Cloud Customer’s Historical Record Based Resource Pricing

Mohammad Aazam, Eui-Nam Huh, Members, IEEE, Marc St-Hilaire, Senior Member, IEEE, Chung-Horng Lung, Ioannis Lambadaris, Members, IEEE

Abstract — Media content in its digital form has been rapidly scaling up, resulting in popularity gain of cloud computing. Cloud computing makes it easy to manage the vastly increasing digital content. Moreover, additional features like, omnipresent access, further service creation, discovery of services, and resource management also play an important role in this regard. The forthcoming era is interoperability of multiple clouds, known as cloud federation or inter-cloud computing. With cloud federation, services would be provided through two or more clouds. Once matured and standardized, inter-cloud computing is supposed to provide services which would be more scalable, better managed, and efficient. Such tasks are provided through a middleware entity called cloud broker. A broker is responsible for reserving resources, managing them, discovering services according to customer’s demands, Service Level Agreements (SLAs) negotiation, and match-making between the involved service provider and the customer. So far existing studies discuss brokerage in a narrow focused way. In the research outcome presented in this paper, we provide a holistic brokerage model to manage on-demand and advance service reservation, pricing, and reimbursement. A unique feature of this study is that we have considered dynamic management of customer’s characteristics and historical record in evaluating the economics related factors. Additionally, a mechanism of incentive and penalties is provided, which helps in trust build-up for the customers and service providers, prevention of resource underutilization, and profit gain for the involved entities. For practical implications, the framework is modeled on Amazon Elastic Compute Cloud (EC2) On-Demand and Reserved Instances service pricing. For certain features required in the model, data was gathered from Google Cluster trace.

Index Terms — cloud broker, resource management, pricing, Inter-cloud computing, cloud federation.

1 INTRODUCTION

Cloud computing has become worldwide because of its increasing demands and requires more heterogeneous infrastructure which results in making interoperability an area of interest. This increasing demand is challenging for cloud customers as far as selection of an appropriate Cloud Service Provider (CSP) is concerned and it connects them to a particular CSP [1]. Inter-cloud computing is needed at this place. This paradigm of inter-cloud computing is in its initial stages but still it permits uninterrupted interoperability between clouds, no matter whatever their underlying infrastructure is. This enables users to shift their workloads across clouds in an easy way. Moreover, resources can be handled effectively through cloud brokerage which is an advantageous aspect of inter-cloud computing [1], [2].

On the clouds, most of the data-intensive applications are now installed. These applications, storage, and data resource are located in so many different manners that they have to reach even cross-continental networks. Due to this issue, the performance of cloud systems and user requests is affected by performance degradation in networks. The need to ensure service quality, specially for bulk data transfer, makes resource reservation and utilization a serious issue [3]. Jayant Baliga et al. [4] stresses upon the fact that more resources are utilized with the increasing digital content which causes more energy consumption. The performance and the overall cost of the services provided, both are affected by greater energy consumption. This intensifies the need for efficient resource management and dynamic pricing.

Ewa Deelman et al. [5] reinforces the fact in their study that significant amount of cost can be reduced by right amount of resource allocation, without having any effect on the performance. If we take customer’s perspective in consideration, honesty is a major concern in resource allocation and pricing. The customers are usually charged on hourly basis in current pay-as-you-go billing mechanisms which is subject to unfairness [6]. Being an economically-oriented idea, cloud computing focuses on fairness as its key feature in a pricing scheme. Personal fairness and social fairness are two types of pricing fairness in the terminology of economics. Personal pricing fairness is meant to be subjective and reasonable for consumers. On the other hand, social fairness refers to overall fairness maintained among the users, using the same service. Charging the same cost for the same service being consumed is called social fairness. If pricing is unfair, it becomes a reason of dissatisfaction for the customers and as a result, service providers would fail to gain the loyalty of its customers [6]. In addition to that, service underutilization also depends upon the loyalty of the customers. It is estimated that many datacenters experience 5% to 20% of utilization of their total resources [42], which shows that the datacenters are significantly underutilized. Underutilization can be countered by having a dynamic and customer’s historical record based resource management and
Cloud computing faces some other vulnerable challenges as well. Cloud federation or inter-cloud computing has been proposed to offer better consistency, accessibility, cost-efficiency, and Quality of Service (QoS). Though the research on inter-cloud computing is still not advanced enough, its efficacy cannot be denied in any way [7]. CSPs have their consumers spread all over the world. CSPs have to establish multiple datacenters at different places to provide services to their consumers. Prevaling systems are not accomplished enough to manage the load dissemination among datacenters, to define optimal location for presenting services to attain anticipated performance. Likewise, consumers’ geographic distribution cannot be projected as well. Hence, load management and service dissemination have to be completed automatically. Inter-cloud computing is intended to resolve these issues. Inter-cloud computing offers measurable provisioning of services with steady performance, under inconstant load and rapidly varying requirements [8]. Inter-cloud computing helps vibrant growth and reduction of resources, and dealing with rapidly varying service demands. A broker’s duty is to categorize suitable CSP, according to the requirements of its consumers, through cloud exchange [8]. To meet user and service requirements, a broker negotiates with the gateway to allot resources. It is not possible that a single cloud may always achieve the required computing and storage resources demands. Cloud customers are searching for better service at a reasonable price. All such circumstances inspire the beginning of cloud federation and cloud broker [2].

With cloud federation, a customer’s request from one CSP is served by another CSP, through the intervention of cloud broker. The broker has to achieve resource provisioning, which may sometimes become a tricky task [9]. This resource provisioning can be ad hoc or in advance. But a broker has to wisely decide what to do, on the basis of behavior or characteristic of consumer and type and price of service(s) [10]. CSPs prefer to have the data about resource arrangement in advance. This permits them to acclimatize to the demands of service and user and sanctions them to do capacity development in a better way [11].

