Empirical analysis of the parallelization of the Monte Carlo simulation and its usage in the field of Economy

Edlira MARTIRI[1], Kozeta SEVRANI[2], Ezmolda BAROLLI [3]

[1, 2, 3]*Department of Mathematics, Statistics and Applied Informatics,*
Faculty of Economy, University of Tirana,
edlira.martiri@unitir.edu.al; kozeta.sevrani@unitir.edu.al; ezmolda.barolli@unitir.edu.al

ABSTRACT

Monte carlo simulation is one of the techniques used to generate pseudo-random numbers which can comprise a certain set of random data. These data are used in different fields of study, especially in those which require large amounts of data to predict a given situation, for example the market in economy, risk management, particle movement in physics, telecommunication, applied statistics and many others.

Since this generation of data requires a lot of time if executed on a single machine, the best approach is to use parallel systems. These systems unfortunately are not used in the albanian reality even though they bring a lot of advantages not only for the academic world, but even to important companies or institutions [2].

In our paper we will analyze some of the most important methods applied in the field of economy. The monte carlo algorithm, as one of the most important and wide used algorithms in economic simulations, will be seen in the parallelization point of view. We will consider empirical characteristics such as speed-up and especially the complexity of this parallelized algorithm. By doing so we will demonstrate some of the advantages of parallel systems such as time and money savings.

Definitions and introduction to parallelism

In the history of science, software has been written in the form of serial computation. Serial computation means that software must be run on a single PC, its instructions must be executed one after the other and only one instruction must be executed at any moment in time. If we see deeper these characteristics then we can see that every one of them can be reconsidered. So, software can be run on many computers and instructions can be executed in parallel at the same time. These reconsiderations yield to "parallel computing". There are two main ways of building a parallel system: multiple processors within a computer, or a number of computers connected together via a network.

Parallel execution dramatically reduces response time for data-intensive operations on large databases typically associated with decision support systems (DSS) and data warehouses. Symmetric multiprocessing (SMP), clustered systems, and large-scale cluster systems gain the largest performance benefits from parallel execution because statement processing can be split up among many CPUs on a single system [6, 7].

Also, owning, managing, or working with a critical business application, means dealing with performance problems. The application can't handle the ever-increasing data volume, it can't scale to meet new demand, or its performance is never good enough or fast enough. You need a higher level of performance so you can multiply the number of transactions your application can handle. In today's computing environment, there's really only one way to get there: Utilize a parallel architecture to run multiple tasks at the same time.

Historically, parallel computing has been considered to be "the high end of computing", and has been used to model difficult scientific and engineering problems found in the real world. Some main fields are: Atmosphere and Environment, Applied Physics, Chemistry, Seismology, Circuit Design and even human or social sciences such as Economy, which will be also the main field treated in this paper. Undependably by the field of study, today, commercial applications require large amounts of data which must be processed in sophisticated ways.

Practical parallelizations in Economy

One of the main parts of Economy which deals with calculations of large amount of data is Econometrics. According to Investor's Words:
"Econometrics is the application of statistical and mathematical methods in the field of economics to describe the numerical relationships between key economic forces such as capital, interest rates, and labor"[22].
In many Econometric models and applications we can have parallelism, since in most of cases we deal with sums and multiplications of vectors and matrices, which are the best place where to introduce parallel computations.
Another important aspect of Economy is the simulation of different models. The most sophisticated approach to forecasting or to predicting the effect of policy changes is to build a full scale simulation model which can be developed for any complex system. Simulation models of a company are often called financial planning models [1].
Computer simulation has to do with using computer models to imitate real life or make predictions. When you create a model with a spreadsheet like Excel, you have a certain number of input parameters and a few equations that use those inputs to give you a set of outputs (or response variables). This type of model is usually deterministic, meaning that you get the same results no matter how many times you re-calculate. As an example we can mention the deterministic Model for Compound Interest [8].

