

# Brokered pricing model for agent-based negotiation of data-driven services in the cloud

Gokay Burak Akkus<sup>1</sup> and Erdem Ucar<sup>2</sup>

<sup>1</sup>Internet and Network Technologies, Yeditepe University, Istanbul, Turkey

<sup>2</sup>Department of Management Information Systems, Trakya University, Edirne, Turkey

Received: 13 March 2021, Accepted: 13 April 2021

Published online: 18 April 2021.

---

**Abstract:** Since the introduction of cloud computing, integration on the web became a reality. Site-to-site integrations and “Software as a service” approach have been widely accepted as industry standards. The true potential of such a distributed infrastructure is realized when these services are combined together and executed as parts of a singular workflow. The collective composition of web services requires adaptive and dynamic systems where all parties may be subject to unexpected changes. Intelligent systems are needed that are able to make use of services as building blocks that meet user requirements with specific functional and non-functional attributes. The most favorable service is then to be selected from a set of discovered candidates. The goal is to provide computing services to users on demand and charge them based on their usage and Quality of Service (QoS) expectations. In cloud environments, in the presence of multiple services meeting the requirements, the price becomes the main discrimination point. In this paper, we present a general framework that includes the proposal for a brokered pricing protocol for agent-based negotiations and a pricing model study for data driven web services domain hosted on a cloud environment.

**Keywords:** Value-added Services; Cloud Computing; Pricing; Cost Based Pricing; Agent Based Negotiation.

---

## 1 Introduction

Web services, as discussed here, are mainly information services. These services provide existing information rather than modify the world. This is as opposed to the so-called “world altering” services. Since these services change the state of the world, such services introduce state complexity into the composition process and make it harder to determine next actions of the composition process. The web services of our interest are built on top of a cloud infrastructure, and they are bound to the underlying resource usage and incurred costs.

The traditional role of service provider in a Cloud Computing environment is divided into two; the infrastructure providers who manage cloud platforms and lease resources according to a usage-based pricing model and service providers who rent resources from one or many infrastructure providers to serve the end users [6]. The cloud makes profit from selling its services at a price that is higher than the actual cost. Setting the right price for a service is a nontrivial problem, because when there is competition, the demand for services grows inversely but not proportionally to the price [3]. The service provider seeks for a revenue maximization solution that uses optimized pricing methods for random demand and service time[1]. There are two major challenges when trying to define an optimal pricing scheme for the cloud service. The first is to define a simplified enough model of the price-demand dependency, to achieve a feasible pricing solution, but not an oversimplified model that is not representative [7]. The second is to make the model aligned with underlying service characteristics, such as the data that is provided through the service and QoS parameters announced within service attributes.

In this research, we present a system that guides service discovery and service composition by integrating a resource based price estimation. In addition, we propose a protocol to leverage the dynamic pricing system to enable agent-based negotiations. As the momentum around Web services grows, there is an increasing need for effective mechanisms of Web services interaction. In order to test our approach, we developed a sample cloud application that evolves with data growth and we harnessed resource usage details per service call. The proposed framework is applied on this prototype to estimate the possible price to execute a specific service with the given attributes. The estimation uses past data, considers data growth and classification of input and output parameters using data mining methods.

Pricing for value-added services is a pricing strategy where the value-added features are provided to differentiate a market offering and support higher prices [8]. To increase their pricing power, many companies adopt this strategy. Rather than cutting prices to match competitors, they attach value-added features and services to differentiate their offers and thus support higher prices. Their focus is not mainly based on the price to boost annual sales; rather, they keep customers loyal by providing service that the customers cannot find anywhere else.

## 2 Services in the cloud

### 2.1 Cloud computing services

Cloud computing can be categorized into three main groups [9]; IaaS, PaaS, and SaaS in increasing order of abstraction.

#### **IaaS (Infrastructure as a Service)**

Infrastructure as a Service (IaaS) is a provision model in which an organization outsources the equipment used to support operations, including storage, hardware, servers and networking components. The service provider owns the equipment and is responsible for housing, running and maintaining it. The client typically pays on a per-use basis [12].