There is no complete existing brokerage model so far which could handle all the significant jobs, including resource calculation, arrangement, billing, and refunding, particularly on the basis of changing features of customers, dilapidation in service quality and different conditions faced during service provision. As the demands are vibrant and may vary any time, it is a challenge for the broker to choose what quantity of resources has to be booked on demand.

There are several services which charge their customers on hourly clock. One well-known example is Amazon Elastic Compute Cloud (EC2). Even if a resource is utilized for a few minutes, the cost would be for the full hour, which is unfair to the users. When the billing duration is even greater, like daily basis (e.g. VPS.NET [12]) it will be a big concern [11]. As users’ requests are varied and change dynamically at any point, it makes it a demanding trial for the mediating broker to correctly decide about the resources it has to reserve [11].

In this paper, we present a cloud brokerage model for dynamically managing resource pricing and refunding, taking [36] further, in which the focus was on resource estimation. Taking into consideration the historical record of cloud customers, the model presents a dynamic and fair way of estimating service prices and managing refunds. For customer’s historical record, it’s relinquish or service give-up probability, overall relinquish probability, earned profit, and base value of service are considered. Service can be quit at any stage. Many factors can be involved in such a situation, like: SLAs violation, connection loss, power shortage, lack of interest in the service, customer finds a better option elsewhere, customer does not deem the service to be affordable anymore. In such a case, the broker has to reimburse its customer, according to the level of utilization. Prior studies also lack this important part of economic model, which we cover here. The reimbursement is done on the basis of level of service utilization, value of unutilized service, SLA versus acquired quality of service. Even though the CSP always try its best to fulfill the promises made during the SLA, however, at times it is not possible. Currently, service provisioning means does not consider this important issue in their reimbursement modules. That is why we have also taken into account acquired service quality while determining the refund amount. For advance reservation of services, pricing and billing is also provided. All in all, the model provides compensation and incentives to the customers wherever applicable. On the other hand, it penalizes the party which violates the SLAs. Penalty is also applicable to the customer when it quits the service before exhaustion of the contract. In this way, resource underutilization is minimized and the service provider is not entirely deprived of the profit it deserves.

For assessing practical implications of our methodology, the modeling was performed on Amazon EC2 On-Demand and Reserved Instances services. For certain historical records of cloud customers, Google Cluster2 trace of 41GB, 12000 machines, for a month-long period was acquired and used in this model.

The rest of the paper is arranged in such a way that details on cloud broker and its architecture are provided in section 2. The historical record based economics model is discussed in the third section. Section 4 presents the outcome of our model. Prior systems and related works are critically discussed in section 5. We conclude our paper in section 6, also providing future insight.

2 Cloud Broker

Cloud federation or inter-cloud computing is a term referred to a situation when two or more clouds have to communicate with each other, or another intermediary comes into play and

---

1. [http://aws.amazon.com/ec2](http://aws.amazon.com/ec2)
2. [https://code.google.com/p/googleclusterdata/](https://code.google.com/p/googleclusterdata/)
federates the resources of two or more clouds. This is done to cater the increasing users’ demands of media content. The intermediary between two clouds is called a broker. Broker is the entity which introduces the cloud customer to the cloud service provider and vice versa [33]. Inter-cloud resource management is also known as Cloud Broker Aggregation [39, 40].

A cloud broker helps in managing, controlling, and monitoring multiple clouds and share resources. It helps the customer in finding out the best provider and service according to its need and specified SLA. Thus, a broker benefits both the parties.

Broker manages commercial services by using a cost management system which includes Application Programming Interfaces (APIs) and a standard abstract API. Different modules perform a specified task in broker’s architecture, e.g., registration of new services is handled by Service Registration Manager. Deployment Manager deploys services and makes them available. Similarly, each module has its own specific utility.

A composite and flexible system of services provide assistance to inter-cloud. Inter-cloud Gateways are responsible for interoperability and transcoding related tasks [37]. Inter-Cloud Exchanges (ICX) are responsible for introducing attributes of cloud environment for inter-cloud computing [38]. This system is responsible for aggregating infrastructure demands from the broker and match them. Inter-cloud Root contains services like, Naming Authority, Directory Services, Trust Authority, etc.

Cloud Service Customer (CSC) can directly access CSP(s) as well but in that case, transcoding related tasks, SLA negotiation, and match-making are done by the CSC itself.

Since services are not yet standardized, it make it difficult to compare the services provided by the CSPs. In other words, cloud services are not really commodity services yet. Though it is possible to identify a common set of core services that CSPs are expected to provide. Even if the services provided by different CSPs have same set of functionalities, there can be substantial difference in terms of their convenience, accessibility, and ease of use. Reputation also plays an important role here. Well-known service providers like Amazon are more likely to be chosen over some less known but cheaper service provider. Broker’s responsibility here is to do match-making and provide the pros and cons of the service in detail to the customer. With the dynamic and historical record based resource management provided in this paper, the concerns discussed above can be somehow addressed, since cloud customers are also treated according to their loyalty and the acquired quality, not the expected quality, of the service.

3 Broker’s Resource Economics Model

The billing of broker is very accommodating. Pay-as-you-go billing model is one of its attributes. It helps the customers to scale their requirements and then pay accordingly.

CSC contacts cloud broker to acquire the required service(s) at best price. Broker performs negotiation and SLA tasks with the CSP. After the contract is finalized, the broker not only provides its services on ad hoc basis, but also predicts and allocates the consumption of resources in advance. Prediction and pre-allocation also depends upon user behavior and its probability of using those resources in future [34, 35]. For this purpose, broker performs pricing and billing in view of that, which is also presented in this section.

