INTEGRATION OF CLOUD COMPUTING PLATFORM
TO GRID INFRASTRUCTURE

Vladislav Falfushinsky, Olena Skarlat, Vadim Tulchinsky

V.M. Glushkov Institute of Cybernetics of NAS of Ukraine,
40 Acad. Glushkov Ave, Kyiv, 03179, Ukraine,
dep145@gmail.com, http://icybcluster.org.ua/eng/

Abstract: Both grid and cloud are used to organize large scale calculations and data processing on remote computers. Grid which became a basic computing infrastructure for the Large Hadron Collider experiments provides unified technical solutions for sharing and merging distributed heterogeneous computing resources within big collaboration groups. Cloud became popular among data centers and computing service providers because of flexibility, manageability and efficient hardware utilization. Both share common ideology "computing as a service", so one can expect additional benefits from their integration. The paper describes our approach to the integration. We propose to use cloud within grid sites for acceleration of application deployment and easy support of multiple virtual organizations by grid sites. The cloud in grid approach has been implemented and tested in Ukrainian National Grid, a part of European Grid Infrastructure. *Copyright © Research Institute for Intelligent Computer Systems, 2013. All rights reserved.*

Keywords: Grid computing; cloud computing; HTC; distributed computing; virtualization.

1. INTRODUCTION

Grid concept, architecture and middleware were developed by Ian Foster, Carl Kesselman and Steven Tuecke [1] since mid 1990th. But the wide application and popularity of grid in scientific calculations have been initiated by grid implementation of the Large Hadron Collider (LHC) data processing [2]. Instead of building a huge computer center to store and process 15 petabytes LHC data yearly [3], it was decided to organize their world-wide distributed storing and processing by the international collaborations of high energy physicists. The developed grid infrastructure and middleware appeared so powerful and flexible that many other researchers started to use it for their scientific calculations. European grid was organized in European Grid Infrastructure (EGI) and integrated with many national grids both within and beyond Europe [4].

Grid implements a paradigm of High-Throughput Computing (HTC) different from the High-Performance Computing (HPC) associated with parallel calculations [5]. When HPC targets minimal execution time for a job, the HTC aim is maximal utilization of all available computer performance to run multiple jobs or to solve a very big problem by parts. When a grid task is subdivided by subtasks the last ones are executed by different grid-sites according to their internal rules and queues without any guaranties of simultaneous run. It is the essence of distributed computing. Certificate based authorization, LDAP based resource catalogues and run environment specification languages create technical basis of grid.

Cloud proposes on-demand hardware usually with the pre-installed software. By sharing hardware of big datacenters cloud reduces the operational costs per task and increases efficiency of the hardware usage for the task flow [6]. It increases elasticity in environment selection and system mobility [7]. Typical requirements to cloud have been formalized by Peter Mell and Timothy Grance in [8]. Cloud providers offer virtualized computational resources with service-oriented provisioning and support "pay as you go" usage-based pricing model. The last feature is important for commercial clouds such as Amazon EC2, GoGrid, FlexiScale, and others.

In theory cloud gives the illusion that unlimited computing resources are available. Users can request, and are likely to obtain, sufficient resources at any time. In practice the illusion may be broken for large workloads or if the task is resource-dependent.

The users are allowed within both grid and cloud to acquire and release resources on-demand. Therefore, as the needs of the workflow change over

time, the releasing of the resources enables the workflow systems easily to grow and shrink their available resource pool. But specification of the requested resources in grid is implemented via its parameters. As the resource description is incomplete, the allocated nodes can be different, and nobody can guarantee their compatibility with the program. Cloud exposes resources with identical pre-installed system environments.

Cloud excels grid in benefits for workflow applications. Cloud applications are distributed in a common unified system and so can efficiently share data and exchange messages. Grid application sub-tasks can be separated both geographically and in time, so communication between them is tricky. Furthermore, cloud enables remote access to the allocated nodes, and thus supports on-line user-directed operations such as dynamic visualization.

