Possibilities of Grid Computing for Realization of Simulation Problems

Ivaylo Penev, Anatoliy Antonov

Abstract: *The paper presents the authors' research on the possibilities of Grid technologies for realization of compute intensive problems. An experimental Grid environment, in which a Monte Carlo simulation for evaluating the market risk of a financial position is proposed. Comparison data, obtained by performing the simulation in an environment, controlled by a traditional high throughput computing system are discussed.*

Keywords: *Grid Computing, Monte Carlo Simulation, High Throughput Computing, Distributed Computing System.*

I. INTRODUCTION

Simulation methods are used to model systems, which are too complex to be analyzed by other methods. One of the most often simulation methods used is the Monte Carlo method [3]. The method simulates the processes of the system directly, by randomly generated, correlated values, used as input for the system. In common an enormous number of simulation trials is performed and finally the desired result is taken as an average over the number of observations. The Monte Carlo realization often requires lots of computing power and the execution of the simulation on a single computer usually takes ineligibly long time.

There exist contemporary technologies for harnessing the unused computing power for realization of high intensive jobs. Long time simulations could be performed in the environment of a local network and also in distributed environment over different administrative domains.

An area where the Monte Carlo is widely applied is Market Risk Evaluation. In [1] is described a realization of the simulation in a local computer network by means of the high throughput computing system Condor. The results show that the proposed approach reduces the time necessary for the process execution. The current work continues the authors' research on the methods for more effective usage of computing power, presenting the same simulation in an experimental Grid environment, which gives possibilities for execution of high intensive computing jobs over different administrative domains in Internet.

First, some essential notes on the concepts of Grid computing are discussed. Second, the whole simulation process is analyzed, sub processes are separated and those one is defined, which could be realized in parallel. Next, an experimental Grid environment in which the sub processes are implemented and tested is proposed. The algorithm of the simulation execution in the Grid is described. Finally, comparison data obtained by the implementation of the same simulation in a Condor pool and in the Grid environment are discussed.

II. Brief Description of the Grid Infrastructure [2,5]

In brief the Grid is a technology used for unifying information resources in a single virtual super computer. Resources include processors, storage devices, transmission data devices, applications. The Grid distinguishes from the traditional systems for distributed computing with the possibilities for gathering resources from different administrative domains. This allows more effective usage of global resources disposed all over the world.

To build a grid structure specific software, called middleware, is used. The main middleware subsystems are:

- Load balancing system – responsible for matching the requirements of a job with suitable resources;
- Data managing system – providing access to data in the Grid, allowing replication between different domains;
- Information system – giving static and dynamic information about a specific resource;
- System for authorization and authentication of users;
- Monitoring system;

III. CALCULATION PROCEDURE FOR MARKET RISK EVALUATION USING MONTE CARLO SIMULATION

III.1. Evaluation methodology and analysis of the simulation process [1,3]

Market Risk involves the uncertainty of future earnings resulting from changes of various independent underlying assets in market environment (prices of assets, interest rates, etc.) for a certain future time (risk horizon). The market risk of a position or of a portfolio is measured by a single value, called market Value at Risk (VaR).
The Value at Risk calculation is defined by the RiskMetrics methodology [4]. It includes calculation of statistic parameters: mean value and standard deviation of relative changes of interest rates and net present values. The methodology is based on the assumption that the interest rates relative changes are normally distributed.
A Monte Carlo simulation engine produces price distribution of a single financial position or of a portfolio aggregate applying numerical calculation procedures on a large set of possible market scenarios derived from statistic data for market moves from historic series.
The methodology for the Monte Carlo simulation is discussed in detail in [1].

III.2. Analysis of the simulation process

The entire process of Monte Carlo simulation could be divided into the following sub processes:

1. **Preparation of scenario set**.
   It includes construction of volatility vector, correlation matrix, covariance matrix, Cholesky matrix, generation of normally distributed relative changes of market drivers (interest rates), scenario vectors.
2. **Evaluation calculations**
   The function producing the price of the financial position (Net Present Value – NPV) is calculated a number of times. As the function calculations are independent each other, they could be performed in parallel in a Grid structure.
3. **Producing the price distribution and calculating the VaR**
   The resulted price distribution is constructed by counting the appearance of the values within many adjacent small ranges and there after the Value at Risk is calculated by numerical integration of the distribution density using confidence percentages, for example 1% and 5%, for the upper limit of the price distribution.

Each of the sub processes defined above is realized by a separate simulation step. The most time-consuming step is the second one. The evaluation function is calculated lots of times with various arguments. The separate calculations are independent each other and therefore the simulation is suitable to be tested in an experimental Grid environment.

