A Financial Option Model for Pricing Cloud Compute Resources

David Allenotor
(correspondent author)
Nig. Tel: 07056009584
Can. Tel: +16478003156
Dept. of Computer Science
University of Benin
PMB 1154 Benin City, Nigeria
dallen@uniben.edu

Oseghale O. Igene
Dept. of Computer Science
University of Benin
PMB 1154 Benin City,
Nigeria

Ruppa K. Thulasiram
Dept. of Computer Science
University of Manitoba
R3T 2N2 MB, Canada
tulsi@cs.umanitoba.ca

Abstract

Cloud compute resources such as managed computing power, storage, platforms, services, CPU cycles, memory, and network bandwidths exist as compute instances. We describe these cloud resources as Cloud Compute Commodities (C3). One of the specific characteristics of C3 is guaranteeing their availability since they exist as instances (or compute cycles) of C3. This specific C3 characteristic feature make pricing them a challenge. Several initiatives (GoGrid, Amazon Elastic Compute Cloud (EC2), Simple Queue Service (SQS)) have developed various frameworks for cloud resource management and cloud economies using resource optimization techniques. However, because it is believed that cloud resources usage is relatively affordable, research efforts to model a standard pricing procedure that capture the realistic priced value of cloud resources have not received attention.

This paper is positioned to develop a novel approach for pricing C3. The novelty in the model design is in the application of the theory of financial option to price instances of C3. To achieve the set objectives, we apply our three research threads; financial option (to model price movements), real option (to capture the realistic value of the priced resources), and fuzzy logic techniques (to characterize availability and hence the user satisfaction and provider profitability measured as Quality of Service (QoS)). We simulate our model using real trace data with cloudbus (a market-oriented cloud computing simulation toolkit) to validate the model.

Key Words: Financial Options, Option Pricing, Distributed Systems, Cloud Computing, Cloud Compute Commodities.

1.0 Background/Introduction

Over the years, there has been an upsurge in the development and application of resource-intensive computations. The Search for Extraterrestrial Intelligence at Home (SETI@Home) [1] and Large Hadron Collider (LHC) [2] are two examples of projects that use resources and computations beyond super-computing. The consequence of the upsurge in resource-intensive computations is the emergence of pools of virtualized resources and service abstractions called computing cloud or cloud computing. Foster et al. [3] describe the cloud as a large-scale of distributed computing paradigm that is driven by economies of scale where a pool of abstractions of virtualized resources (managed computing power, storage, platforms, services, CPU cycles, memory, network bandwidths, throughput, disks, processor, and various measurements, instrumentation tools) are delivered on-demand to external customers over the Internet. We refer to these virtualized resources as Cloud Compute Commodities (*C*3) which are provisioned as needed from a centralized infrastructure. The virtualized resources are delivered over the Internet as services, i.e., they are seen as x-as-a-service (where x is any one type of the C3-s).

Resource virtualization have enabled cloud resource providers (haven taken advantage of the resource virtualization) to scale up resource availability. However, the provider put the cost of access of the resources on metered basis. In Section 2.4, we provide Amazon's EC2 static pricing available in the United States and in Europe in Table 1 and Table 2 respectively. A metered-based cost is a static pricing approach which does not offer flexibility advantage for the user [4]. In a situation where availability is scaled up with multi-tenancy (as it is with cloud resources), an offer of user flexible pricing agility is necessary for the delivery of on-demand user computing requirements. Research efforts in cloud computing have focused mostly on security related issues [5, 6, 7, 8], cloud resource utilization improvement [9], and middleware/infrastructure-based is-sues [10]. Since cloud resources usage has been relatively cheap, there are only a few efforts reported in the literature that price cloud resources. The pricing strategy adopted so far has been based on static price allocation without a standard and justification for the imposed charges. The reason is because pricing instances (or virtual machines) of cloud resources is a major challenge from the view point of a generic pricing problem. Since C3 are virtualized, it follows that their availability are both timed and instance-based. It also follows that C3 usage could occur with flexibilities in usage. These flexibilities between using the cloud resources immediately and using them later in the future. If t_n is the time of usage, then $t_n = 0$ denotes immediate use and $t_n > 0$ denotes future usage time. To price C3 in the presence of these flexibilities, we treat them as real assets and employ our three research threads to obtain their realistic value; financial option (to model price movements), real option (to capture the realistic value of the priced re-sources), and fuzzy logic techniques (to characterize avail-ability and hence the user satisfaction and provider profitability measured as Quality of Service (QoS)). We model the cloud resources as real assets thereby see them as commodities. This enables us to develop an option pricing solution to price the cloud resources. We use a discrete time approach to capture the cloud spot prices starting a simple binomial tree [11] and we then extend to a trinomial in an American option approach for a multi-asset C3. We model the pricing function as a real option problem by formulating the cloud compute commodities pricing problem as real option pricing problem multi-asset C3. The cloud is often seen as an infinitely scaled visualization (projected scalability) of resources in terms of resource availability. However, cloud providers (operators) may not fully support infinite and unlimited cloud resource scalability because of users' requests/computational needs (actual scalability) that does not scale with the same proportion.

1.1 Financial Options

Options are derivative securities because their value is a derived function from the price of some underlying asset upon which the option is written. They are also risky securities because the price of their underlying asset at any future time may not be predicted with certainty. This means the option holder has no assurance that the option will be *in-the-money* (i.e., yield a non-negative reward), before expiry. A financial option is defined (see, for example [12]) as the right to buy or to sell an underlying asset that is traded in an exchange for an agreed-upon 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 forfeits the premium paid at the beginning of the contract. The exercise price *(strike price)* specified 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 to purchase an underlying asset at the specified strike price. On the other hand, a *put option* grants the holder the right to sell an underlying asset at the specified strike price. An *American option* can be exercised at any time during the life of the option contract while a *European option* can only be exercised at expiry.

1.2 Real Options

A real option provides a choice from a set of alternatives. To hold a real option means to have a certain possibility for a given time to either choose for or against making an investment decision without upfront commitment. In the context of this study, these alternatives include the flexibilities of exercising, deferring, finding other alternatives, waiting or abandoning an option. We capture these alter-natives using fuzzy logic [13] and express the choices as a fuzzy number. A Fuzzy number is expressed as a membership function that lies between 0 and 1, i.e., a member-ship function maps all elements in the universal set X to the interval [0, 1]. In this paper, we focus on the valuation of a compute cloud resources as one of the extensions to the research efforts carried out in [4]. We emphasize the provision of a satisfaction guarantee in terms of the QoS requirements for cloud resource users and resource owners through a regulated profit level using a real option approach. We achieve the stated objective by introducing and varying the Option Pricing Factor $(p_o f)$. The contributions of this paper will help to achieve some general cloud system objectives such as: (i) to keep the cloud "busy" for optimal gain (profitability wise); this ensures wait states are minimized, (ii) to provide an assessment for the evaluation of the cost of use of cloud resources and applications in order to justify its design and estimate the future benefits, (iii) to provide a generic plan for managing the resources/infrastructure of a cloud that are essential for meeting the peak demands of cloud resources utilization, (iv) to facilitate continuous re-search activities through timely upgrades on the cloud infrastructure, and (iv) to provide a satisfaction guarantee in terms of the QoS requirements for cloud resource users and resource owners profitability.

