Journal of the Nigerian Association of Mathematical Physics
Volume 25, No. 2 **(November, 2013), pp 55 – 64**
© J. of NAMP

A Financial Option Based Model for Pricing Multiple Grid Compute Cycles

[1]David Allenotor and [2]Ruppa K. Thulasiram

[1]Dept. of Computer Science
University of Benin, PMB 1154 Benin City, Nigeria.
[2]Dept. of Computer Science
University of Manitoba, R3T 2N2 MB, Canada.

Abstract

A large corpus of existing literature in grid computing shows the presence of higher concentration of research efforts in grid middleware framework development, grid resource scheduling, grid economy, and grid security. An important aspect of grid computing that is missing is the pricing of grid resources across multiple grids. Existing research efforts in grid computing that priced grid resources, priced only a single grid resource in one grid. However, in a real life scenario, subscribed grid resource users require more than one resource (which are not available in one grid at all time) to complete a computationally intensive job. Two characteristic challenges of the computational grid that make pricing the resources across multiple grids a hard problem are grid resources availability is transient since they only exists as non-storable grid compute cycles (gcc) and grid resources are distributed geographically across dissimilar organizations with diverse resources usage polices. Therefore, a model that price grid resource across multiple grids must guarantee resources availability measured as Quality of Service (QoS). This paper is positioned to present a novel design of a model to price grid resources across multiple grids using the financial option theory from a real option perspective and value the grid resources by treating them as real assets. We integrate our financial option pricing algorithm into GridSim Toolkit Simulator to simulate our grid environment.

Keywords: Option Pricing; Trinomial Tree; Computational Finance; Pricing Factor; Compute Cycles.

1.0 Introduction/Background

Like the electrical power grid, a computational grid (or simply the grid) is defined [1] as a software and hardware infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities to its subscribers. Over the years, the upsurge in advances in Information Technology (IT) and the applications of resource-intensive computation have grown beyond the capacity for locally affordable compute resources. The Search for Extraterrestrial Intelligence (SETI)@Home [2] project is one example where computational capacity exceeds the locally available compute power. To support it's locally available compute power, SETI@home scavenge CPU cycles or computational resources from volunteers in order to support high performance computation. Similarly, several businesses now prefer to use the grid resources instead of the regular updates of their hardware and software infrastructure. Business owners that subscribe to use gird resources have advantage over those who provide in-house computation power – they save on hardware and software licensing, they save on runtime maintenance of infrastructures, they also save on staffing needs. However, grid resource utilization comes with challenges which have been largely researched in existing literatures. Some of these include middleware framework development [3], grid resources scheduling and resource management [4], and security issues [5]. Another grid challenge is the resource availability constraint. Grid resources are not storable and their use is limited by uncertainty in their availability hence, resource pricing is a concern. Currently, there is no cost for utilizing grid resources especially for research purposes. The grid resources are available for free because of the government funding in establishing grid infrastructures. However, due to the large interest in grid for public computing, a development in the use of

---

Corresponding author: ***David Allenotor,*** E-mail: dallen@uniben.edu, Tel.: +2347056009584

rented services in the grid started recently. Amazon, for example, introduced a Simple Storage Service S3 [6] system and the Elastic Compute Cloud (EC2) [7] for grid users. Amazon's S3 aims at providing data-intensive and low cost data storage system. They provide a benchmark study for data durability, data availability, access performance, and file download via BitTorrent. 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. Requirement for flexibility in grid resource usage is seen from a spectrum of users' choice to determine resource usage time. Such choices include the decision to use the grid resources (for example, in a project, the decision to use computes resources) at a time in the present or at sometime in the future. The decision on resource usage between current time and future time is hard to determine using traditional methods such as Net Preset Value (NPV) or Discounted Cash Flow (DCF) without loosing the realistic value of the decision [8]. The reason why the traditional schemes fail to capture the project uncertainties is because project uncertainties are often unnoticeable during the early stages of the project development.

Generally, we characterize the grid resource availability constraint based on the flexibility of resource use. We consider three important aspects of the grid resource characteristics: (i) Grid resource availability is transient and usage is based on availability (ii) to price them, we treat the grid resources as gccs by applying financial option theory and determine best exercise time for the use of the resources, and (iii) we apply the real option to capture uncertainties that abound in making the decision to exercise the option.

