Computation Of Binomial Option Pricing Model With Parallel Processing On A Linux Cluster

Harya Widiputra

Faculty of Information Technology, Perbanas Institute

Jakarta, Indonesia

harya@perbanas.id

ABSTRACT

Binary Option Pricing Model (BOPM) is one approach that can be utilized to calculate the value of either call or put option. BOPM generally works by building a binomial tree diagram, also known as lattice diagram to explore all possible option values that occur based on the intrinsic price of the underlying asset in a range of specific time period. Due to its basic characteristics BOPM is renowned for its longer lead times in calculating option values while also acknowledged as a technique that can provide an assessment of the most fair option value. Therefore, this study aims to shorten the BOPM completion time of calculating the value of an option by building a computer cluster and presently implement the BOPM algorithm into a form that can be run in parallel processing. As an outcome, a Linux-based computer cluster has been built and conversion of BOPM algorithm so that it can be executed in parallel has been performed and also implemented using Python in this research. Conclusively, results of conducted experiments confirm that the implementation of BOPM algorithm in a form that can be executed in parallel and its application on a computer cluster was able to limit the growth of the completion time whilst at the same time maintain quality of calculated option values.

Indexing terms/Keywords

Binomial Option Pricing Model, Binomial Tree, Lattice Diagram, Parallel Computing, Computer Cluster

INTRODUCTION

called options that allow an investor to have the right to buy or sell on a stock at a certain price agreed upon in the future, one of the main challenges that arise in the area of financial engineering or financial informatics is to be able to calculate or place a considerably fair price for the contract [1].

In this regard, some methods to determine the price of options have been introduced and also used widely. Some of these methods include some techniques that: 1) based on the calculation of the numerical process of derivatives, 2) methods based on simulation of stock price movement over time or method of lattice [2], 3) methods that form the mathematical model [3], and also 4) the Monte Carlo method [4].

Of the several methods described above, the method of calculating the price of options using a mathematical model that was introduced by Fischer Black and Myron Scholes in 1973, later known as the Black-Scholes model is one common method used to calculate the price of options [5].

Interesting to note that in some previous studies, it has been submitted that the option price calculation using the Black-Scholes model gives results that are not fair, especially for American-type options and not the European ones [6]. In addition, other studies have also found that the lattice method as well as the Binomial Option Pricing Model (BOPM) is more suitable to be used in calculating the price of American-type option [7].

However, the main drawback of the BOPM in comparison with the Black-Scholes model is the longer computation time required to complete the valuation of the option price and also the needs of further computing resources at their disposal [7]. The main consequence to this condition is the delay time in making investment assessment while the growing world of financial investments requires an optimal decision of an investment to be taken in a shorter time [8].

Therefore, this study aims to explore a technique that can be used to speed up the process of computing the option price calculation using lattice method. Specifically the approach that will be explored is to distribute the computing process of BOPM to some nodes of different machines or computers so that the completion time of such model to value the fair price of an option in a specific range time period would be faster in comparison to the use of only a single node.

In detail the objectives of this research are outlined as follows:

1. To convert the BOPM algorithm into the form of the problem that can be solved using parallel computing techniques.
2. To build a distributed computing infrastructure consisting of multiple node machines that can then be used to complete the calculation of BOPM.
3. To prove that by changing the BOPM algorithm into a form of problem that can be solved using parallel computing techniques and then run it on a multi-machines distributed computing infrastructure, the option price calculation time can be reduced significantly and still provide a fair result.

The next section of the paper explains the basic concept of option pricing techniques, the BOPM algorithm and also discusses the basic concept of parallel computing and its application on some financial problems. The modified BOPM algorithm so it can be executed in parallel is outlined in the third part of the paper which is followed by results and discussion.

RELATED WORKS

Options and Options Contract

Option is one of derivative securities, or commonly known as derivatives only, which means that the price of such securities is determined, based or "derived from" another underlying securities prices [9]. Options are traded in the form of a contract, so it is also known as option contracts. Being traded as contracts means that options can only be bought or sold in certain period of time and obviously have a maturity date that is similar to futures contracts.

An option contract is divided into two types, namely a contract to buy the underlying asset of an option, also known as a call option and vice versa contract that is to sell the underlying asset of an option or so-called put option.

If an investor has the option to buy, then they are entitled to purchase the underlying asset of the option i.e. company stock, on certain agreed price within a certain period prior to the expiration date the contract [10]. The certain agreed price is called the exercise price or the transfer price. In contrast to the put option, then the investor has the right to sell the underlying asset of the option at the exercise price before the contract expiration date arrives. In this case, the exercise price is only based on the price agreed when the contract traded and does not refer to current asset prices in the market.

To get the right to buy or sell options contracts, an investor is required to pay the purchase price option, referred to as premiums (premium). The premium paid to the seller of the option that is the party who issues the option, or so called the writer. Additionally, as long as the option contract is valid, the writer is obliged to buy or sell the related assets in accordance with the price set in the contract option [9].

On the global financial markets there are two types of options, based on the rules of trade transactions, which often become the benchmark of financial markets in different countries namely the American and European options [1]. The main difference between the American and European options contained in the provisions regarding when an option contract holders can trade their options.

