Bartosz SAWICKI[1], Artur KRUPA[2]

Warsaw University of Technology (1), Warsaw University of Life Sciences (2)

doi:10.15999/48.2019.05.28

Optimization of coil geometry using Monte Carlo method with HTCondor and Microsoft Azure technologies

Abstract. *The paper presents an application of modern computer services known as cloud computing for the simple coil geometry optimization problem. The Monte Carlo method is known for its robustness, but also low convergence. The latter shortcoming could be eliminated by large and affordable computational power offered today by cloud providers. The described architecture of the simulation system is based on Microsoft Azure platform with HTCondor as a job manager.*

Streszczenie. *Artykuł przedstawia wykorzystanie usług obliczeniowych na przykładzie prostego zagadnienia optymalizacji kształtu cewki. Metoda Monte Carlo jest znana ze swojej skuteczności, a jednocześnie z bardzo niskiej zbieżności. Wadę tą można skutecznie ograniczyć poprzez wykorzystaniem dużych i tanich mocy obliczeniowych oferowanych dzisiaj przez dostawców usług 'chmurowych' (ang. cloud computing). Opisana architektura systemu symulacyjnego oparta jest na platformie Microsoft Azure oraz zarządcy zadań HTCondor.* ***(Optymalizacja geometrii cewki metodą Monte Carlo przy wykorzystaniu HTCondor oraz Microsoft Azure)***

Keywords: cloud computing, optimal design, Monte Carlo method
Słowa kluczowe: obliczenia rozproszone, optymalizacja, metoda Monte Carlo

Introduction

An important trend in modern computational science is certainly cloud computing. Although this term has been created by the marketing departments in early 2000's, and a history of distributed and parallel processing in computer science is much longer, one could realize novelty in cloud computing approach. Utilization of commercially managed resources for scientific tasks has significant advantages, such as a flexibility to use different hardware, no need for infrastructure investments. On the other hand, important challenges are raised [1] and it is known that not all types the computing problems benefit from a loosely coupled architecture in the same way.

Groups of problems, which could be easily transferred into the cloud infrastructure are independent simulations, sometimes called as an 'embarrassingly parallel' problems. Sensitivity analysis [2], stochastic simulations [3] are just examples of the problems which require a large number of simulations. Another are DNA alignment in bioinformatics, 3D scene rendering in computer graphics or Monte Carlo methods, which are the main subject of this article.

The Monte Carlo methods are based on probing parameters space with the use of a random generator. Applications of such simple but robust solution are wide, and they are especially compatible with the structure of the cloud services [4]. Stochastic optimization using Monte Carlo sampling is one of them [5].

In this paper, stochastic optimization technique is used to design the magnetic coil system. The developed simulation platform is constructed using HTCondor embedded in the Microsoft Azure environment as presented in Fig. 1. Finite element method solver has been constructed using FEniCS library as described later in the paper. Coil design problem is a test case, which could be also successfully solved using gradient optimization techniques. However it should be treated as a benchmarking tool to study efficiency of the whole simulation system.

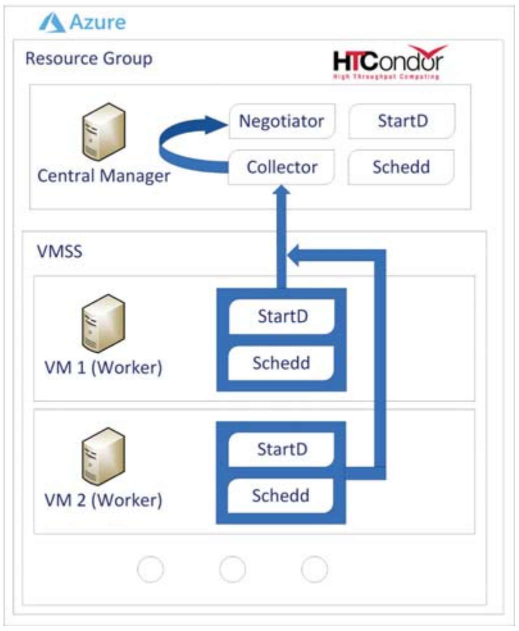


Fig. 1. Architecture of the system build using HTCondor running inside Azure cloud platform

