CASE STUDY BASED ON REVENUE MAXIMIZATION MODEL USING CUSTOMIZEDPLANS IN COMPUTER SERVICE ALLOCATION

SHOWKAT AHMAD DAR
Department of Computer Science & Engineering
Islamic University of Science and Technology
J&K, India
showkatme2009@gmail.com

***Abstract*— The rapid development of the Internet and the emergence of computing technologies like grid or cloud computing have enabled a novel trend of purchasing and consuming Information Technology services. These services are offered at different prices using various pricing schemes and techniques. End users will favor the service provider offering the best quality with the lowest price. Therefore, applying a fair pricing model will attract more customers and achieve higher revenues for service providers. This work focuses on a novel static/dynamic pricing model which is able to satisfy advance users requirements based on normal fixed price model. This paper considers many factors that affect pricing and user satisfaction, such as fairness, QoS, SLA, and more, by highlighting their importance in recent markets and propose a flexible model which tries to utilize all resources to the highest capacity and offers low prices for underutilized resources. The simulated results show the appropriateness of new pricing model for sharing of computing resources, where providers want to have more customers as a managerial decision and even more income in total.**

Keywords — *Computer Resource; Pricing; Resource Allocation..*

I. INTRODUCTION

The rapid development of the Internet and the emergence of computing technologies like grid or cloud computing have enabled a novel trend of purchasing and consuming Information Technology (IT) services [1]. Network-based computing has brought new opportunities to the Information and Communication Technology (ICT) industry allowing businesses to outsource their IT facilities to providers and avoid expensive upfront investments of establishing their own infrastructure and consequent costs of maintenance and upgrades [2]. By means of computer services, customers can access all their required capabilities (i.e., computational re-sources, data, and applications) over the Internet and Intranets, use what they need, and pay for what they use without being concerned with the underlying infrastructure. As a result, customers experience the comfort of traditional utilities such as water, electricity, gas, and telephony. Advantages such as a utility model in addition to accessibility, scalability, and ease of management have created an industry-wide shift towards computing solutions like cloud computing.

According to a forecast from International Data Corporation (IDC), the worldwide spending on such services like public cloud services is expected to surpass $107 billion in 2017. Among different forms of delivering resources, IDC recognize such models as one of the fastest growing categories with compound annual growth rate of 27.2%. IaaS is a promising solution for enabling on-demand access to an elastic pool of configurable and virtual computational services (e.g., computing power, storage, and networks) in a pay-as-you-go manner. Service provider owns the data center(s) and all the required equipment and is responsible for hosting, running, and maintaining them. Providers offer computational services in different forms such as Virtual Machine (VM) instances with specific resource characteristics such as computing power, memory, and disk along with operating system type and installed applications. Amazon EC2, Windows Azure [3], Rackspace, Google Compute Engine [4] and GoGrid are examples of commercial cloud offerings. The same structure is used in other types of services like ISPs who provide Internet services.

This work focuses on the profit maximization problem of computer resource and service providers. We investigate different market and economics-inspired mechanisms such as resource management, future markets, revenue management, and mechanism design to address the providers' profit maximization problem as shown in Figure 1. Our proposed techniques are developed for two main different scenarios: 1) when the provider acts exclusively using their in- house resources to serve customers and 2) when it participates in a federation and benefits from outsourcing requests The remaining parts of this chapter detail the need for provider profit maximization and discuss the research problems, contributions, and organization of the thesis.

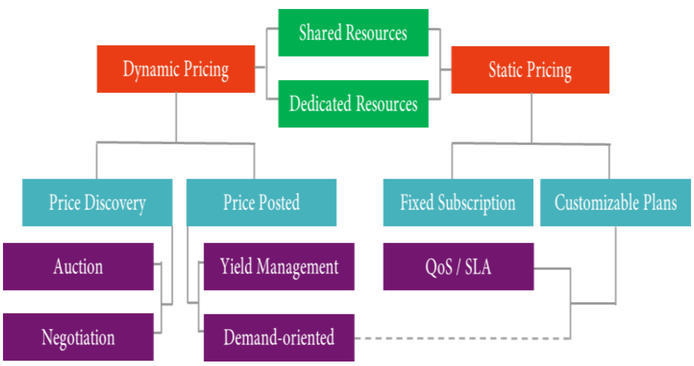

Fig. 1. Computer Resources Pricing Models

II. BACKGROUND

Computer resource service has been coined as an umbrella term to describe anything that involves delivering of computing/networking services over the world. An important aim of such services is to provide on-demand access to computational resources on remote-access basis similar to the way in which we obtain services from public utility services such as water, electricity, gas and telephony [5]. Especially Cloud Computing is the last technology that provide such ability and serve many services [6]. Such services were initially offered by commercial providers such as Amazon, Google, and Microsoft and over the years, several technologies such as Virtualization, Grid computing, and Service-Oriented Architecture (SOA) significantly contributed to make cloud computing viable [7].