A financial option is defined (see, for example [9]) 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. On the other hand, a *put option* grants the holder the right to sell the underlying asset at the specified strike price. An *American option* can be exercised any time during the life of the option contract; a *European option* can only be exercised at expiry. Options are derivative securities because their value is a derived function from the price of some underlying asset on which the option is written. They are risky securities because the price of their underlying asset at any future time cannot be predicted with certainty. This means the option holder has no assurance the option will bring profit before expiry. 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 investment decision. For this study, these alternatives include the flexibilities of exercising, deferring, finding other alternatives, waiting or abandoning an option contract. In [10], we designed a financial option model and proposed a justification for price adjustments in [11]. Further study [12] shows that without unnecessarily charging for grid resource usage, we can satisfy the user in terms of the Quality of Service (QoS) and the providers' incentive (profitability) by using a global base prices and a control factor $p_o f$. A single grid resource was assumed in [10], [11], and [12]. However, in real world scenario, grid resource users often require more than one resource. The request could span across multiple grids. This paper evaluates our financial option based model presented in [11] using the GridSim [13] grid simulator. The novelty of this paper over our previous efforts in [11] is the extension to price multiple resources across multiple grids.

The rest of this paper is structured as follows. In Section 2 we review related work. Section 3 descriptions the GridSim toolkit simulator, the middleware, and provide implementation of our model using the GridSim. In Section 4 we present the mapping between GridSim toolkit simulator and grid the GridSim framework model as a mapping between our mathematical model and the GridSim toolkit simulator. Section 5 we describe the simulation environment and the discussion of results of our experiments. Finally, Section 6 concludes this paper with a discussion of future work.

2.0 Related Work

A large body of research efforts in grid middleware framework, resource allocation, resource scheduling, and grid economy is relevant to this study. In general, much attention has been focused on resource distribution using market-based and resource scheduling allocation concepts [14]. Tycoon [14] provided a solution for proportional share and allowed users to differentiate the value of their jobs. A major limitation of the scheme is in the time spent in handling the auction mechanism. Since the unsuccessful bids takes some of the resources including communication and computing. In a like manner, Stuer et al. [15] focused on user-based service provisioning for the grid using economic forces of demand and supply. Their idea about a grid equilibrium price is based on the interaction between demand and supply where a provider allocate a resource according to a user if the average past price of that resource is less than the current asking price. In this scheme, providers may take undue advantage of a higher demand and make grid services less accessible to users. Other research efforts focus on contingent bids in auctions [16], and resources pricing [17], [18], and [19]. Researchers under these fora have followed two distinct approaches; (a) they extended existing standardized grid middleware (b) present some novel work with focus on grid economy with specific references to resource share and resource management. Bhargava and Sundaresan in [16] model a computing utility and examine the possibility to extend the pay-as-you-go pricing scheme using the auction system. Zhang et al. [19] develop a pricing idea using real option Monte Carlo simulation. They focus on resource management which is not only about scheduling large and compute-intensive applications (or resources), or some form of advanced reservations but the manner of putting compute resources to work for the benefit of the user and owner. However, this scheme did not consider the effects of critical technological changes that are capable of determining users' patronage to the grid. In [20], we provided a pricing framework for a single asset. However, in a real world scenario, a user may want to price many resources across multiple platforms.

Chunlin and Layuan [21] presented an optimization-based resources pricing algorithm that focus on increasing the grid providers' effective gain and Sulistio et al. [22] evaluate the effectiveness of grid revenue management using resource reservations. These studies show that charging user with differentiated prices will increase the effective total revenue for the resource in question. They also showed that their scheme guarantees a fair share of the resources applications with highest computing priorities. The focus of the study given in [17], [18], and [22] is on resource sharing and resource scheduling with references to market economy. In our recent paper [11], we focused on balancing the grid profits as seen from the perspectives of the grid resources provider. In [12], using traces from real grids, we enhanced our model and applied our base prices. The highlighted research efforts [16], [17], [18], [19], and [21] have common goals; the application of economic principles to decide a fair share of grid resources and uses the principles of resources redistribution and scheduling. Currently, there is no charge for using grid resources. However, a trend is developing because of large interest in grid for public computing and this could lead to a sudden explosion of grid use in near future. In this study, we focus on the evaluation of our model that applies financial option theory to price grid resources using the GridSim toolkit simulator. The objective is to keep the grid busy by attracting more patronage for the provider's incentive and to satisfy the user's QoS.

3.0 GridSim Toolkit Simulator

The GridSim uses four auction types [13] to make reservations; first-price sealed bid auction, continuous double auction, Dutch auction, and English auction (and their respective reverses). In a first-price sealed bid auction, the auctioneer sends call for bids and the bidders send their bids in a closed form. The auctioneer then waits a given time for the bids and then allocates the resource to the bidder who values the resource highest. On the other hand, the continuous double auction asks the bidders and sellers continually to submit bids and asks to the auctioneer. The auctioneer waits when a match is found and then informs the auctioneer and the respective bidder and the seller. The English auction and the Dutch auction type behave opposite. In an English auction, the auctioneer starts with a low price and then gradually increases the price until one bidder remains and this last bidder wins the auction. In the Dutch, the auctioneer starts with a high price and gradually decreases the price until a bidder decides to bid. The first bidder willing to accept the price wins the auction. Since the objective of our pricing model is to attract more patronage to the grid by lowering the grid resource prices, we disburse resources to the largest group as we offer the resources for use.