Dissimilar provisions regarding when the asset underlying the option can be traded had an influence on the method suitable for use in the process of pricing fair option values. For American options, due to the rule that the underlying asset can be sold or bought at any time up to the expiry date then of course the basic asset price volatility need to be taken into consideration [10]. In contrast, for the European options the pricing method is premised on the belief that during the term of contract the volatility of the underlying asset price is stationary [10].

Binomial Option Pricing Model (BOPM)

A binomial option pricing approach works base on a very simple assumption, that is at the end of the validity period of the option contract only two possible conditions can materialize. The first condition is the value of shares would increase to achieve the highest price while the second one is it will decrease until it reaches the lowest price that may be achieved within this period in accordance with applicable rules.

One option pricing model that utilizes this technique is the Binomial Option Pricing Model or the BOPM [11]. This model is widely used as it can be applied to many conditions where other models are not capable to work because of the limiting basic assumptions [1],[12].

BOPM generally perform calculations for option pricing by propagating the possibility of changes in stock prices over time and not just at one particular point in time. Therefore, BOPM is suitable to be used in pricing the American options which entitle the holder to purchase or sell shares at any time without having to wait for the expiry date or maturity time [12].

The calculation techniques used in BOPM are considerably uncomplicated when seen from mathematical modeling point of view. However, the main constraint is the relatively longer counting completion time. The length of counting completion time on BOPM is caused by the concepts that form the basic workings of BOPM. In this case, the option pricing is not only calculated at one point in time, which is the expiration date, but at each point in time during the contract period runs by including the growth probabilities of stock prices at each point time. The main stages into the core processes in BOPM option pricing is outlined as follows:

1. Step 1, development of binomial tree that traverses possible stock price movement over time. This tree is also referred to as lattice, therefore the BOPM also known as the method of lattice option pricing models.
2. Step 2, calculation of option values at any time point toward the expiry date of the contract.
3. Step 3, sequential calculation of option values at each point in time that precedes a particular position.

The Cox, Ross and Rubinstein method, hereinafter referred to as the CRR is widely known and has been used to build the lattice in particular the binomial price tree. Previous study has confirmed that the method provides assurance that the constructed binomial tree is recombinant [11]. In addition, the CRR method allows the value of stock at any point in time to be calculated directly without having to build its first tree with mathematical equations as follows:

$$S_n = S_0 \times u^{N_u - N_d} \qquad (1)$$

where S_n is the value of price of stock that acts as the underlying asset at time point n, S_0 is the initial stock price, N_u is the number of climb points and N_d is the number of drop-off points at a specific time point n. To calculate the value of the options at each end point which is the leaf of constructed binomial tree the following formula is applied:

$$\max\,[S_n - K, 0], \text{ for call option} \qquad (2)$$

$$\max\,[K - S_n, 0], \text{ for put option} \qquad (3)$$

where K is the exercise price and S_n is again the price of underlying asset at n. The value of options at each end point is automatically becomes a candidate for the fair value of the execution price of these underlying assets [11].

BOPM characteristics as defined in the calculations above illustrates that the model works by calculating all possible values of the options at each point in time within the option contract period. This gives the advantage that BOPM presents the results of a comprehensive assessment.

However, due to the counting process is carried out at each point in time BOPM computing completion generally takes longer in comparison to the other methods [7]. Additionally, Figure 1 illustrates how an option price binomial tree is constructed using the BOPM.

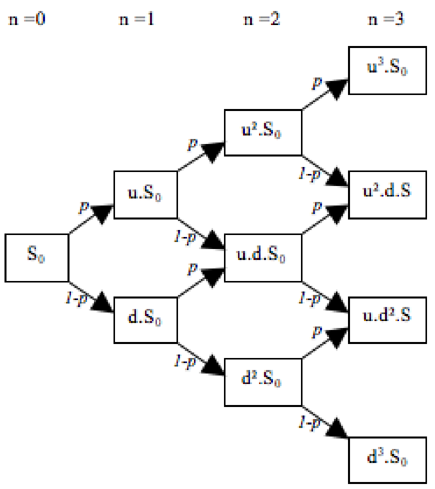


Fig 1. Illustration of the binomial tree development process using the BOPM with the number of time points n=3.

