

Ognjen Radović<sup>1</sup>  
Maša Georgiev<sup>2</sup>  
Milica Đorđević<sup>3</sup>  
University of Niš, Faculty of Economics

ORIGINAL SCIENTIFIC ARTICLE  
DOI: 10.5937/ekonomika2104037R  
Received: Jun, 27. 2021.  
Accepted: September, 14. 2021.

## TOTAL COST OF OWNERSHIP OF INFRASTRUCTURE AS A SERVICE FOR SME

### Abstract

*Cloud Computing offers significant cost benefits for SMEs that often do not manage internal IT infrastructure and start-up companies that do not have their own IT infrastructure. This paper presents a total cost of ownership (TCO) approach for cloud computing services with an emphasis on the infrastructure as a service (IaaS) model. Also, the paper presents a methodology for estimating total cost of ownership (TCO) when running computer instances in the IaaS cloud using the GARCH model to predict transaction volatility. The research results show that it is possible to successfully use GARCH models when there is historical data on the number of transactions. In addition, simulation shows that, when there are large oscillations in the number of transactions, the best choice is to reserve instances according to the Partial-Upfront price model. In contrast, if the transaction number is relatively stable, the best choice is the All-Upfront model.*

**Key words:** Cloud computing, Total Cost of Ownership (TCO), GARCH model

**JEL classification:** M41, L86

## УКУПНИ ТРОШКОВИ ВЛАСНИШТВА ИНФРАСТРУКТУРЕ КАО СЕРВИСА ЗА SME

### Апстракт

*Клауд рачунарство нуди значајне трошковне предности за SME које често не управљају интерном IT инфраструктуром и start-up компаније које не поседују сопствену IT инфраструктуру. У овом раду приказан је приступ укупних трошкова власништва (TCO) за услуге рачунарства у облаку са нагласком на модел инфраструктуре као сервиса (IaaS). Такође, у раду је приказана методологија за процену укупних трошкова власништва (TCO) при покретању рачунарских инстанци у IaaS клауду коришћењем GARCH модела за предвиђање волатилности трансакција. Резултати истраживања показују да је могуће успешно користити GARCH моделе када постоје историјски подаци о броју трансакција. Поред тога, на основу симулације, када постоје велике осцилације у броју трансакција најбољи избор је резервација инстанци*

<sup>1</sup> ognjen.radovic@ekonomski.rs, ORCID ID 0000-0001-7428-565X

<sup>2</sup> kulicmasa@yahoo.com, ORCID ID 0000-0002-7572-9547

<sup>3</sup> milica.djordjevic@ekonomski.rs, ORCID ID 0000-0003-1652-4186

*по ценовном моделу Partial-Upfront. Насупрот томе, ако је број трансакције релативно стабилан, најбољи избор је модел All-Upfront.*

**Кључне речи:** *Клауд рачунарство, укупни трошковИ власништва (TCO), GARCH модел*

## Introduction

Cloud computing, as a new paradigm, is developing rapidly and taking precedence among technologies that modern companies are adopting. NIST (National Institute of Standards and Technology) defines cloud computing as “a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (for example, networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service-provider interaction” (Mell & Grance, 2009; Makhlouf, 2020).

There are several models of cloud computing delivery, with three forms being the most common: infrastructure as a service (IaaS), platform as a service (PaaS), software as a service (SaaS), and business process as a service (BPaaS) (Rosati et al., 2019). The essence in any case is the delivery of computer resources at the user request via the Internet, whereby these resources are not bought but rented. According to the chosen model, the user pays only for as many resources as they actually use, so they no longer have to worry about the purchase of hardware and installation, or software maintenance (Ђорђевић et al., 2018).

However, although cloud technology is based on renting, rather than buying computer resources, recent research on the market of cloud computing services calls into question the profitability of using cloud services (Ramchand et al., 2018; Makhlouf, 2020). In some cases, the total cost of using cloud services is “unexpectedly” much higher than initial investment in their implementation, due to continuous maintenance and other hidden costs (Zimmerman, 2014). This makes the choice of using cloud computing services a complicated and expensive process, the more efficient implementation of which can be achieved by more comprehensive inclusion of all possible categories of cloud computing costs, i.e. by applying the Total Cost of Ownership (TCO) approach. TCO is one of the most important cost-oriented approaches that is widely used in various research areas (Strebel & Stage, 2010).

The paper presents a formal mathematical model for calculating the total cost of ownership of cloud computing services focusing mainly on cost factor to determine costs. The work is organized as follows. The first part of the paper explains the total cost of ownership of cloud computing services (TCO) and gives a schematic overview of cloud computing pricing (IaaS). The second part of the paper presents a methodology for analyzing and estimating the number of reserved instances using the GARCH model to predict the volatility of daily transactions of a hypothetical SME simulated by the Ethereum cryptocurrency transactions collected every hour.

## TCO of cloud computing – theoretical background

The decision to use cloud computing services is largely conditioned by adequate cost analysis because the increasing complexity of managing the entire infrastructure of different information architectures and distributed data and software has made computing more expensive than before. The costing model most commonly implemented in research and practice is the TCO model because it provides companies with a comprehensive overview of cost factors (Rosati et al., 2017; Strebel & Stage, 2010). The application of the TCO approach to cloud computing services is vital for users of these services because it avoids the vague assessment of additional costs, both indirect and for the entire life of the service (Cegielski et al., 2012; Vouk, 2008; Makhlouf, 2020). Specifically, the benefits of TCO lie in improving customer-supplier communication and cost analysis throughout the service lifecycle (Ellram & Siferd, 1993). With the help of a predefined scheme, the TCO approach enables the analysis of individual cost components of one IT service (Martens et al., 2012), and the procedure is significantly simplified by including only a limited number of carefully selected cost factors. A summary of the costs by category according to the recommended TCO model for cloud computing is shown in Table 1.