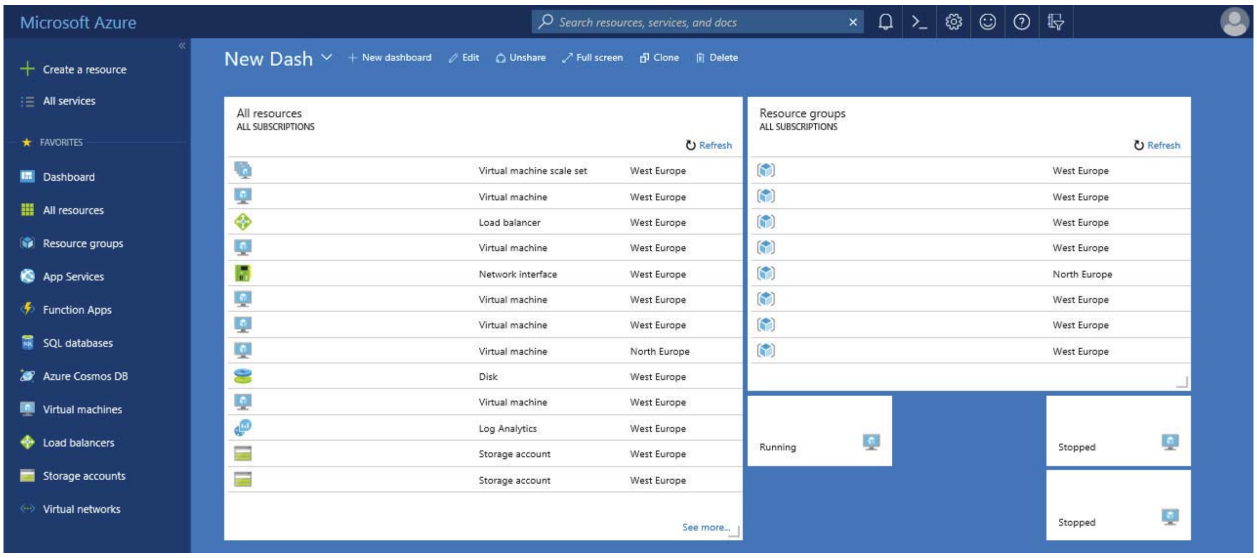
Simulation platform

A commercial offer of computational resources is wide. There are services, like local instances of VPSes (Virtual Private Servers) and DSes (Dedicated Servers) which are expensive when they are not in a constant use. Scientific numerical simulations are usually irregular and unpredictable by nature. That's why we drive our attention to cloud service providers, like AWS (Amazon Web Services) or GCP (Google Cloud Platform). As presented in Table 1 their prices are hourly based with variation related to the power of a rented machine.

It this research we decided to use services from Microsoft with their Azure platform. The main reason for that computational grant which we obtained from Microsoft company. Warsaw University of Technology has an agreement with Microsoft providing free access to services for teaching and scientific purposes.

The aim was to create a system based on fully configurable virtual machines, with its own CPU, RAM, and storage to use them as a fully functional standalone computer.

Azure is an environment in PaaS model (Platform as a Service), where many service are ready to use - open source and paid as well. One can configure them manually using from command line interface (PowerShell or CLI/Bash) or build system on preconfigured solutions with a wizards



All resources
ALL SUBSCRIPTIONS

Virtual machine scale set	West Europe
Virtual machine	West Europe
Load balancer	West Europe
Virtual machine	West Europe
Network interface	West Europe
Virtual machine	West Europe
Virtual machine	West Europe
Virtual machine	North Europe
Disk	West Europe
Virtual machine	West Europe
Log Analytics	West Europe
Storage account	West Europe
Storage account	West Europe

Resource groups
ALL SUBSCRIPTIONS

West Europe
West Europe
West Europe
West Europe
North Europe
West Europe
West Europe
West Europe