Parallel Computing on Financial Problems

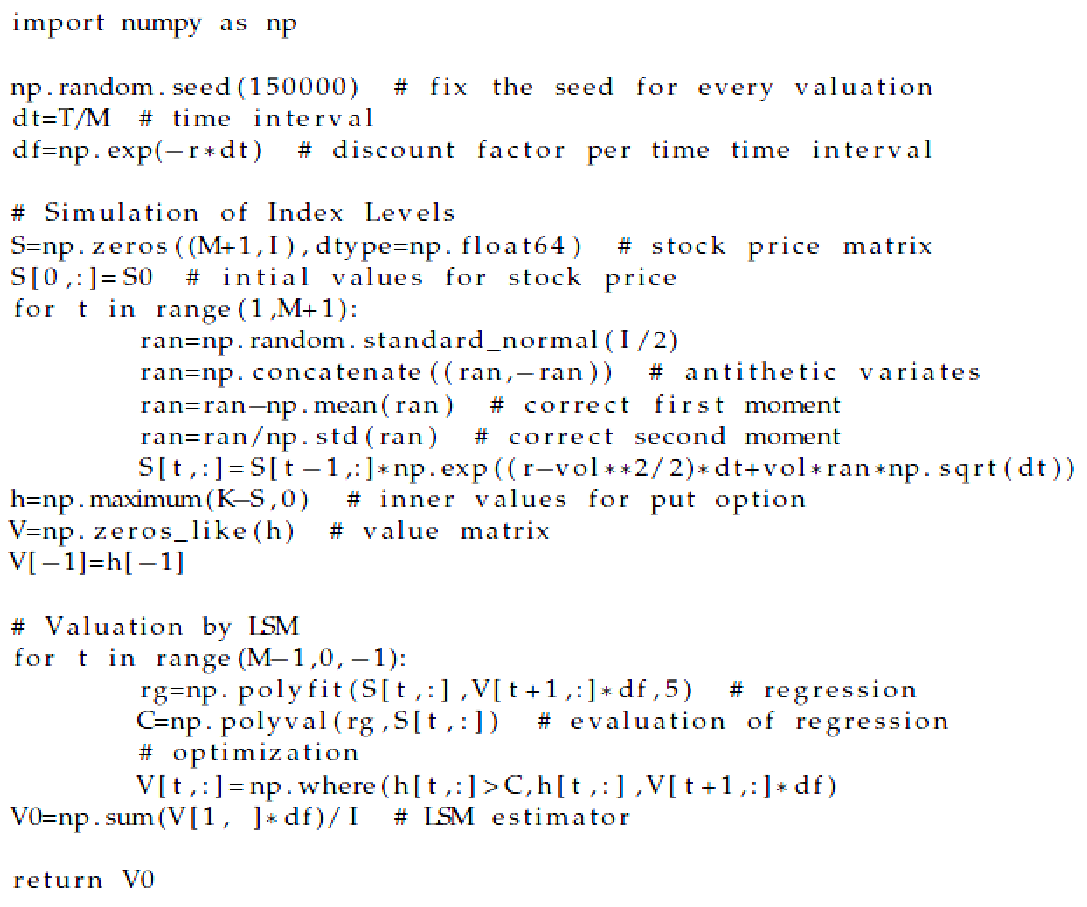
Parallel computing is a technique to solve a problem or mathematical model by running multiple computing processes simultaneously. Parallel computing is done by breaking the problem to be solved into smaller pieces of several sub-problems which are then solved using a number of computational processes that work in tandem. The computation outcome of each process will then be combined to produce the final output [13].

In reality, parallel computing is closely linked to existence of parallel computer architectures. In this design several computers are linked together to form a unitary grid and work together to accomplish a particular task [14].

Need a concern that not all problems or mathematical models can be solved using parallel computing technique as an approach. In the case of completion of a model or a mathematical equation requires a process that works sequentially and cannot be broken down into sub-problems that can be solved individually, then the use of parallel computing is not effective [13].

In line with this fact, in the application of parallel computing a conversion process of the mathematical model or equation into the form of sub-problems that can be solved independently is required. Furthermore, a programming technique that allows the generated sub-problems to be distributed and processed at different nodes of computer is also an absolute requirement for the realization of parallel computing.

In some previous studies, parallel computing has been used in the fields of grouping a text message [15], the image processing on the computer [16], data processing chemicals and genome [17], and also in terms of the financial settlement models [18].



```
import numpy as np

np.random.seed(150000)  # fix the seed for every valuation
dt=T/M  # time interval
df=np.exp(−r∗dt)  # discount factor per time time interval

# Simulation of Index Levels
S=np.zeros((M+1,I),dtype=np.float64)  # stock price matrix
S[0,:]=S0  # intial values for stock price
for t in range(1,M+1):
        ran=np.random.standard_normal(I/2)
        ran=np.concatenate((ran,−ran))  # antithetic variates
        ran=ran−np.mean(ran)  # correct first moment
        ran=ran/np.std(ran)  # correct second moment
        S[t,:]=S[t−1,:]∗np.exp((r−vol∗∗2/2)∗dt+vol∗ran∗np.sqrt(dt))
h=np.maximum(K−S,0)  # inner values for put option
V=np.zeros_like(h)  # value matrix
V[−1]=h[−1]

# Valuation by LSM
for t in range(M−1,0,−1):
        rg=np.polyfit(S[t,:],V[t+1,:]∗df,5)  # regression
        C=np.polyval(rg,S[t,:])  # evaluation of regression
        # optimization
        V[t,:]=np.where(h[t,:]>C,h[t,:],V[t+1,:]∗df)
V0=np.sum(V[1, ]∗df)/I  # LSM estimator

return V0
```

Fig 2. Implementation of Monte Carlo Least Square algorithm for the parallel computation of options price.

