A negotiation based dynamic pricing heuristic in cloud computing

Gaurav Baranwal
Department of Computer Science and Engineering,
Madan Mohan Malaviya University of Technology,
Gorakhpur, Uttar Pradesh, India
Email: gaurav.vag@gmail.com

Dinesh Kumar*, Zahid Raza and Deo Prakash Vidyarthi
School of Computer and Systems Sciences,
Jawaharlal Nehru University, New Delhi, India
Email: dineshyadav2571@gmail.com
Email: zahidraza@mail.jnu.ac.in
Email: dpv@mail.jnu.ac.in
*Corresponding author

Abstract: Over the years, cloud computing has emerged as a good business platform for IT related services. In cloud, prices of computing resource act as lever to control the utilisation of the resources. That is the reason, when the number of cloud customers started increasing, cloud service providers started offering resources with various pricing schemes to attract customers. This work proposes a negotiation based heuristic for dynamic pricing that considers behaviour of both the service provider and the customer and tries to optimally satisfy both for pricing. Both customer and provider are reluctant to reveal information about their utility to each other. The designed utility function for provider considers payment offered by customer and opinion of the provider about customer. Similarly, utility function for customer considers price offered by the provider and opinion of customer about provider. This will encourage both to offer their true value. Performance study indicates that the proposed method performs well and is a potential candidate for its implementation in a real cloud.

Keywords: cloud pricing; negotiations; cloud market; cloud agent; trustability.


Biographical notes: Gaurav Baranwal is a Faculty in Department of Computer Science and Engineering, MMMUT, Gorakhpur, UP, India. His research interests include resource provisioning and service coordination in cloud computing.

Dinesh Kumar is PhD student in Computer Science at the School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India. His research interests include resource provisioning and pricing in cloud computing.

Zahid Raza is a faculty in the School of Computer and Systems Sciences, Jawaharlal Nehru University, India. He is M.Sc. in Electronics, M. Tech in Computer Science. He did his Ph.D. in Computer Science from Jawaharlal Nehru University, India. His research interests include parallel and distributed systems, evolutionary algorithms and multi-objective evolutionary algorithms. He has published many peer-reviewed articles and has proposed various scheduling models for computational grid, cloud and cluster systems.

Deo Prakash Vidyarthi is Professor in the School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi. He was associated with the Department of Computer Science of Banaras Hindu University, Varanasi for more than 12 years before joining JNU as Associate Professor. Dr. Vidyarthi has published around 75 research papers in various peer reviewed International Journals and Transactions (including IEEE, Elsevier, Springer, Wiley, World Scientific etc.) and around 45 research papers in proceedings of various peer-reviewed conferences in India and abroad. He also has contributed chapters in many edited books. His Research interest includes Parallel and Distributed System, Grid and Cloud Computing, Mobile Computing and Evolutionary Computing.
1 Introduction

Cloud computing is a successful implementation of utility computing, which provides physical computing resources such as CPU, memory, storage, sensor etc. and logical resources such as operating system, file system, bandwidth etc. in form of utility using virtualisation techniques (Sajid and Raza, 2013). Various Cloud service models include Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) (Mell and Grance, 2011). Of these service models, IaaS mainly deals with physical resources by conceiving Virtual Machines (VMs) with various configurations. Transparently, users operate on these VMs as a dedicated physical host. Cloud service providers use various pricing schemes to offer cloud resources such as virtual machines, storages etc. in IaaS.

As Cloud is in commercial domain, pricing model for the Cloud services must be effective and efficient. It should be beneficial for both; cloud service providers as well as cloud customers. In the nascent stage of Cloud, service providers offered static pricing schemes for their services though associated with certain drawbacks (Kumar et al., 2017). A simple example better illustrates this; suppose a provider offers its computing resources on fixed price. At any time, these resources may be over-utilised or under-utilised but since the price for the resources is fixed, it will not be able to affect its monetary gain/loss and resource utilisation. Also, in fixed pricing, provider is not able to utilise demand flexibility of customer as well, because customer does not get any incentive to expose its demand flexibility (Ishakian et al., 2012). In recent times, providers started offering computing resources using dynamic pricing schemes (AWS, 2016; Nash, 2015). In dynamic pricing, the prices of the resources vary depending upon their usage and availability. The providers start offering resources at low prices in order to attract more customers when utilisation of the resources gets low. In case of high resource utilisation, providers increase the price not only to control the upcoming requests but also to earn better. So, dynamic pricing is beneficial for the provider as well as the customer; provider is able to maximise its revenue even with excess capacity of resources and customer is able to avail services with minimum price. For example, Amazon EC2 (Amazon, 2010) provides on-demand instances and reserved instances with associated prices that come under fixed pricing scheme. In addition, it also provides spot instances under dynamic pricing schemes. The spot instances (AWS, 2016), introduced by Amazon, allow customers to rent unused computing instances at very low prices. In spot pricing (an auction mechanism), user has to bid maximum price that he/she is willing to pay for renting the unused computing instances. Meanwhile, if market price (bid) for the rented resource becomes more than user’s bid, resources will be taken back from the current user. Though it reflects a weakness of the spot pricing, nonetheless spot pricing is very popular in the Cloud market and so is the dynamic price.

Since providers offer variety of computing resources to different types of customers, they must float varying and attractive pricing schemes giving users the liberty to choose one according to their budget and resource requirement. A price model is attractive from the customers’ point of view, if it serves them effectively in least price. From providers’ perspective, it is effective if it earns better profit with increased resource utilisation. By the strategic planning, a provider can keep the reasonable cost of its services in order to attract good number of customers which would be appropriate for long term benefits. For such a conflicting environment, negotiation can be a better approach which is observed as most flexible approach for procurement in the field of e-commerce (Lomuscio et al., 2003). In e-commerce, artificial intelligence helped to create automated negotiation in which software agents participate on behalf of human agents (Lomuscio et al., 2003; More et al., 2014).