The rest of this paper is organized as follows. In Section 2 we provide a review of related work and provide the model formulation and assumptions in Section 3. We dis-cuss the pricing infrastructure in Section 4. In Section 5 we present the results of our experiments and in Section 6 we provide some concluding remarks.

2 Related Work

Large corpuses of related work have applied financial option pricing to price financial derivatives. One of the most prominent works is the popular Black-Scholes Model (BSM) [14]. The BSM is applied in situations where we experience continuous price jumps (with a higher volatility) and it requires solving a partial differential equation before we can compute the price movement. The binomial lattice could also be used to approximate discrete price jumps. One example of its application is seen in the area of deregulating electricity and of course gas market in order to design an efficient distributed network for the pricing framework. Allenotor and Thulasiram in [15] applied the theories of financial option to price grid resources given uncertainty in demand. The work in [15] is generally based on the risk factors associated with pricing; whether the user will make any gain with respect to obtaining the expected (stated QoS) and whether the provider will make any gain in terms of profitability considering the cost of infrastructure in the computational grid.

Rhaman et al. in [16] proposed to utilize option pricing technic to mitigate the risk of price fluctuations in spot markets. Their main idea is to apply a combination of spot instances as well as option instances to schedule work-load. Basically, they developed a cloud provider-side option model based on binomial option pricing model. The approach in Rahman et al. [16] characterized the cost of European options. Recall in Section 1 on financial option background, European option cannot be exercised any time before the expiration even when the option value at the prevailing time is optimal or profitable. However, American option (that we choose to apply in this paper) gives the user a handle of flexibility to exercise the option any time before the expiration date. This means at any point in time, the user can exercise the option. In this paper, we apply American options to provide flexibility as distinct from Rahman et al. [16].

Currently, the cost of using cloud resources is only marginal when compared to owned physical computing infrastructures. However, a trend is developing due to large interest in cloud for public computing. Hence, a sudden explosion of cloud usage is expected in near future geared towards pushing cost, management, upgrade, and maintenance of infrastructures to the provider. With an outburst of cloud resources usage, the marginal cost may rise above a threshold value which nullifies the intended advantage. To avoid such bottleneck, Amazon has introduced a Simple Storage Service (S3) [17] for grid consumers. S3 offers a pay-as-you-go online storage, and as such, it also provides an alternative to in-house mass storage. In the literature, existing studies have not focus attention on the dynamics of cloud resource pricing. They have principally put research efforts on capacity storage and security concerns. For example, Palankar et al. in [17] reviewed the features of Amazon S3, focusing on the core concepts, the security model, and data access protocols. After characterizing science storage clouds in terms of data usage characteristics and storage requirements, they proceed to benchmark S3 with respect to data durability, data availability, access performance, and file download via BitTorrent (in order to reduce cost). With this information as a baseline, they evaluate S3's cost, performance, and security functionality. Palankar et al. concluded by observing that many science cloud applications do not necessarily need all three of S3's most desirable characteristics – high durability, high availability, and fast access. Finally, Palankar et al. noted that S3's current security architecture lack sufficiently support for delegation and auditing, and built-in trusts which are necessary for developing pricing and cost models.

To date and to the best of our knowledge, this is the first time research efforts to price cloud compute resources using the theories of financial option are made. However, some related efforts in the literature such as Rhaman et al. [16] applied financial options to schedule cloud workload and characterize the cost of scheduling cloud workload based on European options. Other commercial cloud resource providers such as Amazon (EC2), GoGrid, FlexiScale, Mosso, ElasticHost, Joyent, AppNexus, Google Application Engine, and Microsoft Grid offered arbitrary charges without user-side consideration for flexible pricing and they did not use financial options.

2.1 Simple Queue Service (SQS)

Amazon's Simple Queue Service (SQS) allows users to create one or more named queues. SQS supports three basic operations. A named message consisting of up to $256K$ of data and $4K$ of metadata can be written into a queue; one or more messages can be read from a queue; and one or more named messages can be deleted from a queue. When a message is read from SQS the reading process specifies a time lock. While the message is locked, no other read request will return the message. The reading process must delete the message before its time lock expires, otherwise another concurrent process may read the message instead.

2.2 Elastic Cloud Compute (EC2)

Amazon charges separately for computer resources consumed and for bandwidth. Amazon's pricing for EC2 is $\$0{:}10$ per hour for each instance, with fractional hours rounded up. Instances must be shut down with the EC2-terminate-instances command. Instances that have crashed and not automatically rebooted continue to acquire charges. Storage for S3 is charged on a flat basis of $\$0{:}15$ per gigabyte stored per month, with the amount of data stored being calculated twice each day. Amazon S3 uses buckets (similar to file directory) to store files. A few important S3 commands include PUT (put filename.xyz s3 – to load a file to a bucket (if bucket is s3)), LIST (ls s3 – list the contents of a bucket), and GET (get s3 – download the contents of a bucket). Starting June 1, 2007, Amazon has also charged a per-transaction fee of $\$0{:}10$ for every 1,000

PUT or LIST requests, and $0:10 for every 10,000 GET requests. Use of SQS is charged at $0:10 for every thou-sand messages sent. Originally pricing for bandwidth for the Amazon Web Services (AWS) services was charged on a flat basis of $0:20 per gigabyte transferred in or out of the Amazon network. Under the new pricing model Amazon charges $0:10 per gigabyte sent from the Internet to Amazon. Bandwidth from Amazon is charged at $0:18 per gigabyte transferred for the first 10 TB of data transferred per month, $0:16 per gigabyte for the next 40 TB transferred, and $0:13 for each gigabyte transferred thereafter. There is no charge for moving data between EC2, S3, and SQS.

In EC2, the resource charging concept is usage based. For example, the cost of computation (to use a CPU (based on time)) is obtained by multiplying the price per compute cycle with the number of compute cycles that the computation used. This type of static pricing scheme does not provide a fair charge for computation that requires shorter time to complete. Amazon EC2 charges at instant hour. For ex-ample, it offers computational power equivalent to a server with 1:7GHz Xeon CPU, 1:75GB RAM, 160GB HDD, and 250Mb/s of network speed, is priced at $0:10 per instant-hour [18]. In a similar approach, SUN Grid charge $1:00 per CPU hour. In contrast to Amazon, SUN does not pro-vide any specification for equivalent of hardware they offer for sale. SUN Grid idea is similar to paying a high penalty for jobs that takes longer to complete on the average. Al-though that may seem to charge a flat rate of $1:00 per CPU hour, however, users actually pay for the cycles they do not use. The goal of SUN Grid and EGEE (EGEE also use the same charging technique as SUN Grid called Price Authority [19]) is to minimize the queue waiting time using economic scheduling. The cloud is a dynamic resource reservoir as a result, static charging (usage-based charging, flat-rate charging, and waiting time charging) do not capture the essential goals that we address in this paper.

