Association for Information Systems
AIS Electronic Library (AISeL)

ECIS 2012 Proceedings

European Conference on Information Systems (ECIS)

5-2-2012

CLOUD SERVICE REVENUE MANAGEMENT

Tim Püschel
University of Freiburg

Guido Schryen
University of Regensburg

Diana Hristova
University of Regensburg

Neumann Dirk
University of Freiburg

CLOUD SERVICE REVENUE MANAGEMENT

Püschel, Tim, University of Freiburg, Kollegiengebäude II, Platz der Alten Synagoge,79085 Freiburg, Germany, tim.pueschel@is.uni-freiburg.de

Schryen, Guido, University of Regensburg, Universitätsstraße 31,93053 Regensburg, Germany, guido.schryen@wiwi.uni-regensburg.de

Hristova, Diana, University of Regensburg, Universitätsstraße 31,93053 Regensburg, Germany, diana.hristova@ wiwi.uni-regensburg.de

Neumann, Dirk, University of Freiburg, Kollegiengebäude II, Platz der Alten Synagoge,79085 Freiburg, Germany, dirk.neumann@is.uni-freiburg.de

Abstract

Successful Internet service offerings can only thrive if customers are satisfied with service performance. While large service providers can usually cope with fluctuations of customer visits retaining acceptable Quality of Service, small and medium-sizes enterprises face a big challenge due to limited resources in the IT infrastructure. Popular services, such as justin.tv and SmugMug, rely on external resources provided by "cloud computing" providers in order to satisfy their customers' demands at all times. The paradigm of cloud computing refers to the delivery model of computing services as a utility in a pay-as-you-go manner. In this paper, we provide and computationally evaluate decision models and policies that can help cloud computing providers increase their revenue under the realistic assumption of scarce resources and under both informational certainty and uncertainty of customers' resource requirement predictions. Our results show that in both cases under certainty and under uncertainty applying the dynamic pricing policy significantly increases revenue while using the client classification policy substantially reduces revenue. We also show that, for all policies, the presence of uncertainty causes losses in revenue; when the client classification policy is applied, losses can even amount to more than 8%.

Keywords: cloud computing, revenue management, fuzzy optimization

1 Introduction

Recent developments in electronic business (e-business), distributed systems and communications technology have resulted in the emergence of cloud computing, a new paradigm that offers a variety of E-business service innovations. This new paradigm refers to the delivery model of computational e-business services as a utility in a pay-as-you-go manner. Dedicated cloud providers such as Amazon offer services at different levels ranging from Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-service (SaaS). IaaS refers to the model where hardware resources such as CPU, storage and bandwidth are made available over a network, typically the Internet. PaaS offers not only services on the infrastructure level, but also provides the whole platform including the operating system and associated services over the Internet. Lastly, SaaS denotes the delivery model where software being hosted by the SaaS provider is made available to the customers. Technically, all those novel delivery models can be combined so that the distinction into those pure categories becomes academic.

As cloud computing infrastructures gradually mature – exemplified through a number of commercial offerings, such as Amazon’s EC2 and S3, CycleComputing, and SalesForce.com, – and academic projects, such as Nimbus, Eucalyptus, and OpenCirrus, the benefit of such an infrastructure becomes apparent to various user communities. Academic users, for instance, recognize the benefits of scaling their campus-wide infrastructure using additional capacity, on-demand, from external vendors. Similarly, commercial users recognize the benefits of outsourcing some of their data storage and computation requirements, thereby saving on the overall cost of ownership and operation of their IT infrastructure, while maintaining their Service Level Agreements (SLAs). With the emerging demands on minimizing operational costs, especially on energy, core cloud computing technologies such as (i) virtualization; (ii) elastic IP address allocation; and (iii) storage clustering, have also found favor independently. As the quantity of user generated content as well as the need for computation is literally exploding, the management of cloud services is becoming a challenge. Three key challenges, however, remain – and are likely to have a significant impact on the commercial uptake of cloud computing technologies for offering e-Business services:

- Designing interfaces and protocols that allow interoperability among different cloud systems. Currently, clouds are isolated such that applications can access only one cloud at a time.
- Resolving issues of privacy and security as well as piracy within a cloud.
- Designing economic models that balance the dynamic costs of provisioning on the cloud service provider side and the consumer benefit on the user side.

While the first two key challenges have been accepted for quite a while, the third issue has quite recently been raised (Buyya, 2009). Cloud service providers will offer their services only if they can realize sufficient profit. This can become complicated taking demand spikes into consideration, which are associated with clouds as these can be accessed from anywhere in the world at any time. Thus, cloud service providers continuously need a concise real-time view on their revenue streams and costs in order to remain successful. Recent work has suggested the application of revenue management techniques for cloud computing. As a rationale, these papers argue that cloud services are comparable in nature with airplane seats. The capacity is fixed in the short run and needs to be allocated to different groups of clients with different willingness to pay. The decision space of the cloud service providers includes two factors: the price at which services are offered and the decision whether an actual service request is accepted.