Running

Stopped

Stopped

Fig. 2. The main window of the Microsoft Azure web Portal

Table 1. Comparison of virtual machines for different providers

Cloud provider	Machine Type	CPU	RAM [GB]	HDD [GB]	Price [$/h]
Microsoft Azure	D2 v3	2	8	100	0.096
Microsoft Azure	D4 v3	4	16	200	0.192
Amazon WS	t2.large	2	8	30	0.056
Amazon WS	t2.xlarge	4	16	30	0.111
Google CP	n1-standard-2	2	7.5	30	0.067
Google CP	n1-standard-4	4	15	30	0.133

accessible by the Internet web-browser Portal[1] (see Fig. 2).

First and important step is to have a working Azure account with active budget. Free account can be obtained at website with $200 credit available to use in 30 days (credit card required to activate account). As an option can be reached voucher for 60 days with the same amount of money without necessity of having credit card. Contact with local Microsoft branch to require the latter option.

Having an active subscription and one can reach the Portal. After log-in, creation of your cluster could be started. Just click on a '+' symbol on a top-left side of a portal ("Create a resource").

The main part of the system is a job manager software. It is responsible for sending computational tasks into available machines, in such way to keep high utilization of the resources. In previous works [6] authors investigated Apache Hadoop as a job manager. It has proved to be flexible and efficient tool. However its main purpose are problems with unstructured big data sets. Many features provided by Hadoop does not match type of analysis required by the Monte Carlo methods. In presented optimization problem, large number of time-consuming simulations need to be solved. Solution will be performed by external application.

This kind of requirements are the perfect environment for High Throughput Computing system - HTCondor [7], which is designed to deliver large amounts of processing capacity over long periods of time. HTCondor is very stable and well known platform developed for over 30 years, and used in the leading computational centers all around the world.

HTCondor provides command line interface (CLI) which is much more efficient than graphical user interface (GUI). It is not because it is a natural form of a communication between computer and human, but because many parameters or functions are only available through commands which definitely influences into overall performance.

The GAHP project is a solution for an efficient communication between our HTCondor and Microsoft Azure [8]. Unfortunately at this moment, it has limited functionality to simple operations like creation of virtual machines or VM sets, starting them, listing and deallocating. It is promising solution of future, but after preliminary tests authors decided to stay with more stable solution and fully configurable environment without additional "connectors" inside our platform.

The heart of the developed solution is Azure Virtual Machines Scalable Set (VMSS), which can automatically start needed machines. To configure VMSS two virtual machines has to be created. One to be manager, and a second – a template for simulation workers with all preconfigured packages, sets, settings and features may be in use at anytime. After configuring worker machine, it has to be deallocated it and generalized. System and applications are universal, but each machine even having the same system image, has to have different, separate hard drive with personal configuration. This is to make drive image generalized. Details of the configuration process will be described in the following subsection.

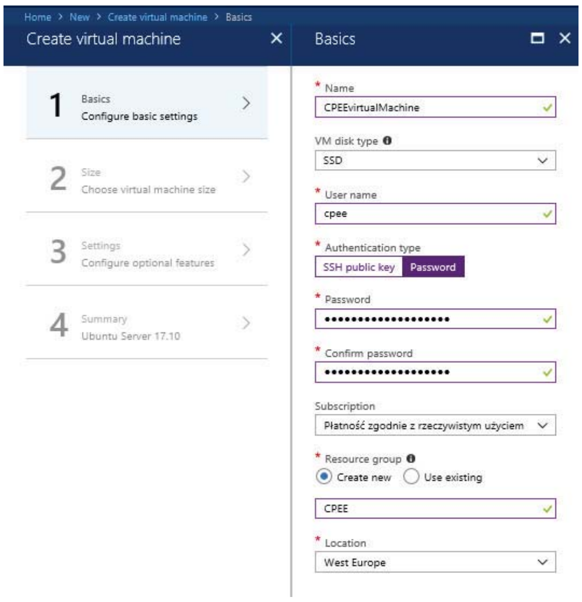
Configured distributed computing cluster is build on the top of virtual machines based on Ubuntu Linux open-source operating system. Creation of the simulation system require proper initialization of the Azure cluster with setting up HTCondor job manager and workers with simulation software. Description of this process will be divided into four phases: creation of virtual machines, setting up master node, creation of worker node template and finally creation of scalable cluster.