2.3 Market-Oriented Cloud Simulators (CloudSim)

A cloud infrastructure (large distributed system), consists of numerous parameters and executes complex interactions between the users and resources. These sets of complexity in the grid interaction make the analytical modeling of the cloud almost impractical without simulating the action of the cloud using a simulator. In this paper, we use a simulator. The fundamental advantage of the simulator is that it is independence to the execution platform. Therefore, simulating a mechanism of one million nodes distributed system on a single personal computer is not rare. This ad-vantage is made possible because the simulator does not run the real distributed system but an abstract model of it.

GridSim Toolkit [20] is similar to SimGrid [21] (Sim-Grid2 provides similar abstraction through the notion of Agents) in that it is a discrete event simulator. However, compared to SimGrid the GridSim's original design con-siders the existence of several brokers. GridSim manages several abstractions also called entities. These include user, broker, resource, grid information service, input and output. Users in GridSim are characterized by job type (execution time, number of parametric replications, etc.), the scheduling optimization strategy, activity rate, time zone, absolute deadline and budget and their associated relaxation parameters. When the brokers receive the tasks submitted by users carry out their scheduling algorithm. However, because the users must compete for the same set of resources (the resources are supposed to be finite); brokers have to find tradeoffs (meeting project deadline) between users requirements. GridSim describe its resources as number of processors, cost of processing, performance, internal scheduling policy, workload, and time zone. Significant difference be-tween GridSim and other simulators is seen in the management of inputs and outputs in two separate ways as a means to express the performance differences between parameters and results communication.

CloudSim [22] is a higher-level simulator designed to investigate interactions and interferences between compute instances and to manage their occurrences. Calheiros et al. [22] describes the layered implementation of the CloudSim (the enthusiastic reader is encouraged to see the layered architecture in [22]) software framework and architectural components. The lowest level of CloudSim consists of discrete event simulation engine. This engine implements the core functions for higher-level simulation such as queuing jobs, event processing, creation of processing elements or component systems such as services, host, data center, resource broker, and virtual machines. The simulation engine also managements the simulation clock. Basically, the CloudSim is implemented by functionally and programmatically extending libraries of the GridSim. The CloudSim layer also has the capability maintaining concurrency among thousands of system components.

2.4 Commercial Cloud Resource Providers

Consequent to the increasing cost of systems upgrades, maintenance, and rapid depreciation, various IT businesses (e-business and e-commerce activities) prefer to rent re-sources (services) instead of buying them for computation purposes. Several companies are now involved in offering resource as a service. Some of these companies include: AppNexus [23], GoGrid [24], Joyent [25], Google Application Engine [26], and Microsoft Grid [27]. The pioneering efforts of Amazon evolved from its requirement to power its own www.amazon.com. Amazon Web Services (AWS) include Simple Service Storage [17] (S3), Elastic Compute Cloud (EC2), and Simple Queue Service [18] (SQS). EC2 is a Web service that provides resizable compute capacity which is designed to make Web-scale computing easier for developers, while the related Amazon Simple Storage Service (Amazon S3) is a cloud storage that provides storage on demand [18]. Cloud computing is a system that involves dynamic scaling of visualized resources which are provided as a service (storage, platform, computing power, application, software, and hardware) – resource offered as a service (Resource as a Service (RaaS)) over the Internet.

The Amazon EC2 is a Web-scale computing service that provides scalable compute capacity [28]. It offers on-demand computing resources in the form of a virtual ma-chine that is accessible using the Internet. Using the Amazon EC2, a user has full control of virtual machines equivalent to a 1:7GHz Xeon CPU, 1:75GB RAM, 160GB HDD, 250Mb/s network at a price of $0:10 per instance-hour (or part hour) [18]. Amazon's EC2 uses the XEN [29] virtual image platform to offer on-demand operating system that consist of a complete virtual computer with a CPU, memory, and disk space. The pricing scheme offered by Amazon is simple. Charging is done per instance hour used. How-ever, it is not possible to apply dynamic prices or reserve computational power [28]. The Tables1 (1 Month = 4 weeks (24*7*4 = 672 hours)) and 2 shows the per hour static prices (as of November 2011) of the Amazon EC2 instance machine in the United States and Europe respectively. An Amazon instance machine could either be a 32bit or a 64bit machine. For instance, Table 1 shows a small 32 bit system that consists of 1:7GB of RAM and 160GB Hard Disk space. This machine could run a Linux Operation System (OS) for a $0:10 per hour (ph) or a Windows OS for $0:125 ph. Similarly, Table 2 shows a small 32bit instance machine of the same configuration but for Linux Operation System (OS) of $0:11 per hour (ph) or a Windows OS for $0:135 ph. Table 3 shows the summery of monthly resource cost in six companies that provide re-source as service.

[2] 1 Month = 4 weeks (24 7 4 = 672 hours)

Table 1: Instance of EC2 in the United States

	RAM	**DISK**	**LINUX**	**Windows**	**SQL Server**
Small (32*bit*)	1:7GB	160GB	$0:10/h $72/m	$0:125/h $90/m	
Large (64*bit*)	7:5GB	850GB	$0:40/h $284/m	$0:50/h $360/m	$1:10/h $792/m
Extra Large(64*bit*)	15GB	1690GB	$0:80/h $568/m	$1:00/h $720/m	$2:20/h $1584/m
High CPU(32*bit*)	1:7GB	350GB	$0:20/h $142/m	$0:30/h $214/m	
High Ex-tra CPU(64*bit*)	7GB	1690GB	$0:80/h $568/m	$1:20/h $864/m	$2:40/h $1728/m

Table 2: Instance of EC2 in the Europe

	RAM	**DISK**	**LINUX**	**Windows**	**SQL Server**
Small (32*bit*)	1:7GB	160GB	$0:11/h $79/m	$0:135/h $97/m	
Large (64*bit*)	7:5GB	850GB	$0:44/h $316/m	$0:54/h $388/m	$1:14/h $792/m
Extra Large(64*bit*)	15GB	1690GB	$0:88/h $633/m	$1:08/h $777/m	$2:28/h $1584/m
High CPU(32*bit*)	1:7GB	350GB	$0:22/h $158/m	$0:32/h $230/m	

High Extra CPU$_{(64bit)}$	7GB	1690GB	$0:88/h $633/m	$1:28/h $921/m	$2:48/h $1785/m

3 Model Formulation and Assumptions

The literature provides several schemes to price financial options. One of the existing schemes is the famous Nobel prize Black-Scholes Model [30] (BSM). The BSM requires a satisfaction of the solution of the partial differential equation of the option price. Another scheme is the application of a discrete time and state binomial model of the under-lying asset price. This requires the application of the dis-counted expectations [31, 32]. In this paper, we use the trinomial model [12] to solve the financial option pricing problem manifested as partial differential equation. This involves using a discrete time approach to capture the dis-counted expectations in a trinomial-tree structure.