Cloud can simultaneously run different operational systems on the same physical servers. So it provides more opportunities to combine independent software in solution of complex problems. E.g., data computed by a 32-bit Linux program can be than visualized in 64-bit Windows application within a common workflow.

Both grid and cloud propose ready to use computing environments on demand [9]. But cloud is less scalable and grid is less flexible and requires more administrators' efforts to fit contradictive system requirements of multiple virtual organizations. This is why integration of grid and cloud is in focus of modern researches.

The simplest kind of the integration is merging multiple geographically distributed computers in a single cloud based grid-site. Such way is proposed by the project "Experimental Deployment of an Integrated Grid and Cloud Enabled Environment in BSEC Countries on the Base of g-Eclipse" started in 2013 [10]. The project approach, however, is poor suitable for HPC tasks which need high performance parallel computations and use grid primary as a common access point to multiple clusters. However such approach can give more utilization power for big number of old hardware that can be used to virtualize nodes for some sorts of power HPC tasks.

More general approach is developed within the EGI-InSPIRE project established a Federated Cloud Task Force [11]. Some EGI grid-sites are already offering private cloud services for local research organizations. The project helps to unify their cloud architecture. Every site of the EGI Federated Cloud exposes the same programming interfaces for virtual machine setup and data manipulation operations, therefore applications that are built for one site of EGI Federated Cloud can run at any of the EGI cloud sites. In other words the EGI Federated Cloud provides an extensible set of reusable virtual

machine (VM) images from the EGI VM marketplace. The images contain installed and set up scientific programs. The approach supposes complete virtualization of the EGI Federated Cloud grid-sites, which can decrease their HPC performance and prevent use of non-virtualized hardware (e.g. non-virtualized GPU accelerators). Besides, the whole grid resource pool is subdivided by two parts of cloud and ordinary grid-sites (Fig. 1).
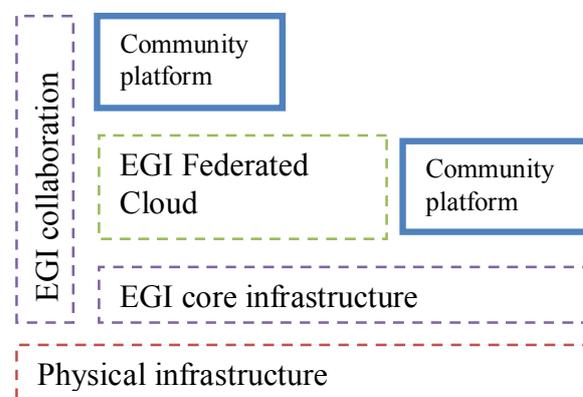
Fig. 1 – The EGI Platform Architecture.

Our idea is slightly different. We propose to run virtual machines as ordinary grid tasks and manage them similarly by grid tasks [12]. As result grid tasks can be run either in virtual or in physical environment. Thus routing problem solution can be accelerated by special efforts of either grid site or virtual organization administrators who install the necessary application software on physical servers. At the same time, a user can immediately run a special program version or configuration without its preliminary installation on multiple target grid-sites. He can easily add the preinstalled or own software to virtual machine, too. If the task is still running he can add some configuration issues to the virtual machine OS. Each user has its own set of virtual machines. In more restricted environment the administrator can control the list of available virtual machines for security reasons. If there is no appropriate template of virtual machine in repository, user can contact administrator and provide a preconfigured template.

In such architecture the cloud platform can be used also for secure testing the new software. Besides, it creates ability to allocate some resources for their online use (without queue) during the virtual machine grid task lifetime. Moreover there is an ability to make a shared access to some devices, that are installed on hypervisor i.e. accelerators, additional storage devices, cache etc. Such capability is absent in classical grid.