Create virtual machine

Having an active subscription and one can reach Azure Portal after log-in and start creation of your cluster. Clicking on a "+" symbol on a top-left side of a portal ("Create a resource") and choosing create virtual machine takes you to the dialog presented on the Fig. 3.

Then choose Ubuntu Server in a newest version (now. 17.10) and set server with a basics settings like machine name, user name, password and resource group for it.

[1] https://portal.azure.com/



Home > New > Create virtual machine > Basics

Create virtual machine

1 Basics — Configure basic settings

2 Size — Choose virtual machine size

3 Settings — Configure optional features

4 Summary — Ubuntu Server 17.10

Basics

* Name: CPEEvirtualMachine

VM disk type: SSD

* User name: cpee

* Authentication type: SSH public key [✓ Password]

* Password: ••••••••••••••••••

* Confirm password: ••••••••••••••••••

Subscription: Płatność zgodnie z rzeczywistym użyciem

* Resource group: [✓ Create new] Use existing

CPEE

* Location: West Europe

Fig. 3. View of the Virtual Machine creation window

Next step is to choose virtual machine size. Each size has estimated costs of usage per month. For this system it has been decided to choose one of cheaper machines, but not limiting necessary size of memory and number of cores. Machine D2 is equipped with 2 cores and 7 GB of RAM which is enough to run our HTCondor as a manager and central repository for all data gathering processes. A capacity of a system hard drive has been increased to 100 GB (from 64 GB by default). It is worth to disable "boot diagnostics" option, which takes additional disk space and makes the boot of the machine longer.

The last step is to wait for creation of a resource group with all required dependencies: virtual machine with hard drive, network interface with virtual network related to it and the most important – network security group.

After creation one should to make sure that new virtual machine is accessible from the same internal address in Azure network. Change IP to static for default assigned (10.0.0.4) as well as to be accessible from the Internet on the same name.

When machine is finished, it could be accessed by a ssh protocol. The system should be then updated and all necessary modules and packages installed.

Set-up Master node

Having fully updated operating system, installation of job manager with command should start:

```
sudo apt-get install condor
```

After that one needs to reconfigure HTCondor itself [9]. There are not much options to configure. The most important settings are listed below:

- Daemon List: Schedd, Collector, Negotiator and Master
- Condor Admin: *username@internal_ip*
- Filesystem Domain: $(FULL_HOSTNAME)
- Condor Host: *internal_ip*
- Allow Write: *

Because of the high number of the tasks planned in the experiment, it is required to add two variables to exceed limits of jobs per task and per user:

- Max Jobs Per Submission: 100000
- Max Jobs Per Owner: 200000

Final step is to create SSH key to share it among all possible machines in a pool:

```
ssh-keygen -t rsa
```

Create Worker node template

Repeat almost all the steps from beginning (creating virtual machine) to the step of a connection. For worker template machine don't set any internal IP nor set DNS name. After successful connection the machine has to be configured as Worker for HTCondor Manager node. Set values as stated below:

- Daemon List: Startd and Master
- Condor Admin: *username@mgr_internal_ip*
- Filesystem Domain: $(FULL_HOSTNAME)
- Condor Host: *internal_ip*
- Allow Write: *
- Trust UID Domain: TRUE

Then all additional applications necessary for simulations have to be installed. Optimization task described in this paper is about magnetic field distribution using Finite Element Method. To solve such problem library FEniCS [10] had been used, and it has to be installed on every working node.

HTCondor service should be set to start with system boot every time. It is required because automatic scalable machines does not have memory.

