Digital Innovations & Contemporary Research in SCIENCE, ENGINEERING & TECHNOLOGY
Vol. 10. No. 2, June 2022

Article Citation Format
Allenotor, D. (2022):
Simulation and Evaluation of a Financial Option Based Model for Cloud Resources Management
Journal of Digital Innovations & Contemporary Research in Science, Engineering & Technology. Vol. 10 No. 2. Pp 109-126
DOI: dx.doi.org/10.22624/AIMS/DIGITAL/V10N2P8

Article Progress Time Stamps

Article Type: Research Article
Manuscript Received: 12th February, 2022
Review Type: Blind
Final Acceptance:: 19th May, 2022
Published: 22nd June, 2022

Simulation and Evaluation of a Financial Option Based Model for Cloud Resources Management

Allenotor David
Department of Computer Science
Federal University of Petroleum Resources Effurun, Nigeria
E-mails: allenotor.david@fupre.edu.ng
Phone: +234-8100534069

ABSTRACT

Cloud computing, where the use of resources is seen as a service (Resources-as-a-Service (RaaS)) has developed extensively for executing computationally resource-intensive applications. As a result, commercial services of such resources are becoming norm of the day and pricing them have become an important problem in finance. Only a few economic models have been reported for pricing cloud resources. In this paper, a novel application of financial option pricing theory to the management of distributed computing resources especially for pricing is addressed. First, the importance of finance models for the given problem is highlighted following an explain on how option theory fits well to price the distributed computing cloud resources. Various cloud resources such as memory, storage, software, and compute cycles are seen as individual commodities and pricing of the resources is done in isolation and in combination of various resources. Second, `we design and develop pricing model and generate pricing results for usage of such commodities for various resources. In the absence of cloud resource pricing benchmarks/standards, firstly, cloud resources usage is simulated in order to justify the pricing model using CloudSim toolkit. In this part of the work, the integration of a financial option-based pricing model with CloudSim framework is implemented using a cloud simulation tool to price cloud compute resources. Secondly, the model is evaluated using cloud metadata obtained from Amazon Web services. The analysis of cloud resources utilization from simulation and real cloud trace data shows the feasibility of a financial option-based model for pricing cloud resources. With a large number of experiments carried out, a justification of the pricing model is obtained by comparing a simulated system to real cloud trace data based on the spot price for the cloud resources.

Keywords: Financial Option, Strike Price, CloudSim, Quality of Service, Compute Commodities

1. INTRODUCTION/BACKGROUND

Cloud computing [1] is a paradigm-shift from the conventional computation methods. It presents an on-demand availability of computer system resources, especially data storage (cloud storage) and computing power, without direct active management by the user for a fee [2]. Large clouds often have functions distributed over multiple locations, each location being a data center. Cloud computing relies on sharing of resources to achieve coherence and typically using a "pay-as-you-go" model which can help in reducing capital expenses but may also lead to unexpected operating expenses for unaware users. Majorly, there are over 5 areas of cloud application viz storage, Marketing, Education, Government, Healthcare. For the purpose of this study, attention is focused on educational clouds. These includes Google Cloud Platform, Microsoft Azure, Salesforce, Amazon web services [3], IBM Cloud, Digital Ocean, SAP, VMware, and google application engine. This paper also introduced the concept of Committed On-Demand Users (CODU) which will help to check the excesses of unexpected operating expenses for unaware users in the cloud. Cloud resources such as CPU cycles, memory, network bandwidths, throughput, computing power, disks, processor, and various measurements and instrumentation tools are non-storable compute commodities.

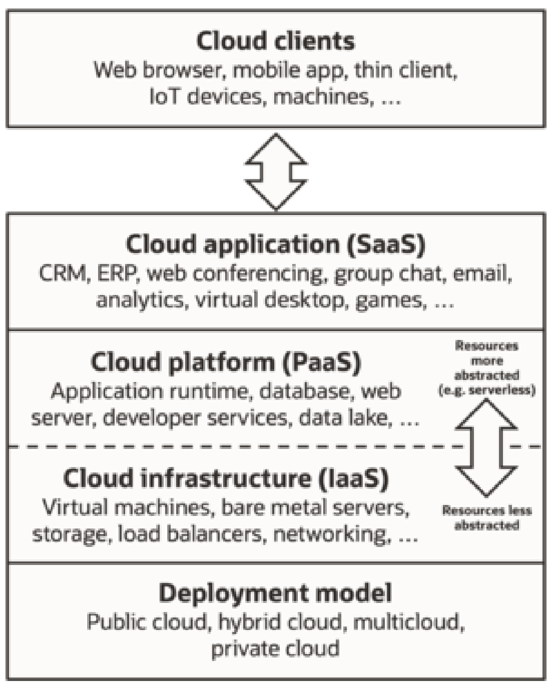

Figure 1: A Stack of Cloud Computing Service Model [37].

Pricing these compute commodities is challenging because of the specific characteristics of the cloud resources [4]: heterogeneity of resources (geographically dispersed ownership and time zones affects availability of resources) and volatility of resources since they exist as compute cycles. These two characteristics (among others) accounts for a high level of flexibilities (uncertainty and fluctuation) of resources. The trace data analysis in [5] shows that in the presence of such flexibilities, guaranteeing resource availability and profitability to users and cloud services providers respectively is hard to determine using traditional methods such as Discounted Cash Flow (DCF) or Net Preset Value (NPV)

[6] which were attempted in previous research efforts. Figure 1 shows a stacked computing service model for cloud where resources are mirrored to the user as a service, i.e., as a Resources-as-a-Service (RaaS). Research in cloud services is important finance problem because of the need to reduce capital infrastructure expenses in network and data computing centers. Similarly, since commercial services of such resources offered in the cloud are becoming norm of the day, it becomes necessary to focus research attention on pricing cloud resources. However, only a few economic models have been reported for pricing the resources. A novel application of financial option pricing theory to the management of distributed computing resources especially for pricing resources in the cloud is further addressed.

Currently, the amount charged for the usage of cloud resources is only arbitrage and based on a case-by-case. However, as a result of the large interest in cloud for public computing, cloud computing is experiencing a mushrooming of many service providers. Amazon, for example, introduced a Simple Storage Service S3 [7] system and the Elastic Compute Cloud (EC2) ([8], [9]) for cloud 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 not flexible, they are rather static. For example, Amazon S3 charges [3] statically $0.023 per GB for the first 50TB and increment this charge later wards. Some other initiatives include AppNexus [10], GoCloud [11], Google App Engine [12], Microsoft Azure Services [13], and Joyent Accelerator [14]. Requirement for flexibility in cloud resource usage is seen from the choices made available to users. Such choices include the decision to use the cloud 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 [6]. To price the cloud resources, the cloud resources are treated as cloud compute commodities (ccc) following the application of financial option theory in order to determine best exercise time for the use of the resources.

