Vol 8. No. 2 Issue 2 – August, 2015
African Journal of Computing & ICT

www.ajocict.net

A Financial Option Model for Pricing Cloud Computational Resources Based on Cloud Trace Characterization

D. Allenotor, D. Oyemade & A. Ojugo
Department of Mathematics & Computer Science
Federal University of Petroleum Resources
Effurun, Delta State, P.M.B. 1221, Nigeria
allenotor.david@fupre.edu.ng, ojugo.arnold@fupre.edu.ng, oyemade.david@fupre.edu.ng

ABSTRACT

Cloud computing has developed extensively for executing resource-intensive applications. As a result, many commercial and industrial services are now hosted on the cloud for computation using resources that would have been unaffordable if owned privately. The hosting, storage, and big data computation using cloud resources are becoming norm of the day and the pricing of the resources have become an important problem. Existing literature shows that a few economic models have been reported for pricing cloud resources using static approaches. In this paper, we address a novel application of financial option pricing theory to the management of distributed computing resources for pricing. First, we highlight the importance of finance models for the given problem and then we provide a justification for the fitness of option theory to price the distributed computing resources especially resources (memory, storage, software, and compute cycles) for the cloud. Second, we design and develop pricing model and generate pricing results for the usage of such resources. We use a large number of experiments to provide justification for our proposed pricing model. We compare the simulated system to real cloud trace data based on the spot price for the cloud resources.

Keywords: Cloud resources, financial options, price volatility, compute cycles, opting pricing, cloud trace.

African Journal of Computing & ICT Reference Format:
S. M. Tudunkaya (2015) Design of the Affine Ciphers of Rhotrices. Afr J. of Comp & ICTs. Vol 8, No. 2, Issue 2. Pp 83-92

1. INTRODUCTION/BACKGROUND

Many cloud resources such as CPU cycles, memory, network bandwidths, throughput, computing power, disks, processor, and various measurements and instrumentation tools exist as state artifacts. Their existence/availability are transient and their valuation can be described with the same theories that support financial option principles in commodity and asset management. Unlike tangible assets such as gold, silver, iron ore, crude oil, or other solid mineral resources, cloud resources are not easily storable. Hence, we characterize them as non-storable computational resources. Since they are non-storable, their existence can only be vetted in the financial option computational paradigm as compute cycles or compute seed.

Therefore, one of the means to value them is by the use of financial options. The pricing of cloud resources is a challenging and an important task because of the characteristic nature of cloud computational resources; (i) they are heterogeneous and numerous [1] (geographically dispersed ownership and time zones affects their availability), (ii) they exists as compute cycles with a high volatility of their availability, (iii) security of resources and policy requirement differs from one geo-political region to another, (iv) resource management policies are administered differently, (v) there is unreliability of resources and environments, and (vi) the resources are connected by heterogeneous, multi-level networks. These pose computation challenges for resource management. Among the characteristics of the cloud resources enumerated, (i) and (ii) accounts for a high level of volatility of the resources. In the existing literature, research efforts focus on the application of traditional methods such as Discounted Cash Flow (DCF) or Net Preset Value (NPV) [2] to value transient resources (CPU cycles, memory, network bandwidths, throughput, computing power, disks, processor, and various measurements and instrumentation tools). However, the valuation of non-storable compute cycles cannot be modeled exactly if the target model cannot manage heterogeneity of the resources as well as the presence of high volatility in their availability. In the absence of flexibility of pricing cloud resource and services we engage two strategies to develop a model to price them. First, we simulate cloud resources usage in order to justify our proposed pricing model using the CloudSim [3] toolkit. CloudSim is a framework for modeling and simulation of cloud computing infrastructures and services. In this part of the work, we integrate a financial option based pricing model with CloudSim framework and use it as a cloud simulation tool to price cloud compute resources. Secondly, we evaluate our proposed model using the data from real clouds trace data. The Analysis of usage of the cloud resources from simulation and real cloud trace data shows the feasibility of a financial option based model for pricing cloud resources.

Vol 8. No. 2 Issue 2 – August, 2015
African Journal of Computing & ICT

www.ajocict.net

Currently, the cost for using cloud resources especially for research, commercial, and industrial purposes is fixed. It means a flat rate is charged for executing jobs on the clouds. For example, supposed that a fixed cost of $2.00 is charged for 8GB RAM, it may be considered as too expensive if the 8GB RAM will not be completely used up for the purpose it was requested. A flexible charging means paying less than $2 to execute the job on the cloud. Despite the current static charging system, there is a large interest in big data for computing. As a result, cloud computing is experiencing a mushrooming of many service providers. Amazon, for example, introduced a Simple Storage Service S3 [4] system and the Elastic Compute Cloud (EC2) ([5], [6]) for users. Amazon's S3 provides data-intensive, low cost, and highly available data storage system. EC2 provides on-demand computing resource as a virtual machine. One of the drawbacks of these services is that the resource prices are static.

Other initiatives include AppNexus [7], GoGrid [8], Google App Engine [9], Microsoft Azure Services [10], and Joyent Accelerator [11]. Requirement for flexibility in resource usage is seen from the choices made available to users. Such choices include the decision to use the resources at a time in the present or at some time in the future. It is hard to make decision using NPV or DCF without losing the realistic value of the decision [2]. To price the cloud resources, we treat them as computational assets and we formulate a pricing model using the theories of financial option to compute option value and the best exercise time for resource usage.