To price cloud resources, we make the following three assumptions. First, we assume that the user has the right but not the obligation to use the cloud resources which he has paid for upfront in the stated times. Stating these assumptions ensures that the user gets the sole right to exercise the option any time before the expiration (American put or call option). Second, since the resources exists as instances, we value them as real assets which make them fit into the general stream of investment valuations that we can valuate in the real option valuation approach. Since the cloud resources are instances, their availability is characterized by a high volatility σ. Hence cloud resources utilization times are shorter relative to life of option in financial valuation methods. Third, we assume that a holder of the option to use the cloud resources has an obligation-free chance of exercising the right.

Table 3: Cost Comparison

Cloud Providers	Memory	HD	Cost/h	Cost/m
Amazon	7GB	1690GB	$0.08	$537.60
GoGrid	8GB	480GB	$0.037	$255.35
Flexisca;le	8GB	100GB	$0.53	$358.50
Mosso	15GB	620GB	$0.096	$645.12
ElasticHost	8GB	1862GB	$0.76	$510.72
Joyent	32GTB	100GB	$5.95	$4,000.00

The obligation-free assumption enables us to apply existing finance option valuation theory to model our pricing scheme. Consider (for example) an asset whose price is initially S_0 and an option on the asset whose current price is f . Suppose the option last for a time T and that during the life of the option the asset price can either move up from S_0 to a new level S_0u with a payoff from the option value of f_u or move down from S_0 to a new level, S_0d and with a payoff from the option value of f_d where $u > 1$ and $d < 1$. This leads to a two-step binomial in Figure 1. Similarly, a multi-Step binomial tree is obtained with more time steps. An algorithmic analysis of the binomial model is given in [32].

3.1 Discretized Real Option

The trinomial-tree model was introduced in [33] to price primarily American-style and European-style options on a single underlying asset. Option pricing under the Black-Scholes model [30] requires the solution of a stochastic partial differential equation (continuous time approach) and satisfied by the option price. Instead, option prices are obtained by building a discrete time and state binomial model of the asset price and then apply discounted expectations [18]. A generalization of such a binomial valuation model [12] to a trinomial model to price option is useful since solving the partial differential equation of the option price by the explicit finite difference method is equivalent to performing discounted expectations in a trinomial-tree [12]. The asset price in a trinomial-tree moves in three directions compared with only two for a binomial tree.

Consider an asset whose current price is S, and r is the risk-less and continuously compounded interest rate, the stochastic differential equation for the risk-neutral Geometric Brownian Motion (GBM) model of an asset price is given [12] by the expression:

$$dS = rSdt - \sigma Sdz. \tag{3.1.1}$$

which in terms of logarithm of asset prices can be given as

$$dx = vdt + \sigma dz. \tag{3.1.2}$$

where $v = \delta - \sigma^2 / 2$ and $x = \ln S$. Consider a trinomial model of asset price in a small time interval δt where we set the asset price changes by δx.The price changes with probabilities of an up movement p_u, probability of steady move (without a change) p_m, and probability of a downward movement p_d. Figure 1 shows a one-step trinomial lattice expressed in terms of δx and δt. The drift (due to known factors) and volatility (σ, due to unknown factors) parameters of the asset price can be captured in the simplified discrete process using δx, p_m, and p_d.

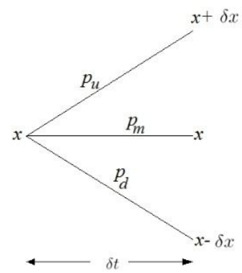

Figure 1: One-Step Trinomial Lattice.

The space step can be computed (with a choice) using $\delta x = \sigma\sqrt{3\delta t}$. A relationship between the parameters of the continuous time process and trinomial process (a discretization of the GBM) is obtained by equating the mean and variance over the time interval δt and imposing the unitary sum of probabilities, i.e.,

$$E[\delta x] = p_u(\delta x) + p_m(0) + p_d(-\delta x) = vdt. \tag{3.1.3}$$

From Equation (3.1.3),

$$E[\delta x^2] = p_u(\delta x^2) + p_m(0) + p_d(\delta x^2) = \sigma^2\delta t + v^2\delta t^2. \tag{3.1.4}$$

where the unitary sum of probabilities can be presented as

$$p_u + p_m + p_d = 1. \tag{3.1.5}$$

Solving Equations (3.1.3), (3.1.4), and (3.1.5) yields the transitional probabilities;

$$p_u = 0.5*((\sigma^2\Delta t + v^2\Delta t^2)/\Delta x^2 + (v\Delta t/\Delta x). \tag{3.1.6}$$

$$p_m = 1-((\sigma^2\Delta t + v^2\Delta t^2)/\Delta x^2). \tag{3.1.7}$$

$$p_d = 0.5*((\sigma^2\Delta t + v^2\Delta t^2)/\Delta x^2 - (v\Delta t/\Delta x). \tag{3.1.8}$$

The trinomial process of Figure 1 is repeated a number of times. As an example in Figure 2a, we repeat it for 8 steps. For number of time steps (horizontal level) n = 8, the number of leaves (height) in such a tree is given by $n+1=17$. At any level, the number of nodes i is given as $2i+1$. We index a node by referencing a pair (i, j) where i points at the level (row index) and j indicates the distance from the top (column index). Time t is referenced from the level index by $i : t = i\Delta t$. From Figure 2(a), 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).

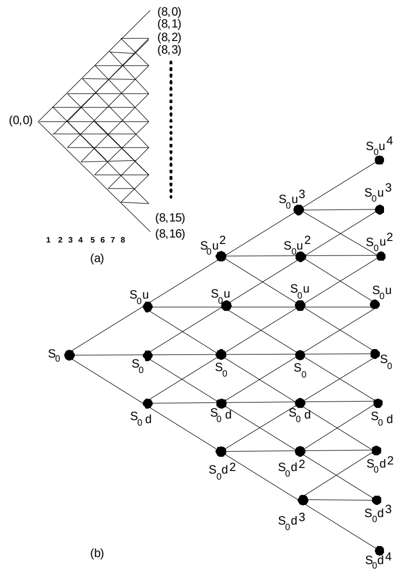

Figure 2: Trinomial Lattice: (a) 8-Step (b) 4-Step.