A financial option is defined (see, for example [15]) 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 may expire if the right is not exercised on or before a specific period and the option buyer loses the premium paid at the beginning of the contract. The exercise price (strike price) mentioned in an option contract is the stated price at which the asset can be bought or sold at a future date. A call option grants the holder the right (but not obligation) to buy the underlying asset at the specified strike price. Yeo and Buyya [16] provided a pricing function to improve utility for the producer on the basis that only jobs that have higher budgets will be accepted in the cluster workload. The pricing function also supports four essential requirements for pricing of utility-driven cluster resources: flexibility, fairness, dynamism, and adaptivity. In this article, the assumption that users will provide the exact computational requirements (faithfully) is only a hypothetical case. However, it is hard to capture users' honest computational requirements even with a promise of incentives [17]. In other previous work, the measure of a flexibility for the use of cloud resources is characterized by user opportunities from the decisions to utilize the resources. Such flexibilities can be accurately captured using real options so that asset prices could be computed using financial options pricing techniques.

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

2. RELATED WORKS

A primary area of focus in cloud economics is related to resource allocation and resource management. Several research studies in the field of resources allocation and management apply market mechanisms to allocate scarce compute resources to job requesting services. For example, Feldman et al. [18] formulate a resource allocation game and study the efficiency and fairness of the Nash equilibria (see for example [19]) that results while Wellman et al. [20] propose an auction mechanism to allocate distributed resources to users. Other related research efforts on market-based resource allocation includes those presented in [21], [22], [23], and [24]. Market-based resource allocation has also been implemented in the Tycoon system [25].

A. Resource Management: Scheduling and Reservations

To guarantee resource availability in the cloud, robust and efficient scheduling algorithm is required. A number of scheduling approaches have been proposed in the literature including the Tycoon [26] and Condor system [27]. Most current systems [17] assume that resources requirements can be estimated a priori by the users. The scheduling systems performs a matchmaking function by matching jobs to resources based on resource requirements and available resources. However, a number of empirical studies ([17], [18], and [28]) have found out that users provide a very inaccurate estimates of resources required and job runtime. The estimates provided by these studies are often 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 [18] 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 estimate ensuring that users do not quote low resource and runtime estimates. The inaccuracies in the estimates despite these incentive mechanisms show that a fundamental problem; it is inherently hard for users to estimate resource requirements.

Poor estimates of resource requirements can significantly undermine the efficiency of the scheduling algorithms. Further, job submitters will estimate their costs based on these resources requirements estimates and known pricing policies. Thus, 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 caused poor estimates of resource requirements so that buyers can better estimate costs, schedulers can better assign jobs to resources and resource providers can better plan their capacity decisions.

Similarly, the focus of the studies given in [29], [30], [31] is on resource allocation and resource scheduling with references to market economy. However, Mutz et al. [17] has some interesting schemes that points to current research. Mutz et al. critically re-examined a batched queue-environment. They considered a simple form of batched-queue of jobs j_i for $i = 1, 2, \cdots, n$ waiting to be to be granted resources; where j_i receives service before j_{i+1}. The resources granted is based on the owners' parameters. Their basis for modelling the payment function depends on the user's behaviour which impose some undesirable externality constraints on the jobs on queue. With specific reference to the job value *vi* (currency based), and the delay in total turnaround time *d* expressed as a tolerance factor. Mutz et al. obtained a job priority model using efficient design mechanism in [32]. They also proposed a compensation function based on how the propensity with which a job scheduled for time t_{n-1} wishes to be done at time earlier.

The compensation that is determined by *d* is paid by the job who wishes to be done earlier and disbursed in the form of incentives (say more ccc) to the jobs before. The scheme given in [17] complies with a typical airline reservation system. For example, the holder of an air ticket may apply any of the following after a full purchase and confirmation is made. The holder may decide to postpone, travel before the previous specified date, or cancel. In any of these cases, the ticket holder pays a compensation.

Consideration for payment mechanism is also given in [17]. The research efforts in this article work are novel in its application of real options and consideration for QoS. However, in the presented cloud resources pricing model, the compensation function originally given in [38] and similarly applied in [8] is also applied. This is similar in principle to the compensation function to an airline system in the model. Since the airline systems demonstrates a similar real options operation: in a computational cloud system, the holder of an option to use resources may decide on any one of the following; abandon, execute, postpone, defer the options to use the resources. In [29], Kang et al. considered one of the major challenges in managing the shared resources of a computational cloud in the presence of user QoS. This challenge becomes more prevalent when resources are orchestrated from less reliable desktop PCs, a user's requirement is biased toward some specific constraints: e.g., a request for 99% resource reliability for 24 hours. Therefore, matching the cost of a resource usage based on the forces of demand and supply may yield undesirable pricing results.

Existing pricing mechanisms (market-based economy) [33] has limitation in controlling mismatches between users QoS and market supplies. They do not foster the utilization of the underutilized low-quality resources. A Highly Available Job Execution Service (HA-JES) [29] dynamically and transparently visualizes underlying low-level computational resources to meet imbalance and unpredictable resource usage. HA-JES sits between a user and cloud resources and composes the underlying underutilized low-quality resource to build a high-quality resource that satisfy user requirement. HA-JES applies the von Neumann's approach for replicating redundant copies of tasks to have a high probability of one of them becoming highly available. Some issues include consistency and redundancy. The results show that utilization of low-quality resources (instead of being wasted) allows for the increase in the total computational capacity pool. Using this scheme may not necessarily guarantee a high and continual profit for the provider since she must maintain a consistent level of a high volume of resources pool to guarantee availability. Closely related is the work of Tan and Gurd [31] where they modelled cloud resources as a dynamic, distributed, two-sided market. The problem they attempt to solve is the uncooperative habits of cloud resources users where concurrent components need to be co-allocated. They applied the novel Stable Continuous Double Auction (SCDA), based on the more conventional Continuous Double Auction (CDA) (proposed). In their results, the SCDA delivered a continuous resource matching, high efficiency and low cost, allied with low price volatility and low bidding complexity and hence achieved superiority in terms of performance over the CDA. The work that is presented in this paper is the performance evaluation using real data to test the previously proposed model [4] on cloud resources pricing.

3. PRICING MODEL

A. Base Prices and Real Options

The following assumptions to aid the development of the presented model are hereby presented.

i. Assumption 1: The base prices for the ccc-s are set to preset values. These assumed prices are the prices that reflect the current real sale prices but discounted almost as close to 100%. For example, if a 1GB of Random Access Memory (RAM) cost \$70, a price, as low as \$0.005 a week for 0.5MB memory.

ii. Assumption 2: Since the resources exists in non-storable (nonstable) states, they are valued 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 it is assumed that the resources are nonstable, a high volatility (σ) affects the resources availability. This is responsible for a shorter time of use 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 the pricing scheme. 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 from S_0 to a new level S_0u with a payoff value of fu or move down from S_0 to a new level, S_0d and with a payoff value of f_d where u > 1 and d < 1. This led to a one-step binomial. A job using the cloud is defined as a service that need one or more of the ccc-s from start to finish.