2. INTEGRATED PLATFORM ARCHITECTURE

The proposed integrated platform consists of 3 levels (Fig. 2):

1) the grid middleware responsible for the job delivery to appropriate clusters, for the data transfer, and for the user authentication;

2) the cluster scheduler responsible for the job queuing and distribution its subtasks between the cluster nodes;

3) the cloud platform software responsible for managing the private cloud as the cluster virtual part.
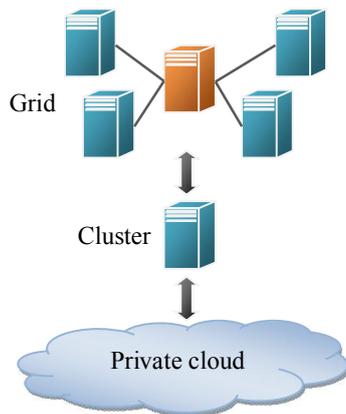
Fig. 2 – The cloud platform integrated in the grid infrastructure.

The grid user generates a proxy-certificate for an appropriate lifetime period, creates the job specification as a file (e.g. in the xRSL format) and launches the job using the grid submit command. The grid scheduler directs the job on a proper grid-site according to the job resource specifications. If the job specification contains links to files, grid automatically downloads them from user's local file system, FTP, HTTP, the grid storage or other place and puts them to a folder or a block device which is shared to the virtual machine file system. The cluster scheduler accepts the job, inserts its command into the queue and runs according to the queue rules. The integrated cloud platform does not affect the grid workflow.

Since the virtual systems need unified path to the input files, the calculation results, and the intermediate files which are download/uploaded by the grid middleware during the job submitting, the integrated platform is based on the unified storage structure rules. Within the Ukrainian National Grid (UNG) [13] all grid application files are managed according to the next rules:

• each grid user accesses UNG resources only under a certificate associated with a Virtual Organization (VO);

• each VO has its own directory in the grid storage;

• the filenames of system files and folders in the VO directory are started from dot (to be invisible for users);

• each VO directory contains system folders ".apps" and ".cloud" for user applications and backups of the system images;

• other subdirectories of the VO directory are shared to all members of the VO.

The cloud platform main unit called *instance* is a virtual node of the cluster. Each instance is created from one of pre-installed VM images. The instances are managed by users and VO administrators via a set of cloud management commands. The commands can be submitted as ordinary grid jobs, according to usual grid and cluster security rules and restrictions, which can deny users direct access to the virtualization system. The cloud management command set is supported by management components, the core of the integrated cloud platform.

Among them there is a routing service which opens a certain port for remote access to the instance through Remote Desktop Protocol (RDP). This is a key to extension of traditional grid functionality by dialog user interface capabilities. RDP remote desktops can be used from personal computers and mobile devices such as smartphones or tablets. Besides, RDP simplifies downloading/uploading files during the instance lifetime. (This is important for long time executed dialog programs, as the traditional grid tools wait for a job finis to download its results.)

Speaking more, the management components create the most important part of the intermediate layer between the private cloud and the grid. Other necessary parts include a virtualization server and a database of VM images & instances, constraints, the virtual network settings, the software licensing information and so on (Fig. 3).
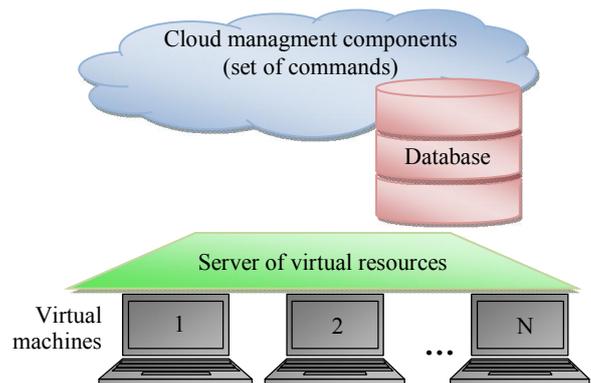
Fig. 3 – Intermediate layer between the cloud platform and the grid.