Pricing method is formulated as under:

\[
\rho_i = \begin{cases} 
(p_{S_P(L)} | p_{S_P(H)}) & \text{if } k > 0, \\
(p_{S_P(L)}) & \text{if } k = 0
\end{cases}
\]  

(1)

In case of a new customer, \( k = 0 \), the relinquish probability is set to low, at 0.3. This is because 0.3 is the average value of low relinquish probability range (0 ~ 0.5). Thus, it is practical to put a new customer in the category of reasonably loyal. When the customer has a history existing, the brokers decides about the price \( (p_{S_P(L)} | p_{S_P(H)}) \) on the basis of previous record.

\[
p_{S_P(L)} = \sum_{i=0}^{n} \mathbb{R} e s_i \int_{0}^{t} (U_i \ast t + uc_L + \Omega \ast \beta)
\]  

(2)

\[
p_{S_P(H)} = \sum_{i=0}^{n} \mathbb{R} e s_i \int_{0}^{t} (U_i \ast t + uc_H + \Omega \ast \beta)
\]  

(3)

\( p_{S_P(L)} \) represents the price for all the resources \( \mathbb{R} e s_i \) in service \( i \), which a customer requests and it has low relinquish probability \( L \) \( (0 < L \leq 0.5) \). \( p_{S_P(H)} \) shows the price determined for the customer of high relinquish probability \( H \).
(0.5 < \( H \leq 1 \)). \( U_i \) represents base price of service \( i \). Duration of the service is represented by \( t \). For customer’s historical record usable in this case, the relinquish probability is incorporated. Represented by \( \gamma \), the Average Overall relinquish Probability (AOP) shows the general record of relinquishing the services a particular customer has consumed so far. AOP gives an overview of the customer’s usage behavior, enabling the service provider decide about resource allocation accordingly. In this way, more realistic resource management and pricing is performed, which in the end helps minimize resource underutilization and the CSP is not deprived of the profit it deserves.

Equation 4 calculates the decision variable for user \( P_H \) (having high ‘H’ relinquish probability) and decision variable for \( P_L \) (having low ‘L’ relinquish probability) is presented by equation 5. The service ratio (e.g., 10% of the total amount) is shown by \( \beta \) which is set by the broker. Business contract and condition are its basis. Relinquish probability and/or user characteristic (\( uc \)) directly affect the final price. Increase in average probability would increase the final price.

\[
uc_H = \frac{\gamma \mu t + \mu_H \beta}{\delta} \tag{4}
\]

\[
uc_H = \frac{\gamma \mu t + \mu_L \beta}{\delta} \tag{5}
\]

Where total profit which is earned through service provider from the recent costumer is indicated by \( \delta \). In case of a new costumer, the average of profit earned is taken as the profit earned by the service provider from the current service on the customer’s request. This current profit is shown as \( \delta_c \). Increase in average profit lessens the value of \( uc \), which in turn reduces the final price.

A costumer can end the current service anytime. This is the point where CSP is required to repay the remaining amount to the costumer after termination of the service. This situation requires CSP to take into consideration two important factors, i.e. consumed resources and the remaining service value of the decided total initial service. The above mentioned case can be formulated through the following equations.

\[
\mathcal{R}_T = Y_{un} + Y_{deg} \tag{6}
\]

\[
\lambda_{un} = 1 - \frac{\mu t}{100} \tag{7}
\]

\[
Y_{un} = \begin{cases} 
Y_{app}, & \text{if } \mu \geq 60\%, \\
Y_{dep}, & \text{if } \mu < 60\%
\end{cases} \tag{8}
\]

In eq. 6, \( \mathcal{R}_T \) shows refunded total amount. \( Y_{un} \) is for the refund amount of consumed resources. \( Y_{deg} \) shows the refund factor which is to repay on the basis of quality degradation. It is not always achievable to keep up the promise made during SLA while delivering the service. This factor is considered in this model to judge CSP according to the decline in service quality. Decline in service quality would increase the value of degradation factor and therefore increase the repay value. Further calculation of \( Y_{un} \) is done through equation 9 to 12, while \( Y_{deg} \) through equation 13. \( \mu \) and \( \lambda_{un} \) denote utilized and unutilized resources respectively.

More profit of CSP is earned from those customers who have used more service. When such customers leave the service, some appreciation amount \( Y_{app} \) can be given to them during refunding. This appreciation amount can be calculated through Appreciation Index \( \omega \). For instance, the customers of 60% or more service utilization are in line for this. Such a case requires an increase in payback amount to make the customer happy. This factor acts as an attractive incentive, which allows the customers to use more services, by returning to the same service provider again over again.

This case can be presented through the following formula.

\[
Y_{app} = ((\lambda_{un} \cdot U_i \cdot t) - \frac{\beta}{\mu}) + \omega \tag{9}
\]

\[
\omega = \log (\mu \cdot t) \tag{10}
\]

\[
Y_{dep} = ((\lambda_{un} \cdot U_i \cdot t) - \frac{\beta}{\mu}) + \epsilon \tag{11}
\]

\[
\epsilon = \ln (\frac{\mu t}{100}) \tag{12}
\]

If a customer uses very less service, the refund amount would be depreciated \( Y_{deg} \), which is on the basis of Depreciation Index \( \epsilon \), deducting some amount from that customer. In our model, Depreciation Index is applicable if resource utilization is less than 60%. Natural log \( \ln \) is used in this particular case because it follows the way of appreciation factor in producing a negative value, which is based on utilization factor. Therefore, it would depreciate the final payback amount. Another factor, i.e., broker’s service ratio, which is shown by \( \beta \) is then deducted when the remaining amount is paid back.

\[
Y_{deg} = \frac{Q_{SLA} \cdot \frac{R}{100} \cdot ((U_i \cdot t) - (Q_a \cdot U_i)) \cdot \left(\frac{\mu t}{100}\right)^2}{Q_a} \tag{13}
\]

During service delivery, SLA violation on the basis of QoS is also made part of our framework. Here, \( Q_a \) stands for acquired QoS. \( Q_{SLA} \) symbolizes the promised QoS. Promised quality in contrast with achieved quality factor is determined by \( Q_{SLA} / Q_a \).