Table 1. Cost categories for cloud computing services

	Service model		
	SaaS	IaaS	PaaS
Management	Strategic decision and selection of cloud services		
Delivery	Implementation, configuration, integration and migration <ul style="list-style-type: none"> <li>▪ Support</li> <li>▪ Initial training</li> </ul>		
Quality	<ul style="list-style-type: none"> <li>▪ Backsourcing or discarding</li> <li>▪ System failure</li> </ul>		
Price	<ul style="list-style-type: none"> <li>▪ Service charge</li> <li>▪ Execution time</li> <li>▪ Costs of input data transfer</li> <li>▪ Costs of output data transfer</li> <li>▪ CPU instance / hour</li> <li>▪ Storage / month</li> </ul>	<ul style="list-style-type: none"> <li>▪ Service charge</li> <li>▪ Load curve</li> <li>▪ RAM</li> <li>▪ Storage and network usage</li> <li>▪ CPU virtual machine hour cost</li> <li>▪ Cost of time contingent</li> <li>▪ Internet bandwidth cost</li> <li>▪ Inbound data transfer costs</li> <li>▪ Outbound data transfer costs</li> </ul>	<ul style="list-style-type: none"> <li>▪ Service charge</li> </ul>
Service	<ul style="list-style-type: none"> <li>▪ Maintenance and modification</li> <li>▪ Permanent training</li> </ul>		
Communication	<ul style="list-style-type: none"> <li>▪ Vendor evaluation and selection</li> </ul>		

Source: Makhlouf (2020), <https://doi.org/10.1186/s13677-019-0149-4>

Table 1 clearly shows that TCO uses the following cost categories: management, delivery, quality, price, service and communication. However, as transaction costs make a significant part of the total costs, we believe that it is necessary to include them in order to more accurately estimate the total costs of using cloud computing services. This ensures fuller TCO application because the process of exchanging goods and services on the market involves a number of transactions with the environment and interactions

with it – negotiations with suppliers, customers, business partners. Transaction costs also include the cost of finding the most favorable suppliers, customers, the cost of negotiating contract terms, overseeing contract signing, adjusting to any changing circumstances, and processing relevant information (Free, 2010). Furthermore, the combined impact of certain human and environmental factors also brings transaction costs. These factor combinations include uncertain environment and limited rationality, opportunism and a small number of partners, risk and specific means and frequency of transactions.

With this in mind, transaction cost theory is a commonly used framework for conducting studies on information technology and cloud computing (Yigitbasioglu, 2014), helping the management structure reach an optimal solution and adequate decision on the adoption or non-adoption of a new technology and implement it in the organization (in this case whether to adopt or reject the decision to use cloud computing services).

Otherwise, the TCO approach for estimating the cost of cloud services takes into account the costs from initial investment to the end of service use, which can be the client transition to other services or suppliers (Rosati et al., 2019). Thus, it is a “procedure that provides a basis for determining the total economic value of an investment”, and the nature of measuring the total cost of cloud computing provides a perfect basis for extremely low-granularity TCO analysis and the ability to perceive how business value can be measured in research and practice (Rosati et al., 2017). Preliminary, i.e. ex-ante, assessment of costs and benefits that investing in the cloud can create is crucial for effective decision-making. This is because the introduction of cloud computing usually dramatically changes the business model and internal organization in the company, and then requires significant resources for the migration process. In addition, the nature of the cloud allows for detailed subsequent cost analysis. In this sense, the total cost of ownership and the TCO cost factors represent the sum of initial investment required to purchase the asset (CapEx) and the operating costs generated by the cloud (OpEx). When choosing appropriate alternatives, it is important to consider both TCO components in order to adequately estimate the cost of using these services. The TCO can be formalized as follows:

$$TCO = CapEx + OpEx$$

When determining TCO in the IaaS cloud, in terms of IaaS operating costs for service providers, the focus should be on computer, storage and network resources as they are usually the most significant costs of this model. IaaS costs can be categorized as (1) fixed (value of reserved/allocated resources, availability, location and other additional and/or premium services) or (2) variable (i.e. use of all relevant IaaS resources) (Rosati et al., 2019).

There are various schemes on today’s market for determining the prices of IaaS services. Most suppliers offer hourly payment (depending on the hours of use), using a cost component rate. Some try to attract customers by low price per GB of storage space while charging hidden costs for inbound and/or outbound data transfer or even for data transfer within the service provider’s infrastructure. In particular, many suppliers offer basic packages at a fixed price that can be extended according to customer needs (variable, direct costs of the service in case the basic package limits are exceeded). It is also possible to pay a fixed price in order to get discounts on prices depending on the use of the service.

The IaaS charging procedure assumes the following: the cost of used computer resources incurred in a given period is calculated by multiplying the number of used processor units by period and the price of one computer unit in that period. The price varies depending on the specific system characteristics (selected type of instance), e.g. number of computer units, storage capacity (in GB), selected operating system (Linux or Windows) and platform (32-bit or 64-bit). The total cost of these factors is determined by summing all costs incurred over all periods. This calculation scheme also applies to storage capacity costs, inbound, outbound and internal data transfer, to other web services from the same provider, and database query costs. The total cost of domains, SSL certificates, software licenses, and basic service charges is determined by multiplying the number of service usage periods by the corresponding cost factor price (Martens et al., 2012).