Monte Carlo Simulation

A Monte Carlo method is a technique that involves using random numbers and probability to solve problems. The term Monte Carlo Method was coined by S. Ulam and Nicholas Metropolis in reference to games of chance, a popular attraction in Monte Carlo, Monaco (Hoffman, 1998; Metropolis and Ulam, 1949, also [10, 11]).
According to Investor's Word, it is an analytical technique in which a large number of simulations are run using random quantities for uncertain variables and looking at the distribution of results to infer which values are most likely.

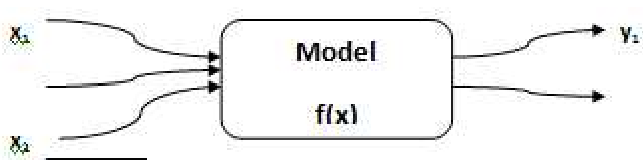

Figure 1. A parametric deterministic model maps a set of input variables to a set of output variables.

[22] http://www.investorwords.com/1638/econometrics.html

Monte Carlo simulation is a method for iteratively evaluating a deterministic model using sets of random numbers as inputs. This method is often used when the model is complex, nonlinear, or involves more than just a couple uncertain parameters [11]. A simulation can typically involve over 10,000 evaluations of the model, a task which in the past was only practical using super computers.

Monte Carlo simulation applications in various disciplines of Economy

Monte Carlo simulation has been successful primarily in fields related to modeling complex systems in biological research, engineering, geophysics, meteorology, computer applications, public health studies, project management and finance.
Monte Carlo simulation does get some exposure through certain **project management practices**. This exposure is primarily in the areas of cost and time management to quantify the **risk level** of a project's budget or planned completion date [12, 13].
In **time management**, Monte Carlo simulation may be applied to project schedules to quantify the confidence the project manager should have in the target project completion date or total project duration. Project manager and subject matter experts assigns a probability distribution function of duration to each task or group of tasks in the project network to get better estimates. A three-point estimate is often used to simplify this practice, where the expert supplies the most-likely, worst-case, and best-case durations for each task or group of tasks. The project manager can then fit these three estimates to a duration probability distribution, such as a normal, Beta, or triangular distribution, for the task. Once the simulation is complete, the project manager is able to report the probability of completing the project on any particular date, which allows him / her to set a schedule reserve for the project [13].
In **cost management**, project manager can use Monte Carlo simulation to better understand project budget and estimate final budget at completion. Instead of assigning a probability distribution to the project task durations, project manager assigns the distribution to the project costs. These estimates are normally produced by a project cost expert, and the final product is a probability distribution of the fi nal total project cost. Project managers often use this distribution to set aside a project budget reserve, to be used when contingency plans are necessary to respond to risk events [11-13].
Monte Carlo simulation can also be used in other areas of project management, primarily in **program and portfolio management** when making **capital budgeting and investment decisions**. Smith (1994) outlined how simulation assists managers in choosing among different potential investments and projects. He explained that by replacing estimates of net cash flow for each year with probability distributions for each factor affecting net cash flow, managers can develop a distribution of possible Net Present Values (NPV) of an investment instead of a single value [10]. This is helpful when choosing between different capital investment opportunities that may have similar mean NPV but differing levels of variance in the NPV distribution [13].

In recent years the complexity of numerical computations in financial applications has increased enormously, putting more demands on computational speed and efficiency. Monte Carlo simulation is one of main tools in **computational finance**. Monte Carlo simulation is a standard approach in derivatives pricing, mortgage pricing [5], to estimate the value of capital budgeting projects, and many other areas of finance [11] and beyond [10].
Monte Carlo simulation has played an increasingly important role in modem computational finance. This is due to the flexibility of the method in handling increasingly **complex financial derivatives** and the advent of the powerful computing facilities, which has significantly reduced the execution time required for the simulation. However, the main drawback of Monte Carlo simulation is its computational burden. A large number of simulation runs may be required to value complex derivative securities and to calculate hedge parameters such as delta and gamma with reasonable accuracy. So, we need an algorithm which can speed up computation for the Monte Carlo simulation. A parallel Monte Carlo algorithm can give a way to break up the computation into parallel sub-computations.

SEQUENTIAL MONTE CARLO METHOD

There are eight steps involved in simulation using Monte Carlo[23].