In this paper, we address the research question of how we can increase revenue by adapting admission control for cloud service providers both under certainty and uncertainty. We introduce novel service acceptance models, which maximize the revenue of the cloud service providers with respect to the acceptance decision, thus contributing to service management innovation. We advance current research in revenue management by explicitly incorporating uncertainty of required job resources into our acceptance decision. Typically, customers have only vague estimates with respect to the resources that are needed for running the job on the cloud. Classical forecasting models have failed to predict the resources adequately due to the inherent complexity. In our approach the cloud service provider has no longer reserve the maximum amount of resources but can include this uncertainty into his optimization. The main contributions of our paper are threefold:

- We provide and evaluate decision models that can help service providers increase revenue.
- Unlike any other related approach, our model can take uncertainty with respect to resource requirements into account.
- We show how this uncertainty can affect revenue, thus helping providers to estimate the cost related to it.

The remainder of this paper is structured as follows: In the second section we present the theoretical background, define requirements on our service acceptance models, and we discuss how related work meets these requirements. In the third section, we present optimization models. The fourth section comprises a computational evaluation, where we test our models with respect to their goodness. The fifth section concludes the paper with a summary and an outlook over new research avenues.

2 Theoretical Background

As previously mentioned, the key question for providers in the short run is how to decide whether a job is accepted or not. The standard approach would be to always accept the job if there is enough capacity available to fulfil the requirements. Typically, capacity is allocated on a first come first serve basis. However, this approach does not deliver an optimal allocation with respect to the cloud provider's revenue. Thus, it is necessary to take a closer look at the requirements and the basic underlying decision problem that cloud providers face.

2.1 Motivational scenario

A service provider offers a number of different cloud services on the market, similar to Amazon.com, Salesforce.com or mor.ph. Most of these services are accessible within very short time span - so that there is no long time to deploy the service and some further advertise certain annual availability percentages. Amazon for example names an annual uptime of at least 99.95%. The service is considered unavailable if the running instances "have no external connectivity during a five minute period and you are unable to launch replacement instances". Naturally, each service has a different profile of resource requirements. Some service are computationally intensive, others require large amounts of storage or bandwidth. Currently, the cloud provider has no mechanisms available to adequately deal with utilization approaching maximum capacity. Thus, all incoming jobs are accepted as long as there is capacity left. However, as demand for the different services varies greatly in situations where one type of resources (e. g. CPU) operate at nearly full capacity while in other situations there is still enough buffer left. The cloud provider typically operates a separate infrastructure that provides the same services to either internal users or key customers. This separation of the infrastructure helps to guarantee that these users can always access a certain amount of resources, but also leads to inefficiencies, e.g. in cases where the internal infrastructure is fully used, but the infrastructure for external customers still has capacity left or vice-versa.

2.2 Requirements

The motivational scenario already addresses some issues a decision model for service admission control has to solve. One issue refers to volatile demand for different services which results in high variations in system utilization. To avoid high utilization requirements due to demand peaks we need to have a mechanism that provides customers with incentives to shift their demand to times of low utilization. A market-oriented or dynamic pricing mechanism is necessary to achieve a balancing of demand peaks (Yeo and Buyya, 2004). Effective mechanisms need to work in online and near time scenarios; accordingly the mechanisms need to exhibit very low computational cost.

One key requirement for customers is Quality of Service (QoS) (Buyya et al., 2009). While some customers accept best-effort based services, a large share of customers considers the fulfilment of adequate service levels a key factor in deciding which providers to choose. Overload situations can lead to reduced overall performance (Nou et al., 2007) and thereby can result in breaking service level agreements between the provider and clients. Thus, it is necessary to have a mechanism that ensures that jobs will not be accepted if they entail an overload situation. Other aspects of QoS cover assuring service levels are kept during normal operation and in case of (partial) resource failure. Since we focus on the admission problem only the first aspect is relevant.

In order to better integrate different groups of clients with varying requirements in a cloud market, it is beneficial to be able to offer different service levels and have some kind of client classification. Certain classes of clients or jobs can have different privileges. According to these privileges, the jobs can be discriminated regarding various factors such as acceptance, pricing (Newhouse et al., 2004), different risk or service levels (Djemame et al., 2006), priority of the running jobs, and reservation of resources for certain clients when the system approaches high utilization. A cloud that is able to accommodate users from different classes promises better efficiency and may also be able to make better use of economies of scale. In general, client classification allows better management of long-term relationships with strategically important customers by considering client classification for the calculation of prices, scheduling of jobs and giving customers further privileges. Furthermore it can enable the provider to improve revenue management, as skimming off customer surplus is facilitated.

