УДК 004.94+004.771 DOI: 10.31474/2074-7888-2020-1-22-27

INTEGRATION OF DISTRIBUTED PARALLEL SIMULATION ENVIRONMENT WITH CLOUD-INFRASTRUCTURES

Shcherbakov O., Polsterer K., Svjatnyj V.

This article describes an integration of the subsystems of the distributed parallel simulation environment with cloud infrastructures. A complex support for simulation of the dynamic network object with distributed parameters on Amazon AWS cloud is provided. As well as a tool helping to significantly save running costs for cloud simulations.

Keywords: *distributed parallel simulation environment, cloud computing, AWS spot-market, cost optimization.*

Introduction. Cloud computing market is rapidly growing. Recent studies [1] predict that spending on cloud services will grow by 16.5% in 2016 – up to $204 billion. The largest increase is expected for cloud services (infrastructure as a service [IaaS]). Cloud in [2] is defined as “a type of parallel and distributed system consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resource(s) based on service-level agreements established through negotiation between the service provider and consumers.”

In [2], and then [3] and others cloud computing called a „fifth utility “(after water, electricity, gas, and telephony), because it is necessary to meet the needs of everyday life and should be available at any time when needed, and be paid for the volume of services used.

The concept of integration of DPSE with cloud-infrastructures. Works [4, 5, 6, 7] describe a distributed parallel simulation environment (DPSE) as a user-friendly system organization of a collaboration of hardware, parallel computing resources, system- and modeling-software, which supports development, implementation and usage of models of complex dynamic systems with distributed and concentrated parameters. DPSE is described as a comprehensive system of hardware and software (system and special). In this aspect a decomposition of DPSE into subsystems was proposed, what on one hand simplifies the development and support of DPSE through a modular approach, on the other hand – preserves the logical and functional integrity of a system being developed.

To determine the aspects of integration of DPSE with cloud infrastructures consider the representation of DPSE on the Software as a Service (SaaS) principle. In this case, the user does not need to own any computing resources. Access to the simulation environment is done through the browser, if necessary specialized applications can be used. Hardware components of such DPSE may include: servers available to the DPSE administrator; HPC-clusters belonging administrator or a particular user, or those to which they have access; cloud resources ordered by administrator or user, or those that can be ordered; user clients (hardware) can be used with limitations. We will later consider the case of “zero ownership” – when DPSE administrator does not own any hardware resources, but can order the necessary resources by cloud providers.

There are different requirements to the subsystems determining hardware resources necessary for their work. Some subsystems have to be always available – *dialog, databases, IT-support*. Others may work “on demand”. According to the needs in computational power we can distinguish subsystems that require a lot of computing resources (including parallel distributed ones) and those for which the parallelization of work is impractical.

Depending on the type of user access, the type of used hardware resources and the type of available HPC resources, following requirements and restrictions may apply due to network security.

The “interface subsystems” – those that provide network interaction with users and remote resources – are located on a server in the local network. DPSE will be accessed only by local users. There are no restrictions on the usage of local and remote HPC-resources. Usage of cloud resources can have slight restrictions – it is not possible to receive notifications about status changes of the cloud due to local network security policies, such information must be retrieved and/or updated on time intervals. Disadvantages of the configuration: external users can not access DPSE, restrictions on feedback from the cloud (and other external) services. Advantages: making external access impossible reduces the risk of unauthorized access, such system can have no restrictions on access to local HPC-resources.

The “interface subsystems” are located in the demilitarized zone (DMZ) [8, p. 388], with external access available. External and local users will have access to DPSE, usually through a global network. Usage of local HPC-resources is possible for the resources in this (or other neighboring) DMZ only. Access to remote HPC and cloud resources is possible, but it is important to keep in mind that in case of DPSE server being compromised, accounts used to access any resources (including cloud) will also be compromised.

The “interface subsystems” are located on servers of external (including cloud ones) provider. Choosing a placement strategy of the subsystems is guided by the availability of hardware resources, funds and human resources to support – systems with an external access are potentially vulnerable and require constant maintenance and keeping their software up to date.