#### **PaaS (Platform as a Service)**

Platform as a Service (PaaS) is a way to rent hardware, operating systems, storage and network capacity over the Internet. The service delivery model allows the customer to rent virtualized servers and associated services for running existing applications or developing and testing new ones. On the downside, PaaS involves some risk of "lock-in" if offerings require proprietary service interfaces or development languages. Another potential pitfall is that the flexibility of offerings may not meet the needs of some users whose requirements rapidly evolve [11].

#### **SaaS (Software as a Service)**

Software as a Service (SaaS) is a software delivery method that provides access to software and its functions remotely as a Web-based service. SaaS allows organizations to access business functionality at a cost typically less than paying for licensed applications since SaaS pricing is based on a monthly fee. Also, because the software is hosted remotely, users don't need to invest in additional hardware. SaaS removes the need for organizations to handle the installation, set-up and often daily upkeep and maintenance [10].

### 2.2 Web services

According to W3C Glossary (Web Services Glossary, <http://www.w3.org/TR/ws-gloss/>) definition, a web service is a service that is accessible by means of messages sent using standard web protocols, notations and naming conventions, including the XML Protocol.

Web Services Architecture is made up of three major components; service requester, service provider and discovery agency. Service provider is the owner of the service and the data provided. It processes the web service request. Service requester is an application or a client requesting the execution of a web service. Discovery agency is an agency through which a Web service description is published and made discoverable [13].

### 3 Pricing schemes

Dynamic pricing is defined as changing the price of a service/product with respect to the market demand. It involves expectations, predictions, competitions, demand profiles, demand functions. It is not an easy task for cloud providers to make informed decisions about pricing their services [23].

The market insights could play a significant role in pricing a service. In order to address such environmental factors, one might devise a Stackelberg game with some structure that reflects the state of the dynamic system and decision variables chosen by game players.

At any time  $t$ , a set of variables called the state variables should be in place. Examples of such state variables are market share, sales, inventory, production cost, goodwill and alike. On top of state variables, there are decision variables which are chosen by the players. These include order/production quantity, advertising, shelf-space decisions. The state variables evolve over time, and this is represented through differential equations on state and control variables. The players try to optimize against these values, each having an objective function [23]. In this paper we assume sequential decision making in noncooperative game format.

### 4 Definition of the problem

In the rapidly changing environment of information technology, it is becoming more important for systems to utilize cloud computing [4]. Cloud service providers mainly adapt a cost based pricing for the services they provide. This cost is generally determined by resource utilization. The main resources affecting the cost are processor and memory usage and total (inbound and outbound) network utilization. However, the pricing can be negotiated for guaranteed resource usage and QoS agreement.

It is important for service providers to calculate the cost of executing a service beforehand so that they can calculate a price for the service requester. The dynamic pricing of executing services requires a dynamic classification of web services. This research proposes a cost estimation model for execution of web services on cloud platforms. Web services can be classified into two; data driven services and services that change the state in their execution environment. Data driven services mainly provide data, rather than manipulating them. These data later can be used on business intelligence, customer relationship management and data mining applications.

Data driven services can be enhanced to provide the pricing of executing them. The pricing would be calculated on the result of the cost estimation model corrected by historical realizations. This enables the service requesters to calculate their cost in advance and make decisions accordingly. To this end, the "naked" data service needs to be wrapped by an external-facing service, which would be expected to conform to a protocol that would allow prediction of costs for a reasonable time period, and would also be expected to facilitate bookkeeping and financial transactions. For simplicity, we consider metered pricing within a cloud environment where the service is provided by another intermediary party to the consumer with constrained capacity. We do not consider external influences where the provider may be increasing its service cooperating with other providers or oppositely, competing with them in order to increase market share [14, 15, 16].

#### *4.1 Data driven web services*

The deployment of web services on the cloud is beneficial for integration services between companies and services providing consolidated output of business intelligence algorithms that run on big data.

The main objective of this research is to come up with a cost estimation model that is beneficial for both service provider and service requester. The main advantage for the service provider is to be able to competitively price the execution of a service by closely estimating the cost. On the other hand, the service requester benefits from knowing the price before executing a service.

#### *4.2 Services that cause change of state in their execution environment*

The web services that update specific parts of data space at run-time might cause a change of state within the execution environment. This change may introduce a significant amount of uncertainty, which could have a large and unexpected impact on predictions. In this research, we do not take these types of services into account. We assume that the underlying data grows in time which can be determined.