Simulation techniques are often used when it is more cost-intensive to use a real system for evaluation. We select the GridSim toolkit for the simulation of our financial option pricing model because of its flexibility in modeling the grid environments. The GridSim [13] is a general purpose discrete event simulation toolkit that is based on SimJava2. Figure 1 shows the six layered architecture of the GridSim. It is a toolkit that administrates time-variable resource assignments. The first layer (at the bottom of Figure 1) consist Java Virtual Machine (JVM). The JVM manages events and components interaction in the GridSim. The second layer consists of the infrastructure components such as network and resource hardware. This layer also enables the design and integration of user interfaces. The third and fourth layers are responsible for the simulation and modeling of computational grid entities. Simulation of the grid resource brokers takes place in the fifth layer. The top layer consists of the grid scenario, user requirements, I/O interface, application configuration, and user customized codes.

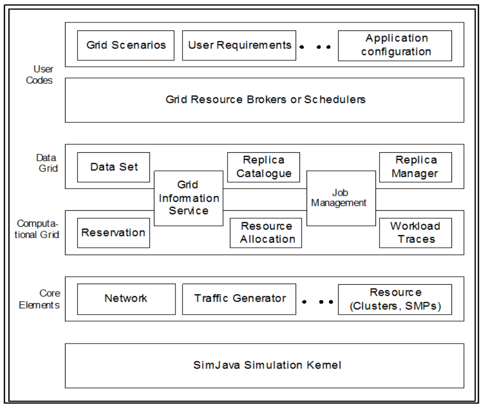

Figure 1: GridSim Toolkit Layered Architecture [13]

Grid resources are non-storable compute commodities. Some of the grid compute cycles include CPU cycles, memory, network bandwidths, throughput, computing power, disks, processor, and various measurements and instrumentation tools. The GridSim toolkit only administrates time-variable resource assignments and it is not suitable for time-constant assignment. Examples of the time-constant resources are RAM, network bandwidths, throughput, computing power, disks, processor, and various measurements and instrumentation tools. For example, bandwidth parameter is a parameter of both a grid and a user which determines the connection speed to the grid and between grid machines.

4.0 Pricing Grid Resources Using GridSim

Several schemes exist in the literature to price financial options. Some of these approaches include (i) application of the Black-Scholes (BS) model [23]. The BS model requires solution of a partial differential equation which captures the price movements continuously; (ii) application of a discrete time and state binomial model of the underlying asset price that captures the price movement discretely [24]. In our simulation, we use the trinomial lattice [9], [25] to solve the real option pricing problem. This is a discrete time approach to calculate the discounted expectations in a trinomial-tree.

Our model objective is to keep the grids busy. In order words, we avoid idle compute cycles by lowering the resource prices. To achieve this objective, we define and apply a price variant factor $(p_o f)$. The $p_o f$ is a multiplier and a real number given as $0.1 \le p_o f \le 1.0$. The value of $p_o f$ 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 changes is captured using crisp values. We set the $p_o f$ as an index for offering prices to grid resource users. For example, given a base price of $0.009 a week for 1 MHz processor, we set the price indices as such that the maximum does not exceed our set base price.

4.1 Grid Resource Matrix (GRM) in GridSim

In our simulation, we set a grid resource base price to a range of [2.00..$3.00] per CPU cycles/week. We assume that the price of one type of grid resource is same in all grids. This assumption gives all grids a fair share to offer resources. For example, if the base price for 1MHz CPU cycle cost $2.0 per week per year in one grid, this price will be same across all grids. To price the grid resources using the GridSim toolkit, we distribute the available resources among users using the Dutch action. The Dutch auctions have been used in on-line auctions, particularly for commodity-type markets. It begins with an initial price that is continuously lowered in a pre-specified bidding interval (bidding round – a bidding round is when every resource is offered to every user for every job) until a trader place a bid. The danger of using the Dutch auction is that a malicious user could take advantage of the continuous lower price and pool all jobs to a resource when the price is cheaper. In our particular approach, we use the reversed Dutch auction to discourage the malicious user intent. The reversed Dutch auction begins at a base price within in a pre-specified contract time until a user place a bid. The value of placed bid is then used to set the offering price. We map the reverse Dutch auction into the trinomial lattice to obtain option values for individual user resources. In our approach, for each job $(j_1, j_2, \cdots, j_N)$, the user $(u_1, u_2, \cdots, u_k)$ select the best resource and choose whether to exercise the option to use it or not. If a user fail to make a bid (i.e., select the best a resource at the offered base price), the user will not be included in the selection of the next-best resources. The user waits for a new bidding round when every other job is assigned. At the end of several bidding rounds (δt), the first (users) would get the resource at the price paid by the last of user. We assume that the users' behavior is normal they do not speculate cheaper prices.