B. Price Variant Factor

An important functionality of the model is the price variance factor (pf). The pf is a fuzzy number, a multiplier and based on the fuzziness (or uncertainty in changes in technology) given as $0 \leq pf \leq 1$. The 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, the resultant changes were captured using fuzzy logic and treating the pf as a fuzzy number. For a use time of (t_{ut}), a fuzzy value, pf, is expressed as a fuzzy membership function that is, $\mu(pf)$. For example, the cloud resources may become underused if users find better and faster ways to solve their computing problems. Therefore, to increase the cloud resources usage with more capacity for computations under same technology, the value of pf (ut) is set to 0.1 and with new technology, the pf = 1. The model therefore, adjusts the price in the use of cloud resources by $(pf(ut))^{-1}$ (for the cloud operator) while providing quality of service set at the Service Level Agreement (SLA) of the contract.

C. Trinomial Lattice Approach

We apply the trinomial-tree model [34] to price mainly American-style and European-style options on a single underlying asset. Options pricing under the Black-Scholes model [35] requires the solution of the partial differential equation and satisfied by the option price. To get option prices, a discrete time and state binomial model of the asset price is built and then the discounted expectations [36] is applied. Suppose S is current asset price and 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 [15] is given by $dS = (r - \delta)S\,dt - \sigma S dz$. For convenience, let x = lnS, this equation can be written as $dx = vdt + \sigma dz$, where $v = r - \delta - \sigma^2/2$. Consider a trinomial model of asset price in a small interval δt, the asset price changes are set to δx. Suppose this change remain the same or changes by δx, with likelihood of an up movement p_u, chance of steady move (without a change) p_m, and chance of a downward movement p_d.

The drift (because of known reasons) and volatility (σ, because of unknown reasons) parameters of the asset price can be obtained in the simplified discrete process using δx, u_p, p_m, and p_d. In a trinomial lattice the price step (with a choice) is given by $\delta x = \sigma\sqrt{3\delta t}$. By equating the mean and variance over the interval δt and imposing the unitary sum of the likelihoods a relationship between the parameters of the continuous time and trinomial (a discretization of the geometric Brownian motion (GBM)) can be obtained, that is,

$$E[\delta x] = p_u(\delta x) + p_m(0) + p_d(-\delta x) = v\delta t \quad (1)$$

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

From Equation (1),

$$E[\delta x^2] = p_u(\delta x^2) + p_m(0) + p_d(\delta x^2) = \sigma^2\delta t + v^2\delta t^2 \quad (2)$$

where the unitary sum of probabilities represented as

$$p_u + p_m + p_d = 1 \quad (3)$$

and $p_u, p_m, and\ p_d$ are probabilities of the price going up, down or remaining same respectively.

Solving

Equations (1), (2), and (3) yields the transitional probabilities;

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

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

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

The 1-step trinomial process could be repeated several times 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. I 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 $i: t = \Delta t$. 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

$$S[i,j] = S[0,0]\left(u^i d^j\right). \quad (7)$$

The options at maturity (that is, when $T = n\Delta t$ for European style options; $T \leq n\Delta t$ for American style options) are determined by the pay off. Therefore, for a call option (the intent to buy an asset at a previously determined strike price), the payoff $C_{n,j} = Max(0, S_{n,j} - K)$ and for a put option (the intent to sell) is given by $C_{n,j} = Max(0, K - S_{n,j})$. The value K represents the strike price at maturity T = $T = n\Delta t$ for a European-style option, and the strike price before, or on maturity for an American-style option. To calculate option prices, I apply the discounted expectations under the risk neutral assumption. For an American put option (for example), for i < n:

$$C_{i,j} = Max\left(e^{-r\Delta t}\left(p_u C_{i+1,j} + p_m C_{i+1,j+1} + p_d C_{i+1,j+2}\right), K - S_{i-j}\right) \qquad (8)$$

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

$$C_{i,j} = e^{-r\Delta t}\left(p_u C_{i+1,j} + p_m C_{i+1,j+1} + p_d C_{i+1,j+2}\right). \qquad (9)$$

While option price starts at $C_{0,0}$, I 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. I now model cloud resources based on the transient availability of the cloud compute cycles, the availability of compute cycles, and the value of volatility of prices associated with the compute cycles [39] and [40]. Given maturity date t, expectation of the risk-neutral value ($\hat{E}$), the future price F(t) of a contract on cloud resources could be expressed as (see for example [15]):

$$F(t) = \hat{E}[S(t)] = S(0)e^{\int_0^t \mu(\tau)d\tau} \qquad (10)$$

Consider a trinomial model of asset price in a small interval ¢t, the asset price increases by Δx, remain the same or decreases by Δx, with probabilities; probability of up movement p_u, probability of steady move (staying at the middle) p_m, and probability of a downward movement p_d. To price the multi-resources system [39], suppose a real option depends on some other variables such as the expected growth rate gcc_μ and the volatility respectively gcc_σ. Let $\frac{d\mathrm{gcc}_i}{\mathrm{gcc}_i} = \mathrm{gcc}_\mu \mathrm{dt} + \mathrm{gcc}_\sigma \mathrm{dz}_i$, for any number of derivatives of ccc such as (ccc_1; ccc_2, . . ., cccn) with prices p (p_1, p_2, . . . , p_n) respectively, then, $\mathrm{dlnS} = \frac{dp_i}{p_i} = \mu_i dt + \sigma_i dz$, where the variables $\mathrm{gcc}_i = \{the\ set\ of\ cloud\ resources\}$. Applying the price variant factor pf for pricing options:

$$\mathrm{d\ ln\ S} = [ccc(t) - pf\ ln\ S]dt + [stochastic\ term] \qquad (11)$$

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, this gives:

$$\mathrm{d\ ln\ S}_i = [gcc_i(t) - pf\ ln\ S_i]dt + \sigma_i dz_i \Big|_{i=1,2,3\cdots,n} \qquad (12)$$

The value of ccc(t) is determined so F(t) = $\hat{E}$[S(t)]; that is, the expected value of S is equal to the future price. A user may need compute cycles (bandwidth) in 3, 6, and 9 months from today and therefore decides to pay some amount, $s to hold a position for the expected increase. I show this using a 3– step trinomial in Figure ***. If the spot price for bandwidth is $\$s_T$ bit per second (bps) and the projected 3-month, 6-month, and 9-month future prices are $\$s_1$, $\$s_2$, and $\$s_3$ respectively. In this case, the two uncertainties are the quantity of bandwidth that will be available and the price per bit. However, the estimate for the stochastic process for bandwidth prices can be obtained by substituting some assumed values of pf and σ (for example, pf = 10%, σ = 20%) in Equation (11) and get the value of S from Equation (12). Suppose $V_{l,j}$ represents the option values at *l* for *l* = 0, 1, 2, . . . , n- 1 level and j node for j = 1, 2, . . . , (2*l* + 1) (for a trinomial lattice only); that is, $V_{1,1}$ represents the option value at level 1 and at p_u.

