

## INVESTIGATING WS-PGRADE WORKFLOWS FOR CLOUD-BASED DISTRIBUTED SIMULATION

Nauman R. Chaudhry, Athar Nouman, Anastasia Anagnostou, Simon J.E. Taylor

Department of Computer Science  
Brunel University London  
Uxbridge, Middlesex, UB8 3PH, UK

### ABSTRACT

Cloud computing enables access to computing infrastructure as a utility without the associated high cost. The impact of Cloud computing on simulation software is just beginning with some simulation vendors using Cloud “solutions” in a similar manner to Web-based Simulation systems. CloudSME Simulation Platform that supports simulation software to be deployed as service is supported by a cloud platform. Distributed simulation has the simple goal of interoperating models together over a network to either (a) build larger models and/or (b) share the processing of a large model. Distributed simulation uses technology typically based on the IEEE 1516-2010 High Level Architecture. Cloud computing allows for the elastic provision of multiple processing nodes, as specified by the simulation end user, based on factors that include the number of simulation runs required, the time available for experiments, and the costs of processing. In this study, we presents a framework for cloud-based distributed simulation.

### 1 INTRODUCTION

Computing power as a utility was introduced in the 1960s. “Cloud computing” refers to access internet-based computing resources first appeared in the mid-2000s. The National Institute of Standards and Technology (NIST) has made efforts to standardize the terminology of Cloud computing. The cloud computing paradigm attracts increasing numbers of Modeling & Simulation (M&S) practitioners wishing to perform their simulations in the cloud. Distributed simulation (DS) can be defined as the distribution of the execution of a simulation program across multiple processors (Fujimoto 2000). The idea behind DS is to either integrate separately designed and developed models or to distribute a model to enhance its performance for accumulating faster results. The IEEE 1516-2010 High Level Architecture (HLA) (IEEE 2010) specifies rules, data interfaces and processes to support DS.

### 2 APPROACH

To enable the execution of DS on the cloud, we propose a framework that simplifies the DS deployment using workflows. The main requirement in the proposed framework is the cloud deployment of the Runtime Infrastructure (RTI), implementation of IEEE 1516-2010, and the simulation package with their dependencies. In our case, we use poRTIco, an open source, fully supported, cross-platform RTI, and Repast Symphony simulator. Both require java runtime installation. The deployment was implemented on an academic cloud provided by the University of Westminster, UK that runs OpenStack middleware and is supported by the CloudSME Simulation Platform (CSSP). This cloud resource supports only Linux applications therefore a Shell Script executable is used to run the DS. Secondly, we need to create the workflow to describe the sequence of tasks to run the DS. WS-PGRADE, developed by MTA SZTAKI (Kacsuk et al. 2012) as an open source software, provides a graphical environment for workflow creation and a web interface for workflow configuration. Figure 1 illustrates the workflow in order to execute the DS. “Initialize”, “Manager”, “Model-1”, “Model-2”, ... “Model-n”, “Execution” and “collect results” are all job nodes. Each job

node has its own input and output ports. “Initialize” job node contains one input and one output port. This job takes a text file, contains the list of federates name to be part of the DS federation, as an input and pass it Manager job node through output port. Manager job node has three input and one output ports. This job node represents the Manager federate that ensures that all federates has joined the federation when starting the execution of the DS. “Manager” job node prepares the Manager Federate executable and pass it to the “Execution” job node. “Model-1”, “Model-2”, ... “Model-n” job nodes contain two inputs and one output ports. Users provide the model (model.tar file) and its data (input.tar file) in the input ports 1 and 2, respectively. “Execution” job node takes all the required files in order to run the distributed simulation. Finally, “Collect Results” job node collect the final results.

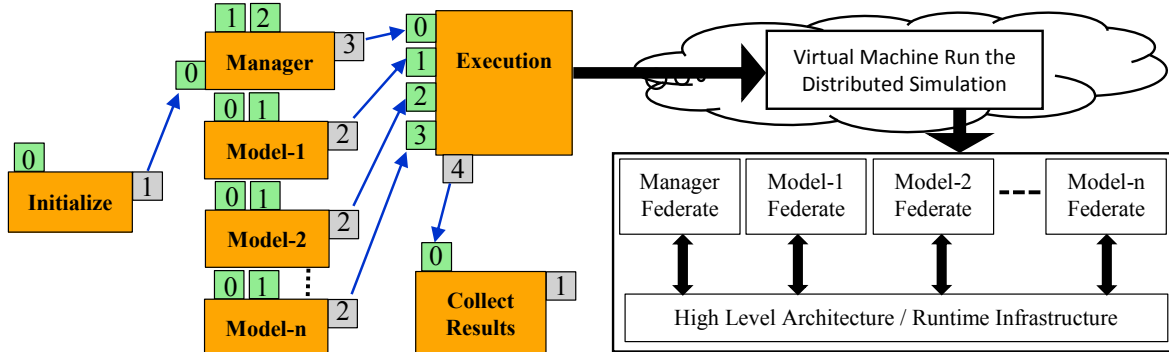


Figure-1: Framework for Cloud-based Distributed Simulation

### 3 CASE STUDY

To test the feasibility of the proposed framework, we deployed the distributed simulation model of a large Emergency Medical Services system, previously executed in a local network (Nouman et al. 2013). The federation consists of an Agent-Based Simulation model of the Ambulance Service and six Discrete Event Simulation models of the Accident & Emergency departments in the coverage area. Currently, the implementation utilises a single multi-core cloud instance and the verification of the cloud-based distributed simulation is performed.

### 4 CONCLUSION

The proposed framework simplifies the deployment of cloud-based distributed simulation using the WSPGRADE workflow layer of the CSSP. It is extensible since it supports the execution of varied size federations on the cloud. Future work involves performance testing and multi-instances implementation.

### ACKNOWLEDGEMENT

Special thanks to Dr. Tamas kiss and Hannu Visti, Centre for Parallel Computing, University of Westminster for support in this study.

### REFERENCES

Fujimoto, R. M. 2000. *Parallel and Distributed Simulation Systems*. New York: Wiley Interscience, Inc.  
 IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules, IEEE Standard 1516, 2010.  
 Kacsuk, P., Z. Farkas, M. Kozlovsky, G. Hermann, A. Balasko, K. Karoczkai, and I. Marton. 2012. “WSPGRADE/gUSE generic DCI gateway framework for a large variety of user communities.” *Journal of Grid Computing* 10(4):601-630.  
 Nouman, A., Anagnostou, A. and Taylor, S.J.E. 2013 “Developing a Distributed Agent-Based and DES Simulation Using poRTIco and Repast.” In Proceedings of the 17<sup>th</sup> IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications (DS-RT 2013), pp. 97-104.