The TCO model used in this paper relies on the following common TCO requirements (Martens et al., 2012):

- Transparency: a detailed description of the model and applied criteria is provided,
- Applicability: the prototype implemented software tool allows easy application of the TCO model with reasonable effort,
- Variability: The TCO model is standardized to a large extent, but the central aspects are variable, so the desired model changes or extensions are possible,
- Comparability: the results of the model analysis are comparable to each other given a pre-defined framework and transparent calculation schemes.
- Decision support: since the calculated costs are structured according to cost types and factors, the model provides a sufficient basis for a comprehensive analysis to yield significant information.

## 2. TOC calculation

Cloud computing enables faster adaptation to changes in the environment and reduction of IT costs. One of the main advantages of cloud services is the ability to optimize costs according to the changing company needs. As we saw in the previous part of the paper, the basic cost elements of IaaS are: computer (number of servers), storage (memory size in GB), and data transfer. AWS offers several price models depending on the product (AWS, 2020):

- On-demand instances –charging computer and stored resources per hour or minute, on user demand, without long-term commitments and prepaid (upfront) resources.
- Reservation – reserving resources in advance.

The On-demand model is the most expensive cost model per hour of cloud resource use. The user pays the full price, which does not include storage and network costs. However, as there are no long-term commitments or prepayments, this is the most flexible option as it allows the use of cloud resources when needed (AWS, 2020). Like Amazon, Google Compute Engine and Microsoft Windows Azure have a similar approach to the on-demand pricing model.

Cloud providers (such as Amazon EC2, Google Compute Engine, Microsoft Windows Azure) offer the possibility of capacity reservation. In relation to equivalent on-demand capacities, reserved capacities, or reserved instances (RI), can bring savings of up to 72% (AWS, 2020). Reserved instances are available in three options: All up-front (AURI), partial up-front (PURI) or no up-front (NURI). Reserved instances is an option that allows the user to pre-reserve instances of certain properties. The user pays in advance or in monthly installments for a one- or three-year term. The provider (Amazon) also gives significant on-demand discounts when the capacity is used up. Amazon distinguishes three types of instance reservations (AWS, 2020):

- All-upfront – payment is made in advance for the entire rent time, without monthly installments.
- Partial-upfront – partial payment is made in advance (usually about two-thirds) and the rest is divided into monthly installments during the instance reservation period.
- No-upfront – there is no advance payment, only monthly installments are paid during the instance reservation period.

Like the usual purchase of instances, when buying reserved instances, the user chooses the instance type and size, platform and region. There are other cloud providers that offer instance reservation as well, but with somewhat different features. Microsoft uses Azure Reserved VM Instances, and, based on Enterprise Agreements, offers various discounts on Azure resources or Microsoft software licenses. Google (Committed Use Discounts – CUD) gives certain discounts at the end of the month based on the time of instance use, without any prior commitments or payments.

Table 2 gives an example of the AWS pricing model per instance (t2.small). The data in the table is used to calculate the planned costs that will be presented in this paper.

*Table 2. Pricing of on-demand and reserved t2.small instances in Amazon EC2 (General Purpose, Linux, EU(Frankfurt), September 1, 2020.)*

Pricing Option		Upfront	Monthly	Hourly
On-demand		\$0	\$0	\$0.0268
1-Year Reserved	No Upfront	\$0	\$13.94	\$0.019*
	Partial Upfront	\$80	\$6.64	\$0.018*
	All Upfront	\$156	\$0	\$0.018*

\*Effective hourly pricing is shown to help you calculate the amount of money that a Reserved Instance will save you over On-Demand pricing. When you purchase a Reserved Instance, you are billed for every hour or second during the entire Reserved Instance term that you select, regardless of whether the instance is running or not. The effective hourly price shows the amortized hourly cost of the instance (this takes the total cost of the Reserved Instance over the entire term, including any upfront payment, and spreads it out over each hour of the Reserved Instance term).

Source: <https://aws.amazon.com/ec2/pricing/reserved-instances/pricing/>

Cloud providers allow monitoring the use of cloud resources in real time. The user can, for example, monitor the cloud instances use pattern on an hourly basis according

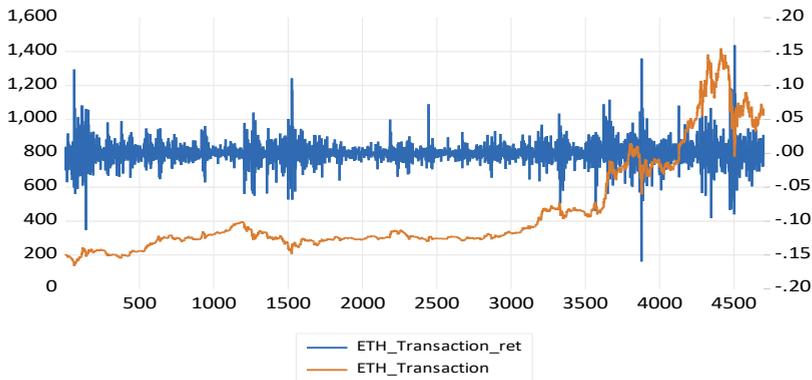
to the performed transactions. Data recording provides the user with information support for estimating the future costs of using cloud resources. Cloud providers state that instance reservation can potentially reduce costs by 70-80% compared to the on-demand model. However, when reserving instances, unused instances are permanently lost. In order to use the full capacity of the reserved instances, all purchased instances must be used. This requires detailed analysis and prediction of future instance needs. Planning the required cloud capacity and calculating the cost of instance rent is a complex task that requires an assessment of computer and storage resources, as well as an assessment of network traffic on a daily basis, but also planning and forecasting several business years in advance. On the provider side, user requirements are constantly changing and all possible combinations of instances and virtual machines should be explored. On the user side, it is extremely difficult to analyze the workload patterns of applications and computer resources used. These analyses are rarely performed or not performed at all. Deeper and more accurate analysis requires the use of artificial intelligence and machine learning tools as well as longer recording of all relevant data from hour to hour. The analysis of ongoing transactions, with large jumps in instance reservations, but also periods of low workload, is particularly complex. High load volatility and instance reservation is a key feature of transactional applications. Insufficiently accurate predictions based on incorrect workload patterns lead to over-reservation of instances or to over-consumption of on-demand instances. In an ideal prediction, we reserve exactly as many instances as the transactions require, in each time period of the prediction.