2.3 Theoretical optimum

As for many other enterprises, the key objective of a service provider is revenue maximization. If we assume that the provider has perfect information and certainty about future events it is possible to calculate the revenue maximizing solution. This necessary information includes incoming service requests, prices, the exact resource requirements of each job and capacity available in the future. In this case a simple instance of this problem would be:

$$\max_{x} \sum_{j \in J} x_j * fp_j \tag{O1}$$

subject to:

$$\sum_{m=1}^{|J|} c_{mr}(t) * x_m \leq c_r(t) \quad \forall t \in T, \forall r \in R \tag{C1.1}$$

where T is the set of all regarded time slots; J is the set of available jobs; R is the set of all resource types; fp_j is the price paid for job j; x_j is a binary allocation variable indicating whether job j was accepted or rejected; $c_{jr}(t)$ is the capacity of resource type r required by job j in time slot t; and $c_r(t)$ is the total capacity available for resource type r during time slot t. (O1) is the objective function and represents the achieved revenue. The constraint (C1.1) is the capacity constraint. It assures that not more capacity can be allocated than is available. In addition to the capacity constraints further requirements could exist, e. g. the requirement to have a certain amount of resources available for a certain client.

This maximization problem can be formulated and solved as a linear program. However it is an instance of the knapsack problem (Kellerer et al., 2004). As such it is NP-hard and therefore computationally intractable. In our scenario the job or service requests represent the items used in the knapsack problem. The job prices represent the valuations and the capacity needed by a service represents its weight. The available capacity represents the weight restriction. Therefore any mechanism which can determine the optimal solution to the service request acceptance problem can also be used to solve the knapsack problem, which is NP-hard. Since this problem has to be solved for each time-slot and the solutions for the knapsack problems in different timeslots are interrelated it is an instance of the temporal knapsack problem (Bartlett et al., 2005). Early on scientists started developing heuristics or approximations to find solutions to the knapsack problem in polynomial time (Ibarra and Kim, 1975).

One option would be to use such an approximation. However, to solve the problem, whether with an exact algorithm or with a heuristic, it is still necessary to have sufficient information on the available jobs, the capacity and runtime they require, the prices, etc. In general this assumption, nevertheless, cannot be made. Instead of trying to predict the exact job information and then running approximation heuristic to solve the knapsack problem with predicted information, a different approach is pursued in this work. The idea is to use policies as a rule of thumb for directly maximizing the revenue without having the exact information about future jobs. These policies can be based on requirements from contracts with clients, company policies and information gained from historic workload traces, utilization curves, prices, etc. Policies can also be based on information about the statistical distribution of job size, runtime and prices. The policies are used for real-time allocation as an additional requirement to the capacity requirement when determining whether to accept a job or not.

2.4 Related Work

There are three different research streams dealing with the admission and management of job requests. The first stream stems from the research area of computer science and focuses on the technical aspects while incorporating basic economic principles such simple market mechanisms. Ferguson et al. (Ferguson et al., 1996), discuss the general applicability of economic theories to resource management. Being a primer, their results are rather general, not specifically associated with actual implementations. An interesting approach to realize high service levels and end-to-end Quality of Service is the Globus Architecture for Reservation and Allocation (Foster et al., 1999). This approach uses advance reservations to guarantee QoS. A very related approach to achieve autonomic QoS aware resource management is based on online performance models (Kounev et al., 2007). They introduce a framework for designing resource managers that are able to predict the impact of a job in the performance and adapt the resource allocation in such a way that SLAs can be fulfilled. Elements of client classification such as price discrimination based on customer characteristics have been address by Newhouse et al. (2004). The authors - albeit technically motivated - move in the direction of applying methods from revenue management. However, they do not consider other discrimination factors, such as priority on job acceptance or higher quality of service. Aiber et al. (2004) present an architecture for autonomic self-optimization based on business objectives. Apparently, the focus is on the technical implementation of economic principles not on the employed principles as such. Boughton et al. (2006) present research on how workload class importance can be considered for low-level resource allocation. They focus on competing workloads in databases and investigate what business policies - directing the relative importance of workloads - can be used to efficiently allocate resources.