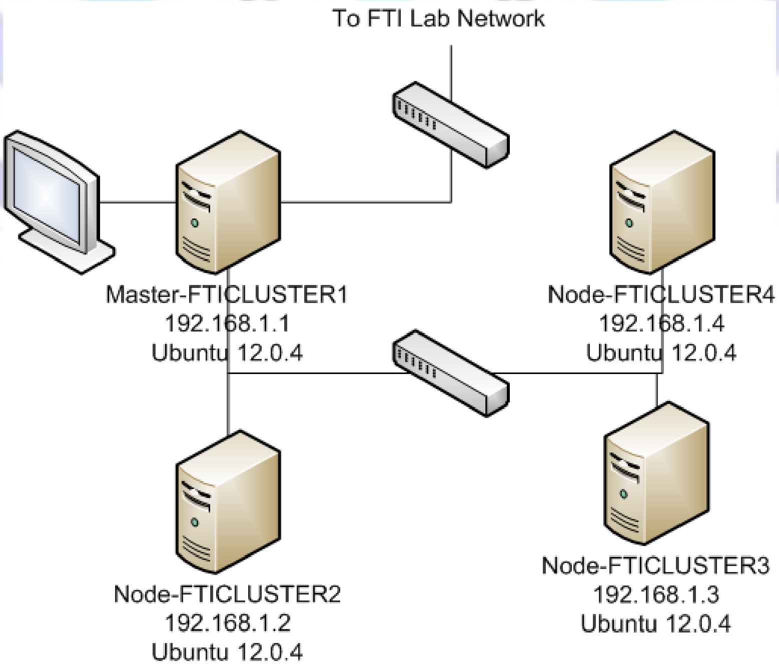


Fig 3. Configuration of built Linux Beowulf cluster that consists of 1 master node and 3 regular nodes.

Additionally, research conducted by Barua, et al. in 2005 also mentioned that parallel computing has been used to calculate the Fast Fourier Transform (FFT) in the financial derivative product pricing [19]. Furthermore, it is worthy to note that there are many previous studies which have also been using parallel computing techniques in calculating the price of options on various options market in the world [20],[21],[22],[23]. The results of these studies explain that the use of parallel computing can speed the counting process is a fair option price so that it can assist investors in making decisions at the time will be invested in the options.

Based on the results presented in previous studies, then this research focuses on the development of BOPM algorithm that can be executed in parallel and also would construct the computer cluster infrastructure, in which the algorithm will be implemented. By realizing these objectives, it is expected that outcomes of the study would help to accelerate the BOPM counting completion time and therefore become assistance for the investor to make more rapidly investment decision.

IMPLEMENTATION

BOPM Algorithm in Parallel Computing

Based on the main concept of BOPM algorithm that has been described in previous section, it can be understood that the key process of BOPM is to build a binomial tree that contains a value that is propagated to the future, both for the values of the underlying asset price, as well as the exercise price of the options, either call or put. With the establishment of the binomial tree the describes possible propagation of asset price and also the upcoming values of the options, then the decision maker, or in this case generally the investors can assess whether the exercise price of the options being offered is a fair price or not.

In this study, the examination of implementation for the BOPM algorithm is conducted using the Python programming environment. This is due to the ease of programs conversion into a form that can be run in parallel on a cluster of computers provided by Python.

The implementation of BOPM algorithm in Python requires an additional library named Numpy, which is an extension in Python that provides functions to perform complex mathematical calculations along with data and matrix in large sizes processing capabilities. The use of Numpy becomes important due to the characteristics of BOPM that has to construct a propagation binomial tree for the upcoming value of underlying asset and the fair settlement price of stock option under observation.

Additionally, the basic BOPM algorithm is also altered by putting the Least-Square Monte Carlo algorithm into place, as proposed by the Longstaff and Schwartz in 2001 [24]. In detail, the implementation of BOPM algorithm that has been adapted to a working source code as given in Figure 2.

Into the bargain, results of previous study have found that the growth rate settling time calculating the value of an option with BOPM will be directly proportional to the increase in the number of processors available in a computer cluster used [25].

Building a Beowulf Cluster

In this research in order to realize the completion of BOPM process with parallel computing technique, a computer cluster is built. The utilized architecture of the computer cluster constructed in this study is the Beowulf cluster that is operating on a Linux platform [26].

The Beowulf Cluster architecture is chosen based on published results of previous study that verify the performance of Beowulf Cluster which are reliable, easy to manage and has been widely used to run various parallel computing process [27]. The schematic configuration of the constructed computer cluster in this research is translated visually in Figure 3.

