

A NEW REVENUE MAXIMIZATION MODEL USING CUSTOMIZED PLANS IN CLOUD SERVICE ALLOCATION

Applied On A Real Company Case Study

M. M. Raja¹, G. Dastghaibfard¹, A. Nazemi²

¹Computer Eng. and IT Department, Shiraz University, Iran

²Department of Accounting, Shiraz University, Iran

Published online: 15 May 2016

ABSTRACT

Cloud computing is emerging as a promising field offering a variety of computing services to end users. 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 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 shows the appropriateness of dynamic pricing for sharing of computing resources, where providers want to have more customers as a managerial decision and even more income in total.

Keywords: Cloud Computing; Digital Pricing; Dynamic Pricing.

Author Correspondence, e-mail: mrja@shirazu.ac.ir

doi: <http://dx.doi.org/10.4314/jfas.v8i3s.272>



1. INTRODUCTION

Cloud computing is emerging as a critical practice for the online provisioning of computing resources as services. This technology allows scalable on-demand sharing of resources and costs among a large number of end users. It enables end users to process, manage, and store data efficiently at very high speed with reasonable prices. Customers of cloud computing do not need to install any kind of software and can access their data worldwide from any computer as long as an Internet connection is available [1].

Many definitions have been presented for cloud computing [2]–[4]. Foster et al., [3] defined cloud computing as “a large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet.” Cloud computing provides various computing services online based on SLAs between the provider and the consumer.

Cloud computing providers offer many services to their customers [5], including infrastructure as a service (IaaS), platform as a service (PaaS), software as a service (SaaS), storage as a service (STaaS), security as a service (SECaaS), test environment as a service (TEaaS), and many more. A cloud computing provider’s typical goal is to maximize its revenues with its employed pricing scheme, while its customers’ main goal is to obtain the highest level of quality of service (QoS) feasible for a reasonable price. Therefore, satisfying both parties requires an optimal pricing methodology. The price charged is one of the most important metrics that a service provider can control to encourage the usage of its services.

Pricing is a critical factor for organizations offering services or products [6]. How the price is set affects customer behavior, loyalty to a provider, and the organization’s success. Therefore, developing an appropriate pricing model will help achieve higher revenues. The price determined for a service or product must consider the manufacturing and maintenance costs, market competition, and how the customer values the service or product offered. Iveroth et al., [7] analyzed the possible sets of price models that different organizations can employ. Their research illustrated how price is connected to a set of many implicit features of the price model. Such an approach helps in resolving many issues regarding pricing between the customer and the provider.

Software vendors utilize many pricing techniques. For example, a typical pricing approach is to pay once for limitless usage. However, this approach is inflexible and does not consider many other factors that affect pricing, such as the age of resources and price fairness [8]. Many

major cloud computing providers (e.g., Amazon Web Services [9] and Google App Engine [10]) employ “pay-per-use fixed pricing,” which charges users according to their overall resource consumption. “Pay for resources” is another technique, in which users are charged according to the storage or bandwidth size provided. Subscription is another pricing technique, in which the customer subscribes with a certain service provider for a fixed price per unit for long periods of time. Moreover, a service level agreement (SLA) is an essential part of cloud computing. It describes the negotiations between the provider and the customer regarding the services provided. The final agreement is verified via a contract between the involved parties. An SLA might involve agreements regarding QoS, pricing, guarantees, and so on.

Samimi and Patel [11] introduced a review and comparison of the recent pricing models in grid and cloud computing and their economic models. They also highlighted the differences in grid and cloud computing by comparing their usage, standardization, virtualization, and SLAs. They studied pricing models thoroughly in grid computing and compared them to those in cloud computing. However, the number of pricing models compared is insufficient to draw conclusions. Moreover, the fairness of each model, which is an important factor to assess pricing models, was not stated.

In our work, we present a thorough comparison between many proposed cloud computing pricing models and schemes. We consider many factors that affect pricing and user satisfaction, such as fairness, QoS, and more, by highlighting their importance in recent markets. We consider recent pricing models and their pricing approaches. We also introduce the pros and cons of each model to provide a solid ground to design future improved models. The rest of the paper is organized as follows. Section II presents background information for cloud computing and pricing. Section III describes a novel method of pricing based on the priority of customer’s types. Section IV presents a thorough comparison between different factors and parameters in our model and Section V includes our conclusions and remarks.

2. BACKGROUND

Different service providers employ different schemes and models for pricing. However, the most common model employed in cloud computing is the “pay-as-you go” model. Customers pay a fixed price per unit of use. Amazon [9], considered the market leader in cloud computing, utilizes such a model by charging a fixed price for each hour of virtual machine usage. The “pay-as-you-go” model is also implemented by other leading enterprises such as Google App Engine [10] and Windows Azure[12]. Another common scheme employed by

these leading enterprises is the “pay for resources” model. A customer pays for the amount of bandwidth or storage utilized. Subscription, where a customer pays in advance for the services he is going to receive for a pre-defined period of time, is also common.

