UDC 004.94; 004.75

V. V. KAZYMYR, A. S. POSADSKA, D. M. SYSA

Chernihiv National University of Technology, Faculty of Electronic and Information Technology, Ukraine

CLOUD SIMULATION ENVIRONMENT BASED ON HLA

This technology of integration of a powerful formal apparatus E-nets and hierarchical aggregate approach which is used in the system of simulation use of E-net Modeling System (EMS), with High Level Architecture (HLA) in cloud environment is proposed. Functional features and structure of general purpose architecture HLA is reviewed. Service-oriented simulation technique of EMS is proposed. Using cloud platform OpenStack for effective solution of the problem of complex distributed simulation is considered. Architecture of synergy EMS and Platform as a Service is described in the article.

Key words: *cloud computing, high level architecture, distributed modelling, EMS, web-oriented interface.*

Introduction

Modern trends in the development of information technology bring to the fore the task of designing systems of distributed simulation, capable to implement additional benefits as a method to investigate complex systems using new technology. Emphasis should be paid to modelling under conditions of restriction of time, workforce, material and energy resources, as well as under conditions of uncertainty.

The urgent modern task is to investigate modeling directions in case when several models that should be combined for further work are required. High Level Architecture (HLA) [1] - a high level technology of distributed simulation is proposed to be used for this purpose. It has been developed at the end of 90s by the USA Department of Defense and is now supported by the IEEE 1516 standard [2]. The HLA has been developed based on processes involving government, academia and industry.

HLA architecture is a set of methods and standards for the design of distributed simulation systems. This technology combines the systems built for different purposes and in different time periods, products and platforms of different companies, providing them an ability to interact in a common synthetic environment. Run-Time Infrastructure (RTI), based on the XML standard, is used for a combination of models in HLA. However, HLA cannot satisfy the modern requirements appeared as results of the development of large-scale distributed simulation systems, because of inefficiently usage of resources, absence of the possibility of load balancing due to weak capacity of fault tolerance and complication of process of modeling deployment. Recent studies show that cloud computing [3] is an appropriate solution for extended simulation. Particularly, a new cloud model - Platform as a Service (PaaS) - is a modern decision for simulation tasks. PaaS is a category of cloud computing services for data processing, which provides application development platform as a service. This model involves the supply of cloud services, such as programming languages, application development environments, libraries, services, tools etc. for a user. Additionally, usage of cloud allows saving resources and other costs.

Considered the importance of the presented problem, the distributed simulation system E-net Modeling System (EMS) [4] has been developed by our researchers. The powerful formal apparatus of E-nets [5] and hierarchical aggregate approach has been applied for creation of models in proposed EMS. EMS has a convenient web-interface that provides the ability for the geographically remote users to work together at the same time with one simulation model. E-nets support display of management data flows and conduction of a quantitative processing at the network transitions provide convenient routing mechanisms of development processes and are significantly better than other network methods in the implementation of logic functions. At the same time, the usage of hierarchical aggregation approach allows conduction of investigation of complex systems from system analysis position.

Therefore, **the aim of this work** is a consideration of synergy HLA with PaaS by integration of EMS into the cloud environment.

1. High Level Architecture

HLA is a software standard that provides a common technical architecture for distributed M&S.

HLA is based on a composable «system of systems» approach (Fig.1):

- no single simulation can satisfy all user needs;
- support interoperability and reuse among Department of Defense (DoD) simulations.

Federations of simulations (federates):

- pure software simulations;
- human-in-the-loop simulations (virtual simulators);
- live components (e.g., reactive systems).

Federation is a set of simulation components (federates), which co-operate for solving a particular problem. Run-Time Infrastructure (RTI) [6] provides interoperability between all components of modeling.