### **5 Cost modeling for data driven web services**

The pricing for web services on cloud systems are mainly calculated per usage bases. The usage is based on the amount of total network traffic, the processing power required, the resource (memory and disk) utilization and QoS agreement. The QoS have been treated as a component of our price modeling scheme, and possible effects have been measured by using public datasets [18].

Another cost calculation scheme might import a base price through matching of services to the consumers of such services using partially-observable consumer and service attributes [2].

#### *5.1 Components of the price*

The components of the pricing for cloud service usage are the cost of resource utilization, insurance allocation for the cost of errors and the profit margin per customer (customer contract) and the market insights (concurrent demand, predictions, number of possible clients, lifetime expectancy of the service). The cost of resource utilization is mostly the main focus considering these price components, it has the highest perceived material weight among all [5].

#### *5.2 Cost modeling*

This modeling can be both used on differentiating among services and development of services based on customers pricing needs.

In this research, the parameters shown in Table 1 are used for cost estimation modeling:

**Table 1:** Service cost calculation.

Parameter	Requirement	Cost Calculation
Md	Data storage	Size (total) × Subscription multiplier × Unit price (storage)
Mb	Computation	Cost (machine)
Mti	Data transfer – IN	Cost (input traffic)
Mtd	Data Transfer – OUT	Cost (output traffic)
Mht	Unexpected errors or timeout occurrences	If the request fails to respond with a probability of P or an unexpected result occurs or an error occurs, then another price is charged.
Mab	Revenue that is based on client identity	The profit parameter should be based on the subscription type and the profit margin should counterbalance the cost of errors and timeouts.
Mi	Market insight	A Stackelberg game formulation

The cost estimation may be formulated using the model parameters as follows.

$$(Md + Mb + Mti + Mtd) + Mab + Mht + Mi \tag{1}$$

The parameters in the formula may vary according to the cloud service provider and market perception of the data-driven service provider. Additionally, these parameters shall be weighted according to internal dynamics of the service.

In many applications or services, before the data becomes readily available, a pre-processing is required. This could be in the form of background processing, the initialization of execution environment, or ETL processes. For simplicity, the effect of such pre-processing on cost calculation is left for future work.

The market insight parameter specified above is a complex structure, and could be a research topic at full-scale. In order to keep the focus of this research mainly in cost components while addressing protocol definitions and price negotiations, the market insights is assumed to be reflected in the formulation as a simplified component, but not totally omitted. This component is kept in cost calculation in order to include lifetime expectancy of a service, for its correlation with the underlying contractual terms of cloud infrastructure rental scheme.

### 5.3 The impact of input parameters

The input parameters used in calling the service have a direct effect on the output. Therefore, the input parameters are used as a basis for the cost estimation modeling. There are two steps while using input parameters within the process. The first step is to map each of the parameters to an input class. As a future work, we propose to make this classification in an automated fashion where a learning algorithm handles the process of both identifying classes and the mapping of each input parameter to the corresponding class. The classification of parameters is realized with the  $f_i$  function (2). This function maps each input parameter value to another value in a specific numeric set.

$$\forall i f_i(p_i) = c_i \tag{2}$$

Classification function  $f_i$  identifies the corresponding numeric parameter class of every parameter. This function is mutually exclusive:

**Example 1.** In case  $p_1 = \text{''Ali''}$  then parameter classification is made as  $c_1 = 1$  and this numeric value is used in predictions.

The prediction is mainly calculated on time series analysis. The historical realizations are used in estimating the future outcomes. The model also takes into account the growing trend of data and the effect of the input parameters on the

output.

The second step is predicting utilization and output values. After classification of service parameters, the parameters are used on two different types of predictions:

- (1) Output prediction (This prediction will also be used on resource utilization prediction).
- (2) Resource utilization prediction (CPU, memory, disk, network bandwidth usage).

## 6 A protocol for dynamic cost negotiation

Once the parameters for cost prediction are known, a protocol needs to be defined for the use of the cloud web service in question while enabling dynamic prediction of price, and agent-based negotiation. The protocol to be used in this case needs to satisfy the following properties:

- (I) For the user of the service, a cost must be calculated for any given query, and that cost should be guaranteed for a determinate, reasonable amount of time.
- (II) The protocol should include either an authentication mechanism, or a way to provide payment information for use of the service.
- (III) The protocol implementation must ensure that a different query may not be executed for a given cost.
- (IV) The protocol must ensure pricing predictions should only be made for valid queries. Conversely, any query that receives a pricing prediction must be able to be properly executed.

