Super Quant Monte Carlo Challenge 2008
V GRID Plugtests

Organized by ERCIM, ETSI and INRIA
(Sophia Antipolis, France,
Monday 20 - Friday 24 October 2008)

Co-Organisers

Event sponsors

European Partners

Registration Deadline: 22 September 2008
INRIA, Sophia Antipolis - FRANCE
Tel: + 33 (0)4 92 38 77 77 Fax: +33 (0)4 92 38 77 65
More info available at : http://www.etsi.org/plugtests/GRID2008/GRID.htm

This page is intentionally left blank.

1 Goal of the "2008 Super Quant Monte Carlo Challenge"

The goal of the "*2008 Super Quant Monte Carlo"* challenge is to design and develop a Grid-enabled derivative pricing application, for pricing high dimensional European options or first generation exotic options, that can cope up with time critical intense computational demand for complex pricing requests.

The participants will have to deploy their option pricing application in a Grid environment using the ETSI GCM (the Grid Component Model) Interoperability deployment standard [15]. We will provide a ready-to-use Open Source implementation of this standard with the ProActive Parallel Suite [8]. The participants can use any middleware to develop their applications as long as they deploy their application with the GCM standard. To test robustness, scalability, and responsiveness, the application will have to go through the challenge of pricing an industry size batch of basket options (derivative products) within an assigned time slot. The application will be judged by a panel of experts and the winning team will take home the title of "*Super Quant 2008*".

The participation to this challenge, including the preliminary qualification tests, is **free of charge**. However, the participants that wish to participate in the **Face to Face challenge** in Sophia Antipolis (20th October – 23rd October 2008) have to register (for free) and pass the preliminary qualification tests (1st September – 1st October 2008).

No prior knowledge of Finance or Derivative pricing is necessary in order to participate in the challenge.

To facilitate the participation of the teams we have provided a fully functional, ProActive based prototype of this challenge. The teams with no finance experties can refer the source code for finance related algorithms. Whereas, the teams with finance background but no knowledge of Grid Computing can also refer the source code and understand how easy it is to develop large scale applications using the ProActive Parallel Suite. The source code is available for download on the Grid Plugtests 2008 website [12].

2 Background

Innovations over the past decades have made options and other derivative products a key part of global financial markets, partially due to computing advances which have allowed market investors to more accurately price these products. The legendary publication by Black-Scholes on an analytical model for pricing derivative financial products [2], have definitely resulted in a new era of derivatives trading. However, from the computational point of view, pricing of highly complex derivative products has been a computationally demanding task which further imposes a tremendous load on the computing infrastructure.

Yet another main problem in this situation is modelling the future asset prices and then estimating the payoff expectations. This typically is done using statistical Monte Carlo simulations and careful selection of the static and dynamic parameters which describe the market and asset. These computations may require considerably long time, upto a couple of days, with existing conventional (serial) computing platforms. In a liquid market, any delays in computing prices are unacceptable due to highly volatile and competitive market conditions. Pricing decisions generally need to be made in seconds or at most in minutes to be useful to market traders, hence the need to explore alternative strategies for pricing of complex options. Better option pricing (that being more accurate, more advanced models, and faster results) and hedging provide a market advantage and are therefore of a great interest to market traders.

Starting in the mid 90's, Grid computing has provided the trading community with tremendous computing power which can be harnessed in order to multiplicate the profits, hedge the risks and gain a competitive advantage in the market. The 2008 *Super Quant Monte Carlo* challenge is an attempt to further encourage the Grid community, as well as the computational finance community to increase the overall awareness towards the use of Grid computing in the domain of financial engineering.

3 What is the challenge?

As a challenge, the participants are asked to price an industry size batch of derivative products within a fixed duration (a time slot assigned to a participant). The participants might not need to understand the theory in order to develop a system. Neverthless, to understand what the overall task will look like, the basics are explained as follows. Experts in Finance can skip this part to take on the ***Challenge***. The terminologies in the following description can also be found in the relevant literature[3].

Consider a scenario that you have the following offer:

"In 3 months' you will have the option to buy Foogle Corp. shares at a price of $25 per share."

Thus, the options represent the right, and not obligation, to take any action, such as to buy, the share of Foogle stock (termed as an asset). Three months from now, you can check the market price and decide whether to buy the option at the price as per the option contract. Rationally, you would "exercise" the option only when the market price was greater than $25, as you can buy the share and immediately re-sell to gain the profit (payoff). Thus, the deal has no downside since after three months you can either make a profit or just walk away. However, the buyer of the option needs to pay a small amount upfront – which infact is the price of that option. (For more detailed descriptions on Put options (rights to sell), Call options (rights to buy), Basket options (whose payoff depends on the value of a portfolio (or basket) of underlying assets), please refer to [3].) There exist several variations of options, for example, a plain vanilla option is the standard type of option, one with a simple expiration date, strike price and no additional features. Another standard type of option is a Barrier option (where the option to exercise depends on the underlying crossing or reaching a given barrier level) [9]. For further theory behind this option and its variations; Up, Down, In and Out, please refer to [3].