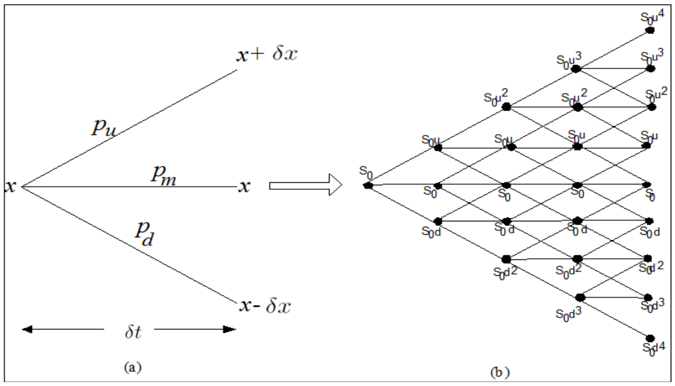

Figure 2: Trinomial Lattice

4.2 Trinomial Tree with Reverse Dutch auction

We apply the trinomial-tree model [9] to price mainly American-style and European-style options on a single underlying asset. To compute option prices, we build a discrete time and state binomial model of the asset price and then apply discounted expectations [9]. Figure 2(a) shows a trinomial tree were 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 [9] is given by $dS = (r - \delta)Sdt - \sigma Sdz$. Consider a trinomial model of asset price in a small interval δt, we set the resource price changes by δx . To map the reverse Dutch auction into the trinomial lattice, consider the interval between bidding round δt the price 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. Figure 2(a) shows a one-step trinomial tree expressed by δx and δt and Figure 2(b) shows the corresponding four-step trinomial tree.

$$\left(cc_1 cc_2 \cdots cc_m\right) * \begin{pmatrix} Pcc_1{}^{g_1} & Pcc_1{}^{g_2} & \cdots & \mathrm{P}cc_1{}^{g_n} \\ Pcc_2{}^{g_1} & Pcc_2{}^{g_2} & \cdots & \mathrm{P}cc_2{}^{g_n} \\ \vdots & \vdots & \ddots & \vdots \\ Pcc_m{}^{g_1} & Pcc_m{}^{g_2} & \cdots & \mathrm{P}cc_m{}^{g_n} \end{pmatrix} \qquad (1)$$

For several gccs, the amount requested for various commodities is presented as a matrix, the price of such commodities as a vector is shown in Equation (1). Various requests are seen as number of rows in the matrix which is also the number of grids $(g_i \forall_i = 1,2,\cdots,m)$. The resulting price from the GRM is the corresponding input to the trinomial lattice (see Figure 2) to compute option value for the resource utilization.

5.0 GridSim Environment and Experimental Setup

In our simulation, we specify the number of users that need to use resources, the number of grids, and number of jobs. We use GridSim toolkit 5.2.4 to provide resource scheduling. We schedule the grid resources based on the reversed Dutch auction where each round is seen as the time to expiration in a trinomial tree (see Figure 2(a)). The trinomial-reversed Dutch auction provides price information about the winners of the bid. We use the resultant winning price to compute option value in the trinomial tree model. The setup grid consists of several jobs, each of which provides resources for the users. Computational resources (gccs) are seen as compute cycles which run jobs in the grid. We run our simulation using eclipse and Windows 8 Service Pack 1. The computer specifications that was used for this simulation are as follows: 3.6 GHz Dual-core Intel Corei7, 8GB RAM, 750GB HD. Additionally, we also used Cloud Garden's Jigloo Standard Widget Toolkit (SWT)/swing Graphic User Interface (GUI) Builder for Eclipse and Web Sphere to reduce the development time and enhance coding. Two important classes in the simulation are the broker and the gridSimPricing. The broker class receives jobs from the user, and behaves as an auctioneer. It then implements the reversed Dutch auction (where jobs are allocated to the resource provider who makes the bids) and submits the jobs to the gridSimPricing class after reversed Dutch auction clears. The simulation is initialized in a configuration setup in five steps: (i) Users setup – the user setup specifies the user, user budget (amount expected to pay for computation) and the bidding policy. The bidding policy could be set for cost or optimal time for the user. We specify the cost optimal bidding policy for all users since our objective is to provide resources at a lower affordable price to the users, (ii) Job type – the type of job is specified for all users as job name, length of time (MIs), file input size, required RAM, and disk size (MB), (iii) Machine name – the machine name specifies the individual name of the machines available for computation, (iv) Simulation parameters – we use the simulation parameters (two variables) to speculate the simulation time and expected delays. the expected simulation time specified is expected to exceed the actual simulation time with minimal delays [13], and (v) Trinomial setup – the trinomial setup consists of strike price, time (years), interest rate, number of time step, and volatility. After setting up the simulation, users bid to get resources using the reverse Dutch auction. The reverse Dutch auction allocates the jobs based on the users' requirements (given in i-v). Figure 3 shows the resources (R_i), machines (M_i), and jobs (J_i) distribution in the simulation. We created three grids g_1, g_2, and g_3; three users u_1, u_2, and u_3.