1.1 Financial Options

A financial option (see, for example [12]) can be defined as the right to buy or to sell an underlying asset that is traded in an exchange for an agreed-on sum. The right to buy or sell an option expires if the right is not exercised on or before a specified period. If this right of exercise is lost, the option buyer will also loose the premium paid at the beginning of the contract. The exercise price which is also called the strike price (mentioned in an option contract) is the stated price at which the asset can be bought or sold at a date in the future. A call option on the other hand, grants the holder the right (but not obligation) to buy the underlying asset at the specified strike price. Two option styles are American option and European option. An American option can be exercised any time during the life of the option contract. On the other hand, a European option can only be exercised at expiry.

The rest of this paper is organized as follows. In Section 2 we review related work. Section 3 provides the model theory, description, and assumptions. Section 4 describes the model architecture and integration with CloudSim. Section 5 presents our experiments, results, and discussions. Section 6 ends the paper and provide directions for future work.

2. RELATED WORK

Earlier developments in grid computing which gave rise to cloud computing offered lead research efforts from security, distributed resources management and scheduling to grid market economy. Because many of the grids were production and research grids, services are made available free of charge or with minimal service charges. However, as the grid idea developed into cloud ideas and many commercial and industrial applications find computation cost saving benefits in cloud computing, charging (billing) became an important aspect of the current research efforts. A greedy approach was static charging for cloud resources use. Therefore, in existing literature, approaches for managing distributed resources applied resource scheduling principles.

These include the Tycoon [13] and the Condor system [14]. In [13] and [14], it was assumed that resource requirements can be estimated a priori by the users. Based on these user estimates, a computation cost was associated. The costing procedure was matchmaking function by matching jobs to resources based on resource requirements and resource availability. As reported by many studies of empirical proof ([16], [17], and [15]), the approach could be misleading because many users are not sincere while they estimate requirements for their jobs – they tend to supply inaccurate estimates of resources required and job runtime. As a result, relying on these erroneous job requirement estimates could lead to poor resource management. The estimates provided by these studies are poor because they allow users to continue to provide estimates even in cases where there are indications of strong incentives for faithful reporting. For instance, scheduling such as backfilling [17] schedules the first job in the queue that could be completed given the available resources. This would mean providing incentives to quote low runtime requirements. Similarly, jobs may be evicted from queue if the actual runtime is higher than the estimated runtime. This helps to ensure that users do not quote low resource and runtime estimates that are not realizable.

When estimates of resource requirements are poorly requested, they can significantly undermine the efficiency of the scheduling algorithms used. The consequences of allowing users to provide their computation requirements is that the realized costs may be considerably different from their estimates, causing considerable ex-post regret when costs exceed expectations. This is also not desirable. Solutions are needed to address inefficiencies that is caused poor estimates of resource requirements so that buyers can better estimate costs and schedulers can better assign jobs to resources and resource providers can make favorable capacity decisions or managements.

Vol 8. No. 2 Issue 2 – August, 2015
African Journal of Computing & ICT

www.ajocict.net

On market economy, Kang et al. in [18], Sulistio et al. in [19], and Tan and Gurd in [20] focus research efforts on resource allocation and resource scheduling with references. However, Mutz et al. [17] has some interesting research observations. They critically re-examined a batched queue-environment which consists of a simple form of batched-queue of jobs j_i, for $1,2,\cdots,n$ waiting to access computing resources; where j_i receives service before j_{i+1}. The resources granted is based on the owners' parameters or behavior which they used to model a payment function. Shrma et al. [26] valuated cloud resources on the basis of the age of technology of the components that constitute the cloud. However, many of the cloud resources such as compute cycles and compute seed do not heavily relay on age. Our work is novel because we use the real trace data to capture the realistic figure of our option values which is a complete replica of real life.

3. PRICING MODEL

Our model is developed using three approached; (a) two basic assumption, (b) Trinomial lattice approach, and (c) introduction of price variant factor into the overall model.

3.1 Assumptions

We made the following assumptions to aid the development of our model.

Assumption 1: We set some base prices for the cloud resources. These assumed prices are the prices that reflect the current real sale prices but discounted almost as close to 100%.

Assumption 2: Since the resources exists in non-storable (nonstable) states, we value them as real assets. This assumption qualifies them to fit into the general stream of investment included in the real option valuation approach. This assumption also justifies resources availability. Since the resources are nonstable, a high volatility (σ) affects the resources availability. This is responsible for a shorter use time of cloud resources compared with the life of option in financial valuation methods. A holder of the option to use the cloud resources has an obligation-free chance of exercising the right. The obligation-free status enables us to apply existing finance option valuation theory to model our pricing scheme.

3.2. Trinomial Lattice Approach to Option Pricing

Consider an asset whose price is initially S_0 and an option on the asset whose current price is f. Suppose the option has a lifetime of T. It can either move up with a probability p_u from S_0 to a new level S_0u with a payoff value of f_u or move down to p_d from S_0 to a new level, S_0d and with a payoff value of fd where $u > 1$ and $d < 1$. This lead to a one-step binomial. We have trinomial if it could also maintain a steady level (without either moving up or moving down), with a probability p_m. We define a job in the cloud as a service that need one or more of the resources from start to finish.