III.3. Proposed experimental Grid environment for evaluation the market risk of a financial position

The current problem is to evaluate the market risk of a financial position with two payments – one (50 currency units) after 6 months and the other (80 units) after 12 months. The interest rates for each month for one year are known.

The evaluation function F, calculating the Net Present Value (NPV) of the position, is given by discounting the two payments to current date:

$$F = \frac{50}{1 + x * 0.5} + \frac{80}{1 + y * 1}.$$

As a middleware for the simulation the Globus Toolkit 4 is used. It is an open source software toolkit, developed by the Globus Alliance, for building Grid systems and applications. The system provides the following most key features [6]:

Service-oriented architecture. GT4 software is designed to support applications in which sets of services interact via standard protocols. The software includes both complete services and libraries implementing useful protocols.
Infrastructure services. GT4 includes built in services for accessing, monitoring, managing, and controlling access to such infrastructure elements as computational and data resources.
Web services. GT4 makes heavy use of industry-standard Web services protocols and mechanisms for service description, discovery, access, authentication, authorization, and the like.
GT4 containers. The GT4 software includes components that can be used to construct GT4 containers for hosting Web services written in Java, C, and Python.
Security. A highly standards-based security subsystem addresses message protection, authentication, delegation, and authorization.
Standards. Wherever possible, Globus implements standards or other broadly adopted specifications, so as to facilitate the construction of operable and reusable components and the use of standard tools.
The experimental Grid environment consists of three connected computers – central manager (Host A) and two executing nodes (Host B and C). In terms of the simulation logic the participants perform the following functions:

- Host A

Prepares the scenarios set;
Divides the set into parts;
Submits the parts to the executing nodes;
Collects the results returned by the nodes;
Produces the price distribution in 40 numerical ranges;

- Host B and host C - calculate the evaluation function with the provided scenarios independently each other.

In terms of the Grid the hosts perform the following roles:

- Host A – Certificate Authority;

Each host and user in the computational Grid needs a certificate in order to submit a job. Host A acts as a certificate authority, responsible for generating certificates to hosts and users. The certificate contains X.509 credentials. This function is performed by the Simple Certificate Authority (Simple CA) component, available in the Globus Toolkit 4.

- Host A and host B – computational nodes;

The hosts A and B act as computational nodes in the Grid, calculating the evaluation function independently each other. Each node performs the following GT4 components:

- GridFTP – protocol for transferring files in the Grid;
- Web services container – a set of web services written in Java, C or Python;
- RFT (Reliable File Transfer Protocol) – protocol, complementing Grid FTP;
- WS GRAM (Grid Resource Management and Allocation) – service, responsible for matching a job with suitable computational resources;

All hosts keep file transfer information in local Postgres databases and work under control of Linux operating system.

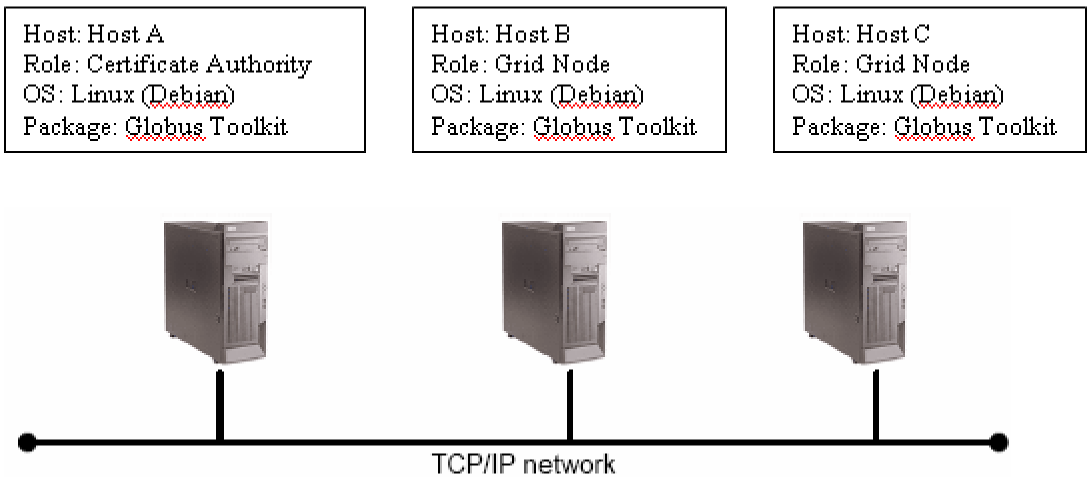

Fig. 1 Schema of the Grid environment

III.4. Algorithm of the Monte Carlo simulation, executed in the computing Grid

The following sequence of actions for starting the simulation in the Grid is performed:

- Generating of certificates for users and hosts in the Grid by the certificate authority;
- Certificate validation;
- Starting the application in the central manager;

For realization of the Monte Carlo simulation logic the Java language is used. Each job has XML description:

```
<?xml version="1.0" encoding="UTF-8"?>
<job>
   <executable>/usr/java/j2sdk1.4.2_18/bin/java</executable>
   <argument>calc</argument>                //The application Java class
   <stdin>${GLOBUS_USER_HOME}/stdin</stdin>       //Standard input for the node
   <stdout>${GLOBUS_USER_HOME}/stdout</stdout>//Standard output for the node
```

<stderr>${GLOBUS_USER_HOME}/stderr</stderr> //Standard error output for the node
</job>

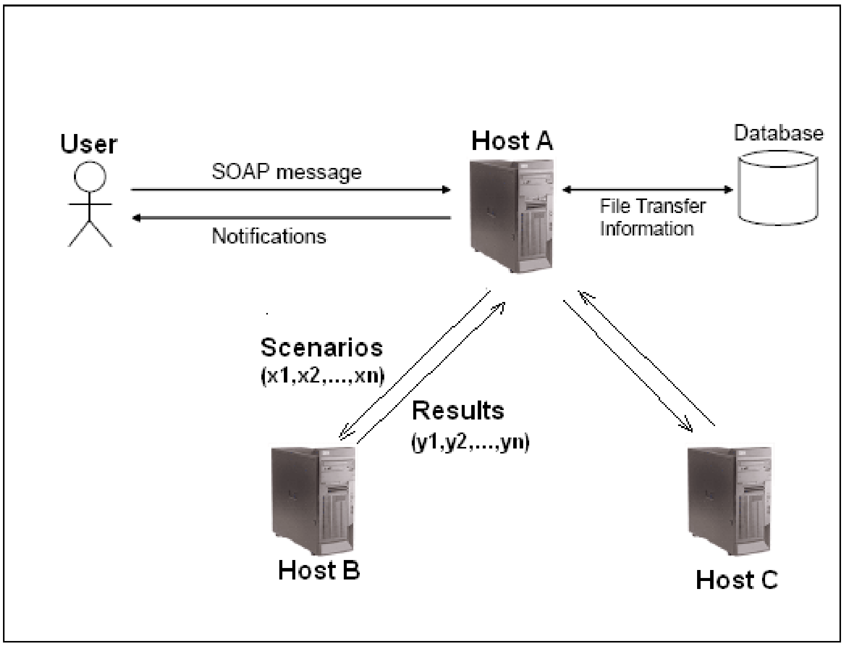

Fig. 2 Algorithm of the simulation in the Grid

IV. TEST AND RESULTS

The simulation ends with constructing the distribution of the evaluation function values into 40 numerical ranges and drawing the distribution chart:

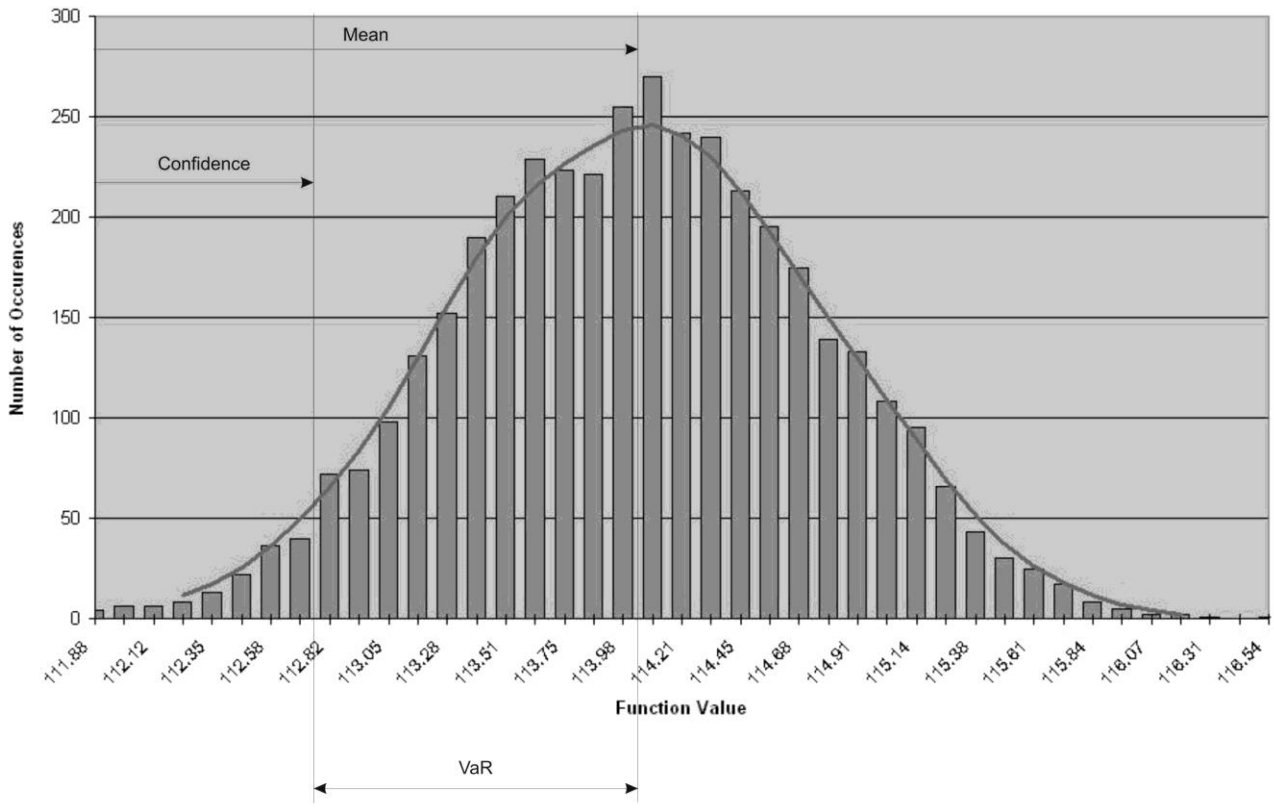

Fig. 3 Chart of the NPV distribution

The Value at Risk is calculated for two values of probability – 95% and 99%. The VaR is equal to the difference between the mean net present value and the value from the distribution ranges, responding to the relevant confidence percentage [4].

Result Analysis

Probability	Results	VaR
	114,093	
95,00%	112,888	1,205
99,00%	112,522	1,571

Fig. 4 Result Analysis

In [1] the simulation is tested on one computer only and in an environment, consisted of a pool of three connected computers, working under the control of the Condor high throughput computing system (implemented by the University of Wisconsin). The comparison data shows that the execution of the simulation in the pool achieves better time performance than the same simulation, executed on a single computer.
The following results of the execution of the same simulation process in the experimental Grid environment are obtained:

Simulation executed in the Condor pool		
Simulation step	**Time (sec)**	**%**
Producing the scenario set	9	16.57%
Submitting the scenario set to the machines	2.3	4.24%
Calculation	42	77.35%
Result evaluation	1	1.84%
Total	54.3	100.00%

Simulation executed in the Grid environment		
Simulation step in the Grid	**Time (sec)**	**%**
Request for certificates	3	4.98%
Certificate validation	3	4.98%
Producing the scenario set	9	14.93%
Submitting the scenario set to the machines	2.3	3.81%
Calculation	42	69.65%
Result evaluation	1	1.66%
Total	60.3	100.00%

Fig. 5 Results from testing the simulation in the Condor pool and in the Grid environment

The comparison data show that the execution in the Grid environment adds a new step in the whole simulation process – requests for user and host certificates and validation of certificates. This is the value of the secure usage of resources, distributed in different administrative domains.

V. CONCLUSIONS AND FUTURE WORK

The paper presents a further research of the authors’ work on the possibilities of the distributed computing technologies for salvation of compute intensive jobs. A Monte Carlo

simulation for evaluation the market risk of a financial position is implemented and tested in an experimental Grid environment.
The future work will be connected to the adoption and test of complex problems for distributed execution in real Grid structures.

VI. REFERENCES

[1] Penev I., Implementation of Monte Carlo simulation in a distributed computing environment, CompSysTech Gabrovo, 2008
[2] Atanasov E., Gurov T., Karaivanova A., Computing Grid: structure and applications, Automation and Information, 2006
[3] Haugh, M. The Monte Carlo Framework, Examples from Finance and Generating Correlated Random Variables. Course Notes, 2004
[4] http://riskmetrics.com: Long Run Technical Document
[5] http://www.ogf.org: Open Grid Forum
[6] http://www.globus.org: The Globus Project

ABOUT THE AUTHORS

Dr. Anatoliy Antonov
Eurorisk Systems Ltd.
31, General Kiselov Str., 9002 Varna, Bulgaria
E-mail: antonov at euroriksystems dot com

Ivaylo Penev
PhD student, Department of Computer Sciences and Technologies
Technical University of Varna