BID PRICE CONTROL AND DYNAMIC PRICING IN CLOUDS

Arun Anandasivam, University of Karlsruhe, Englerstr. 14, 76131 Karlsruhe,
arun.anandasivam@kit.edu

Marc Premm, University of Karlsruhe, Englerstr. 14, 76131 Karlsruhe,
marc.premm@student.kit.edu

Abstract

The term Cloud Computing represents a paradigm for offering different kind of Web services, which can be dynamically developed, composed and deployed on virtualized infrastructure. This work will extend the concepts known from the revenue management to the specific case of Cloud Computing and propose two models, bid price control and a variant of dynamic pricing, that will compete with the commonly used static pricing. Both models will try to maximize revenues by controlling the availability or price of every offered fare class. The aim is to understand from a Cloud Computing company's perspective, how decisions about the pricing and the optimal allocation of the given resources for the various Cloud Services can be supported. As expected, simulation results show that an optimally adjusted dynamic pricing model will outperform any pricing model with static prices and will simultaneously contribute to slightly smoother resource utilization in some cases. However, we will see that the adjustment itself is difficult to realize, and if conducted suboptimal, it may also have certain disadvantages compared to static prices. In combination with a reasonable product differentiation, the bid-price method performed very solid and in nearly any case better than the pure static pricing model.

Keywords: Cloud Computing, revenue management, pricing, IT services

1 INTRODUCTION

Cloud Computing has recently become very popular as a new paradigm to shift IT resources and software from locally independent computers to a more collaborative level (Hayes, 2008). The hub-and-spoke system known from the client-server model is replaced by a geographically distributed cluster system. The concept of cloud computing comprises many existing technologies and architectures like on-demand or utility computing, software as a service and AJAX. Providers like Amazon or Rackspace offer products on the infrastructure level to run individually configured operating systems. Server farms remain idle most of the time. 99 percent of the entire computing capacity of a company is only used 10 percent of the time (Weiss, 2007). Additional revenue can be earned by utilizing resources. The resource demand is mostly both dynamic and unpredictable. Defining demand based pricing policies can influence the behavior of price sensitive customers (Desiraju and Shugan 1999). Customers, whose utility is highly depending on prices, would shift their computational jobs to a time, where resources are cheap. Business customers, though, are less flexible and are thus less price-sensitive. Their utility is increased by getting the job done on time or getting a better service. When prices are relatively fixed and demand outweighs supply, providers have to manage their capacity by deciding whether to accept and incoming request or waiting for a future request demanding a higher valued service. This will probably induce higher revenue. This problem is also known as the capacity management problem from the Revenue Management domain (Talluri and van Ryzin, 2004). Solutions have been successfully applied to the airline sector in the majority of cases (McGill and van Ryzin, 1999).

In this paper, we analyze how Revenue Management concepts can be useful to the Cloud Computing domain. The aim is to understand from a Cloud Computing company's perspective, how decisions about the pricing and the optimal allocation of the given resources for the various Cloud Services can be supported. Our contribution is threefold. At first, the requirements for applying Revenue Management to Cloud Computing have to be analyzed. In section three we derive a mathematical model to determine appropriate policies when to accept or reject requests for Cloud Services. Two models are compared, namely bid price control and dynamic pricing. As a supplementary benchmark, we use a static pricing model. The proposed bid price model and dynamic pricing models are evaluated via simulation in section four. Section five concludes the paper.

2 RELATED WORK

2.1 Cloud Computing

The term *Cloud Computing* is currently used in various ways and is often confounded with the term Grid Computing. Grid Computing accrued in the mid 1990s and originally denoted a scientific network to share computation power for computationally intensive jobs. The main characterization of a Grid is the distribution of a computing job in a somehow connected network. Accordingly, jobs for Grids are divided in several small jobs which are distributed to independent servers or desktop computers (Foster et al., 2001). This opens the possibility to share resources between institutions that are dispersed among different geographical locations. However, sharing computing resources is also a central aspect of Cloud Computing, but with a focus on virtualized instances that usually run on a server cluster (Buyya et al., 2008). A server cluster typically distributes computation power among several locally connected servers and hosts the virtual instance, which has been allocated to a certain customer. In scientific Grids the institutes are provider and consumer similar to P2P networks. Cloud Computing is commercially driven and most of the providers do not consume the resources they offer. The role between provider and consumer is currently disjointed. Some providers offer services in Clouds, which are based on other services like Rackspace, Rightscale or MorphExchange. In particular, services in this case are distinguished. For example, the product MorphExchange is a

platform providing a Tomcat server for uploading and hosting Java web applications (Platform as a Service). It uses Amazon's Simple Storage Service (Infrastructure as a Service) for storing the applications. On top these services, software or data can be provided as a service as well like StrikeIron offering a data service in combination with Salesforce's Customer Relationship Management software service. Lawton describes the cloud applications (Software as a Service) as Web-based applications accessed via browsers but with the look and feel of desktop programs (Lawton 2008). Skillicorn already emphasized in 2002 the advantages of composing simple services together to a new Web service with an added value for the customer (Skillicorn 2002).

Another aspect is the relationship between Clouds and Grids. Grid and Cloud Computing can be viewed as integrating several instances of one or multiple Clouds in a Grid. For example, we can use Grid technology for connecting multiple Clouds that do not have to be allocated at the same geographical location to combine computation power and furthermore distribute utilization (Boss et al., 2008). This definition implies a standard protocol to interact between several cloud instances. However, this is strictly depending on the providers, who supply Web Services to their products. Moreover, currently the Grid is more academia driven, while Cloud Computing is focusing on commercial application (Weinhardt et al., 2009). Earlier approaches from IBM or SUN Microsystems to commercialize Grid Computing have not been successful yet.