A second stream stream aims to adapt more sophisticated concepts and from revenue management and admission problems in service environments. Ramalho (1998) examines the application of fuzzy logic to the CAC (Connection Admission Control) traffic control function in ATM (Asynchronous Transfer Mode) broadband communication networks. Based on observations of the traffic in the ATM link, the paper derives a Fuzzy Logic Based CAC (FCAC) for the maximum cell loss ratio from adding a new connection to a given traffic scenario under uncertainty. The use of concepts from Revenue Management for Internet Service Providers was researched by Nair and Bapna (2001). They considered the decision to accept or reject customers but did not take different service types or advanced reservation into account. An approach to measurement-based admission control for multiclass networks with link sharing for "applications with ill specified traffic characteristics" using adaptively measured maximal rate envelopes is proposed by Qiu and Knightly (2001). By assuming that future packet arrivals will not exceed past maximal rate envelop they develop a method that among other indicators reflects the uncertainty of the prediction of future workloads and allows to control important QoS parameters such as loss probability. Poggi et al. (2009) propose an architecture for admission control on e-commerce websites that prioritizes user sessions based their intentions to buy a product. They analyze customer behavior, such as navigational clicks, on an e-commerce web site. Based on these behavior predictions about the user's the intention to buy a product can be inferred. Consequently, Quality of Service for user sessions is shaped based on user's intentions.

The third stream deals with Revenue Management and admission control in Utility Computing environments. Yeo and Buyya (2004) show an approach for a pricing function depending on a base pricing rate and a utilization pricing rate. The idea behind utilization based pricing is that when utilization is high, demand for the resources is high as well and therefore customers are willing to pay higher prices for resources. If utilization is low, lower prices are charged to attract more customers. One of the first papers analyzing revenue management concepts for cluster systems was published by Dube et al. (2005). The suggested model offers one resource for different prices. By assuming the customer behavior follows a logit model, the authors analyzed an optimization model for a small number of price classes and provided numerical results. A framework for the application of revenue management in the field of Grid Computing was introduced by Anandasivam and Neumann (2009). They propose a theoretic model for revenue management that complies with certain requirements for pricing of bundles of resources.

The related work covers different aspects of related research problems. However, the proposed mechanisms are only applicable to the cloud market to some extent and an overall decision model taking the characteristics and requirements of the Cloud market as well as the effect of uncertainty into account is still missing. To address this issue we present a novel service request acceptance model in the following section.

3 Service Request Acceptance Models

In this paper we consider two groups of cloud service admission control models. While all models of the first group assume that resource requirements are certain, the models of the second group assume that customers' predictions of the resources required by a job are subject to uncertainty. The intra-group difference between the models of each group is based on the criterion which is used to make the decision of whether or not a job is accepted. The intra-group distinction is profit-oriented and is thus endogenous, while the inter-group distinction reflects the exogenous (un)certainty of information.

3.1 Basic Model and Policies

The mentioned features allow the provider to include important factors when taking the decision which service requests to accept. However, it still needs a decision model to help take the actual decision which jobs to accept. To this end the following basic policies are introduced. They represent the some of the basic options, such as dynamic pricing with reservation prices, and client classification with strict priority. The reservation price denotes the minimum price at which the provider is going to sell the service.

A simple first come first served policy (I) is used as a benchmark. Any incoming job is accepted if there is enough capacity available. This type of policy represents a simple system without any enhancements. The following mathematical formulation can be used to represent the FCFS Policy:

$$\max_{x} \sum_{j \in J} \left(\frac{1}{2} * x_j \right)^{j} \tag{O2}$$

subject to:

$$\sum_{m=1}^{|J|} c_{mr}(t) * x_m \leq c_r(t) \quad \forall t \in T, \forall r \in R \tag{C2.1}$$

where T is the set of all regarded time slots; J is the set of available jobs; R is the set of all resource types; x_j is a binary allocation variable indicating whether job j was accepted or rejected; $c_{mr}(t)$ is the capacity required by job j in timeslot t for resource r; and $c_r(t)$ is the total capacity available for resource type r during time slot t. The objective function (O2) is a generating function which is used to represent the sequential nature of the policy. (C2.1) represents the capacity constraints which assure that not more capacity is allocated than is available for each resource type, thus fulfilling the requirement of not degrading QoS due to the acceptance of too many jobs. The two subsequent policies use the same generating function and capacity constraint but add further constraints.

The second policy (II) uses dynamic pricing based on different reservation prices for different utilization levels.

$$\sum_{l=1}^{r}\sum_{k=1}^{n}\left[H_1\left(U_l(t)-l_{k-1}\right)-H_1\left(U_l(t)-l_k\right)\right]$$

$$*c_{jl}(t)*rp_{k-1}*x_j \leq \frac{fp_j}{s_j}\forall t \in T, \quad \forall j \in J \qquad \text{(C2.2)}$$

(C2.2) introduces utilization based pricing based on a certain reservation price p_n that has to be achieved once the utilization surpasses the threshold l_n. It represents a number of constraints, depending on the number of reservation prices and thresholds. For each resource type the highest applicable reservation price per unit is chosen and multiplied with the resources needed. The reservation price per time slot for the job is then compared to the actual price per time slot. The variable fp_j represents the price for job j. s_j is the runtime of job j. $U_l(t)$ describes the utilization of resource l in time slot t and is defined as followed:

$$U_l(t) := \frac{1}{c_l(t)} * \sum_{m=1}^{|J|} c_{ml}(t) * x_m$$

$H_1(n)$ is the Heavyside step function. $H_1(n)$ is 0 for $n < 0$ and 1 for $n \geq 0$.