Cloud computing, also known as cloud, refers to both the applications delivered as services over the Internet and the hardware and software in the data centers that provide those services [8]. Essentially, there are two main stakeholders in this environment, which are the Providers (service producers) and Customers (service consumers or clients). Customers can be either software/application/network service providers who have their own service consumers or end users (e.g., organization or businesses) who use these services directly. A provider is a company or vendor that offers economically efficient services using the hardware, software and/or network resources provisioned from other providers or supplied from within its own data centers. When services are available to the public, it is called public service provider and when belongs to a business or an organization, not made available to the public, it is called private/internal service provider. In Cloud Computing there are three main types of service models usually offered by providers which is also known as cloud service stack [9]– [11].

The computer resource market paradigm has introduced new performance metric which is cost. This has shifted a traditional distributed system into a two-party computation, providers and customers, with pricing and cost as the bridge [12]. There is a large body of research devoted to minimizing cost for customers using computing services, however, relatively less work has been done on the provider's side to maximize revenue and reduce cost of service production. In this thesis, we study market and economics-inspired mechanisms to maximize providers' revenue [13]. Therefore, this chapter is devoted to review the related literature and to position the thesis.

When providers understand how much it costs to provide the service, how much competitors are charging for the same service, and how customers perceive the value of services, it is time to figure out what type of pricing model they should utilize [14]. The most commonly used pricing models in markets, especially in infrastructure as-a-service cloud marketplaces, are usage-based, subscription-based, and demand-oriented pricing models (Figure 2).

Usage-based pricing model: Basically, resource providers can be defined as delivery of on-demand access to computing services on a pay-as-you-go basis in cloud service providers. In the usage- based pricing model, providers often charge for services only on a fixed-rate basis. Fixed rate pricing is a relatively simple model and most often requires easily controllable cost-plus pricing strategy. There is a large body of literature on cost analysis of running applications considering the usage-based pricing model in clouds [12], [15], [16].

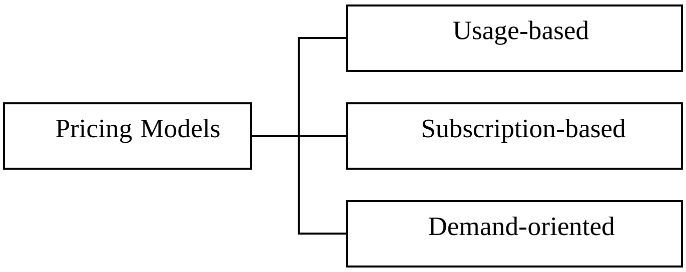

Fig. 2. Common pricing models in computer resource environments

A related work by Sharma et al. [17] developed a resource pricing model that uses financial option model to give a lower bound on the prices and compounded-Moores law taking into account the metrics such as initial investment, rate of depreciation, and age of resource to give a upper bound on prices for what they call compute commodities.

Subscription-based pricing model: is a pricing model that allows customers to pay a subscription fee to use the service for a particular time period. For example, in the case of GoGrid, to use its prepaid plan, customers pay a subscription fee to reserve VM instances for monthly or annual contracts and after which the usage is free for the contract period. In Amazon Web Services [18], the customer pays an upfront reservation fee to reserve an instance for a one or three year term and usage-based rate for that instance is heavily discounted.

Niu et al. [19] propose a guaranteed service model for bandwidth reservation, where each customer does not require to estimate the absolute amount of bandwidth, he/she needs to reserve. Their objective is to determine the optimal policy for pricing bandwidth reservations in the presence of demand uncertainty such that the social welfare is maximized, that is, the sum of the expected profits for all customers and the provider is maximized.

Meinl et al. [20] discuss the application of reservation systems in resource provider environments and point out the benefits for vendors as well as their customers. The authors analyzed the application of derivative pricing techniques and yield management to create a model that can be utilized in real world systems.

Mohammadi et al. [21] propose a novel reservation mechanism to protect both providers and customers from the cost overhead of over-provisioning resources. In their reservation mechanism, consumers can communicate their workload forecasts as a pre- reservation and then claim the pre- reserved resources if the need actually arises for the softly reserved resources in future. Pre- reservations capture the estimated number of resources that will be required by a customer at a given future point of time as well as the probability of actually needing these resources. The proposed approach encompasses mechanisms to exploit the required information to be exchanged between the provider and the customer in a way that it leverages benefits of both providers and customers.

Similarly, Lu et al. [22] provide a solution for the resource reservation problem in IaaS providers with limited resource capacity. Their proposed method investigates the feasibility of each submitted reservation request and if the provider is not able to accept the request, an alternative way of accommodating the request with backward or forward shifting in time is suggested. They utilize computational geometry to tackle the problem.