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 asset price could be computed from the number of up and down moves required to reach (i, j) from $(0,0)$ and is given by

$$S[i, j] = S[0,0](u^i d^j). \tag{3.1.9}$$

The option prices at maturity (i.e., when $T = n\Delta t$ are determined by the pay-off. For a call option (the intent to buy an asset at a previously determined strike price), the pay-off $C_{n,j} = Max(0, S_{n,j} - K)$ 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 $T = n\Delta t$ for a European-style option, and the strike price at any time be-fore or on maturity for an American-style option. To compute option prices, we apply the discounted expectations un-der the risk neutral assumption (see, [12]). For an American put option (for example), for $i < n$:

$$C_{i,j} = Max(e^{-r\Delta t}(p_u C_{i+1,j} + p_m C_{i+1,j+1} + p_d C_{i+1,j+2}), K - S_{i-j}). \tag{3.1.10}$$

whereas for a European call option (exercised on maturity only), for $i < n$,

$$C_{i,j} = e^{-r\Delta t}(p_u C_{i+1,j} + p_m C_{i+1,j+1} + p_d C_{i+1,j+2}). \tag{3.1.11}$$

While option price starts at $C_{0,0}$, we apply the expression for $C_{n,j}$ along with Equations (3.1.9), and (3.1.10) or (3.1.11) to obtain the option price at every time step and node of the trinomial-tree. We now model cloud resources based on the transient availability (a reserved quantity at a certain time (t_{n-1}) may be unavailable at t_n.) of the cloud 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 cloud resources could be expressed as;

$$F(t)=\hat{E}[S(t)]=S(0)e^{\int_0^t \mu(\tau)dr}. \qquad (3.1.12)$$

Consider the cloud as a resource system with multiple resources (C3s) $C3i=\{C3_1, C3_2, \cdots, C3_n\}$ where n is a finite number (the number of available cloud resources). To price the multi-resources system, we suppose a real option depends on some other variables such as the expected growth rate gcc_μ and the volatility respectively gcc_σ. Then if we let

$$dC3_i / C3_i = C3_\mu + C3_\sigma dz_i. \qquad (3.1.13)$$

for any number of assets (cloud resources) of C3 such as $(C3_1, C3_2, \cdots, C3_n)$ with prices p $(p_1, p_2, \cdots, p_n)$ respectively, we have:

$$d\ln S = dp_i / p_i = \mu_i dt + \sigma_i dz_i. \qquad (3.1.14)$$

where the variables $C3_i = \{\text{the set of resources}\}$. Applying the Option Pricing Factor ($p_o f$) for pricing options, we have:

$$d\ln S = [C3(t) - p_o f \ln S]dt + [\textit{stochatic term}]. \qquad (3.1.15)$$

where dz is the stochastic term. The strength of the $p_o f$ is determined by the value of its membership function (high for $p_o f > 0$). For a multi-asset problem, we have:

$$d\ln S_i = [C3_i(t) - p_o f \ln S_i]dt + \sigma_i dz_i \,|_{i=1,2,\cdots,n}. \qquad (3.1.16)$$

The value of $C3(t)$ is determined such that $F(t)=\hat{E}[S(t)]$ i.e., the expected value of S is equal to the future price, a scenario similar to what we may get is a user who suspects that he might need more compute cycles (bandwidth) in 3, 6, and 9 months from today and therefore decides to pay some amount, \$s upfront to hold a position for the expected increase. We illustrate this process using a 3 step trinomial process. If the spot price per bit of bandwidth is $\$s_T$ and the projected 3, 6, and 9 months future prices are $\$s_1, \$s_2$, and $\$s_3$ respectively. In this scenario, the two uncertainties are the amount of bandwidth that will be available and the price per bit. However, we can obtain an estimate for the stochastic process for bandwidth prices by substituting some reasonably assumed values of $p_o f$ and (e.g., $p_o f$ = 10%*;* = 20%) in Equation (3.1.15) and obtain the value of S from Equation (3.1.16). Suppose $V_{i,j}$ represents the option values at l for $l=0,1,2,\cdots,n-1$ level and j node for $j=1,2,3$ (for a trinomial lattice only); i.e., $V_{1,1}$ rep-resents the option value at level 1 and at p_u. Therefore, the displacement for the node is $V_{l,j}+\alpha_j$. If there are displacements $\alpha_{p_u}, \alpha_{p_m}$, and α_{pd} for p_u, p_m, and p_d respectively, the expected future price for bandwidth is given as:

$$\hat{E}[S(t)] = p_u e^{v_{l,j}} + p_m e^{v_{l,n}} + p_d e^{v_{l,n}}. \qquad (3.1.17)$$

where $l=0,1,2,\cdots,n-1$ and $j=1,2,3$.

3.2 Fuzzy Logic Framework

We express the value of the C3 flexibility opportunities as:

$$gcc: t_n = t_{ut}. \qquad (3.1.18)$$

where t_n denotes the time-dimension and given as $0 \le t_n \le 1$ and t_{ut} describes the corresponding utilization time. If $t_n = 0$, C3 usage is "now" or "today", if $t_n = 1$, C3 has a usage flexibility opportunity for "the future" where future is not to exceed 6 months (for example). Users often request and utilize C3 at extremely high computing power but only for a short time for $t_n = t_{ut} \approx 0$. Therefore, disbursing the C3 on-demand and satisfying users' quality of service (QoS) requires that the distributed resources be over-committed or under-committed for $t_n = 1$ or 0) respectively in order to satisfy the conditions specified in the service level agreements (SLAs) document. Such extreme conditions (for example, holding C3 over a long time) re-quires some cost in the form of storage. Therefore, we express utilization time t_n as a membership function of a fuzzy set T. A fuzzy set is defined (see for example [13]) as:

$$T = (t, \mu(t)) \mid t \in T, \mu_T(t) \in [0,1]. \tag{3.1.19}$$

Thus, given that T is a fuzzy set in a time domain (the time-dimensional space), then $\mu_T(t_n)$ is called the membership function of the fuzzy set T which specifies the degree of membership (between 0 and 1) to which t_n belongs to the fuzzy set T. We express the triangular fuzzy membership function as follows:

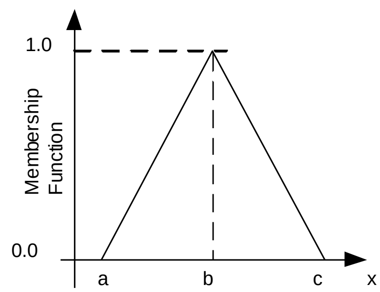

Figure 3: Triangular Fuzzy Membership Function for C3 Utilization Time.