Negotiation is a process to settle the differences between provider and customer. Negotiation improves relationship among providers and customers as it reaches to a satisfactory and a long term beneficial agreement on pricing. Both providers and customers can decide on some price range i.e. a range with best price and worst price. If user’s range and provider’s range intersects, negotiation is settled and closed. This work considers a Cloud market, a place that brings all Cloud customers and Cloud providers at one place. Here, providers advertise their offers for the computing resources and gives customer a leverage to choose the required services at competitive price. Though current Cloud market lacks such implementation, this is an innovative and upcoming proposal and has the potential for its possible implementation. Some proposed works support the idea of Cloud market (Garg et al., 2013a; Altmann et al., 2008; Neumann et al., 2007) and their practical approach. Its popularity with intensive research may make it feasible for real Cloud implementation in future. Though, there is an incompatibility issue among cloud providers, an open standard will provide interoperability in Cloud. Cloud Computing Interoperability Forum (CCIF), DMTF Cloud Standards Incubator, Open Cloud Manifesto etc. are some organisations that are trying to make some open standards. Open Virtualisation Format (OVF), supported by few Cloud vendors, provides facility to switch from one vendor to another. Interoperability is mandatory for Cloud prevalence.

Cloud provides various types of resources such as storage, computing power, virtual network etc. Cloud service providers may offer these resources individually or in the form of bundle designed for specific type of application. Amazon provides S3 storage services besides virtual machines of different configuration called as instance. These are bundles of computing resources, e.g., R3 instances provided by Amazon are suitable for memory intensive applications and r3.large, a model of R3, is a combination of two vCPU, 3.75 GB Memory and 2×16 GB SSD Storage (Amazon, 2010). This work assumes that provider offers bundles or instances and user negotiates on price to avail such instances. Instance, bundle and virtual machine are used.
interchangeably in this work. Since negotiation between provider and customer may take some time, a time limit is fixed expiring which both provider and customer may renegotiate or may proceed to some other party.

Proposed negotiation algorithm, in this work, is inspired by Lai et al. (2006) that treats provider and consumer as two negotiating agents. One agent gives offer to the other agent in a given time period. If second agent agrees with the offer, it is accepted. Otherwise it is rejected. If rejected, second agent generates counter offer in the next time period based on a heuristic which is close enough to first agent’s offer. In a given time, if any agent gets satisfied, agreement is done otherwise negotiation is terminated. Though, some works exist in the literature of cloud related to negotiation (discussed in detail in related work section), to the authors’ best knowledge there is no fruitful research on the price negotiation between Cloud provider and customer that considers pricing based on physical resource utilisation as well as behaviour of both provider and customer. Thus, a huge possibility of pricing improvement by negotiation in Cloud computing exists. The main contributions, of this work, are as follows:

- Negotiation based pricing scheme, a practical approach as it improves the relationship between provider and customer to meet out the differences, is designed for implementation of dynamic pricing in cloud.
- Since both provider and customer both are self-interested agents, behaviour of these agents is considered in negotiation for decision making.
- This work considers price as well as non-price attributes to define utility of agents. Providers’ utility introduces fairness by giving opportunity to customers who could succeed less number of times. This, in turn, maintains the competition in the market.
- Resource pricing is done on the basis of utilisation of the resources leading to maximum resource utilisation. This improves the revenue of the provider whereas customers get the resources within their budget using negotiation.
- Proposed model is analysed through simulation on various settings. Using simulation, the causal effect of the behaviour of provider is studied depending on the behaviour of customer.

The rest of the work is structured as follows. Section 2 defines the problem addressed in this work. The proposed model of negotiation mechanism is presented in Section 3. Subsection 3.2 formulates utilisation of provider and customer which includes value of price and opinion of one agent for opponent. Negotiation algorithm is described in subsection 3.3. Pricing of resources, on the basis of utilisation of total resources, is formulated in subsection 3.4. Section 4 studies the performance of the proposed model on various system parameters. The proposed work is compared and contrasted with some recent related models in Section 5. Finally, Section 6 concludes the work.

2 The problem

Negotiation is a common technique used to resolve conflicts among the parties and reaching an agreement. Negotiation generally involves two parties, called negotiation agents, with common interest and conflicts. Effective auto negotiation techniques equip both the agents, in this case cloud providers and customers, to respond adaptively according to the behaviour of negotiating partners and emerging negotiation scenario. Agents’ unwillingness to reveal their negotiation strategy, in order to safeguard them from exploitation, complicates auto negotiation. This work considers bilateral negotiation i.e. negotiation between two agents; Cloud Provider P and Cloud Customer C. Both the agents negotiate over resource pricing by generating counter proposals in terms of price concession. Cloud customers cannot wait indefinitely for an agreement as resources are needed to run their applications. Thus, a time limit is introduced in the negotiation expiring when negotiation is terminated. Literature contains some works on negotiation (Lai et al., 2006; Ehtamo et al., 1999; Faratin et al., 1998) though not for the Cloud resources. Some works consider that negotiation strategies are known to all agents (cooperative agents). Some consider centralised negotiation in which agents are not self-interested. In Cloud computing, some research works propose negotiation algorithms considering agents to be co-operative. Such relevant works are discussed in Section 5. In a dynamic pricing scenario of Cloud computing, provider and customer are self-interested and non-cooperative i.e. they do not want to share their utility information with each other. Cloud provider and customer being self-interested and non-cooperative makes it possible to design a heuristic to define counter proposal. This work proposes a negotiation model for dynamic pricing in Cloud computing using a heuristic for counter offer and also defines the pricing of resources at the provider side by considering the resource utilisation as discussed in Section 1.