Wang et al. [23] study the resource reservation management issues inside cloud environments. They propose an adaptive resource reservation approach by selectively accepting reservation requests. The decision is made to maximize the provider revenue while it ensures the quality of service (QoS) for transactional applications.

Demand-oriented pricing model: Demand-oriented pricing model is the process of establishing a price for a service based on the level of demand. The service price is changed according to its demand in a way that when the demand is high the price goes up and when it is low the price goes down. Among all pricing models discussed here, this is the least common pricing model at real- world marketplaces; however, it has received the highest attention from researchers in academia due to its complexities. Amazon is one of the IaaS cloud providers that publicly offers a demand- oriented pricing model for selling IaaS resources. The resources are called spot instances and are sold according to a dynamic pricing model that varies the price of instances in real-time based on supply and demand according to Amazon's claim.

A relevant study has done by Niyato et al. [24] where they present an economic analysis of the resource market. Three types of resource market between private customers and service providers have been considered, i.e., monopoly, competitive, and cooperative oligopoly. Repeated game model has been used to analyze the cooperation behavior of the providers to reach efficient and fair profit.

Kantere et al. [25] presented an optimal pricing method achieved through the dynamic pricing for a caching service offered by the service provider. They propose a novel price-demand model designed for a cloud cache and a dynamic pricing scheme for queries executed in the cloud cache. They also discuss qualitative aspects of the solution that allows the consideration of customer satisfaction together with cloud provider profit maximization.

A dynamic pricing scheme suitable for federated cloud environments with rational and self- interested parties where resource demand and supply fluctuate as customers join and leave the system has been proposed by Mihailescu and Teo [26]. They compare the performance of their proposed strategy-proof dynamic pricing model with fixed-rate usage-based pricing using simulations, and show that customer welfare and the percent- age of successful requests are increased when their dynamic pricing model is applied.

As the importance of dynamic pricing in resource pricing has been recognized by the literature [12], [26], [27], we devote the next section to discussing and reviewing dynamic pricing related work.

Additionally, dynamic pricing helps providers in more-effective resource management and capacity planning. By ensuring that prices match the market conditions, fully flexible dynamic pricing mechanisms also empowers customers to manage their cost efficiently. However, this dynamism of prices makes providers' pricing decisions and customers' budget planning further challenging. This can be the main reason that currently fewer number of providers are offering dynamic pricing models. Nonetheless, as competition among providers in cloud computing marketplace for example, grows and more complex pricing models appear, dynamic pricing models would gain more popularity and approval gradually.

In general, dynamic pricing can be determined by a provider revenue maximization problem in a monopoly market, or by a social welfare maximization problem in a competitive market with multiple providers [28]. Mihailescu and Teo [26] present dynamic pricing scheme for a cloud environment aiming at maximizing individual participants' welfare in the IaaS cloud market with large number of providers and customers. Similarly, Hassan et al. [29] study the design of theoretical game models using a price-based resource allocation strategy among the IaaS cloud providers in a federated cloud environment. They develop cooperative and non-cooperative games and examine the social welfare maximization under each game model. Menache et al. [30] use pricing as a means to maximize the social welfare in cloud environments. They show that the socially optimal operating point is unique, and can be induced by a per-unit pricing model with the same price to all customers.

One of the main techniques used in the literature to maximize providers' revenue is dynamic pricing. Xu and Li [27] present an infinite horizon stochastic dynamic program to maximize the cloud provider's revenue with stochastic demand arrivals and departures. They rely on dynamic pricing as the main technique to maximize revenue. The problem of revenue maximization with dynamically allocating resources to spot markets has been investigated by Zhang et al. [31]. Supply adjustment and dynamic pricing are used as a means to maximize revenue and meet customer demand. Wang et al. [32] propose an optimal mechanism for a dynamic pricing in spot market based on the uniform price auction. The problem of optimal capacity segmentation between requests form usage-based market and dynamic pricing market has been also formulated and addressed. In order to maximize the provider's revenue, a dynamic pricing model based on genetic algorithms has been proposed by Mac´ıas and Guitart [33]. The proposed method has been compared with demand-oriented pricing model proposed in previous works from the same authors. In the context of pricing, we propose a strategy-proof dynamic pricing mechanism for allocating dedicated resources with multiple resource types. We assume a Standalone computer resource market where rational users cannot provide (=Sell) but they can utilize resources (=Buy). Rational users represent either an individual or an organization. Interoperability provides the buyers with uniformity and elasticity.

On the other side some service providers use fixed static pricing method with fixed plans. They provide services with some simple pricing metrics and users are not able to choose what they really need. The managerial side of this story is somehow different. Although all service providers like to have more revenue but there are some commercial tricks which are not understandable by most of the proposed pricing methods. There are some issues:

- It is not acceptable to power off a resource which is available in an environment since the company has spent money on that resource and would like to make money on that. (Power Efficiency is not the only side of this story and the revenue model should be discussed too. That means Non-Storable Resources during time.)