4. MODEL ARCHITECTURE AND INTEGRATION WITH CLOUDSIM

A. Model Architecture

In this model architecture, I normalize base prices for the cloud resources using SLA and QoS as constraints for individual (local) clouds. I also consider economic and market behaviours for resources conflict in the cloud. For a detailed discussion on the model architecture see [4].

We setup the base prices for ccc-s based on real and current market value per node. For example, a 1GB of RAM costs about $70. For a minimal 2 years for the 100% return on investment for a cloud operator, I charged $\$47.95*10^{-6}$ per day per MB. Similarly, suppose it costs $50.00 for 100GB hard disk, a base charge of $\$34.25*10^{-8}$ per day per MB was set as a charge for the cloud storage. Similarly, I set a base price of $\$34.25*10^{-6}$ per day per MHz of CPU cycles for a 1.00 GHz dual-core processor.

These base prices that I choose are as low as possible because of repeated use of the ccc-s by many consumers (users). This means the resource provider could still get the return on their investments despite the small charges and small charges for the consumers mean they will be satisfied to a large extent for the return on their premium payments. However, the base prices do not take into account the overhead cost due to other infrastructures that are not ccc-s. Such infrastructures include operational expenses that are incurred – power, building infrastructures, and air-conditioning expenses, are that the ccc-s are fully utilized.

B. CloudSim

In [4], I designed a pricing architecture – abstract representation that comprises of cloud services layer and the price optimization layer. In this architecture, the cloud services layer houses the middleware while the pricing model integrates on the top layer of the middleware. In the current study, I integrate the top layer of the pricing model (price and usage optimization level) onto the top layer of CloudSim [37]. Figure 2 shows the layered architecture of the CloudSim. It is a toolkit that administrates time variable resource assignments. The first layer (at the bottom of Figure 2) consists of Java Virtual Machine (JVM), the CloudSim core simulation engine. The JVM 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 modelling of computational cloud entities. Simulation of the cloud resource brokers takes place in the fifth layer. The top layer consists of the cloud scenario, user requirements, I/O interface, and application configuration. Figure 3 shows the pricing architecture (which is integrated into the CloudSim toolkit in Figure 2) to price the cloud resources using financial option-based model [4]. The integration of the financial option-based pricing model to the CloudSim toolkit is at the top levels of the CloudSim toolkit architecture.

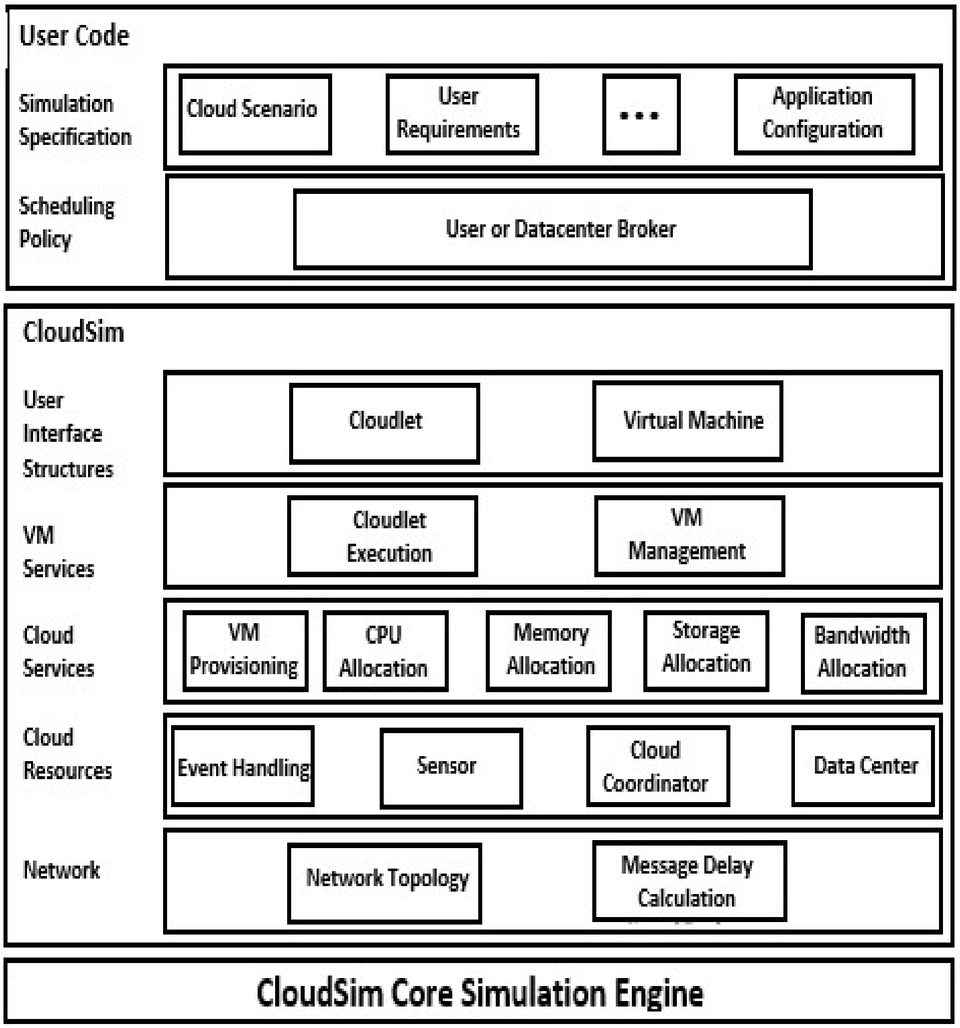

Figure 2. CloudSim Toolkit Layered Architecture/Middleware [37]

C. Integration

We develop user application that runs trinomial lattice (to compute option price) and optimize resources usage (reverse Dutch auction) and deployed to the user code layer of the CloudSim toolkit. A classical Dutch auction begins with a high initial price which is constantly reduced until all items are sold. The price reduction happens through incremental discounts. In the approach using the inverse Dutch auction, the time between successive bids is similar to the time steps of the trinomial lattice. In the inverse auction – cloud resource users begin with an initial willingness to use the cloud resources (that is, to exercise the option). This means that the users are willing to pay a regular price without any discount (from the view point of the cloud resource provider, this amount is a high price) and the provider continues to increase the willingness to reach the capacity of the resources by increasing the discounts to the users.