Acquired quality factor is generated by \( (Q_a \cdot U_i) \) on the basis of utilized service, which is then deducted from the actual price. This factor is changed by the type and price of service, producing quality degradation factor respectively. \( \left(\frac{\mu t}{100}\right)^2 \) decides the amount of utilized resources for duration \( t \).

CSP decides about the registration of the service when it is demanded by \( n \) number of customers. CSP’s decision depends on the relinquish probabilities of each customer. CSP sets a threshold value \( \theta \). This threshold value is affected by many factors, like, type, duration, and cost of service. It then adds up the relinquish probability of each customer \( P_i \). The service is registered only if the added probability is greater than or equal to the threshold value, otherwise, the service is not granted. The reason is that, a particular minimum number of customers are supposed to register, with the provision of some services.
Else, the service would become unaffordable for the customers or leaving service provider with a less profit. This case can be formulated as under.

$$P_T = \hat{x} (\sum_{i=0}^{n} P(L|H)_i)$$  \hspace{1cm} (14)

Service is provided if $P_T > \theta$

For some of the many different types of services provided by the CSP, advance booking is required or appreciated. The CSC is required to pay a small amount in case of advance booking. It is called Premium Amount here. The CSC can either utilize the service reserved in advance or cancels it. Hence, there are two stages. In stage 1, a small premium amount is paid by the CSC for advance reservation. In stage 2, the CSC pays the cost after the consumption of reserved resources and the Premium Amount is adjusted in the final cost. In case the customer cancels the reservation, it still has to pay the cost of reservation, since they were held for that customer. We present the formulation of this case in equation 15 to 19.

Stage 1: CSC reserves resources in advance, Premium Amount $\rho_{prem}$ would be:

$$\rho_{prem} = U_i \times t \times P_L \times uc_{prem}$$  \hspace{1cm} (15)

$$uc_{prem} = \begin{cases} uc_{prem1} & \text{if } \theta_p < \delta > U_i, \\ uc_{prem2} & \text{if } \theta_p \leq \delta \leq U_i, \\ 0.3, & \text{otherwise} \end{cases}$$  \hspace{1cm} (16)

$$uc_{prem1} = \frac{U_i + \delta \times P_L}{\delta}$$  \hspace{1cm} (17)

$$uc_{prem2} = \left(\frac{(U_i + \delta) \times P_L}{\delta + \delta_f} \right)$$  \hspace{1cm} (18)

$$\delta_f = \frac{\delta}{\delta_c}$$  \hspace{1cm} (19)

As we have used both $L$ and $H$ relinquish probabilities in the actual/initial price determination that is why $H$ is not included in this calculation. There is no need to use it again in Premium Amount calculation, as it is supposed that all the customers who are making service reservation have low ($L$) chances to cancel it. So that the CSCs are in the end encouraged to go ahead with the eventual service consumption. Therefore, only low relinquish probability $P_L$ is made part of the calculation here.

For Premium Amount calculation, the user characteristic is represented by $uc_{prem}$. The user characteristic is in contrast with the profit earned from current customer. It means more profit would make the value of $uc_{prem}$ smaller and as a result payable Premium Amount would also be smaller. If the profit is at least greater than profit threshold ($\theta_p$); only then $uc_{prem1}$ and $uc_{prem2}$ are applicable. This condition applies because of a set minimum price, called here as profit threshold ($\theta_p$). In our model, minimum priced service is of USD 6.336. Default $uc_{prem}$ value is used in a situation where profit does not reach the minimum target of USD 6.336. The default value used in this case is 0.3. $\delta_f$ represents current profit ($\delta_c$) factor to the total profit ($\delta$).

In equation 15, increase in $uc_{prem1}$ would raise the Premium Amount. Three cases are shown by equation 16. Third one is the default case which is already discussed above. Case 1 represents a situation where profit earned so far exceeds profit threshold ($\theta_p$) USD 6.336 and the price of requested service as well, then $uc_{prem1}$ is applied. Logic here is to reward those CSCs who made CSP able to earn more profit (greater than the price of currently requested service). Greater profit is inversely proportional to $uc_{prem1}$ and as a result Premium Amount would also become lesser.

Premium Amount is to be set carefully in a case where the cost of needed service is more than earned profit. This situation demands CSP to be focused on avoiding any sort of loss, if the CSC does not fully consume the resources afterwards. Broker deals with the CSP and CSC just according to their conditions as it plays a key role of mediator. Equation 18 shows an increase in the service price as the price and current profit factor ($U_i + \delta$) raise $uc_{prem2}$. Costing of the needed service is decided by price-to-current-profit factor. The divisor $\frac{U_i + \delta}{\delta_f + \delta}$ shows the proportion of contrast in price of presently demanded service from the acquired profit so far. Two factors are discussed with currently demanded service. First factor is that if service is costly and profit to service cost difference is of great deal, it requires CSP to be more cautious with that CSC and decide Premium Amount by keeping risk factor in view. Second factor includes giving profit to the CSP through current service. Now this merit and demerit are to be adapted in proportion. The same we attempted to do in eq. 18 and 19. The greater earned profit $\delta$ and current profit factor $\delta_f$ would raise the value in divisor, which consequently makes $uc_{prem2}$ smaller.

Stage 2: Case 1: CSC continues with the resource consumption, the final cost $\rho_{net}$ is determined as:

$$\rho_{net_{CSC}} = ARR \times \rho_{prem} + (P_{SP(L)} - \rho_{prem})$$  \hspace{1cm} (20)

$$\rho_{net_{CSC}} = ARR \times \rho_{prem} + (P_{SP(H)} - \rho_{prem})$$  \hspace{1cm} (21)

ARR represents Advance resource Reservation service Ratio. In our model, this ratio is set as 10%.