Nevertheless, many useful theoretical studies for cloud computing pricing have been introduced. Sharma et al., [13] proposed a novel financial economic model capable of providing a high level of QoS to customers. They employed the financial option theory and treated the cloud resources as assets to capture their realistic value. The price determined using this model represented the optimal price that the service provider should charge its customers to recover the initial costs. The financial option theory gave a lower boundary on the price that should be charged to customers. The upper boundary of the price was determined using a proposed compounded Moore’s law. This law, presented by the authors, combined Moore’s law [14] with the compounded interest formula. The authors claimed that, if the price was set between these two boundaries, it would be beneficial for both customers and service providers. This approach was interesting; however, it did not take into consideration the maintenance costs. The authors also assumed that the initial costs would be the same for clients and providers, which is not true. Service providers get discounts for buying a larger amount of assets.

Wang et al., [15] proposed an algorithmic solution to optimize data center net profit with deadline-dependent scheduling by jointly maximizing revenues and minimizing electricity costs. They developed two distributed algorithms for the net profit optimization: Net Profit Optimization for Divisible jobs (NPOD), and Net Profit Optimization for Indivisible Jobs (NPOI). An indivisible job is a job that cannot be interrupted, while a divisible job can be interrupted or divided. The authors proved via simulations their algorithm’s capabilities to increase revenues and reduce electricity costs by comparing it to the Largest Job First (LJF) algorithm. However, the authors considered only static job arrivals and departures. They also assumed that the servers at all data centers were homogenous, which is not realistic. Macias and Guitart [16] proposed a genetic model for pricing in cloud computing markets.

Choosing a good pricing model via their genetic algorithms involved three main steps: define a chromosome, evaluate it, and finally select the best pairs of chromosomes for reproduction and discarding those with the worst results. The results of the simulation illustrated that genetic pricing acquired the highest revenues in most of the scenarios. The proposed genetic model with a flexible genome was proven to be more stable against noise and earned more money than the one with the rigid genome. The proposed genetic model is easy to implement, flexible, and easily adapted to a set of various parameters that influence pricing. The genetic pricing

approach can be further explored by defining relations between the parameters that influence pricing.

Mihailescu and Teo [17] introduced a dynamic pricing scheme for federated clouds, in which resources are shared among many cloud service providers. Federated clouds are implemented to improve reliability and scalability for both users and providers. Users in the federated environment were assumed to be capable of both buying and selling resources. In the case of high market demand, fixed pricing would minimize seller welfare because he would not be capable of raising his price. Similarly, when demand was low, user utility would be minimized because he would be charged more than the market price. Therefore, dynamic pricing would be beneficial in such environments because it would set the price according to the levels of supply and demand. It would also allow the offering of many types of resources to end users. The authors carried out simulations to determine the efficiency of this approach by comparing it to a fixed pricing scheme. They found that dynamic pricing achieved better average performance with increasing buyer welfare and numbers of successful requests up to 200%. However, fixed pricing achieved better scalability in the case of high demand in the market.

Yea et al., [18] described the difference between fixed and variable prices. Fixed prices were easier to understand and more straightforward for users. However, fixed pricing could not be fair to all users because not all users had the same needs. Their study proposed charging variable prices with advanced reservation. Charging variable pricing with advanced reservation would let users know the exact expenses that are computed at the time of reservation even though they were based on variable prices. The paper emphasized the importance of implementing autonomic metered pricing to increase revenues. Autonomic metered pricing can also be straightforward for users through the use of advanced reservations. The advantage of advanced reservations is that users can not only know the prices of their required resources in the future but are also able to guarantee access to future resources to better plan and manage their operations.

Rohitratana and Altmann [19] analyzed four dynamic pricing schemes: derivative-follower (DF), demand-driven (DD), penetration (PN), and skimming (SK). They developed an agent-based simulation of a software market that allowed the trading of two types of software licensing models. The two types of software licensing models were SaaS and perpetual software (PS). Rohitratana and Altmann's simulation results indicated that the DD pricing scheme was the best scheme in ideal cases. However, in the real world, obtaining perfect information about customers and competitors is almost impossible. This makes the DD pricing

scheme difficult to implement. PN and SK pricing schemes can be implemented easily, while they give results close to the DD pricing scheme.

Nähring [20] focused his study on four basic pricing strategies. The basic pricing strategies were cost-based pricing, customer-based pricing, competition-based pricing, and value-based pricing. Nähring highlighted the pros of cons of each one of the pricing strategies. On the other hand, Jäättmäa [21] emphasized strongly that a “pay-per-use” pricing mechanism was regarded as the key characteristic of cloud computing pricing. The study found that pay-per-use pricing significantly changed the risk-sharing model between the service provider and the customer as the customer’s commitment decreased. In addition, a pay-per-use mechanism could decrease the service provider’s incoming cash flow. Jäättmäa, therefore, proposed a new form of generic cloud computing pricing that balanced the commitment between the service provider and the customer.

Li et al., [22] proposed a pricing algorithm for cloud computing resources. This proposal used the cloud bank agent model as a resource agency because it could provide the proper analysis and assistance for all members. The authors used a price update iterative algorithm to determine the price. It analyzed the historical utilization ratio of the resources; iterated current prices constantly, assessed the availability of resources for the next round, and determined the final price. The model included a user request broker (GCA), cloud banking, a cloud service agent (CSA), and a cloud resource agent (GRA). The proposed pricing model was comparatively fixed because it could not adapt to the rapid changes that typically occur in the market. However, it could reduce the costs to providers and maximize their revenues, allowing resources to be used more effectively.