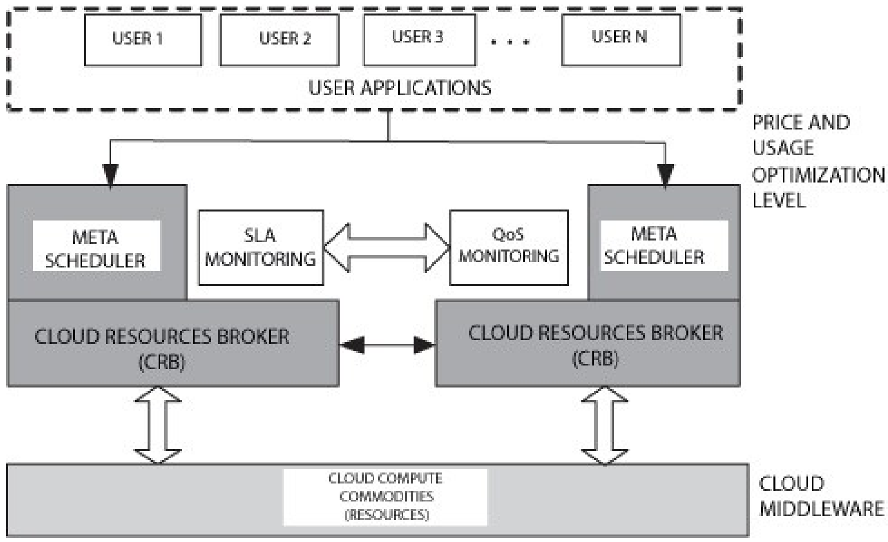

Figure 3. Pricing Architecture

D) Integrated Architecture: The resultant architecture is given in Figure 3.

D. Pricing Cloud Resources in CloudSim

Cloud Simulation Algorithm
1.Begin: CloudSim;
2. Begin: Create cloud scenario;
/* Create the environment scenario and initialize the CloudSim Toolkit*/
3. Start: for each cloud resource do; /*Ri*/
4. Create new processing elements; /*PE*/
5. Create new machines;/*Mi*/
6. Create new resources Ri; /*where Ri have one or more Mi that also have one or more PEs
7. End: Create cloud scenario
8. Begin: Create users' scenario
9. For each user do
10. Create a cloud task;
11. End: users' scenario;
12. Begin: bid and trinomial;
13. For each user do
14. For cloud resource do
15. Resource bid and utilization;
16. Apply trinomial;
17. compute option value;
18. End: resource use and trinomial;
19. Start CloudSim simulation;
20. Obtain simulation data;
21. End CloudSim simulation;
22. End:CloudSim;

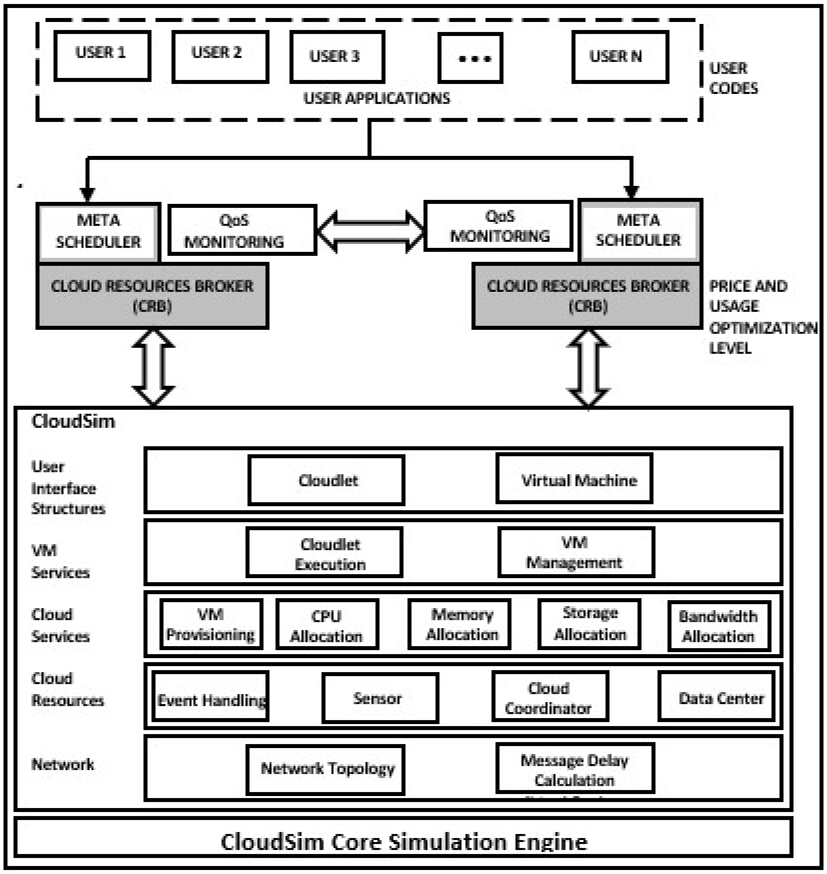

Figure 3. Integrated Architecture

The algorithm in Section 4 (d) shows the integration procedure in my simulation in Figure 3. The simulation starts with CloudSim initialization and the creation of CloudSim resources with different configurations. This is followed by the creation of CloudSim users. Each user has different requirements from other users. The modelling of the CloudSim resource starts with the CPUs (also called Processing Elements (PEs)). Each PE is created with several Million Instructions Per Second (MIPS) and the PEs combine to form one machine while several machines form one single CPU node in the CloudSim simulation. The second phase of the simulation setup is the specification for the trinomial lattice for option pricing. The parameters are configured as follows: Asset Price S, Strike Price K, Time to Maturity T, Interest Rate r, Number of Steps N, Interval time between 2 steps Δt, and Volatility σ.

5. EXPERIMENTS, RESULTS, AND DISCUSSION

We setup base prices for the various ccc-s using real market and current market values as follows. A charge of $95.89*10^{-6}/day/MB is set for a 2GB storage, similarly, a charge of $68.49* 10^{-8}/day/MB for a 200GB hard disk, and a charge of $68.49*10^{-6}/day/MHz for CPU cycles are respectively set.

Digital Innovations
& Contemporary Research in SCIENCE, ENGINEERING & TECHNOLOGY
Vol. 10. No. 2, June 2022

A. Real Data Collection, Results, and Discussions

Usage pattern (metadata) from Amazon S3 were collected. The trace usage pattern collected include number of processors, memory, CPU time, run time, and wait time. First, I analyse these traces. To price the ccc-s, I run the trinomial lattice using the following model parameters: For example, for a one-step trinomial tree I use strike price (K = \$0.70), resources price (S = \$0.80), expiration time (T = 0.5 in years), interest rate (r = 0.06), volatility (σ = 0.2), and the number of time steps (Nj = 2N + 1). I extend this study by varying the volatility σ in steps of 0.0, 0.1, . . . , 0.7 and N = 4, 8, 16, 24. For a 6 month contract, for example, N = 3 would mean a 2 month step size and N = 12 would mean a 2 week step size.