Case 2: CSC decides to revoke the booking, CSC has to pay the reservation cost, which is given in eq. 22.

$$\rho_{cancel} = ARR \times \rho_{prem}$$  \hspace{1cm} (22)

4 Implications and Outcome

This section presents the outcome of our model. The model defined though algorithms was first implemented in Java using CloudSim toolkit to know the working. Later, to evaluate the practical implications, the methodology was modeled on Amazon EC2 services. For that purpose, Google Cluster trace was also extracted and incorporated in the model. In the end, the feasibility and rationality of the model was evaluated.

4.1 Description of Evaluation Setup

Various parameters are made part of this model, also incorporating heterogeneous services and customers. In Table
1, basic setting of key parameters is presented. There are two categories of services used in this model: (i). On-demand, (ii). Advance reservation. In this regard, various services provided by Amazon EC2 are used for modeling the algorithms which determine pricing, billing, and refunding. To get parameters like heterogeneous types of customers, relinquish probabilities, trend of utilization, type of service, Google Cluster trace was applied. Some of the services did not have our required relinquish probabilities in Google Cluster trace, which were then supposed in the model, keeping in view the trend in the available trace.

### TABLE 1
Parameters' Setting for Evaluation

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Level Agreement (Q_{SLA})</td>
<td>0-9</td>
</tr>
<tr>
<td>Acquired service quality (Q_{a})</td>
<td>0.9,0.8,0.7,0.6,0.5,0.4,0.3,0.2,0.1</td>
</tr>
<tr>
<td>Service Price (ρ)</td>
<td>USD 1 \sim USD 1000</td>
</tr>
<tr>
<td>User characteristic (L or H)</td>
<td>L &gt; 0 &amp;&amp; \leq 0.5, H &gt; 0.5 &amp;&amp; \leq 1</td>
</tr>
<tr>
<td>Unutilized resources (α)</td>
<td>90% \sim 10%</td>
</tr>
<tr>
<td>Service utilization</td>
<td>10% \sim 90%</td>
</tr>
<tr>
<td>Number of registered services</td>
<td>20</td>
</tr>
<tr>
<td>Service duration(t) in days</td>
<td>30</td>
</tr>
<tr>
<td>Broker’s service ratio (β)</td>
<td>10%</td>
</tr>
<tr>
<td>Advance reservation service ratio (ARR)</td>
<td>10%</td>
</tr>
</tbody>
</table>

### 4.2 Service valuation for new CSCs

Price or markup is an essential element in resource management. Markup rate or assessed financial value needs to be dynamic because of the discrepancy in the types of CSCs and the services provided. If markup is too inclusive, it will lead to injustice. Prices are calculated on the basis of traits of a CSC and kind of service demanded. CSC can either be by-and-large new, existing but demanded service S for the first time, or current and asked for service S before as well. Subsection 4.3 gives the third synopsis. In synopsis 1, since service provider has no earlier proof of service consumption, it takes the CSC as almost trustworthy and calculates the markup rate on the basis of low relinquish probability (i.e., 0.3). In synopsis 2, the service provider possesses the common record of CSC’s earlier activities, so its AOP can be calculated. But when CSC has asked for service S for the very first time, SOP cannot be assessed. Markup rate is thence calculated on the basis of AOP. For simplicity sake, Figure 2 shows the difference in markup according to the variance in AOP, by showing two categories, as $Price - L$ and $Price - H$. For a base USD 9.36 per month t2.micro Amazon EC2 service, providing 1 vCPU and 1 GB Memory, markup rate for low relinquish probability ($Price - L$) is USD 9.68 (a CSC gets various added services, like refunding). USD 10.09 will be decided $Price - H$. Thus, CSC will be dealt accordingly, which will finally help to create equality. Consequently, when the particular CSC let go the service, the service provider must have earned suitable profit. Like, where the resources are released, the dealer needs not to be in the debit.

### 4.3 CSC characteristics based valuation

In the third synopsis, if service S has been demanded by a CSC prior as well, then cost will be determined on the basis of earned profit too, other than relinquish probabilities. This is to ensure that CSC has been provided more inducement, from whom CSP has successfully gained higher profit. This motivates the CSC to consume maximum service and to have more trust on the CSP. Horizontal axis in Figure 3 exhibits various CSC samples. Vertical axis represents cost in USD. The service give up probability is 65\% in case of CSC 1 and gained profit is low, USD 13. As a result, for t2.micro Amazon EC2 service, having base price USD 9.36, CSC 1 needs to pay 16.33. The cost is calculated realistically and in a just way, as CSP has the chronological data of the CSC.

When CSC 1 of subsection 4.2 (Figure 2) is compared with this CSC 1, it informs us that if the CSC is new, it has to pay low amount, even though it comes in the range of Price-H. The pricing is more appealing to one’s senses based on CSC’s traits in the example presented here, because of being well informed about the chronological details of CSC 1. In the case of CSC 2, payable price is USD 12.66, because of comparatively lower give up probability, 32\%, with CSP earned comparatively more profit, USD 32. In case of CSC 3, give up probability is 65\%, profit gained so far is USD 29, greater than the above two cases. Therefore, CSC 3 needs to pay USD 16.07. Among all these cases, CSC 4 is considered to be the best case in the result discussed so far, logically as having lowest relinquish probability and greatest gained profit for CSP, has to pay USD 12.32.
4.4 Value of unutilized service with fixed acquired service quality

CSC can choose to stop the service it has been consuming due to many apprehensions. In such a case, the service provider has to take three steps. First, assess the worth of utilized resources. Second, figure out the cost of unutilized resources. Third, pay back the leftover amount to the CSC. Five such examples are manifested through Figure 4, when the service was stopped by a CSC. Utilization level is represented by left vertical axis in the figure, whereas, right vertical axis exhibits refund amount in USD. For a fixed Amazon EC2 t2.micro On-Demand service with 1 vCPU and 1 GB memory priced at USD 9.36 per month, CSC 1 gets USD 6.7 with only 24% consumption. Refund depends upon not only the type of service and consumption level, but also on broker’s service ratio. CSC 4 and 5 have used more than half of the resources. Therefore, the value of unconsumed resources is calculated accordingly.

