Maximum revenue-oriented resource allocation in cloud

Guofu Feng*

Department of Computer Science and Technology,
Nanjing Audit University,
77 Beiwei Road, Nanjing 210029, China
Email: njufgf@gmail.com
*Corresponding author

Rajkumar Buyya

Cloud Computing and Distributed Systems (CLOUDS) Laboratory,
Department of Computing and Information Systems,
The University of Melbourne,
VIC 3010, Australia
Email: rbuyya@ unimelb.edu.au

Abstract: Cloud computing is distinguished from such conventional computing paradigms as grid computing and cluster computing in that it provides a practical business model for customers to use the resources remotely. It is natural for service providers to allocate the pooled cloud resources dynamically among the differentiated customers to maximise their revenue. This paper addresses the problem of the revenue maximisation through the SLA-aware resource allocation. Firstly, two TSF (Time Service Factor) based pricing models are proposed since TSF is a widely used metric to determine the billings of internet services with variable performance. Then the resource allocation problem is formulised with queuing theory and its optimal solutions are proposed. The optimal solution considers various Quality of Service (QoS) parameters such as pricing, arrival rates, service rates and available resources. Finally, the experiment results, both with the synthetic dataset and traced dataset, are presented. They have validated our optimal resource allocation solutions and shown that our algorithms outperform the related work.

Keywords: cloud computing; service level agreement; resource allocation; time service factor; pricing mechanism; pooled resources.

Reference to this paper should be made as follows: Feng, G. and Buyya, R. (201x) 'Maximum revenue-oriented resource allocation in cloud', *Int. J. Grid and Utility Computing*, Vol. x, No. y, pp.xx–xx.

Biographical notes: Guofu Feng is an Associate Professor in Nanjing Audit University. He obtained his PhD in Computer Science and Technology from Nanjing University, Nanjing, China in 2006. His research interests are distributed computing systems and cloud computing.

Rajkumar Buyya is Professor of Computer Science and Software Engineering, Future Fellow of the Australian Research Council, and Director of the Cloud Computing and Distributed Systems (CLOUDS) Laboratory at the University of Melbourne, Australia. He is also serving as the founding CEO of Manjrasoft, a spin-off company of the University of Melbourne, commercialising its innovations in cloud computing. He has authored over 450 publications and four text books including *Mastering Cloud Computing* published by McGraw Hill and Elsevier/Morgan Kaufmann, 2013 for Indian and international markets respectively. His research interests include distributed computing, utility computing and cloud computing.

This paper is a revised and expanded version of a paper entitled [title] presented at [name, location and date of conference].[AQ1]

AQ1: If a previous version of your paper has originally been presented at a conference, please complete the statement to this effect or delete if not applicable.

1 Introduction

Cloud computing is a technology that uses remote centralised servers via internet, whereby all applications are provided as a service. By means of cloud computing, customers can expand or shrink resources for their applications adaptively and simultaneously save themselves from the complex IT management and maintenance. Cloud computing makes it possible for any company or consumer to operate sophisticated and expensive applications without having

to purchase, install and maintain the actual software and hardware.

From the perspective of service providers, cloud is a pool computing resources providing services for end-users. The goal of cloud providers is to lease various services and obtain the revenue. The business model is the key point to distinguish cloud computing from conventional distributed computing paradigms (Gong et al., 2010). Cloud services are delivered to customers in a 'prepaid' or 'pay-as-you-go' manner. The customers only rent the required resources or services and only pay for the consumed services.

This paper considers two scenarios: (1) some big multinational corporations rent the IT infrastructure and services from cloud as a virtual data centre for their branches all over the world; (2) a cloud provides electronic transaction services for some small electronic commerce companies.

Under both scenarios, Service Level Agreement (SLA) plays an indispensable role in facilitating the transactions between customers and service providers. Consumers indicate the required service level through Quality of Service (QoS) parameters, which are noted in SLAs established with providers. SLA usually specifies a common understanding about responsibilities, guarantees, warranties, performance levels in terms of availability, response time, etc. A pricing model, including the pricing mechanism and penalties in case of non-compliance of SLA, is also specifically defined in SLA. The cloud services are usually charged according to achieved performance level. For example, Amazon EC2 offers three types of compute services i.e., on-demand, spot and reserved, at different prices based on their service/performance levels.

We define a *service instance* as a combination of a customer and a certain type of rented service with binding SLA. The service instances certainly have different attributes such as arrival rate, execution time, performance requirement and pricing mechanism. Even for the same service instance, its arrival rate can vary with different distributions at different times.

To a service provider, the proper resource allocation among the service instances is of vital importance because the revenue can vary quite widely under different allocation strategies even with the same resources. Therefore, a fundamental problem faced by any cloud service provider is how to maximise their revenue by managing resource allocation and providing differentiated performance levels dynamically based on SLAs and measurable service attributes.

This paper focuses on how a Cloud data center maximises the SLA-based revenue by proper resource allocation and two optimal allocation algorithms are presented. The basic idea in this paper is to re-allocate the resources among different service instances adaptively based on the dynamically collected information. Our main contributions in this paper include:

1 We have proposed a Queuing Theory based mathematical model to formulate the resource allocation problem. The formulation models the application situations with various parameters such as resource quantity, request arrival, service time and pricing model.

2 We have proposed two optimal pricing mechanism based resource allocation algorithms, by which cloud providers can maximise their revenue given a pricing model. The algorithms outperform related work under any situation as they are obtained from the theoretical analysis but not experience or inspiration.