In my experiments, I simulated the cloud compute commodities (ccc) and watch user's requests. For a call option, I simulate the effects of time of use of one of the ccc-s such as memory (RAM), hard disk (HD), and CPU. In other words, I study effect of time of exercise of the option. I start with memory (one of the ccc-s) using the following parameters: S = $\$6.85*10^{-7}$, T = 0.5, r = 0.06, N = 4,8,16,24, σ = 0.2, and Nj = 2N + 1; I varied K for both advantageous (in-the-money option) disadvantageous (out-of-the-money) direction for the users. These values reflect the market value of this raw infrastructure, in general. There is no certainty about the RAM availability in the example clouds. However, it was easy to map the parametric values to the infrastructure available in the clouds. This is true study the variation in several step sizes. The effects of the variations (uncertainty) that exists between the total period of the option contract and the time of exercise on option value was further analysed. Figure 4 (a) shows an in-the-money option value for RAM while Figure 4 (b) shows out-of-the-money call. Over the number of step sizes, the option value reaches a steady state. This depicts that the number of jobs running increases steadily almost throughout the experiment.

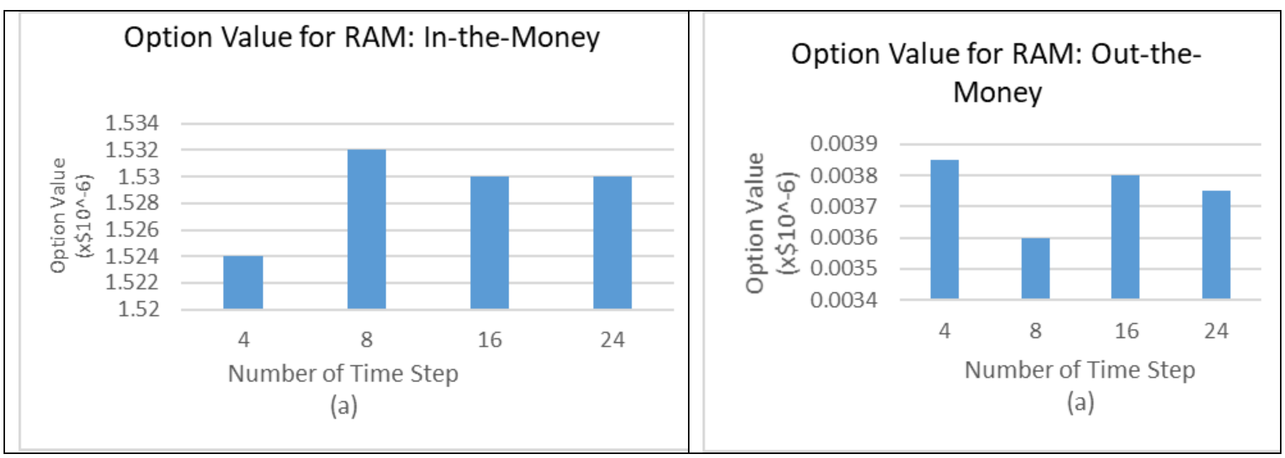

Figure 4. Option Value for RAM: (a) In-the-Money (b) Out-of-the-Money.

Similarly, the option values for both in-the-money and out-of-the-money for CPU using the limits S = \$68.49 and K = \$68.47 and \$80.47 (all values scaled at ($*10^{-6}$)) and simulated for a varying time step of 4,8,16,24 were obtained from the simulation. Figure 5 (a) shows the in-the-money option value for CPU. The option values for in-the-money for other cccs under the current study include RAM in Figure 4(a) and HD in Figure 6. They show an increasing option value which increases with the number of time steps.

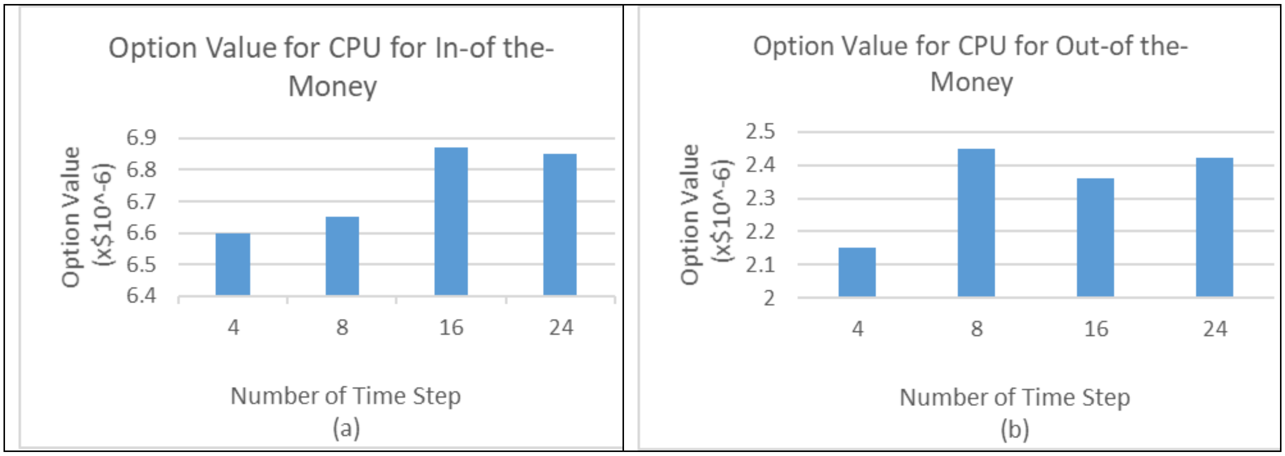

Figure 5. Option Value for CPU: (a) In-the-Money (b) Out-of-the-Money.

The observed behaviour shows that at any given time, a consumers' cost for using the cloud resources is the base cost and the extra cost which depends on the time of use of the ccc. However, for an equilibrium service-profit, I imposed a price modulation – price variant factor called pf (see Section 3(b)). The value of the pf depends on changes in the technology or architecture of the cloud infrastructure. These variations are unknown before exercising the options to hold the use of cloud resource. Therefore, deciding the exact price of ccc in real life is uncertain and hard to predict. Hence, to increase ccc use (ut) with more computing facilities and with same technology, I set the value of pf (ut) to 0.1 and with new technology, the pf = 1.0. Fuzzified boundary value of pf is set up as pf (ut) = [0.1, 1.0] to simplify fuzzification. The model, therefore, adjusts the price in the use of cloud resources by $(pf\ (ut))^{-1}$ (for the cloud operator) while providing quality service to the user. For example, applying pf reverses an unprofitable late exercise of an out-of-the-money option value to an early exercise of in-the-money option value in a 10% adjustment. Figure 5 (b) shows a corresponding out-of-the-money option value for CPU.