- Having more customers during the long lasting period of time is much more important than income, sometimes. There are many companies who are advertising and they spend money to achieve a better situation in public and build a good vision in the mind of professionals. Then, it is not true only to think about revenue maximization in our optimizations as there are some commercial vision and rules in business.

- Rejecting a customer's request is not as easy as returning a FALSE value in a function. It has many side effects on business when someone leaves your order process. Assume that going to create a Gmail account results: "We do not provide any email account now, please try again later". No one will return again, Would you?

Our proposed method is going to achieve these important characteristics:

- The main goal of the method is to sell our available resources as much as possible. Not only CPU but also Memory and Storage (Our Assumptions are these three resource types).

- We do not want to lose a customer when we have available resources but we really like to sell the available resources to those customers whose requests are more compatible to the available resources scales (Not to have Injured Resources). In such a situation we will use prediction based on the history of provider which will be a good issue of future discussions.

- While it is very important not to reject new customers because of their dissatisfaction, also it would be useful to offer some discounts as a satisfactory feature. We would like to sell normal plans in normal cases and we would like to give discounts to people who could utilize our unused resources in unbalanced scale. Example: When in one slice of time available resources are 2 CPU Cores + 3GB Memory + 20GB Storage , it would be much more acceptable to sell it as one VM instead of selling two VMs each unit 1 CPU Cores + 1GB Memory + 10GB Storage, which will cause 1GB of wasted memory.

In probability theory, the normal (or Gaussian) distribution is a very common continuous probability distribution. Normal distributions are important in statistics and are often used in the natural and social sciences to represent real-valued random variables whose distributions are not known.

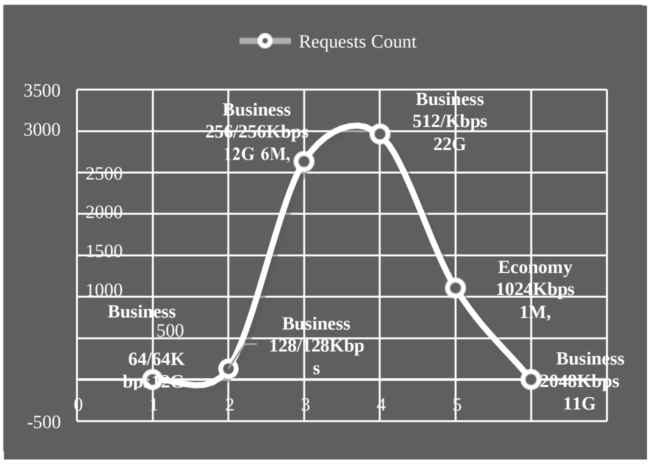

Fig. 3. The arrival request of an Internet Service Provider

The normal distribution is useful because of the central limit theorem. In its most general form, under some conditions (which include finite variance), it states that averages of random variables independently drawn from independent distributions converge in distribution to the normal, that is, become normally distributed when the number of random variables is sufficiently large. A random variable with a Gaussian distribution is said to be normally distributed and is called a normal deviate. In this work we use the Normal distribution to model the requested plans customers do agree on the arrival. To be sure that this is the best choice of the model we accessed one of the most famous ISP companies in the city and analyzed the customers invoices to see if the strategy is correct or not. The result is shown in the following figure 3.

III. Environments

The experiments presented in this section were developed using EasySim discrete-event simulator. The simulated scenario is composed of computer resource providers. The number of servers is one of the simulation parameters, and we evaluate the effect of the policies considering different numbers of members.

For the simplicity, we assume only six types of service is offered by providers. The configuration is inspired by a real company small instances mentioned below in table 1. Adding different types to the model can be considered as an extension of the current work.

Table 1. Default Plans of a Real Company

	Plan 1	Plan 2	Plan 3	Plan 4	Plan 5	Plan 6	Plan 7	Plan 8
CPU Core	1	2	4	6	9	12	16	24
RAM	256MB	512MB	1GB	2GB	4GB	8GB	16GB	24GB
STORAGE	25GB	50GB	100GB	200GB	500GB	1000GB	1000GB	2000GB
Price $	$18.75	$37.5	$75	$130	$255	$450	$570	$980

Figure 4 shows that providers follow the pricing of several companies other than amazon spot price [34] at the time of experiments. That is, all providers charge their customers based on monthly usage per on-demand resource. In the case of spot resource, the provider charges customers based on the spot price, which fluctuates periodically according to the minimum bid in the system and resource availability. The price for each spot is set at the beginning of each period of time for the entire duration. ($10 ~ Per CPU Core $10 ~ Per 1GB of Memory $0.25 ~ Per 1GB of Storage)

Fig. 4. Same Pricing on Many Accessed Providers

Each simulated data center contains 100 servers, and each server supports 24 Separate CPU Cores

+ 24GB Memory + 2TB Storage. So, each provider is able to concurrently host 2400 plan I VM. We assumed that operational costs are constant and the same for all the providers, so they are not considered in the experiment. For accuracy, each experiment is tried to be carried out several times by using different workloads and the average of the results is reported. We explain the workload setup details in the following subsection.

