider in order to be attractive for the customers, meaning $p(r_j)<<COD(r_j)$. The hardware requirements for each VM are denoted as $P(v_i)$:processor speed, $M(v_i)$:memory, $S(v_i)$:storage, $n_c(v_i)$:number of cores and the requests arrive in groups (i.e. per hour) A_i. The aforementioned optimization problem is subject to $M(v_i) \leq M(r_j)$, $P(v_i) \leq P(r_j)$, $S(v_i) \leq S(r_j)$ and $n_c(v_i) \leq n_c(r_j)$.

To solve the above problem, alternative scheduling heuristic algorithms are presented below using different criteria for prioritizing customer requests (Nesmachnow et al., 2013).

- **Best Fit Resource (BFR):** Each request for VM is assigned to the *RI* that best fits (the same number of cores and the closest amount of memory for the requested value), defying the time limit. It seeks to leverage the assignment of VM to those RIs that most suit, letting the most restrictive requests to run in larger RIs.

- **Earliest Finish Time (EFT):** Priority is given to requests that are completed the soonest. The availability of each *RI* determines the time of completion and as a result the basic idea of the algorithm is the execution of requests that are finished the soonest, in order to increase the availability of the *RIs*.
- **Shortest Task First (STF):** Priority is given to VMs with the shortest execution time in order to minimize the completion time. The algorithm searches the soonest inactive request and it assigns it to the lower-cost VM instance.
- **Earliest Deadline First (EDF):** Priority is given to the requests with shorter deadline (without taking into account the arrival time) and each VM is assigned to the suitable RI with the earliest availability. The purpose is to take advantage of the prompt execution of the more restrictive requests to avoid penalization of buying on-demand instances due to deadline violations.
- **Cheapest Instance (CI):** The cheapest VM RI that allows the execution of each request is selected. The basic idea is to reduce the average waiting time of VM requests. An on-demand instance is rented only if there is absolutely no reserved instance satisfying the deadline of the request.
- **MaxProfit (MaxP):** Is an algorithm of greed for profit that uses the contribution of each request to the total profit function. The request with the largest contribution is assigned to the cheapest RI that meets the hardware requirements.
- **Shortest Request to Cheapest Instance (SRCI):** The requests are sorted according to their duration and the cheapest resource that allows the execution of each one is chosen. The purpose is to maximize the profit and minimize the response time. Once the shortest requests are asked to be executed first, they will be completed earlier and the users with demands of short computing time will find that their requests are completed fast enough.

The simulation for the performance evaluation of the aforementioned algorithms in (Nesmachnow et al., 2013) is based on C programming language, using the standard stdlib library, the GNU gcc, a Xeon E5430 processor at 2.66 GHz, 8GB RAM, and the CentOS Linux 5.2, from the Cluster FING. The problem instances are defined based on information about VM requests (memory, storage, processor speed, number of cores) and the relevant data for a set of RIs from the virtual broker (available memory, storage, processor speed, number of cores, cost, pricing values). A set of 400 problem instances were solved, considering batches of 50, 100, 200, and 400 VM requests with different durations, taking into account pre-booked cloud infrastructure of 10, 20, 30, and 50 RIs for the virtual broker. The VMs include small and average machines, large machines and instances with large memory, CPU and/or storage. The results revealed that the MaxP takes a precedence over the other approaches accounting for the profit of the virtual broker, and being among the best ones in terms of QoS of the solutions.

Derivative Contracts Options

This model describes the financial method of a cloud broker using derivative contracts to purchase cheaper instances for the users and facilitate providers to predict the future demand for services. Options contracts are common types of derivatives contracts which give buyers the legal right, but not an obligation, to purchase a resource for an agreed price on some later delivery date. Derivative contracts are used by the broker as a strategy to avoid the risk for uncertainty over future demand and supply (Rogers et al., 2012). This kind of financial methods is widely used in either natural of virtual commodity and product markets.

Typically, the role of the broker is to facilitate the assignment to the demand and offer in the market. Companies that offer this kind of business services can gain a profit by charging service subscriptions and /or leveraging spreads by buying resources at lower prices and selling them at higher. The main objective is to purchase in advance long-term contracts (36 months) for resources and repack them as contracts of one month offered to customers at a higher price.