The remainder of this paper is organised as follows. Section 2 presents some related work. Section 3 introduces two pricing models in terms of TSF (Time Service Factor). Section 4 formulates the problem of resource allocation and provides the exact answers to these two problems. In Section 5, we have carried out some simulations to verify our solutions. Finally, Section 6 concludes our paper.

2 Related work

Cloud computing provides the practical business models to facilitate the trades between providers and customers, which distinguishes cloud computing from previous typical computing paradigms (Gong et al., 2010). Therefore, SLA plays an important role in that it provides mechanisms and tools for customers to express their requirements and constrains such as response time and price scheme. It is very natural but challenging for service providers to transform the service-oriented contract metrics into resource-oriented metrics and allocate the resources dynamically among the customers, thereby maximising the revenue. Buyya et al. (2009) argues that commercial offerings with clouds must be able to derive proper market-based resource management strategies and leverage VM (virtual machine) technology to assign resources dynamically.

There is an extensive literature on resource management techniques for commercial data centres. Utility is often used to evaluate resource allocation, especially when the objectives and criterion are multi-dimensional. Amato et al. (2013) proposed a touristic context-aware recommendation system by means of cloud computing infrastructure that allows to process big collections of data and numerous user accesses. Walsh et al. (2004) discussed a distributed architecture for dedicated data centre with dynamic virtual pool. However, it emphasised on the utility of resource usage rather than the utility of data centre. Buyya et al. (1997) proposed a QoS-aware resource allocation model Q-RAM to maximise the utility under multi-dimensional QoS constraints. Q-RAM was further enhanced with scalability and ability (Ghosh et al., 2003; Hansen et al., 2004). Householder et al. (2014a, 2014b) have proposed the oversubscription technology in cloud infrastructure to diminish the sum of unutilised resources.

Many works illustrated how to meet the QoS and SLA requirements by proper resource allocation. Daniel et al. (2001) and Mohammed and Daniel (2005) proposed an approach based on hill climbing techniques to cope with short-term fluctuations in the workload and guide the search

for the best combination of configuration parameters of a multilayered architecture. Chandra et al. (2003) presented techniques for dynamic resource allocation in shared web servers. Levy et al. (2003) presented a prototype implementation with performance management, where cluster utility was used to encapsulate business value. The system dynamically allocates server resources, balances the load among multiple classes according to performance demand. Li et al. (2005) took the minimisation of resource consumption as the objective and proposed a strategy for autonomic computing to meet requirements in terms of response time and server utilisation. Kertesz et al. (2014) presented a self-manageable architecture for SLA-based service virtualisation that provides a way to ease interoperable service executions in a distributed and virtualised world of services. Garg et al. (2011) proposed a technique to maximise the utilisation of Cloud data centre with different SLA requirements, where utilisation is indicated by the number of used hosts for a given workload.

Moreover, many works considered the economic issues related to SLAs. Goudarzi et al. (2012) considered a resource allocation problem, which aimed to minimise the total energy cost of cloud system while meeting SLAs. Zhang and Ardagna (2004) proposed a resource allocation controller to maximise the provider's revenues associated with multi-class SLA, where the revenue depended on discrete QoS levels. Liu et al. (2001) proposed a theoretical model to maximise the revenue of a hosting platform subject to multi-class SLAs. Püschel et al. (2010) presented a framework that linked technical and economical aspects to the management of computational resources. It combined some technical methods such as dynamic pricing, different job priorities, and client classification into an economically enhanced resource management, which increases the revenue for the local resource sites. Villela et al. (2007) studied how a service provider should allocate the application tier of an e-commerce application subject to QoS constraints. Our work is different from them in that we adopt a continuous price function and provide the formal precise answer to the problems. Decentralised economic approaches are proposed in (Goiri et al., 2012; Bonvin et al., 2011) to utilise the resources dynamically to meet SLA performance and availability goals in a federated cloud.

There were several works sharing the similar scenarios with our work. Zhu et al. (2001) proposed an allocation strategy of server resources among customers to minimise the mean response time. However, this work did not consider the economic model, thereby the parameter of weight q in optimal solution lacks of the specific practical meaning. Goudarzi and Pedram (2011) and Wu et al. (2011) also addressed the problem of SLA-based resource allocation. However, they emphasised the situation where services have multi-dimensional resources such as memory, bandwidth and CPU. This paper masks the resource diversity. The work of Mazzucco (2009) is similar to ours. It provided two strategies for the resource allocation, Heuristic and Greedy. Greedy is optimal but it often costs an impractically long execution time while the improved algorithm does not always work well. Heuristic is simple but our following work have displayed that its validity is affected much by the environment parameters.

3 Models

Cloud computing should incorporate autonomic resource management according to the signed SLA that effectively satisfies service demands and obligations. In this paper, we contribute towards this aim of cloud computing and therefore, consider a similar scenario where a cloud provider offers various services to customers at different SLA levels. All services are hosted within a data centre using certain number of virtual infrastructure, which can grow and shrink. The objective is to find a proper allocation of servers among the service instances so that maximum revenue can be achieved given a charging/pricing model.

3.1 Mathematical model

We assume there are all N servers in the cloud data centre. Service provider has signed long-term SLAs with m customers. Each customer is allocated with servers to provide services. We consider n_i servers allocated to each service instance i as one super server. The capability of each super server is proportional to the server number. This assumption is reasonable especially for those computing tasks that can be divided into several pieces and dispatched to many servers to execute concurrently. For example, many dynamic web pages are composed of many parts that should be computed separately; or some tasks can be decomposed for parallel computing.