The *option pricing problem* is to compute a fair value for an option [1]. It is the fair value at which the option can be bought and sold in a well arbitraged market. Not only the price of a option but the traders also need to know the Greeks values such as Delta, Gamma, Rho, Theta, and Speed. Each Greek measures the sensitivity of the value of a portfolio to a small change in a given underlying parameter, so that component risks may be treated in isolation, and the portfolio rebalanced accordingly to achieve a desired exposure [14]. For further mathematical notions behind the Greeks, please refer to Appendix. There exist several pricing methods including Finite Difference Methods for PDEs [6] or Monte Carlo Simulations [4], [5]. We chose Monte Carlo simulations for the **Challenge** for the following reasons:

1. The simulations are preferred because of the absence of straight forward closed form solutions for several financial models. For several state variable (several sources of uncertainties), real options models suffer the problem of the curse of dimensionality, which limit the model solution with others methods. For example, for more than three or four state variables, both lattice and finite-difference methods face several difficulties and are not practical.
2. Other main reason for using Monte Carlo methods is their suitability for pricing high dimensional options. The simulation permits simulate several

sources of uncertainties that affect the real value of the option (provided we control the optimal rule for exercise). The price of the option is expressed as an expectation, which in case of multi-asset options, is a multi-dimensional integral. With other pricing methods, the dimension is one limiting factor as the complexity of computation grows exponentially which is not the case with Monte Carlo simulations.
3. And most of all their embarrassingly parallel nature. The Monte Carlo based algorithms can be easily scaled to achieve almost linear scalability.

In the Appendix, we present a simple Monte Carlo based option pricing algorithm for pricing Barrier options. The sample code based on the algorithms described in this document is given on the Grid Plugtests 2008 website. The source code uses the Monte Carlo Framework available in the ProActive Parallel Suite [8].

For a quick start, non-finance experts are advised to have a look at the article, Option Pricing by Simulation [10] and [11].

The Challenge

Most of the available option pricing applications tend to evaluate low dimensional basket options (upto 1-10 underlying assets). As a "*Super Quant 2008*" challenge, the participants will have to develop a scalable pricing application that can utilise the computational Grid connecting up to several thousands of computing cores to price basket options of up to 100 underlying assets.

The "**2008 Super Quant Monte Carlo Challenge"** is to price a batch of the following configuration:

* The batch of test cases contains a set of 1000 derivative products/options (vanilla and barrier options). Each test case consists of a basket option of high dimensions. The underlying assets are derived from a pool of 1000 correlated assets. The correlation matrix and the initial start prices for these assets are given a priori to the contestants in the descriptor files (in XML format). The sample program and example tests are available on the Grid Plugtests 2008 website [13].

Further, the configuration of the batch for the final contest is as follows :

* 500 10-dimensional derivative products with 2 years time to maturity (the time at which the option expires)
* 240 30-dimensional derivative products with 9 months time to maturity
* 240 50-dimensional derivative products with 6 months time to maturity
* 20 100-dimensional derivative products with 3 months time to maturity

Other necessary market and asset parameters such as interest rate or dividend etc will be provided to the participants.

Monte Carlo simulations rely on two major factors for the accurary as well as the computational time for convergence:

1. Number of Monte Carlo simulations i.e. The number of simulated trajectories of the future asset prices that are used to compute the pay off expectations.

2. The time step for the trajectory simulation of these future prices in the interval from time t = 0 (today) to the maturity date (2 years, 9 months and so on). For the challenge, this time step has been fixed to 1 day for each product for each maturity.

All the necessary parameters will be provided in a standard format as defined in the sample test cases. For the input and output formats, please refer to *the schema* in the sample source code. ***Note:*** Your code must comply with these input/output formats as demonstrated in the sample code.

If you have any doubts or questions please post them on the PLUGTESTS-GRID mailing list (PLUGTESTS-GRID@LIST.ETSI.ORG) as well as the ProActive mailing list (proactive@ow2.org).

4 How to participate?

By participating in the contest, the participants agree to be bound by the contest rules defined as follows.

Teams willing to take part in the challenge will have to go through the following steps:

1. To get started, download the following files from the website:
   - ProActive 4.0 from the website http://www.activeeon.com/downloads.html
   - The sample source code from the Grid Plugtests Website (http://www-sop.inria.fr/oasis/plugtests2008/ProActiveMonteCarloPricingContest.html) or the ActiveEon Website(http://www.activeeon.com/downloads.php)

   1.1 Run the preliminary tests available with it.

2. Pass the preliminary remote qualification phase (requires free registration). A more elaborative set of sample test cases will be published in the early weeks of September 2008. The participants will then be able to test their application for these sample test cases or fine tune the application and any algorithms. These test cases will be representative of the test cases in the final challenge.

3. Participate in the **Face to Face challenge** during which the participants will execute their code on the computational Grid provided during a fixed and measured time slot (typically one hour) to calculate the highest number of solutions as fast as possible. The points-based judging criteria is described later in the document. The highest points earning team will win the contest.