The derivative contracts model includes the following steps (Clamp et al., 2013):

- Each month the broker asks from the customers to express their needs for future resources, by selling option contracts and determine how many reserved RIs will purchase from them providers.
- Then, the customer should decide whether or not to buy additional long reserved instances (usually for 3 years).
- The next month the customers submit their demands about the instances they need, activating their contracts. If there are available RIs from previous purchases, then the broker sells them with a profit gain. Otherwise, the broker has to pay additional on-demand instances to the provider in higher prices to fulfill the obligation to the customer.

The contract options model is based on the pricing model originally developed by Wu, Zhang and Huberman (Wu et al., 2008) at the HP. In this case the customers take a discount privilege if they are able to declare a true probability of using the VMs they need in the future. Every month, each customer *i* estimates his own probability p_i. He then submits his p_i to the broker to buy a contract for the resources he needs. The next month the customer is charged *Used*(p_i) if the contract is activated or otherwise *Unused*(p_i) according to the following equations (Clamp et al. 2013).

$$U_{sed}\left(p_i\right) = 1 + \frac{k}{2} - kp_i + \frac{kp_i^2}{2} \; U_{nused}\left(p_i\right) = \frac{kp_i^2}{2}$$

According to derivative contract options, the broker observes historical resource demand of customers for reserved instances, during the previous 3 years (36 months) $H = [h_{t-36}, ..., h_t]$, and compares against the future resource capacity, such as the number of reserved instances that the broker has currently available $F = \{f_t, .., f_{t+36}\}$ during the following 3 years. Then the deficit profile D is estimated for each forthcoming month, by subtracting historical demand from future expected demand $D=F-H$.

For each resource requested, a variable called Marginal Resource Utilization (MRU) is used to describe the possible utilization of an additional reserved resource over the next 3 years, based on historical demand and it is the proportion of item if $D > 0$. In addition, the broker uses another variable named threshold θ, which determines whether the broker should buy a new instance during the forthcoming 3 years. The broker has the ability to take a risk by varying the threshold value in the interval [0 1]. In case where MRU > θ, the broker is advised to purchase additional reserved instances, which will very probably be utilized in the following months and this decision is expected to be profitable. On the other hand if MRU ≤ θ, the broker should purchase new instances on-demand, being more profitable than purchasing reserved instances in advance.

Every month clients can demand instances from the broker by exercising their options contracts. If the broker has available capacity to satisfy the demand of the client, instances are sold to clients at a higher value than the purchased one. Otherwise, the broker has to buy on-demand instances and provide them to the client in order to fulfill its obligation.

The simulation for the performance evaluation of the derivative contracts model according to (Cartlidge et al., 2013) was based on Python programming language, taking into account a set of submitting probabilities of 1000 customers with different thresholds θ, for reserved instance contract lengths of 12 and 36 months. The results revealed that in the worst case scenario the broker still makes nontrivial profit. Modifications of the operating instances of the broker, such as purchasing longer term reserved instance contracts, seem to improve profits by 30%. Considering past performance can also reward the broker with increased profits, by 36%. Taking into account ideal market conditions, where longer term contract terms are used, considering an optimum threshold, profit was increased by up to 165%.

Quantized Billing Cycles

According to Quantized Billing Cycles (QBC) the user pays the same price for an on-demand instance, regardless if the time of usage is smaller than the whole billing cycle (Saha et al., 2015). For example, a user may pay the same price using the VM for 1 hour or for a few minutes. However, this type of pricing model is not proper for customers of sporadic demand. For example, sometimes user demands for VMs are higher than the available resources and hence brokers have to purchase more resources. Given the fact that the peak of demand is transient, the resources are charged for an hour, while their use will be only for a few minutes. Consequently, the main QBS problem is the occasional customer demand and the higher the sporadic nature, the greater the loss.

This problem is addressed with a dynamic pricing approach. Increasing the sales price leads to the decrease of demand and increase of revenues. For example, in case of static pricing where there is demand for a number of resources, the revenues will be determined by a fixed interest rate. When the selling price of a VM is rising (dynamic pricing) then the demand falls, but the revenues are more. Dynamic pricing brings more profit than static. This is because in the latter case many VMs remain idle and hence they don't contribute to revenues. The basic idea behind dynamic pricing is: it is preferable to suffer a small loss of revenue for a limited time rather than buying VM and then suffering greater loss in subsequent slots due to low demand.