Table 1 Symbols and their meanings

Symbol	*Meaning*
λ	Arrival rate
ρ	Service intensity
N	Number of all the servers
R	Demand of response time
r	Variable of response time
b	Base price of services
μ	Service rate per server
m	Number of service instances (customers)
c	Margin per unit resource
A	Demand of assurance factor
a	Variable of assurance factor
B	Revenue from a service provision

We assume that the requests from each service instance arrive at the system in a Poisson distribution with average arrival rate λ and the processing time by one server follow a

negative exponential distribution with average service rate $1/\mu$ (μ is the number of processed requests per unit time). We also assume that it costs much for servers to shift their running environments. For example, it needs a long time to read the commercial data of a new customer into cache from the external memory. Therefore, each group of servers associated with one service instance can be modelled as a FIFO (First In First Out) *M/M/1* queue. Here we define service intensity ρ as the ratio of arrival rate to service throughput of one server,

$$\rho = \lambda / \mu \tag{1}$$

The symbols used in this paper are listed in Table 1.

3.2 TSF (Time Service Factor) based pricing models

3.2.1 Service performance metrics

The cloud services are usually charged based on achieved performance. There are two often used metrics associated with response time, *MRT* (*Mean Response Time*) and *TSF* (*Time Service Factor*). MRT is a commonly used metric to express the service performance. However, it cannot reflect the reality when the response time varies over a large range. TSF is another metric to evaluate the service performance. It means the percentage of services answered within a definite timeframe, e.g., 80% in 20 seconds (Wikipedia, 2014). Time service factor is widely used because it can reflect the response time as well as response time distribution more precisely. We formulate TSF as a 2-tuple <*r, a*>, where *r* is response time and *a* means the percentage of services with the response time less than *r*. Symbol *a* also is referred as to *Assurance Factor*.

In this paper, we propose two user's demand based metrics, *ASF* (*Assurance Satisfaction Factor*) and *RSF* (*Response Satisfaction Factor*). Both ASF and RSF reflect the deviation of achieved performance to user's demand. We denote user's demand in terms of TSF by <*R, A*>. And then we define ASF of a service instance with a demand <*R, A*> in SLA as,

$$f_A = \frac{a - A}{1 - A} \tag{2}$$

where *a* is the actually achieved assurance factor with response time demand *R*, namely the ratio of services with response time less than *R*. f_A means the offset of actually achieved performance to user's demand. Expression (2) implies that the achieved performance meets user's demand when f_A is larger than or equal to zero. f_A is less than zero when the achieved performance fails to meet user's demand. Expression (2) also shows us that the same offset is more sensitive for those services having severe response time requirement.

We denote user's demand in terms of TSF by <*R, A*>, and define RSF of a service instance with demand <*R, A*> as,

$$f_R = \frac{R - r}{R} \tag{3}$$

where *r* is the achieved *A*-th response time. Expression (3) implies that the achieved performance meets user's demand when f_R is larger than or equal to zero. f_R is less than zero when the achieved performance fails to meet user's demand.

The percentile mechanism (e.g. the well-known 95th percentile) is a widely used mathematical calculation to determine billings for internet services that are provided as 'burstable' (variable rate) performance. To calculate *A*-th percentile response time of a service instance during a time slot, we

- record all the response time of all the services during a time slot of a service instance;
- sort all the records in an ascending order;
- select the *A*-th record from the ordered sequence.

3.2.2 Pricing models

In this paper, we propose two demand-driven pricing models in terms of ASF and RSF respectively. We partition the provision time of a service instance into slots with fixed length. If <*R*, *A*> denotes the proposed demand by customer, we can obtain the service performance in terms of ASF and RSF. If the achieved performance meets users' demands, the services are charged with the base price. If the actual performance fails to meet the demand, the service provider will be penalised at the basis of base price. Base price *b* is determined by the attributes of service instances. The algorithm to obtain *b* is presented in Subsection 3.2.3.

Here we define two pricing models in terms of ASF and RSF,

$$B = \begin{cases} b, f_A \geq 0 \\ b + bf_A, f_A < 0 \end{cases} \tag{4}$$

Or,

$$B = \begin{cases} b, f_R \geq 0 \\ b + bf_R, f_R < 0 \end{cases} \tag{5}$$

where price *B* of a service provision is a linear function of f_A and f_R respectively.

Both pricing models are also illustrated using Figure 1. We denote these two pricing models by *ASF* and *RSF* in the following.

3.2.3 Base price

According to the conclusions on Queuing Theory of *M/M/1* model, the cumulative distribution function of sojourn time is (Thomas, 2000),

$$w(t) = 1 - e^{(\lambda - \mu)t} \tag{6}$$

We assume that a service instance with demand <*R, A*>is assigned *n* servers. To substitute <*R*, *A*> into Expression (6),

$$A = 1 - e^{(\lambda - n\mu)R} \tag{7}$$

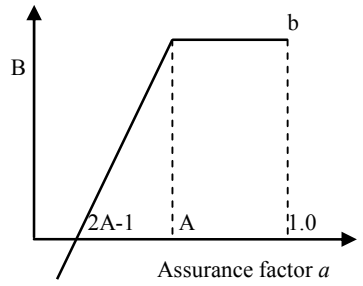
$$n = \rho - \frac{\ln(1 - A)}{\mu R} \tag{8}$$

Expression (8) implies that n servers are required to guarantee the customer's performance demand $<R, A>$. If we assume c is the expected margin revenue per unit resources, then b is the expected revenue from n servers,

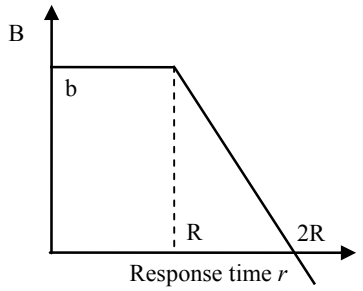
$$b = cn = c\left(\rho - \frac{\ln(1-A)}{\mu R}\right) \tag{9}$$

Arrival rate may vary dynamically. We can adopt the statistical mean arrival rate by sampling.