DPSE subsystems that do not need to be constantly available – equations solver, visualization, etc. – can be started “on demand”, usually on HPC resources. Running all subsystems that do not use parallelization on the same server is possible, but it can cause high server load causing difficulties with access to the DPSE.

Consider a typical scenario of working with DPSE.

1. User being authorized in the system.

2. Selects, load or describes the network desired topology of a network object.

3. Topology being analyzed.

4. Equations describing a given topology being generated in a convenient form for simulation.

5. The analysis of available (to the user) and of necessary parallel simulation resources is being made.

6. Generated equations being adapted to available resources.

7. Prepared simulation model being sent to the queue for running on the HPC.

8. Simulation results being post processed and stored.

9. Results being presented to the user.

For this scenario DPSE needs the following hardware resources for its work: a server for subsystems that provide interaction with the user and other subsystems; resources for other subsystems – HPC-resources for parallel simulation, other available servers.

Consider the following scenario of use: access to DPSE is available to external users, thus DPSE server is located outside the security perimeter and has no access to the local cluster. DPSE can run parallel simulations on HPC resources available to its users or can order computing resources by cloud providers. We consider DPSE integration with cloud services of Amazon. A significant advantage of this provider for scientific simulations is the offer of spot-instances [9] – free resources are sold on the principle of trading. This allows the user to save a lot. Stated that the price of these instances is defined by bets of all users of the service and the number of free instances offered for trades. The disadvantage of this proposal consist in having no guarantees of continuous work of a spot-instance. If the auction price rises above the bet of the user, his instance(s) will be shut down and given to someone else. There is an automatic notification shortly before the shut down but it is not guaranteed.

DPSE subsystems for *dialog*, *databases* and *IT-support* must always be available to users, thus they may run only on standard “on demand” instances. Other DPSE subsystems related to the preparation of simulation models, simulation and processing of results, can be run on spot-instances and other HPC-resources available to the user. Interrupting their work is less critical, in addition their algorithm may provide a mechanism of restoring the interrupted simulation (checkpointing). It is worthwhile to order cloud resources for these subsystems only when users run a simulation and to free them when the simulation is complete.

Prototype of the simulation environment DPSE – cloud-services. Work with cloud resources on more or less complex systems needs automation. First of all placing the bets must be automated for spot-instances when starting the simulation, deleting unused instances after simulation, analysis of prices for spot-instances.

A prototype of DPSE was developed [10]. It runs completely in cloud environments and provides support for the simulation of dynamic network object (DNO) at all stages – from model development to retrieving the results after parallel execution on cloud resources. Amazon AWS Services are used as a cloud provider. Ten subsystems of DPSE, as described in [4, 5, 6, 7], are represented in the various components of the prototype. The subsystems for the *IT-support* and *dialog* run on the EC2 on-demand instance. Amazon RDS service represents both software and hardware parts of the *database* subsystem. Subsystem for *load balancing* as well as a subsystem for *data exchange* are based on the CfnCluster [11] framework. Gnuplot [12] is used as a simple implementation of the *visualization* subsystem. Other DPSE subsystems – for *topological analysis*, for *equations generation*, for *virtual parallel simulation models*, for *parallel equation solvers* – provide at first hand the simulation of DNO. They run on a cloud cluster, running simulation on other HPC clusters is also supported. Software parts of these subsystems are briefly described below.

Currently the DNO model with distributed parameters is implemented and available for all users. A model developer has the possibility to prepare a description of different topologies. This description is provided as a text file in a specific format. A user can select one of the available topologies for simulation and define different model parameters: numerical method, maximum number of parallel processes and so on. Whereat the topology analysis, equations generation and analysis of virtual parallel simulation-models are performed automatically for the model with these parameters.

The numerical solution of the Cauchy problem (for previously generated equations) is provided by GNU Scientific Library (GSL) [13], which is quite popular in the scientific community (over 1000 citation of GSL documentation according to Google Scholar), is cross-platform and has a simple and well structured source code. GSL can be easily extended with the implementation of other numerical methods. Implementation of the block methods is planned for the future [14, 15]. The results of the simulation are visualized by Gnuplot as three-dimensional plots and stored in a git repository. This way of storing and accounting the results is not suitable for models with huge amount of output data, and for long time storage, but it is implemented to demonstrate the work of DPSE and to simplify models debugging. Results of the simulation are stored in a local (DPSE server) repository with mirroring to the external one – GitHub and AWS CodeCommit. The question of optimally storing and managing simulation results is still open.