In this paper, we will try to give a rough estimate of the required instances for the changing market requirements of a small business in the e-commerce segment. Predictions are based on statistical properties and GARCH-modelled variance estimates in changes in the number of transactions on a daily basis. The number of required on-demand instances on a daily basis represents a deviation from the planned reserved instances on a monthly and annual basis.

### **2.1. Data analysis and volatility model**

Data on the use of cloud resources in public IaaS is often confidential and not publicly available. In addition, cloud providers do not publicly disclose information about the volume of transactions performed by e-commerce companies that use their services. Because of this, researchers often use alternative data series. For example, Wang et al. (2015) use Google cluster-usage traces. For this reason, as a substitute for e-commerce transactions performed in IaaS, in this paper we have chosen a historical data series on Ethereum cryptocurrency transactions every hour, in the period from 2017-07-01 12-PM to 2018-01-25 06-PM (5000 observations).

Figure 1. Bitcoin transaction and transaction first difference



Source: Author's calculation

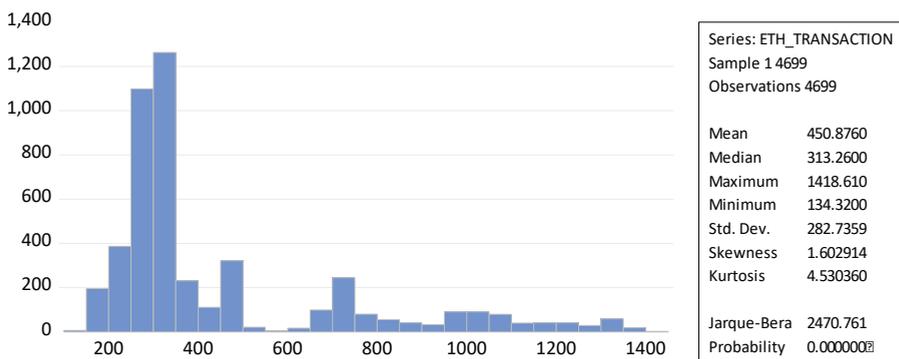
Like most financial time series, Ethereum transaction time series data is non-stationary while returns are stationary (Figure 1). In order to transform the time series of daily Ethereum transactions into stationary series, a time series of simple hourly returns on Ethereum transactions per hour is created. Simple return can be defined as:

$$R_t = \Delta ETH_t (ETH_t - ETH_{t-1}) / ETH_{t-1} = \Delta ETH_t / ETH_{t-1} \quad (1)$$

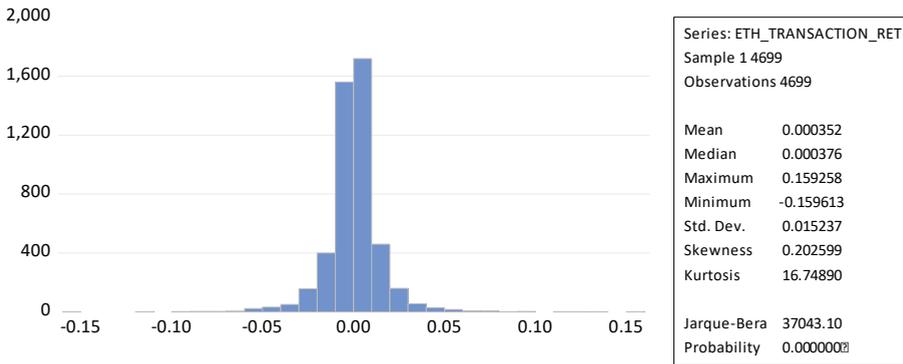
Where  $ETH_t$  is the value of the Ethereum transaction at time  $t$ , and  $\Delta ETH_{t-1}$  the Ethereum transaction difference.

Descriptive statistics on transactions and return on Ethereum transactions is summarized in Figure 2. The mean value of return is positive, which indicates that the number of transactions increased during the observed period. The statistical values of skewness, kurtosis, and Jarque-Bera test imply that the return series shows fat tailed properties and does not follow normal distribution.

Figure 2. Descriptive statistics and histogram



a) Ethereum transaction



b) Ethereum transaction returns

Source: Author's calculation

Checking the stationarity of the observed return time series includes testing for the presence of unit roots. In this paper, we use standard tests: Augmented Dickey-Fuller (ADF) test for unit roots (Dickey & Fuller, 1979), Phillips-Perron test (Phillips & Perron, 1988) and KPSS test (Kwiatkowski et al., 1992). Testing is done in the Eviews 10 software package. Based on the results shown in Table 3, we reject the null hypothesis of a single root at standard significance levels (1%, 5% and 10%, respectively) with a probability of p-values < 0.0001.