- Average user welfare (Discounts)
- Number of successful requests, for buyers
- Number of allocated resources, for sellers.

The eco-system has three phases in operation:

• Assign Normal Dedicated Plans up to X% of the system's bottleneck resource (Which is CPU Cores in our environment as if we sell only normal plans.)

Assign Priority-Based Dedicated Expert Plans based on professional usages up to Y% using Multi-Variable Knapsack or Solving a Linear Model Optimization (as we did) to see who is more accepted in using the company resources and give more discount to him (Example: Extra Large Storage in addition to normal plan for Hadoop usage or Extra CPU cores in Parallel Computing.)

• Assign Shared Plans based on remaining resources up to Z% using IaaS ONE VM to see who is more accepted in using the company resources and give more discount to him (We do not have any guarantee on this type of resource. It is very cheap service in compare with others).

Note: X, Y and Z, are three managerial parameters set on the system to apply the managerial decisions (Here the assumption is: 75/20/5). One may place 0 in each. Like what current companies can be modelled as 100/0/0.

Arrival Request (P,RC, RR, RS,…)	**//The Allocation Function**
{	
if(**Check Total**()<X)	//Cheking Remaining Resources in
{	Total Cloud Environment in Cycle
	#1
Assign if Req ∈ **LPModel**(RC, RR, RS,…)	//Finding Best Selling Option
}	
Else if(**MonitorTotal**()<Y)	Else if(MonitorTotal()<Y) //Checking Remain
{	{ Total Cloud Envir
	#2
If(P ∈ PriorityPlans)	If(P ∈ PriorityPlans)
Assign if Req ∈ **LPModel**(RC, RR, RS,…)	Assign if Req ∈ LPModel (RC, RR, RS,…)
}	}
ELSE	ELSE // No More Space,
Negotiation on Shared Assignment	Negotiation on Shared Assignment
}	}

RC, RR, RS, Requested Resources, CPU, RAM, Storage, …

P: Requested Plan

IBM CPlex will solve this program with such a model code:

Maximize (Maximizing the revenue based on number of sold packages multiply with price of each)

obj: 18.75 x1 + 37.5 x2 + 75 x3 + 130 x4 + 255 x5 + 450 x6 + 570 x7 + 980 x8

Subject To (Summation of each resource – c1: CPU, c2: Memory, c3: Storage) c1: x1 + 2 x2 + 4 x3 + 6 x4 + 9 x5 + 12 x6 + 16 x7+ 24x8 <= 2400

c2: 0.25 x1 +0.5 x2 + 1 x3 + 2 x4 + 4 x5+ 8 x6+ 16 x7 + 24 x8 <= 2400

c3: 25 x1 + 50 x2 + 100 x3 + 200 x4 + 500 x5 + 1000 x6 + 1000 x7 + 2000 x8 <=

200000

Bounds

0 <= Where all xn End

IV. Experimental Results

Results presented for profit and utilization are the normalized values for each metric using the result obtained from the normal policy as the base value. Since the policy reflects the situation where providers do not explore capacities of the total environment, the use of normalized values allows us to quantify the benefits of high level view policies on each provider.

To reach these key features we have evaluated the proposed dynamic pricing scheme both for economic and computational efficiency. Using simulation, we compare our pricing scheme with fixed pricing, currently used by many providers. We implement our framework as an application built on top of the data from a real service provider environment as a case study (Because of the commercial secrecy in cloud sellers, we do not have the permission to publicly announce their brand name).

For simplicity, we used a centralized market-maker to compare the efficiencies of the two pricing schemes. A centralized implementation has the advantage of allowing the measurement of economic and computational efficiency with a simple setup for a simulated network. Moreover, the use of an API-based simulation allows us to address the scalability issue in our future works and even the real implementation accordingly. Thus, our simulated environment contains one market-maker and 100 nodes, where each node can be checked separately. Order and Resource check processes are sent to the market-maker node, which then performs the assignment using the first-come-first-serve policy and computes the payments in the simplest way.