The optimization problem of QBS is presented below, taking into account that the user pays the broker based on per-request basis.

$$\max_{\{\gamma_t, \upsilon_t\}} P = \sum_{t=1}^{T} \left(\gamma_t d_t - \upsilon_t\right)$$

where

$$\sum_{i=t-\tau+1}^{t} \upsilon_i \geq d_t, \forall t = 1, 2, \ldots, T$$

and

$$d_t = f\left(d_t^*, \gamma_t\right), \forall t = 1, 2, \ldots, T$$

P is the profit to be maximized. The term $\gamma_t d_t$-υ_t is the gain at t^{th} time slot and γ_t is the selling price per time slot. The term d_t is the number of VMs need to serve a new request, υ_t is the number of VMs bought in the t^{th} time slot, and d_t* the actual demand. The symbol τ refers to the period of one billing cycle and $\sum_{i=t-\tau+1}^{t} \upsilon_i$ is the number of active VMs within the t^{th} time slot. The term d_t* is the modified demand for VMs when the selling price is equal to γ_t. The relationship relating the actual demand d_t and the modified d_t* is based on the price-demand function *f*(*). If $\gamma_t = \gamma^*$, then $d_t = d_t^*$. The revenues of selling a VM at a price γ^* for a billing cycle is $\gamma^*\tau$ and if $\gamma^*\tau>1$ then it seems profitable for the broker, where 1 is the cost of a VM.

The above maximization problem is equivalent to the following minimization formula.

$$\min_{\{\gamma_t,\upsilon_t\}} L = \sum_{t=1}^{T}\left[\overbrace{\left(\gamma^* d_t^* - \gamma_t\, d_t\right)}^{\text{Demand Loss}} + \underbrace{\upsilon_t}_{\text{VM Loss}}\right]$$

In equation ($\gamma^* d_t^*$-$\gamma_t d_t$) and υ_t refers to the reduction in demand and VMs respectively. In case of unexpected increase in demand d_t* for a short time, then according to the selling prices will increase to reduce demand. In such a case the broker will suffer a small demand loss. On the other hand, the case of purchasing many VMs to support the demand hike is not the ideal solution as there is a possibility for the broker to suffer a huge VM loss in subsequent time intervals due to underutilized machines. Nevertheless, if demand is high for a long time it would be wiser to purchase the VMs. The algorithm based on belongs to the category of offline algorithms and hence it cannot be known in advance whether an increase in demand will last for a long time or not. The challenge here is to design algorithms that make decisions online based on present and past data. Such algorithms are called online.

The above algorithms described in and belong to the category of ski-rental problems. In this category the broker has to decide whether to buy or will continue to rent a resource without knowing in advance the time period of usage and future demand. If the period of usage is short then renting is preferable, while for long-term usage buying is cheaper. The ski-rental problems face the dilemma of whether to buy or rent the RIs without knowing beforehand the time period of usage. Breakeven point is used to design online algorithms, suggesting the point after which buying is better than renting.

The simulation for the performance evaluation of the proposed pricing model implemented in (Saha et al., 2015) was based on Google cluster usage traces, conducting comparative studies for the effect of demand prediction and demand threshold for switching between renting and buying. The results revealed the significant importance of demand prediction and determined the appropriate breakeven points for different thresholds.

Two-Sided Auction Mechanism

Two-sided auctions allow for many-to-many price negotiations, meaning that on the one hand there is a number of providers and on the other a number of buyers. Each side submits its bids on resources. This type of model is quite efficient in relation to unilateral auctions. However, further research is required to bridge the gap between the two sides so that both their benefits and requirements are rightly met.

The proposed model strives to meet the needs of the providers and customers as well as the broker in an attempt to attract them. This model seems to benefit all users by properly exploiting the transactions of large data services to facilitate their trade and use. This mechanism is called Two-Sided Mechanism for Trading Big Data Commodities (2-SAMBA) and determines the price each user has to pay to the broker for the resources he is going to use and the revenue the cloud provider will receive (Mashayekhy et al., 2014).