$$\mu_T(t) = \begin{cases} 1 & \text{for } x = b \\ \dfrac{x-a}{b-c} & \text{for } a \le x \le b \\ \dfrac{c-x}{c-b} & \text{for } b \le x \le c \\ 0 & \text{otherwise i.e., } x \notin [a,c] \end{cases} . \tag{3.1.20}$$

where [a, c] is called the universe of discourse or the entire life of the option. Therefore, for every C3 at utilization time t_n, availability of the C3 expressed as member-ship function is the value compared to stated QoS conditions given in the SLA document. Figure 3 shows the tri-angular fuzzy membership function for the cloud resources utilization corresponding to Equation (20). A Service Level Agreement [34] (SLA) document is an agreement between a service provider and a service consumer related to the service level (quality of service). Such an agreement can be reached by signing a formal and legally binding contract, or informally in case of different departments of a company using the services. This is referred to as an operation level agreement (OLA). In terms of quality, the SLA implies a mutual agreement with respect to security, priorities, responsibilities, guarantees, and billing modalities. In addition, the SLA specifies metrics such as availability, throughput, response times, and others. By nature, SLAs always consider the output side, i.e. they are drafted from the service consumers' perspective. The implication of a service constraint that guarantees QoS and meets the specified SLA conditions within a set of intermittently available C3 is a system that compromises the basic underlying design objective of the cloud as a commercial computing service resource [35]. Therefore, Equation (3.1.18) becomes:

$$C3: t_{ut} = t_n \,|_{QoS \approx SLA} \,. \tag{3.1.21}$$

To satisfy QoS-SLA requirements, we apply a real options pricing scheme, which differs from a generic market-based resource sharing where all jobs are expected to receive some resource [36] based on the offered price or the application of demand and supply to set prices. Using the demand and supply economic market principles, the price of cloud resource tend to be higher than affordable. The reason is that since the resources exist as compute cycles and this makes availability hard to guarantee, supply could be low (most often) and low supply raises price where more us-age is expected. Hence economic market principles that are guided by demand and supply (equilibrium price) do not sufficiently support the general structure for pricing cloud commodities.

4 Pricing Infrastructure

Figure 4 shows our abstract representation of a cloud pricing infrastructure. Basically, our pricing infrastructure consists of four layers; cloud services layer, middleware layer, pricing and usage optimization layer, and the Internet layer. The cloud services layer houses the virtual cloud resources. The middleware layer integrates the pricing to the user applications that runs/utilizes the cloud resources. The Internet layer offers a user interface for querying cloud resources. Specific services available in the middleware layer include Resource Modeling (RM). The RM provides a description of the available resources, application capabilities, resource discovery, provisioning and defines inter-component relationships between the various clusters that comprise the cloud. Other functionalities of the cloud service layer include advanced monitoring and notification. The updates include notifications for changes in projected utilization levels and application notification regarding services changes. The price optimization layer executes the process of accounting and auditing and actually applies our pricing scheme to charge resource utilization. At this layer, services are price-based and they range from buying cloud compute commodities, such as bandwidth, processor cycles or memory to retailing computes time. The functions executed at the price optimization layer are user-application based. At this layer, several authenticated users log in to the cloud to assess compute commodities. The objective of a logged-on user is to gain access to the computing commodities as soon and quickly as possible for a small price while occupying the highest level of QoS as defined in the SLA. To achieve this objective, the Cloud Resources Broker (CRB) maps physical resources (requirements) onto virtual resources (the C3) while guaranteeing a service agreement between the QoS and SLA.

5 Results and Discussions

We start with analysis of the Amazon EC2 trace. Figure 5 shows the projected instance for 2011 (01 January 2011 up until 31 December 2011).

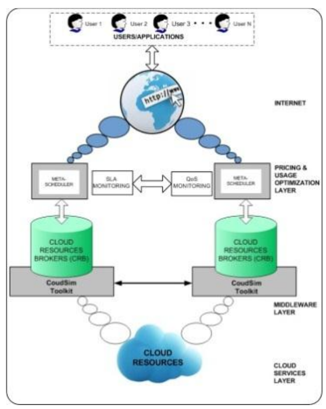

Figure 4: Cloud Resources Pricing Infrastructure.

During these timeline, Amazon EC2 was expected to have sufficiently scaled at least 64 GB independent instances possible to support up to 13 million jobs. However, Statistics from Figure 5 shows that a direct mapping of the projected statistics does not scale to realistic values. As seen from Figure 5, the actual capacity barely supports 5 million jobs with 16 GB of instances at one given time. This variation observed in the instance trace depicts that static charges in the cloud computing system does not favor the user. The user would not have to worry about future cloud resource usage if he paid an upfront value (i.e., if he entered into a usage agreement). To circumvent Amazon's problem (of applying static

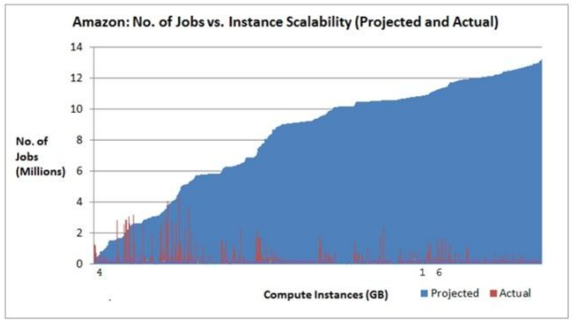

Figure 5: Amazon: Scalable Instances Vs. Number of Jobs.

charges), we introduced artificial spot prices for the C3 at various times of the contract period as exemplified by the trinomial tree structure of the solution space. As we mentioned earlier, from the date of signing the contract to the actual date of utilization (for European style option it is at maturity; and for American style option it is any time be-fore maturity) the price variation is due to various factors such as change in the demand on the cloud resources and change in technology. Based on these changed prices of the cloud resource commodities (in other words, the underlying assets for the option) the option values are computed using our model in Equation (3.1.10).

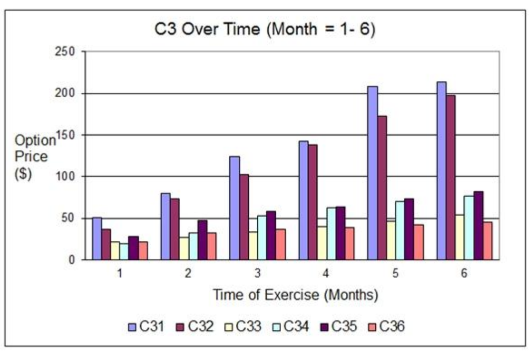

Figure 6: Effects of Time of Exercise on Cloud Compute Cycles: CPU Cycles ($C3_1$), Bandwidth ($C3_2$), Memory ($C3_3$), Throughput ($C3_4$), Disks ($C3_5$), Processors ($C3_6$).

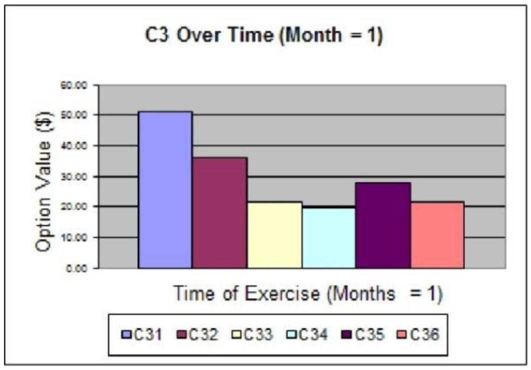

Figure 7: Effects of Time of Exercise on Cloud Compute Cycles in the First Month.

Further in our experiments, we simulate (run trinomial on CloudSim) the cloud compute commodities and obtain option values (prices) and study the variation in a space of 6 months to determine the effects of time of exercise on option value. Time of exercise here means the time at which the cloud compute commodities are going to be utilized, up to six months in the future. Figure 6 shows the effects of time of exercise on C3. The time of exercise (months) shows the time the user may wish to exercise the options.