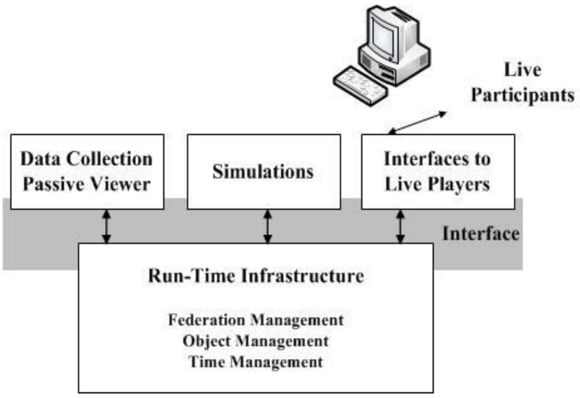

Fig. 1. Functional View of an HLA Federation

HLA includes the following components:

1. Federate Interface Specification.
2. Framework and Rules.
3. Object Model Template (OMT).

The first component specifies the services that each federation can use for communication. The second is a set of rules that ensure appropriate interaction within the federation. In case of the last component, the HLA technology does not impose restrictions on the internal structure of federates, but defines a standard for describing information about the objects of simulation - object model template. This standard provides interaction of federates and allows their multiple use regardless of the internal structure in different federations for simulation.

Since all communication between federates going through RTI, the exchange mechanism is implemented as a «subscription», it means that federate, which is interested in obtaining certain attributes, must subscribe to them using the RTI service.

2. EMS and HLA

E-net Modeling System (EMS) is a web-simulation system, which architecture is based on a few servers which interact with each other and allow creating, editing, debugging models and conducting experiments in a distributed environment (Fig. 2) [7].

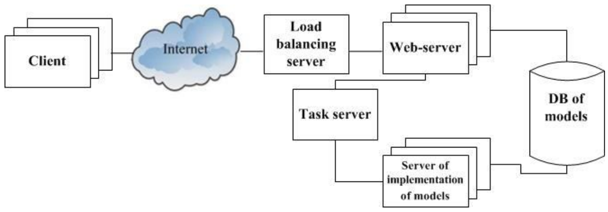

Fig. 2. EMS architecture

User has an opportunity to create, run and get the results of the modeling using only a browser thanks to the system web-interface (Fig.3). Possibility of providing the multiuser mode of the system requires several web-servers, where the load is distributed evenly between them via the load balancer server. Creating an imitation model, downloading and saving model experiment forming and processing of its results are provided interface subsystem and performed exclusively on web-servers. Simulation models require a large amount of computing resources, which can lead to web-server overload. Thus, it is necessary to use a separate server of implementation of models to ensure efficient use of resources and reliable operation of the web-servers, which performs the task of the immediate processing of user requests.

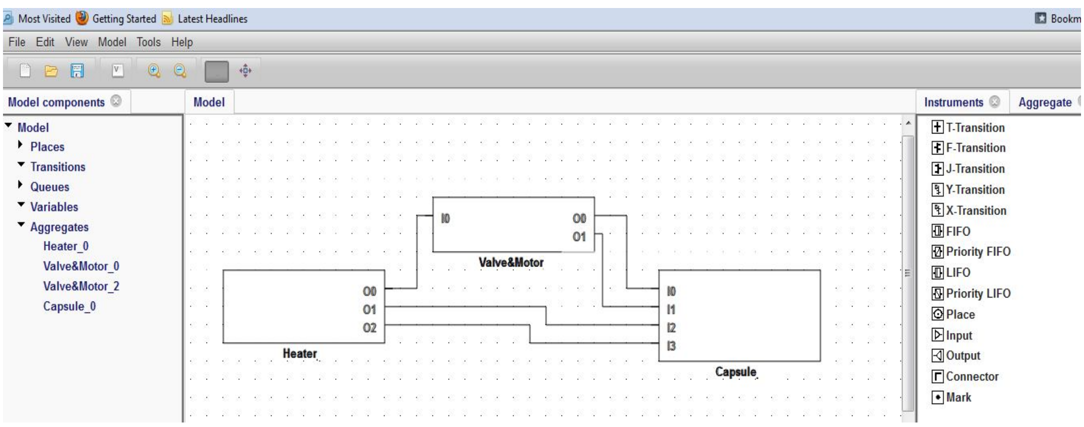

Fig. 3. EMS web-interface

Server problems perform a uniform distribution of tasks between the individual servers of implementation of models. Simulation models, created by users, the results of the simulations, as well as data about users of the system are stored in a DB of models that is available for all web-servers of the system [7].

Special graphics editor is designed for creation of models in the EMS, which makes it possible to develop easily perceived models and eliminates the need to study a special programming language, thereby greatly simplifies the process of simulation of systems with a complex structure.