4.5 Refunding for various Amazon EC2 services

Payback amount is discussed in this section for various categories of utility with CSCs, possessing different usage levels. Figure 5 horizontal axis shows such proof of CSCs, while vertical axis exhibits payback amount in USD. For a fixed Amazon EC2 t2.micro On-Demand service with 1 vCPU and 1 GB memory priced at USD 9.36 per month, the amount of unused capital after 35% consumption is estimated as USD 5.8 for CSC 1. CSC 2 consumed 18% of t2.small Amazon EC2 On-Demand service (1 vCPU, 2 GB memory) at USD 18.72 per month. CSC 2 then receives unutilized resources value of USD 14.79. Correspondingly, depending on the usage level and service type, unutilized values of capitals differ.

4.6 Refunding according to service quality degradation

Consumer contentment and trust towards a certain service provider depends upon the service quality, which no doubt plays a pivot role. The major interest of the service quality for which CSP needs to think about SLA compliment. Though it is not always easy and at times, dissatisfied customers may change service. This issue has been pondered over in the suggested model, which shows that if SLA is not accomplished, CSP will be punished and CSC gets the raised payback amount according to the decline in service quality. The payback amount differ according to the attained SLA or service quality. If further degrading of service is noticed, greater refund would be paid. In this section, we change SLA covering from 0.1 to 0.9. Figure 6 shows service degradation, with 0.1 as the worst quality, while 0.9 is considered to be the best quality.

The payback amount for Amazon EC2 On-Demand t2.micro USD 9.36/month service, based on service quality, with fixed consumption of 70% in this case. It clearly manifests that refund amount is directly proportional to the degradation in service. The refund amount is not USD 2.889, if the attained quality is 0.1 with 70% of the service consumed. On the contrary, due to the worst quality being provided, USD 6.52 are paid back. Simultaneously, with better quality, refund is adjusted accordingly. This is to ensure justice to attain customer contentment.

4.7 Refunding for different Amazon EC2 services – varying service quality

Various usage levels for unlike services, with different accomplished quality levels are represented in Figure 7. It has been illustrated that in the first case, USD 9.36 service was demanded, the attained quality was 0.7. Though it is thought out to be acceptable, but it was below the assured quality of 0.9. 80% service was used by the customer, hence the payback amount is USD 2.1, instead of USD 1.8. Utilization is inversely proportional to refund, i.e. greater the usage, lesser would be the payback. Opposite to this, higher is the reduction in quality of service, greater would be the refund factor and thus the payback amount. In this way, decline in service quality is directly proportional to greater refund amount. In addition to this, in case of CSC 2, service demanded was USD...
18.72 (Amazon EC2 On-Demand t2.small, 1 vCPU, 2 GB memory), usage was 72%, and achieved service quality was 0.6. The payback is USD 5.82. Correspondingly, the process of payback depends on the basis of usage level and attained service quality, providing CSC inducement and punishing CSP, where befitting.

4.9 Depreciation in refund amount

When service consumption is below a certain level set by the service provider (60% in our model), it results in low earned profit for the CSP. In such case, resources are underutilized, which has to be minimized, if not prevented. CSP handles underutilization by refunding a depreciated amount to the CSC. Figure 9 represents cases having 25% utilization of Amazon EC2 On-Demand services. In case 1, for USD 9.36 service, depreciated refund is calculated as USD 5.23 instead of USD 6.62. Service prices effect the depreciation factor in the same way.

4.10 Premium Amount for Amazon EC2 Reserved Instances – the case of new CSC

Certain services are needed to be held in reserve. In this case of reservation, CSP charges an advance payment, called here as “Premium Amount” because broker allocates resources and
time slots for the requests made by the CSCs. User characteristic is not considered in case of advance booking because they are already taken into account while calculating prices and other concerned issues. The final total price would already be according to that. Therefore, it is not needed to include it again in advance reservation. Moreover, broker’s objective is to hold resources and interest CSCs to use the booked service. Hence, type of service determines the premium amount, and low relinquish probability of requesting CSC is assumed. Figure 11 demonstrate different Amazon EC2 Reserved Instances services on the horizontal axis and payable Premium Amount on the vertical axis. For t2.micro Reserved Instances service 1, priced at USD 6.336, calculated Premium Amount is USD 0.58. For t2.small USD 12.672 service, the Premium Amount is 1.15. In the same way, Premium Amount is determined based on the value and type of service. The presented result shows the scenario where CSC is existing (not new) and the earned profit from it is less than the threshold value (as discussed with eq. 16). In such case, as the earned profit is still very low, if the Premium Amount is determined keeping in front case 1 or 2 of the eq. 16, then the value would be high, which demotivates the CSC. That is why the third case of eq. 16 is made operational instead. The threshold is set at the minimum priced service, i.e., USD 6.336.

4.11 Premium amount according to earned profit– the case when profit > price

For those CSCs, from whom the service provider and broker have already earned high profit, incentive must be provided while reserving resources. That is, more is the profit, lesser should be the reservation fee. On the basis of this concept, results are presented in Figure 12. If the earned profit is USD 36, in case of CSC 1, for USD 6.34/month t2.micro Amazon EC2 Reserved Instances service, the calculated Premium Amount is USD 0.1. Premium Amount is determined in accordance with the earned profit and the value of service. In the second case, when service price is USD 12.68, the calculated Premium Amount is not increased as much, calculated as USD 0.17. The reason is the amount of earned profit, which is USD 85, much higher than the CSC 1. This shows the effect of profit and the way incentive is provided to attract the deserving CSC. In such a way, both the factors, the price and the earned profit, are considered while determining reservation cost.