In a trinomial approach, we apply the trinomial-tree model [21] to price mainly American-style and European-style options on a single underlying asset. Options pricing under the Black-Scholes model [22] requires the solution of the partial differential equation and satisfied by the option price. In order to compute the option prices, we need to build the discrete time and state binomial model of the asset price and then apply discounted expectations [23]. Suppose S is the current asset price and that r is the riskless and continuously compounded interest rate, the risk-neutral Black-Scholes model of an asset price paying the continuous dividend yield of δ for each year [12] is given by:

$$dS = (r-\delta)S\delta t - \sigma S dz \qquad (1)$$

For convenience, let $x = lnS$, Equation (1) can be written as $dx = vdt + \sigma dz$, where $v = r - \delta - \frac{\sigma^2}{2}$. Let us consider a trinomial model of asset price in a small interval δt, we set the asset price changes by δx. Suppose this change remain the same or changes by δx, with likelihood of an up movement pu, chance of steady move (without a change) pm, and chance of a downward movement pd.

The drift (as a result of known reasons) and volatility (σ, because of unknown reasons) parameters of the asset price can be obtained in the simplified discrete process using $\delta x, pu, pm$, and pd. In a trinomial lattice the price step (with a choice) is given by $\delta x = \sigma\sqrt{3\delta t}$ and imposing the unitary sum of the likelihoods, we obtain a relationship between the parameters of the continuous time and trinomial (a discretization of the Geometric Brownian Motion (GBM)), that is,

$$E[\delta x] = pu(\delta x) + pm(0) + pd(-\delta x) = v\delta t \qquad (2)$$

where $E[\delta x]$ is the expectation as mentioned before.

From Equation (2),

$$E[\delta x^2] = pu(\delta x^2) + pm(0) + pd(-\delta x^2) = \sigma^2\delta t + v^2\delta t^2 \qquad ($$

We can present the unitary probability sum as $pu + pm + pd = 1$, where $pu, pm, and\ pd$are the probabilities of the price going up, down or remaining same respectively. We solve Equations (2), (3), and (4) to yield the transitional probabilities:

$$pu = \frac{1}{2}\left(\left(\frac{\sigma^2\Delta t + v^2\Delta t^2}{\Delta x^2}\right) + \frac{(v\Delta t)}{\Delta x}\right) \qquad (4)$$

$$pm = 1 - \left(\frac{\sigma^2\Delta t + v^2\Delta t^2}{\Delta x^2}\right) \qquad (5)$$

$$pd = \frac{1}{2}\left(\left(\frac{\sigma^2\Delta t + v^2\Delta t^2}{\Delta x^2}\right) - \frac{(v\Delta t)}{\Delta x}\right) \qquad (6)$$

Vol 8. No. 2 Issue 2 – August, 2015
African Journal of Computing & ICT

www.ajocict.net

We replicate the one-step trinomial process to form an n-step trinomial tree. For number of time steps (horizontal level) n = 4, the number of leaves (height) in such a tree is given by 2n + 1. We index a node by referencing a pair (i, j) where i points at the level (row index) and j shows the distance from the top (column index). Time t is referenced from the level index by i: $t = i\Delta t$. Node (i, j) is thus connected to node (i + 1, j) (upward move), to node (i + 1, j + 1) (steady move), and to node (i + 1, j + 2) (downward move). The option price and the asset price at node (i, j) are given by $C[i, j] = C_{i,j}$ and $S[i, j] = S_{i,j}$ respectively. The number of up and down moves required to reach (i, j) from (0, 0) estimates the asset price and is given by

$$S[i,j] = S[0,0](u^i d^j) \qquad (7)$$

At maturity, that is, when $T = n\Delta t$ for European style options or when $T \leq n\Delta t$ for American style options, the option values are determined by the pay off. Therefore, for a call option whenever the intent is to buy an asset at a previously determined strike price, the payoff is given as:

$$C_{n,j} = Max(0, S_{n,j} - K) \qquad (8)$$

and for an intent to sell, the payoff is computed using Equation (10).

$$C_{n,j} = Max(0, K - S_{n,j}) \qquad (9)$$

where in Equation (9) and Equation (10) the value K represents the strike price at maturity $T = n\Delta t$ for a European-style option, and the strike price before, or on maturity for an American-style option. To calculate option prices, we apply the discounted expectations under the risk neutral assumption.

For an American put option (for example), for i < n:

$$C_{i,j} = Max\left(e^{-r\Delta t}(puC_{i+1,j} + pmC_{i+1,j+1} + pdC_{i+1,j+2}), K - S_{i,j}\right) \qquad (10)$$

For a European call option (exercised on maturity only),

$$C_{i,j} = e^{-r\Delta t}(puC_{i+1,j} + pmC_{i+1,j+1} + pdC_{i+1,j+2}) \qquad (11)$$

While option price starts at $C_{0,0}$, we apply the expression for $C_{n,j}$ with Equations (7), and (8) or (9) to get the option price at every time step and node of the trinomial-tree. We now model grid resources based on the transient availability of the grid compute cycles, the availability of compute cycles, and the value of volatility of prices associated with the compute cycles. Given maturity date t, expectation of the risk-neutral value $(\hat{E})$, the future price F(t) of a contract on grid resources could be expressed as $F(t) = \hat{E}[S(t)]$ where $\hat{E}[S(t)] = S(0)e^{\int_0^t \mu(\tau)d\tau}$ (see for example [12]):

Consider a trinomial model of asset price in a small interval Δt, the asset price increases by Δx, remain the same or decreases by Δx, with probabilities; probability of up movement pu, probability of steady move (staying at the middle) pm, and probability of a downward movement pd.