Figure 1 Pricing models in terms of time service factor



(a) Pricing model ASF



(b) Pricing model RSF

4 Optimal resources allocation in theory

4.1 ASF based optimal allocation

We assume that the service instance i is assigned n_i servers and its performance demand is $<A_i, R_i>$. The assurance factor a_i actually is the probability that the response time of a service request is less than or equal to R_i. According to (6), the assurance factor a_i will be,

$$a_i = 1 - e^{(\lambda_i - n_i\mu_i)R_i} \tag{10}$$

By substituting (10) into (2), the performance of service instance i in terms of metric ASF is,

$$f_A^i = \frac{1 - e^{(\lambda_i - n_i\mu_i)R_i} - A_i}{1 - A_i} \tag{11}$$

According to the price model (4), a service provision of service instance i brings the mean revenue is,

$$g_i = b_i\left(1 + f_A^i\right) = b_i\left(1 + \frac{1 - e^{(\lambda_i - n_i\mu_i)R_i} - A_i}{1 - A_i}\right) \tag{12}$$

Then the overall revenue from service instance i per unit time is,

$$G_i = \lambda_i b_i\left(1 + \frac{1 - e^{(\lambda_i - n_i\mu_i)R_i} - A_i}{1 - A_i}\right) \tag{13}$$

Thus, our optimisation problem can be formulated as,

$$\begin{aligned} &Max \sum_{i=1}^{m} \lambda_i b_i\left(1 + \frac{1 - e^{(\lambda_i - n_i\mu_i)R_i} - A_i}{1 - A_i}\right) \\ &s.t.\ \sum_{i=1}^{m} n_i = N \end{aligned} \tag{14}$$

Constructing Lagrange composite function,

$$\begin{aligned} L(n_i) = &\sum_{i=1}^{m} \lambda_i b_i\left(1 + \frac{1 - e^{(\lambda_i - n_i\mu_i)R_i} - A_i}{1 - A_i}\right) \\ &+ \overline{\lambda}\left(N - \sum_{i=1}^{m} n_i\right) \end{aligned} \tag{15}$$

where $\overline{\lambda}$ is Lagrange multiplier.

Letting $dL / dn_i = 0$, $i = 0,1,2...m$,

$$\frac{\lambda_i b_i \mu_i R_i e^{(\lambda_i - n_i\mu_i)R_i}}{1 - A_i} - \overline{\lambda} = 0 \tag{16}$$

$$n_i = \rho_i - \ln\left(\frac{\overline{\lambda}(1 - A_i)}{\lambda_i b_i \mu_i R_i}\right) / \mu_i R_i \tag{17}$$

Substituting (17) into the constrain condition in (14),

$$N = \sum_{j=1}^{m} \rho_j + \sum_{j=1}^{m} \ln\left(\frac{\lambda_j b_j \mu_j R_j}{1 - A_j}\right) / \mu_j R_j - \ln\overline{\lambda} \sum_{j=1}^{m} \frac{1}{\mu_j R_j} \tag{18}$$

$$\ln\overline{\lambda} = \frac{\sum_{j=1}^{m} \ln\left(\frac{\lambda_j b_j \mu_j R_j}{1 - A_j}\right) / \mu_j R_j + \sum_{j=1}^{m} \rho_j - N}{\sum_{j=1}^{m} \frac{1}{\mu_j R_j}} \tag{19}$$

Substituting (19) into (17), we can obtain the results,

$$\begin{aligned} n_i = &\rho_i + \ln\left(\frac{\lambda_i b_i \mu_i R_i}{(1 - A_i)}\right) / \mu_i R_i \\ &- \frac{\sum_{j=1}^{m} \ln\left(\frac{\lambda_j b_j \mu_j R_j}{1 - A_j}\right) / \mu_j R_j + \sum_{j=1}^{m} \rho_j - N}{\mu_i R_i \sum_{j=1}^{m} \frac{1}{\mu_j R_j}} \end{aligned} \tag{20}$$

It is assumed that arrival rate of requests of each service instance can be modelled by Expression (6). However, (6) is valid only when the arrival rate of each service instance is less than service processing rate. Otherwise, the response time of a queue with FIFO does not converge and the response time always increases as time elapses. Therefore, our conclusion of (20) holds only if arrival rate is less than service processing rate,

$$\lambda_i < n_i \mu_i \tag{21}$$

$$n_i > \rho_i \tag{22}$$

Figure 1(a) also shows us that the revenue stops to rise any more once a_i equals to A_i. Thus, there is no use any more to increase the resources for service instance i once the response demand is met. According to (8),

$$n_i \le \rho_i - \frac{\ln(1-A_i)}{\mu_i R_i} \tag{23}$$

Therefore, Expression (22) is the lower resource bound and (23) is the upper threshold for instance i.

4.2 Optimal allocation based on RSF

We assume that the service instance i is assigned n_i servers and its performance demand is $<A_i, R_i>$. We also assume that the A_i-th response time of service instance i is r_i.