4.12 Premium amount according to earned profit – the case when profit < price

In the contrary case when earned profit is low, i.e., less than the value of currently requested service, but higher than or equal to the threshold (minimum priced service in our model), then Premium Amount is calculated in a different way. CSP has to make sure that since the chronological record does not suggest that the requesting customer is truly reliable, thus, broker has to make sure that CSP does not go into loss at the end. Being an intermediary, it is broker’s responsibility to act rationally and be justified with both the involved parties. Therefore, based on case 2 of eq. 16, the Premium Amount is determined. As the profit increases, the Premium Amount would experience decline according to eq. 16 case 2.

For CSC 1 in Figure 13, earned profit recorded is USD 5.07. Service currently requested is Amazon EC2 t2.micro Reserved Instance, priced at USD 6.34/month. The Premium Amount assessed is USD 0.68. Comparing this case with the CSC 1 of Figure 15, it is understandable that when profit was higher than the price of requested service, the Premium Amount was subsidized accordingly. This shows the effect of Premium Amount calculation on the basis of all applicable scenarios.

CSC 8 has the highest Premium Amount, USD 61.06, for Amazon EC2 m4.xlarge service, priced at USD 426.32/month. This is because the earned profit is relatively low (USD 87.9), while the request made is of a costly service. Hence, broker makes cautious calculations.
5 RELATED WORK

There is no standard architecture yet available for data communication, media storage, compression and media transfer other than Inter cloud computing, which is still at the initial stage. All previous studies stress either architectural blue print or Inter-cloud resource management but in a very superficial manner.

Using a pair of proxy was introduced by S. Ferretti et al. [13], at the user’s side a client proxy and at the cloud side a server proxy, in order to incorporate the cloud to the wireless unit unlined. Daniel Diaz et al. [14], [15] also introduce proxy as a bridge between home cloud to other home clouds and to the outside, public media clouds for the purpose of sharing items. This proxy performs multiple tasks of arranging the multimedia items, allowing public cloud to form search database and the categorization of content. The users get opportunity to look for the contents of their own choice through discovery service.

The proxy for transcoding and transfer of media is given by H. Zixia et al. [16]. While J. Xin et al. [17] suggests usage of peer to peer (P2P) for transferring media stream outside media cloud. It establishes an intercrossed architecture P2P as well as media cloud.

A large number of resources are needed for transcoding and compression of media. R. Pereira et al. proposed an architecture in [18] [19], the usage of Map-Reduce model for this purpose, in private and public clouds. J. Feng et al. [20] has presented the concept of stream oriented cloud and stream-oriented object. The authors present stream-oriented cloud with a high-level description.

Assessment for security and privacy in cloud storage was presented Wang Cong et al [21], [22], Yongdong Wu et al. introduce their work based on access control in cloud federation environment [23]. Yau et al. Concentrates on data integrity in clouds [24]. Zhifeng Xiao et al. [25] also emphasizes security and privacy concerns in cloud computing.

Rogers Owen et al. [10] has presented resource allocation mechanism but did not consider estimation of resources, their valuation, and refund management. Park Ki-Woong et al. [26] has presented a secure mutually verifiable billing system to resolve different future disputes. Their work does not focus on overall resource management, pricing, refunding or similar important features of cloud broker but their only focus is on the reliability of transactions made in purchasing and consuming resources. Wei et al. [11] proposed a brokerage service for instance reservation. The authors proposed a brokerage service which is for on-demand reservation or resources for IaaS clouds. And their work has limitation to only on-demand jobs while they do not present anything beyond that.

Jrad Foued et al. has given a generic broker architecture. They presented broker’s handling of SLA management and interoperability of resources [1], [2].

Yang Yichao et al. [3] presents algorithm of resource allocation in a very simplistic way. Ewa Deelman et al. [5] has presented performance tradeoffs of different kinds of resource provisioning plans. With that they also presented tradeoffs of Amazon S3 storage space. But their work does not include pricing strategies and other tasks regarding resource management. Shadi Ibrahim et al. proposed the fairness principle in pricing with regard to micro-economics [6], without mentioning the pricing procedure for different types of services. Their work is only confined to micro-economics pricing concept. Nikolay Grozev et al. discusses basic classification for inter-cloud architecture [7]. Buyya et al. presents architectural fundamental of inter-cloud computing [8]. David Villegas et al. present in [9] how creating a cloud federation environment has an impact over multiple clouds.

Kan Yang et al. present a vigorous auditing protocol for ensuring the incorporation of stored data in the cloud. They suggest an assessment framework for cloud storage [27]. Zhen Xiao et al. present a resource allocation system that uses virtualization technology to dynamically distribute resources, according to the demands of the service [28]. In their study, they present measuring the off-balance in resource usage. Other resource management issues and inter-cloud environment is not under consideration in this study. Resource information sharing through proxies was introduced by D. Cenk Erdil, in [29]. In cases where clouds are not available and there is no direct control, usage of proxies can make resource information available to them. The emphasis of this study is the significance of resource information sharing. Rakpong et al. discuss resource allocation in mobile cloud computing environment in their work [30]. They mention about communication/radio resources and computing resources, but only for mobile cloud.

According to Hong Xu et al. [31], dynamic pricing strategy is still a challenge and static pricing strategy is dominating the cloud computing market. Customer behavior and demand changes randomly. Dynamic allocation and pricing of resources are required for meeting customer demands and to fairly distribute resources and their pricing. Author is discussing the Amazon, its spot price history drawbacks, and motivation for dynamic pricing development. According to Zhou et al. [32], in today's cloud computing related research, pricing and cost optimization are the hot topics. Further, the authors say that ad hoc optimization strategies which are most
available strategies, have failed to optimize for different workloads.