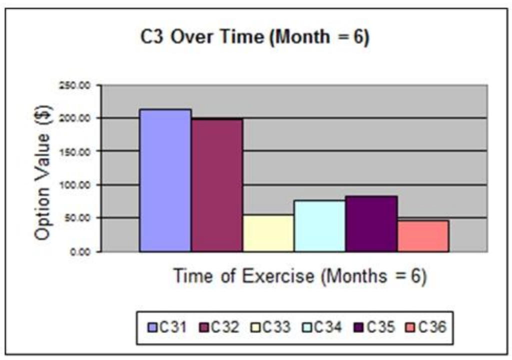

Figure 8: Effects of Time of Exercise on Cloud Compute Cycles in the Sixth Month.

In the Figure, 1 month means the first after signing the con-tract, and 2 month means the second month after signing the contract etc. We isolate the first month and the sixth months in Figure 7 and Figure 8 respectively for ease of comparison. We study some selected instances of C3 such as CPU cycles, memory, bandwidth, throughput, disk capacity, and processors. The effects of applying Option Pricing Factor $p_o f = (p_o f)^{-1}$ shows that at any given time, the cloud satisfies the users' computing needs by granting computing requests at lower prices during off-peak demands for C3 so that users take advantage of the low prices and use more of the C3. The resource prices are increased during peak demand (to make more profits) period to provide all the avail-able resources at its full without compromising on the QoS. Changing the value of price varying factor $p_o f$ statically amounts to early exercise of an American option when a favorable situation arises for the user. This also means that the provider can benefit for certain values of $p_o f$ in which case the contract holder will not exercise early. That is, provider can execute the jobs of users willing to pay higher prices for the resources. Therefore, the original contract holder still has the time value on his/her option to exercise at a later date. This implies that both the user and the provider are availing the best opportunities for their benefits. In other words, the option pricing factor $p_o f$ helps in achieving the quasi-static equilibrium between the quality of service that the user requires and the profit level that the service provider would expect. Since the value of $p_o f$ is changed for a given experiment it is not dynamic. Changing the value of $p_o f$ dynamically is bit complicated and we leave that that issue for a future work. Figures 9 and 10 shows our results for the simulation of the American put and American call option. In these simulations, we show the effects of volatility on the option price. Figures 9 and 10 also demonstrate the effects of time of exercise on C3 computed from our model.

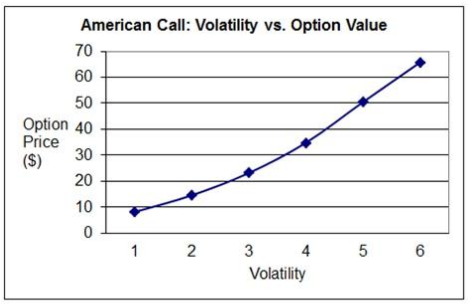

Figure 9: Effects of Volatility on Option Value (American Call).

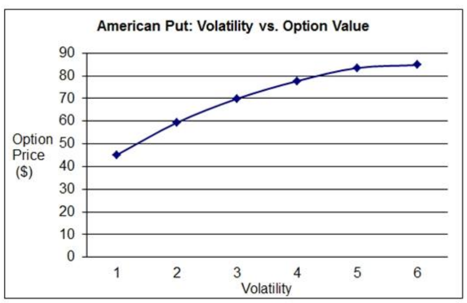

Figure 10: Effects of Volatility on Option Value (American Put).

In our experiments, we run our developed trinomial lattice on American put option and American call option using some hypothetical values for $K = 100.0, S = 100.0, T = 1,2,\cdots,6$ (months), $r = 0.06, N3$ (for trinomial), $\sigma = 0.2, dx = 0.2, N_j = N$. We vary σ in steps of 0.1 i.e., $\sigma = 0.0,0.1,\cdots 0.7$. We obtain option values (prices) and study the variation in a space of 6 months to determine the effects of fluctuations (i.e., volatility) value of the option computed and the time of exercise on option without $p_o f$. We also compute option value for individual resources while applying $p_o f$ in the trinomial lattice. As an example, we simulate Random Access Memory (RAM), one of the C3-s and monitor users' request for utilization. For a call option, we simulate the effects of time on exercising the option to use RAM. We use the following parameters: $S = \$6.941.00 * 10^{-7}, T = 0.5, r = 0.06, N = 4,8,16,24, \sigma = 0.2$ and

$N_j = 2N + 1$.The price are base values that reflects existing cloud market value. Figure 11 shows the option value for RAM. The computed option values from our experiments are reasonably low com-pared to the static and fixed charges in Table 3 or the charges for cloud instances in Table 1 and Table 2. The option value reaches a steady state as the number of step sizes increases. These option values take into account a balance between the QoS required by the users and a marginal profit level for the cloud service providers. We observe that as we increase the number of time step to 24 and above, the option value starts to maintain a steady value. We stop at this stage which indicates that we reach a value where the user can obtain satisfaction.

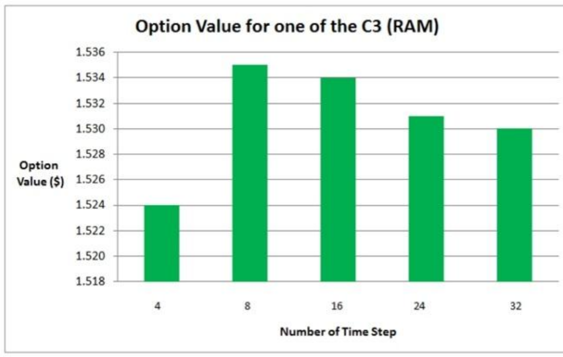

Figure 11: Amazon: Option Value for RAM Computed with p_of varied between 0:0; ; 1:0.

6 Conclusion

In this paper we presented a proof of concept of the application of financial option for pricing instances of cloud resources. Our results have been validated by cloud market simulation middleware (cloudbus) using real trace data. A future line of work will focus on testing our proposed model in real cloud computing market environments.

We have only examined the utilization traces from Amazon and Joyent for the sake of comparison in this paper. From the utilization trends in Joyent, the generic cloud problem – satisfying diverse and multiple requests and guaranteeing resources (non-storable reservoir of compute commodities) availability - persists. We designed and developed a pricing model that meets the users' satisfaction guarantee with profitable outcome to the provider. We proved this with two real cloud nodes. Therefore, this paper has both academic and industrial value/contributions. Cloud economy, as a new emerging research area, presents several issues to be considered by a service provider as well as a user of cloud computing commodities – provision of virtual cloud resources that are in much demand and this meets expected level of user QoS. For the industry, our model and results would provide a novel approach for assessing the profitability of the cloud and would demonstrate the need to manage the cloud infrastructure in order to meet the peak load demand for cloud resources.

We have assumed that the user will use only one of C3 at any time. However, in reality, a user would request for multiple commodities for their jobs – for example, while asking for compute cycles they would also ask for storage, memory, bandwidth and possibly some databases and soft-ware. This involves a combination of several instances and pricing them applies multi-asset financial pricing approach. A multi-asset pricing problem make the option to have multiple underlying asset. This paper can be extended for experimentation on such options with multiple underlying as-set. This is a very difficult problem. As a next step also, work is currently underway to integrate the model with automated resource managers for developing a smooth service and pricing system.