4. Provide their logs and source code to the jury. After the contest all source code produced by the participants will be made available to public as open source code.

Face to Face challenge

Participants to the **Face to Face challenge** will be given access to a worldwide Grid of interconnected processors (approximately 3000/5000 cores [11]). Their code must be deployed and executed during the assigned time slot, and the final results will be evaluated by the jury.

Judging:

After the **Face to Face challenge**, the results of each team will be examined by a jury formed of experts in the Finance and the Grid computing domains. The primary evaluation criteria is explained below and note that it is subject to change and registered team will be duely notified about it in well advance on the mailing lists. Therefore participants are strongly adviced to register on the mailing lists mentioned earlier.

To evaluate the participants' programs objectively, we have outlined the following points based system.

1. The main evaluation criteria is the total number of derivative products, N, that are priced during the assigned time slot. For each price computed, the team gets +10 points. Thus a team can earn upto +10 * N points.

2. The accuracy and the precision achieved for each computed price:
    2.1 If the computed price is within the expected precision*, the team gains +5 points
    2.2 If the computed price is above the expected precision, the team gains +10 points
    2.3 If the computed price is below the expected precision, the team is penalized with -10 points.

*The tolerable error in computing the prices and other Greek values is 10^{-3}. (Since the accuracy of the values computed relies on the start prices of the underlying assets, we will consider the **relative errors** with respect to our reference prices and Greek values. These reference values will be computed with much larger Monte Carlo simulations. To know whether their systems are performing correctly, the participants are adviced to test their system against the test cases in the qualification tests which will be released later.) If the values achieve better accuracy (less than 10^{-4}), then the team will qualify for a bonus point for each value computed with such precision. Note, since the Monte Carlo methods are based on random numbers, the participant's algorithm might get lucky once in a while, but our judging system will ignore such anomalies (No!, we do not believe in *the hot hand fallacy*).

3. The additional Greeks computed along with the price of a derivative product;
    3.1 For each Greek, namely Delta, Gamma, Rho, Theta, and Speed that is precisely computed, the team will get 2 points per Greek. The greeks must be computed by a finite difference method with a fixed step size, chosen by the jury (see the input descriptor) see the sample code for further details. You will have to compute, Delta, Gamma, and Speed for each asset in the basket. Note that if non-precise, the values will not be given any points.

4. Finally, the time required for the entire computation.
    4.1 For each minute saved out of the total time limit, the team will gain 1 point.

The evaluation/analysis of the results and announcement of outcome of the contest is the sole responsibility of the jury. In the (unlikely) case where several teams would be placed equal first, the jury include other criteria to decide the winner: (some may include - in random order) total cpu-time used, number of grid nodes used in the competition, efficiency of the Grid usage, etc.

Although participation to the **Face to Face challenge** is free of charge, challengers have to have passed the preliminary remote qualification tests and/or to be registered:

Participants will:

1. Use the preparation day (Monday, 20th October) to warm-up, get familiar with the local platform and the Grid. This will give them the opportunity to tune and adapt their code in order to get it running smoothly on the Grid. The format of the input/output will remain the same as provided in the sample code on the contest website.

2. Use their time-slot during the challenge (21-23 October) to execute their code in real time on the Grid.

3. Provide their logs and source code to the jury. After the contest all source code produced by the participants will be made available as open source code.

4. The time slots schedule (participating order) will be announced at the beginning of the event. If a team is not ready at the time of its time slot, that team will be re-scheduled after all other participants.

5 References:

[1] D. J. Higham, Black-Scholes for scientific computing students, Computing in Science and Engineering, 6 (2004), pp. 72–79.

[2] F. Black and M. S. Scholes, The pricing of options and corporate liabilities, Journal of Political Economy, 81 (1973), pp. 637–654. Available at http://ideas.repec.org/a/ucp/jpolec/v81y1973i3p637-54.html

[3] Options, Futures, and Other Derivatives by John Hull.

[4] Monte Carlo methods in finance by Peter Jäckel.

[5] Monte Carlo Methods in Financial Engineering by Paul Glasserman.

[6] www.premia.fr

[7] Résolution de modèles d'évaluation de produits dérivés financiers sur des architectures de grilles informatiques. I. Muni Toke. PhD Thesis, *Ecole Centrale de Paris*, july 2006.

[8] http://proactive.inria.fr/

[9] http://en.wikipedia.org/wiki/Barrier_option

[10] www.scieng.ed.ac.uk/Students/Awards/Optionpricing.pdf

[11] http://www.etsi.org/plugtests/grid2008/Providers.html

[12] Option pricing by simulation http://finance.bi.no/~bernt/gcc_prog/recipes/recipes/node12.html

[13]http://www-sop.inria.fr/oasis/plugtests2008/ProActiveMonteCarloPricingContest.html

[14] http://en.wikipedia.org/wiki/Vega_(finance)

[15] http://www.etsi.org

Appendix

Due to extensive mathematical nature we have provided all the necessary algorithmic details in the accompanying (Appendix.pdf) pdf document.