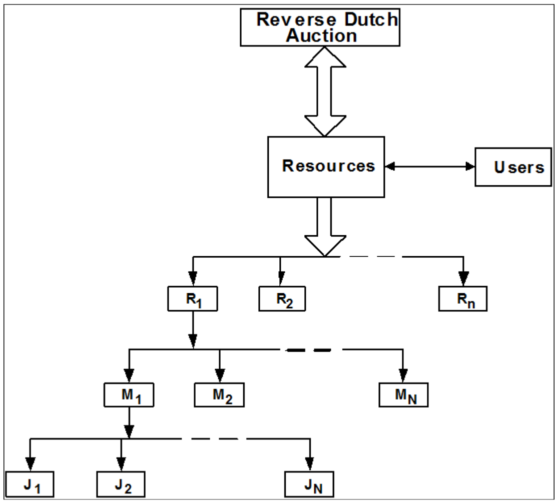

Figure 3: Experimental Grid Setup

We let the users provide specifications for their jobs. The reversed Dutch auction allocates the jobs based the users' needs.

5.1 Handling Time-Constant Resources in GridSim

The GridSim simulation toolkit administrates time variable resource assignments. Some factors affect the overall simulation. These include speed, which depends on the characteristics of the resource (high performance or low performance), time taken to execute a task, which depends on how busy the resource is and the bandwidth, that influences the whole simulation. Any user who wants the offered resource must book the resource for the expected duration of his task. A booked resource becomes unavailable to users until the current task is completed. After every user decided to put their job into some resource (or wait without doing it because the user thinks they are expensive), the resources that were not booked in the current round get a new offer round, with a discount price (lowered by a factor of $p_o f$). Users get another chance to book them. This is repeated until the offered price is lower than a 10% of the base price. The bidding is complete when every booked resource executes its task. The process is repeated if there are unfinished jobs. Our idea is that whatever a user uses for computation, the grid is seen as a task oriented architecture in a like manner as the GridSim toolkit. The executed tasks would actually be resources like HD and RAM, which would affect the amount of instructions of that task (for example, occupying 1MB of RAM implies reading and writing 1MB of RAM at least once. This could be interpreted as 2MB instructions (assuming RAM read/writing was an instant operation), or more 4MB instructions (using the empirical assumption that a read or write cycle takes the same time as 2 CPU instruction cycles). Hence, our simulation takes all tasks as CPU instructions to enable GridSim accommodate them. The RAM and HD therefore acts as a factor which increased CPU instructions (using more HD or RAM implies more instructions because accessing HD and/or RAM requires processor time as well) and also as limiting factors. For example, if 10 tasks use 2GB RAM each at the same time, it cannot be scheduled if availability is only 10GB. This is the remarkable difference with the GridSim toolkit.

5.2 Experimental Results

We setup three grids (g_1, g_2, and g_3), three users (u_1, u_2, and u_3), and eight jobs ($j_1, j_2, \cdots, j_8$). We provided cost of compute cycles per grid (base prices) as $3.00, $2.50, $2.00 for g_1, g_2, and g_3 respectively. Users' jobs run on the grid based on their bids and the set base price. We assume that the base price is the same for the similar type of resource across various grids. At the completion of the simulation grid g_1 with a set base price of $3.0 had only one job, g_2 with a base price of $2.50 had two jobs, and g_3 with a set base price of $2.0 had five jobs. The base price values were reversed (g_1 =$2.0; g_2 =$2.50; g_3 =$3.0). Figure 4 and Figure 5 shows the effect of the chosen base prices. These figures depict the number of individual jobs on the grids. Figure 5 demonstrates the reversed base prices for the grids in Figure 4. We observe that there is a higher concentration of jobs executed in grids with lower base prices in other words when the price variant factor is lower. Using a multi-step trinomial for number of time step N = 4, 8, 16, 24, 32, for various strike price K and resources price S, time ($T = 0.5$ in years), interest rate ($r = 0.06$), volatility ($\sigma = 0.2$), and the number of time steps

($N_j = 2N+1$). We extend our study by varying the volatility σ in steps of $0.0, 0.1, \cdots, 0.7$, we computed the option values for the various three grids g_1, g_2, and g_3. In Figure 7, we notice that as the strike price increase the call option value decreases.