3. PROPOSED METHOD

In a resource market with a large number of providers (sellers) and users (buyers), fixed pricing does not reflect the current market price resource price due to the changing demand and supply. This leads to lower user welfare and to imbalanced markets, and even imbalanced resource allocation. Figure 1 shows the welfare lost by a seller that uses fixed pricing. In the case of under-demand, the fixed price tends to be higher than the market price and buyers may look for alternative providers. In the case of over-demand, the fixed price limits the seller welfare, which could be increased by using a higher resource price.

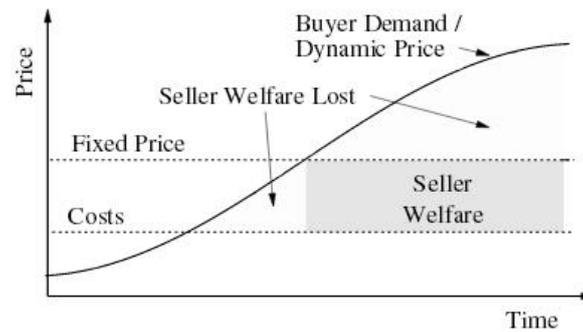


Fig.1. Fixed Pricing Limits Seller Welfare [23]

In a cloud market, dynamic pricing sets resource payments according to the forces of demand and supply. Moreover, the use of dynamic pricing facilitates sellers to provide multiple resource types. Early cloud services such as Sun Grid Compute Utility were restricted to one resource type, e.g. CPU time. More recent services, such as Amazon S3 and EC2, introduced more resource types, i.e. storage and bandwidth. Currently, Amazon has expanded its offer to 10 different virtual machine instance configurations, with different prices for each configuration, and practice tiered pricing for storage and bandwidth. We see this as step towards dynamic pricing, where users can request for custom configurations with multiple resource types based on available slots.

In the context of cloud pricing, we propose a strategy-proof dynamic pricing mechanism for allocating dedicated (and optionally shared if applicable) resources with multiple resource types. We assume a Standalone Cloud Service resource market where rational users can not 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 there are some service providers most of whom are using a 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 of their business but there are some commercial tricks which are not understandable by most of proposed pricing methods in computer-science papers. There are some sample issues:

1) It is not acceptable to power off a resource which is available in a cloud. The company spent money on that to be profitable for him. (Power Efficiency is not as simple as what computer scientists think)

2) Sometimes, having more customers is much more important than income. There are many companies who are advertising and they spend money to comply their resources availability. Then it is not true only to think about revenue maximization in our optimizations.

3) Saying “No” to a customer is not as easy as running a scheduling function. It has many side effects on business when someone leaves your order process like figure 2. Assume that going to create a Gmail account results: “We do not provide any email account now, please try later!!!”. Would you try again?

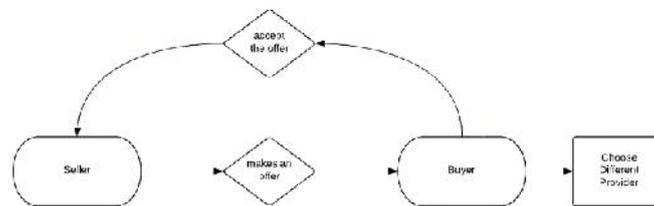


Fig.2. Normal Plan-Based Service Provider Order Process

Our proposed method is going to achieve these important characteristics:

1) The master goal of the method is to sell our available resources as much as possible like what you see in figure 3. Not only CPU but also Memory and Storage (Our Assumptions are these three resources).

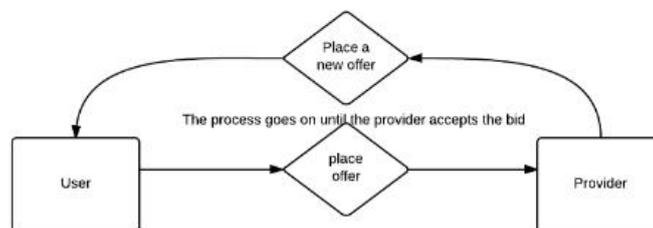


Fig.3. The Negotiation step of proposed service model

2) We do not want to lose a customer when we have available resources but we really like to sell the available resources to whom that may be more match with the available resources

scales. (Not to have Injured Resources¹). In a such situations we will use prediction based on the history of cloud which will be a good matter of future discussions.

3) 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 free resources scales. Example: When we have available resource like (2 CPU Cores + 3GB Memory + 20GB Storage) it would be much more acceptable to sell it as one VM instead of selling two (1 CPU Cores + 1GB Memory + 10GB Storage) which will cause 1GB of wasted memory.

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 cloud 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). We exported three main data from their services.

1) *Available Hardware Devices in Scale.*

100 x Servers with these specifications:

24 Separate CPU Cores + 24GB Memory + 2TB Storage

2) *Orderable Cloud VM Plans in eight types.*

- 1 Core,256MB Memory,25GB Storage
- 2 Cores,512MB Memory,50GB Storage
- 4 Cores,1GB Memory,100GB Storage
- 6 Cores,2GB Memory,200GB Storage
- 9 Cores,4GB Memory,500GB Storage
- 12 Cores,8GB Memory,1000GB Storage
- 16 Cores,16GB Memory,1000GB Storage
- 24 Cores,24GB Memory,2000GB Storage

Note: Entrance rate was assumed as normal distribution.