6 CONCLUSION AND FUTURE WORK

Since inter-cloud computing is a novel paradigm, prior research works still lack a lot in providing a dynamic and more rational way of resource management and resource pricing. In a successful business process, all involved parties must be in a win-win situation. This is possible only if the mediating broker manages resources in a dynamic way, also considering the previous usage record and traits of the customers. In this way, the problem of resource underutilization is addressed, which is becoming a concern for the cloud datacenters. In addition to that, since cloud arena is not very mature yet, trust is also a matter of concern for the service provider and the consumer. Customer has to make sure that the promise made during the SLA would be fulfilled and in otherwise case, compensation would be made. Similarly, service provider has to make sure that the customer is provided with the resources according to its usage behavior, in such a way that more profit could be earned and resource does not go underutilized. All such aspects were made part of our presented customer characteristics and historical record based dynamic resource pricing and refund management model. In our framework, customer’s relinquish probabilities for currently requested service and for all the other services generally, the profit earned from it, the value of service, degradation in quality, SLA based decided quality, and the level of utilization were considered while designing the dynamic methodology. On the basis of Google Cluster trace of 41GB, gathered from 12000 machines for a month-long period, then modeled on various On-Demand and Reserved Instances services of Amazon EC2, the results are presented. This shows the practical applicability of our model, along with the outcomes it gives. The results and their discussions motivate the usefulness of such model and endorse our model’s utility in cloud business process.

In the future, our plan is to apply Quality of Experience (QoE) based more focused quality degradation refund management.

ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No.NRF-2013R1A1A2013620). This work was also supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (B0101-15-0535, Development of Modularized In-Memory Virtual Desktop System Technology for High Speed Cloud Service). The corresponding author is Prof. Eui-Nam Huh.

REFERENCES

BIOGRAPHIES

Mohammad Aazam is a Post-Doctoral Fellow in the Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada. He did his Ph.D. from the Department of Computer Engineering, Kyung Hee University, Korea, in 2015. His MS was from Mohammad Ali Jinnah University, in 2011 and BS from Gomal University, in 2006. He has served as Lecturer at Federal Urdu University, Mohiuddin Islamic University, Bahria University, and Allama Iqbal Open University, Islamabad, Pakistan during 2007 to 2012. He has also been Adjunct Faculty at SZABIST and Research Engineer at Mohammad Ali Jinnah University, Islamabad, Pakistan. He is currently serving Sensors & Transducers Journal as Editor and IEEE Communications Magazine as Associate Editor. Besides, he is serving several IEEE Transactions as Reviewer Board Member and is a Technical Program Committee member of several conferences. He has, so far, published more than 80 journal and conference papers. He received Best Research Output awards consecutively for all the three years of his Ph.D. He received President's Scholarship for his Ph.D. studies. Before that, he received Erasmus Mundus scholarship for Post-Graduate Research at Bradford University, UK, in 2009. He also received fellowship for ISOC APAN (Asia Pacific Advanced Network)'s 36th meeting, in 2013. He is a Member IEEE, Member IEEE Communications Society, Member IEEE Cloud Computing, Member IEEE Internet of Things, Member IEEE Smart Cities, Member ISOC, Member ISOC-APAN, and Member KIPS (Korea Information Processing Society).

Eui-Nam Huh has been Professor in the School of Computer Engineering, Kyung Hee University, since 2011. He is also founding Director of Realtime Mobile Cloud Research Center (RmCRC) and Chair ISOC Asia Pacific Advanced Network (APAN). He did his PhD from The Ohio University, in 2002. His MS was from University of Texas, in 1995, while he did his BS from Busan University, in 1990. His research interest lies in Cloud Computing, Internet of Things, Wireless Sensor Networks, HPC, and Real-time Mobile Cloud Computing. He is serving several highly reputed journals as Editor, including, but not limited to KSII Transactions on Internet and Information Systems, International Journal of Distributed Sensor Networks. He is a Member IEEE, Member ACM, Member Korea Information Processing Society (KIPS), and several other technical committees.

Marc St-Hilaire is currently an Associate Professor in the School of Information Technology with a cross appointment to the Department of Systems and Computer Engineering at Carleton University, Canada. He is a senior member of the IEEE and is involved in the organisation of several national and international conferences and workshops. His research interests include wireline and wireless networks, mobile computing, performance analysis, telecommunication network planning and mathematical modeling. He authored or co-authored over 90 technical papers.

Chung-Hong Lung received the B.S. degree in Computer Science and Engineering from Chung-Yuan Christian University, Taiwan and the M.S. and Ph.D. degrees in Computer Science and Engineering from Arizona State University. He was with Nortel Networks from 1995 to 2001. In September 2001, he joined the Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada, where he is now a professor. His research interests include: Communication Networks, Software Engineering, and Distributed Systems.

Ioannis Lambadaris (M'94) was born in Thessaloniki, Greece. He received the Diploma degree in electrical engineering from the Polytechnic School, Aristotle University of Thessaloniki, Thessaloniki, Greece, in 1984, the M.Sc. degree in engineering from Brown University, Providence, RI, USA, in 1985, and the Ph.D. degree in electrical engineering from the Department of Electrical Engineering, Systems Research Center (SRC), Institute for Systems Research (ISR), University of Maryland, College Park, MD, USA, in 1991. After finishing his graduate education, he was a Research Associate at Concordia University, Montreal, QC, Canada, from 1991 to 1992. Since September 1992, he has been with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada. Currently, he is a Professor in the same department. His interests lie in the area of applied stochastic processes and their application for modeling and performance analysis of computer communication networks and wireless networks. His current research concentrates on quality-of-service (QoS) control for IP and evolving optical networks architectures and stochastic control/optimization in emerging wireless networks. His research is done in close collaboration with his students and colleagues in the Broadband Networks Laboratory http://www.sce.carleton.ca/bbnlab/bnlhome.shtml. Dr. Lambadaris received a Fellowship from the National Fellowship Foundation of Greece (1980–1984) during his undergraduate studies. He also received the Technical Chamber of Greece Award (ranked first in graduating class). He was a recipient of a Fulbright Fellowship (1984–1985) for graduate studies in the U.S. While at Carleton University, he received the Premiers Research Excellence Award, and the Carleton University Research Excellence Award (2000–2001), for his research achievements in the area of modeling and performance analysis of computer networks.