3 The proposed model

This work is motivated by a decentralised model for multi-attribute negotiations (Lai et al., 2006). The model is decentralised as the agents are self-interested and do not share their utility information. The model is based on an alternating offer, where in each period the proposing agent makes a limited number of offers. The responding agent can either choose the best offer or can reject it. In case of rejection, agents exchange their roles and the negotiation proceeds for the next time period. To make counter offers, an agent first chooses on an indifference curve (or surface), the offer that is closest to the best offer made by the opponent in the previous period, and then with this offer as the seed, chooses several other offers randomly in a specified neighbourhood of the seed offer. In Lai et al. (2006) the concept of negotiation is given but is not designed for Cloud. It also does not consider the resource utilisation in the price determination.
An agent (provider and customer) $i$ prepares a counter offer in time period $t$ closest to its opponent’s offer in time period $t-1$, where $i \in \{P, C\}$. To define this heuristic, a time dependent strategy (Faratin et al., 1998) as given in equation (1), is used

$$U_i(t) = 1-(1-\alpha_i)^{\left(\frac{t}{T_i}\right)}^{\beta_i}$$

where $U_i(t)$ is a utility that agent $i$ expects in time period $t$. If this utility is greater than the utility of opponent’s offer in period $t-1$, agent $i$ generates counter offer i.e. offered price by opponent is not acceptable to agent $i$ . Otherwise, it accepts the opponent’s offer. $\alpha_i$ is a reservation utility that agent $i$ can accept for each negotiation period, $T_i$ is maximum negotiation period of agent $i$ and $\beta_i$ represents the behaviour of agent $i$. Two types of behaviour of an agent i.e. boulware behaviour and conceder behaviour, as given in Faratin et al. (1998), are considered in this proposed work. Agent with boulware behaviour moves slowly to its reservation value whereas agent with conceder behaviour moves fast to its reservation value. Small value of $\beta_i$ represents agent’s boulware behaviour i.e. agent takes shorter steps until in proximity of deadline whereas large value of $\beta_i$ represents agent’s conceder behaviour i.e. agent takes longer steps targeting an early agreement. Through this whole work agents are participants in the negotiation i.e. agent refers both Cloud provider and Cloud customer. Above heuristic is defined for both customer and provider. Customer is opponent for provider and vice-versa.

### 3.2 Utility functions

In negotiation, upon receiving a proposal from other agent, an agent evaluates the proposal. Utility, a concept from economics, represents the degree of satisfaction. In this work, each agent (provider and customer) defines and evaluates the proposal through a utility function. Suppose for the two proposals $a$ and $b$, their utilities for an agent are $U(a)$ and $U(b)$ respectively. If $U(a) > U(b)$, then proposal $a$ is better than proposal $b$. Since in this work, price has to be negotiated, proposal will be in form of certain value of price. Provider and customer both define a range of price, i.e., best price and worst price. At customer side, least price is best price as a customer always wants to pay minimum while worst price is maximum price. For a provider, best price is maximal price a provider can earn and worst price is minimum price a provider is willing to charge. A provider generates value of best price and worst price based on the utilisation of resources discussed in Section 3.4. Let $BP_p$ be the best price and $WP_p$ be the worst price offered by an agent $i$, where $i \in \{provider(P), customer(C)\}$.

Generally, range of utility is between 0 and 1 so utility of best price and worst price always be 1 and 0 respectively as depicted in equations (2) and (3).

$$u(WP) = 0 \quad \text{for each } i$$

$$u(BP) = 1 \quad \text{for each } i$$

Though in economics and computer science, various utility functions have been defined, for simplicity, a linear and monotonic utility function is considered in this work. Utility function of a particular value of price $p$ for agent $i$ is defined in equation (4).

$$u_i(p) = \frac{WP_p - p}{WP_p - BP_p} \quad \text{and } u_i(p) \in [0,1]$$

For customer $C$, best price is less than worst price i.e. $BP_C < WP_C$, utility of customer for price $p$ can be written using equation (5).

$$u_c(p) = \frac{WP_c - p}{WP_c - BP_c}, \quad BP_c \leq p \leq WP_c$$

For provider $P$, best price is greater than worst price i.e. $BP_p > WP_p$, utility of provider for price $p$ can be written using equation (6).

$$u_p(p) = \frac{p - WP_p}{BP_p - WP_p}, \quad WP_p \leq p \leq BP_p$$

The above equations for utility are defined on the basis of price only. In Cloud market, a Cloud provider foresees long term benefit and so the provider must not focus on price only. It must give an opportunity to low budget new customers to win their trust besides keeping in mind the importance of loyal customers. On the other hand, customer looks for services from a trustable provider that has high reputation. Thus, the utility of an agent $i$ can be extended by including the opinion of the agent $i$ about the opponent agent. It may encourage agents to offer their true price range for the resources.

The extended utility $U_i(p)$, for agent $i$, is defined as given in equation (7).

$$U_i(p) = \omega_1 \times u_i(p) + \omega_2 \times of_i$$

where $of_i$ is opinion function of agent $i$, and $\omega_1$ and $\omega_2$ are weights of utility of price and opinion function of agent $i$ respectively such that $\omega_1 + \omega_2 = 1$, and $i \in \{P, C\}$.

In the competitive Cloud market, provider needs to retain existing customers as well as attract new customers. Provider may design some mechanisms to attract new customers, but these mechanisms must not neglect importance of existing loyal customers for long term benefits. That is why provider’s opinion, about the customer, can be quantified based on the loyalty of the customer and fairness in the market. Fairness here means provider may give some priority to the new
Customers or those that are getting VMs less frequently. This maintains the competition in the market (Baranwal and Vidyarthi, 2015). Detail of fairness can be found in Baranwal and Vidyarthi (2015). The opinion function of the provider of $p$ is defined as in equation (8).

$$\text{of}_p = w_1 \times \text{FE} + w_2 \times \text{LoC}$$  \hspace{1cm} (8)

where $\text{FE}$ is the function for fairness. Equity in resource allocation is defined in equation (9). $\text{LoC}$ is loyalty of customer in equation (10). $w_1$ and $w_2$, the weights of $\text{FE}$ and $\text{LoC}$ functions, are used to give priority to fairness and loyalty of customer respectively.

To ensure fairness and equity in the resource allocation, it is assumed that provider gives priority to those customers who requested for VMs earlier but could succeed fewer times. This facilitates those customers who may leave the system soon because of low budget. Such concept of priority has successfully been used in Lemaître et al. (2003), Murillo et al. (2011) and Pla et al. (2014). Thus, $\text{FE}$ for customer $j$ is defined as given in equation (9).

$$\text{FE} = 1 - \left( \frac{1}{1 + nt_{\text{gotVM}}} \right) \left( 1 + nt_{\text{triedVM}} \right)$$  \hspace{1cm} (9)

where $nt_{\text{gotVM}}$ is number of times customer $j$ gets VM and $nt_{\text{triedVM}}$ is number of times customer $j$ tried to get VM, i.e., participated in negotiation.

