Grid computing of Monte Carlo based American option pricing: Analysis of two methods *

Viet Dung Doan, Mireille Bossy, Françoise Baude, Ian Stokes-Rees
INRIA Sophia Antipolis, CNRS - I3S - Univ. Nice Sophia Antipolis
2004, Route des Lucioles, BP 93 - 06902 Sophia Antipolis Cedex, France
First.Last@inria.fr

October 25, 2007

Abstract - This paper aims to provide an overview and a performance comparison of some parallel and distributed algorithms for Bermudian-American option pricing. We use two Monte Carlo based methods to address such pricing in the case of a high number of assets (high-dimension) through continuation values classification and optimal exercise boundary computation. Our implementations are supported by a Java based technology grid-aware software framework offering fault tolerance, dynamic and aggressive load balancing. This master-workers grid software framework and the distributed pricing algorithms have been designed to serve in the context of financial applications.
Keywords : Grid computing, Bermudian-American option, optimal exercise boundary, parallel distributed Monte Carlo simulation.

1 Introduction

The PicsouGrid project was started in 2006 to investigate the application of grid computing techniques to financial option pricing. The project has been funded by the French national research agency (ANR) and has had industry partners from banks (BNP, Calyon and EDF), financial software companies (Pricing Partners Paris), and two groups from INRIA (OASIS and TOSCA). The work to date has focused on fault tolerance and load balancing of parallel European option pricing (see [1], [2]). European option pricing consists of fairly standard Monte Carlo (MC) simulations of the trajectories of an asset price, with the option exercise date fixed in advance. The latest work has moved on to parallel implementions of Bermudian-American (BA) option pricing algorithms, and is discussed here. American option pricing, with a variable exercise date (and its discrete time version, Bermudian options with a fixed set of equally spaced exercise dates) are much more computationally intensive, and therefore various approximation methods exist to improve the tractability of the MC pricing simulation.

Recent work has explored parametrization MC methods for the state space or the optimal exercise boundary (for short boundary) for sequential calculation of BA option pricing. However the computational intensity of these approaches, particulaly in the case of high-dimension, is an significant barrier to adoption. We address this problem by using a parallel approach. For instance, there are already some parallel algorithm for BA option pricing, such as Huang (2005) [6] or Thulasiram (2002) [9], which are based on the binomial lattice model and Muni Toke (2006) [10] which uses a MC approach but with a very limited number of processor. In this paper we examine on two approaches, the first one from Ibanez and Zapatero (2004) [7] which computes the optimal exercise boundaries and the second from Picazo (2002) [5] which uses the classification of continuation values. These two methods are similar in the recursive time programming so that at a given exercise opportunity we use many small independent packages of MC simulation to compute the continuation values. Next, they classify these values or use them

*This paper is an extended abstract accepted for the workshop of Monte Carlo for Financial Mathematics of the sixth International Association for Mathematics and Computers in Simulation seminar on Monte Carlo Methods (MCM2007), hosted by the Centre for Advanced Computing and Emerging Technologies, part of the University of Reading, June 2007. The full version has been submitted for review in *Mathematics and Computers in Simulation* (Elsevier Press)

to compute the boundary then in the final phase they use a standard, but large, MC simulation to pricing the option with the parameters calculated ealier. The MC method is often used in option pricing but still results in long computaion times for in high-dimensional problems. Ibanez and Zapatero (2004) state that the boundary calculation for 5 assets takes two days [7], which is clearly too long for modern financial markets. The use of grid computing to achieve practical run times has been gaining in computational finance. Luckily, the standard MC simulation is easily for parallelized so we can profit from this to decrease the computational time by using the grid.

In this paper, we have implemented a client - server architecture and exercised the distributed algorithms on it with a large number of processors. For the performance comparison, we validated the accuracy of our algorithms for pricing the option on one asset. We then also demonstrated the parallel efficiency of our algorithms for the case of multiple assets.

2 The Classification - MC approach

The MC approachs for BA option pricing is largely based on the continuation value estimation. For exemple the popular Least Square method of Longstaff and Schwartz [8] uses the regression method to estimate the continuation value at a given time then compares this value to the exercise value. We have implemented also his method for refrence. Here we are interested in describing the Classification - Monte Carlo approach (for short CMC) which was introduced by Picazo [5]. This approach is based on the observation that at a given exercise time the option holder makes his decision in view of $(exercise\ value - continuation\ value)$ is positive or negative. Picazo also uesd a classificantion algorithm to do this instead of a regression algorithm of higher complexity.