Now we consider EMS integration into the HLA architecture.

Every EMS simulation model is viewed as one of federates, which, in turn, refers to the federation – it is a set of models of the system, which is being investigated. Special modules - Connect Module (CM), which perform the function of linking between models and RTI in the simulation, have been developed to ensure interoperability of simulation models, created in EMS under architecture HLA (Fig. 4) [7].

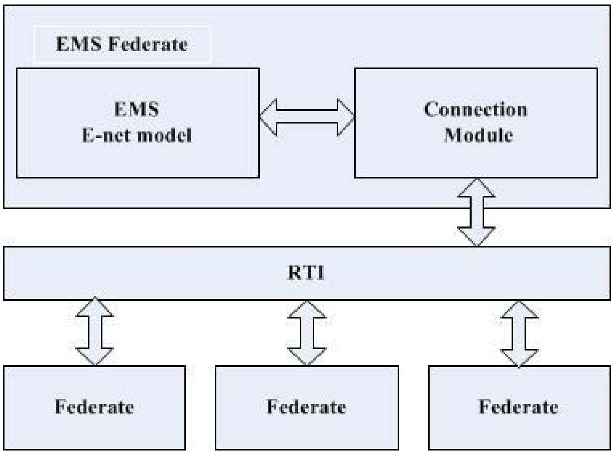

Fig. 4. The interaction of EMS federates with HLA architecture

CM receives attributes and interactions that come from other federates, transmits them to federates according to the mechanism of subscription and synchronizes the local time of execution of each federate according to the global time of work of all federation. Thus, the model that is developed in the EMS cooperates with the RTI HLA via CM.

During the distributed simulation task of collecting statistical data for each imitation model is assigned to a subsystem of model experiment. The collected statistics is also transmitted through the CM.

Interaction with the RTI services is provided by the presence in CM the unique inputs and outputs (RTIInput, RTIOutput respectively) and based on the usage of the methods of basic interfaces RTI: RTIAmbassador и FederateAmbassador. When the mark goes into RTI INPUT, settoken method is triggered. It sets new attribute in the E-net attributes, which is received from other federate, or updates the attribute. Also, the values of the input variables (parameters) of the network may be changed if necessary. The possibility of forming queues (lists) of messages is implemented in CM. It is necessary for synchronization of federates. Since each federate works in its local time, which is certainly not the same as the local time of the other federate, there is the need to create list of messages and the time storage of their receipt and the required implementation, in order not to lose the message with a time mark that exceeds the federate model time and not skip where time mark is less. Thus, during the transmission of messages a time mark indicating the time of the update of attributes is transmitted and synchronizes the operation of federates except attribute values and variables. Accordingly, settoken method is overridden for message transfer to the RTI when the mark goes to RTIOutput.

3. EMS in cloud

Considering the complication of the distributed simulation process in the EMS, the transfer of the system into the cloud is proposed.

3.1. Cloud Computing

Cloud Computing [8, 9] is a technology that allows usage the Internet for placing computing resources and provision of services to users. Usage of cloud computing allows to transfer service of applications and data storage to systems that have a high level of reliability, almost unlimited resources, and provide ready service for users.

Cloud computing promises several attractive benefits for industry and end users. Three of the main benefits of cloud computing includes:

– Self-service provisioning: End users can spin up computing resources for almost any type of workload on-demand;

– Elasticity: Companies can scale up as computing needs increase and then scale down again as demands decrease;

– Pay per use: Computing resources are measured at a granular level, allowing users to pay only for the resources and workloads they used.

Cloud computing services can be private, public or hybrid.

It is divided into three broad service categories:

– SaaS (Software as a Service) – it is ready cloud software for automatization of business processes: CRM, ERP, ECM, etc.;

– PaaS (Platform as a Service) — it is ready for using software, solution or platform in the «cloud»: databases, data processing systems;

– IaaS (Infrastructure as a Service) – it is a server with practically unlimited size and speed.

PaaS is proposed to be used for solving our problem. Platform-as-a-Service is a model, where the ability to use cloud infrastructure for placing basic software for subsequent placement new or existing applications is given to user.