Then according to (6), Expression (24) holds,

$$A_i = 1 - e^{(\lambda_i - n_i\mu_i)r_i} \tag{24}$$

Then the A_i-th restime of service instance i is,

$$r_i = \frac{\ln(1-A_i)}{\lambda_i - n_i\mu_i} \tag{25}$$

By substituting (25) into (5), the performance of service instance i will be,

$$f_R^i = 1 - \frac{\ln(1-A_i)}{R_i(\lambda_i - n_i\mu_i)} \tag{26}$$

According to the pricing model (5), a service provision of service instance i brings the mean revenue is,

$$g_i = b_i\left(1 + f_R^i\right) = b_i\left(2 - \frac{\ln(1-A_i)}{R_i(\lambda_i - n_i\mu_i)}\right) \tag{27}$$

Then the overall revenue from service instance i per unit time is,

$$G_i = \lambda_i b_i\left(2 - \frac{\ln(1-A_i)}{R_i(\lambda_i - n_i\mu_i)}\right) \tag{28}$$

Thus, our optimisation problem can be formulated as,

$$\begin{aligned} Max & \sum_{i=1}^{m} \lambda_i b_i\left(2 - \frac{\ln(1-A_i)}{R_i(\lambda_i - n_i\mu_i)}\right) \\ s.t. & \sum_{i=1}^{m} n_i = N \end{aligned} \tag{29}$$

Constructing Lagrange composite function,

$$\begin{aligned} L(n_i) = & \sum_{i=1}^{m} \lambda_i b_i\left(2 - \frac{\ln(1-A_i)}{R_i(\lambda_i - n_i\mu_i)}\right) \\ & + \bar{\lambda}\left(\sum_{i=1}^{m} n_i - N\right) \end{aligned} \tag{30}$$

where $\bar{\lambda}$ also is Lagrange multiplier.

Letting $dL/dn_i = 0$, $i = 0,1,2...m$,

$$\frac{\lambda_i b_i \mu_i \ln(1-A_i)}{R_i(\lambda_i - n_i\mu_i)^2} - \bar{\lambda} = 0 \tag{31}$$

$$n_i = \rho_i - \sqrt{\frac{-1}{\bar{\lambda}}}\sqrt{\frac{-\lambda_i b_i \ln(1-A_i)}{\mu_i R_i}} \tag{32}$$

Substituting (32) into the constrain condition in (29),

$$N = \sum_{j=1}^{m}\rho_j - \sqrt{\frac{-1}{\bar{\lambda}}}\sum_{j=1}^{m}\sqrt{\frac{-\lambda_j b_j \ln(1-A_j)}{\mu_j R_j}} \tag{33}$$

$$\sqrt{\frac{-1}{\bar{\lambda}}} = \frac{\sum_{j=1}^{m}\rho_j - N}{\sum_{j=1}^{m}\sqrt{\frac{-\lambda_j b_j \ln(1-A_j)}{\mu_j R_j}}} \tag{34}$$

Substituting (34) into (32), we can obtain the results,

$$n_i = \rho_i - \frac{\sum_{j=1}^{m}\rho_j - N}{\sum_{j=1}^{m}\sqrt{\frac{-\lambda_j b_j \ln(1-A_j)}{\mu_j R_j}}}\sqrt{\frac{-\lambda_i b_i \ln(1-A_i)}{\mu_i R_i}} \tag{35}$$

Owing to the same reason as previous subsection, Expression (22) also is the lower resource bound and (23) is the upper threshold for service instance i.

5 Performance evaluations

In this section, we present our experimental results on the efficiency of our algorithms for optimising the resource provisioning technique in the cloud environment. We provide two types experiments in the following, whose requests come from synthetic dataset and traced dataset respectively.

The experiments presented in this section are obtained from our simulator, which is developed with C language. The simulator is implemented with a time-driven model. Simulation clock increases at a constant rate of one millisecond. After each millisecond, we check and handle those events happen at the current time point. The events mainly include four types: request arrival, request departure, resource reallocation, and output the experiment results.

We use revenue from services as our main metric to evaluate the strategies. In the evaluation we use the strategy of Heuristic, a resource allocation algorithm proposed by Michele in his thesis (Mazzucco, 2009), as our target to compare with, because this work is the most similar to ours among all the related works.

In the following, we use ASF, RSF, and Heuristic to denote our optimal allocation in terms of ASF, our optimal allocation in terms of RSF, and the heuristic allocation algorithm proposed by Michele respectively. The related parameters and their default values are listed in Table 2.

Table 2 Parameters and their default values in experiments

Parameter	*Default values*
Arrival Distribution of Requests	Poisson
Service Time Distribution	Negative Exponential
Arrival Rate λ	Random (10…50)/minute
Service Rate μ	Random (5…15)/minute
Assurance factor demand	Random (0.7…0.98)
Response demand	Random (3…10)second
Number of Types m	20
Margin c	1$

5.1 Simulations with synthetic data

Figures 2 and 3 show the comparison of revenue with different pricing mechanisms and server resources between ASF, RSF, and Heuristic. The time slot is set an hour in the simulations. We calculate and output the revenue outcomes when the time slot expires.

Figure 2 Revenue under ASF pricing model versus server number (see online version for colours)