3) *Price per Month (Near to HP Cloud Prices):*

- \$10 ~ Per CPU Core

¹ For Example: When we have 1% Free CPU Cores available but there are around 20% Memory and 30% Storage in Total of Cloud, you are not able to sell these resources as there are not any plan to match these requirements then you loss many resources and your income will be much less than your maximum.

- \$10 ~ Per 1GB of Memory
- \$0.25 ~ Per 1GB of Storage

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.

Economic systems measure efficiency with respect to normal price for resources (utility). Consequently, in a Pareto efficient system, where economic efficiency is maximized, a user's utility cannot improve without decreasing the utility of another user. We try to have dedicated resources not to face such issues. Economic efficiency is a global measure and represents the total buyer and seller welfare. More specifically, there are three factors that affect the economic efficiency:

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

We write out all the plans in the system of constraints and rewrite the system, including the objective function[24] to see the best plan to have best revenue (x_i = No. of Customers using Plan i):

$$z = \frac{75}{4}x_1 + \frac{75}{2}x_2 + 75x_3 + 130x_4 + 255x_5 + 450x_6 + 570x_7 + 980x_8 \rightarrow \max$$

$$\left\{ \begin{array}{l} 1x_1 + 2x_2 + 4x_3 + 6x_4 + 9x_5 + 12x_6 + 16x_7 + 24x_8 \leq 2400 \\ \frac{1}{4}x_1 + \frac{1}{2}x_2 + 1x_3 + 2x_4 + 4x_5 + 8x_6 + 16x_7 + 24x_8 \leq 2400 \\ 25x_1 + 50x_2 + 100x_3 + 200x_4 + 500x_5 + 1000x_6 + 1000x_7 + 2000x_8 \leq 200000 \end{array} \right.$$

We convert the linear programming problem to the canonical form[25][26]:

To accomplish this, we will perform the following steps:

Introduce residual variable $s_i = 0$ into the constraints of form "

Introduce redundant variable $S_i = 0$ into the constraints of form "

Table 1. Initial Simplex Table

	x1	x2	x3	x4	x5	x6	x7	x8	s1	s2	s3	BFS
z(max)	$\frac{265}{12}$	$\frac{265}{6}$	$\frac{265}{3}$	115	112.5	10	$\frac{250}{3}$	0	$\frac{245}{5}$	0	0	98000
x8	$\frac{1}{24}$	$\frac{1}{12}$	$\frac{1}{6}$	$\frac{1}{4}$	$\frac{3}{8}$	$\frac{1}{2}$	$\frac{2}{3}$	1	$\frac{1}{24}$	0	0	100
s2	$-\frac{3}{4}$	$-\frac{3}{2}$	-3	-4	-5	-4	0	0	-1	1	0	0
s3	$-\frac{175}{3}$	116.667	233.333	300	250	0	333.333	0	$-\frac{250}{3}$	0	1	0

$$z = \frac{75}{4}x_1 + \frac{75}{2}x_2 + 75x_3 + 130x_4 + 255x_5 + 450x_6 + 570x_7 + 980x_8 \rightarrow \max$$

$$\begin{cases} 1x_1 + 2x_2 + 4x_3 + 6x_4 + 9x_5 + 12x_6 + 16x_7 + 24x_8 + 1s_1 = 2400 \\ \frac{1}{4}x_1 + \frac{1}{2}x_2 + 1x_3 + 2x_4 + 4x_5 + 8x_6 + 16x_7 + 24x_8 + 1s_2 = 2400 \\ 25x_1 + 50x_2 + 100x_3 + 200x_4 + 500x_5 + 1000x_6 + 1000x_7 + 2000x_8 + 1s_3 = 200000 \end{cases}$$

Now we can make an initial simplex table.

Iteration 1

We insert into the basis x8. Let us find a lead row:

$$\theta = \min_{\substack{i \\ \alpha_{ip} > 0}} \left\{ \frac{\beta_i}{\alpha_{ip}} \right\} = \min\{100, 100, 100\} = 100$$

We withdraw from the basis s1. We construct a simplex table.

Since in the row "Solutions" of the optimal simplex table there are some zeros, then the resulting solution is degenerate.

Since in the objective function row there is no negative coefficients (except R-columns), we have found the optimal point!

Table 2. 2ND Iteration Simplex Table

	x1	x2	x3	x4	x5	x6	x7	x8	s1	s2	s3	BFS
z(max)	$\frac{75}{4}$	$\frac{75}{2}$	-75	-130	-255	-450	-570	-980	0	0	0	0
s1	1	2	4	6	9	12	16	24	1	0	0	2400
s2	$\frac{1}{4}$	$\frac{1}{2}$	1	2	4	8	16	24	0	1	0	2400
s3	25	50	100	200	500	1000	1000	2000	0	0	1	200000

$$\text{Optimal point: } \left\{ \begin{array}{l} x_1 = 0 \\ x_2 = 0 \\ x_3 = 0 \\ x_4 = 0 \\ x_5 = 0 \\ x_6 = 0 \\ x_7 = 0 \\ x_8 = 100 \end{array} \right. \text{ Value of the objective function at the optimal point: 98000.}$$