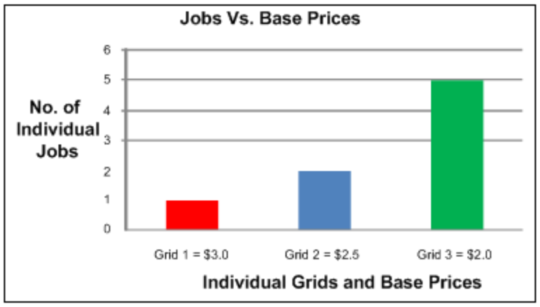

Figure 4: Based Prices Setup

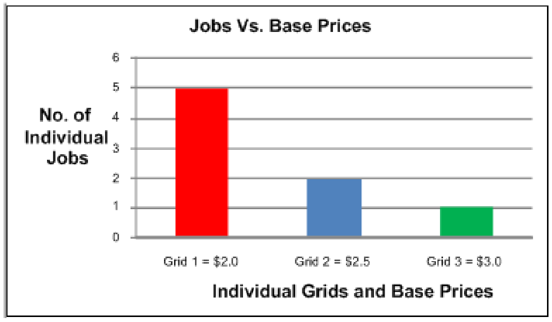

Figure 5: Reversed Base Prices Setup

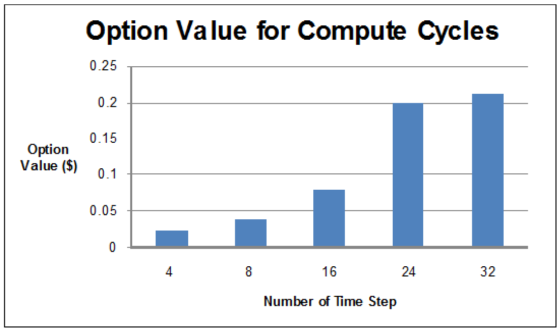

Figure 6: Option Value at Base Price of $2.00 in g3

These computed option values from the integrated GridSim trinomial behaves as expected. The computed option values are significantly smaller than value of the specified base prices at the individual grids. The smaller values of these option values shows that the grid can be made busy while a larger number of users are attracted for optimal grid provider incentives. Figure 6 shows the results of more simulations carried out using same parameters (as above) to verify the effects of time of exercise and the option values. Figure 6 shows that while the computed option values remains lower compared to the specified base prices, early exercise is more profitable. Since more users are most likely to exercise early were the option values are lower than the offered prices, our integrated pricing approach shows that it supports large user patronage and hence grid resource utilization can be significantly improved by extrapolating current usage patterns into the future. Figure 7 shows that despite option values occur at a relatively lower option value than the specified base prices. A user may decide to wait longer before exercising. Exercising late does not lead to profitability for the user because option values may become higher. For example, at higher time step (24 and 32 in Figure 6), the results of our trinomial option value computation converges and hence it does not profit waiting beyond these time steps. We also show a joint plot of strike price for the three grids under study and compare the option values at increasing strike prices. We varied the value of K between $1.5 and $3.55 in steps of $0.5. We observed that a user can actually combine resources across the three grids. Higher strike prices gave the lowest option value across the multiple grids.

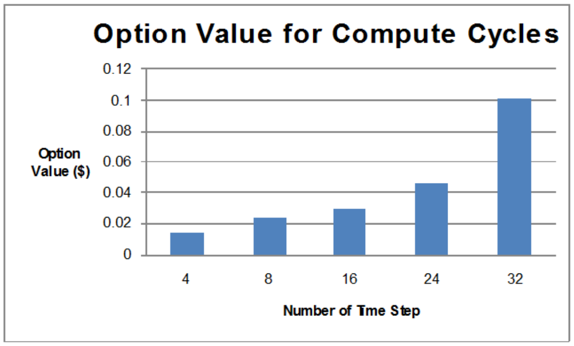

Figure 7: Option Value at base Price of $2.00 in G1

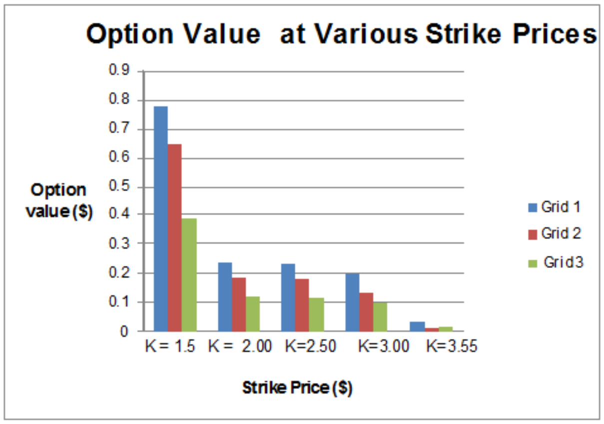

Figure 8: Option values at Various Strike Prices for Individual Grids

The jobs that ran in the two grids could wait for a longer time to be allocated resource(s) at a lower price. The users could be seen to have tradeoff; either to wait longer with a possibility of exercising at lower price or pay more to execute their jobs without waiting.