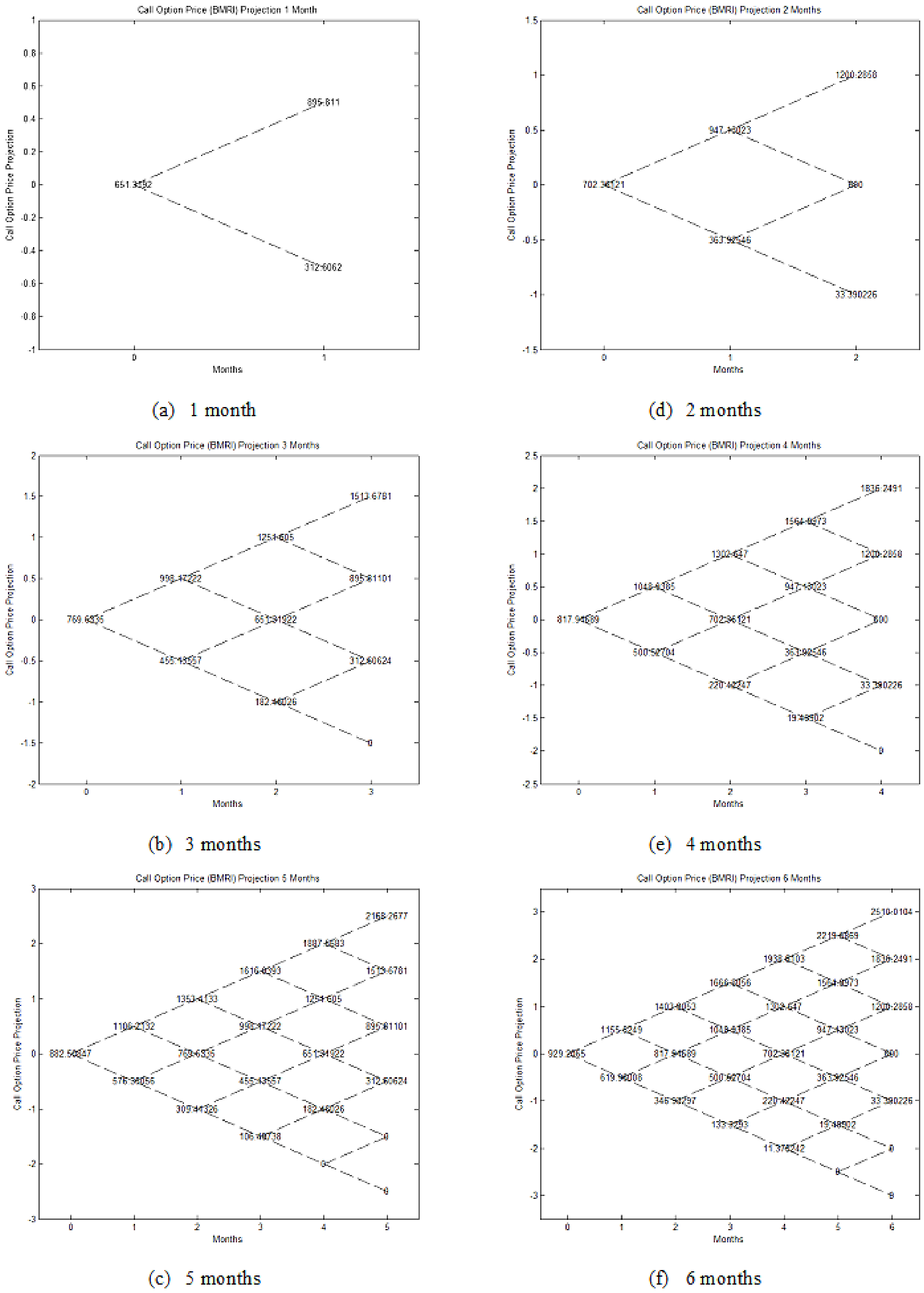
Furthermore, to run and manage the process of parallel computing on the clusters of computers that have been built, a parallel processing protocol called Message Passing Interface (MPI) is used. In essence, the MPI administers parallel computing processes by exchanging messages through an interface, in the form of programs that exist on each computer node in the cluster.

RESULTS AND DISCUSSION

In this study, the trials for the implementation of the BOPM algorithm that has been converted into a form that can be run in parallel on a computer cluster is done by calculating the option price for one of the shares listed on the Indonesian Stock Exchange. Option values that are to be calculated will be derived from the stock prices of one of the biggest bank owned by the government of Indonesia, namely PT. Bank Mandiri, Tbk. whose stock code is BMRI.

Calculations are performed to value both the call option and put option with maturity or execution time ranging from 1 to 6 months. Some parameters that are used to calculate the options price are defined as follows:

- Asset base value or price = 10,100
- Option exercise price = 9,500
- Risk free interest rate = 6.5% (using the interest rate of Bank Indonesia, the central bank of Republic of Indonesia)
- Asset volatility = 10% (based on data obtained from the Indonesian Stock Exchange)