Postoptimal analysis: Determining the values of resources[27]

This method of finding the value of the resources uses the following formula: $(V^0)^T = c_B^T B^{-1}$

Vector c_B^T - coefficients of the objective function corresponding to the basic variables of optimal simplex table, ie $c_B^T = (c_{x_8} \ c_{s_2} \ c_{s_3}) = (980 \ 0 \ 0)$

We extract from the optimal simplex table matrix B-1, corresponding to the optimum. To do this, we note that in the initial acceptable basic solution base identity matrix consists of columns

$$B_0 = I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = (a_{x_8} \ a_{s_2} \ a_{s_3})$$

The inverse basis matrix consists of the same columns as the matrix B0, but of the optimal simplex table.

$$B^{-1} = \begin{pmatrix} \frac{1}{24} & 0 & 0 \\ -1 & 1 & 0 \\ -\frac{250}{3} & 0 & 1 \end{pmatrix}$$

Thus,

$$c_B^T = (980 \ 0 \ 0) \begin{pmatrix} \frac{1}{24} & 0 & 0 \\ -1 & 1 & 0 \\ -\frac{250}{3} & 0 & 1 \end{pmatrix} = (40.833 \ 0 \ 0)$$

Hereby,

$$\begin{cases} y_1 = 40.833 \\ y_2 = 0 \\ y_3 = 0 \end{cases}$$

Finding the intervals of variation of the right side coefficients of constraints, in which the optimal solution is given by the current acceptable basis solution.

The variable $x_1 > 0$ nonbasis, therefore, the resource 1 is scarce. The original constraint has a sign " $>$ ".

In this case, the range of valid relative change of the resource is defined as:

$$\max_{\substack{i \\ \alpha_{is} > 0}} \left\{ \frac{\beta_i}{-\alpha_{is}} \right\} \leq \Delta_1 \leq \min_{\substack{i \\ \alpha_{is} < 0}} \left\{ \frac{\beta_i}{-\alpha_{is}} \right\}$$

Then the relative interval of stock change of 1-th resource, where the optimal solution given the current basis would be:

$$\begin{aligned} \max\{-2400\} \leq \Delta_1 \leq \min\{0; 0\} \\ -2400 \leq \Delta_1 \leq 0 \end{aligned}$$

Since the absolute interval is calculated using the formula: $b_1^{new} = \Delta_1 + b_1$, then it would be:

$$0 \leq b_1 \leq 2400$$

The variable $x_2 > 0$ (basis, in the optimal simplex table equals 2400), therefore, the resource 2 is not scarce. The original constraint has the sign " $>$ ". In this case, the range of valid relative change of the resource is defined as:

$$-s_0 \Delta_2$$

Then the relative interval of stock change of 2-th resource, where the optimal solution given the current basis would be:

$$0 \leq \Delta_2$$

Since the absolute interval is calculated using the formula: $b_{2\text{new}} = \Delta_2 + b_2$, then it would be:

$$2400 \leq b_2 <$$

The variable $x_3 > 0$ (basis, in the optimal simplex table equals 200000), therefore, the resource 3 is not scarce. The original constraint has the sign " \leq ". In this case, the range of valid relative change of the resource is defined as:

$$-\infty \leq \Delta_3$$

Then the relative interval of stock change of 3-th resource, where the optimal solution given the current basis would be:

$$0 \leq \Delta_3$$

Since the absolute interval is calculated using the formula: $b_{3\text{new}} = \Delta_3 + b_3$, then it would be:

$$200000 \leq b_3 <$$

Finding the intervals of coefficient variation of the objective function, in which the optimal solution is given by the current acceptable basis solution.

This is a maximum problem:

Variable x_1 is not a basis variable in the optimal simplex table, therefore to find relative range of variation of their coefficients of the objective function, we use the formula:

$$-\infty < \Delta_1 \leq d_{10}$$

$$-\infty < \Delta_1 \leq \frac{265}{12}$$

Since coefficient of the objective function for this variable is $75/4$, we get the following absolute range of sustainability:

$$-\infty < C_1 \leq \frac{75}{4} + \frac{265}{12} + \frac{75}{4}$$

$$-\infty < C_1 \leq \frac{245}{6}$$

Variable x_2 is not a basis variable in the optimal simplex table, therefore to find relative range of variation of their coefficients of the objective function, we use the formula:

$$-\infty < \Delta_2 \leq d_2^0$$

$$-\infty < \Delta_2 \leq \frac{265}{6}$$

Since coefficient of the objective function for this variable is $75/2$, we get the following absolute range of sustainability:

$$-\infty < C_2 \leq \frac{75}{2} + \frac{265}{6} + \frac{75}{2}$$

$$-\infty < C_2 \leq \frac{245}{3}$$

Variable x_3 is not a basis variable in the optimal simplex table, therefore to find relative range of variation of their coefficients of the objective function, we use the formula:

$$-\infty < \Delta_3 \leq d_3^0$$

$$-\infty < \Delta_3 \leq \frac{265}{3}$$

Since coefficient of the objective function for this variable is 75 , we get the following absolute range of sustainability:

$$-\infty < C_3 \leq 75 + \frac{265}{3} + 75$$

$$-\infty < C_3 \leq 163.333$$

Variable x_4 is not a basis variable in the optimal simplex table, therefore to find relative range

of variation of their coefficients of the objective function, we use the formula:

$$-\infty < \Delta_4 \leq d_4^0$$

$$-\infty < \Delta_4 \leq 115$$

Since coefficient of the objective function for this variable is 130, we get the following absolute range of sustainability:

$$-\infty < C_4 \leq 130 + 115 + 130$$

$$-\infty < C_4 \leq 245$$

Variable x5 is not a basis variable in the optimal simplex table, therefore to find relative range of variation of their coefficients of the objective function, we use the formula:

$$-\infty < \Delta_5 \leq d_5^0$$

$$-\infty < \Delta_5 \leq 112.5$$

Since coefficient of the objective function for this variable is 255, we get the following absolute range of sustainability:

$$-\infty < C_5 \leq 255 + 112.5 + 255$$

$$-\infty < C_5 \leq 367.5$$

Variable x6 is not a basis variable in the optimal simplex table, therefore to find relative range of variation of their coefficients of the objective function, we use the formula:

$$-\infty < \Delta_6 \leq d_6^0$$

$$-\infty < \Delta_6 \leq 40$$

Since coefficient of the objective function for this variable is 450, we get the following absolute range of sustainability:

$$-\infty < C_6 \leq 450 + 40 + 450$$

$$-\infty < C_6 \leq 490$$

Variable x7 is not a basis variable in the optimal simplex table, therefore to find relative range of variation of their coefficients of the objective function, we use the formula:

$$-\infty < \Delta_7 \leq d_7^0$$

$$-\infty < \Delta_7 \leq \frac{250}{3}$$

Since coefficient of the objective function for this variable is 570, we get the following absolute range of sustainability:

$$-\infty < C_7 \leq 570 + \frac{250}{3} + 570$$

$$-\infty < C_7 \leq 653.333$$

Variable x8 is a basis variable in the optimal simplex table, therefore to find relative range of variation of their coefficients of the objective function, we use the formula:

$$\max_{\substack{j \\ \alpha_j < 0}} \left\{ \frac{d_j^0}{-\alpha_j} \right\} \leq \Delta_8 \leq \min_{\substack{j \\ \alpha_j > 0}} \left\{ \frac{d_j^0}{-\alpha_j} \right\}$$

$$\max \{-530; -530; -530; -460; -300; -80; -125; -980; -980\} \leq \Delta_8 < \infty$$

$$-80 \leq \Delta_8 < \infty$$

Since coefficient of the objective function for this variable is 980, we get the following absolute range of sustainability:

$$-80 + 980 \leq C_8 \leq 980 + \infty$$

$$900 \leq C_8 < \infty$$

The eco-system has three phases in operation:

1) *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.)*

2) *Assign Priority-Based Dedicated Expert Plans based on professional usages up to Y% using the model above(OR using Multi-Variable BucketSort) 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.)*

3) *Assign Shared Plans to cover final non-allocated resources. (CPU Cores which are not needed to be dedicated for FTP Account which needs only Dedicated Storage)*

Note: *X* and *Y* are two managerial parameters set on the system to apply the managerial decisions. (Here : 70/20)

4. RESULTS

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 figure 4. The total income with this structure is mentioned here:

- Total CPU Sold: 2274 Core x \$ 10 =**22740**
- Total Ram Sold: 1058500 Byte x \$ 10/1000 =**10585**
- Total Storage Sold: 117050 GB x \$ 0.25 =**29262.5**
- Total Income = \$ **62587.5** / month After Saturation