To price the multi-resources system, we suppose a real option depends on some other variables such as the expected growth rate r_μ and the volatility respectively r_σ. Then if we let $\frac{dr_i}{r_i} = \mu r_i dt + \sigma r_i dz$ for any number of derivatives of r_i such as $(r_1, r_2, \cdots, r_n)$ with prices p $(p_1, p_2, \cdots, p_n)$ respectively, we have $dlnS = \frac{(dp_i)}{p_i} = \mu_i dt + \sigma_i dz$, where the variables r_i is the set of resources .

Applying the price variant factor pf for pricing options, we have:

$$dlnS = [r_i(t) - pf\,lnS]dt + [the\ stochastic\ term] \quad (13)$$

where the stochastic term is σdz. The value of its membership function (high for pf > 0) control the strength of the pf . So for a multi-asset problem, we have:

$$dlnS_i = [r_i(t) - pf\,lnS_i]dt + \sigma_i dz_i \forall i > 0 \qquad (14)$$

The value of $r_i(t)$ is determined such that $F(t) = \hat{E}[S(t)]$. This shows that the actual and expected value of S is equal to the future price, p. A user may need compute cycles (bandwidth) in the first quarter, second quarter, third quarter, and fourth quarter of the year from today and therefore decides to pay some amount, $p to hold a position for the expected increase. We show this using a 3−step trinomial for the spot price for bandwidth as $pT bit per second (bps) and for the projected first quarter, second quarter, third quarter, and fourth quarter of the year future prices $\$p_1, \$p_2, \$p_3$ and $\$p_4$ respectively. In this case, the two uncertainties are the quantity of bandwidth that will be available and the price per bit. However, we can get an estimate for the stochastic process for bandwidth prices by substituting some assumed values of pf and σ (for example, pf = 10%, σ = 20%) in Equation (13). Suppose $V_{l,j}$ represents the option values at l for $l = 0,1,2,\cdots,n-1$ level and j node for $j = 1,2,\cdots,(2l+1)$ (for a trinomial lattice only); that is, $V_{1,1}$ represents the option value at level 1 and at pu.

3.3. Price Variant Factor

An important functionality of our model is the price variance factor (pf). We define pf as $0 \leq pf \leq 1$. Its value depends on changes in technological developments such as new and faster algorithms, faster and cheaper processors, and changes in access rights and policies. The certainty in predicting the effects caused by these is hard using crisp schemes. As a result, we capture the resultant changes using fuzzy logic and treat pf as a fuzzy number. For a use time of (t_{ut}), we express pf as a fuzzy membership function that is, $\mu(pf)$. For example, the cloud resources may become under used if users find better and faster ways to solve their computing problems.

Vol 8. No. 2 Issue 2 – August, 2015
African Journal of Computing & ICT

www.ajocict.net

Therefore, to increase the cloud resources usage with more capacity for computations under same technology, we set the value of pf(ut) to 0.1 and with new technology, the $pf = 1.0$. we can then adjust the price in the use of cloud resources by $(pf(ut))^{-1}$ while providing QoS set at the Service Level Agreement (SLA) of the contract.

3.4. Fuzzy Logic Framework

To fuzzify the utility of the cloud, we express the quality of the resources availability as a function of the time when resources are needed and the time the resources become available for use. That is the resources (r_i), $r_i = f(tut, tn)$, where tn is the life of the contract and is given as $0 \leq tn \leq 1$, and tut is the actual utilization time. A best scenario is when $tn = tut$, i.e., when the resources are available when $r_i$is needed or $tn = 0$ (no wait time). If $tn = 0$, r_i use is "now" otherwise, $tn = 1$ and usage is in the future until the end of the contract period (say 6 months). Users often request and use r_i for computation and expects a best scenario where service provided meets expectations or when $tut - tn \xrightarrow{t} 0$ for a high QoS. In this instance, it is hard to guarantee provision of the r_i on-demand and satisfy the users' QoS without additional r_i to satisfy the conditions named in the SLAs document. To capture the fuzziness of the parameters tn, tut, and QoS, we express them in terms of their fuzzy membership functions.

That is, $\mu(tn), \mu(tut)$, and μ(QoS) respectively. If T is a fuzzy set, the membership function is defined (see for example [24]) as $T = (t, \mu(t)), \mu T(t) \in [0,1]$ To price the cloud resources, we consider only its heavily utilized resources. The RAM and CPU cycles are most requested resources provided in the Cerit Scientific Cloud [25]. The generated workload trace shows 91% normalized memory availability in Cerit Scientific Cloud. Therefore, we use memory availability index set at 0.91 in our simulation. This index, idx, can be expressed as a fuzzy membership function, $\mu_{idx}(p_{idx})$. Where p_n is the calibrated set of prices $(p_1, p_2, \cdots, p_n)$ where $p_{idx} \in \{p_1, p_2, \cdots, p_n\}$. We provide the fuzzy membership function of the range of prices in Equation (14).

$$\mu_{idx}(p_n) = \begin{cases} 1, for\ x = p_{idx}\ for\ 100\%\ availability \\ \dfrac{x - p_1}{p_{idx} - p_1}, & for\ p_1 \leq x \leq p_{idx} \\ \dfrac{p_n - x}{p_n - p_{idx}}, & for\ p_{idx} \leq x \leq p_n \\ 0, & otherwise, that\ is, if\ x \in [p_1, p_n] \end{cases} \quad (15)$$

4. MODEL ARCHITECTURE AND INTEGRATION WITH CLOUDSIM

In our model architecture, we normalize base prices for the grid resources using SLA and QoS as constraints for individual (local) grids. We also consider economic and market behaviors for resources conflict in the grid. For a detailed discussion on the model architecture see [1].