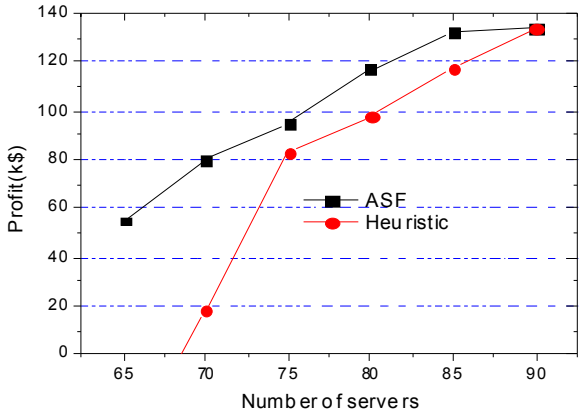
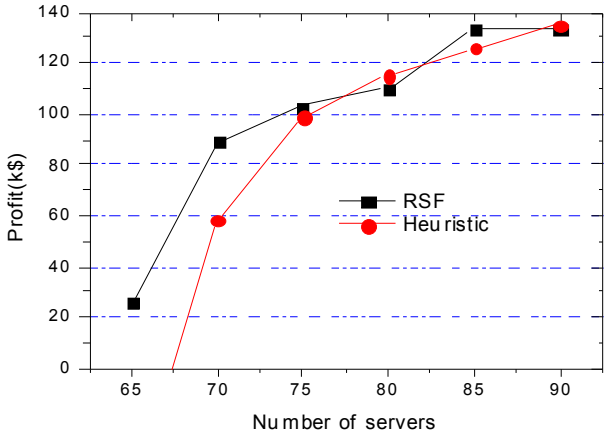


Figure 3 Revenue under RSF pricing model versus server number (see online version for colours)



Figures 2 and 3 show us that both resource allocation strategies ASF and RSF outperform Heuristic. This is because that ASF and RSF are the results of our theoretical analysis. The simulations partially support that our conclusions are correct. The revenue of Heuristic is 75.5% and 16.2% lower than ASF when the number of servers is 70 and 80. The revenue of RSF is 53.4% higher than Heuristic when the number of servers is 70. The curves of Heuristic are close to ASF and RSF with the increasing of servers. The values are not notable any more when the number of servers gets to 90. This is because that most service instances are assigned the same resources with the upper threshold when the cloud data centre has sufficient server resources for all the service instances.

Figures 2 and 3 show us that the superiority of ASF and RSF is more remarkable when the resource is relatively rare. Therefore, ASF and RSF are valuable to improve the revenue through proper allocation especially when the resource is rare or the service instances are numerous.

5.2 Simulations with traced data

We use the traced data to simulate the requests and allocate the resources adaptively according to probed parameters. The traced data come from (Internet Traffic, 2011). All the data are records of HTTP requests to WWW servers. We intercept consecutive request records of 8 hours from the traces to simulate the arrival of service instances. The detailed information is shown in Table 3 and Figure 4.

We partition the time into slots, each with a length of 5 minutes. During the execution, we count the number of arrived requests at each time slot. Then we predict the average arrival rate of next slot according to the records of previous and current slot. The predicting algorithm is formulated as,

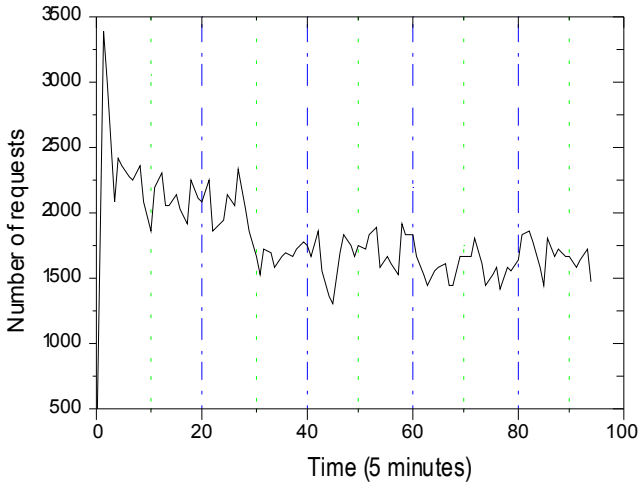
$$\lambda_{post} = \lambda + (\lambda - \lambda_{pre}) \tag{36}$$

where λ_{post} denotes the arrival rate of next slot, λ_{pre} and λ mean the measured arrival rate of previous slot and current slot respectively.

Table 3 Metadata about traces

#	*Source*	*Date*	*Time*	*#Records*
1	EPA-HTTP	30 Aug. 1995	09:00–17:00	31,385
2	EPA-HTTP	30 Aug. 1995	16:00–24:00	14,714
3	SDSC-HTTP	22 Aug. 1995	09:00–17:00	15,479
4	SDSC-HTTP	22 Aug. 1995	16:00–24:00	7178
5	NASA-HTTP	01 Jul. 1995	00:00–08:00	16,481
6	NASA-HTTP	01 Jul. 1995	09:00–17:00	24,021
7	NASA-HTTP	01 Jul. 1995	16:00–24:00	25,476
8	NASA-HTTP	25 Jul. 1995	00:00–08:00	9360
9	NASA-HTTP	25 Jul. 1995	09:00–17:00	34,965
10	NASA-HTTP	25 Jul. 1995	16:00–24:00	20,652

Figure 4 Evolution of total arrival requests over time (see online version for colours)



The servers are partitioned into ten groups for every service instance, each group with a FIFO (First In First Out) waiting queue. At the end of each time slot, the system re-allocates the resources among all the service instances according to the predicted arrival rate in next time slot. The specific parameters are listed in Table 4.

Table 4 Parameters and their default values

Parameter	*Default values*
Service Time Distribution	Negative Exponential
Service Rate μ	Random (5…15)
Number of service instances m	10
Assurance factor demand	Random (0.7…0.98)
Response demand	Random (3…10) second
Number of Servers N	90

Figures 5 and 6 are the ASF and RSF pricing mechanisms based revenue every 5 minutes. Ninety servers are deployed in the simulations. It can be seen that both ASF and RSF outperform Heuristic. The average revenue every 5 minutes (excluding the first 30 minutes) of ASF and Heuristic is 8.2k$ and 1.6k$ respectively. The former is four times higher than the latter. The average revenue every 5 minutes (excluding the first 30 minutes) under RSF pricing mechanism of RSF and Heuristic is 8.7k$ and 6.6$. The former is 32% higher than the latter. Figures 2 and 3 also show us that ASF and RSF are more significant if the available computing resources are relatively rare in the simulations.