There is a set of $C = \{C_1, C_2, ..., C_m\}$ providers available to offer a large amount of resources to cloud users through a reservation system for a set of different time slots, defined as T. Each provider C_j offers a cluster to users for each time slot, reporting a minimal cost. In addition, the preferences of the providers are denoted by $a= (a_1, ..., a_m)$ where each element i denotes the minimum cost of a provider C_j. The cost of each C_j is a_j/T for each time slot. There is also a set of users U and each user i requests to use a cluster for a certain time slot and determines his bit preference as the maximum value he is willing to pay for the cluster at time t. Both users and providers report their requirements (bid and demand respectively) to the broker, who is responsible for executing and implementing the auction to determine user resource allocation and pricing. Both participants send this information to the broker a-priori, ensuring the privacy of the choices.

Based on the above, the problem of trading big data computing commodities (TBDCC) is to determine the distribution of clusters to users but also their price based on the submitted bid and demand. One mechanism for resolving this issue involves two phases:

- **Winner Determination:** The assignment of clusters to users over time is determined. If user i receives a cluster from provider C_j at time t, then the binary decision variable $x_{ij}^t = 1$, otherwise 0. Also, when the resources of a cloud provider C_j are allocated to a user then the binary decision variable $y_j = 1$.
- **Price Determination:** The amount of π_i^u that each user i has to pay to the broker and the amount π_j^c that each provider C_j receives from the broker. Users have almost linear utility. This is the case if the user i is granted with the resources, the utility u_i^u will be the difference between the valuation and the amount of money transferred, $u_i^u = b_t^i - \pi_u^i$ or 0 otherwise. If a provider C_j allocates a cluster to users, then its own utility will be $u_c^j = \pi_c^j - a_j$ and 0 otherwise.

The monetary payoff of the broker is defined as the total payment received by users minus the receipts of the providers, expressed by $\sum_{i \in U}^{T} \pi_i^u - \sum_{j:C_j \in C}^{T} \pi_j^c$.

When the monetary payoff is not negative, then the auction is ex-post budget balanced. This property gives the incentive to broker to set up the auction. Each participant attempts to maximize its utility. In order to promote the transactions and attract the interest of users and cloud providers this model seeks to optimize the utility for both users (customers and providers) and brokers' payoff. The phase of optimal winner determination is based on the following equation:

$$\text{Maximize} \sum_{t \in T} \sum_{i \in U} \sum_{j:C_j \in C} b_i^t x_{ij}^t - \sum_{j:C_j \in C}^{T} a_j y_j$$

where

$$\sum_{i \in U} x_{ij}^t \leq 1, \forall j : C_j \in C, \forall t \in T, \quad \sum_{j:C_j \in C} \sum_{t \in T} x_{ij}^t \leq 1, \forall i \in U$$

Moreover $y_j \geq x_{ij}^t$, $x_{ij}^t = \{0,1\}$ and $y_j = \{0,1\}$. The objective function is to maximize the social welfare.

The simulation for the performance evaluation of the 2-SAMBA according to (Mashayekhy et al., 2014) is based on IBM ILOG CPLEX Optimization Studio Multiplatform Multilingual e-Assembly, C++, AMD 2.4GHz Dual Proc Dual Core, 16 GB RAM of WSU Grid System. The whole procedure includes a comparative study of SAMBA and another TBDCC algorithm called VCG-TBDCC based on Amazon request data for a total number of 350 users. The results revealed that 2-SAMBA algorithm is very fast finding solutions in less than 75 seconds, being suitable in two-sided markets with high demand. In conclusion, it turns out that 2-SAMBA is suitable for trading big data computing commodities.

Comparison of Pricing Models

In this section a comparative study of the aforementioned pricing models is presented revealing the key issues that have to be considered, when implementing each algorithm for cloud brokerage.