Client classification with strict priority on job acceptance is the proposed policy III:

$$(1-cc_j) * \frac{\sum_{m=1}^{|J|} c_{mr}(t) * x_m}{c_r(t)} \leq l_c \quad \forall t \in T, \forall r \in R, \forall j \in J \qquad \text{(C2.3)}$$

Constraint (C2.3) assures that once a certain utilization threshold l_c is exceeded only jobs from important customers, further on called gold clients, who have the binary client classification variable cc_j set to 1 are accepted.

The basic model ensures that not more capacity is allocated than available, it features dynamic pricing and client classification. Furthermore it has very low computational cost. It requires however that customers deliver exact information about their capacity requirements.

3.2 Extended Model and Policies

Cloud providers can face information uncertainty in many respects. Instances of uncertainty can occur when impreciseness is present in the description or estimation of the exact amount of resource the execution of a job or delivery of a service will need. This impreciseness is due to a lack of information, belief, and linguistic characterizations, which all are deemed some of the most important roots of uncertainty (Zimmermann, 2000). Accounting for particular types of uncertainty being prevalent for Cloud service providers, we select fuzzy set theory (Zadeh, 1965) among the available uncertainty theories.

The appropriateness is particularly based on the fact that *“a (fuzzy set theory based) framework provides a natural way of dealing with problems in which the source of imprecision is the absence of sharply defined criteria of class membership rather than the presence of random variables.”* (Zadeh, 1965). Fuzzy set theory generalizes traditional set theory in such a way that it provides for a degree of membership with which an element belongs to a fuzzy set; in contrast to (crisp) set theory, wherein an element explicitly either comes with a set or not. A specific type of a fuzzy set is a fuzzy number (Klir and Yuan, 1995), which is formally defined by {(x, $\mu_{\tilde{N}}$(x)|x ∈ □}, $\mu_{\tilde{N}}$: □ → [0; 1] where Ñ is referred to as fuzzy number. $\mu_{\tilde{N}}$ is denoted as the membership function of Ñ; it outputs the degree with which x ∈ □ belongs to Ñ. For example, the fuzzy number $\widetilde{10}$, which is to be equivalently seen as “real numbers close to ten” may be given by the membership function $\mu_{\widetilde{10}}(x) = (1 + (x - 10)^{-2})^{-1}$. The membership function differs from a probability density function in two regards: (1) $\int_{-\infty}^{\infty} \mu_{\tilde{N}}(x)dx$ does not need to equal 1, and (2) it mirrors the subjective attitude of an individual rather than reflecting statistical evidence. This is advantageous in cases where probabilities or exact data is not available, but subjective estimates of customers or experienced experts are given.

Fuzzy set theory provides a formal framework for arithmetic and logic operations on fuzzy numbers and crisp numbers (Buckley and Jowers, 2008). These sound theoretical concepts allow for flexibly extending our predefined models. In our extended model we utilize the aforementioned concepts of fuzzy set theory by fuzzifying resource requirements of the customers’ jobs or service requests. Besides, the mode of operation resembles the one of the basic model.

The objective function (O2) is the same as for the basic model. In the constraints the resource requirements are now a fuzzy number and denoted by $\tilde{c}_{mr}$. Consequently the current utilization $\tilde{U}_l(t)$ is now also a fuzzy number. Furthermore adequate operators for the addition, division, and less equal which can be used with fuzzy numbers have to be chosen. The chosen operators and their implementation are described in the description of the evaluation scenario and simulator implementation.

$$\max_x \sum_{j \in J} \left(\frac{1}{2} * x_j\right)^j \tag{O2}$$

subject to:

$$\sum_{m=1}^{|J|} \widetilde{c}_{mr}(t) * x_m \leq c_r(t) \quad \forall t \in T, \forall r \in R \qquad (\widetilde{C2.1})$$

$$\sum_{l=1}^{r} \sum_{k=1}^{n} \left(\left[H_1 \widetilde{U}_l(t) - l_{k-1} \right) - H_1 \left(\widetilde{U}_l(t) - l_k \right) \right]$$

$$* \widetilde{c}_{jl}(t) * rp_{k-1} * x_j \leq \frac{fp_j}{s_j} \forall t \in T, \quad \forall j \in J \qquad (\widetilde{C2.2})$$

$$(1 - cc_j) * \frac{\sum_{m=1}^{|J|} \widetilde{c}_{mr}(t) * x_m}{c_r(t)} \leq l_c \quad \forall t \in T, \forall r \in R, \forall j \in J \qquad (\widetilde{C2.3})$$

The combination of (O2) and $(\widetilde{C2.1})$ represents the FCFS-Policy, for the dynamic pricing constraints $(\widetilde{C2.2})$ are added. The strict priority policy is represented by of (O2), $(\widetilde{C2.1})$, and $(\widetilde{C2.3})$.