Besides DNOs other types of models are supported and can be launched, as well as creating and managing standalone cloud clusters. User can specify the desired cluster configuration to be created, and a maximum price for the spot-instances. Because CfnCluster framework instances that are not used are automatically turned off and are turned on again when required, costs and administrative time is saved.

A variety of additional DPSE features improve the work of simulation model developers and users. The developed prototype has a built-in version control system (git) for the models, supports accounting of experiments on HPC and allows to open terminal sessions directly in a browser. Instant notifications on a mobile phone about the status of cloud cluster and simulations helps to organize simulation workflow and to save time.

A function for monitoring and analysis of spot instance prices allows user to understand the dynamic (fig. 1) of AWS spot prices and helps to choose cheapest spot-instances or to choose a proper bet for certain instance. Fig. 2 shows an example price comparison, instances are sorted by the rising cost of virtual performance (conditionally “price per gigahertz”).

Assume we have a model to launch on the cloud cluster with the lowest budget. The model has been executed with test data on different types and different numbers of instances. For simplification we take into account only two types of EC2 spot-instances: *c4.large* and *c4.8xlarge*. *Frankfurt* is chosen as an AWS region. Execution time of this simple benchmark model in minutes is shown in table 1. Having this benchmark done helps to choose a better type of instances with respect to cost/performance for models of this type and to make a better parallelization. Table 2 shows a price comparison for one simulation running on the *onDemand*- and on *spot*-instances. Full utilization is assumed. Average spot prices were computed by the spot-market analysis tool of DPSE (fig. 2).

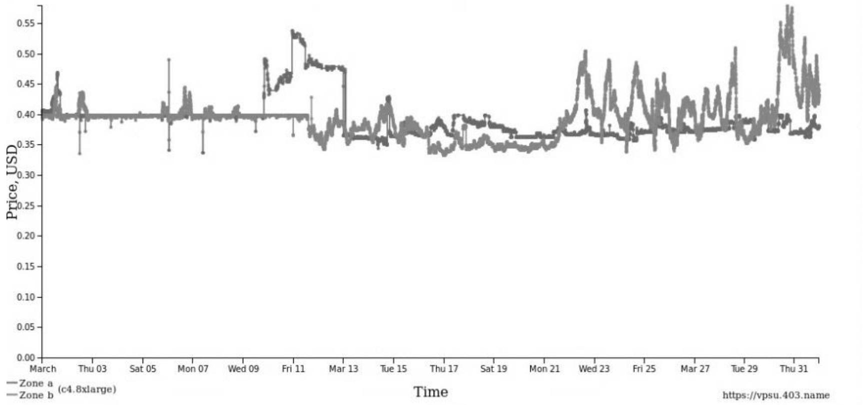
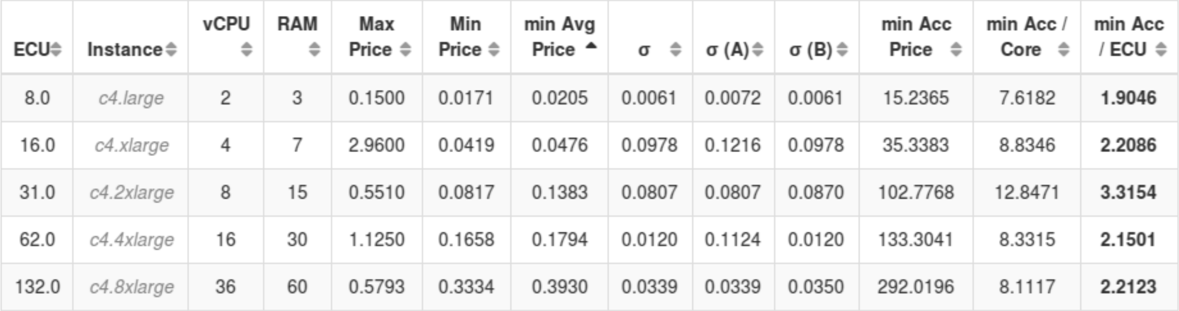