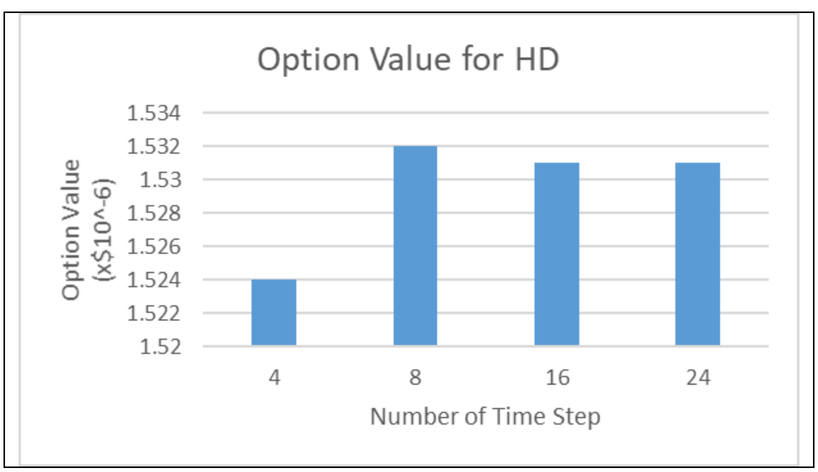

Figure 6. Option Value for HD – In-the-Money.

This procedure was repeated for the various clouds and for the various ccc-s first individually and then using a combination of the individual ccc-s. Figure 7 shows execution time for HD, CPU, and RAM at various time steps. Analysing the option values captured from the experiments it was observed that the option values converge (error level set at 0.1% for academic purpose) in 24 steps. Increasing the computation beyond 24 steps did not yield better solution for the option values it only increases the computational cost. This is in contrast to the finance market where the stock prices are highly volatile and for convergence one needs to go for very small step sizes (in other words large number of steps). Since the time required to achieve a steady state in option value increases with the number of nodes as shown in Figure 7 without yielding better solution, so at 24 steps, the simulation was terminated. Figure 8 shows option values computed for various commodities. In these experiments, the option value for individual ccc was also captured.

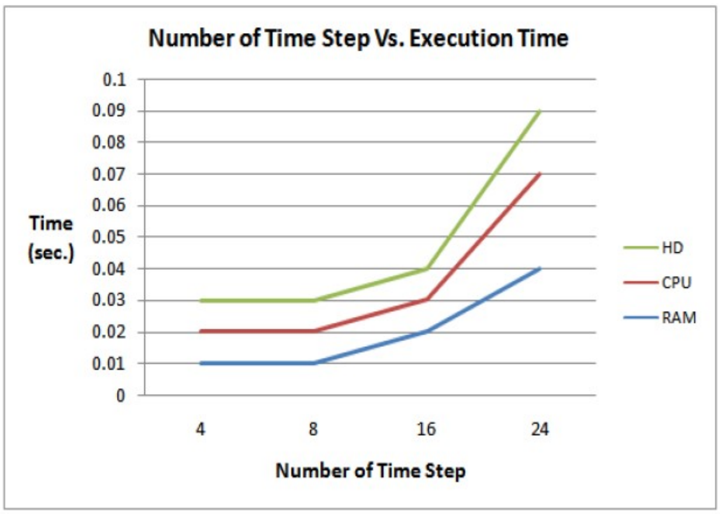

Figure 7. Execution Time for Various Commodities

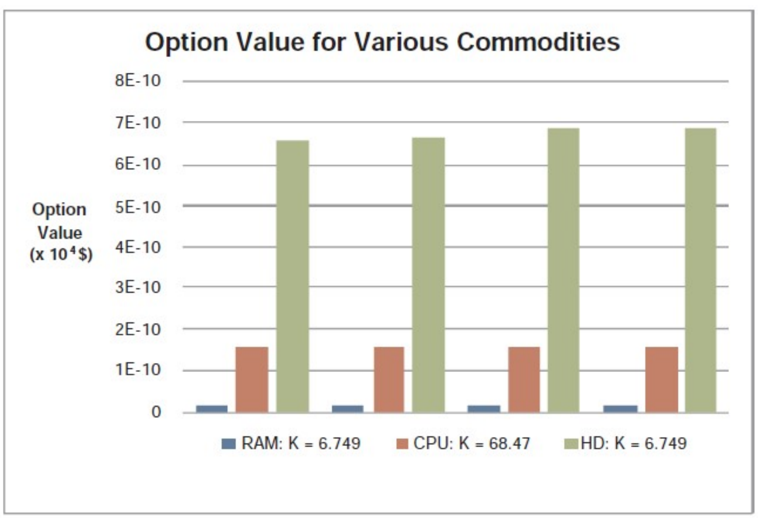

Figure 8. Option Value for Various Commodities.

However, it is noted that most cloud jobs use resources in a combination of more than two. A good correlation between such combined set of cccs is being worked to model a price that will be needed to compute the option values for such combination of cccs.

6. CONCLUSIONS AND FUTURE WORK

In this paper, I have studied, analysed, and made comparison for cloud resources utilization using traces form two educational clouds. I focus on the resources usage patterns to simplify the design and to develop a cloud resources pricing model. The results from the trace analysis show that some clouds could provide 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 same cloud resources usage patterns obtained from the traces of these two real clouds were used to develop a novel pricing model as a real option problem. Two important contributions are: (i) option value computation for cloud resources usage and to select the best point of exercise of the option to utilize any of the cloud resources. This helps the user as well as the cloud resources provider to optimize resources for profitability; in other words, I achieve an equilibrium condition; (ii) the study also incorporate a price varying function *pf* which controls the price of the resources and ensure the cloud 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. The future work will focus on the larger problem of pricing cloud resources for applications that use diverse resources across varied clouds simultaneously. For example, if an application needs memory in Microsoft Azure and the early CPU time in IBM cloud simultaneously, then I 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).

REFERENCES

[1] Ray Partha Pratim. "An Introduction to Dew Computing: Definition, Concept and Implications - IEEE Journals & Magazine". *IEEE Access*. **6**: 723–737. doi:10.1109/ACCESS.2017.2775042. S2CID 3324933.

[2] Montazerolghaem, Ahmadreza; Yaghmaee, Mohammad Hossein; Leon-Garcia, Alberto (September 2020). "Green Cloud Multimedia Networking: NFV/SDN Based Energy-Efficient Resource Allocation". *IEEE Transactions on Green Communications and Networking*. **4** (3): 873–889. doi:10.1109/TGCN.2020.2982821. ISSN 2473-2400. S2CID 216188024.

[3] Amazon S3 Pricing [Online] Available: https://aws.amazon.com/s3/pricing/ [visited on February 17, 2022]

[4] D. Allenotor and R. Thulasiram, "Cloud resources pricing: A novel financial option-based quality of service-profit quasi-static equilibrium model," in The 9th IEEE/ACM International Conference on Cloud Computing (Cloud 2008). September 29-October 01, Tsukuba, Japan., September 2008.