(a) 1 month

(d) 2 months

(b) 3 months

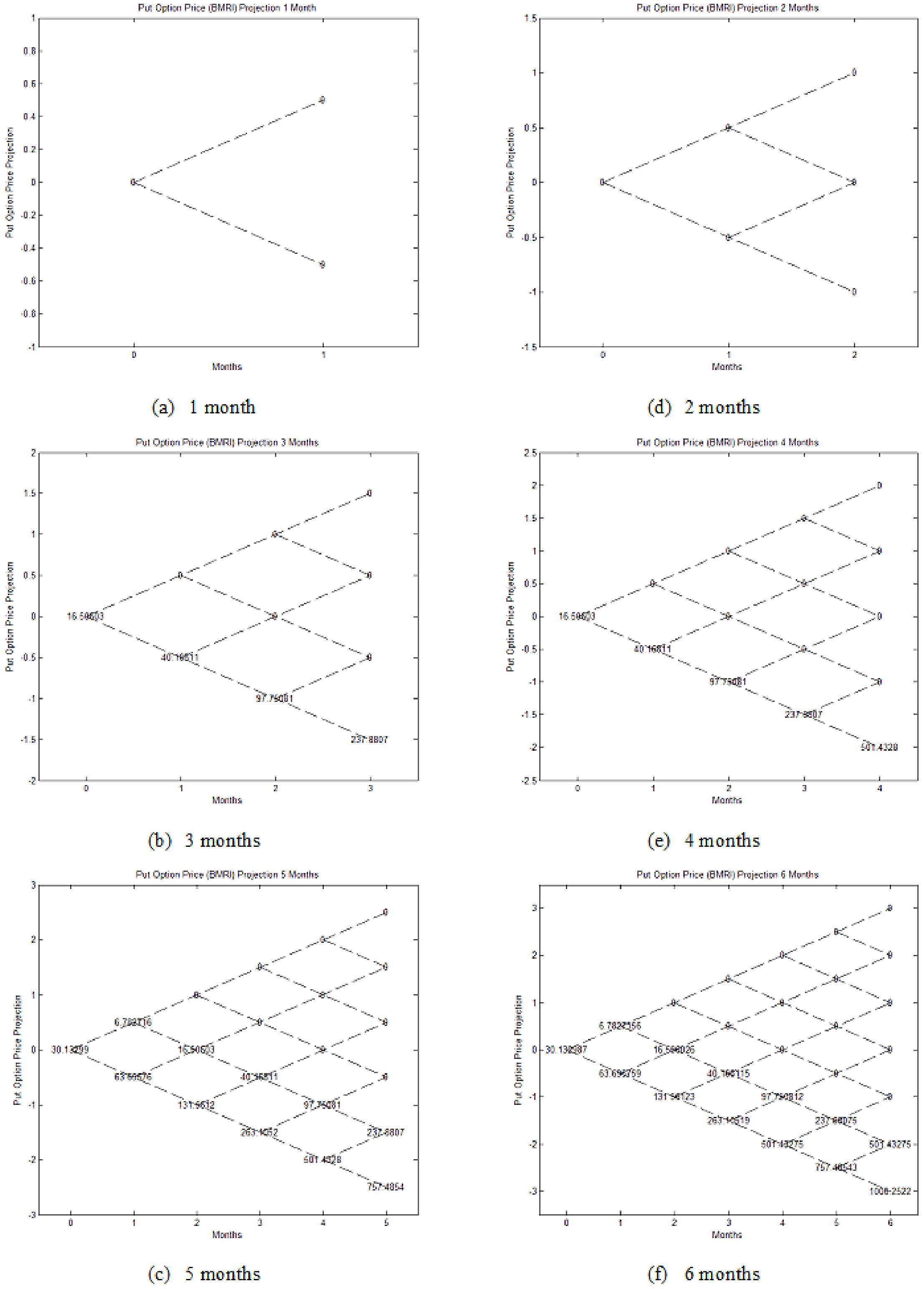
(e) 4 months

(c) 5 months

(f) 6 months

Fig 5. Lattice diagram of projected BMRI call option prices in 6 months period.

Furthermore, the projected possibility of asset price movements for up to 6 months ahead can be seen in Figure 4. Projected possibility of asset price movements are generated by the BOPM calculation that run on a Linux cluster that has been previously explained and built in this study. Based on the results obtained it then can be observed that the conversion of the BOPM algorithm into a form that can be run in parallel on a computer cluster is capable of constructing the option price binomial tree or lattice in a high-quality manner.



(a) 1 month

(d) 2 months

(b) 3 months

(e) 4 months

(c) 5 months

(f) 6 months

Fig 6. Lattice diagram of projected BMRI put option prices in 6 months period.

The calculation results of call option prices on the BMRI stock for execution time ranging from 1 to 6 months ahead is shown in Figure 5a to 5f. While the projection of put option prices on the BMRI stock for equal time period is depicted in Figure 6a to 6f. The main difference appears in the resulting lattice diagram for the values of the call and put option is the emergence of "0" which only occurs in the lattice diagram of the put option.

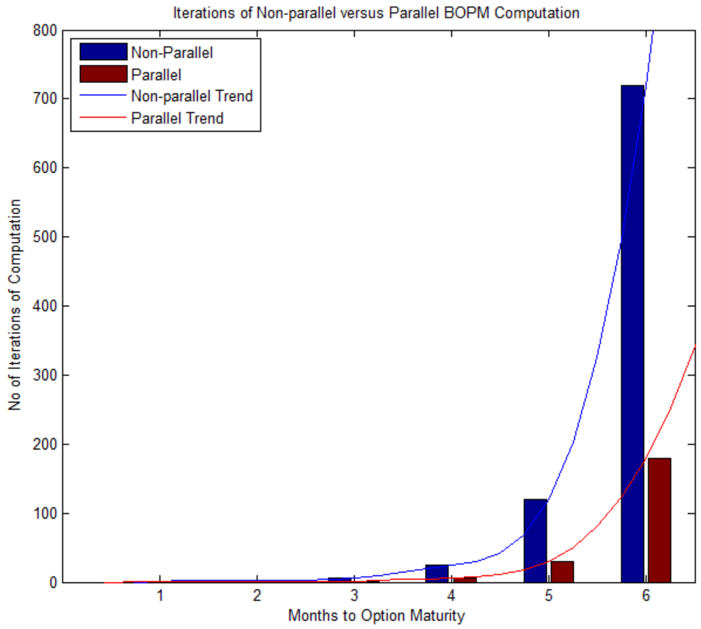


Fig 7. Computing time comparison between BOPM completion on a standalone system against the parallel system.