Table 3. Test of stationarity time series of bitcoin transaction return

Tests	ADF	PP	KPSS	
	t-Statistic (Probability)	Adj. t-Stat (Probability)	LM-Statistic	
	-53.80061 (0.0000)	-71.31998 (0.0001)	0.111334 -	
ARCH-LM test (Probability)	19.94688 (0.0000)			
t-critical	1%	-3.431560	-3.431559	0.739000
	5%	-2.861960	-2.861959	0.463000
	10%	-2.567036	-2.567036	0.347000

Notes: ADF - Augmented Dickey-Fuller test, PP - Phillips-Perron test, KPSS - Kwiatkowski-Phillips-Schmidt-Shin test

Source: Author's calculation

Figures 1 and 2.b clearly show the presence of spikes and clustering volatility. Mandelbrot describes clustering volatility as a phenomenon in which “**large changes** tend to be **followed by large changes**, of either **sign**, and **small changes** tend to be **followed by small changes**” (Mandelbrot, 1963). To test the presence of heteroskedasticity in the residual Ethereum transaction return series, we use the Lagrange Multiplier test for Autoregressive conditional heteroscedasticity (ARCH-LM) (Engle, 1982). The results

of ARCH-LM test provide strong evidence to reject the null hypothesis that the series is not heteroskedastic. This indicates the existence of ARCH effects in the return residues of Ethereum transactions. The generalized autoregressive conditional heteroskedasticity (GARCH) is a model used to estimate and model volatility in time series (Bollerslev, 1986). Mathematically, the GARCH(1,1) model is expressed as:

$$r_t = c + \epsilon_t, \epsilon_t \sim (0, \sigma_t^2) \quad (2)$$

where

$$\sigma_t^2 = \omega + \beta_1 \sigma_{t-1}^2 + \alpha_1 \epsilon_{t-1}^2 \quad (3)$$

and  $\alpha_1, \beta_1$  are the model parameters.

Threshold GARCH (TGARCH) integrates the impact of good and bad news on volatility. TGARCH (1,1,1) can be expressed as follows:

$$\sigma_t^2 = \omega + \beta_1 \sigma_{t-1}^2 + \alpha_1 \epsilon_{t-1}^2 + \lambda \epsilon_{t-1}^2 I_{t-1} \quad (4)$$

where

$$I_{t-1} = \begin{cases} 1, & \text{if } \epsilon_{t-1} \leq 0, & \text{bad news} \\ 0, & \text{if } \epsilon_{t-1} \geq 0, & \text{good news} \end{cases} \quad (5)$$

Table 3 shows the results of volatility modeling of Ethereum return using GARCH (1,1) with normal distribution, GARCH (1,1) with student t-distribution and TGARCH (1,1,1) with student t-distribution. Based on Log-Likelihood values and AIC criteria, the best model is TGARCH (1,1,1) with t-distribution (TGARCH (1,1,1)-t). All model coefficients ( $c, \omega, \alpha_1, \beta_1$ ) are statistically significant at the level of 1% (significant at 1% level) while the coefficient is significant at the level of 10%.

*Table 4. Models for volatility of the return of Ethereum hourly transaction*

Parameters	GARCH(1,1)	GARCH(1,1)-t	TGARCH(1,1,1)-t
$c$	0.000289 (0.0362)	0.000291 (0.0051)	0.000253 (0.0015)
$\omega$	3.45E-06 (0.0000)	1.53E-06 (0.0000)	1.55E-06 (0.0000)
$\alpha_1$	0.148487 (0.0000)	0.186598 (0.0000)	0.160038 (0.0000)
$\beta_1$	0.847200 (0.0000)	0.848685 (0.0000)	0.848955 (0.0000)
$\lambda$	-	-	0.045263 (0.0745)
Distribution parameter	-	DoF 3.553699 (0.0000)	DoF 3.563540 (0.0000)
Log-likelihood	14310.56	14751.28	<b>14753.10</b>
AIC	-6.089195	-6.276348	<b>-6.276698</b>

*Notes: AIC - Akaike info criterion*

*Source: Author's calculation*

## 2.2. Simulation results

In this paper, we test five pricing models: On-demand, No-Upfront reserved instance, Partial-Upfront reserved instance, All-Upfront reserved instance, and Max-Instance. In the first model, On-demand, the user rents all computer resources, i.e. instances, to fit their own needs, and there is no possibility to reserve instances. This model is common in situations where the user does not know the computer requirements in advance or with a time-varying workload (AWS, 2020).