4 Simulation and Evaluation

To validate the models and estimate the effect of different degrees of uncertainty on the revenue a thorough evaluation based on real world workloads is done. First, the evaluation scenario, the simulation setting, the workloads used for the evaluation and the implementation of fuzzy numbers and operators are explained. Subsequently, the results are presented. The simulator was implemented as an object-oriented program in the numerical computing environment MATLAB.

4.1 Scenario Description

For the scenario based on real workloads, data from the Parallel Workload Archive (Feitelson, 2009) was used. The SHARCNET log, which was graciously provided to the Parallel Workload Archive by John Morton (john@sharcnet.ca) and Clayton Chrusch (chrusch@sharcnet.ca), was used as a basis for this scenario. It contains 1,195,242 jobs sent to a set of 10 clusters in Ontario, Canada from a period starting in December 2005 and ending in January. The SHARCNET log was chosen as a basis because it contains a large variety of jobs with different runtimes, numbers of used CPUs, and varying submit and start times. The workload further shows high variation in demand over time.

Jobs running less than one hour or more than 10 days were filtered. Subsequently runtimes were rounded down to full hours to allow a timeslot based allocation. After filtering invalid jobs 566,701 jobs were left. Based on these workloads nine joblists with different prices and service type assignment were generated as described in the following paragraphs. For the evaluation we considered the following three types of services with requirements for processing power, memory, and storage according to Table 1.

Service	CPU	Storage	Bandwidth
1	4	4	16
2	8	16	4
3	16	8	8

Table 1. Service types

Table 2 shows the contents of the job lists and the source of the data used. Data regarding user ID, submission time, start time, and run time was used as stored in the workload trace. The service type was drawn from a discrete uniform distribution. The number of instances was adapted from the job requirements present in the workload trace. The pricing information was generated using normal distribution.

Variable	Source	Adaption / Distribution
Submission Time	Workload trace	None
Start time	Workload trace	None
Runtime	Workload trace	None
Number of instances	Workload trace	Adapted from job requirements to for the three service scenario
User ID	Workload trace	None
Service Type	Generated	Uniform Distribution {1, 2, 3}
Price	Generated	Normal distribution N(1, 0.5) per resource unit and timeslot

Table 2. Workload information

As a further condition a minimum unit price of 0.1 was required. Full prices were calculated by multiplying unit prices with the number of instances and time slots and rounding the value up to the next integer. It was assumed that gold clients are willing to pay a markup of 20% for the priorities. In this scenario, the capacity is determined by CPU, Storage, and Bandwidth capacity and limited to 1050. This capacity was chosen to accommodate some of the larger jobs from the SHARCNET log but still have time periods where demand exceeds supply. Jobs can either start in the same timeslot in which they are submitted or in a later timeslot. The Extended Model makes use of the concept of Triangular Fuzzy Numbers. A triangular fuzzy number N=(a,b,c), a<b<c, {a,b,c} □ □, is a fuzzy set over R, with the membership function:

$$\mu^n(x)=\begin{cases}\frac{x-a}{b-a}, & if\ a \leq x < b\\ \frac{c-x}{c-b}, & if\ b \leq x \leq c\end{cases}$$

If l:=(c-b)=(b-a), then the triangular fuzzy number is symmetric. In our implementation, we use symmetric fuzzy numbers with l=0.1*b. Thereby, we account for the situation in which the maximally expected deviation of the required level of resources from the predicted level does not exceed 10%. The following fuzzy operators and the respective implementations were used in our simulator: Addition of fuzzy numbers: Given that x = (a, b, c) and y = (d, e, f) are two triangular fuzzy numbers, we add them by calculating x + y = (a + d; b + e; c + f) i.e. component wise. Division of a fuzzy number and a scalar: If g is a scalar we divide x over g, by creating a new fuzzy number z = (g, g, g) and then calculating x/g= (a/g, b/g, c/g). A comparison operator of a fuzzy number and a scalar: A fuzzy number is less or equal than a scalar if its right vertex is less or equal than it, for example x < g is true if and only if c < g is true. Analogously, x ≥ g if and only if a ≥ g i.e. the left vertex of the fuzzy number is greater or equal than the scalar.