2.2 Revenue Management

The term *Revenue Management* is most commonly used for the theory and practice of maximizing expected revenues by opening and closing different fare classes or dynamically adjusting prices for products (Talluri and van Ryzin, 2004). The development of scientific research methods in this discipline started after the deregulation of the American Airline industry in 1978. They relaxed restrictions over standardized prices and profitability targets enabling dynamic pricing and resource allocation, although the first approach was back in 1972 by Littlewood (Littlewood, 1972, Belobaba, 1987). (Netessine and Shumsky, 2002) define Yield Management as a part of Revenue Management although boundaries between both are often ambiguous.

The perishability of the offered products is one of the main characteristics of Revenue Management. For example, a hotel cannot save up one room for the next day. Instead, an unused room becomes worthless without creating any revenue (Netessine and Shumsky, 2002). To get an efficient return from utilizing Revenue Management techniques the arriving customers have to be segmented into different classes. Every customer segment can be distinguished by different needs and thus by higher or lower reservation prices. Consequently, the main goal in Revenue Management theory is to find the combination of customers which seems to provide the highest possible revenue (Kimes, 1989). Therefore, we have to choose between opening a certain product class for sale or protecting it for a more profitable customer. However, future demand is not certain what causes this problem to get far more complex. If the vendor decides to protect a product for future demand, he takes the risk of ending up without selling the product (Goldman et al., 2002, Netessine and Shumsky, 2002). Although the vendor would face lower risk by immediately using the possibility to sell the product, we assume for all models presented in this work that these transactions are repeated often enough to justify a risk-neutral approach (McGill and van Ryzin, 1999, Bitran and Caldentey, 2003).

2.3 Application of Revenue Management in Clouds

The application of Revenue Management to other domains is only feasible, if several requirements are fulfilled. The requirements enable a comparison to traditional Revenue Management approaches which will be useful to formulate and optimize the unique optimization problem analogously to established strategies.

a. Time horizon: Like the Amazon EC2, the duration of every service is defined as a finite time period. With this assumption, we can handle every full hour as a separated product, what makes it comparable with a seat on one flight leg in airline industry or a room for one night in a hotel (Kimes, 1989).

b. Perishability: The perishability of the offered products is one of the main characteristics that have to be fulfilled for an appropriate use of Revenue Management (Weatherford and Bodily, 1992). The usage of resources on servers is limited to a point of time. Afterwards they are worthless, since they are not storable. As long as enough IT resources are left to provide at least one product, the revenue is still not maximized.

c. Production Inflexibility: IT resources are largely fixed and can only be extended at disproportional costs in relation to their running expenses. Product supply is thus limited and not extensible in a specific time horizon (Weatherford and Bodily, 1992). As most Cloud Computing providers offer their services on relatively huge server cluster. The integration of an additional resource like buying a new server will incur high additional fixed costs due to integration and increasing maintenance cost for the provider of infrastructure services.

d. Possibility of booking future products: Advance reservation has some relevant benefits (Boss et al., 2008, Dube et al., 2005). On the one hand, it gives the user the possibility to ensure a prospective computation demand by reserving the required products for the desired time. On the other hand, it provides the seller with the ability to easily discriminate customers by their valuation.

e. Customer segmentation: The offered products have to be differentiated by adding or restricting certain features to reach the desired customers for every price class. In most cases, the demand for these price classes differs particularly in their valuation for special product features and price sensitivity (Talluri and van Ryzin, 2004). For example a service level might indicate the minimum percentage of availability. Additionally, we might also identify different needs for resource combinations between price segments and with future booking enabled. Moreover, restrictions for reservation changes can be adopted commonly known from the airline industry.

f. Multiple products based on same resources: For Cloud Computing we have the desired advantage of flexible products. The resources of a Cloud Computing center are able to provide different service offerings using the same IT resources. Several products or product bundles can be defined, e.g. containing different amounts of CPU power or memory and defining individual service levels as aforesaid. The flexibility of the initial capacity provides plenty of resource combinations (Bitran and Caldentey, 2003).

h. Overbooking: The Cloud Computing overbooking procedure would be to sell more of the computing and storage capacity than the computing center has. In this case, not every costumer will exploit its reserved resources completely. Overbooking of Cloud Computing resources allows more flexibility. (Urgaonkar et al., 2002) show that the usage of overbooking techniques can increase utilization drastically: Already an overbooking rate of just 1% may increase the utilization of the whole computing cluster by a factor of two without losing meaningful availability guarantees.

It is important to consider customer needs. On the one hand, we believe, that in contrast to most traditional Revenue Management industries there is a market for open-end requests, which explicitly do not have certain end dates. That is because these virtual computing resources are likely to replace local computers which are in most cases also bought without a fixed time horizon for its usage. This issue is considered in our model. On the other hand, these requests would lead to the loss of an accurate resource schedule for future booking requests, so we will assume for the following, that there are no open-end requests, instead these have to be formulated as a long-term request with a finite time horizon (Boss et al., 2008).

3 PRICE DETERMINATION

The goal of a provider is to identify the appropriate price to maximize his revenue and satisfy the consumer demand. To give the customer the highest possible flexibility, we will allow every possible hardware configuration based on the given resources $k = 1...K$. This approach can already be observed

in the industry. For example the *Enterprise Cloud* of *terremark*[1] has the capability to offer individually configured servers. The user can assemble the desired product by choosing in each case a certain amount of processing power, storage size and memory size.