To be sure everything is working we checked on Manager status of an existence working machines with command:

```
condor_status
```

which should give a summary of all working machines connected to Manager. Only then we are sure that system is working properly and can be generalized. Template machine needs to be clean without any dependencies on account – clean system with only installed applications. It should be done with a command:

```
sudo waagent -deprovision+user -force
```

After turning off a virtual machine, one could prepare generalization of an image, which will be ready to deploy at the Azure infrastructure. Generalization is done by the command:

```
az vm generalize
```

The hard drive address of deallocated virtual and generalized machine has to be remembered. It will be used to create a virtual machine scale set.

Create scalable cluster (VMSS)

Final step is to create Virtual Machine Scalable Set (VMSS) with PowerShell interface exactly with a command "az vmss create" presented on Fig. 4. Use remembered VHD URI as a source image of a system drive and used previously generated SSH key to be sure, that each newly created machine in a set is able to connect to the Manager.

When VMSS cluster is ready, the any number of virtual machines can be set. One can choose size of them (from various number of dedicated solutions) and see status of performance, errors or notifications using metrics sub-tab.

The VMSS is a scalable solution working with any configuration, number of instances and sizes. In our case the only limitations is number of instances (40) and number of cores (500). They are related with our license agreement and

```
az vmss create -n <SET_NAME> -l <REGION> -g <RESOURCE_GROUP> --use-unmanaged-disk
--instance-count <NUMBER> --image "https://<STORAGE_ADDRESS>.blob.core.windows
.net/<STORAGE_HARD_DRIVE>.vhd" --os-type linux --authentication-type ssh --admin-username
 <USERNAME> --ssh-key-value <SSH_KEY_STRING>
```

Fig. 4. Shell command to create Virtual Machine Scalable Set

could manually be exceeded with the help of Azure support team.

Performance tests

Performance advantages of the system containing many computational units are obvious. However any distributed computing system suffers from issues related with its network complexity. Data transfer, coordination of tasks, changeable structure are the main reasons which deteriorate computational power.

Maximum theoretical speedup is equal to the number of cores connected to the system. For problems consisting of many independent tasks (so call "embarrassingly" parallel problem) such theoretical speedup is nearly achievable. To measure how close we are to the perfect solution, we used overhead factor as it is formulated on equation:

$$\text{(1)} \quad \textit{overhead} = \frac{\textit{time\_real}}{\textit{time\_one\_task} \times \textit{number\_of\_tasks}} - 1$$

Our previous research showed that systems based on Apache Hadoop software have overhead on the level 20-30% [6]. We constructed similar experiments with HTCondor.

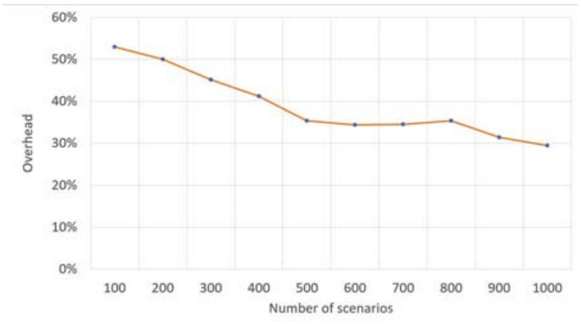


Fig. 5. Performance overhead for different numbers of simulations

The set of simulations had been performed at least three times to calculate average and eliminate unpredictable nature of public service environment. Results for simulations with different number of cases are presented on Fig. 5. One can observe expected effect, that overhead drops if problem consists of higher number of cases.

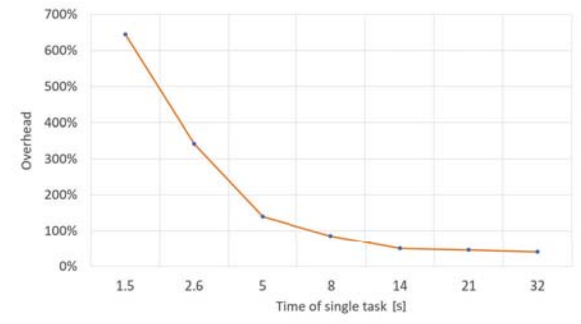


Fig. 6. Performance overhead as a function of single simulation time

On the plot of Fig. 6 influence of single simulation time is analyzed. The longer time means that computing node has better utilization and less often data are transmitted over the network. If single task is shorted that 5 seconds, overhead is rapidly growing and its value exceeds 100%. It does not mean that for such cases parallelization is not justified. One just have to be aware that more that half of computer power is wasted. What is especially important when public cloud computing services are used, where the fee is directly related with time of machine allocation, and not with the parameters of utilization.