4.2 Simulation Results

This section summarizes the results achieved in the scenario under certainty (scenario 1) and in the scenario under uncertainty. The runtime of one instance of the simulation with 566,701 jobs was roughly between 10 and 30 seconds for the different policies in scenario one. With runtimes between 1,500 and 4,000 seconds the runtimes for the scenario under uncertainty (scenario 2) was significantly higher, however each job decision took only fractions of a second, so that all models are applicable in practice for making real-time decisions. Due to the capacity constraint no overload situation due to acceptance of too many jobs occurred, assuring QoS.

The basic first-come first-served policy which is used as a benchmark results in a mean revenue over all simulations of 21,249,352.78. Table 3 shows a comparison of the results for scenario 1. It contains the policies, mean revenue μ, the coefficient of variation of revenue (σ/μ), the revenue increase compared to the benchmark policy I, and the mean ratio of accepted gold jobs μΓg as well as the coefficient of variation for the gold acceptance ratio (σ/μ)Γg.

Policy	μ rev.	σ/μ rev.	rev. increase	$\mu\Gamma_g$	$(\sigma/\mu)\Gamma_g$
I	21,249,352.78	0.0187	0.00%	4.22%	0.0299
II	24,722,267.67	0.0169	16.34%	4.33%	0.0462
III	17,297,469.78	0.0074	-18.59%	11.93%	0.0110

Table 3. Revenue and gold acceptance rate for scenario under certainty (scenario 1)

Policy II, using dynamic pricing, beats the FCFS policy by 16.34% concerning revenue. Policy III, featuring client classification has 18.59% lower revenue. It might still be preferable for providers, e.g. due to contractual obligations, or other revenue generated by gold clients, which is not considered here. Furthermore, it outperforms both policy I and II concerning the ratio of accepted gold jobs.

Table 4 shows similar results for the scenario under uncertainty. Thus, the presence of certainty or uncertainty has no substantial impact on the quality of the applied policies.

Policy	μ rev.	σ/μ rev.	rev. increase	$\mu\Gamma_g$	$(\sigma/\mu)\Gamma_g$
I	19,956,299.11	0.0117	0.00%	4.14%	0.0397
II	24,431,015.33	0.0086	22.42%	4.36%	0.0432
III	15,858,202.11	0.0086	-20,54%	11.02%	0.0069

Table 4. Revenue and gold acceptance rate for scenario under uncertainty (scenario 2)

We now investigate the intra-policy impact of having uncertain information (costs of uncertainty). Table 5 shows that, for all policies, the presence of uncertainty causes losses in revenue. The loss even exceeds 8% when policy III is applied. The costs of uncertainty mirror the disadvantage for the cloud provider that figures on required resources are uncertain and that therefore the provider loses leeway in using his resources. From an economic perspective, costs of uncertainty provide an upper bound for financial incentives that the provider should give to his customers for increasing their preciseness of resource estimates. Regarding the gold acceptance rate, uncertainty does not have a significant impact.

Policy	Scenario	μ rev.	σ/μ rev.	Costs of uncertainty	μ r_g	σ/μ r_g
I	1	21,249,352.78	0.0187	---	4.22%	0.0299
	2	19,956,299.11	0.0117	-6.09%	4.14%	0.0397
II	1	24,722,267.67	0.0169	---	4.33%	0.0462
	2	24,431,015.33	0.0086	-1.18%	4.36%	0.0432
III	1	17,297,469.78	0.0074	---	11.93%	0.0110
	2	15,858,202.11	0.0086	-8.32%	11.02%	0.0069

Table 5. Impact of uncertainty on revenue and gold acceptance rate

5 Conclusion and Outlook

In this work we show the need for a cloud service admission control decision model both under certainty and uncertainty, and we provide a decision model for cloud providers that allows determining the optimal job acceptance behaviour in terms of revenue. Due to the complexity of this approach and the unavailability of information on resource requirements, we suggest using policies as heuristics in both cases where information on required resources is certain and uncertain. Uncertainty is modelled based on fuzzy set theory and fuzzy optimization, which allows accounting for informational uncertainty in the absence of statistical data.

We evaluate the models using simulations that are based on real world workloads. Our simulations are implemented in the numerical computing environment MATLAB. Our results show that in both cases under certainty and under uncertainty applying the dynamic pricing policy significantly increases revenue while using the client classification policy substantially reduces revenue. However, the results also show that the latter policy, which gives gold clients a priority on job acceptance, can substantially increase the acceptance ratio for gold customers.

We further analyse the impact of having uncertain information on each the three policies (costs of uncertainty). For all policies, the presence of uncertainty causes losses in revenue, in client classification policy even more than 8%. These costs of uncertainty indicate an upper bound for the financial incentives that cloud providers should provide to the customers for enhancing the quality of their resource predictions.