We propose the following protocol that satisfies these conditions:

Let us call the “naked” service  $\mathbf{S}$  and its parameter set  $\mathbf{P}$ . The goal of the user is to call the service with parameters  $\mathbf{P}$ , that is, execute  $\mathbf{S}(\mathbf{P})$ . To this end, the service provider implements two wrapper services,  $\mathbf{W}_1$  and  $\mathbf{W}_2$ .

$\mathbf{W}_1(\mathbf{P}, \mathbf{A})$  is the pricing service, and takes the parameters of the naked service  $\mathbf{P}$ , authentication or payment information  $\mathbf{A}$ . It first validates the authentication or payment information, and rejects the call if that fails. If not, it validates  $\mathbf{P}$ . If the validation fails, it returns an error. Otherwise, it creates a ticket  $\mathbf{T}$  for the service user, including the predicted cost (using the cost model), and estimates for the execution time and amount of data to be returned. The cost is final (that is what the user will pay) but the rest are understood to be estimations. It also returns a validity date for the ticket – this reflects the dynamic nature of the service. It may be of the order of minutes, hours, or days depending on the nature of the service and load patterns.

$\mathbf{W}_2(\mathbf{T})$  is a wrapper for the naked service, but it takes the ticket  $\mathbf{T}$  generated by  $\mathbf{W}_1$  only. The actual calling parameters must be stored by the service provider, as well as the expiration time of the ticket. The actual service  $\mathbf{S}$  will be called only if the ticket  $\mathbf{T}$  is valid and has not expired. Ultimately,  $\mathbf{W}_2$  returns  $\mathbf{S}(\mathbf{P})$  to the user, while also charging the user’s account with the amount in the ticket.

## 7 Machine learning and test setup

### 7.1 Machine Learning

**Definition 1.** *A computer program is said to learn from experience  $E$  with respect to some class of tasks  $T$  and performance measure  $P$ , if its performance at tasks in  $T$ , as measured by  $P$ , improves with experience  $E$ . In this research, we implement a predictive model and then test it with different classification algorithms. Predictive modeling is used to estimate the optimal probability of an outcome. Mostly, the model is constructed in a way, where the probability of an outcome is estimated against a given set of input data with historical data.*

## 7.2 Test setup

Machine learning methods are conducted on WEKA (Waikato Environment for Knowledge Analysis) tool. WEKA is data mining software developed on Java [17]. In the experimental design, WEKA version 3.6.9 is used on a computer running MS Windows 10 with 4GB of RAM, 1.5 GHz CPU and 100 GB SSD.

## 7.3 Data collection and execution of predictive algorithms

We created a new sample application on top of a data set from a web application that indexes job advertising sites. A job ad contains information about the title of the company, the job title, requirements and job description. Also, the publication date of the ad is on record. For our purposes, we built a web service that accepts the following parameters to search for job ads:

- (I) Search start date
- (II) Search end date
- (III) Job title query (may include wildcards, or may be null)
- (IV) City (may be null, or be a specific city)

The service was called using a set of queries derived from actual usage logs of the (now defunct) web site ([www.islerburada.com](http://www.islerburada.com)). The following information was collected for each call by wrapping the actual service call with another logging function:

- (I) The CPU load on the machine during query execution
- (II) The number of job ads returned as a result of the query
- (III) The total amount of data transmitted
- (IV) The amount of I/O performed by the machine for the query
- (V) The CPU time used by the query
- (VI) The real time elapsed during query execution

### Classification function

The classification function is one of the main fields of interest for this research. It builds the backbone for the algorithms to function within an expected performance range. In this research, for the sake of simplicity and to increase accuracy, we implemented a manual guided classification scheme, where a domain expert defines the classification rules that will enable the mapping of each value of parameters to a set of values, which will be used within the prediction algorithm.