3.1 Scenario

Products are requested over time. Within a product planning period T_p resources can be planned and allocated at $t_p = \{0, \ldots, T_p\}$. Every product and thus every resource k for every point in time t_p, will be offered in different fare classes c. The price differences between classes will be justified with different service levels. For example, the higher valued class A will ensure 99.9% availability while the lower valued class B will only provide 99%. Other services may be faster reaction times or higher security specifications like redundancy checks. Following (Bitran and Caldentey, 2003), the connection between price and demand dependent on the time t will be expressed by the stochastic demand function $D\ (t, p, \xi)$ with noise parameter ξ. This assumption implicates myopic customer behavior.

Furthermore, we will not enclose any approach that considers demand between fare classes as dependent. The introduction of probability dependencies like passenger diversion or degradation costs (Botimer and Belobaba, 1999) would make the problem far more complex and too hard to compute. Variable costs of a Cloud Computing center are relatively low. Because servers and computers in general only have a minimal abrasion, the costs of one additional customer can be reduced to the energy consumption costs that increase because of the higher utilization. However, empirical studies have shown that an idle server needs nearly 90% of the average power consumption (Fan et al., 2007). Thus variable costs will be neglected in this paper. Consequently, with marginal costs being close or equal to zero, the marginal profit equals the turnover and so every sold product will contribute to the gained profit, regardless of the price.

3.2 Basic model

A common assumption is that for every time t there is at most one request from any customer. However, with Cloud Computing services being a time dependent service, the time interval of the offered product has to be predetermined. In the majority of the current business cases these are full hours (e.g. Amazon's EC2 product). To overcome this problem and to relax this assumption, a second time scale $t_r = 0, \ldots, T_r$ is introduced. Simultaneously a restriction is required for the number of arriving requests per time interval which will be set to R. Then, the connection between the product planning time t_p and the request time t_r can be defined as $t_t(t_r) = \left\lfloor \frac{t_r}{R} \right\rfloor = t_p$. Accordingly, the time horizon for the request time T_r can be formulated as $T_r = R\ T_p$. This term allows us, on the one hand, modeling the complex situation and, on the other hand, using the assumption that there is at most one request in every time interval Δt_r between to sequent points of t_r.

The capacity left for sale in t_r will be represented by the matrix $X_{t_r} = \left(x_{k,t_p}^{t_r}\right)$ for every request time t_r. As we reckon that initial resource capacity Γ_k of product k will not change over the time horizon T_p, we can define the initial capacity matrix for every resource k:

$$X_o = \left(x_{k,t_p}^{o}\right) \; with \; x_{k,t_p}^{o} = \Gamma_k \; \forall \; t_p \in \{1, \ldots, T_p\}.$$

The next important step is to unify a booking request. As we have stated above, the request has to include the amount of desired resources for every possible resource $k = 1, \ldots, K$. So, we define a vector $\mathbf{r}$ with elements r_k for every resource k. Additionally, we also need the information about the requested start time $s \in \{t_t(t_r)+1, \ldots, T_p\}$ and the duration of the service $d \in \{1, \ldots, T_p - s + 1\}$. Consequently, every submitted request has to contain the vector $\mathbf{r}$ as well as the variables s and d. The

[1] http://www.theenterprisecloud.com/

dependency of these variables is expressed in the request matrix R^{t_r} for every request time t_r as follows:

$$R_{t_r} = \left(r_{k,t_p}^{t_r}\right) = \begin{pmatrix} 0 & \cdots & 0 & r_{1,s} & \cdots & r_{1,s+d-1} & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & r_{K,s} & \cdots & r_{K,s+d-1} & 0 & \cdots & 0 \end{pmatrix}$$

Let $R_{t_r}^{t_p}$ denote the column t_p of R_{t_r}. This definition will be important for the model formulation and will even allow requests that have varying resource needs over time. The goal of Revenue Management is to maximize the profit. Therefore, let $V_t(X)$ be the expected revenue at time *t* and with the state of available resources *X*. Next, we present different models to determine the expected profit.

3.3 Static pricing

The first model that we will look at fixes all prices for the whole time horizon. Therefore, it has to set the price vector $\mathbf{p}^c = (p_0^c, \dots, p_k^c, \dots, p_K^c)^\top$ which contains prices for every resource *k* for every fare class *c*. Marbach introduced a similar problem which tries to optimize the general static pricing problem for a network service provider. However, they do not consider multiple resources (Marbach, 2004). Because prices are fixed for the whole time, the optimization for the static pricing model has to be done only once, more precisely before the first time interval of the time horizon. To accomplish this task, we have to solve the general optimization problem, depending on the price-response-function $D_c(t, p, \xi)$ for class c like it has been introduced above:

$$\max_{\mathrm{p_k^c}} V_{t_r}(X_{t_r}) = \sum_{t_{p=1}}^{T_p} \sum_{k=1}^{K} \sum_{c=0}^{C} p_k^c \; E\left[D_c(t_p, p_k^c, \xi)\right] \tag{1}$$

As long as we do not know anything about $D_c(t, p, \xi)$ yet, we assume that all prices p_k^c are given by the market and we have no possibility to change them at any point.