From equation (9), it can be observed that a customer who attempts but does not get the resources even once has the highest priority. A customer who attempted and succeeded for the resources always will have least priority. Priority of a new customer, i.e., one in negotiation process for the first time, also has least priority as it is quite unlikely that a new customer will leave the system at its first attempt.

Relationship between provider and customer is an important factor in negotiation. Since Cloud is a market of computing resources, pricing strategy should establish a pool of profitable and loyal customers in the long run like any other market scenario. Customers try to churn to some other service provider if not satisfied with the services. Since cloud resources are available online and can be obtained very easily, increased satisfaction may not lead to increased loyalty (Shankar et al., 2003). Now one can understand importance of a loyal customer. Loyalty of customer can be accomplished by deciding on why a customer will churn to some other provider in direct marketing. Recency, Frequency and Monetary (RFM) techniques have been used to observe the behaviour of customers (Shih and Liu, 2003). Unfortunately, churning identification, prediction and prevention is a big and new research area in Cloud (Xie et al., 2009; Tsai and Lu, 2009; Baranwal and Vidyarthi, 2016a). Therefore, without diving in much detail here, it is assumed that customers are ranked on the basis of loyalty ($\text{LoC}$) which is defined by equation (10). 

$$\text{LoC} = 1 - \left( \frac{\text{rank of customer}}{10} \right)$$  \hspace{1cm} (10)

where \text{rank of customer} lies between 1 to 10, 1 being the highest rank.

Customers try to get services from the best providers based on their requirements, level of other customers’ satisfaction, trust and reputation of Cloud provider in the market etc. Basically, this is provider selection problem and has been well deliberated (Garg et al., 2013b; Li et al., 2010; Baranwal and Vidyarthi, 2014). This work considers that customers are interested to extend their utility by incorporating their opinion about the provider without going in details of the best provider selection. Since creating a reputation model, trust model etc. to define opinion function in itself is a research challenge, these are out of scope of this proposed work. In the proposed work, ranking of the provider is considered in obtaining the opinion of the customer. Opinion function of customer of $c$ is defined as given in equation (11).

$$\text{of}_c = 1 - \left( \frac{\text{rank of provider}}{10} \right)$$  \hspace{1cm} (11)

where rank of provider lies between 1 and 10 with 1 being the highest rank.

\subsection{3.3 Negotiation algorithm}

Each agent $i$ alternatively offers a price at time instances $t = 0, 1, 2, \ldots$ until agents reach on an agreement or time period exceeds the specified maximum negotiation period where $i \in (P, C)$. Let $p_{j}^{t-1}$ be the price of the resource offered by agent $j$ at time $t-1$ which is received by agent $i$ at time $t$ where $i$ can be provider $P$ or customer $C$ and $i \neq j$.

$U_i\left(p_{j}^{t-1}\right)$ is an extended utility of agent $i$ for $p_{j}^{t-1}$ as given in equation (7). The following algorithm performs negotiation decision at time $t$.

\begin{itemize}
  \item \textbf{Step 1:} if $(t > T_i)$
  \item \textbf{Step 2:} Agent $i$ will withdraw
  \item \textbf{Step 3:} exit
  \item \textbf{Step 4:} end if
  \item \textbf{Step 5:} calculate heuristic $U_i\left(t\right)$
  \item \textbf{Step 6:} if $\left(U_i\left(p_{j}^{t-1}\right) \geq U_i\left(t\right)\right)$
  \item \textbf{Step 7:} Agreement is done on price $p_{j}^{t-1}$
  \item \textbf{Step 8:} exit
  \item \textbf{Step 9:} end if
  \item \textbf{Step 10:} calculate counter offer
  \item \textbf{Step 11:} $p_i = \arg \min_{s} \left\| p - p_{j}^{t-1} \right\|$
  \item \textbf{Step 12:} such that $\left\| \right\|$ is euclidian distance
  \item \textbf{Step 13:} offer $p_i$ to agent $j$
  \item \textbf{Step 14:} exit
\end{itemize}
Initially customer submits its requirement to the provider. An agent, provider or customer, can start negotiation by initiating an offer. Suppose customer is the opponent and at time \( t = 1 \) the customer had given its offer i.e. \( p_c^{1-1} \). Now provider at time \( t \) checks if the negotiation period has been expired, i.e., \( t > T_p \), provider leaves the negotiation. If no, provider calculates \( U_p(t) \) in step 5 using heuristic defined in equation (1) i.e. provider’s expectation in the current round at time \( t \). In step 6, provider calculates the utility of the offer by customer i.e. \( U_p(p_c^{1-1}) \) using equation (7). If this calculated utility is as per expectation of provider, agreement is reached in step 7 and negotiation is done. Otherwise provider starts calculating the counter offer. In step 11, provider explores a price \( p'_p \) such that the calculated utility using equation (7) should be equal to what provider was expecting from customer at time \( t \) i.e. \( U_p(p'_p) = U_p(t) \). Now provider gives this calculated price \( p'_p \) as an offer to customer. Now the role of customer and provider is reversed i.e. provider acts as opponent and customer will observe the given offer by provider.

### 3.4 Resource pricing based on resource utilisation

Cloud provider creates VMs (workload) on physical machines using hypervisor. In this work, utilisation of resources is considered as overall utilisation of all physical machines of the Cloud provider. In order to maximise revenue and resource utilisation, Cloud provider must define pricing function to estimate the utilisation of the resources. For this, the provider defines a range of utilisation of resource with its base price. If the resource utilisation is below this range, provider reduces the price of resources to attract more customers. If utilisation of resource is above the range, the provider increases the price of the resources to control upcoming requests. Increasing number of customers in case of under-utilisation of resources and decreasing number of customers in case of over-utilisation makes efficient resource utilisation. Since in this work negotiation algorithm is used, provider defines a range of price for the resource at a particular value of utilisation of resource i.e. a minimum price of resource which gives an acceptable profit to provider and a maximum value of price, so that customer can negotiate in this range.