The list of cloud management commands for VM control in the integrated grid infrastructure includes:

• `VMmanager` to list the images, to list, create, start and shutdown the virtual nodes;

• `VMRunApp` to execute a user command within an instance (on the corresponded virtual machine);

• `VMRegisterApp` to install an application in an instance;

• `VMDBCreate` to test the system, create and restore the cloud platform DB. (The command permissions may be restricted to administrators only);

• `VMAddImgTpl` to manage the instance templates which relate the instance parameters to the physical node parameters.
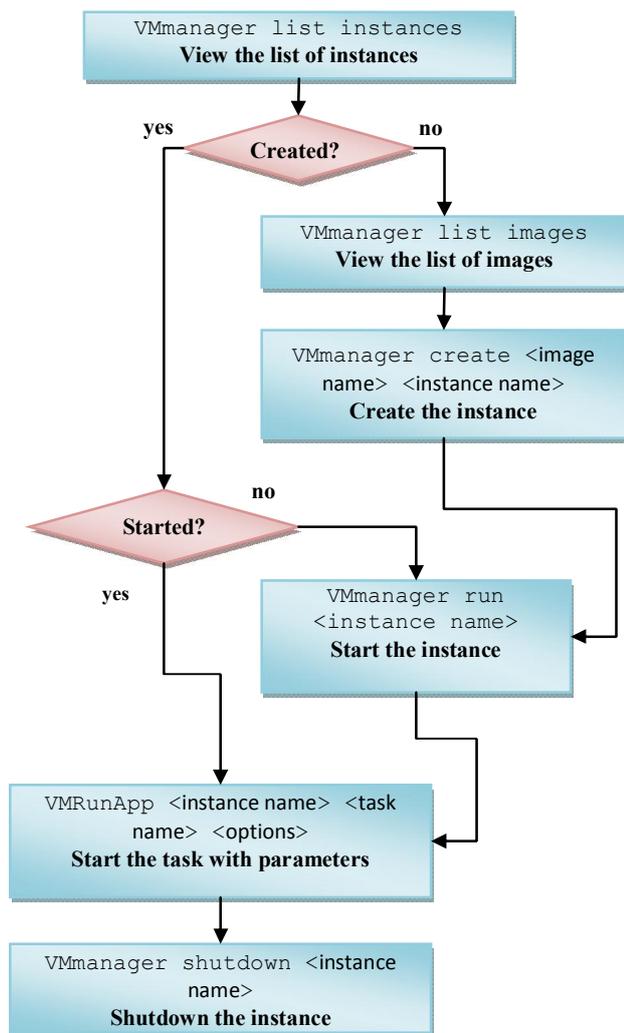
Fig. 4 – Cloud management commands in use.

The cloud management commands application can be illustrated by the next scenario.

At first, the user inspects the list of the available instances by VMmanager. If the requested instance is not running, it can be started by

```
VMmanager run <instance name>
```

After successful launching the instance the command

```
VMRunApp <instance name>
     <task name> <parameters>
```

is called to run the task with parameters on the specified instance. In order to free allocated resources, the instance can be stopped by

```
VMmanager shutdown <instance name>
```

If the requested instance is absent it can be created from an image. The list of images can be retrieved by VMmanager list images. The new instance is created by

```
VMmanager create <image name>
          <instance name>
```

The cluster administrator can call management commands for managing templates and cloning images. The whole scenario can be described as one grid script for the integrated cloud platform (Fig. 4). After the script is started the user is allowed to acquire additional connection options (like RDP address, port, user and password) calling the status command from grid utilities.

3. INTEGRATED PLATFORM IMPLEMENTATION

Core components of clouds are VM hypervisors which host all the VM instances. The integrated platform accesses hypervisor via the opensource toolkit libvirt [14]. This library acts as a middleware between hypervisor and cloud management components. It is compatible with most popular hypervisors, supports recent Linux versions and provides modern set of virtualization capabilities:

• management of virtual machines, virtual networks and storage;

• portable client API for Linux, Solaris and Windows;

• remote control with authentification and encryption;

• unified management of multiple hypervisors from one access point;

• a driver-based architecture and common hypervisor-independent API;

• integrated services: HTTP, DHCP server, VPN, SSH, built-in shell.