3.4 Bid price model

The bid-price model can be seen as an extension to the static pricing model introduced above (Goldman et al., 2002). They both have in common, that prices for every fare class and for every resource are fixed over the whole time horizon. Consequently, to find optimal prices, a similar optimization problem has to be solved (Williamson, 1992, Talluri and Ryzin, 1998). For every request matrix R^{t_r} that arrives at request time t_r with start point *s* and duration *d*, we can define the revenue that is gained by accepting this request as

$$\sum_{t_p=s}^{s+d-1} \left(p^{c\,\top} R_{t_r}^{t_p}\right) \tag{2}$$

In case the provider would accept a delivered request R^{t_r}, he will gain the profit like we have just seen, but simultaneously the available resource for the next request time t_r +1 will lower exactly by the amount of the request to $X_{t_r+1} = X_{t_r} - R_{t_r}$. So, the expected revenue for the remaining capacity X_{t_r+1} will be

$$V_{t_r+1}(X_{t_r+1}) = V_{t_r+1}(X_{t_r} - R_{t_r}). \tag{3}$$

However, the provider, who receives a request, does not have very much room for different strategies. He simply has to choose whether to accept or reject an incoming request. The decision is defined by the indicator variable $u \in \mathcal{U} = \{0,1\}$ with one for acceptance and zero for rejection. With the preceding formulas, we can formulate the Bellman equation similar to (Talluri and Ryzin, 1998) to find the optimal decision $u_{t_r}^*$

$$V_{t_r}(X_{t_r}) = \max_{u_{t_r} \in \mathcal{U}} \sum_{t_p=s}^{s+d-1} \left(p^{c\,\top} R_{t_r}^{t_p} u_{t_r}\right) + V_{t_r+1}(X_{t_r} - R_{t_r} u_{t_r}) \tag{4}$$

$$s.t.\; x_{ij} - r_{ij} \geq 0 \;\forall\, i, j$$

$$V_{T_r}(X) = 0 \ \forall X.$$

The bid-price approach is very intuitive by comparing immediately realizable revenue with their opportunity costs. Let $S_k^{t_p} = \left(s_{k,t_p}\right)$ a $K \times T_p$ matrix where the element s_{k,t_p} equals 1 and every other element equals 0. With the assumption that the function for the expected revenue $V_t(X)$ has a derivative $\frac{\partial}{\partial x_k} V_t(X)$ for the resource *k*, the approximated condition for accepting a request for a single resource *k* at time t_p with price p_k^c is (Talluri and van Ryzin, 2004):

$$p_k^c \geq V_{t_r}(X_{t_r}) - V_{t_r}\left(X_{t_r} - S_k^{t_p}\right) \approx \frac{\partial}{\partial x_k} V_t(X_{t_r}) =: \pi_{t_p}^k \ . \tag{5}$$

Next, (5) has to be extended for a whole request, including start time *s*, duration *d* as well as multiple resources *k*. This simply can be done by summing up all relevant prices for the requested fare class *c* and comparing it with the total sum of all relevant bid-prices for the given request. This approach allows us to compensate resources that may be sold under value with ones that will be sold for a higher price than expected. For pre-calculated bid-prices vectors $\boldsymbol{\pi}_{t_p} = (\pi_{t_p}^1, \dots, \pi_{t_p}^K)^\top$ we can reduce the necessary inequality that a request for fare class *c* has to satisfy to

$$\sum_{t_p=s}^{s+d-1} p^{c\top} R_{t_r}^{t_p} \geq \sum_{t_p=s}^{s+d-1} \boldsymbol{\pi}_{t_p}{}^\top R_{t_r}^{t_p} \ . \tag{6}$$

This condition ensures that the provider will only sell resources in fare classes that at least gain the revenue that he is expecting to gain. Consequently, an incoming request should be accepted if and only if the inequality (6) is fulfilled. We can formulate the optimal decision function as

$$u_{t_r}^* = \begin{cases} 1 & (6)\ is\ satified\ \ and\ \ x_{k,t_p}^{t_r} - r_{k,t_p}^{t_r} \geq 0 \ \forall\ k, t_p \\ 0 & otherwise. \end{cases} \tag{7}.$$

Because we only know the actual request at time t_r we cannot directly use the recursive definition of (4). Instead, we have to find a heuristic that uses another method to get the expected revenue of any remaining capacity *X*. In turn, the knowledge of the expected revenue together with (5) will enable the calculation of every necessary bid-price $\pi_{t_p}^k$.

For the following, we want to present one of several possibilities to value a certain state *X* of the available resources *k*. In any case, we need to model the expected demand to say anything about the valuation of the resources that are left. Thus, a demand function $\widehat{D}\ (k, t_p)$ is required that will return the estimated demand for resource *k* at time t_p. However, the valuation cannot exceed the available capacity stored in the vector $X_{t_r}^{t_p}$. and the demand function has to be adapted by

$$D\ (k, t_p) = \min\left[\widehat{D}\ (k, t_p), x_{kt_p}^{t_r}\right]. \tag{9}$$

Let $P_k^{t_p}(C\ |C = c)$ the probability of any demand for resource *k* at time t_p to equal fare class *c* *considering that higher valued classes have to be handled primarily to the disfavor of the lower valued ones*. The expected revenue can be calculated by

$$V_{t_r}(X_{t_r}) = \sum_{t_p=t_t(t_r)+1}^{T_p} \sum_{k=1}^{K} \sum_{c=0}^{C} p_k^c\ D\ (k, t_p)\ P_k^{t_p}(C\ |C = c) \tag{10}$$

$$s.t. \sum_{c=0}^{C} P_k^{t_p}(C\ |C = c) = 1 \ \forall\ k.$$

Algorithm 1 Transform $\widehat{D}\ (c)$ to observe capacity restrictions

Require: expected demand $\widehat{D}\ (c)$ for fare class c
Ensure: expected demand $D(c)$ under capacity restrictions

```
X ⇐ capacity left
for c = 0 to C do
    if X > D̂(c) then
        D(c) ⇐ D̂(c)
        X ⇐ X − D(c)
    else
        D(c) ⇐ X
        X ⇐ 0
    endif
endfor
```

3.5 Dynamic pricing

The following Pricing Model does not try to derive a benefit from denying requests for lower valued fare classes, instead it attempts to adjust prices for every fare class. Consequently, we can increase the price for a product to maximize profits rather than just limiting the capacity of a product and leaving the rest of the demand unused. The intuitive issue behind this approach is that fluctuations in demand can be compensated while simultaneously increasing revenues.