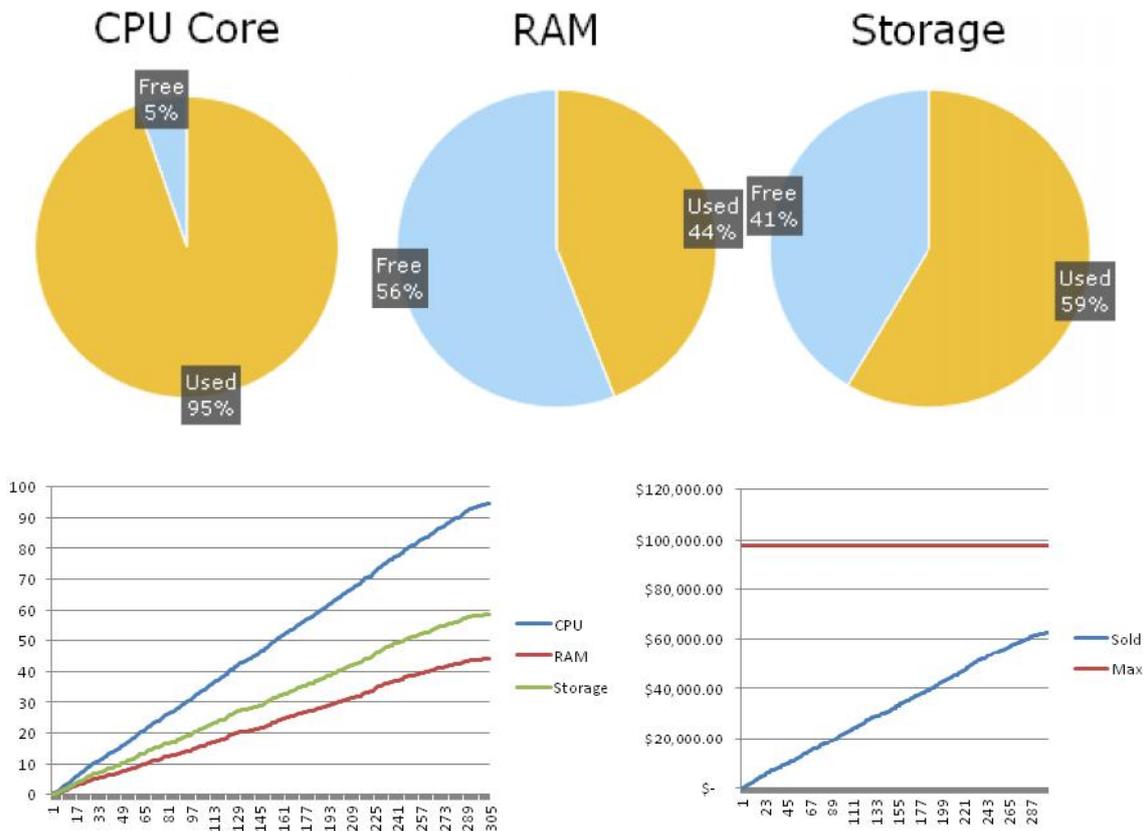


Fig.4. Case Res. Saturation with Wasted Resources/Income

The proposed dynamic price method used by our case study with the default plans for first phase (*X*=70%) and Specialists Plans (*Y*=20% | Total : 90%) take the cloud to a much better statistical situation like what the simulator exported in figure 5 and shows around 25% better

income without any extra hardware or maintenance cost. The total income with this structure is mentioned here:

- Total CPU Sold: 2269 Core x \$ 10 =**22690**
- Total Ram Sold: 2069400 Byte x \$ 10/1000 =**20694**
- Total Storage Sold: 174025 GB x \$ 0.25 =**43506.25**
- Total Income = \$ **86890.25** / month After Saturation

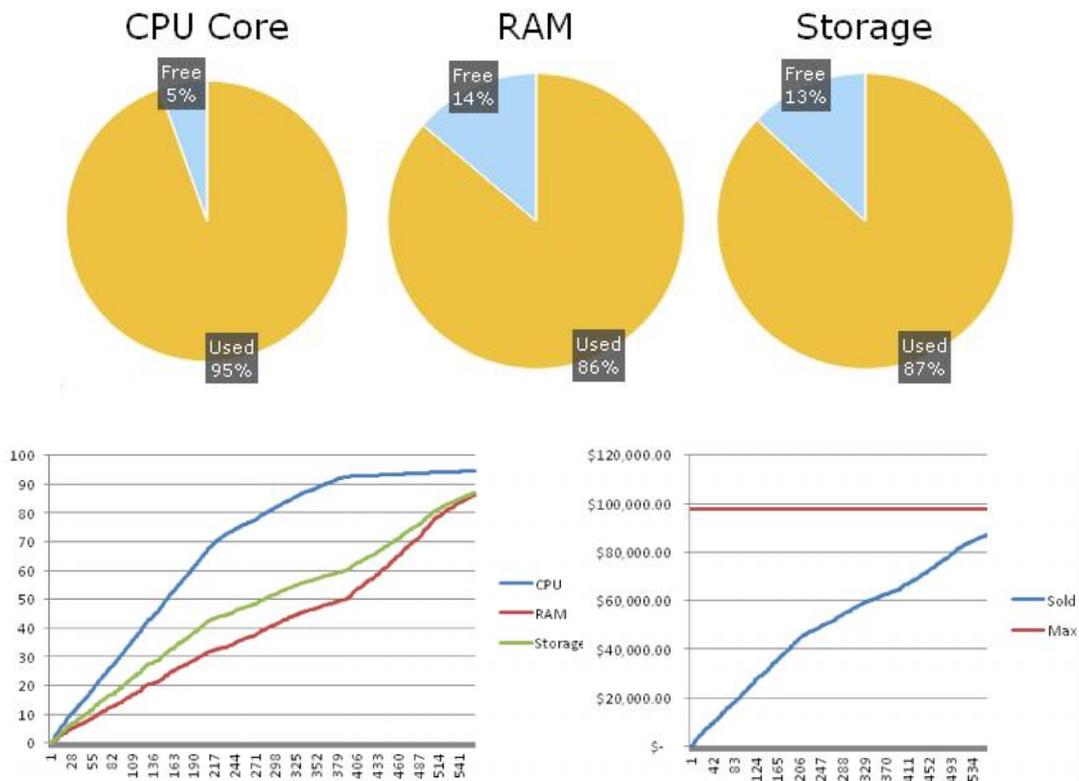


Fig.5. The 3-Phase Proposed Method Results

5. CONCLUSIONS AND FUTURE WORKS

This paper discusses current resource allocation models for cloud computing service provider and shows that dynamic pricing is more suitable for sharing of computing resources, where they want to have more customers as a managerial decision and even more income.