In this case, the value of "0" is defined as the absence of profit that can be obtained by investors. In other words, it suggests that buying a stock option contract with exercise price of 9,500 for BMRI share will not give any yield when being exercised at that particular time point.

As an illustration, in Figure 5a the lattice diagram of BMRI call option for a period of 1 month does not show any "0" at all and therefore it indicates that it is a good decision to invest in the call options of BMRI with an exercise price of 9,500 for a period of 1 month ahead if the price of the underlying asset at the moment is 10,100. In addition, the lattice diagram also shows a range of profit that may be obtained by the investor, starting from 312.60, when asset prices fell by up to 9,812.90, to 895.81, when asset prices rise by up to 10,395.81.

Contrary to the above conditions, the BMRI put options lattice diagram for 1 month period as shown in Figure 6a, are showing only "0". This situation explains that investing in BMRI put option with an exercise price of 9,500 for only the upcoming month ahead is not a good decision because it will not provide any benefit. This condition is caused by the movement of the projected prices of the underlying assets that will not likely to fall below 9,500 for the first months, as shown in Figure 4, while the put option contract will require the investor to sell its assets at the price of 9,500 when the market value of these assets is still predicted above 9,500.

The results obtained from the calculation of the BMRI call and put option values, which then represented in the form of lattice diagrams, are showing that the implementation of BOPM algorithm in the form of parallel computing has been successfully carried out. It is the can be confirmed that the conversed BOPM algorithm running on the built Linux clusters is still capable of doing the settlement of calculating the options price.

However, further analysis related to the performance of parallel computing process, in terms of the time of completion of the pricing, still needs to be done to show that the main purpose of this research activity has been accomplished. To analyze the performance of a single computing being compared to parallel computing on the completion of the BOPM algorithm for pricing the call or put options equal experiment on calculating the BMRI options as describe above has also been performed on a standalone system with single computational technique. A comparison chart of performance given by the two settings is depicted in Figure 7.

It can be clearly observed that the computational time required to complete the calculation of the BMRI option values on the Linux clusters is less than that of the standalone system.

Despite of the growth pattern of the computation time that holds the exponential form for both parallel and standalone system, the graph evidently shows that using parallel processing the completion time of BOPM is significantly reduced. The explanation for the exponential growth of the computation time lies on the nature of the BOPM algorithm itself when it generates the binomial tree. However, in this study, the alteration of the BOPM algorithm focused more in the area so that the algorithm can be executed simultaneously in parallel and not in the process to diminish the complexity of the algorithm itself.

Nonetheless, in the end results as shown in Figure 7 have confirmed that the main objective of this research, which is the application of BOPM algorithms in parallel on a computer cluster in order to expedite the completion of the calculation of fair value of the options have been reached.

CONCLUSION

Based on experimental results obtained and the analysis carried out in the course of this research, some conclusions can be drawn and described as follows:

1. The BOPM algorithm can be implemented and executed in parallel machines to significantly shorten the completion time of calculating the fair price for both call and put options in comparison to when being applied on a single machine.
2. The BOPM algorithm can be implemented in the form of parallel computing using Python and give a better performance, in terms of turnaround time compared to the completion of the model calculation on a single system.
3. The Least Square Monte Carlo algorithm is required in the conversion process of BOPM algorithm into the form that can be executed simultaneously on a computer cluster.
4. The Beowulf cluster architecture is proven to be suitable as the supporting infrastructure for parallel computing process in particular for the completion of option pricing practice using BOPM.
5. The ability to reduce completion time of options pricing calculation using BOPM algorithm that runs over a computer cluster could be of a help for investors in assessing the fair price of the options and therefore increase the potential to gain benefits in a shorter period of time.

FUTURE WORK

In this study, the conversion process and implementation of the BOPM algorithm emphasized more on the intention that the algorithm can be run in parallel over a computer cluster since the key objective is to be able to speed up the turnaround time of calculating fair option values using the BOPM algorithm. However, the characteristic of the algorithm which has a growing complexity in the form of an exponential function is still not explored further and therefore remains. For that reason, to improve the performance of the BOPM algorithm, in terms of complexity, in the future the alteration of the BOPM algorithm should focus more on the reduction of the algorithm computational complexity.

ACKNOWLEDGMENTS

This research is supported by research funding provided by the Asian Banking Finance and Informatics Institute Perbanas, Jakarta, Indonesia for the year of 2015/2016.