Step 1. Describe system probability distributions
Step 2. Decide on the measures of performance.
Step 3. Compute cumulative probability distributions.
Step 4. Assign representative ranges of numbers.
Step 5. Generate random numbers and compute the system's performance.
Step 6. Compute measures of performance.
Step 7. Stabilization of the simulation process.
Step 8. Repeat steps 5 and 6 until measures of system performance stabilize.

In 1973, Robert C. Merton published a paper[24], presenting a mathematical model, which can
be used to calculate a rational price for trading options [14].
Mertons work expanded on that of two other researchers, Fischer Black and Myron Scholes, and the pricing model became known as the Black-Scholes model. The model depends on a constant (Sigma), representing how volatile the market is for the given asset, as well as the continuously compounded interest rate r. The Monte Carlo Method approach takes M number of trials as input, where M could be 1,000

[23] The Monte Carlo Simulation Technique Applied in the Financial Market, Cornelia MUNTEAN, Faculty of Economic Sciences, West University of Timişoara, Romania, pp. 114.

[24] F. Black and M. S. Scholes. The pricing of options and corporate liabilities. In Journal of Political Economy, pp 637-654, 1973.

to 1,000,000 large depending on the accuracy required for the result [14]. The pseudo code for one individual experiment is shown below[25].

```
// A Sequential Monte Carlo simulation of the Black-Scholes model
1: for i = 0 to M do
2:        t := S * exp ((r - 0.5*sigma^2)* T + sigma * sqrt(T) * randomNumber())
3:        trials [ i ] := exp(-r * T ) max{t - E, 0}
4: end for
5: mean := mean( trials )
6: stddev := stddev( trials , mean)
```

Where:
S : asset value function, E: exercise price, r: continuously compounded interest rate, Sigma: volatility of the asset, T : expiry time, M : number of trials [14].

In a parallel implementation, the computation in line (2-3) can be computed by concurrent threads of execution. The function randomNumber() must be generating uncorrelated (pseudo)random number sequences [4, 14].

EMPIRICAL ANALYSIS OF THE MONTE CARLO PARALLELIZATION

In order to show the advantages of parallel computations of the Monte Carlo Method, we will show in this section some of the main factors which influence the performance of the abovementioned algorithm in a parallel system.

a. *Concurrency*
   Definition: an algorithm is considered concurrent in the case where a large number of instructions can be parallelized [6].
   Monte Carlo algorithm: since we have a **for-loop**, we can divide a number of iterations in each PC.
b. *Grain Size*
   Definition: it is a measure of the parallelism size. It refers to the number of instructions which can be executed in parallel [3].
   Monte Carlo algorithm: In this case we have a large number of instructions which can be executed in parallel; therefore we say that it is a **coarse-grain size** algorithm.
c. *Speed-Up*
   Definition: Is the ratio between the sequential computation time and the parallel computation time [3].

[25] Monte Carlo Methods, Jike Chong, Ekaterina Gonina, February 15, 2010, pp. 11.

Monte Carlo algorithm: We will consider a situation where we need to predict with a sample M=100. We will calculate the Speed-Up S for 8 different number of PCs: 1, 2, 4, 5, 10, 20, 50, 100.

d. *Efficiency*
Definition: Efficiency of an algorithm which is executed on p-processors is defined as the ration between Speed-Up and the number of processors [6].

e. *Effectivity*
Definition: it is one of the most important factors. An algorithm is considered effective if it maximizes the Sp*Ep product [3].

Analyses of the algorithm: charts and results

In our short form of the sequential Monte Carlo simulation of the Black-Scholes model we have two instructions inside the for-loop:

```
2: t := S * exp ((r - 0.5*sigma^2)* T + sigma * sqrt(T) *
randomNumber())
3: trials [ i ] := exp(-r * T ) max{t - E, 0}
```

These instructions will be sent together at every node of the system, since they are dependent on each other. And we have two instructions outside the for-loop.

```
5: mean := mean( trials )
6: stddev := stddev( trials , mean)
```

These instructions will also be sent at a single node, since even they are dependent on each other.