Simultaneously, both figures show us that the revenue of ASF and RSF during the 47th time slot decreases sharply, even lower than Heuristic. We believe that it results from the extreme volatility of arrival rate. As displayed in Figure 4, the arrival rate decreases sharply from 42nd to the 45th time slot; while it rises quickly after the 46th time slot. Thereby, the prediction of arrival rate by Expression (36) is not correct, which misleads the resource allocation. What's more, because the algorithms of ASF and RSF are accurate and heuristic is not, the effect of ASF and RSF depends on the arrival rate prediction more sensitively. A more refined predicting algorithm of arrival rate improves the quality of ASF and RSF much.

Figure 5 Evolution of revenue over time under ASF pricing model (see online version for colours)

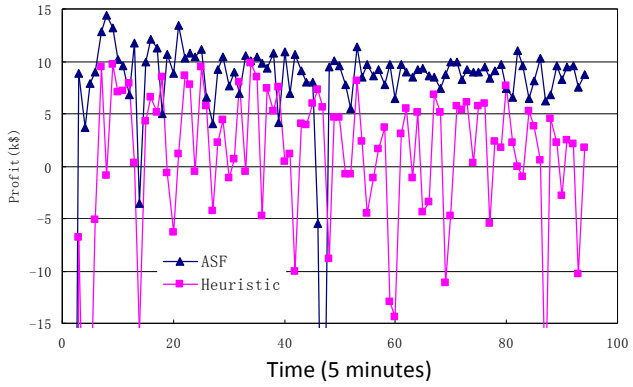
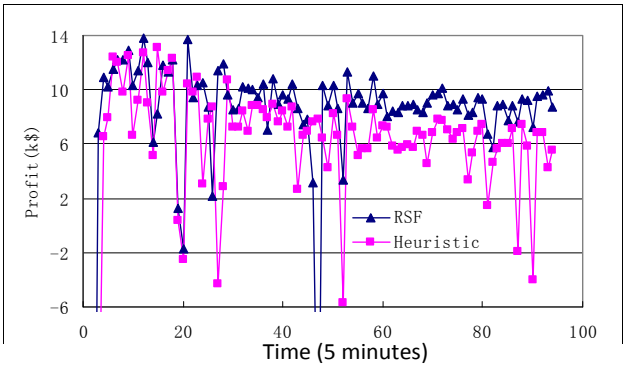


Figure 6 Evolution of revenue over time under RSF pricing model (see online version for colours)



6 Conclusions

Cloud computing has changed the way how applications are delivered to customers. In this new computing paradigm, service level agreements play an important role to facilitate the collaboration between end-users and service providers. This paper addresses how to maximise providers' revenue under the pricing models in terms of TSF (Time Service Factor) in SLAs. This paper has formulated the optimisation problem and given the optimal results by the Method of Lagrange Multiplier. Our simulations have also validated the conclusions. The experimental results have shown that the proposed algorithms in this paper always outperform related work. The proposed algorithms are of higher significance especially when the Cloud is faced with computing resource shortage.

Pricing model is the foundation of our work. A nice price model plays an important part not only in theory but also in practice. We will further these problems in the future work.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 61073028), Qing Lan Project and PAPD Program of Jiangsu Province. We thank Saurabh Garg for his guidance on improving the quality of this paper. The first author would like to thank Jiangsu Provincial Government and the CLOUD Laboratory for supporting and hosting his visit to the University of Melbourne, Australia.

References

Amato, F., Moscato, V., Mazzeo, A. and Picariello, S. (2013) 'Exploiting cloud technologies and context information for recommending touristic paths', *IDC 2013: Proceedings of International Distributed Computing*, Prague, Czech Republic, pp.281–287.

Bonvin, N., Papaioannou, T.G. and Aberer, K. (2011) 'Autonomic SLA-driven provisioning for cloud applications', *CCGRID 2011: Proceedings of the 11th International Symposium on Cluster, Cloud and Grid Computing*, Newport Beach, CA, USA, pp.434–443.

Buyya, R., Lee, C., Lehoczky, J. and Siewiorek, D. (1997) 'A resource allocation model for QoS management', *RTSS 1997: Proceedings of 18th IEEE Real-Time Systems Symposium*, San Francisco, California, USA, pp.298–307.

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 fifth utility', *Future Generation Computer Systems*, Vol. 25, No. 6, pp.599–616.

Chandra, A., Gong, W. and Shenoy, P. (2003) 'Dynamic resource allocation for shared data centers using online measurements', *IWQoS: Proceedings of 11th IEEE/ACM International Workshop on Quality of Service*, Berkeley, CA, USA, pp.381–400.

Daniel, A.M., Daniel, B. and Ronald, D. (2001) 'Preserving QoS of e-commerce sites through self-tuning: a performance model approach', *ACM EC 2001: Proceedings of 3rd ACM Conference on Electronic Commerce*, Tampa, Florida, USA, pp.224–234.

Garg, S., Srinivasa K. and Buyya, R. (2011) 'SLA-based resource provisioning for heterogeneous workloads in a virtualized cloud datacenter', in Xiang, Y., Cuzzocrea, A., Hobbs, M. and Zhou, W. (Eds): *Algorithms and Architectures for Parallel Computing*, Springer-Verlag Berlin, Heidelberg, Germany, pp.371–384.