Let \([RU_{min}, RU_{max}]\) be the range of utilisation of resources for which provider charges a base price, where \(RU_{min}\) is minimum utilisation of resource and \(RU_{max}\) is maximum utilisation of resource. Let \(P_{best}^b\) be the minimum base price, \(P_{best}^{UL}(RU)\) be the price function when utilisation \(RU\) is less than \(RU_{max}\), \(P_{best}^{UL}(RU)\) be the price function when utilisation is higher than \(RU_{max}\). Inspired by Zheng and Veenavalli (2012), pricing function to decide best price with respect to utilisation of resources can be written as in equation (12).

\[
P_{best}(RU) = \begin{cases} 
P_{best}^{UL}(RU) & 0 \leq RU < RU_{min} \\
P_{best}^{UL}(RU) & RU_{min} \leq RU \leq RU_{high} \\
P_{best}^{UL}(RU) & RU_{high} < RU \leq 1 
\end{cases}
\]

Here, it is assumed that \(P_{best}^{UL}(RU)\) and \(P_{best}^{UL}(RU)\) are functions of base price and utilisation of resource respectively because provider can use any appropriate method to decide base price \((P_{best}^b)\). When utilisation is low, exponential form of \(P_{best}^{UL}(RU)\) will decrease base price rapidly and linear function will decrease base price constantly. So a polynomial function is considered to define \(P_{best}^{UL}(RU)\). To define \(P_{best}^{UL}(RU)\), it is assumed that when all the resources are utilised, price tends to infinity to discourage customers strongly. To consider the above aspect, logarithmic barrier function is suitable for \(P_{best}^{UL}(RU)\). \(P_{best}^{UL}(RU)\) and \(P_{best}^{UL}(RU)\) are defined as given in equations (13) and (14) respectively.

\[
P_{best}^{UL}(RU) = P_{best}^{b} \left(1 - a \left(RU_{min} - RU\right)^{n}\right) \quad (13)
\]

\[
P_{best}^{UL}(RU) = P_{best}^{b} \left(1 + \left(1/\left(1\right)\right) \left(log\left(1 - RU\right)\right)\right) \quad (14)
\]

where \(a\), \(n\) and \(t\) are constants and act as tunable parameters for the provider.

For a certain value of resource utilisation \((RU)\), maximum price \(P_{worst}(RU)\) i.e. worst price is defined by equation (15).

\[
P_{worst}(RU) = P_{worst}(RU) + f \quad (15)
\]

where \(f\) may be any function to decide worst price at particular value of utilisation of resource. For simplicity, it is assumed that \(f\) is a constant. The whole pricing scenario is given in Figure 1.

![Figure 1: Pricing function for utilisation of resource](image)
4 Performance evaluation

For the performance study of the proposed model, a simulated Cloud environment is created because it allows the evaluation of the models more accurately in a repeatable and controllable environment by generating good amount of data using limited resource and limited time. Currently, no real cloud market traces are available to extract data for experimental purposes. So, parameters of the simulation described are generated randomly. It has been our attempt to cover all possibilities in the simulated environment close to real Cloud.

As mentioned in Section 3.4, utilisation of physical resources decides the price. The arrival rate ($\lambda$) of users’ requests for VM follows Poisson distribution similar to McManus et al. (2004) and Wolff (1982). Total 1000 VMs are created and result in each experiment is taken as average of 100 generated results for more precision. Performance metric, in each experiment, are revenue of Cloud provider ($R$), total utility ($TU$) and speed of negotiation ($SoN$). To calculate profit, more number of parameters are needed but base price is defined in such a way that always ensures an acceptable profit for the provider. Thus, if revenue of Cloud provider ($R$) increases, profit will definitely increase.

When negotiation between provider and customer is successful for a particular price $p$ of the resource, total utility (equation 16) is calculated as average of extended utility of both provider and customer for that price.

$$TU = \frac{(U_p(p) + U_c(p))}{2}$$

(16)

Speed of negotiation means how fast a successful negotiation is reached. It can be defined (equation 17) as a ratio of number of rounds ($nr$) in successful negotiation and minimum of negotiation time limit of agents ($\min(T_p, T_c)$).

$$SoN = 1 - (nr / \min(T_p, T_c))$$

(17)

Higher the value of speed of negotiation i.e. $SoN \to 1$, higher the speed is.

Extended utility is defined in Section 3.2 by considering price and opinion of agents, but for clear observation of performance measures and behaviour of agents, the values $w_1 = 1$ and $w_2 = 0$ are considered for both provider $P$ and customer $C$. Experiments are performed to explore different perspective of the proposed model by changing the value of the system parameter. Since all system parameters do not change in every experiment, the common values of system parameters are given below and varying value of system parameters are mentioned in each experiment.

Resource utilisation limit $\left[RU_{max}, RU_{min}\right] = [0.3, 0.7]$ as in Zheng and Veeravalli (2012); Base price of VM is $P^{base} = \$0.5$ so that price of VM changes (increases or decreases) according to the utilisation of resource. Though $f = 0.3$ is taken to quantify worst price but behaviour of system is observed also for various value of $f$.

Values of some of the parameters, randomly drawn using uniform distribution to ensure dynamic behaviour and effectiveness of proposed model, are as follows.

As discussed before, worst price of customer should be greater than its best price, best price of customer is $BP_c = U(\$0.4, \$0.5)$ where $U$ represents uniform distribution and worst price of customer is $WP_c = U(\$0.8, \$1.0)$; $\beta_i$ represents behaviour of agent $i$, to generate customer and provider of both boulware behaviour and conceder behaviour $\beta_c = U(-1, 1); \beta_p = U(-1, 1);$ Reservation utilities are $\alpha_p = U(0.15, 0.25)$ and $\alpha_c = U(0.15, 0.25)$ and negotiation times are $T_p = U(30, 70)$ and $T_c = U(30, 70)$, to show the presence of customer and provider respectively with various needs.