4.1. The CloudSim Simulation Architecture

Figure 1 shows the three main layers of the CloudSim architecture – the top user layer has two other sub layers (a) the simulation specification layer and (b) the scheduling policy layer. The second (middle) layer from the top is the CloudSim core layer.

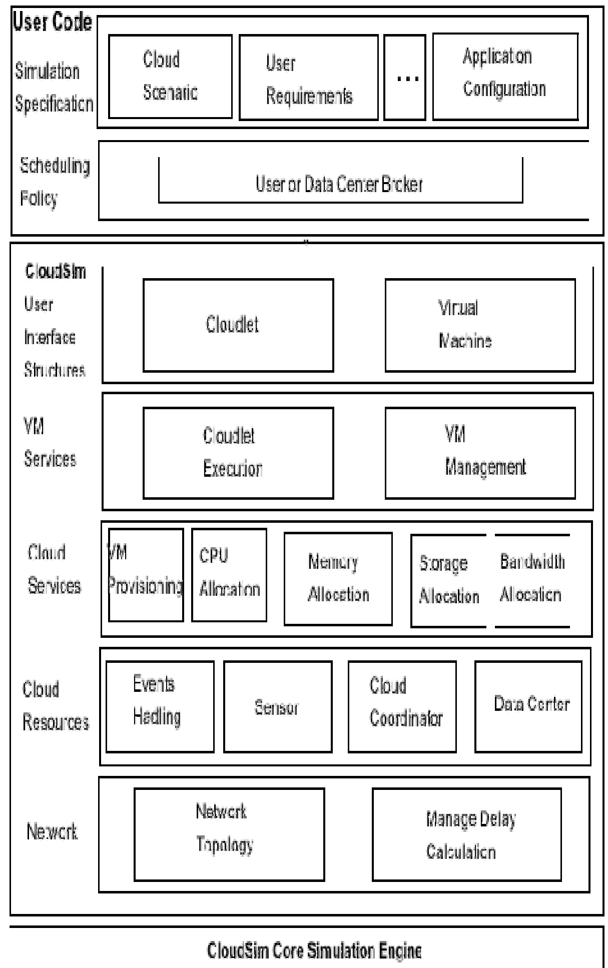

Figure 1.0: Layered CloudSim Architecture [3]

It houses the (i) user interface structures, (ii) the virtual machine (VM) services, (iii) the cloud services, (iv) the cloud resources, and (v) the network architecture. The third core layer is the CloudSim core simulation engine. The CloudSim simulation specification layer provides support for modeling and simulation of virtualized Cloud-based data center environments. This include the dedicated management interfaces for VMs, memory, storage, and bandwidth. Other fundamental functions handled by the CloudSim simulation specification layer are provisioning of hosts to VMs, managing application execution, and monitoring dynamic system state, are handled by this layer.

Vol 8. No. 2 Issue 2 – August, 2015
African Journal of Computing & ICT

A Cloud provider must implement strategies at this layer if the interest is to monitor the different policies in allocating its hosts to VMs (VM provisioning). Such implementation requires a great deal of programming that extends the core VM provisioning functionality. This layer is also associated with executing applications from the provider's defined QoS levels. In the current study, we integrate the top layer of our pricing model (price and usage optimization level) onto the top layer of CloudSim. The VM manages events and components interaction in the CloudSim. The second layer consists of the infrastructure components such as network and resource hardware. This layer also enables the design and integration of user interfaces. The third and fourth layers are responsible for the simulation and modeling of computational grid entities. Simulation of the cloud resource broker takes place in the CloudSim layer. The top layer consists of the cloud scenario, user requirements, I/O interface, and application configuration.

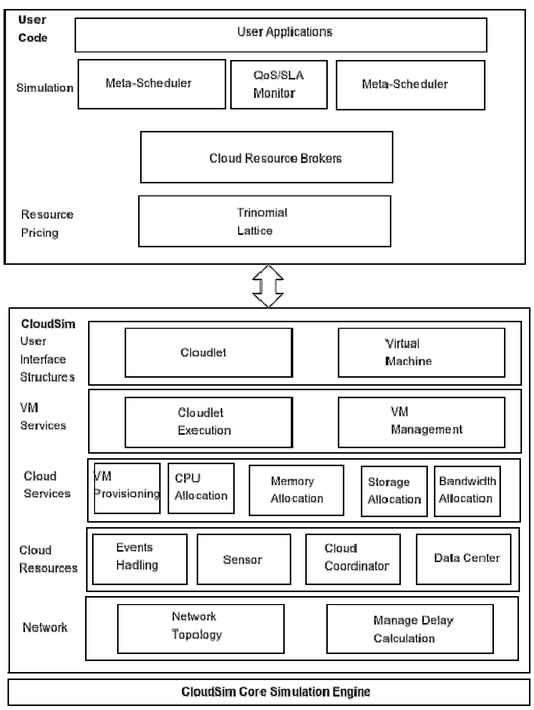

Figure 2.0: Integrated Pricing Architecture

Figure 2.0 shows our adapted integrated architectural model. The cloud resource pricing was carried out at the user code level of the CloudSim architecture. The QoS/SLA monitoring was achieved using a fuzzy classification approach.