The recommended VM hypervisor tested in our platform is Oracle VirtualBox [15]. The choice was motivated by its low resource consumption, fast configurability, multi-system VM (Linux, Windows and MacOS), and parallel run of multiple VM containers. The licensing policy favorable for Academia also affected the decision.

Special efforts targeted user authorization and authentication in the integrated platform. It is important to keep the extended access capabilities within the standard grid rules and use nothing but

the time-limited proxy-certificates of grid users. The proposed solution is based on the proxy sharing via Myproxy Credential Management Service [16]. It combines the online credential repository with the online certificate authority, and allows users to obtain securely credentials when and where needed. The system is configured to encrypt all private keys in the repository with user-chosen passphrases and server-enforced policies for passphrase quality. The technique also provides delegation of the credentials from one user to another without using certificate files and its passphrases.

To put the integrated platform in operation the web user's interface (Fig. 5-7) has been developed.
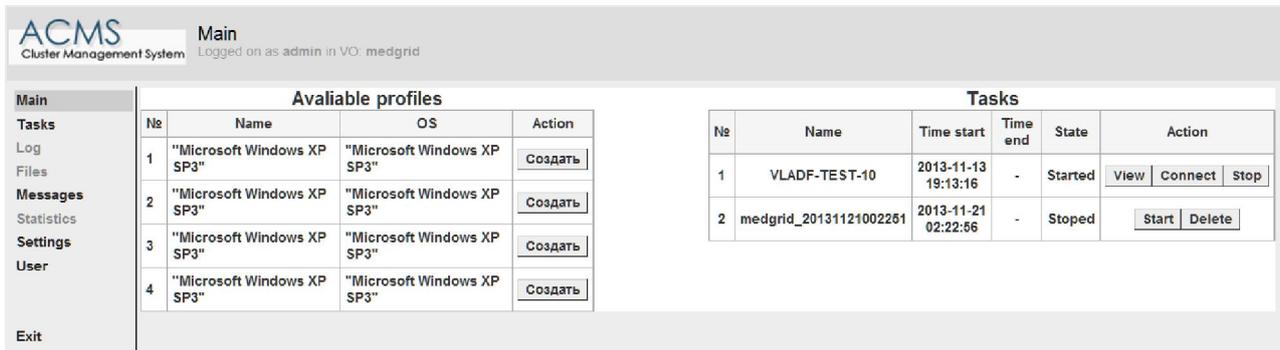
Fig. 5 – Web interface of the integrated platform.
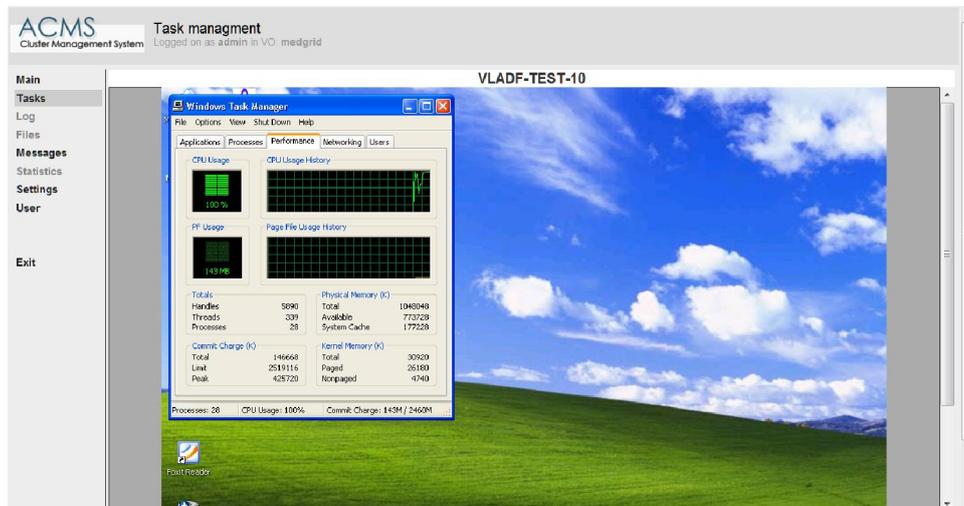
Fig. 6 – Windows XP run as a VM task on Linux cluster via ARC grid middleware.
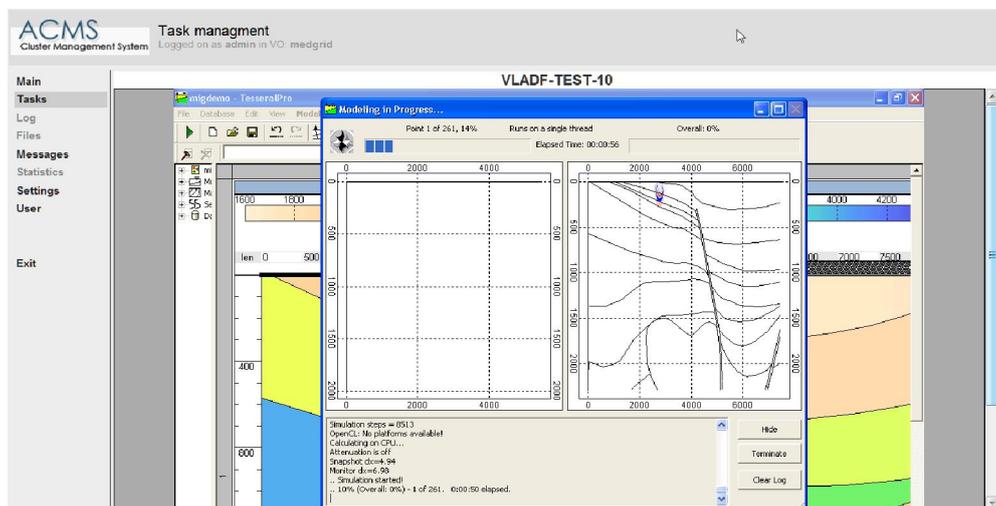
Fig. 7 – Seismic modeling program Tesseral Pro for Windows executed in grid in the integrated platform and available from the web interface.