4.1 Experiment 1: fixing the values of $a, n, t$

Best price $P_{best}(RU)$ for provider depends on utilisation of resource. For a particular value of utilisation of resource, minimum and maximum value of $P_{best}(RU)$ depends on three parameters ($a, n, t$) as used in equations (12), (13) and (14). To decide the value of these three parameters, value of one parameter is varied keeping the other two fixed. First, the value of $n$ and $t$ are fixed as $\sqrt{3}$ and 4 respectively with varying $a$. Results obtained are shown in Figure 2. As the value of $a$ increases, minimum value of $P_{best}(RU)$ increases while maximum value is fixed. Next, the value of $a$ and $t$ are fixed as 2 and 4 respectively with varying $n$. Results obtained from this experiment are shown in Figure 3 which shows that as the value of $n$ increases, minimum value of $P_{best}(RU)$ decreases while maximum value is fixed. Finally, the values of $a$ and $n$ are fixed to 2 and $\sqrt{3}$ respectively with varying $t$. Results obtained from this experiment are shown in Figure 4 depicting that as the value of $t$ increases, maximum value of $P_{best}(RU)$ decreases while minimum value is fixed. For a real Cloud scenario, minimum value of $P_{best}(RU)$ should not be very less in comparison to base price $P^{base}$ and maximum value of $P_{best}(RU)$ should not be very high. So, with the help of Figures 2, 3 and 4, the values of $a$, $n$, $t$ may be fixed as $a = 2, n = \sqrt{3}$ and $t = 4$ to generate a simulated environment close to a real one.

Figure 2 Range of best price for provider w.r.t $a$ ($n$ and $t$ fixed)
4.2 Experiment 2: different behaviour of provider

In this experiment, variations in revenue, total utility and speed of negotiation are observed when $P_{\beta}$ varies. This variation is also observed with different arrival rates. Results obtained are shown in Figures 5, 6 and 7. It can be observed that as the arrival rate of requests increases, the revenue of provider is increased while total utility and Speed of Negotiation (SoN) decreases. It is because, if the number of requests is more, all VMs are quickly allocated. But after a significant increment in arrival rate (after $\lambda = 50$), all the three performance measures get saturated as requests are increasing for same fixed resources.

For different arrival rates, the optimal value of $P_{\beta}$ is different. If $P_{\beta}$ is small i.e. attitude of provider is bouluwar, most of the negotiations are unsuccessful because provider is taking short steps in negotiation. It means provider is decreasing the price very slowly and customers only with higher budget are satisfied. If provider’s attitude is conceder i.e. $P_{\beta}$ is large, customers with fewer budgets also get satisfied and resources are allocated quickly. This can be observed from Figure 7 i.e. SoN is increasing as $P_{\beta}$ is increasing. Figure 7 depicts that for large values of $P_{\beta}$, customer’s utility increases but utility of provider decreases very rapidly resulting in decrease of total utility.

Thus, it can be concluded that the optimal behaviour of provider depends on the arrival rate. If arrival rate is high, provider’s attitude should be bouluwar, and attitude should be conceder in case of low arrival rate. But, in any case, provider does neither show its attitude exceedingly bouluwar nor exceedingly conceder.

4.3 Experiment 3: optimal value of $P_{\beta}$

In this experiment for different arrival rate of requests, optimal value of $P_{\beta}$ is observed for the customer with different attitude. We fixed value of $P_{\beta}$ between -0.9 and 0.9 to show bouluwar attitude and conceder attitude of customer respectively. Optimal value of $P_{\beta}$ is identified from the provider’s revenue curve. Obtained result, from this experiment, is shown in Figure 8. It depicts that optimal
value of $\beta_p$ decreases as arrival rate increases. Also, for boulware customer, provider’s behaviour should be conceder comparatively and vice versa.

**Figure 8** Optimal $\beta_p$ for different arrival rates for boulware and conceder customers

![Optimal $\beta_p$ for different arrival rates for boulware and conceder customers](image)

### 4.4 Experiment 4: varied behaviour of provider and customer

Since both provider and customer can act as boulware and conceder, in this experiment we observed performance for different behaviour of provider and customer i.e. boulware provider & boulware customer (PBCB); boulware provider & conceder customer (PBCC); conceder provider & boulware customer (PCCB); conceder provider & conceder customer (PCCC). Parameters for boulware behaviour and conceder behaviour are given in Table 1.

<table>
<thead>
<tr>
<th></th>
<th>$T_i$</th>
<th>$\alpha_i$</th>
<th>$\beta_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boulware</td>
<td>U(40,80)</td>
<td>U(0.2,0.3)</td>
<td>U(-1,-0.4)</td>
</tr>
<tr>
<td>Conceder</td>
<td>U(20,60)</td>
<td>U(0.1,0.2)</td>
<td>U(0.4,1)</td>
</tr>
</tbody>
</table>

**Table 1** Input parameters for behaviour

Results from this experiment are shown in Figures 9, 10 and 11. Revenue of provider increases while total utility and speed of negotiation decreases with arrival rate for all mixed behaviour. Figure 9 depicts that when customer is conceder, provider should act as boulware while Figures 8 and 9 both conclude that when customer is boulware, provider should also be boulware. For maximum revenue of provider, if value of $\beta_c$ is low, value of $\beta_p$ should also be low but $\beta_p$ should be more than $\beta_c$. Figure 11 confirms that in case of PCCC, SoN is highest because both provider and customer are in hurry during negotiation and in case of PBCB, SoN is lowest because both are taking short steps during negotiation.

### 4.5 Experiment 5: varied budget of customer

In previous experiment, it is observed that provider get highest income when its attitude is boulware and customer’s attitude is
This experiment is performed on the same behaviour of customer and provider (i.e., system parameters are same as in PBCC in previous experiment) to observe the performance measures when the customer increases its budget range \([BP_C, WP_C] = [U(0.30,0.50), U(0.70,0.90)], [U(0.40,0.60), U(0.80,1.00)], [U(0.50,0.70), U(0.90,1.10)], [U(0.60,0.80), U(1.00,1.20)]\). Figures 12, 13, and 14 are drawn from results of this experiment and it clearly shows that on the increase in the customers’ budget, all performance measures are increasing.