6.0 Conclusion

This paper presented the integration of a financial option based pricing scheme that prices multiple grid resources using the GridSim toolkit simulator. Our approach improves over other toolkits without a pricing integrated by using a Dutch action-like bidding system to allocated grid resources based on users' specifications and by applying base prices. The applied base prices and trinomial pricing injected lower option values for the grid user. In this paper, we have demonstrated flexibility in pricing grid resources using the price variant factor which have the effects of allowing users to take advantage of lower prices.

From the results, it can be seen that at any time the computed option values are low as specified using the base prices. We also show that the grid resource pricing using financial option theory significantly improved resource utilization. In the future, we plan to consider using actual trace from real grids (Western Canada Research Grid (WestGrid) with a node located at the University of Manitoba, Shared Hierarchical Academic Research Computing Network (SHARCNet), and Grid5000).

The GRM provides a mapping of gccs to various grids. In previous research efforts [10], [11], and [12] one resource in one grid or several resources of the same grid was presented. Sharma et al. in [26] priced cloud resources using financial economic model. Their model priced a single cloud resource. In the current paper, we priced multiple resources across multiple grids whose characteristic features (ownership, currency, policy, and zonal time) differ. In the grids g_1, g_2, and g_3 we pieced individual and combined resources cc_1, cc_2, and cc_3 of many users. The generated trinomial lattice represents option value for one grid. Figure 9 shows multidimensional grid and resource pricing for base prices $p_1, p_2, \cdots, p_n$.

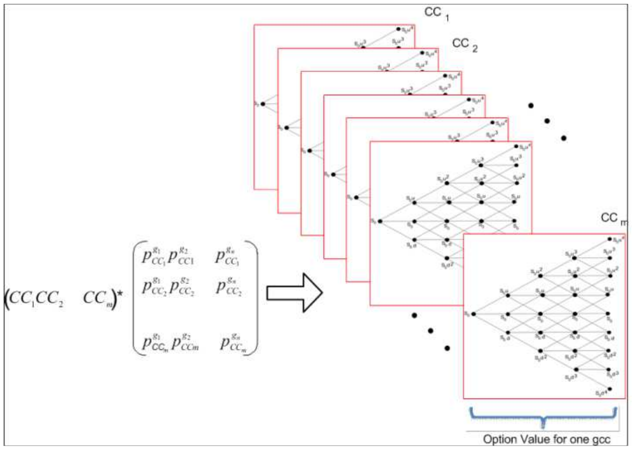

Figure 9: Multidimensional Grid and Resource Pricing

We treat each instance of $\left[cc_m^{g_n} * p_{cc_m}^{g_n}\right]$ in Equation (1) as a trinomial tree whose solution requires large computational resources of the grid because of its large size. Figure 9 idealizes the pricing multiple resources across multiple grids.

Acknowledgement

The authors acknowledge the partial financial supports from the University Research Grants Program (URGP) of University of Manitoba. The second author acknowledges the financial supports from the Natural Sciences and Engineering Research Council (NSERC) of Canada for this research.

References

[1] I. Foster and C. Kesselman, *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers Inc., 1999.

[2] D. Werthimer, J. Cobb, M. Lebofsky, D. Anderson, and E. Korpela, "SETI@HOME-massively distributed computing for SETI," *Computer Science Engineering*, vol. 3, no. 1, pp. 78–83, 2001.

[3] I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," *Int. Journal of Supercomputer Applications*, vol. 15, no. 3, pp. 200–222, 2001.

[4] K. M. Sim, "Grid Commerce, Market-Driven G-Negotiation, and Grid Resource Management," *Systems, Man, and Cybernetics*, vol. 36, no. 6, pp. 1381–1394, 2006.

[5] I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke, "A security Architecture for Computational Grids," in *ACM Conf. on Computer and Comm. Security*, 1998, pp. 83–92.

[6] 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.

[7] EC2. (2012, March) Amazon Elastic Compute Cloud (Amazon EC2). [Online]. Available: http://aws.amazon.com/ec2/ (visited October 2013)

[8] D. M. Chance and P. P. Peterson, *Real Options and Investment Valuation*. The Research Foundation Publication of AMIR, First Edition, 2002.