In our description of the factors above, we considered the number of iterations M=100.

The table below shows the values of T_p, S_p, E_p and $S_p * E_p$ according to the given numbers of PCs.

Table 1. Calculations of main factors depending on the number of PCs

PC	T_p	S_p	E_p	$S_p * E_p$
1	102	0.411765	0.411765	0.16955
2	52	0.807692	0.403846	0.326183
4	27	1.555556	0.388889	0.604938
5	22	1.909091	0.381818	0.728926
10	12	3.5	0.35	1.225
20	7	6	0.3	1.8
50	4	10.5	0.21	2.205
100	3	14	0.14	1.96

The Speed-up of the algorithm is almost linear and this can be seen in the graph below:

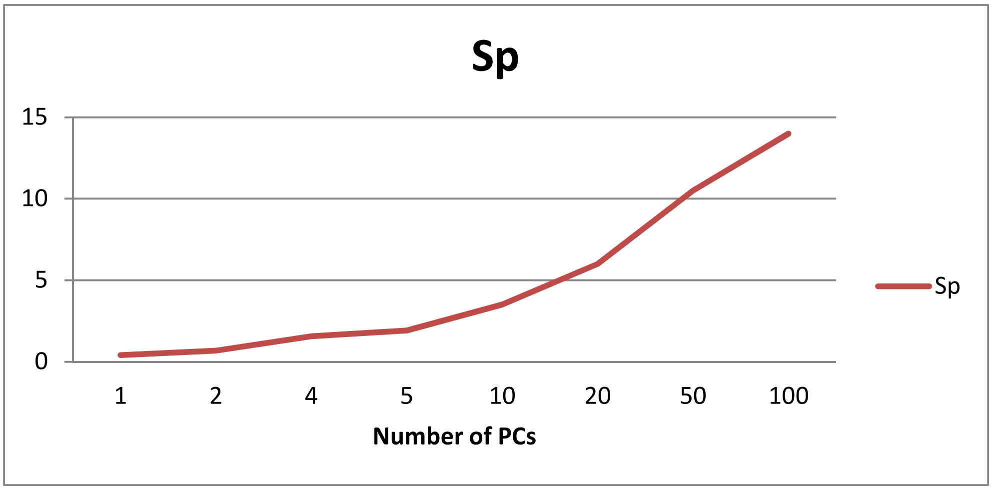

Figure 2. Graph of Speed-up in relation to the number of PCs

According to the function Sp * Ep, there is a maximum at number of PCs equal to 50. This means that for this sample, 50 is the optimal number of PCs.