Figure 13 Total utility on arrival rate for different budget range of customer

Figure 14 Speed of negotiation on arrival rate for different budget range of customer

4.6 Experiment 6: varied value of \(f\)

Similar to previous experiment, performance measures are observed for varying values of \(f\) (i.e., for different range of provider’s offered price). Other system parameters are same as in PBCC in experiment 4 in Section 4.4. Since the base price is fixed, when the value of \(f\) increases, acceptance of customers with higher budget is more likely which in turn increases the revenue of the provider. Though after a significant increment of \(f\), revenue of provider stagnates because customers cannot afford. Figures 16 and 17 confirm that higher price range will reduce utility and speed of negotiation.

5 Related work

As Cloud market is growing, many providers are coming up with attractive offers to their customers which, in turn, makes Cloud pricing a challenging issue. In the recent past, some research work is done to handle this challenge. This section elaborates the current research works, related to pricing issues, along with its comparison and contrast with the proposed work. Fox et al. (2009) identifies six cost factors. It compares Cloud with in-housing infrastructure using trade-off formula.
and finds that migration to Cloud is beneficial. It gives an idea about Cloud cost but does not supply any cost model. Similarly, Fox et al. (2009) and Maurer et al. (2012) consider 11 cost factors to cover all aspects of Cloud for economic comparison. Chandrakant and Amip (2005) calculate the cost, for running a data centre, considering three parameters: power, cooling and space. It estimates the cost for all three parameters one by one then adds these costs. Though Chandrakant and Amip (2005) give an idea for the cost model, they do not provide details to sell the Cloud resources as a service. Since computing resources in grid market and Cloud market are similar to the resources in business market, some economics model such as commodity model, auction model (Baranwal and Vidyarthi, 2016b), bargaining model (Zheng et al., 2010) and game theoretic approach (Künsemöller and Karl, 2014) are widely used for pricing in cloud. We have already discussed that static (fixed) pricing, dynamic pricing and spot pricing (an auction mechanism) are modern practically implemented pricing schemes.

Fixed pricing of Cloud resources has several drawbacks. Because of over-utilisation and under-utilisation of resource, fixed pricing is not efficient and generally does not reflect equilibrium price for resource which may cause low revenue for the provider (Zaman and Grosu, 2013). In fixed pricing, if a user realises that cost of resource is higher than expected, he/she does not have option for re-adjustment of price. In comparison to static pricing, the proposed work decides the price of resources on the basis of resource utilisation and customers have the option for negotiation. Thus, customers are satisfied to avail the services on minimum price.

In Cloud computing, some research works have modelled pricing using game theory, treating provider and customers as a player (Hadji et al., 2011; Künsemöller and Karl, 2014; Zheng et al., 2010). In cooperative game, both provider and customer know about each other’s payoff function. So, to establish a Nash equilibrium is easy in cooperative game. But in non-cooperative game, neither player knows the other player’s payoff/utility information, a scenario similar to Cloud. Some works consider the scenario of resource pricing as a non-cooperative game. In Hadji et al. (2011) Cloud provider tries to find optimal price and optimal user demands. Hadji et al. (2011) models interaction between Cloud provider and consumer as a non-cooperative and constrained game. Game is based on Stackelberg game and Stackelberg/Nash equilibrium. But users adjust their demands according to the proposed price to maximise their utility. In current competitive Cloud market, it is less likely that customers will adjust their demand as in Hadji et al. (2011). So in the proposed work, unlike Hadji et al. (2011), customers need not to adjust their demand rather adjust price of resources by negotiation. Künsemöller and Karl (2014) created a game theoretic model for predicting the future prices of on-demand resources in which there are two players, a provider and a customer, where provider tries to maximise its profit and customer tries to minimise its expenses. In Künsemöller and Karl (2014), provider offers a fixed price to customer and the model assumes that provider has strong knowledge about customer’s load distribution unlike our proposed model in which provider does not require the knowledge of load distribution. Zheng et al. (2010) considers web-service negotiation between provider and customer as a bargaining game. It assumes that each player has knowledge about taste and preference of the other player. As discussed before, provider and customer do not want to reveal their strategy to each other to prevent themselves from exploitation. The proposed model does not reveal one’s strategy to another i.e. Cloud provider and customer both are self-interested and non-cooperative.

Ren et al. (2016) considered negotiation for web-based service environments. But they considered multiple negotiations with different goals at the same time and proposed graph based approach to settle the negotiation which considers interdependency among multiple negotiations as well. Since such type of multiple negotiations is processed concurrently, this is called concurrent negotiation. Like Ren et al. (2016), Wu et al. (2013) proposed an automated negotiation framework in which SaaS provider negotiate with multiple providers to achieve different objectives of customers. While this proposed work is bilateral negotiation in the cloud environment.

Bidding or Auction, a concept of economics, is also a well-known method for pricing of Cloud resources. Spot pricing used by Amazon is popular in Cloud research (Wee, 2011; Ben-Yehuda et al., 2011; Yi et al., 2010). Auction has two types; one-to-many and many-to-many. Dutch auction, English auction, Second-pricing auction are some famous one-to-many auctions in which an agent initiates the auction and other agents bid for the resources. Second-pricing auction mechanism is proposed in Lin et al. (2010) which uses the concept of marginal bid to decide the price of Cloud resource where marginal bid is highest bid among all unsuccessful bids. In case of large number of users and resources, Lin et al. (2010) overcome strategic deviation and revenue inferiority. Zaman and Grosu (2013) used combinatorial one-to-many auction for VM allocation and pricing. It proposed two mechanisms for auction; CA-LP (Combinatorial Auction – Linear Programming) and CA-GREEDY (Combinatorial Auction - Greedy). Zaman and Grosu (2013) compared CA-LP and CA-GREEDY with fixed pricing scheme and found that proposed schemes are better choice. The drawback of one-to-many auction is monopoly of one side. Double auction is widely accepted many-to-many auction that prevents monopoly and is more efficient than one-to-many auction as in double auction both sides submit their bids (Kumar et al., 2017). To form a standard for interoperability, a challenging issue in Cloud, Shang et al. (2010) first proposed a framework for formation of global Cloud market and then proposed a knowledge-based continuous double auction (CDA) model using a learning algorithm (based on historical trading information) to determine the price of cloud resources. Samimi et al. (2015) proposed a Combinatorial Double Auction Resource Allocation (CDARA) model. CDARA considered two models proposed in Zaman and Grosu (2013) and Li et al. (2009) and efficiently allocated the Cloud resources. Involved economic efficiency and incentive compatibility were two criteria used by Samimi et
al. (2015) for evaluation of CDARA. Experiments on CDARA showed that model is efficient and cost effective for user as well as provider. Generally, auction is organised by a broker or by Cloud provider. Auction for pricing and resource allocation in Cloud computing is efficiently used but auction is a relatively public event and customer may avoid this because of fear of bad reputation in market. The other drawback of auction is that conditional customer feels excluded. In the proposed work, there is no involvement of a broker or mediator and customers freely participate in negotiation. Any of the above discussed works do not consider opinion of provider (customer) about customer (provider).