First end-user experience of application the integrated platform in VO "Medgrid" [17] and "Geopard" [18] resulted in improvements and development of additional functions:

1. Delayed start of grid tasks. The grid tasks related to cloud management are submitted in background to unfreeze the task manager during the long enough procedure.

2. Accelerated work with remote grid storage. Virtual copy of the VO folder directory is stored in database. In addition to acceleration of usual operations it helps to synchronize or re-install local storage. Current recovery speed for all directory structure is about 1000 files per minute. The synchronization tool is run periodically in background.

3. Cache of VO user certificates was added to web interface. This function allows to read statistics and to communicate with other VO members without generating proxy certificates. (Access to grid is not possible.)

4. PHP API for ARC implementation of the integrated platform with basic functions of monitoring, submitting, interruption and deleting tasks, file operations in grid storage.

5. Web interface API for remote administration of the web interface allows easy integrate the solution in another management system.

4. CONCLUSION

The offered integration approach unites the principles of both grid and cloud computing. The described implementation of the cloud platform integration in grid-infrastructure has been tested and applied in operation within Ukrainian National Grid which is an integrated part of EGI. UNG is partially based on ARC middleware. But the approach is suitable for gLite and EMI grids as well. The developed set of commands is sufficient for flexible command line interaction with the integrated in grid cloud platform. Main advantages of the proposed solution are:

1) quick deployment of new or alternative program versions within VO, less administrator's efforts,

2) arbitrary mix of grid and cloud/grid tasks on the same clusters,

3) dialog and on-line environments run in grid for immediate user's operations (to exclude delays for submitting and re-submitting grid tasks),

4) automated data flow and distributed storing.