The Normal Fixed price method used by our case study with the default plans take the cloud to waste many resources like what the simulator exported in figures 5,6 and 7. The total income with this structure is mentioned here:

- Total CPU Sold: 2275 Core x $ 10 =22750
- Total Ram Sold: 1021700 Byte x $ 10/1000 =10217
- Total Storage Sold: 114175 GB x $ 0.25 =28543.75
- Total CPU+RAM+Storage= **$ 61510.75** /month

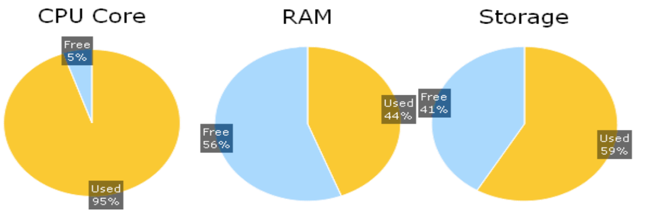

Fig. 5. Companies Default Sold Resources

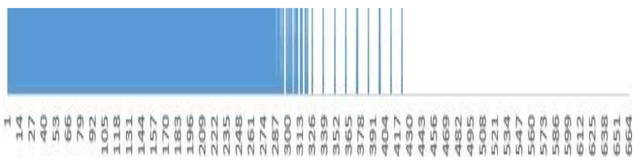

Fig. 6. Rate of Arrival Request Acceptance

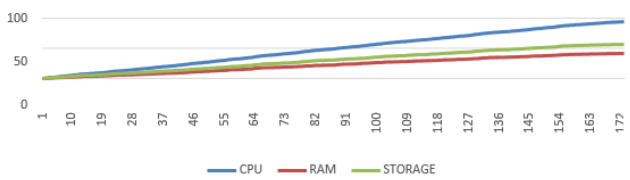

Fig. 7. Resource Allocation Rates based on Number of Users

The proposed dynamic price method used by our case study with the default plans for first phase (X=75%) and Specialists Plans (Y=20% | Total: 95%) take the cloud to a much worse statistical situation like what the simulator exported in figure 8,9 and 10 and shows around 23% revenue decrement. We would make it better in next round. The total income with this structure is mentioned here:

Iteration 1 (75% Normal):

- Total CPU Sold: 1768 Core x $ 10 =17680
- Total Ram Sold: 792500 Byte x $ 10/1000 =7925

- Total Storage Sold: 87850 GB x $ 0.25 =21962.5
- Total CPU+RAM+Storage= **$ 47567.5** /month

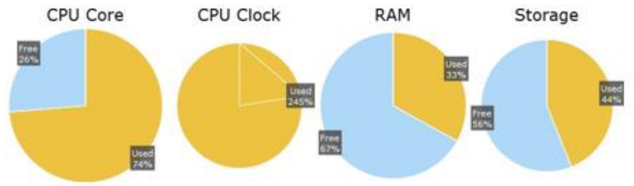

Fig. 8. Companies Sold Resources with Limitation

Fig. 9. Rate of Arrival Request Acceptance

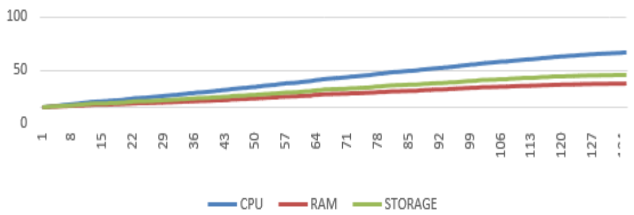

Fig. 10. Resource Allocation Rates based on Number of Users

Iteration 2 (20% Advanced Plans)

- Total CPU Sold: 2220.875 Core x $ 10 =22208.75
- Total Ram Sold: 2083100 Byte x $ 10/1000 =20831
- Total Storage Sold: 185525 GB x $ 0.25 =46381.25
- Total CPU+RAM+Storage= $ 89421 /month.

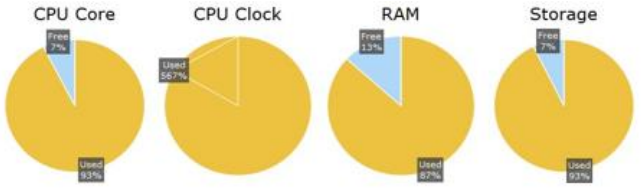

Fig. 11. Companies Sold Resources with Limitation in Cycle #2

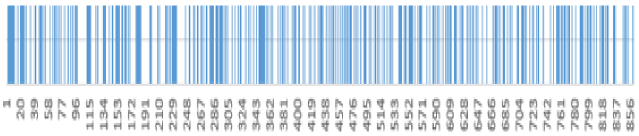

Fig. 12. Rate of Arrival Request Acceptance

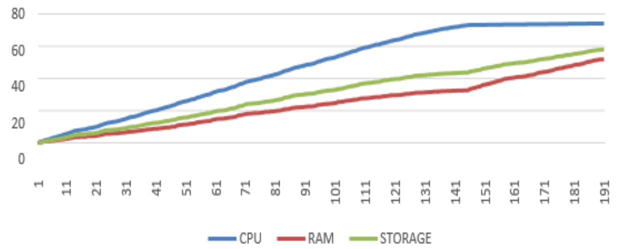

Fig. 13. Resource Allocation Rates based on Number of Users

Last Iteration (5% Shared):