We describe here the algorithm that is used to characterize the boundary at each opportunity. Denote x a variable in an input space $\Omega \subset \mathbb{R}^d$ and $y = \beta(x, \epsilon)$ for some $k - dimensional$ random vector ϵ. We focus to find the regions of the input space Ω where $\mathbb{E}(y|x) > 0$ as well as where $\mathbb{E}(y|x) < 0$. In case of option pricing where the underlying's price S follows a geometric Brownian motion (GBM) with risk free rate, the variable $y = \beta(x, \epsilon)$ would be the difference between the value of continuation and the exercise value at a given time step t where $\epsilon \sim N(0, 1)$ and $x = S_t$. The problem becomes to find the function $F(x)$ having the same sign with $\mathbb{E}(y|x)$ using the AdaBoost algorithm [5].

In the second step the option is priced by standard MC simulation taking the advantages of the characterization of $F(x)$, so for the i^{th} we get the optimal stopping time τ_i = min $\{t \in \{t_1, t_2, ..., t_M\} | F_t(S_t^i) > 0\}$.

3 The MC - optimal exercise boundary computation approach

Ibanez and Zapatero focused on building a full boundary in a form of a polynomial curve with the dimension depends on the number of underlying assets. At a given exercise opportunity, they do a linear-interpolation or regress a quadratic or cubic function, and get a parameterization of this boundary. In the last phase, a MC simulation is run until the price trajectory reaches the dynamic barrier (boundary). Following Broadie and Detemple (1997) [3], Ibanez and Zapatero comment that for an option on maximum of M assets there are M separate exercise regions, characterized by M boundaries which are monotonic and smooth curves $\in$ space $\mathbb{R}^{(M-1)}$. However, for the problems where these properties can not be easily established (e.g. pricing swaptions in a general enough multi-factor term-structure model),the algorithm would have to be revisited. The main advantages of this method is that it provides to the option holder a full parameterization of the boundary and the exercise rule. The another convenience is the greeks hedging but this work will not be involved in this paper.

4 The parallel distributed implementations

4.1 The Classification - MC approach

4.1.1 The complexity of the algorithm

Let $\mathbb{O}()$ denote the complexity of the algorithm. In the classification phase we have to simulate N_1 paths with a finite number of exercise opportunities $m = 1, ..., N_T$. At a given opportunity, we simulate another N_2 paths to compute the continuation values. In this step we repeat a non-parametric regression on N_2 points to get the classification of the characterization of the boundary at this given date and this computation takes n iterations to converge. The complexity will be $\mathbb{O}(N_1 \times m \times N_2 \times n)$. Then we go to the second phase to compute the option value by simulating N a large number of standard Monte Carlo with m exercise opportunities. The complexity of the final phase will be $\mathbb{O}(N \times m)$. In total, we have the complexity of the method is: $\mathbb{O}(N_1 \times m \times N_2 \times n + N \times m)$. It can be observed that :

- Each point of the set N_1 points will be a seed which we use to simulate an independent package of N_2 simulation paths.
- From the exercise opportunity m backward to $m-1$ we use the Brownian motion bridge to simulate the price of the underlying.
- The complexity of the method is linear in the case of multiple assets option.

With these observations, we present our parallel approach in the next section.

4.1.2 The parallel algorithm

At $m = N_T$ we generate N_1 paths of the price of the underlying S_m^i, $i = 1, .., N_1$ then apply the Brownian motion bridge to get the price at time $m = m - 1$. Denote $x_i = S_m^i$, we then divide the N_1 paths by nb tasks then distribute them to a number of workers, each task has $\frac{N_1}{nb}$ paths. On receipt of a task, each worker simulates N_2 paths from each point in the task to compute the corresponding continuation value, then calculate the value $y_j = (exercise\ value - continuation\ value)$, $j = 1, .., \frac{N_1}{nb}$. The server collects the y_j from the workers until it has sufficient nb tasks. It then does a non-parametric regression with the set $(x_i, y_i)_{i=1}^{N_1}$ to get the signature of the function $F_m(x)$ mentioned above in section (2). After having the signature of $F_m(x)$ at time m, we repeat the same procedure in a recursive way for all earlier time intervals $[m-1, 1]$, then go to the second phase which uses a standard MC simulation to find the value of the option. We try also to figure out a parallel approach for the non-parametric regression.