There are several available open source PaaS platforms: Windows Azure, Amazon web services, Google app engine etc. Cloud technology OpenStack will be used by us.

3.2. OpenStack

OpenStack [10] is open source cloud computing software for creating, managing and deploying infrastructure cloud services. It consists of following components (Fig.5):

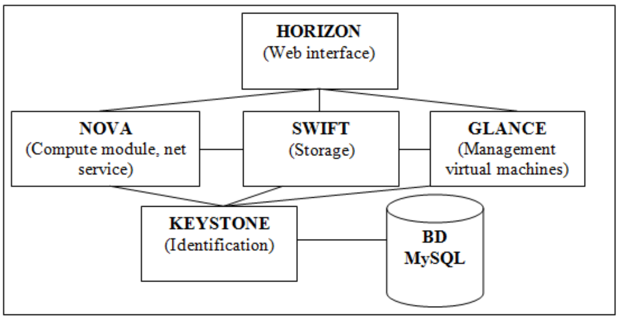

Fig. 5. OpenStack structure

OpenStack logical parts: Nova - computing module, network service, controller of computing resources, Swift - object storage, Glance - image management service of virtual machines, Keystone - identification service, Horizon - Control web-portal. Each module requires installation and configuration of specific components and necessary system packages of OS for their operation. Communication between components is performed by AMPQ protocol through a dedicated connection broker - computing controller. It is necessary to run the network and computational agents on each production server in order to provide the computing module operation. These agents interact with the management controller. System settings are stored in an SQL database (MySQL, PostgreSQL). The system can easily be decentralized through dividing services on different physical servers.

OpenStack functions:

1. Implementation of computing subsystems, interaction with hypervisors.

Computing Module nova-compute is installed on each working server and controls an operation of hypervisors and virtual machines through a local execution of system commands supported by the hypervisor. Computing module communicates with the computing controller (nova-api), the authentication service (keystone), the network service (nova-network), the task manager (nova-scheduler) and the other services of OpenStack. Management of block devices and their connection to the virtual machines in the Essex release is also performed by this module (nova-volume package). Implementation of the block devices is performed by a functionality of Linux LVM system software or by connection of external block of storage systems iSCSI.

2. The implementation of the storage subsystem for centralized storage and management of images and template settings of virtual machines.

Glance module, which by default works in conjunction with the Swift object repository, manages the storage and usage of virtual machine images. Ordinary file system also can be used as a repository for images. Swift Object Store allows converting the servers in the scalable data repository with built-in features to ensure fault tolerance. The system automatically makes several redundant replicas of data between servers and in case of failure of one of the servers, data integrity is not violated. It is not a file system, and works badly with OLTP data. It is also intended for long-term storage of large objects (virtual machine images, multimedia content) and is an analogue of Amazon S3 service.

3. Implementation of the virtual machines migration between cluster nodes without interruption of running services.

Migration of running virtual machines is performed exclusively by the functionality of the hypervisor is possible only between the nodes with installed same type hypervisors.

4. Implementation of the network subsystem.

Network settings are implemented through the usage of embedded OS Linux mechanisms for network management: the creation of bridges or VLAN-s. Nova-Network module manages the network.

5. Implementation of graphical user interface.

The graphical user and administrator interface is realized through a modular web-server written in Python using the Django framework. It is a graphical interface to all major OpenStack services.

It's definitely more mature product than other. Furthermore, there are more than 150 companies (AMD, Brocade, Dell, HP, IBM, VMware, and Yahoo), who are contributing to development. It's the leader in the cloud platform management and the momentum around its growth continues.

3.3. System architecture

EMS extends HLA with cloud computing technology to realize the service-oriented simulation support environment. Figure 6 shows the system architecture of Cloud EMS (CEMS).

Simulation application layer includes E-net the model system and the connect module. Simulation layer provides cloud RTI and federates. In this way, simulation models do not need to be placed on the local. Users can invoke them on demand through service access over the wide area network. And multi-users can access different instances of the same model service simultaneously. Services can communicate with each other in spite of the diversity of programming languages and platforms. Cloud infrastructure layer exploits cloud platform to integrate and operate various resources, and supports the implementation of simulation layer.