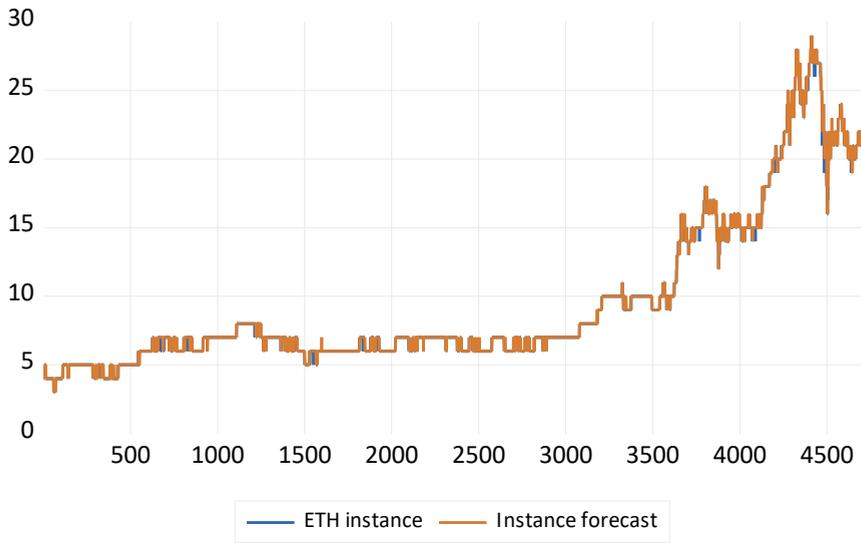
The next model group (No-Upfront reserved instance, Partial-Upfront reserved instance, All-Upfront reserved instance) is based on 1-Year Reserved Instance according to the Amazon EC2 model. The first model in this group, No-Upfront reserved instances, includes the costs of monthly reservation of instances based on the prediction of needs of the selected TGARCH model and the costs of using instances according to the on-demand model in contingencies. In the second model in this group, the Partial-Upfront reserved instances, the costs include partial instance reservation on an annual basis, monthly instance reservation costs, and the cost of on-demand usage of instances in contingencies. The cost simulation of the third model in this group, the All-Upfront reserved instance, includes the cost of a one-year reservation of predicted instances and the costs of using contingent on-demand instances.

The last model simulated, Max-Instance, is based on the unrealistic assumption that the maximum number of necessary instances is reserved for uninterrupted business throughout the year. In this model, the total cost is the cost of reserving instances on an annual basis and there is no cost of on-demand instance usage.

All costs are normalized to unit costs (first 300 pieces of data) assuming that one instance is required for that period. The volume of transactions, after the initial period, is reduced to a multiplier of the number of transactions in relation to the initial number of transactions. For example, if it is predicted that ten instances need to be reserved, this in our simulation means that ten times more instances need to be reserved than in the initial period. Also, the initial period is used to form the TGARCH model.

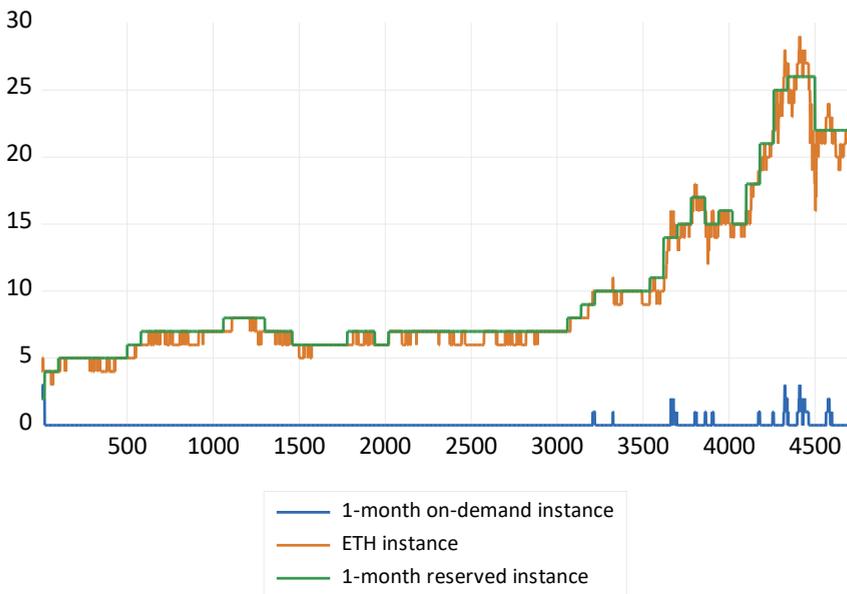
Figures 3-6 show graphical simulation results for pricing models. Table 5 shows the numerical results of cost calculation for all price models. Figure 3 shows the monthly prediction of the TGARCH volatility model in relation to the actual number of required instances for Ethereum transactions. Figure 3 clearly shows that the TGARCH model is acceptable as the volatility model. Based on the forecast of required instances for the next month, the Amazon EC2 pricing model is used to calculate reservation costs, as shown in Table 2. All figures show the actual overruns within Ethereum transactions that require the reservation of on-demand instances. The exception is the Max-Instance model where there is no on-demand usage.

Figure 3. 30-day forecast transaction vs. Ethereum transaction



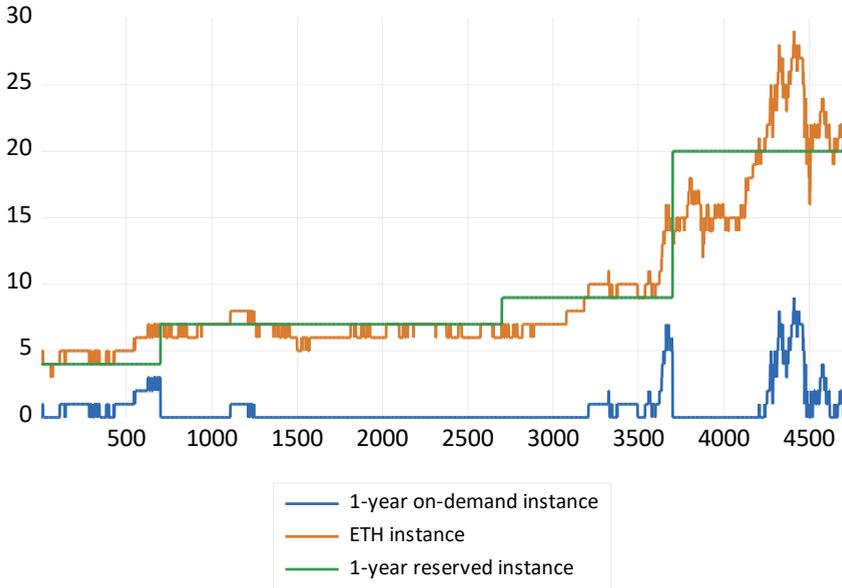
Source: Author's calculation

Figure 4. One-month reserved instance and on-demand instance vs real Ethereum instance – No Upfront and Partial Upfront Pricing options