REFERENCES

1. Bodie, Z. (2009). Investments. Tata McGraw-Hill Education.
2. Boyle, P. P. (1988). A lattice framework for option pricing with two state variables. Journal of Financial and Quantitative Analysis, 23(01), 1-12.
3. Jiang, L. (2005). Mathematical modeling and methods of option pricing. Singapore: World Scientific.
4. Boyle, P. P. (1977). Options: A monte carlo approach. Journal of financial economics, 4(3), 323-338.
5. Black, F., & Scholes, M. (1973). The pricing of options and corporate liabilities.The journal of political economy, 637-654.
6. Hendrawan, R. (2010). PerbandinganModel Opsi Black-Scholes dan Model Opsi GARCH di Bursa Efek Indonesia. Jurnal Keuangan dan Perbankan, 14(1), 13-23.
7. Ganesan, N., Chamberlain, R. D., & Buhler, J. (2009). Acceleration of Binomial Options Pricing via Parallelizing along timeaxis on a GPU. Performance Computing.
8. Lander, D. M., & Pinches, G. E. (1998). Challenges to the practical implementation of modeling and valuing real options. The quarterly review of economics and finance, 38(3), 537-567.
9. Saunders, A., Cornett, M. M., & McGraw, P. A. (2006). Financial institutions management: A risk management approach (Vol. 8). McGraw-Hill/Irwin.
10. Smit, H., & Trigeorgis, L. (2009). Valuing infrastructure investment: An option games approach. California Management Review, 51(2), 82-104.
11. Cox, J. C., Ross, S. A., & Rubinstein, M. (1979). Option pricing: A simplified approach. Journal of financial Economics, 7(3), 229-263.
12. Georgiadis, E. (2011). Binomial options pricing has no closed-form solution. Algorithmic Finance, 1(1).

13. Asanovic, K., Bodik, R., Catanzaro, B. C., Gebis, J. J., Husbands, P., Keutzer, K., & Yelick, K. A. (2006). The landscape of parallel computing research: A view from Berkeley (Vol. 2). Technical Report UCB/EECS-2006-183, EECS Department, University of California, Berkeley.

14. Barney, B. (2010). Introduction to parallel computing. Lawrence Livermore National Laboratory, 6(13), 10.

15. Darriba, D., Taboada, G. L., Doallo, R., & Posada, D. (2012). jModelTest 2: more models, new heuristics and parallel computing. Nature methods, 9(8), 772-772.

16. Lee, C. K., & Hamdi, M. (1995). Parallel image processing applications on a network of workstations. Parallel Computing, 21(1), 137-160.

17. Sangket, U., Mahasirimongkol, S., Chantratita, W., Tandayya, P., & Aulchenko, Y. S. (2010). ParallABEL: an R library for generalized parallelization of genome-wide association studies. BMC bioinformatics, 11(1), 217.

18. Zenios, S. A. (1999). High-performance computing in finance: The last 10 years and the next. Parallel Computing, 25(13), 2149-2175.

19. Barua, S., Thulasiram, R. K., & Thulasiraman, P. (2005). High performance computing for a financial application using Fast Fourier Transform. In Euro-Par 2005 Parallel Processing (pp. 1246-1253). Springer Berlin Heidelberg.

20. Peng, Y., Gong, B., Liu, H., & Zhang, Y. (2010). Parallel computing for option pricing based on the backward stochastic differential equation. In High Performance Computing and Applications (pp. 325-330). Springer Berlin Heidelberg.

21. Sak, H., Ãzekici, S., & Bodurog, Ä. (2007). Parallel computing in Asian option pricing. Parallel Computing, 33(2), 92-108.

22. Surkov, V. (2010). Parallel option pricing with Fourier space time-stepping method on graphics processing units. Parallel Computing, 36(7), 372-380.

23. Wilbertz, B. (2012). GPGPUs in computational finance: Massive parallel computing for American style options. Concurrency and Computation: Practice and Experience, 24(8), 837-848.

24. Longstaff, F. A., Schwartz, E. S. (2001). Valuing American Options by Simulation: A Simple Least-Squares Approach. Rev. Financ. Stud. 14 (1). 113-147.

25. Hilpisch, Y. J. (2013). Python in Finance. Night of Talks of New York City Python Meetup Group, 27 Agustus 2013, Interactive Space.

26. Becker, D. J., Sterling, T., Savarese, D., Dorband, J. E., Ranawak, U. A., & Packer, C. V. (1995, August). BEOWULF: A parallel workstation for scientific computation. In Proceedings, International Conference on Parallel Processing(Vol. 95).

27. Saldaasa, R., Garcia, J., Muga II, F., & Yu, W. (2012). Development of a Beowulf- Class High Performance Computing System for Computational Science Applications. Science Diliman, 13(2).

Author' biography

Harya Widiputra received his PhD in Computer Science at the Auckland University of Technology, New Zealand. He is a Lecturer at the Faculty of Information Technology of Perbanas Institute, Jakarta, Indonesia. His research interests include issues related to computational intelligence, time-series analysis and modeling, data mining, evolving systems. He is author of a great deal of research studies published at national and international journals, international conference proceedings as well as book chapters, one of which has been edited by Springer Publishers.