Future work includes further research on the relation between the degree of uncertainty and revenue. We also plan to research which effects can be observed when both the demand side, i.e. job requirements, and the supply side (available capacity) show uncertainty.

References

Aiber, S., Gilat, D., Landau, A., Razinkov, N., Sela, A. and Wasserkrug, S. (2004). Autonomic self-optimization according to business objectives, In Proceedings of the First International Conference on Autonomic Computing, Washington, DC, USA, 206–213.

Anandasivam, A., and Neumann, D. (2009). Reputation, Pricing and the E-Science Grid, Autonomic Systems. Springer, Germany, 25-43.

Bartlett, M., Frisch, A.M., Hamadi, Y., Miguel, I., Tarim, S.A., and Unsworth, C. (2005). The Temporal Knapsack Problem and Its Solution, In Integration of AI and OR Techniques in

Constraint Programming for Combinatorial Optimization Problems (R. Barták and M. Milano Eds.), Lecture Notes in Computer Science, vol. 3524. Berlin, Springer-Verlag, 34-48.

Boughton, H., Martin, P., Powley, W., and Horman, R. (2006). Workload class importance policy in autonomic database management systems, In Proceedings of the Seventh IEEE International Workshop on Policies for Distributed Systems and Networks,Washington, DC, USA, 13–22.

Buckley, J.J., and Jowers, L.J. (2008). Monte Carlo Methods in Fuzzy Optimization, Springer, Berlin, Heidelberg.

Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., and Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility,"Future Gener. Comput. Syst., 25 (6), 599-616.

Djemame, K., Gourlay, I., Padgett, J., Birkenheuer, G., Hovestadt, M., Kao, O., and Voß, K. (2006). Introducing risk management into the grid, In Proceedings of the 2nd IEEE International Conference on e-Science and Grid Computing, Amsterdam, Netherlands, 28.

Dube, P., Hayel, Y., and Wynter, L. (2005). 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), 99-102.

Feitelson, D. G. (2009). Workload Modeling for Computer Systems Evaluation.

Ferguson, D. F., Nikolaou, C., Sairamesh, J., and Yemini, Y. (1996). Economic models for allocating resources in computer systems, 156–183.

Foster, I., Kesselman, C., Lee, C., Lindell, B., Nahrstedt, K., and Roy, A. (1999). A distributed resource management architecture that supports advance reservations and co-allocation, In Proceedings of the 7th International Workshop on Quality of Service, London, UK, 62–80.

Ibarra, O.H., and Kim, C.E. (1975). Fast Approximation Algorithms for the Knapsack and Sum of Subset Problems, Journal of the ACM, 22 (4), 463-468.

Kellerer, H., Pferschy U., Pisinger D. (2004). Knapsack Problems, Springer, Berlin.

Klir, G.J., and Yuan, B. (1995). Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice Hall, Upper Saddle River, N.J., USA.

Kounev, S., Nou, R., and Torres, J. (2007). Autonomic qos-aware resource management in grid computing using online performance models, In Proceeding of the 2nd International Conference on Performance Evaluation Methodologies and Tools, Nantes, France.

Nair, S., and Bapna, R. (2001). An application of yield management for Internet Service Providers, Naval Research Logistics, 48 (5), 348–362.

Newhouse, S., MacLaren, J., and Keahey, K. (2004). Trading grid services within the uk e-science grid, In Grid resource management: state of the art and future trends, 479–490.

Nou, R., Julià, F., and Torres, J. (2007). Should the grid middleware look to selfmanaging capabilities?, In Proceedings of the 8th International Symposium on Autonomous Decentralized Systems, Sedona, Arizona, USA, 113–122.

Poggi, N., Moreno, T., Berral, J.L., Gavaldà, R., and Torres, J. (2009). Self-adaptive utility-based web session management, Computer Networks, 53 (10), 1712-1721.

Qiu, J. and Knightly, E. W. (2001). Measurement-based admission control with aggregate traffic envelopes. IEEE/ACM Trans. Netw., 9 (2), 199-210.

Ramalho, M. (1998). Uncertainty measures associated with fuzzy rules for connection admission control in ATM networks, In Applications of Uncertainty Formalisms, (1455/1998), (Hunter, A. and Parsons, S., Eds.), Springer, Berlin.

Yeo, C. S., and Buyya, R. (2004). Pricing for Utility-driven Resource Management and Allocation in Clusters, In Proceedings of the 12th International Conference on Advanced Computing and Communications, Ahmedabad, India, 32–41.

Zadeh, L. (1965). Fuzzy sets, Information Control, 1965 (8), 338-353.

Zimmermann, H.-J. (2000). An application-oriented view of modeling uncertainty, In European Journal of Operational Research, 122 (2), 190-198.