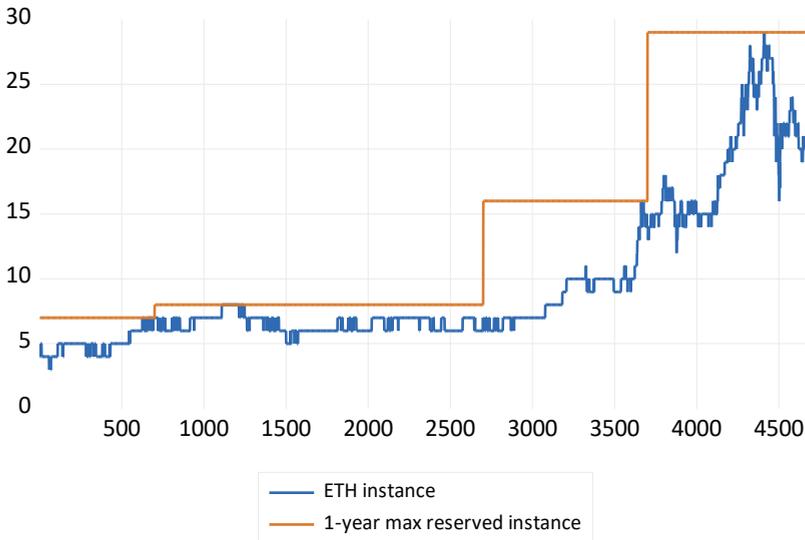
Source: Author's calculation

Figure 5. One-year reserved instance and on-demand instance vs real Ethereum instance – All Upfront Pricing option



Source: Author's calculation

Figure 6. One-year maximum reserved instance vs real Ethereum instance – Max instance Pricing option



Source: Author's calculation

The results of numerical analysis of the cost of computer resource renting, for all price models, are shown in Table 5. The Partial-Upfront model proves to be the best choice. However, all models with one-year instance reservation give approximately similar results. Our simulation does not take into account any discounts that providers give on on-demand instances with models with reserved instances already taken, so the advantage of instance reservation is even more obvious compared to the on-demand model. The Max-Instance model proves to be the worst model. In addition to extremely high costs, compared to models with reserved instances, Figure 6 clearly shows that a large number of reserved instances remain unused. This model is similar to the on-demand model in terms of costs, which shows that the overestimation of the required instances is equally expensive as when there is no assessment, but all instances are used as needed.

*Table 5. Total Cost of Ownership (TCO) of running cloud instances in IaaS clouds*

Pricing Option		TCO
On-demand		\$28,932.55
1-Year Reserved	No-Upfront	\$22,480.67
	Partial-Upfront	\$20,674.68
	All-Upfront	\$21,522.24
	Max-Instance	\$29,429.32

*Source: Author's calculation*

The results of the simulation suggest that it is best for small and medium-sized e-commerce companies to start with an on-demand model of computer resource use in the first year. After the first year, based on historical data on the volume of daily (or per hour) transactions, they can choose one of the models with a one-year reservation. The better the historical data, the more accurate the prediction of future cloud service costs. If the volume of business is such that the fluctuations in the number of transactions are quite large, the Partial-Upfront model is recommended. If the transaction number is relatively stable, the best choice is the All-Upfront model. Many case studies show that effectively combining the use of these two models with instance reservation leads to a significant cost reduction (Chaisiri et al., 2012; Wang et al., 2013; Huang, 2013; Clamp & Cartlidge, 2013; AWS Case studies, 2019). This research shows that it is possible to qualitatively estimate the future costs of using cloud computing among small and medium enterprises that are already engaged in e-commerce or intend to start e-commerce in the cloud.

## Conclusion

Cloud computing for small and medium enterprises (SMEs) promises a number of advantages over on-premise installations, including cost, operational and organizational efficiency. However, for SMEs in underdeveloped or middle-developed countries, the weaknesses of cloud implementation are: low quality of infrastructure, unresolved problems with legislation related to cloud data handling, dependence on cloud vendors, and loss of control over data. Total cost of ownership (TCO) is a widely used approach

that provides SMEs with a comprehensive overview of cost factors. The application of the TCO approach to cloud computing is important for quality assessment of indirect costs and costs that last over the life of the system, as well as for the overall value of the chosen cloud model. In this paper, we identify the key cost factors of the cloud infrastructure model as a service (IaaS). TCO supports SMEs and start-up companies that do not manage their own IT infrastructure.

This paper presents a methodology for analyzing and estimating the number of reserved instances using the GARCH class of volatility prediction models. The analysis focuses on the number of daily transactions of the hypothetical SME, simulating the Ethereum cryptocurrency transactions collected every hour. The research results suggest that the TGARCH model with student t-distribution can be used successfully when there is historical data on the number of daily transactions. In addition, the results suggest that in situations where there are large oscillations in the number of transactions, the best choice of instance reservations is according to the Partial-Upfront price model. In contrast, if the transaction number is relatively stable, the best choice is the All-Upfront model.