The frame of the optimization problem is very similar to the bid-price model of the preceding section. However, the big difference will be the variable that we are going to optimize. In both preceding models we had to decide whether to accept an incoming request or not, what was represented by the decision variable u. In this case, we will primarily accept any incoming request that will not exceed our remaining capacity, but we will influence demand by adapting the price. Consequently, the variable that we have to optimize is the price for every fare class c. As we will not allow every real number as a possible price for a fare class c, we assume that we have to select one element of the set $\mathcal{P}_{c,k}$, which contains all possible prices for fare class c and product k. The use of such a discrete set is very common in business practices. Especially conventional retailing industries often use price steps of at least one full monetary unit or in higher price regions even steps of integer multiples of those (Talluri and van Ryzin, 2004).

The difference to static pricing models is that we have to know something about the relation of price and demand to solve most optimization problems. Since we do not know the exact price-demand function, we assume that we have such a function $\widehat{D}\left(k, t_p, c\right)$ which returns the demand dependent on the price p. Analogous to the bid-price model, we have to transform this function with the algorithm 1 or with (9) to fulfill capacity restrictions. A probability distribution has to distribute the demand to the different fare classes (similar to the bid-price model). The revenue calculation also has to be slightly changed to include the price dependent demand function and different prices for every time interval. So the optimization problem that has to be solved is $\max_{p \in \mathcal{P}_{c,k}} p\ D\ (k, t_p, c)$. As this optimization problem has to be done for every resource k, fare class c and planning time t_p, we can formulate the Bellman equation that will recursively return the expected revenue:

$$V_{t_r}\left(X_{t_r}\right) = \sum_{t_p=s}^{s+d-1} \sum_{k=1}^{K} \sum_{c=0}^{C} \left(\max_{p \in \mathcal{P}_{c,k}} p\ D\ (k, t_p, c, p)\right) + V_{t_r+1}\left(X_{t_r} - R_{t_r}\right). \quad (11)$$

There are still two practical problems to directly use this term: On the one hand, we have to know or approximately assume a specific relation between price and demand. On the other hand, it is obvious that this calculation will be very complex, as we have to solve at least $d\ (C+1)\ K$ different optimization problems and we still do not know the valuation of the future time period t_{r+1}. Therefore, we want to present a heuristic that will adjust prices depending on the variables k, c and t_p. This heuristic will calculate a parameter γ for every resource k, time t_p and fare class c by setting a

specific relation between the expected demand $D\left(k, t_p, c\right)$, the capacity left x_{k,t_p} and the time that is left until the requested start time $s - t_t(t_r)$:

$$\gamma\left(k, t_p, c\right) := \frac{Expected\ Demand}{Capacity\ Left} \ln(Time\ To\ Go) = \frac{D\left(k,t_p,c\right)}{x_{k,t_p}} \ln(s - t_t(t_r)). \quad (12)$$

We assume that every parameter $\gamma\left(k, t_p, c\right)$ is positively correlated to the valuation of the customers and can serve as an approximation for the price-demand relation that we still do not know. Let $\mathcal{P}_{c,k} = \left\{\hat{p}_{k,1}^c, \dots, \hat{p}_{k,P}^c\right\}$ the set of possible prices for fare class *c* and resource *k* with the ascending order $\hat{p}_{k,1}^c > \hat{p}_{k,2}^c > \cdots > \hat{p}_{k,P}^c$. We can define the function

$$p\left(k, t_p, c\right) := \begin{cases} \hat{p}_{k,1}^c & \gamma\left(k, t_p, c\right) > \tau_1(k,c) \\ \hat{p}_{k,2}^c & \tau_1(k,c) \geq \gamma\left(k, t_p, c\right) \geq \tau_2(k,c) \\ \vdots & \vdots \\ \hat{p}_{k,P-1}^c & \tau_{P-1}(k,c) \geq \gamma\left(k, t_p, c\right) \geq \tau_P(k,c) \\ \hat{p}_{k,P}^c & otherwise. \end{cases} \quad (13)$$

to get the proper price for resource *k*, time t_p and fare class *c* with the threshold values $\tau_i(k,c)\ \forall\ i \in \{1, \dots, P\}$. We assume that all $\tau_i(k,c)$ will be defined manually at the beginning of our time horizon, primarily based on empirical knowledge. A possible orientation might be the expected value of the parameter $\gamma\left(k, t_p, c\right)$, so every $\tau_i(k,c)$ might be expressed as

$$\tau_i(k,c) := \alpha_i\ E\left[\gamma\left(k, t_p, c\right)\right] \quad (14)$$

with the freely chosen real multiplier $\alpha_i > 0$ and the expected value $E\left[\gamma\left(k, t_p, c\right)\right]$ of γ for every t_p.