- Total CPU Sold: 2269.775 Core x $ 10 =22697.75
- Total Ram Sold: 2155340 Byte x $ 10/1000 =21553.4
- Total Storage Sold: 189445 GB x $ 0.25 =47361.25
- Total CPU+RAM+Storage= **$ 91612.4** /month

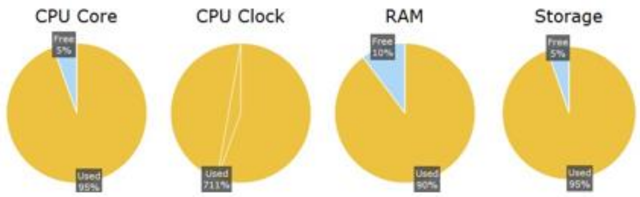

Fig. 14. Companies Sold Resources with Limitation in Cycle #3

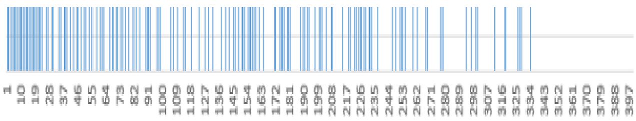

Fig. 15. Rate of Arrival Request Acceptance

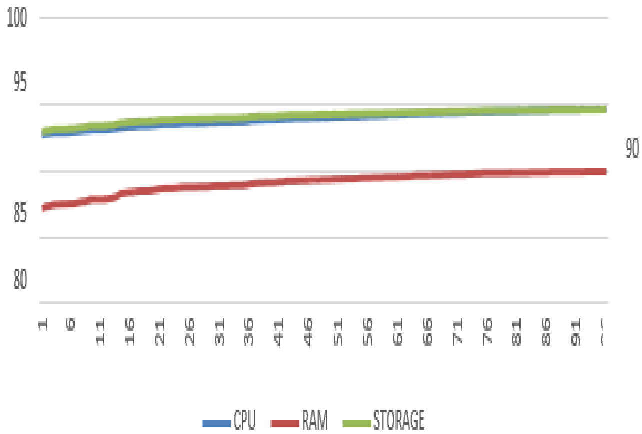

Fig. 16. Resource Allocation Rates based on Number of Users

Total revenue shows 49% of increase in selling services and more than 100% increase in number of customers.

V. CONCLUSION

Using our policies, we evaluated the impact of different parameters such as ratio of shared services to the total number of services, percentage of persistent services, number of providers, and provider's load on various performance metrics such as profit, utilization, and rejection rate. Results demonstrated that our policies help providers to increase profit and to reject fewer requests, while keeping utilization at an acceptable level. Experimental results also allowed us to derive some guidelines for providers. For example, running on-demand requests locally is more profitable if ratio of requests to total number of shared resources is high and termination of old services may lead to less discontinuation of the service consumption. Moreover, outsourcing is more profitable when services are scarce and termination may result in discontinuation of the service consumption. Experimental results showed that financial option-based contracts between providers would help them to exploit the underutilized reserved capacity without any concern to acquire the needed resources at any given time. Using our model, the provider can increase the profit while keeping the rejection rate of reserved requests at a negligible level. The model therefore, contributes to obtaining a trust and goodwill from the provider's client base. Computer resources and computing has evolved as a key IT platform that aims at providing computing resources for hosting applications as a utility. Market and economics-inspired mechanisms explored in this paper will enable providers to increase their return on investment while they honor QoS requirements of customer applications. Research similar to what has been done in this paper, therefore, becomes more important in driving further innovation and development in cloud computing.

References

[1] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," Journal of internet services and applications, vol. 1, no. 1, pp. 7–18, 2010.

[2] M. A. Smith and R. L. Kumar, "A theory of application service provider (ASP) use from a client perspective," Information & management, vol. 41, no. 8, pp. 977–1002, 2004.

[3] T. Redkar and T. Guidici, Windows Azure Platform, vol. 1. Springer, 2011.

[4] E. Ciurana, "Google App Engine," Developing with Google App Engine, pp. 1–10, 200

[5] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," Future Generation computer systems, vol. 25, no. 6, pp. 599–616, 2009.

[6] A. Monaco, "A View inside the Cloud," theinstitute. ieee. org, 2012.

[7] W. Voorsluys, J. Broberg, and R. Buyya, "Introduction to cloud computing," Cloud computing: Principles and paradigms, pp. 1–44, 2011.

[8] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and others, "A view of cloud computing," Communications of the ACM, vol. 53, no. 4, pp. 50–58, 2010.

[9] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds: towards a cloud definition," ACM SIGCOMM Computer Communication Review, vol. 39, no. 1, pp. 50–55, 2008.

[10] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud computing and grid computing 360-degree compared," in Grid Computing Environments Workshop, 2008. GCE'08, 2008, pp. 1–10.

[11] A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and

[12] Stoica, "Above the clouds: A Berkeley view of cloud computing," Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS, vol. 28, p. 13, 2009.