The presented cost analysis serves as a support for decision-making in the evaluation of cloud computing services and the selection of the appropriate pricing model. However, the TCO approach should be seen as part of a much broader overall IT cost management in SMEs as well as a starting method for estimating cloud computing service and costs for start-up companies.

## Reference

- AWS (2020). Amazon Web Services (AWS) - Cloud Computing Services <https://aws.amazon.com/> (1.9.2020).
- AWS Case studies (2019), <http://aws.amazon.com/solutions/case-studies/>. (15.06.2019).
- Bollerslev, R. (1986) Generalized Autoregressive Conditional Heteroskedasticity. *Journal of Econometrics*, 31, 307-327.
- Cegielski, C.G., Allison Jones-Farmer, L., Wu, Y., & Hazen, B.T. (2012). Adoption of cloud computing technologies in supply chains: an organizational information processing theory approach. *Int J Logistics Manage* 23(2), 184–211.
- Chaisiri, S., Lee, B.-S., & Niyato, D. (2012). Optimization of resource provisioning cost in cloud computing. *IEEE Trans. Services Comput.*, 5, 2, 164–177.
- Clamp, P., & Cartlidge, J. (2013). Pricing the cloud: an adaptive brokerage for cloud computing. *5th International Conference on Advances in System Simulation (SIMUL-2013)*, IARIA XPS Press, Venice, 113–121.
- Dickey, D.A., & Fuller, W.F. (1979). Distribution of the Estimators for Autoregressive Time Series with a Unit Root. *Journal of the American Statistical Association*, vol.74, 366, 427-431.
- Đorđević, M., Radović, O., & Bonić, Lj. (2018). Potentials for applying cloud technology in accounting. *Ekonomika*, vol.64, 23-32.
- Ellram, L.M., & Siferd, S.P. (1993). Purchasing: the cornerstone of the total cost of ownership concept. *Journal of Business Logistics*, 14(1), 163-184.

- Engle, R. (1982). Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. *Econometrica*, 50, 4, 987–1008.
- Free, R.C. (Ed.). (2010). 21st century economics: A reference handbook. Thousand Oaks, USA: SAGE Publications.
- Huang, J. (2013). Pricing Strategy for Cloud Computing Services. *PACIS 2013, Proceedings* 279.
- Krivokuća, M., Čočkaló, D., & Bakator, M. (2021). The potential of digital entrepreneurship in Serbia. *Anali Ekonomskog fakulteta u Subotici*, 57(45), 97-115. <https://doi.org/10.5937/AnEkSub2145097K>
- Kwiatkowski, D., Phillips, P.C.B., Schmidt, P., & Shin, Y. (1992). Testing the null hypothesis of stationarity against the alternative of a unit root. *Journal of Econometrics*, 54, 159-178.
- Makhlouf, R. (2020). Cloudy transaction costs: a dive into cloud computing economics. *Journal of Cloud Computing: Advances, Systems and Applications*, 9:1 <https://doi.org/10.1186/s13677-019-0149-4>
- Mandelbrot, B. (1963). The Variation of Certain Speculative Prices. *The Journal of Business*, 4, 36, 394-419.
- Martens, B., Walterbusch, M. & Teuteberg, F. (2012). Costing of Cloud Computing Services: A Total Cost of Ownership Approach. *45th Hawaii International Conference on System Sciences, Maui, HI*, 1563-1572.
- Mell P., & Grance, T. (2009). The NIST definition of cloud computing. *Natl Inst Stand Technol*, 53(6):50.
- Phillips, P., & Perron, P. (1988). Testing for a unit root in time series regression, *Biometrika*, 75, 2, 335–346.
- Ramchand, K., Chhetri, M.B., Kowalczyk, R. (2018). Towards a comprehensive cloud decision framework with financial viability assessment. In: *Proceedings of the 22nd Pacific Asia Conference on Information Systems (PACIS 2018)*, P(32).
- Rosati, P., Fowley, F., Pahl, C., Taibi, D., & Lynn, T., (2019). Right Scaling for Right Pricing: A Case Study on Total Cost of Ownership Measurement for Cloud Migration. *Cloud Computing and Services Science*, Springer International Publishing, 190–214.
- Rosati, P., Fox, G., Kenny, D., & Lynn, T. (2017). Quantifying the financial value of cloud investments: a systematic literature review. *IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, 194–201.
- Strebel, J., & Stage, A. (2010) An economic decision model for business software application deployment on hybrid cloud environments. *Multikonferenz Wirtschaftsinformatik 2010*:47.
- Vouk, M.A. (2008). Cloud computing - Issues, research and implementations. *ITI 2008-30th International Conference on Information Technology Interfaces*, 31-40.
- Wang, W., Liang, B., & Li, B. (2015). Optimal online multi-instance acquisition in IaaS clouds. *IEEE Transactions on Parallel and Distributed Systems*, 26,12, 3407-3419.

- Wang, W., Niu, D., Li, B., & Liang, B. (2013). Dynamic cloud resource reservation via cloud brokerage. *ICDCS'13 Proceedings of the 2013 IEEE 33rd International Conference on Distributed Computing Systems*, 400-409.
- Yigitbasioglu, O. (2014). Modelling the Intention to Adopt Cloud Computing Services: A Transaction Cost Theory Perspective. *Australasian Journal of Information Systems*, 18(3), 193–210.
- Zimmerman, D.K. (2014). Five cloud essentials for the boardroom: what banking and financial markets executives need to know about cloud computing. *J Payments Strategy Syst* 8(1), 84–93.