4 EVALUATION

4.1 Scenario

As requirements mostly cannot be foreseen even by the customers, we will only allow requests for a fixed amount of every resource *k* for a certain time period $\Delta t = d$. Forecasting of future demand and utilization are essential for overbooking decisions in the specific Cloud Computing case. Since this task is not in the focus of this paper, so some relatively simple, but intuitive approaches will be applied. Demand is independently distributed over time, so a very intuitive step is to use historic demand data to estimate future demand. To eliminate smoothing effects, not all the historical data will be considered, but a limitation to the last N_D values. In this case, the historical data consists of the observed demand $\overline{D}(k,t,c)$ for product k, time t and fare class c. The expected demand at the request time t_r for every future time $t_p \geq t_t(t_r)$ can be expressed as

$$D_{t_r}(k,c) = \frac{1}{N_D} \sum_{t=t_t(t_r)-N_D}^{t_t(t_r)-1} \overline{D}(k,t,c) \quad (15)$$

where the last N_D values of the observed demand are taken into account. The variable N_D might be increased for a smoother demand forecasting or decreased for a quicker adaptation to market changes.

For utilization, we will assume that the utilization behavior will change over the time of day, while the progress will be similar from day to day. For overbooking decisions in a Cloud Computing environment, we have to set some kind of a service level which indicates that controls how many resources may be additionally sold. Let $\overline{U}_n(t,k)$ the recognized usage of costumer n for the product k at time t. With N_U last values and resource k, we can define the average value and the variance of the past utilization as

$$\mu_{k,t_r} := \frac{1}{N_U}\sum_{t=t_t(t_r)-24\cdot N_U}^{t_t(t_r)-24}(\bar{U}(k,t)\cdot I(t,t_r)) \tag{16}$$

$$\sigma^2_{k,t_r} := \frac{1}{N_U}\sum_{t=t_t(t_r)-24\cdot N_U}^{t_t(t_r)-24}\left(\left(\mu_{k,t_r}-\bar{U}(k,t)\right)^2\cdot I(t,t_r)\right)$$

$$s.t.\, I(t,t_r) := \begin{cases} 1 & t \bmod t_t(t_r) - \left\lfloor \frac{t_t(t_r)}{24} \right\rfloor \cdot 24 \equiv 0 \\ 0 & otherwise. \end{cases}$$

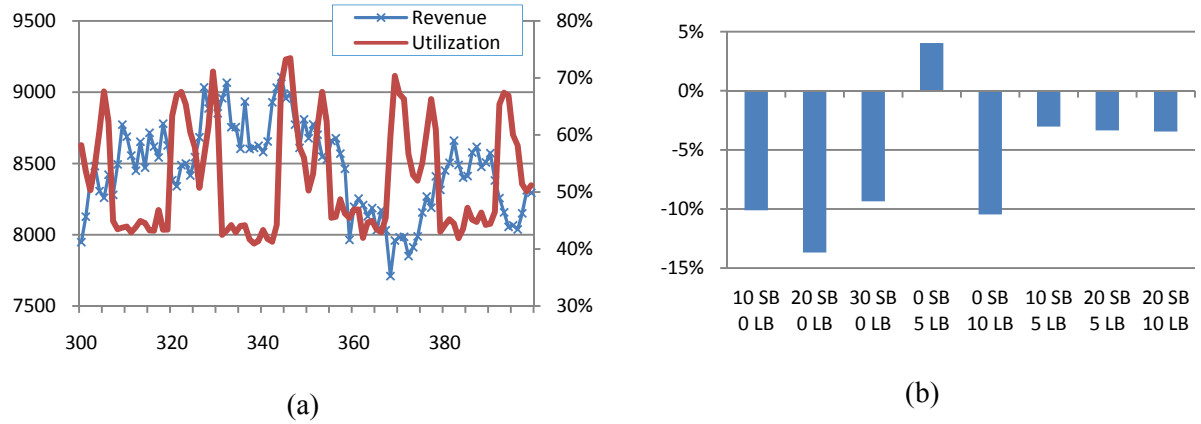
With a predefined service level SL and the assumption that the utilization of every product follows a normal probability distribution, we can solve the following equation and adapt the available capacity accordingly to ensure that we can provide our services with the probability of SL:

$$P(\hat{u} \le SL) = \frac{1}{\sigma_{k,t_r}\sqrt{2\Pi}}\int_{-\infty}^{\hat{u}} e^{-\frac{1}{2}\left(\frac{t-\mu_{k,t_r}}{\sigma_{k,t_r}}\right)^2} dt \le SL \tag{17}$$

We have already defined the behavior of the sellers by formulating the different pricing models above, but furthermore, we have to explicitly determine the buyers' behavior. We will implement two different types of buyers, namely *short term buyers* who tend to requests the product last-minute and *long term buyers* who are more likely to reserve earlier (Shen and Su, 2007). The amount of delivered workloads over the course of a day is modeled according to (Calzarossa and Serazzi, 1985).

4.2 Simulation results

The first simulation set tested the performance of each seller in a monopolistic environment under different demand situations (e.g. 10 short-term buyers). The planning horizon T_p of every seller has been set to 1500 what approximately equals two month when one tick is considered as a full hour. While providers have no information about demand and utilization at the beginning of each simulation run, the first ticks are not very representative, so only the time span from 300 to 1020 has been considered for the evaluation. This time span contains 720 ticks or hours what equals one month. For utilization and demand forecasting, parameters have been set to N_D = 50, N_U = 5 and SL = 99.5%. Three possible fare classes were defined with the price vector p_1^c used by the static and bid-price seller for resource 1 (Table 2). Other resources are adapted according to the relation used in the Amazon EC2. In contrast to the other models the dynamic seller needs to have a price matrix which includes prices for every resource and fare class. For the simulation, the prices have been chosen in accordance with the prices for the static models with additional ones below and above. Additionally, the threshold values τ have to be set to define price change behavior for every price class c and resources k (Table 3, setting I).