5. EXPERIMENTS, RESULTS, AND DISCUSSIONS

We setup the base prices for cloud resources based on real and current market value. For example, if a 10GB of RAM costs about \$20. For a minimal 2 years for the 100% return on investment for a cloud owner, we charge \$0.50 $\times 10^{-6}$ per day per MB. Similarly, suppose it costs \$60.00 for 1TB hard disk then we fix a base price of \$3.43 $\times 10^{-8}$ per day per MB as a charge for the cloud storage. Similarly, we set a base price of \$3.43 $\times 10^{-6}$ per day per MHz of CPU cycles for a 1.00 GHz processor. These base prices that we choose are as low as possible because of repeated use of the cloud resources by many users. We use the data sets generated from Torque traces [26] at the national Center CERIT-SC (CERIT Scientific Cloud). The trace data were collected from January to April 2015. These data sets contain over 102,657 job descriptions. The jobs are divided per-user, with specified batches and their mutual dependencies. The shared memory machines run 8 clusters and has approximately 4,000 CPUs.

The cloud resources trace that we collected from the CERIT Scientific Cloud, include number of processors, memory, CPU time, run time, and wait time. First, we analyze these traces. To price the usage, we run the trinomial lattice using the following model parameters: For example, for a one-step trinomial tree we use strike price (K = \$20.00, \$22.50, \$25.50, \$27.50, \$30.00), resources price (S = \$0.80), expiration time (T = 90 days or 0.25 years), interest rate (r = 0.06), volatility (σ = 0.3), and the number of time steps $(Nj = 2N + 1)$. We extend our study by varying the volatility σ in steps of 0.1, 0.2, …, 0.7 starting from 0.2 and N = 0, 1, 2, 4, 8, 16, 32, 64, 128, 256. For a 3-month contract, for example, N = 3 would mean a 2 month step size and N = 6 would mean a 1 week step size.

Experiment 1: For the European call, we used two test values of strike price K = \$20.00 and \$22.50 to gauge the behavior of the option in the money. For the first value of K, with the increase of the time step to 128 the option value converged to \$7.34. Figure 3.0 shows that there was a jump in the value of the option after the first step, this could be due to the effect of the volatility became more pronounced in the option price at this period. For the second value of the strike price. In Table 1.0 we show that the option value converged to \$5.33, after 64 time steps. There was also a jump in the option value after the first time step. At the later time steps, the option value converged smoothly to a particular value. At the money, we observed the option value converged to \$3.62 after time step 128, there was a downward slide in the value of the option after time step 1, from \$3.78 to \$3.38. This could be due to the effect of the volatility being more pronounced at this period. We used these two values of the strike price K = \$27.50 and \$30.00 to gauge the behavior of the option out-of-the-money. In Table 1.0, we observed that for the first value of K, the option price converged to \$2.29 after time step 128. There was no observable jump in the option value for this strike price.

www.ajocict.net

For the second value of K, the option value converged to \$1.32 after time step 32. It is observable that the option converged in an earlier time steps compared to other strike prices. This can be attributed to the option value being deep out-of-the-money at this period. Table 1.0 shows the time steps and option values for European call using $\sigma = 0.2$.

Table 1.0: Time steps and Option Values for European call $\sigma = 0.2$

Time Steps (N)	0	1	2	4	8	16	32	64	128	256
f@K=\$20.00	5.00	7.26	7.37	7.32	7.34	7.33	7.34	7.34	7.34	7.34
f@K=\$22.50	2.50	5.46	5.37	5.38	5.35	5.32	5.34	5.33	5.33	5.33
f@K=\$25.00	0.00	3.76	3.38	3.49	3.55	3.59	3.60	3.61	3.62	3.62
f@K=\$27.50	0.00	2.06	2.34	2.33	2.29	2.25	2.29	2.28	2.29	2.29
f@K=\$30.00	0.00	0.36	1.31	1.17	1.33	1.33	1.35	1.35	1.35	1.35

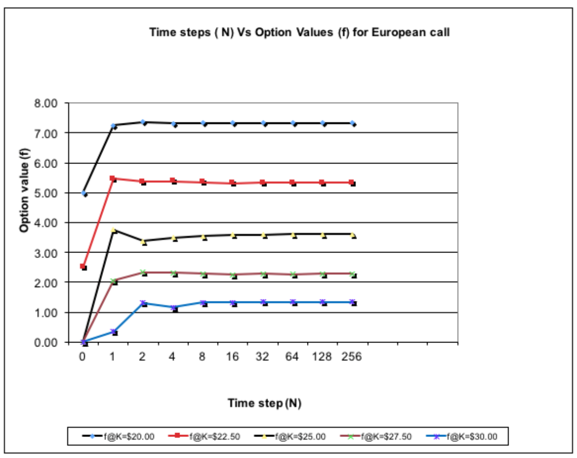

Figure 3.0: Computed Option Values for European Call.