Henzinger et al. (2010) proposed FlexPRICE (Flexible Provisioning of Resources in a Cloud Environment) a flexible framework. In this, customer submits its job to the cloud for execution. Cloud find multiple schedules corresponding to a quote to the user that contains price for job execution and finish time of the job. Then user chooses a particular schedule and cloud run the job according to chosen schedule. Same as Henzinger et al. (2010), in Macias and Guitart (2010), when provider receives a proposal from customer, the provider proposes an offer to the customer. The offer should maximise the provider’s utility with consideration of economic factors, resource availability etc. But Henzinger et al. (2010) and Macias and Guitart (2010) both are basically a unilateral negotiation where provider proposes an offer and customer decide whether to accept or reject. Ranaldo and Zimeo (2015) proposed an approach similar to Macias and Guitart (2010) i.e. maximisation of utility but in bilateral negotiation. Ranaldo and Zimeo (2015) exploits capacity planning to optimise the utility of provider in a more flexible manner. But Ranaldo and Zimeo (2015) does not consider the behaviour of provider or customer in negotiation and does not decide the price of the resources on the basis of resource utilisation.

Zheng et al. (2014) proposed multi-attribute bilateral negotiation mechanism for computing resources in Cloud. Authors considered five quality of service (QoS) metrics and two negotiation strategies: Concession and Tradeoff. In concession strategy, agent reduces its utility to reach an agreement while in tradeoff strategy an agent yields on less important QoS metrics but demands more on relatively more important QoS metrics. Authors suggested that if information is incomplete it is better to adopt concession strategy otherwise tradeoff strategy.

Violation of SLA may occur because of changing workload condition and/or failure of software and hardware components. Inspired by the autonomic computing, Kertesz et al. (2014) provides a self-adaptive architecture for on-demand resource provisioning with SLA. This architecture also includes a negotiation component that can be a good motivation for negotiation based architecture of cloud resources. Dastjerdi and Buyya (2015) extended their work of Dastjerdi and Buyya (2012) to propose a negotiation based model which decides the price on the basis of provider’s resource capabilities. It also considers reliability aspect of providers in negotiation. The reliability of offer is calculated using recorded data from monitoring services.

But Dastjerdi and Buyya (2015) do not consider both customer’s and provider’s boulware and conceder behaviour. Our proposed work not only helps agents in deciding the optimal behaviour in different possible scenario but also finds that arrival rate, budget of customer and the price range of the resources also play an important role for better results in negotiation.

6 Conclusion

In summary, the novelty of the proposed work is negotiation with dynamic pricing for Cloud resources in which provider and customer both are self-interested and non-cooperative. Utility of an agent $i$ is defined on the basis of the price offered by other agent $j$ as well as opinion of $i$ about $j$. Provider defines price of cloud resources on the basis of resource utilisation and provides fairness in resource allocation using its opinion function which gives higher priority to those customers that succeeded less number of times in negotiation. In addition, opinion function of provider considers loyalty of the customer. Since pricing is done on one-to-one interaction, customers feel that they are paying lowest price for a resource. Proposed work also considers the behaviour of both provider and customer. Experiments, carried for the proposed work, help to decide on the optimal behaviour of agents upon various parameters.

To study the performance of the proposed model, simulation is used. Three performance parameters, revenue of provider, total utilisation and speed of negotiation, have been studied. Various parameters (e.g. arrival rate, budget, attitude etc.) are varied and the consequences are investigated. The proposed model helps the Cloud provider to find various trade-offs. Provider sets his/her attitude towards conceder when request arrival rate is low and shifts towards boulware as the request arrival rate increases. When arrival rate is low, provider tries to allocate the resources by early agreement. This way, provider allocates resources at low price margin. When arrival rate is high, provider tries to allocate resources by late agreement, i.e., provider allocates resources to customer at higher price margin. Offering resources at low price during off-peak time (when arrival rate is low) attracts those customers that have no disadvantage requesting resource at off-peak time at the lower price. This facilitates both customer and provider as the customer can obtain lower price while provider can earn during off-peak time. Provider should behave optimally in negotiation. It can be clearly concluded, from the experiments, that when provider’s attitude is extremely boulware, speed of negotiation is much less. In this case, not all available resources are allocated. When provider’s attitude is extremely conceder, resources are allocated at lower price but all the resources are allocated quickly. It can further be concluded from the experiments that optimal attitude of Cloud provider should be comparatively boulware when customer’s attitude is conceder and vice versa, while it should never be extremely boulware or extremely conceder. Although customers with higher budget are beneficial for provider, for enhancing fairness in the allocation of resource, extended utility of provider also
A negotiation based dynamic pricing heuristic in cloud computing

95

considers those customers who have succeeded fewer times in negotiation by giving them higher priority. Since both provider and customer include their opinion of others in utility function, it will encourage them to offer their true price range. Proposed work presents a market oriented tool in which provider decides price of resources on the basis of resource utilisation to deal with dynamic resource demands under various system settings, and customer participates to fulfil its own requirements. Both negotiate with each other to make an agreement.

Acknowledgements

Authors would like to acknowledge UPE-II, JNU and DST-PURSE for the financial support in carrying out this work.

References