(a) (b)

Figure 1. Utilization and revenue of a dynamic provider facing20 short and 5 long term customer for 100 timeslots (excerpt for t = {300, …, 400}) (Figure a) and difference between simulation setting I and II (Figure b).

$p_1^c = \begin{cases} 0.3 & c = 0 \\ 0.2 & c = 1 \\ 0.1 & c = 2 \end{cases}$	$\mathcal{P}_{c,1} = \begin{cases} \{0.2, 0.25, 0.3, 0.5, 0.9\} & c = 0 \\ \{0.15, 0.18, 0.2, 0.3, 0.5\} & c = 1 \\ \{0.08, 0.09, 0.1, 0.2, 0.3\} & c = 2 \end{cases}$

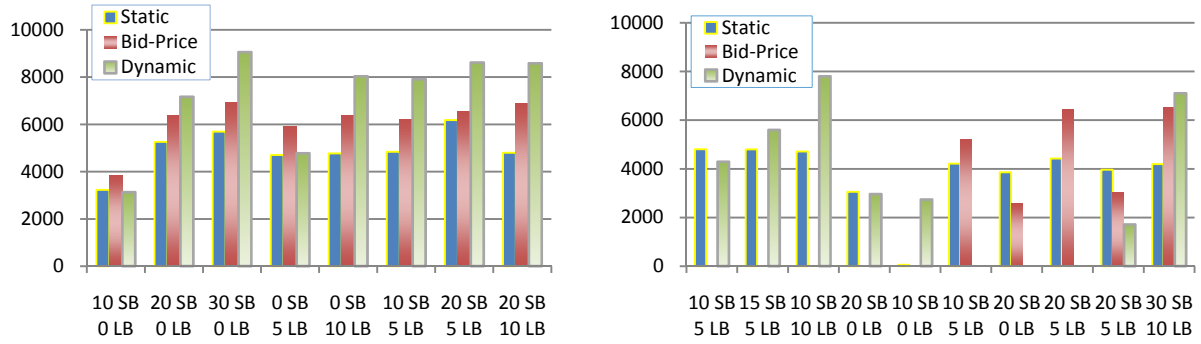
Table 2. Price vector and price matrix for bid-price and dynamic price sellers.

It can be observed that the dynamic seller performs particularly well in cases where demand is high. Even when demand is low, the dynamic seller achieved some acceptable outcome. The worst case, when there were only ten short term buyers, the dynamic seller had only 3.0 % lower revenue than the static seller. In the best case, he even gained 78.9 % more revenue. Also the bid-price seller performed surprisingly good and achieved as expected always higher revenue than the static seller, even when demand was low. In these cases, he even outran the dynamic seller (Figure 4a). It can be summarized that that both revenue management adaptations performed very solid in the monopolistic environment and managed to significantly increase revenues in nearly any cases. However, the threshold values τ for the dynamic seller have been adjusted on simulation results. When we chose to slightly change these values for every price class *c* and resources *k* (Table 3, setting II), the dynamic seller behaves different and results in considerably decreased revenues in most cases (see Figure 1b). Consequently, an optimally or nearly optimal adjustment of the dynamic seller has to be done what may be difficult and expensive to achieve in some cases. Figure 1a illustrates the utilization and revenue trend of a dynamic seller. The time period is exemplarily set to 300...400. We can also observe how the utilization over the time of day changes.

$\tau_i(k, c) = \begin{cases} 0.8 & i = 1 \\ 0.4 & i = 2 \\ 0.2 & i = 3 \\ 0.1 & i = 4 \end{cases}$	$\tau_i(k, c) = \begin{cases} 1.0 & i = 1 \\ 0.6 & i = 2 \\ 0.4 & i = 3 \\ 0.2 & i = 4 \end{cases}$
Simulation setting I	Simulation setting II

Table 3. Price vector and price matrix for bid-price and dynamic price sellers.

The second simulation scenario was based on similar parameter, however, the sellers contend with each other for the potential customers. Tests have shown that if demand is low, so the available capacity cannot be sold completely, the dynamic seller has the advantage of lowering prices. Consequently, a myopic and totally informed customer will always choose the dynamic seller if capacity is available. However, this is only the case if prices have been adjusted correctly. This requirement is important though a suboptimal adjusted dynamic seller may also ask too expensive prices so he will not get any customer at all. Figure 4b shows that the dynamic seller outperformed the static model when demand was low or high. With 10 short and 5 long term buyers or only 20 short term buyers the dynamic seller did not have any advantage at all, but still managed to increase the revenue per utilized resource and to simultaneously decrease the utilization variance. The bid-price seller showed his main advantages in high demand situations what fits the theory as closing of fare classes makes no sense in most underutilized scenarios.



(a) (b)

Figure 4. Average revenue of one provider in a market with short term (SB) and long term (LB) buyers (Figure a) and a competitive market with two and three sellers respectively (Figure b).