Fig. 1. Spot-price changes for *c4.8xlarge* instance in March 2016, region *Frankfurt*



ECU	Instance	vCPU	RAM	Max Price	Min Price	min Avg Price	σ	σ (A)	σ (B)	min Acc Price	min Acc / Core	min Acc / ECU
8.0	c4.large	2	3	0.1500	0.0171	0.0205	0.0061	0.0072	0.0061	15.2365	7.6182	**1.9046**
16.0	c4.xlarge	4	7	2.9600	0.0419	0.0476	0.0978	0.1216	0.0978	35.3383	8.8346	**2.2086**
31.0	c4.2xlarge	8	15	0.5510	0.0817	0.1383	0.0807	0.0807	0.0870	102.7768	12.8471	**3.3154**
62.0	c4.4xlarge	16	30	1.1250	0.1658	0.1794	0.0120	0.1124	0.0120	133.3041	8.3315	**2.1501**
132.0	c4.8xlarge	36	60	0.5793	0.3334	0.3930	0.0339	0.0339	0.0350	292.0196	8.1117	**2.2123**

Fig. 2. Prices analysis results for spot instances in March 2016, region *Frankfurt*

Table 1. Simulation benchmarking for different types of instances

# of instances	Instance type	Duration (min)	Runs/hour
1	c4.large	14	4.3
2	c4.large	8	7.5
3	c4.large	6	10
4	c4.large	5	12
6	c4.large	5	12
1	c4.8xlarge	1	60

The performance of the simulations on 4 and 6 instances is the same because of relatively slow interconnections of instances done with 1Gbit-Ethernet and intensive data exchange during the simulation of this model.

Table 2. Prices per run for the benchmarked model

# of instances	Instance type	*OnDemand* price/run	*Spot* price/run
1	c4.large	0.0312	0.0048
2	c4.large	0.0357	0.0055
3	c4.large	0.0402	0.0062
4	c4.large	0.0447	0.0068
1	c4.8xlarge	0.0356	0.0066

Values for 6 c4.large instances are removed as they are redundant and will wrongly increase a calculation of the resulting benefit.

Table 2 lets to compute the benefit from using spot-instances. For running a test model in March 2016 it can be about 650%. Running a benchmark on the instances of different type and choosing the best one lets to benefit by about 140%. Both together can give 910%.

At the moment of writing this article an automation for described benchmarking and further research on cost optimization are under development.

Conclusions. The growing popularity of cloud service providers is caused by their affordability and flexibility. For scientific purposes the possibility to order unused resources with a substantial discount (spot instances), for tasks not requiring real-time simulation, but still having need for powerful (CPU speed, available memory, etc.) resources, seems to be attractive.

A prototype of DPSE providing support for all phases of modeling and simulation of dynamic network objects and other models was developed. Running DPSE entirely on cloud resources allowed to prepare the ground for further experiments on the feasibility of using cloud resources for different classes of simulations.

An example of saving up to 910% of simulation costs for running a certain model in cloud was shown. This was achieved by choosing an optimal type of instance for the presented simulation model and by carefully observing the prices on the spot-market. This benefit will differ for other types of models.

Calculations was made without taking into account the prices for EBS storage, EBS I/Os and data transfer for downloading the results of the simulations. This will be done in future research.

References

[1] Gartner says worldwide public cloud services market is forecast to reach 204 billion in 2016, 01 2016. http://www.gartner.com/newsroom/id/3188817 (retrieved 29.03.2016).

[2] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation computer systems, 25 (6):599–616, 2009.

[3] Jose Moura and David Hutchison. Review and analysis of networking challenges in cloud computing. Journal of Network and Computer Applications, 60:113 – 129, 2016.

[4] L.P. Feldmann, V.A. Svjatnyj, M. Resch, and M. Zeitz. Forschungsgebiet: Parallele Simulationstechnik. Modelling and Computer Graphics - 2009, 2009.