5) Linux/Windows portability,

6) tolerance to differences in operational environments and VM hypervisors of different grid sites.

6. REFERENCES

[1] I. Foster, C. Kesselman, S. Tuecke, The anatomy of the grid enabling scalable virtual organizations, *International Journal of High Performance Computing Applications*, (15) 3 (2001), pp. 200–222.

[2] M. W. Browne, Physicists discover another unifying force: Doo-Wop, The New York Times, April 2008, pp. 4–6.

[3] R. Lucio, Superconductivity: its role, its success and its setbacks in the Large Hadron Collider of CERN, *Superconductor Science and Technology*, (23) 3 (2010), pp. 17-21.

[4] European Grid Infrastructure, http://www.egi.eu

[5] J. Schiff, Grid computing and the future of cloud computing, *Proceedings of the Enterprise Storage Forum*, January 21, 2010.

[6] R. Bias, Grid, Cloud, HPC … What's the Diff?, http://www.cloudscaling.com/blog/cloud-computing/grid-cloud-hpc-whats-the-diff/, November 18, 2010.

[7] J. Myerson, Cloud computing versus grid computing, http://www.ibm.com/developer works/library/wa-cloudgrid/, March 03, 2009.

[8] P. Mell, T. Grance, Special Publication 800-145 The NIST Definition of Cloud Computing, *National Institute of Standards and Technology*, September, 2011.

[9] I. Foster, Yong Zhao, I. Raicu, Lu Shiyong, Cloud Computing and Grid Computing 360-Degree Compared, *Proceedings of the Grid Computing Environments Workshop*, 2008, pp. 1-10.

[10] Experimental deployment of an integrated grid and cloud enabled environment in BSEC countries on the base of g-Eclipse, http://blacksea-cloud.net/

[11] Federated Cloud, http://www.egi.eu/infrastructure/cloud/

[12] Vladislav Falfushinsky, Olena Skarlat, Vadim Tulchinsky, Cloud computing platform within grid infrastructure, *Proceedings of the 7th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2013)*, Berlin, Germany, 12-14 September 2013, pp. 717-720.

[13] Ukrainian National Grid, http://ung.in.ua/

[14] Libvirt, http://libvirt.org/

[15] Oracle VirtualBox VM, https://www.virtualbox.org/

[16] MyProxy Credential Management Service, http://grid.ncsa.illinois.edu/myproxy/

[17] Medical Grid-system for population research http://medgrid.immsp.kiev.ua/secondary_pages/home_en.html

[18] UNG Virtual Organization for grid applications in Earth sciences, http://vo.igmof.org.ua/ (in Ukrainian)

[19] Intelligent computers «Inparcom», http://www.inparcom.com

[20] Uber-Cloud Experiment, http://hpcexperiment.com

Vladislav Falfushinsky, *Junior researcher and PhD student at V.M.Glushkov Institute of Cybernetics of NAS of Ukraine. Master of Computer Science: Intelligent systems for decision making, National University of Kyiv-Mohyla Academy, 2010. Interests: grid, cloud, large scale and distributed software systems, parallel application software (primary for geophysics), databases and big data processing.*

Olena Skarlat, *Junior researcher at V.M.Glushkov Institute of Cybernetics of NAS of Ukraine. Master of Computer Science: Information Control Systems and Technologies, National University of Kyiv-Mohyla Academy, 2010. Interests: Analysis of the Ukrainian regulatory framework and modelling business-processes for e-Services – Software assessment – ISO and ETSI standards harmonization– Long-term preservation strategies– Adopting grid-technologies for investigations.*

Dr. Vadim Tulchinsky, *Head of High-Performance and Distributed Computing Lab at V.M.Glushkov Institute of Cybernetics of NAS of Ukraine. Interests: parallel and distributed computing; big data, seismic data processing, databases; simulation, computational methods in geophysics.*