Overhead observation has been confirmed by our largest experiment. Cluster consisting of 40 machines with 8 cores CPU each created network of 320 working nodes. Theoretical speed up of such system should be 320x. Single simulation takes 5 seconds, so 100,000 cases should take 500,000 seconds (139 hours). HTCondor system was able to solve all the simulation with about 2 hours, which gives overhead value approx. 150%.

Optimization problem

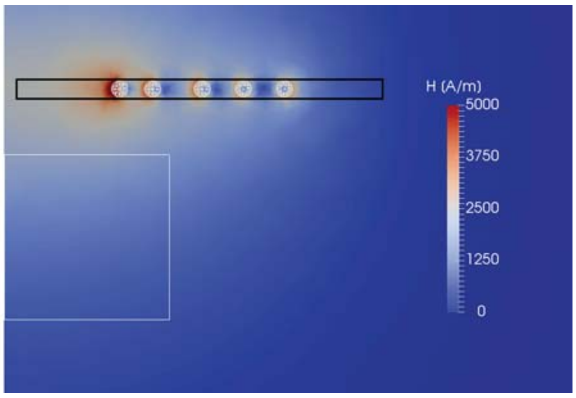


Fig. 7. Initial point of the coil optimization. Distribution of the magnetic H-field in the 2D cylindrical coordinate system. Black rectangle is to show optimization constrains.

Designing problem used to validate developed platform is a simple magneto-static case. As seen on Fig. 7 coil is made of five toroidal rings (cylindrical coordinate system). The case was inspired by the magnetic stimulation coil used for magnetic fluid hyperthermia [11]. Initial radii of the coils (see Fig. 7), which are subject of optimization are taken directly from the mentioned Yamada research.

Diameter of the wire and current density (**J**) are given and fixed. Designing variables are radii of the coil rings, and objective function is to maximize energy by the field into the region marked by the white square. Black rectange shows constrains of designing variables.

Distribution of the magnetic field could be described by the vector laplacian equation (2) for magnetic vector potential (**A**).

$$\text{(2)} \quad -\frac{1}{\mu}\nabla^2 \mathbf{A} = \mathbf{J}$$

where μ is magnetic permeability.

Objects are axis-symmetric, so significant reduction of the complexity could be achieved by transferring equation (2) into 2D. Source current density has only one J_φ component, so in cylindrical coordinate system $(r,\ \varphi)$ problem is described by following equation:

$$-\frac{1}{\mu}(\nabla^2 A_\varphi - \frac{A_\varphi}{r^2}) = J_\varphi \tag{3}$$

Then magnetic induction (**B**) has to be calculated:

$$\mathbf{B} = \nabla \times \mathbf{A} \tag{4}$$

$$\mathbf{B} = [-\frac{\partial A_\varphi}{\partial z}, \frac{A_\varphi}{r} + \frac{\partial A_\varphi}{\partial r}] \tag{5}$$

Objective function is to maximize magnetic energy W provided into the selected region.

$$W = \int \frac{\mathbf{B}^2}{2\mu}\,\mathrm{d}v \tag{6}$$

The problem is solved using FEniCS library with zero boundary condition on a large bounding box. Single simulation takes approx. 5 seconds. Convergence of Monte Carlo method is low, so 100,000 of trials where needed to obtain significantly better result. The best solution is presented on Fig. 8. Value of the energy in the aim area has been increased by 26% comparing with original shape of the coil. One can realize that radii of coils are nicely matched with the size of integration area. However small improvement related with the position of the fifth coil is visible.