Acknowledgement

We thank the officials at the Joyent and Amazon for granting us permission to use their web resources. The first and third authors acknowledge the partial financial supports from the University Research Grants Program (URGP) of University of Manitoba and the Natural Sciences and Engineering Research Council (NSERC) of Canada.

References

[1] D. Werthimer, J. Cobb, M. Lebofsky, D. Anderson, and E. Korpela. SETI@HOME-Massively Distributed Computing for SETI. *Computer Science Engineering*, 3(1):78–83, 2001.

[2] LCG-LHC. Large Hadron Collider (LHC), May 2008.

[3] I. Foster, Y. Zhao, I. Raicu, and S. Lu. Cloud Computing and Grid Computing 360-Degree Compared. In *Grid Computing Environments Workshop (GCE'08)*, pages 1–10, 2008.

[4] D. Allenotor. *A Fuzzy Real Option Model for Pricing Grid Compute Resources*. PhD Thesis, University of Manitoba, 2010.

[5] W. Itani, A. Kayssi, and A. Chehab. Privacy as a Service: Privacy-Aware Data Storage and Processing in Cloud Computing Architectures. In *Proceedings of the 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing*, DASC '09, pages 711–716, Washington, DC, USA, 2009. IEEE Computer Society.

[6] M. R. Tribhuwan, V. A. Bhuyar, and S. Pirzade. Ensuring Data Storage Security in Cloud Computing through Two-Way Handshake Based on Token Management. In *Proceedings of the 2010 International Conference on Advances in Recent Technologies in Communication and Computing*, ARTCOM '10, pages 386–389, Washington, DC, USA, 2010. IEEE Computer Society.

[7] D. Velev and P. Zlateva. Cloud Infrastructure Security. In *Proceedings of the 2010 IFIP WG 11.4 international conference on Open research problems in network security*, iNet-Sec'10, pages 140–148, Berlin, Heidelberg, 2011. Springer-Verlag.

[8] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li. Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing. *IEEE Trans. Parallel Distrib. Syst.*, 22:847–859, May 2011.

[9] P. Marshall, K. Keahey, and T. Freeman. Improving Utilization of Infrastructure Clouds. In *Proceedings of the 2011 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, CCGRID ’11, pages 205–214, Washington, DC, USA, 2011. IEEE Computer Society.

[10] G. Reese. *Cloud Application Architectures: Building Applications and Infrastructure in the Cloud*. O’Reilly Media, Inc., 2009.

[11] A. V. Gerbessiotis. Architecture independent parallel binomial tree option price valuations. *Parallel Computing*, 30(2), 2004.

[12] J. Hull. *Options, Futures, and Other Derivatives*. Prentice Hall, Sixth Edition, 2006.

[13] G. Bojadziew and M. Bojadziew. *Fuzzy Logic for Business, Finance, and Management Modeling*. World Scientific Press, 2nd Edition, 1997.

[14] F. Black and M. Scholes. The Pricing of Options and Corporate Liabilities. *Journal of Political Economy*, 81(3), 1973.

[15] D. Allenotor and R. K. Thulasiram. Grid Resources Pricing: A Novel Financial Option Based Quality of Service-profit Quasi-static Equilibrium Model. In *Proceedings of the 2008 9th IEEE/ACM International Conference on Grid Computing*, GRID ’08, pages 75–84, Washington, DC, USA, 2008. IEEE Computer Society.

[16] M. R. Rahman. Risk Aware Resource Allocation for Clouds. Technical report, University of Illinois at Urbana-Champaign, 2011.

[17] M. Palankar, A. Onibokun, A. Iamnitchi, and M. Ripeanu. Amazon S3 for Science Grids: a Viable Solution? In *Poster: 4th USENIX Symposium on Networked Systems Design and Implementation (NSDI’07)*, April 2007.

[18] Amazon. Amazon Elastic Compute Cloud (Amazon EC2), January 2012.

[19] GRIA. DataGrid Accounting System (DAGAS), July 2009.

[20] R. Buyya, M. Murshed, and D. Abramson. GridSim: A Toolkit for the Modeling and Simulation of Distributed Re-source Management and Scheduling for Grid Computing. In *Journal of Concurrency and Computation: Practice and Experience (CCPE) Issue 13-15, Wiley Press, Nov.– Dec.*, volume 14, pages 1175–1220, 2002.

[21] A. Legrand, L. Marchal, and H. Casanova. Scheduling Distributed Applications: The SimGrid Simulation Framework. In *CCGRID ’03: Proceedings of the 3rd International Symposium on Cluster Computing and the Grid*, page 138, 2003.

[22] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya. CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms. *Softw. Pract. Exper.*, 41:23–50, January 2011.

[23] AppNexus. Appnexus, January 2012.

[24] GoGrid. Cloud hosting: Instant windows and linux cloud servers, January 2012.

[25] Joyent. Accelerator, January 2012.

[26] Google. App engine, January 2012.

[27] Microsoft Corporation. Azure services platform, January 2012.

[28] 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*, pages 1–6, New York, NY, USA, 2007. ACM.

[29] X. Zhang and Y. Dong. Optimizing Xen VMM Based on IntelR Virtualization Technology. In *ICICSE '08: Proceedings of the 2008 International Conference on Internet Computing in Science and Engineering*, pages 367–374, Washington, DC, USA, 2008. IEEE Computer Society.

[30] F. Black and M. Scholes. The Pricing of Options and Corporate Liabilities. *Journal of Political Economy*, 81(3), 1973.

[31] J. C. Cox, S. Ross, and M. Rubinstein. Option Pricing: A Simplified Approach. *Journal of Financial Economics*, 7(7), 1979.

[32] R. K. Thulasiram, L. Litov, H. Nojumi, C. T. Downing, and G. R. Gao. Multithreaded Algorithms for Pricing a Class of Complex Options. In *Proceedings (CD-ROM) of the International Parallel and Distributed Processing Symposium (IPDPS), San Francisco, CA*, 2001.

[33] P. P. Boyle. Option Valuing Using a Three Jump Process. *International Options Journal*, 3(2), 1986.

[34] S. Pantry and P. Griffiths. *The Complete Guide to Preparing and Implementary Service Level Agreements*. Library Association Publishing, First Edition, 1997.

[35] C. S. Yeo and R. Buyya. Integrated Risk Analysis for a Commercial Computing Service. In *Proceedings of the 21st IEEE International Parallel and Distributed Processing Symposium (IPDPS 2007, IEEE CS Press, Los Alamitos, CA, USA)*, 2007.

[36] R. Wolski, J. Plank, J. Brevik, and T. Bryan. Analyzing Market-Based Resource Allocation Strategies for the Computational Grid. *Int. Journal of High Performance Computing Applications*, 15(3):285–281, 2001.