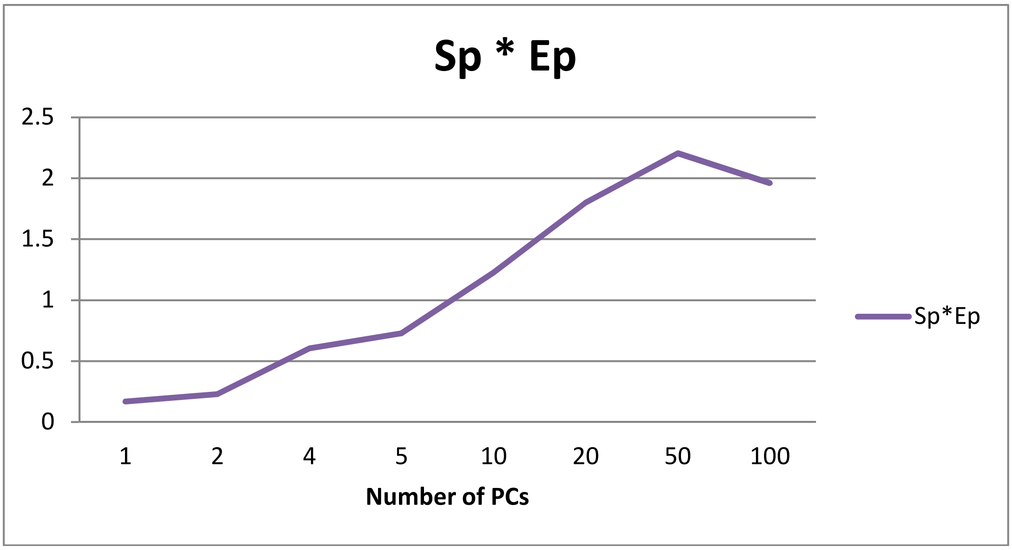

Figure 3. Graph showing the relation of Sp * Ep according to the number of PCs

If we try different sample number M, we always get the same result: **"the optimal number of PCs is equal to half the number of iterations M"**. We tested this for M = {10, 100, 500, 1000}. This shows that Monte Carlo algorithm can be suitable for every kind of parallel system, be it a simple cluster or even on High Performance Computers.

CONCLUSIONS AND FUTURE WORK

We presented in this paper parallelism which is the idea of breaking down a task so that, instead of one process doing all of the work many processes do part of the work at the same time. Parallel execution dramatically reduces response time for data-intensive operations. Today we need a higher level of performance so that we can multiply the number of transactions that an application can handle. In today's computing environment, there's really only one way to get there: Utilize a parallel architecture to run multiple tasks at the same time. Especially in the field of Economy we think that parallel systems must be adopted as soon as possible in our countries. Especially now in the era of technology, our rajon needs the fastest and complete solutions.

As the main part of this paper, we showed one method which is very useful not only in Economy but also in other fields of study which deal with the generation of pseudo-random numbers or statistical calculations. The Monte-Carlo method is highly parallelizable and this is a very good characteristic of the algorithm lying behind it.

We showed that this algorithm has a coarse-grain size and a quasi linear speed-up. We analyzed the different factors related to parallelism such as speed-up, effectivity, etc and showed the optimal number of processors of the system where the Monte Carlo algorithm could run. Our result showed that if M is the number of samples, then the optimal number of PCs is M/2.

As future work we plan to do the same calculations but on the practical side. We will compare the factors mentioned above with the empirical results of this paper.

REFERENCES

[1] Donald Erdman, "Simulation Models for Business Planning and Economic Forecasting", SAS Institute Inc., Cary, NC

[2] Edlira Martiri, Gloria Tuxhari, Albana Gorishti, "Parallel Computing: still missing in the Albanian reality", International Conference, Economic & Social Challenges and Problems, Faculty of Economy, University of Tirana, December 2010.

[3] Fatmir Hoxha, "Elemente te njehsimit paralel", Textbook, SHBLU, Tirana, 2004.

[4] Fatmir Hoxha, Andrea Kotro, "Matematike e Zbatuar", Textbook, SHBLU, Tirana, 2002.

[5] J. Boudoukh, R. F. Whitelaw, M. Richardson, and R. Stanton. "Pricing mortgage-backed securities in a multifactor interest rate environment: a multivariate density estimation approach", Review of Financial Studies, 20(3):769 – 811, 2005.

[6] Blaise Barney, Lawrence Livermore, "Introduction to Parallel Computing", National Laboratory.

[7] Ian Foster, Designing and Building Parallel Programs, Addison-Wesley, 1995

[8] Cornelia MUNTEAN, The Monte Carlo Simulation Technique Applied in the Financial Market, Faculty of Economic Sciences, West University of Timişoara, Romania.

[9] Monica Nedelcu, Bogdan Oancea, "Parallel algorithms for large scale Macroeconometric models", Journal of Applied Quantitative Methods, winter 2007

[10] A. Dubi. "Monte Carlo Applications in Systems Engineering". Wiley, 2001.

[11] Don L. McLeish. "Monte Carlo Simulation and Finance". Wiley, 2005.

[12] D. Smith, "Incorporating Risk into Capital Budgeting Decisions Using Simulation" . *Management Decision* . 32 (9) , p. 20 .

[13] Young Hoon Kwak, Lisa Ingall, "Exploring Monte Carlo simulation applications for project management", Risk Management Journal 2007, 9, (44–57) 2007 Palgrave Macmillan Ltd

[14] Kathy Yelick, "A Parallel Monte Carlo Simulation for Black-Scholes", University of California, Computer Science Division