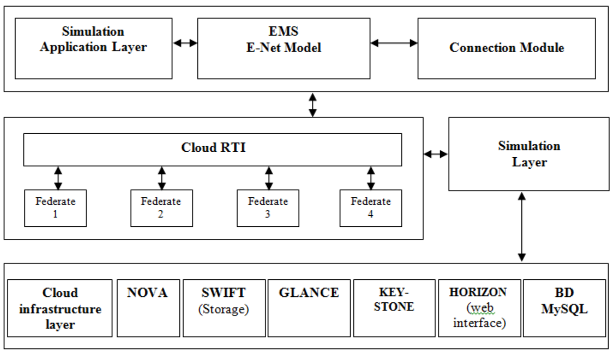

Fig. 6. CEMS architecture

The Cloud-RTI component encapsulates the functions of the traditional RTI as cloud services to support large-scale simulations on CEMS. CEMS can also support simulations on traditional RTIs by connecting traditional federates to the traditional RTI. Thus CEMS has good compatibility, which makes it practical.

Thus, self-organized simulation environment via adaptive platform is proposed: CEMS configures appropriate simulation environment on the virtual machines (VMs) for federation execution. Each federates of the federation and dedicated RTI instances are deployed on each VMs. Deployed federation is launched and monitored on each VM automatically.

Conclusion

Cloud computing can enhance the capability of HLA and provide an effective solution for complicated distributed simulation systems. In this paper, an efficient and cloud-based E-net Modeling System has been proposed and its architecture has been described.

The technology of integration of a formal apparatus E-nets and hierarchical aggregate approach with HLA architecture allows using distributed scheme of using EMS in cloud with all the benefits of this scheme.

OpenStack is proposed to use as PaaS because it is an open source platform with wide functionality.

Further work will be aimed at modifying CEMS for solving tasks of network planning in real time mode using formal apparatus of temporal logic.

References (GOST 7.1:2006)

1. Fujimoto, R. M. Distributed Simulation Systems [Text] / R. M. Fujimoto. – NY : A Wiley-Interscience publication, 2000. – 303 p.

2. 1516-2010 - IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules [Text] // IEEE Standard. – New York : IEEE, 2010. – 26 p.

3. Buyya, R. Cloud computing: The next revolution in information technology [Text] / R. Buyya // In Proc. of the lst International Conference on Parallel Distributed and Grid Computing. – Los Alamitos : IEEE, 2010. – P. 2-3.

4. Казимир, В. В. Розподілена система імітаційного моделювання EMS [Text] / В. В. Казимир, Г. А. Сіра, І. І. Мушкетик // Вісник Чернігівського державного технологічного університету. – 2011. – № 3. – С. 144 – 153.

5. Nutt, G. Evaluation Nets for Computer Systems Performance Analysis [Text] / G. Nutt // FJCC, AFIPS PRESS. – 1972. – P. 279 – 286.

6. Замятина, Е. Б. Современные теории имитационного моделирования: Программа специального курса [Text] / Е. Б. Замятина. – Пермь : ПГУ, 2007. – 119 с.

7. Казимир, В. В. Розподілене моделювання в EMS на основі архітектури HLA [Text] / В. В. Казимир, Г. А. Сіра // Математичні машини і системи. – 2011. – № 4. – С. 125-135.

8. Chu-Carroll, Mark C. Code in the Cloud [Text] / Mark C. Chu-Carroll. – Texas. : The Pragmatic Bookshelf, 2011. – 292 p.

9. Kazymyr, V. Grid workflow design and management system [Text] / V. Kazymyr., O. Prila, V. Rudyi // International Journal «Information Technologies & Knowledge». – 2013. – Vol. 7, No. 3. – P. 241-255.

10. Open source software for creating private and public clouds [Electronic resource]. – Access mode: https://www.openstack.org/. – 12.02.2016.

References (BSI)

1. Fujimoto, R. M. *Distributed Simulation Systems.* New York, A Wiley-Interscience publication Publ., 2000. 303 p.

2. *1516-2010 - IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules. IEEE Standard.* New York, IEEE Publ., 2010. 26 p.

3. Buyya, R. Cloud computing: The next revolution in information technology. *In Proc. of the lst International Conference on Parallel Distributed and Grid Computing.* Los Alamitos, IEEE Publ., 2010, pp. 2-3.