Considering the DCRR and the VMMP, the users receive a lower price when trading with the broker. There is no need for upfront payment for reservations and no money wasted on idle reservation instances. The broker makes profit by leveraging the wholesale model. However, the main aspect the broker has to deal with is to what extent it makes proper predictions about future demand. On the one hand the broker faces the risk to buy a pool of VMs probably not to be used in the near future. On the other hand, an incorrect estimated of future demand can lead to a lack of resources and customer service. In this case the broker will be forced to buy more expensive instances on-demand directly from cloud providers. Consequently, the result in both cases is common: profit loss.

Regarding the derivative contracts approach, it seems more profitable for the broker to purchase long-term option contracts. Moreover, the past performance of the customers benefits the broker. The main practical problem of derivative contracts is due to the inherent risk related with the unsteady nature of the cloud market. Moreover, it is very risky to take into account future probabilities of customers on demand, if we consider the case that customers may reveal a mistaken possibility and hence the broker will inaccurately forecast the reservation of the resources. In this case the broker will purchase resources that will remain unused, which results in profit loss.

As far as the QBS model is concerned, dynamic pricing turns out to make more profit than static pricing, mainly due to the underutilization of the VMs in the latter approach. The key problem that the broker has to deal with is the demand prediction and the time duration for which it has to continue to rent the VMs and the right time that the broker has to decide if purchasing the resources will be more profitable than renting.

Finally, the two-sided auction mechanism seems to be profitable for big data applications area, where there is a need to develop market mechanisms for managing, trading, and pricing big data computing services. In this case users require entire clusters for their big data applications. Such demand necessitates the design and deployment of markets for big data services in which entire clusters are the tradable goods.

Based on the aforementioned analysis and the general needs of cloud market, the pricing models for cloud brokerage offer economic benefits to both customers and providers, while at the same time being profitable for the broker.

According to pricing models adopted by the broker, presented in this book chapter, the broker reserves instances from cloud providers based on past performance of customers, using either a probability which reveals the utilization of instances for the next month or an online reservation strategy to make decisions based on history. In addition, a broker may collect tariffs from the provider market and assesses them by calculating the cost performance of each tariff always according to the priorities of customers for resources. Dynamic pricing is also presented as approach for aiming to regulate the demand on resources based on the underutilization of the VMs or minimize the service cost of the broker using dynamic programming and approximate algorithms. Moreover, the auction mechanism seems to be suitable for big data applications. In this context, the development of flexible pricing procedures is an issue of high concern, since the existing ones seem to not adequately address the pricing of cloud services.

CONCLUSION AND FUTURE OUTLOOK

This chapter gives an overview of cloud computing and its services and models, while emphasizing the need for cloud brokerage and its benefit. The most promising cloud brokerage seems to play an essential role in the increasingly complex cloud computing scenarios and in profit making in the context of future cloud market.

In the market of cloud computing, a broker functions in the same way as it does in other, real-world, markets. It matches the demands of users with supplies of providers, aiming to succeed in settling the best financial agreement between these two sides of the market. The purpose of the broker to make profits for his own offering at the same time high quality services to customers reveals a successful commodity market.

The overview of the cloud broker discussed in this book chapter focuses on the numerous benefits of this widely known business model. From a business oriented perspective, the broker assists enterprises to develop themselves, makes cost savings, creating at the same time a competitive environment with more job opportunities and challenges. Cloud brokering has a substantial potential for cloud service providers and small, upstart entrepreneurs, who gain improved profitability and new revenue opportunity.

This book chapter also emphasizes the need of pricing models adopted by the broker, presenting a review of previous literature in this area, from a cost saving perspective for the broker. These cloud brokerage pricing models serve the needs of users by providing the resources they have leased earlier to different providers either dynamically or on-demand and determining pricing according to the existed supply and demand, depending on market conditions. These approaches are based on economic strategies and algorithms taking into account the cost of resources and service quality, having the ultimate goal of maximizing profit as well as minimizing possible spending and cost savings.

As the cloud broker business model is still developed, there are several important aspects to be further explored, mainly towards the direction of developing and adopting more efficient pricing methods and the role of the broker into the reduction of costs. An important issue of future research is the development of a meta-algorithm able at forecasting future demand for more efficient pooling of resources in order to better meet the expected demand to minimum possible costs. Moreover, the possibility for a broker to adopt more than one pricing models and switch among them according to demand and recourses conditions seems to be very promising for future brokerage services. Finally, research must be extended to accommodate the SaaS and PaaS models as well, which are also expected to diffuse quickly in the coming years, raising the imperative need for new, innovative, business models.