The classification scheme may be automated as a future work, again with the help of data mining methods or neural networks, fuzzy approaches, Support vector Machines [19]. Within the scope of this work, the parameters of the method call were manually classified and preprocessed as follows:

Since it was observed that only the number of days present in the query was of predictive value (the number of ads per day does not significantly change), for prediction purposes, the two date limit parameters were mapped to a single "days" parameter. The job title parameter was classified to the following values:

- (i) Null.
- (ii) does not contain.
- (iii) ends with .
- (iv) contains (not in the first character).

The classification for the city was as follows:

- (i) Null.

- (ii) Is a "large" city (four cities with most job ads)
- (iii) Is another city.

In addition, the CPU load on the host machine was also included as a numeric value. This is in fact a factor independent of the given query, but useful in determination of the response time. It is also a factor that reflects the dynamic nature of the prediction model.

## 8 Analysis of test results

### 8.1 Training Set

The main data set is divided into two portions. The first part, that is 50% of the data, is used for statistical model development and training. The rest of the data is then used for the validation of the predictions and verification of using different clustering and prediction algorithms. Since we have two numeric (day count and CPU load) and two classified (job title and city) parameters to be used in the prediction model, analysis using WEKA was found to be most fruitful using the M5P[20][21] algorithm. This algorithm combines a conventional decision tree with linear regression functions at the leaf nodes.

### 8.2 Prediction model

Using the M5P algorithm, a prediction model was generated for each of time, CPU, I/O and total data transmission values. The prediction model gave reasonably good correlation when tested with 50% of the data, and the results are shown in Table 2.

**Table 2:** Predicted cost parameters

Parameter	Correlation Coefficient	Relative Mean Absolute Error	Number of Instances
Time	0.9415	18.6 %	48600
Cpu	0.9436	17.5 %	48600
I/O	0.7361	34.3%	48600
Data	0.7344	34.4 %	48600

## 9 Conclusion and future work

Pricing schemes, cost prediction and a protocol for the utilization of data driven services running on top of a cloud infrastructure have been investigated in this research.

A web service was built on top of real-world data, with both classified and numeric parameters. In addition, a dynamic service-side parameter was also included in the prediction model. The M5P algorithm was found useful for such a setup; and the predicted values showed reasonable correlation with the actual patterns.

A protocol has also been described, ensuring easy implementation and useful features for the user and provider of the service. The protocol is sufficiently general to be utilized by any web service with little to no modification and can be implemented as a software library. The prediction algorithm uses historical data. Some of the data can be extracted directly from "traditional" logs; however for a useful prediction, additional parameters may need to be logged. We have had to implement and use a logging wrapper function to this end to log all required information.

The protocol might also be extended as a future work, to include some reservation models in order to address the service-time component of a service call while identifying options to specify time-slots and fix a reserved price for a specific period of time.

The market insights component described in the paper might be a focus of further research while referring to economics of services market in a world of AI supported autonomous personal digital assistants that will allow machine-to-machine interactions commonly available.

## Competing interests

The authors declare that they have no competing interests.

## Authors' contributions

All authors have contributed to all parts of the article. All authors read and approved the final manuscript.