Table 2.0: Strike price Vs Option values for European call

Strike Price K	f@ N=0	f@ N=8	f@ N=64
20.00	5.00	7.34	7.34
22.50	2.50	5.35	5.33
25.00	0.00	3.55	3.61
27.50	0.00	2.25	2.28
30.00	0.00	1.33	1.35

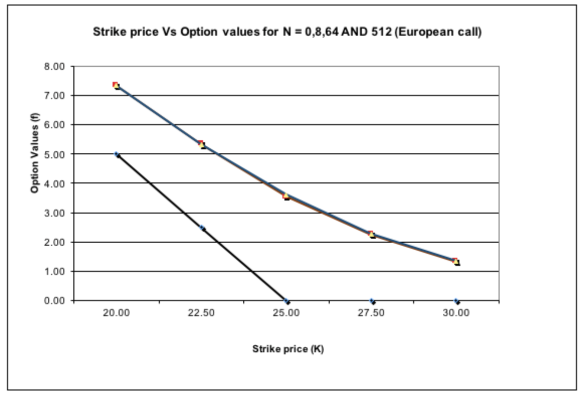

Figure 4.0: Strike Price and Option Values for European Call

Simulation 2: American put

For the American call, we used two strike prices K =\$27.5 and 30.00 to gauge the behavior of the put option when it is in-the-money, it was observed in Table 3.0 and in Figure 5 that the value of the option was constant at \$2.50 for the first strike price and \$5.00 for the second strike price. This could be attributed to the expected payoff being greater than the computed payoff at all time steps. It was also discovered that the option value converged earlier than the corresponding European call.

This substantiates the early exercise of American options at-the-money. Table 3.0 shows the option value converged at time step 15 with \$1.10. There was a no remarkable jump in the option value at the various time steps. The spot price of the option was nil. At this price, for example, we interpret this cost to mean that there is a benefit/leverage for the user for using cloud resources. Therefore, the cost of using cloud resources could be priced at any value as low as \$1.10 for the favor of the user and the provider without any losses.

For the out-of-the-money, we used 2 strike prices K = \$20.00 and \$22.50 to estimate the behavior of this option. Table 3.0 shows that using the first strike price the option value converged at time step 15 at \$0.09. There was no unusual behavior observed, except for the early convergence of the option price compared to the European call. The option value converged at time step 15 at 0.38. Figure 5.0 shows the results obtained gives credence to the early exercise of American put options. Table 3.0 shows the simulated number of time step and option values for the American put option using $\sigma = 0.2$.

www.ajocict.net

Table 3.0: Time steps and Option Values for American Put $\sigma = 0.2$

Time Steps (N)	0	1	2	3	5	7	9	12	15	17	19
f@K=$20.00	0.00	0.00	0.10	0.09	0.09	0.08	0.09	0.08	0.09	0.09	0.09
f@K=$22.50	0.00	0.42	0.33	0.34	0.34	0.35	0.36	0.37	0.37	0.37	0.37
f@K=$25.00	0.00	0.94	0.99	1.12	1.07	1.09	1.11	1.09	1.10	1.10	1.10
f@K=$27.50	2.50	2.50	2.50	2.50	2.52	2.52	2.55	2.56	2.59	2.59	2.59
f@K=$30.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00

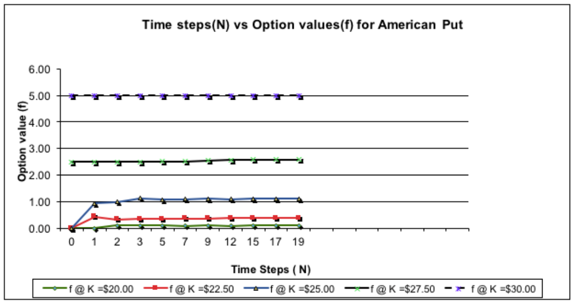

Figure 5.0: Computed Option Values for American Put at Various Time Steps.

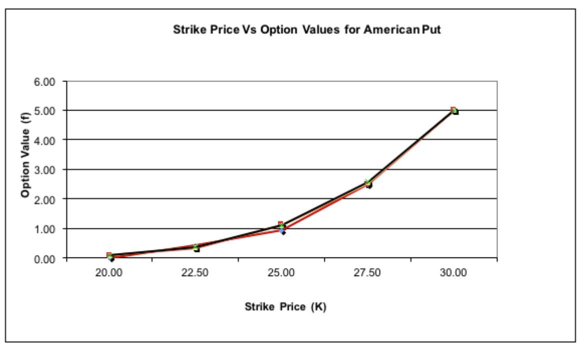

Figure 6.0: Computed Option Values for American Put at Differing Strike Price

Table 4.0: Strike price Vs Option values for American Put

Strike Price K	f@ N=1	f@ N=3	f@ N=9	f@ N=19
20.00	0.00	0.09	0.09	0.09
22.50	0.42	0.34	0.36	0.37
25.00	0.94	1.12	1.11	1.10
27.50	2.50	2.50	2.55	2.59
30.00	5.00	5.00	5.00	5.00

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have studied, analyzed, and made comparison for cloud resources utilizing using traces from CERIT Scientific Cloud. Our results show that it is possible to offer cloud compute resources to the user at a high value at one time and unable to support the same application at other times. In other words, resources availability varies while a measure of their certainty is hard to guarantee. The computed option value for cloud resources usage and to select the best point of exercise of the option to utilize any of the grid resources. This helps the user as well as the grid resources provider to optimize resources for profitability; in other words, we achieve an equilibrium condition; (ii) our study also incorporate a price varying function pf which controls the price of the resources and ensure the grid users gets the maximum at best prices and the resources provider also make reasonable revenue at the current base price settings. At the same time cloud operators do not unduly over-commit cloud resources whether the system is in-the-money or out-of-the-money conditions from the user perspective.
Future work will focus on the larger problem of pricing cloud resources for applications that use diverse resources across various clouds simultaneously. This will have to deal with a more complex, computationally intensive, and a multidimensional option pricing problem. This would need a more complex optimization of the solution space of the cloud resources usage as well as finding out the best node (time) to exercise the option (utilize the resources).