[5] L.P. Feldmann, V.A. Svjatnyj, M. Resch, and M. Zeitz. Forschungsgebiet: Parallele Simulationstechnik. In ASIM 2014 – 22. Symposium Simulationstechnik, pages 3–7, 2014.

[6] V. Svjatnyj, V. Kushnarenko, O. Shcherbakov, and M. Resch. Dekomposition der verteilten parallelen simulationsumgebung. Scientific papers of Donetsk National Technical University. Series "Problems of modeling and design automation", 2012.

[7] O. Shcherbakov and V. Svjatnyj. Decomposition into subsystems and organization of work of distributed parallel simulation environment on the web. Informatics and Computer Technologies, 2010.

[8] E. Maiwald. Network Security: A Beginner’s Guide, Second Edition. Beginner’s Guide. McGraw-Hill Companies,Incorporated, 2003.

[9] Amazon. Ec2 spot instances. https://aws.amazon.com/ec2/spot/.

[10] Oleksandr Shcherbakov. Distributed parallel simulation environment. https://vpsu.403.name/.

[11] AWS. Cfncluster. https://aws.amazon.com/hpc/cfncluster/.

[12] Thomas Williams, Colin Kelley, et al. Gnuplot 4.4: an interactive plotting program. Official gnuplot documentation, http://sourceforge.net/projects/gnuplot, 2010.

[13] Brian Gough. GNU scientific library reference manual. Network Theory Ltd., 2009.

[14] A. Guseva and V. Kushnarenko. Discrete simulation-model of dynamic network object with distributed parameters based on the block numerical method. Scientific papers of Donetsk National Technical University. Series "Problems of modeling and design automation", 9(179):356, 2011.

[15] V. Kushnarenko. Simulating a branch of dynamic network object with lumped parameters based on the block numerical methods. Scientific papers of Donetsk National Technical University. Series "Problems of modeling and design automation", (210):66–74, 2013.

Надійшла до редакції 25.04.2019 р.

Shcherbakov Oleksandr – PhD-student, Heidelberg Institute for Theoretical Studies (Schloss-Wolfsbrunnenweg 35, Heidelberg, 69118, Germany).
E-mail: oleksandr.shcherbakov@h-its.org.
Kai Polsterer – Dr., Heidelberg Institute for Theoretical Studies (Schloss-Wolfsbrunnenweg 35, Heidelberg, 69118, Germany).
E-mail: kai.polsterer@h-its.org.
Svyatnyy Volodymyr Andriiovych – Doctor of Technical Sciences, Professor, State Higher Education Establishment “Donetsk National Technical University”, (Shybankova square, 2, Pokrovsk, Donetsk region, 85300, Ukraine).
E-mail: vsvjatnyj@gmail.com

WEB APPLICATION DEVELOPMENT FOR MODELING BULK MATERIALS BY DISCRETE ELEMENTS METHOD

If the language of the article is different from Ukrainian, an annotation (not less than 1800 characters) and a list of keywords are given in Ukrainian. If the language of the article is different from English, an annotation (not less than 1800 characters) and a list of keywords in English are given. The abstract should be contain information about the purpose, methods and results of the research, scientific novelty and / or practical significance of the results. The main text of the article should be structured and consist of the following sections: introduction, analysis of recent research and publications on the topic, purpose of the research, materials and methods, results and their discussion, conclusions. The total volume of the article is 6-10 pages.

Ключові слова: *MDE, bulk materials, modeling, mobility, simulation.*

Щербаков Олександр Сергійович – аспірант, Гейдельберзький університет (Schloss-Wolfsbrunnenweg, 35, м. Гейдельберг, 69118, Німеччина).
E-mail: oleksandr.shcherbakov@h-its.org.
Кай Польстерер – доктор наук, Гейдельберзький університет (Schloss-Wolfsbrunnenweg, 35, м. Гейдельберг, 69118, Німеччина).
E-mail: kai.polsterer@h-its.org.
Святний Володимир Андрійович – доктор технічних наук, професор, Державний вищий навчальний заклад «Донецький національний технічний університет» (пл. Шибанкова, 2, м. Покровьск, 85300, Україна).
E-mail: vsvjatnyj@gmail.com