4. Kazymyr, V. V., Sira, H. A., Mushketyk, I. I. Rozpodilena systema imitatsiynoho modelyuvannya EMS [Distributed simulation system EMS]. *Visnyk of Chernihiv State Technological University*, 2011, no.3, pp. 144 – 153.

5. Nutt, G. Evaluation Nets for Computer Systems Performance Analysis. *FJCC, AFIPS PRESS*, 1972. pp. 279 – 286.

6. Zamjatina, E. B. *Sovremennye teorii imitacionnogo modelirovanija* [Modern simulation theory: Program for special course]. Perm, PHU Publ., 2007. 119 p.

7. Kazymyr, V. V., Sira, H. A. Rozpodilene modelyuvannya v EMS na osnovi arkhitektury HLA [Distributed modeling in EMS based on HLA]. *Journal «Mathematical Machines and Systems»*, 2011, no. 4, pp. 125 - 135.

8. Chu-Carroll, Mark C. *Code in the Cloud.* Texas, The Pragmatic Bookshelf, 2011. 292 p.

9. Kazymyr, V., Prila, O., Rudyi, V. Grid workflow design and management system. International *Journal «Information Technologies & Knowledge»*, 2013, vol. 7, no. 3, pp. 241 - 255.

10. *Open source software for creating private and public clouds*. Available at: www.openstack.org/ (accessed at 12.02.2016).

Поступила в редакцию 25.02.2016, рассмотрена на редколлегии 14.04.2016

МОДЕЛИРОВАНИЕ В ОБЛАЧНОЙ СРЕДЕ НА ОСНОВЕ HLA

В. В. Казимир, А. С. Посадская, Д. Н. Сыса

В статье предлагается технология интеграции мощного формального аппарата Е-сетей и иерархического агрегатного подхода, примененного в системе имитационного моделирования E-net Modeling System (EMS), в архитектуру высокого уровня HLA в облачной среде. Рассмотрены функциональные особенности и структура архитектуры общего применения HLA. Предложена сервисно-ориентированная технология моделирования EMS. OpenStack рассматривается как облачная платформа для эффективного решения задачи комплексного распределенного моделирования. Также в статье описана архитектура синергии EMS и Platform as a Service.

Ключевые слова: облачные вычисления, HLA, распределенное моделирование, EMS, веб-ориентированный интерфейс.

МОДЕЛЮВАННЯ В ХМАРНОМУ СЕРЕДОВИЩІ НА ОСНОВІ HLA

В. В. Казимир, А. С. Посадська, Д. М. Сиса

У статті пропонується технологія інтеграції потужного формального апарату Е-мереж та ієрархічного агрегатного підходу, застосованого в системі імітаційного моделювання E-net Modeling System (EMS), в архітектуру високого рівня HLA в хмарному середовищі. Розглянуто функціональні особливості та структура архітектури загального застосування HLA. Запропонована сервісно-орієнтована технологія моделювання EMS. OpenStack розглядається як хмарна платформа для ефективного вирішення задач комплексного розподіленого моделювання. Також в статті описана архітектура синергії EMS та Platform as a Service.

Ключові слова: хмарні обчислення, HLA, розподілене моделювання, EMS, веб-орієнтований інтерфейс.

Казимир Володимир Вікторович – д-р техн. наук, Чернігівський національний технологічний університет, Чернігів, Україна, e-mail: vvkazymyr@gmail.com.

Посадська Аліна Сергіївна – аспірант, Чернігівський національний технологічний університет, Чернігів, Україна, e-mail: alinka.posadskaya@gmail.com

Сиса Дмитро Миколайович – магістр, Чернігівський національний технологічний університет, Чернігів, Україна, e-mail: dmitriy.sysa@gmail.com.

Kazymyr Volodymyr - Doctor of Technical Sciences, Chernihiv National University of Technology, Chernihiv, Ukraine, e-mail: vvkazymyr@gmail.com.

Posadska Alina – PhD student, Chernihiv National University of Technology, Chernihiv, Ukraine, e-mail: alinka.posadskaya@gmail.com.

Sysa Dmytro – MSc student, Chernihiv National University of Technology, Chernihiv, Ukraine, e-mail: dmitriy.sysa@gmail.com.