REFERENCES

Bellman, R. (2013). *Dynamic programming.* Courier Corporation.

Buyya, R., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, *25*(18), 599–616. doi:10.1016/j.future.2008.12.001

Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, *25*(6), 599–616. doi:10.1016/j.future.2008.12.001

Cartlidge, J., & Clamp, P. (2014). Correcting a financial brokerage model for cloud computing: Closing the window of opportunity for commercialisation. *Journal of Cloud Computing*, *3*(1), 2. doi:10.1186/2192-113X-3-2

Clamp, P., & Cartlidge, J. (2013). Pricing the cloud: an adaptive brokerage for cloud computing. In *5th International conference on advances in system simulation (SIMUL-2013)*. IARIA XPS Press.

Felipe, A., Sanchez, D., Felipe, A., Sanchez, D., & Diaz-sanchez, F. (2016). Cloud brokering: New value-added services and pricing models TELECOM ParisTech Spécialité "Informatique et Réseaux" Cloud brokering. *Nouveaux services de valeur ajoutée et politique de prix*.

Hassan, Q. F. (2011). Demystifying Cloud Security. *Crosstalk*, 6–21. Retrieved from http://www.crosstalkonline.org/storage/issue-archives/2011/201101/201101-Hassan.pdf

Iturriaga, S., Nesmachnow, S., & Dorronsoro, B. (2017). Optimizing the Profit and QoS of Virtual Brokers in the Cloud. In Cloud Computing (pp. 277-300). Springer International Publishing. doi:10.1007/978-3-319-54645-2_11

Marinescu, D. C. (2012). Cloud Computing. *Theory into Practice*, 1–403. PMID:22333270

Mashayekhy, L., Nejad, M. M., & Grosu, D. (2014, October). A two-sided market mechanism for trading big data computing commodities. In *Big Data (Big Data), 2014 IEEE International Conference on* (pp. 153-158). IEEE. 10.1109/BigData.2014.7004225

Mell, P., & Grance, T. (2011). The NIST definition of cloud computing. *NIST Special Publication*, *145*, 7. doi:10.1136/emj.2010.096966

Nesmachnow, S., Iturriaga, S., Dorronsoro, B., El-Ghazali, T., & Bouvry, P. (2013). List scheduling heuristics for virtual machine mapping in cloud systems. *VI Latin American Symposium on High Performance Computing (HPCLatam)*, 37-48.

Nesmachnow, S., Iturriaga, S., & Dorronsoro, B.IEEE. (2015). Efficient heuristics for profit optimization of virtual cloud brokers. *IEEE Computational Intelligence Magazine*, *10*(1), 33–43. doi:10.1109/MCI.2014.2369893

Powell, W. B. (2007). *Approximate Dynamic Programming: Solving the curses of dimensionality* (Vol. 703). Wiley-Interscience. doi:10.1002/9780470182963

Rogers, O., & Cliff, D. (2012). A financial brokerage model for cloud computing. Journal of Cloud Computing: Advances. *Systems and Applications*, *1*(1), 2.

Saha, G., & Pasumarthy, R. (2015, September). Maximizing profit of cloud brokers under quantized billing cycles: a dynamic pricing strategy based on ski-rental problem. In *Communication, Control, and Computing (Allerton), 2015 53rd Annual Allerton Conference on* (pp. 1000-1007). IEEE. 10.1109/ALLERTON.2015.7447117

Sean Marston, Z. L. (2011). Cloud computing — The business perspective. *Elsevier BV.*, *51*(1), 176–189.

Wang, W., Niu, D., Li, B., & Liang, B. (2013, July). Dynamic cloud resource reservation via cloud brokerage. *Distributed Computing Systems (ICDCS), 2013 IEEE 33rd International Conference on*, 400-409. 10.1109/ICDCS.2013.20

Wu, F., Zhang, L., & Huberman, B. A. (2008). Truth-telling reservations. *Algorithmica*, *52*(1), 65–79. doi:10.100700453-007-9107-5