5 CONCLUSION

Cloud providers like Amazon, Google or Rackspace have to consider to whom which products are offered at a certain time to increase revenue, when resources are scarce. Future incoming requests are oftentimes unpredictable. Revenue Management concepts enable to implement a decision policy to accept or reject requests and to influence the behavior of the consumers. Price sensitive consumers will shift their demand to low-demand time, when prices are low as well. Depending on the utilization of his resources the provider can set prices or decide about accepting a request.

At first we analyze if revenue management requirements are fulfilled by a Cloud Computing scenario. The scenario comprises several products using the same pool of multiple resources. Then, we compare dynamic pricing and bid price control according their revenue outcome. In contrast to traditional Revenue Management approaches, we used a matrix to interpret a request and introduced a second time scale which allows multiple requests in one interval of the planning horizon. The proposed model allows the service providers to decide about accepting or rejecting incoming service requests based on expected revenue, expected demand and a given set of resources.

In future, the modeling of buyer according to the customer choice behavior is an interesting challenge. Then, dependencies between fare classes have an impact on the revenue. Customers, who are not allocated, could probably decide to change to a higher fare class which increases the revenue. Furthermore, the bid price as well as the dynamic algorithm can be optimized by more complex heuristics and learning algorithms.

References

Peter Paul Belobaba. Air travel demand and airline seat inventory management. PhD thesis, 1987.

Gabriel Bitran and Rene Caldentey. An Overview of Pricing Models for Revenue Management. Manufacturing & Service Operations Management, 5(3):203–229, 2003.

Greg Boss, Padma Malladi, Dennis Quan, Linda Legregni, and Harold Hall. Cloud Computing. IBM High Performance On Demand Solutions. Technical report, 2008.

Theodore Botimer and Peter Belobaba. Airline Pricing and Fare Product Differentiation: A New Theoretical Framework. The Journal of the Operational Research Society, 50(11):1085–1097, 1999.

Rajkumar Buyya, Chee Shin Yeo, and Srikumar Venugopal. Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities. High Performance Computing and Communications, 2008. HPCC '08. 10th IEEE International Conference on, pages 5–13, 2008.

Maria Calzarossa and Guiseppe Serazzi. A Characterization of the Variation in Time of Workload Arrival Patterns. IEEE Transactions on Computers, 34(2):156–162, 1985.

Raman Desiraju and Steven Shugan. Strategic Service Pricing and Yield Management. Journal of Marketing, 63(1): 44-56, 1999.

Parijat Dube, Yezekael Hayel, and Laura Wynter. Yield management for IT resources on demand: analysis and validation of a new paradigm for managing computing centres. Journal of Revenue and Pricing Management, 4(1):24–38, 2005.

Xiaobo Fan, Wolf-Dietrich Weber, and Luiz Andre Barroso. Power provisioning for a warehouse-sized computer. In ISCA '07: Proceedings of the 34th annual international symposium on Computer architecture, pages 13–23, New York, NY, USA, 2007. ACM.

Ian Foster, Carl Kesselman, and Steven Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal of Supercomputer Applications, 15:2001, 2001.

Paul Goldman, Richard Freling, Kevin Pak, and Nanda Piersma. Models and techniques for hotel revenue management using a rolling horizon. Journal of Revenue and Pricing Management, 1(3), 2002.

Brian Hayes. Cloud computing. Communications of the ACM, 51(7):9–11, July 2008.

Sheryl Kimes. Yield management: A tool for capacity-constrained service firms. Journal of Operations Management, 8(4):348–363, 1989.

George Lawton, “Moving the OS to the Web,” Computer, 41(3): 16–19, 2008.

Ken Littlewood. Forecasting and control of passenger bookings. In 12th AGIFORS, pages 95–117, 1972.

Peter Marbach. Analysis of a static pricing scheme for priority services. Networking, IEEE/ACM Transactions on, 12(2):312–325, April 2004.

Jeffrey McGill and Garrett van Ryzin. Revenue Management: Research Overview and Prospects. Transportation Science, 33(2):233–256, 1999.

Serguei Netessine and Robert Shumsky. Introduction to the Theory and Practice of Yield Management. INFORMS Transactions on Education, 3(1):34–44, 2002.

Zuo-Jun Max Shen and Xuanming Su. Customer Behavior Modeling in Revenue Management and Auctions: A Review and New Research Opportunities. Production and Operations Management, 16(6):713–728, 2007.

David Skillicorn, “The Case for Data-Centric Grids,” Proc. IEEEInt’l Parallel and Distributed Processing Symp., pp. 247–251, 2002.

Kalyan Talluri and Garrett van Ryzin. An Analysis of Bid-Price Controls for Network Revenue Management. Management Science, 44(11):1577–1593, 1998.

Kalyan Talluri and Garret van Ryzin. The Theory and Practice of Revenue Management. Springer, New York, 2004.

Bhuvan Urgaonkar, Prashant Shenoy, and Timothy Roscoe. Resource overbooking and application profiling in shared hosting platforms. In OSDI '02: Proceedings of the 5th symposium on Operating systems design and implementation, pages 239–254, New York, NY, USA, 2002. ACM.

Lawrence Weatherford and Samuel Bodily. A taxonomy and research overview of perishable-asset revenue management: yield management, overbooking, and pricing. Operations Research, 40(5):831–844, 1992.

Christof Weinhardt, Arun Anandasivam, Benjamin Blau, Jochen Stoesser. Business Models in the Service World. In IEEE IT Professional, Special Issue on Cloud Computing, 11(2):28-33, 2009.

Aaron Weiss. Computing in the clouds. netWorker, 11(4):16–25, 2007.

Elizabeth Williamson. Airline network seat control. PhD Thesis, 1992.