[5] D. Allenotor, R. Thulasiram, and P. Thulasiraman, "A financial option based cloud resources pricing model: Towards an equilibrium between quality of service for users and profitability for service providers," in 4th IEEE Intl. Conf. on Cloud and Pervasive Computing (GPC 2009). May 04-08 Geneva., May 2009.

[6] D. M. Chance and P. P. Peterson, Real Options and Investment Valuation. The Research Foundation of AMIR, First Edition, 2002.
[7] M. Palankar, A. Onibokun, A. Iamnitchi, and M. Ripeanu, "Amazon s3 for science clouds: a viable solution?" in Poster: 4th USENIX Symposium on Networked Systems Design and Implementation (NSDI'07), April 2007.
[8] Amazon. Amazon elastic compute cloud (amazon ec2). [Online, viewed (2022, June)]. Available: http://aws.amazon.com/ec2/
[9] A. Caracas and J. Altmann, "A pricing information service for cloud computing," in MGC '07: Proceedings of the 5th international workshop on Middleware for cloud computing. New York, NY, USA: ACM, 2007, pp. 1–6.
[10] AppNexus. (2009, September) Appnexus. [Online]. Available: http://www.appneus.com/
[11] GoCloud. Cloud hosting: Instant windows and Linux cloud servers. [Online, viewed (2022, June)]. Available: http://www.gocloud.com/
[12] Google. App engine. [Online, viewed (2022, June)]. Available: http://code.google.com/intl/de-DE/appengine/
[13] M. Corporation. Azure services platform. [Online, viewed (2022, June)]. Available: http://code.google.com/intl/de-DE/appengine/
[14] Joyent. Accelerator. [Online, viewed (2022, June)]. Available: http://joyent.com/accelerator/
[15] J. Hull, *Options, Futures, and Other Derivatives*. Prentice Hall, Ninth Edition, 2015.
[16] C. S. Yeo and R. Buyya, "Pricing for utility-driven resource management and allocation in clusters," International Journal of High-Performance Computing Applications, vol. 21, no. 4, pp. 405–418, 2007.
[17] A. Mutz, R. Wolski, and J. Brevik, "Eliciting Honest Value Information in a Batch-Queue Environment," in The 8th IEEE/ACM International Conference on Cloud Computing. September 1921, Austin, Texas, USA., September 2007.
[18] 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.
[19] C. Daskalakis, G. Schoenebeck, G. Valiant, and P. Valiant, "On the complexity of Nash Equilibria of Action-Graph Games," in SODA '09: Proceedings of the Nineteenth Annual ACM -SIAM Symposium on Discrete Algorithms. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2009, pp. 710–719.
[20] M. P. Wellmana, W. E. Walsha, P. R. Wurmanb, and J. K. MacKie-Masonc, "Auction Protocols for Decentralized Scheduling," Games and Economic Behaviour, vol. 35, no. 1–2, pp. 271–303, 2001.
[21] R. Wolski, J. Plank, J. Brevik, and T. Bryan, "Analyzing Market-Based Resource Allocation Strategies for the Computational Cloud," Int. Journal of High-Performance Computing Applications, vol. 15, no. 3, pp. 285–281, 2001.
[22] C. A. Waldspurger, T. Hogg, B. A. Huberman, J. O. Kephart, and S. W. Stornetta, "Spawn: A Distributed Computational Economy," IEEE Transaction on Software Engineering, vol. 18, no. 2, pp. 103–117, 1992.
[23] R. Buyya and S. Vazhkudai, "Compute Power Market: Towards a Market-Oriented Cloud," in The First IEEE/ACM International Symposium on Cluster Computing and the Cloud (CCCloud 2001), 2001.
[24] R. Buyya, "Economic models for resource management and scheduling in cloud computing," Ph.D. dissertation, Monash University, 2002.

[25] K. Hosanagar, J. Chuang, R. Krishnan, and M. D. Smith, "Service adoption and pricing of content delivery network (cdn) services," Management Science, vol. 54, no. 9, pp. 1579–1593, 2008.
[26] K. Lai, L. Rasmusson, E. Adar, L. Zhang, and B. A. Huberman, "Tycoon: An implementation of a distributed, market-based resource allocation system," Multiagent Cloud Syst., vol. 1, no. 3, pp. 169–182, 2005.
[27] D. Thain, T. Tannenbaum, and M. Livny, "Distributed computing in practice: The condor experience," Concurrency and Computation: Practice and Experience, pp. 1–36, 2004.
[28] 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.
[29] 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 Cloud Computing. September 1921, Austin, Texas, USA., September 2007.
[30] A. Sulistio, W. Schiffmann, and R. Buyya, "Using Revenue Management to Determine Pricing of Revervations," in Proc. 3rd International Conference on e-Science and Cloud Computing (eScience 2007) Bangalore, India, December 10-13, 2007, 2007.
[31] Z. Tan and J. R. Gurd, "Market-Based Cloud Resource Allocation Using a Stable Continuous Double Auction," in The 8th IEEE/ACM International Conference on Cloud Computing. September 1921, Austin, Texas, USA., September 2007.
[32] V. Krishna and M. Perry, "Efficient mechanism Design, 1998 [online]. Available http://ratio.huji.ac.il/dp/dp133.pdf," 2007.
[33] R. Buyya, D. Abramson, and S. Venugopal, "The Cloud Economy," in IEEE Journal, 2005.
[34] P. P. Boyle, "Option Valuing Using a Three Jump Process," International Options Journal, vol. 3, no. 2, 1986.
[35] F. Black and Scholes, "The Pricing of Options and Corporate Liabilities," Journal of Political Economy, vol. 81, no. 3, 1973.
[36] P. Wilmott, Derivatives: The Theory and Practice of Financial Engineering. John Wiley and Son Ltd, 1998.
[37] A. Sulistio, U. Cibej, S. Venugopal, B. Robic, and R. Buyya, "A toolkit for modelling and simulating data clouds: an extension to CloudSim,"Concurrency and Computation: Practice and Experience (CCPE), vol. 20, no. 13, pp. 1591–1609, 2008.
[38] D. Allenotor, "A Fuzzy Real Option Model for Pricing Grid Compute Resources," PhD., Department of Computer Science, University of Manitoba, Winnipeg, Canada, 2010. December, 2010.
[39] D. Allenotor, D. Oyemade, & A. Ojugo. "A Financial Option Model for Pricing Cloud Computational Resources Based on Cloud Trace Characterization," African Journal. of Computing & ICTs. Vol 8, No. 2, Issue 2. 2015. Pp 83-92.
[40] D. Allenotor and R. Thulasiram. "A Financial Option Based Model for Pricing Multiple Grid Compute Cycles, " Journal of the Nigerian Association of Mathematical Physics Volume 25, No.2, 2013, 55 - 64