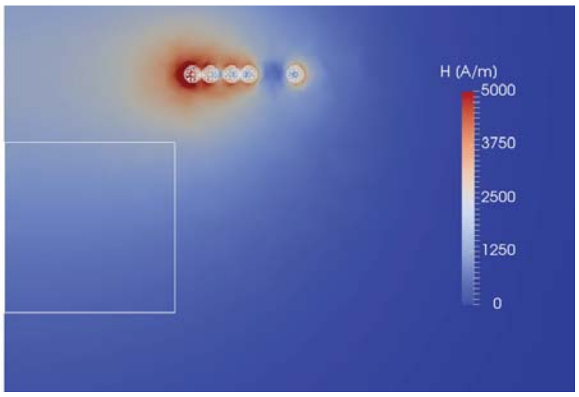


Fig. 8. The best coil design found after 100,000 simulations with Monte Carlo method. Objective function was to maximize energy provided into a area marked by the white square.

Cost of 100,000 simulations running on developed cloud platform is low. In our largest experiment with 40 instances * 0.049 EUR/h * 4 hours of work (including time to set-up cluster) it was 7.84 EUR.

Conclusions

We demonstrated how virtual computational resources could be utilized for the real electromagnetic design process. HTCondor running on the Microsoft Azure proved to be an effective solution, however overhead is not negligible. Moreover, the preparatory phase for the platform is quite complicated.

Full benefits of computations running on external cloud services could be reached when the scale of a problem exceeds the capabilities of the local machine. Another recommendation for the system usage is a prediction of its multiple reuses for different Monte Carlo simulations.

Authors: *dr hab. inż. Bartosz Sawicki, Politechnika Warszawska, Instytut Elektrotechniki Teoretycznej i Systemów Informacyjno-Pomiarowych, ul. Koszykowa 75, 01-650 Warszawa, E-mail: bartosz.sawicki@ee.pw.edu.pl; mgr inż. Artur Krupa, Szkoła Główna Gospodarstwa Wiejskiego, Wydział Zastosowań Informatyki i Matematyki, ul. Nowoursynowane 159, 02-776 Warszawa, E-mail: artur_krupa@sggw.pl.*

REFERENCES

[1] Angelo G., Marzolla M., New trends in parallel and distributed simulation: From many-cores to Cloud Computing, *Simul. Model. Pract. Theory*, 126 (2014)
[2] Krupa A., Sawicki B., High-resolution scatter analyse using cloud computing, *Przeglad Elektrotechniczny*, 91 (2015), Nr 12, pp. 140-142
[3] Krupa A., Sawicki B., Measurement-based stochastic models of biological materials, *18th International Conference on Computational Problems of Electrical Engineering (CPEE 2017)*, Kutna Hora, 11-13th September 2017
[4] Wang H., Ma Y., Pratx G., Xing L., Toward real-time Monte Carlo simulation using a commercial cloud computing infrastructure, *Phys. Med. Biol.*, 56 (2011)
[5] Homem-de-Mello T., Bayraksan G., Monte Carlo sampling-based methods for stochastic optimization, *Surveys in Operations Research and Management Science*, 19 (2014)
[6] Krupa A., Sawicki B., Massive Simulations Using MapReduce Model, *IAPGOS*, 5 (2015), No 4, pp. 45-47
[7] Thain D., Tannenbaum T., Livny M., Distributed computing in practice: the Condor experience, *Concurrency - Practice and Experience*, 17 (2005), pp. 323-356
[8] Srirangam R., Patil R., Azure GAHP Server for HTCondor, *Microsoft*, (2017)
[9] Rynge M., Desinghu B., Thapa S., Harstad E., Large Scale Computation with HTCondor, *SWC-OSG Workshop materials*, (2017)
[10] Alnæs M. S., Blechta J., Hake J., et. al., The FEniCS Project Version 1.5, *Archive of Numerical Software*, 3 (2015), No. 100, pp. 9-23
[11] Yamada S., Ikehata Y., Ueno T., Nagae H., Wireless Power Transfer System for Hyperthermia Therapy, *Proceedings of the 2nd Frontiers in Biomedical Devices Conference*, (2007)