## References

- [1] Doan X.V., Lei X., Shen S. Pricing of reusable resources under ambiguous distributions of demand and service time with emerging applications. In: European Journal of Operational Research, Volume 282, Issue 1, 2020, pp. 235-251.
- [2] Izhutov P., Mendelson H. Pricing a Digital Services Marketplace Under Asymmetric Information. In: Miller J., Stroulia E., Lee K., Zhang L.J. (eds) Web Services-ICWS 2019. ICWS 2019. Lecture Notes in Computer Science, vol 11512. Springer, Cham, 2019, <https://doi.org/10.1007/978-3-030-23499-7-2>.
- [3] Singh H. AWS Pricing and Cost Management. In: Practical Machine Learning with AWS. Apress, Berkeley, CA., 2021, <https://doi.org/10.1007/978-1-4842-6222-1-2>.
- [4] Kansal, S., Kumar, H., Kaushal, S. et al. Genetic algorithm-based cost minimization pricing model for on-demand IaaS cloud service. J Supercomput 76, 2020, pp. 1536-1561.
- [5] Lee, I. Pricing schemes and profit-maximizing pricing for cloud services. J Revenue Pricing Manag 18, pp. 112-122, 2019, <https://doi.org/10.1057/s41272-018-00179-X>.
- [6] Bigala P.; Ekabua O.O. Implementation of novel accounting, pricing and charging models in a cloud-based service provisioning environment. In SDWIC 2013 The International Conference on Electrical and Electronics Engineering, Clean Energy and Green Computing (EEECEGC2013), Dubai, UAE, 11-13 December 2013. In The Society of Digital Information and Wireless Communications (SDIWC), Washington, D.C, USA, 2013, pp. 1-9.
- [7] Kantere V.; Dash D.; Francois G.; Kyriakopolou S.; Ailmaki A. Optimal service pricing for a cloud cache. IEEE Transactions on Knowledge and Data Engineering, 2011, 23:1345-1358.
- [8] Kotler P. Marketing Management, 7th ed., New Jersey, NJ, USA, Prentice Hall, 1991.
- [9] Litoiu M.; Woodside M.; Wong J.; Ng J.; Iszlai G. A business driven cloud optimization architecture. In ACM 2010 Proceedings of the 2010 ACM Symposium on Applied Computing (SAC 2010), Sierre, Switzerland, 22-26 March 2010, New York, NY, USA, ACM; pp. 380-385.
- [10] Bhardwaj S.; Jain L.; Jain S. Cloud computing: A study of infrastructure as a service (IAAS). International Journal of Engineering and Information Technology, 2010, 2; 60-63.
- [11] Boniface M.; Nasser B.; Papay J.; Phillips S.C.; Servin A.; Yang X.; Zlatev Z.; Gogouvitis S.V.; Katsaros G.; Konstanteli K. et al. Platform-as-a-service architecture for real-time quality of service management in clouds. In IEEE 2010 Internet and Web Applications and Services (ICIW), 2010 Fifth International Conference; Barcelona, Spain, 9-15 May 2010. New York City, New York, USA, IEEE; pp. 155-160.

- [12] Buxmann P.; Hess T.; Lehmann S. Software as a service. *Wirtschaftsinformatik*, 2008; 50: 500-503.
- [13] Akkus G.B.; Ucar E. Network algorithms for agent based automation of web services composition. In *International Symposium on Innovations in Intelligent Systems and Applications (INISTA 2010)*, Kayseri, Turkey, 21-24 June 2010; pp. 214-218
- [14] Kurose J.F.; Simha R. A microeconomic approach to optimal resource allocation in distributed computer systems. *IEEE Transactions on Computers*, 1989; 38: 705-717.
- [15] Nagle T.T.; Hogan J.E. *The Strategy and Tactics of Pricing: A Guide to Growing More Profitably*, 4th ed., Upper Saddle River, NJ, USA, Prentice Hall, 2006.
- [16] Yeoa C.S.; Venugopalb S.; Chua X.; Buyyaa R. Autonomic metered pricing for a utility computing service. *Future Generation Computer Systems* 2010; 26: 1368-1380.
- [17] Hall M.; Frank E.; Holmes G.; Pfahringer B.; Reutemann P.; Witten I.H. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter*, 2009; 11: 10-18.
- [18] Zhang Y.; Zheng Z.; Michael Lyu R.; WSPred: A time-aware personalized QoS prediction framework for web services. In *IEEE 2011 Proceedings of the 22th IEEE Symposium on Software Reliability Engineering (ISSRE 2011)*, Hiroshima, Japan, 29 November- 2 December 2011. New York, NY, USA, IEEE; pp. 210-219.
- [19] Kalkan H.; Online feature selection and classification with incomplete data. *Turkish Journal of Electrical Engineering & Computer Sciences*, 2014; 22: 1625-1636.
- [20] Quinlan Ross J.; Learning with Continuous Classes. In *5th Australian Joint Conference on Artificial Intelligence*, Singapore, 1992; 343-348.
- [21] Wang Y.; Witten I. H. Induction of model trees for predicting continuous classes. In *Poster papers of the 9th European Conference on Machine Learning*, 1997.
- [22] Li T.; Sethi S.P. A Review of Dynamic Stackelberg Game Models. In *Discrete and Continuous Dynamical Systems Series B*, Volume 22, Number 1, January 2017, pp. 125-159.
- [23] Lee, I. Pricing schemes and profit-maximizing pricing for cloud services. *J Revenue Pricing Manag* 18, 2019, pp. 112-122. <https://doi.org/10.1057/s41272-018-00179-X>.