Acknowledgment
We are grateful to the CERIT Scientific Cloud for graciously proving us with the CERIT-SC workload log.

References

[1] D. Allenotor and R. Thulasiram, “Grid resources pricing: A novel financial Option-based Quality of Service-profit Quasi-Static Equilibrium model,” in The 9th IEEE/ACM International Conference on Grid Computing (Grid 2008). September 29-October 01, Tsukuba, Japan., September 2008.
[2] D. M. Chance and P. P. Peterson, Real Options and Investment Valuation. The Research Foundation of AMIR, First Edition, 2002.
[3] Calheiros RN, Ranjan R, Beloglazov A, De Rose CA, Buyya R (2011). “CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms.” Software: Practice and Experience 41 (1): 23–50.
[4] M. Palankar, A. Onibokun, A. Iamnitchi, and M. Ripeanu, “Amazon S3 for Sciece grids: A Viable Solution?” in Poster: 4th USENIX Symposium on Networked Systems Design and Implementation (NSDI’07), April 2007.
[5] Amazon. (2016, June) Amazon Elastic Compute Cloud (Amazon EC2). [Online]. Available: http://aws.amazon.com/ec2/
[6] A. Caracas and J. Altmann, “A Pricing Information Service for Grid Computing,” in MGC ’07: Proceedings of the 5th international workshop on Middleware for grid computing. New York, NY, USA: ACM, 2007, pp. 1–6.
[7] AppNexus. (2016, July) Appnexus. [Online]. Available: http://www.appneus.com/
[8] GoGrid. (2016, March) Cloud hosting: Instant Windows and Linux Cloud Servers. [Online]. Available: http://www.gogrid.com/
[9] Google. (2016, July) App engine. [Online]. Available: http://code.google.com/intl/de-DE/appengine/
[10] M. Corporation. (2016, July) Azure Services Platform. [Online]. Available: http://code.google.com/intl/de-DE/appengine/
[11] Joyent. (2016, June) Accelerator. [Online]. Available: http://joyent.com/accelerator/
[12] J. Hull, Options, Futures, and Other Derivatives. Prentice Hall, Sixth Edition, 2006.
[13] K. Lai, L. Rasmusson, E. Adar, L. Zhang, and B. A. Huberman, “Tycoon: An Implementation of a Distributed, Market-based Resource Allocation System,” Multiagent Grid Syst., vol. 1, no. 3, pp. 169–182, 2005.
[14] D. Thain, T. Tannenbaum, and M. Livny, “Distributed Computing in Practice: The Condor Experience,” Concurrency and Computation: Practice and Experience, pp. 1–36, 2004.

[15] A. W. Mu'alem and D. G. Feitelson, "Utilization, Predictability, Workloads, and User Runtime Estimates in Scheduling the IBM SP2 with Backfilling," IEEE Trans. Parallel Distrib. Syst., vol. 12, no. 6, pp. 529–543, 2001.

[16] M. Feldman, K. Lai, and L. Zhang, "A Price-anticipating Resource Allocation Mechanism for Distributed Shared Clusters," in EC '05: Proceedings of the 6th ACM conference on Electronic commerce, Vancouver, BC, Canada. New York, NY, USA: ACM, 2005, pp. 127–136.

[17] W. Kang, H. H. Huang, and A. Grimshaw, "A highly Available job Execution Service in Computational Service Market," in The 8th IEEE/ACM International Conference on Grid Computing. September 1921, Austin, Texas, USA., September 2007.

[18] A. Sulistio, W. Schiffmann, and R. Buyya, "Using Revenue Management to Determine Pricing of Revervations," in Proc. 3rd IntI Conference on e-Science and Grid Computing (eScience 2007) Bangalore, India, December 10-13, 2007, 2007.

[19] Z. Tan and J. R. Gurd, "Market-based Grid Resource Allocation Using a Stable Continuous Double Auction," in The 8th IEEE/ACM International Conference on Grid Computing. September 1921, Austin, Texas, USA., September 2007.

[20] P. P. Boyle, "Option Valuing Using a Three Jump Process," International Options Journal, vol. 3, no. 2, 1986.

[21] F. Black and Scholes, "The Pricing of Options and Corporate Liabilities," Journal of Political Economy, vol. 81, no. 3, 1973.

[22] P. Wilmott, Derivatives: The Theory and Practice of Financial Engineering. John Wiley and Son Ltd, 1998.

[23] G. Bojadziew and M. Bojadziew, Fuzzy Logic for Bus., Finance, and Management Modeling. World Scientific Press, 2nd Ed., 1997.

[24] Cerit Scientific Cloud (CERIT-SC) (2016, June) [Online]. Available: http://www.cerit-sc.cz/en/

[25] Bhanu Sharma, Ruppa K. Thulasiram, Parimala Thulasiraman, Saurabh K. Garg, and Rajkumar Buyya, Pricing Cloud Compute Commodities: A Novel Financial Economic Model. in CCGRID '12 Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012) Pages 451-457