4.2 The MC - optimal exercise boundary computation approach

4.2.1 The complexity of the algorithm

It can be seen that the two methods have the same construction for the recursive time programming aspect. With a finite number of exercise opportunities $m = 1, ..., N_T$, at a given opportunity, we simulate only J independent packages instead of N_1 in case of Picazo to compute the option value at this times in order to get J optimal boundary points. This procedure requires n iterations of Newton's method to converge. In case of a basket option (d underlyings) each underlying has its own boundary so the procedure must be repeated d times. The complexity of the boundary computation is: $\mathbb{O}(d \times J \times m \times N_2 \times n)$.
After having all the boundaries at all m exercise opportunities weor can go to the second phase to do a standard Monte Carlo of N paths to compute the price of the basket option (basket of d underlyings). The complexity of this phase would be $\mathbb{O}(d \times N \times m)$. Finally, we have the complexity for the method is: $\mathbb{O}(d \times J \times m \times N_2 \times n + d \times N \times m)$

4.2.2 The parallel algorithm

In fact, we can compute separately d boundaries and on each boundary, the computation of J optimal boundary points at a given exercise date can be simulated independently. Based on these observations, we present our parallel and distributed approach using the same architecture as in the case of Picazo.

We discuss here the computation of one boundary. At $m = N_T$, the boundary is definitively the strike value. Backward to $m = N_T - 1$, we have to estimate J optimal points from J initial good lattice points (see [7], [4]) to regress the boundary at this time. We distribute the J jobs to the workers, on receipt of a job, each worker simulates N_2 paths to compute the approximate points then use Newton's method to converge to the optimal point. The server collects J points to regress the boundary then repeats the same procedure at every point m, in a recursive way, until we reach $m = 1$. This is then repeated to calculate the optimal exercise boundary for all d assets in the basket. To find the price of the option, we can now compute it by using standard MC simulation.

5 Conclusion

We achieved good performances with these two parallel and distributed algorithms for BA option pricing using a large number of processors in a grid environment. Future work will add greeks hedging for BA options with the optimal exercise boundary and further scaling to large grid environment.

References

[1] F. Baude, M. Bossy, V.D. Doan, and I. Stokes-Rees. Experience and advances in grid computing for options pricing. In *Proceedings of the International Symposium on Grid Computing 2007*, Taiwan, March 2007.

[2] S. Bezzine, V. Galtier, S. Vialle, F. Baude, M. Bossy, V.D. Doan, and L. Henrio. A Fault Tolerant and Multi-Paradigm Grid Architecture for Time Constrained Problems. Application to Option Pricing in Finance. *2nd IEEE International Conference on e-Science and Grid Computing*, Netherlands, December 2006.

[3] M. Broadie and J. Detemple. The Valuation of American Options on Multiple Assets. *Mathematical Finance*, 7(3):241–286, 1997.

[4] S. Haber. Parameters for Integrating Periodic Functions of Several Variables. *Mathematics of Computation*, 41(163):115–129, 1983.

[5] F.J. Hickernell, H. Niederreiter, and K. Fang. *Monte Carlo and Quasi-Monte Carlo Methods 2000: Proceedings of a Conference Held at Hong Kong Baptist University, Hong Kong SAR, China, November 27-December 1, 2000*. Springer, 2002.

[6] K. Huang and R.K. Thulasiram. Parallel Algorithm for Pricing American Asian Options with Multi-Dimensional Assets. *Proceedings of the 19th International Symposium on High Performance Computing Systems and Applications*, pages 177–185, 2005.

[7] A. Ibanez and F. Zapatero. Monte Carlo Valuation of American Options through Computation of the Optimal Exercise Frontier. *Journal of Financial and Quantitative Analysis*, 39(2):239–273, 2004.

[8] F.A. Longstaff and E.S. Schwartz. Valuing American options by simulation: a simple least-squares approach. *Review of Financial Studies*, 2001.

[9] R.K. Thulasiram and D.A. Bondarenko. Performance Evaluation of Parallel Algorithms for Pricing Multidimensional Financial Derivatives. *Proceedings of the International Conference on Parallel Processing Workshops (ICPPW02)*, 1530, 2002.

[10] I.M. Toke. Monte Carlo Valuation of Multidimensional American Options Through Grid Computing. *Lecture notes in computer science, Springer-Verlag*, Volume 3743:page 462, 2006.