Using fixed pricing, the average user welfare is constant, since the user utility is also constant and the cloud owner does not have any control on its resource usage scale. In contrast, when using dynamic pricing, the average user welfare fluctuates with the computed payments, according to the resource demand. Moreover, a dynamic pricing scheme is able to balance the number of successful requests and the number of allocated resources depending on the market

condition (~300 VM in Fig.4 against ~550VMs in Fig.5). For example, resource contention in the case of over-demand is balanced by increasing the resource price or even discount on other resources. Similarly, buyers are incentivized by a lower price when the market condition is under-demand. Overall, dynamic pricing achieves better economic efficiency both with higher average user welfare, and a higher number of successful buyer requests and allocated seller resources. From our experiments we find that our method will serve higher utilization and on the other side more customers and lower rejection rate expansion shown in figure 6, while the percentage of succesful requests is also increased up to 180%.

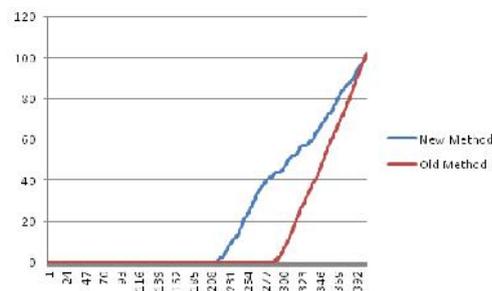


Fig.6. Comparison between Rates of Orders Rejection

Even though the pricing algorithm is polynomial, scalability becomes an issue as the number of resource types in a request increases. We are currently implementing a scheme that uses this model in a full feature framework, where multiple costing parameters should be cared and costing can affect allocation of different resource types at the same time. Applying this method to Federated Clouds is another aspect of this work which is able to be followed.

6. REFERENCES

- [1] M. Al-Roomi, S. Al-Ebrahim, S. Buqrais, and I. Ahmad, "Cloud Computing Pricing Models: A Survey," *International Journal of Grid and Distributed Computing*, vol. 6, no. 5, pp. 93–106, Oct. 2013.
- [2] 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.
- [3] 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.

-
- [4] A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. 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.
- [5] A. Monaco, "A View inside the Cloud," *theinstitute. ieee. org*, 2012.
- [6] 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.
- [7] E. Iveroth, A. Westelius, C.-J. Petri, N.-G. Olve, M. Cöster, and F. Nilsson, "How to differentiate by price: Proposal for a five-dimensional model," *European Management Journal*, vol. 31, no. 2, pp. 109–123, 2013.
- [8] S. Maxwell, *The price is wrong: Understanding what makes a price seem fair and the true cost of unfair pricing*. John Wiley & Sons, 2007.
- [9] A. W. S. Amazon, "Amazon Web Services," Available in: <http://aws.amazon.com/es/ec2/> (November 2012), 2010.
- [10] E. Ciurana, "Google App Engine," *Developing with Google App Engine*, pp. 1–10, 2009.
- [11] P. Samimi and A. Patel, "Review of pricing models for grid & cloud computing," in *Computers & Informatics (ISCI), 2011 IEEE Symposium on*, 2011, pp. 634–639.
- [12] T. Redkar and T. Guidici, *Windows Azure Platform*, vol. 1. Springer, 2011.
- [13] 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.
- [14] G. E. Moore and others, *Cramming more components onto integrated circuits*. McGraw-Hill New York, NY, USA, 1965.
- [15] W. Wang, P. Zhang, T. Lan, and V. Aggarwal, "Datacenter net profit optimization with deadline dependent pricing," in *Information Sciences and Systems (CISS), 2012 46th Annual Conference on*, 2012, pp. 1–6.
- [16] 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.
- [17] 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.

-
- [18] C. S. Yeo, S. Venugopal, X. Chu, and R. Buyya, “Autonomic metered pricing for a utility computing service,” *Future Generation Computer Systems*, vol. 26, no. 8, pp. 1368–1380, 2010.
- [19] J. Rohitratana and J. Altmann, “Impact of pricing schemes on a market for Software-as-a-Service and perpetual software,” *Future Generation Computer Systems*, vol. 28, no. 8, pp. 1328–1339, 2012.
- [20] P. Nährung, “Value-based pricing: The perception of value,” 2011.
- [21] J. Jäättmäa and others, “Financial aspects of cloud computing business models,” 2010.
- [22] H. Li, J. Liu, and G. Tang, “A pricing algorithm for cloud computing resources,” in *Network Computing and Information Security (NCIS), 2011 International Conference on*, 2011, vol. 1, pp. 69–73.
- [23] 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.
- [24] S. I. Gass, *Linear programming: methods and applications*, 5th ed. New York: Dover Publications, 2003.
- [25] B. Kolman and R. E. Beck, *Elementary linear programming with applications*. New York; London: Academic Press, 1980.
- [26] C. Woodford and C. Phillips, *Numerical methods with worked examples*, 1. ed. London: Chapman & Hall, 1997.
- [27] M. A. Goberna and M. A. López, *Post-optimal analysis in linear semi-infinite optimization*. 2014.

How to cite this article:

Raja M M, Dastghaibyfarid G, Nazemi A.. a new revenue maximization model using customized plans in cloud service allocation (Applied On A Real Company Case Study). J. Fundam. Appl. Sci., 2016, 8(3S), BCCE>BCEF?