[13] H. Wang, Q. Jing, R. Chen, B. He, Z. Qian, and L. Zhou, "Distributed systems meet economics: pricing in the cloud," in Proceedings of the 2nd USENIX conference on Hot topics in cloud computing, 2010, pp. 6–6.

[14] S. Dutta, M. J. Zbaracki, and M. Bergen, "Pricing process as a capability: A resource-based perspective," Strategic management journal, vol. 24, no. 7, pp. 615–630, 2003.

[15] S. Maxwell, the price is wrong: Understanding what makes a price seem fair and the true cost of unfair pricing. John Wiley & Sons, 2007.

[16] B.-C. Tak, B. Urgaonkar, and A. Sivasubramaniam, "To Move or Not to Move: The Economics of Cloud Computing.," in HotCloud, 2011.

[17] U. Sharma, P. Shenoy, S. Sahu, and A. Shaikh, "A cost-aware elasticity provisioning system for the cloud," in Distributed Computing Systems (ICDCS), 2011 31st International Conference on, 2011, pp. 559–570.

[18] B. Sharma, R. K. Thulasiram, P. Thulasiraman, S. K. Garg, and R. Buyya, "Pricing cloud compute commodities: a novel financial economic model," in Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012), 2012, pp. 451–457.

[19] H. Xu and B. Li, "A study of pricing for cloud resources," ACM SIGMETRICS Performance Evaluation Review, vol. 40, no. 4, pp. 3–12, 2013. M. M. Hassan, B. Song, and E.-N. Huh, "Distributed resource allocation games in horizontal dynamic cloud federation platform," in High Performance Computing and Communications (HPCC), 2011 IEEE 13th International Conference on, 2011, pp. 822– 827.

[20] I. Menache, A. Ozdaglar, and N. Shimkin, "Socially optimal pricing of cloud computing resources," in Proceedings of the 5th International ICST Conference on Performance Evaluation Methodologies and Tools, 2011, pp. 322–331.

[21] Q. Zhang, Q. Zhu, and R. Boutaba, "Dynamic resource allocation for spot markets in cloud computing environments," in Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on, 2011, pp. 178–185.

[22] W. Wang, B. Li, and B. Liang, "Towards optimal capacity segmentation with hybrid cloud pricing," in Distributed Computing Systems (ICDCS), 2012 IEEE 32nd International Conference on, 2012, pp. 425–434.

[23] M. Macías and J. Guitart, "A genetic model for pricing in cloud computing markets," in Proceedings of the 2011 ACM Symposium on Applied Computing, 2011, pp. 113–118.

[24] J. Varia, "Best practices in architecting cloud applications in the AWS cloud," Cloud Computing: Principles and Paradigms, pp. 457–490, 2011.

[25] A. W. S. Amazon, "Amazon Web Services," Available in: http://aws. amazon. com/es/ec2/(November 2012), 2010.

[26] D. Niu, C. Feng, and B. Li, "Pricing cloud bandwidth reservations under demand uncertainty," in ACM SIGMETRICS performance evaluation review, 2012, vol. 40, pp. 151–162.

[27] T. Meinl, A. Anandasivam, and M. Tatsubori, "Enabling cloud service reservation with derivatives and yield management," in 2010 IEEE 12th Conference on Commerce and Enterprise Computing, 2010, pp. 150–155.

[28] S. V. Mohammadi, S. Kounev, A. Juan-Verdejo, and B. Surajbali, "Soft Reservations: Uncertainty-Aware Resource Reservations in IaaS Environments," in Proceedings of the 3rd International Symposium on Business Modeling and Software Design (BMSD'13), Noordwijkerhout, The Netherlands, 2013.

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

[30] X. Wang, Y. Xue, L. Fan, R. Wang, and Z. Du, “Research on adaptive QoS-aware resource reservation management in cloud service environments,” in Services Computing Conference (APSCC), 2011 IEEE Asia-Pacific, 2011, pp. 147–152.

[31] D. Niyato, S. Chaisiri, and B.-S. Lee, “Economic analysis of resource market in cloud computing environment,” in Services Computing Conference, 2009. APSCC 2009. IEEE Asia-Pacific, 2009, pp. 156–162.

[32] V. Kantere, D. Dash, G. Francois, S. Kyriakopoulou, and A. Ailamaki, “Optimal service pricing for a cloud cache,” Knowledge and Data Engineering, IEEE Transactions on, vol. 23, no. 9, pp. 1345–1358, 2011.

[33] M. Mihailescu and Y. M. Teo, “Dynamic resource pricing on federated clouds,” in Cluster, Cloud and Grid Computing (CCGrid), 2010 10th IEEE/ACM International Conference on, 2010, pp. 513–517.

H. Xu and B. Li, “Dynamic cloud pricing for revenue maximization,” IEEE Transactions on Cloud Computing, vol. 1, no. 2, pp. 158–171, 2013.