Ghosh, S., Rajkumar, R., Hansen, J. and Lehoczky, J. (2003) 'Scalable resource allocation for multi-processor QoS optimization', *ICDCS 2003: Proceedings of 23rd International Conference on Distributed Computing and Systems*, Columbus, Ohio, USA, pp.174–183.

Goiri, I., Guitart, J. and Torres, J. (2012) 'Economic model of a cloud provider operating in a federated cloud', *Information Systems Frontiers*, Vol. 14, No. 4, pp.827–843.

Gong, C., Liu, J., Zhang Q., Chen, H. and Gong, Z. (2010) 'The characteristics of cloud computing', *ICPPW 2010: Proceedings of 39th International Conference on Parallel Processing Workshops*, Pittsburgh, PA, USA, pp.275–279.

Goudarzi, H. and Pedram, M. (2011) 'Multi-dimensional SLA-based resource allocation for multi-tier cloud computing systems', *IEEE CLOUD 2011: Proceedings of the 4th International Conference on Cloud Computing*, Washington DC, USA, pp.324–331.

Goudarzi, H., Ghasemazar, M. and Perdram, M. (2012) 'SLA-based optimization of power and migration cost in cloud computing', *CCGRID 2012: Proceedings of the 12th International Symposium on Cluster, Cloud and Grid Computing*, Ottawa, Ontario, Canada, pp.172–179.

Hansen, J.P., Ghosh, S., Rajkumar, R. and Lehoczky, J. (2004) 'Resource management of highly configurable tasks', *IPDPS 2004: Proceedings of 18th International Parallel and Distributed Processing Symposium*, Santa Fe, New Mexico, USA, p.116a.

Householder, R., Arnold, S. and Green, R. (2014a) 'On cloud-based oversubscription', *International Journal of Engineering Trends and Technology*, Vol. 8, No. 8, pp.425–431.

Householder, R., Arnold, S. and Green, R. (2014b) 'Simulating the effects of cloud-based oversubscription on data center revenues and performance in single and multi-class service levels', *CLOUD 2014: Proceedings of the 7th IEEE International Conference on Cloud Computing*, Anchorage, Alaska, USA.

Internet Traffic Archive (2011) Available online at: http://ita.ee.lbl.gov/html/traces.html (accessed on 23 August 2011).

Kertesz, A., Kecskemeti, G. and Brandic, I. (2014) 'An interoperable and self-adaptive approach for SLA-based service virtualization in heterogeneous cloud environments', *Future Generation Computer Systems*, Vol. 32, No. 1, pp.54–68.

Levy, R., Nagarajarao, J., Pacifici, G., Spreitzer, M., Tantawi, A. and Youssef, A. (2003) 'Performance management for cluster based web services', *IM 2003: Proceedings of IEEE 8th International Symposium on Integrated Network Management*, Colorado Springs, CO, USA, pp.247–261.

Li, Y., Sun, K., Qiu, J. and Chen, Y. (2005) 'Self-reconfiguration of service-based systems: a case study for service level agreements and resource optimization', *ICWS 2005: Proceedings of IEEE International Conference on Web Services*, Orlando, Florida, USA, pp.266–273.

Liu, Z., Squillante, M.S. and Wolf, J.L. (2001) 'On maximizing service-level-agreement profits', *ACM EC 2001: Proceedings of the 3rd ACM Conference on Electronic Commerce*, Tampa, Florida, USA, pp.213–223.

Mazzucco, M. (2009) *Revenue Maximization Problems in Commercial Data Centers*, Unpublished PhD Thesis, University of Newcastle, Newcastle, UK.

Mohammed, N.B. and Daniel, M. (2005) 'Resource allocation for autonomic data centers using analytic performance models', *ICAC 2005: Proceedings of the Second International Conference on Autonomic Computing*, Seattle, WA, USA, pp.229–240.

Püschel, T., Borissov, N., Neumann, D., Macías, M., Guitart, J. and Torres, J. (2010) 'Extended resource management using client classification and economic enhancements', in Neumann, D., Baker, M., Altmann, J. and Rana, A. (Eds): *Economic Models and Algorithms for Distributed Systems*, Birkhäuser Basel, Basel, Switzerland, pp.129–141.

Thomas, G.R. (2000) 'Computer networks and systems: queue theory and performance evaluation', 3rd ed., Springer-Verlag, New York, USA.

Villela, D., Pradhan, P. and Rubenstein, D. (2007) 'Provisioning servers in the application tier for e-commerce systems', *ACM Transactions on Internet Technology*, Vol. 7, No. 1, pp.57–66.

Walsh, W.E., Tesauro, G., Kephart, J.O. and Das, R. (2004) 'Utility functions in autonomic systems', *ICAC 2004: Proceedings of the International Conference on Autonomic Computing*, New York, USA, pp.70–77.

Wikipedia (2014) *Service-level agreement*. Available online at: http://en.wikipedia.org/wiki/Service-level_agreement (accessed on 20 March 2014).

Wu, L., Kumar, S.G. and Buyya, R. (2011) 'SLA-based resource allocation for software as a service provider (SaaS) in cloud computing environments', *CCGRID'11: Proceedings of the 11th International Symposium on Cluster, Cloud and Grid Computing*, Newport Beach, CA, USA, pp.195–204.

Zhang, L. and Ardagna, D. (2004) 'SLA based revenue optimization in autonomic computing systems', *ICSOC 2004: Proceedings of the 2nd International Conference on Service Oriented Computing*, New York City, USA, pp.173–182.

Zhu, H., Tang, H. and Yang, T. (2001) 'Demand-driven service differentiation in cluster-based network servers', *InfoCom 2001: Proceedings of IEEE InfoCom*, Anchorage, Alaska, USA, pp.679–688.