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

[10] D. Allenotor and R. K. Thulasiram, "G-FRoM: Grid Resources Pricing A Fuzzy Real Option Model," in *Proc. (CDROM) 3rd IEEE Int' Conf. on e-Science and Grid Computing (eScience 2007) Bangalore, India, December 10-13*, pp. 388-395, 2007.

[11] D. Allenotor and R. Thulasiram, “Grid resources pricing: A novel financial option-based quality of service-profit quasistatic equilibrium model,” in *Proc. (CD-ROM) 9th IEEE/ACM IntI Conf. on Grid Computing (Grid 2008). Sept. 29-Oct. 01, Tsukuba, Japan.*, pp. 75-84, Sept. 2008.

[12] D. Allenotor, R. Thulasiram, and P. Thulasiraman, “A financial option based grid resources pricing model: Towards an equilibrium between quality of service for users and profitability for service providers,” in *4th IEEE Intl. Conf. on Grid and Pervasive Computing (GPC 2009). May 04-08. Geneva.* vol. 5529, pp. 13-24, May 2009.

[13] The GridSim: A Grid Simulation Toolkit for Resource Modeling and Application Scheduling for Parallel and Distributed Computing website. [Online]. Available: http://www.gridbus.org/gridsim/ (visited October 2013)

[14] Lai Kevin and Rasmusson Lars and Adar Eytan and Zhang Li and Huberman Bernardo A., “Tycoon: An implementation of a distributed, market-based resource allocation system,” *Multiagent Grid Syst.*, vol. 1, no. 3, pp. 169–182, 2005.

[15] Stuer Gunther and Vanmechelen Kurt and Broeckhove Jan, “A commodity market algorithm for pricing substitutable Grid resources,” *Future Gener. Comput. Syst.*, vol. 23, no. 5, pp. 688–701, 2007.

[16] H. K. Bhargava and S. Sundaresan, “Contingent Bids in Auctions: Availability, Commitment and Pricing of Computing as Utility,” in *HICSS 2004: Proc. of the 37th Annual Hawaii IntI Conf. on System Sciences (HICSS’04) - Track 8*, pp. 201-227, 2004.

[17] W. Kang, H. H. Huang, and A. Grimshaw, “A highly available job execution service in computational service market,” in *The 8th IEEE/ACM IntI Conf. on Grid Computing. Sept. 1921, Austin, Texas, USA.*, pp. 275-282, Sept. 2007.

[18] Z. Tan and J. R. Gurd, “Market-based grid resource allocation using a stable continuous double auction,” in *The 8th IEEE/ACM IntI Conf. on Grid Computing. September 1921, Austin, Texas, USA.*, pp. 283 – 290, Sept. 2007.

[19] Zhang Juheng and Bandyopadhyay Subhajyoti and Piramuthu Selwyn, “Real option valuation on grid computing,” *Decis. Support Syst.*, vol. 46, no. 1, pp. 333–343, 2008.

[20] David Allenotor, Oseghale Igene, and Ruppa K. Thulasiram, A Financial Option Model for Pricing Cloud Compute Resources Journal of the Nigerian Association of Mathematical Physics. Vol. 21 (July, 2012), pp. 237-257, 2012.

[21] L. Chunlin and L. Layuan, “Pricing and resource allocation in computational grid with utility functions,” in *International Conf. on Information Technology: Coding and Computing (ITCC’05).*, vol. II, September 2005.

[22] A. Sulistio, W. Schiffmann, and R. Buyya, “Using revenue management to determine pricing of revervations,” in *Proc. 3rd IntI Conf. on e-Science and Grid Computing (eScience 2007) Bangalore, India, Dec. 10-13, pp.* 396-405, *2007*.

[23] F. Black and Scholes, “The Pricing of Options and Corporate Liabilities,” *Journal of Political Economy*, vol. 81, no. 3, pp. 637-654, 1973.

[24] J. C. Cox, S. Ross, and M. Rubinstein, “Option Pricing: A Simplified Approach,” *Journal of Financial Economics*, no. 7, pp. 229-263, 1979.

[25] E. Schwartz, “The Valuation of Warrants: Implementing a New Approach,” *Journal of Financial Economics*, no. 4, pp. 79-93, 1997.

[26] Bhanu Sharma, Ruppa K. Thulasiram, Parimala Thulasiraman, Saurabh K. Garg and Rajkumar Buyya (2012), *Pricing Cloud Compute Commodities: A Novel Financial Economic Model*, Proceedings of the 12th IEEE/ACM *IntI* Symposium on *Cluster, Cloud and Grid Computing (CCGrid12)*, Ottawa, Canada, pp. 451-457, May 13-16, 2012.