Computing and Informatics, Vol. 31, 2012, 743–757

FLEXIBLE ORGANIZATION OF REPOSITORIES FOR PROVISIONING CLOUD INFRASTRUCTURES

Joanna Kosińska, Jacek Kosiński
Sławomir Zieliński Krzysztof Zieliński

Department of Computer Science
AGH University of Science and Technology
Al. Mickiewicza 30, 30-059, Kraków, Poland
e-mail: {`kosinska, jgk, slawek, kz`}`@agh.edu.pl`

Communicated by Jacek Kitowski

Abstract. The paper proposes an architecture of a system automating the provisioning process of cloud computing infrastructures. Its structure and components are specified, based on an analysis of three types of requirements: infrastructure providers, service providers and end users. These considerations have led us to formulate a new infrastructural model, offered to end users as a collection of Virtual Machines (VM) connected by a dedicated Virtual Private Network (VPN) with QoS guarantees. The role of repositories in cloud provisioning systems is specified along with the relevant data acquisition processes. The applicability of the proposed system is illustrated by practical usage scenarios.

Keywords: Virtualization, resource management, repositories, cloud infrastructure

1 INTRODUCTION

The cloud computing paradigm has been known for many years [2] as an evolution of utility computing [9] techniques. However, efforts are still underway to formulate its proper definition. For example, the National Institute of Standards and Technology definition of cloud computing [8] refers to its five essential characteristics: on-demand self-service, broad network access, resource pooling, rapid elasticity and measured services. On-demand services are services that can be rapidly and flexibly provisioned. Typically, there are four types of services that can be provisioned and consumed over the Internet [15]: infrastructure resources (computing power, storage or entire machines), software resources (e.g. middleware, development software), application resources (e.g. collaboration applications, word processors) and business processess (applications exposed as loosely coupled sub-processes or tasks). These resources, in turn, imply the widely-known cloud computing models.

Based on research conducted by large corporations [15, 13] in the areas of service provisioning and design it appears that two key enabling technologies play an especially important role in building computing solutions: virtualization [14] and Service-Oriented Architectures (SOA) [4]. The above-mentioned types of resources are all provisioned using virtualization technologies and represented as services.

In order for clients to be able to access resources, they must first be discovered. Clouds resources are not easy to discover and it is difficult to select and use their services. An analysis presented in [3] shows that even prominent cloud vendors [1, 7, 5, 11] do not provide support for service discovery, although Windows Azure has recently released .NET Service Bus components which include some discovery mechanisms.

In order to support automated provisioning of resources, cloud computing environments should rely heavily on repositories, keeping track of physical infrastructure state and configuration. A significant stage of user requirement processing involves representing requirements as sets of queries to be resolved against the current state of the system (i.e. records kept in repositories).

This paper focuses on a provisioning infrastructure for cloud systems, developed as an ACC Cyfronet AGH contribution to the PL-Grid Project [10]. A system for provisioning automation is proposed and described. Its structure and components are specified based on analysis of three sets of requirements, presented by infrastructure providers, service providers and end users. These considerations have led us to formulate a model of a computing infrastructure allocated for end users as a collection of virtual machines (VM) connected by a dedicated Virtual Private Network (VPN) with QoS guarantees. Our model is more general than the one offered by popular cloud infrastructures such as Amazon EC2 [1], Windows Azure [7] or Google App Engine [5]. These solutions implement Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) models and primarily focus on providing on-demand computational and storage resources. The communication topology of connections between computational nodes is almost neglected in these models. In contrast, the model proposed in our paper originates from research on virtual grids [6].

The provisioning process proposed in the presented infrastructure is rather complex and requires a suitable system to support it. We have determined that the functional and nonfunctional properties of such a system (including its dependability, configurability and flexibility) are all related to the design of repositories used for representing the following information:

1. hardware configuration,
2. VM appliance and
3. virtual infrastructure configuration.

The structure of the paper is as follows. In Section 2 requirements for cloud infrastructure provisioning are presented. The end-user view is highly prominent in this part. The presented considerations underpin the proposed system architecture, which satisfies the stated requirements. This architecture is described in Section 3. It is evaluated from the provider needs point of view. All building blocks of the proposed system are described in detail. Subsequently, in Section 4, the repositories – a core part of our system – are described. Implementation aspects are also presented. Section 5 presents a case study involving the use of the constructed system. The paper ends with conclusions.

2 REQUIREMENTS OF CLOUD INFRASTRUCTURE PROVISIONING

Both grid computing and cloud computing are attempts at utility computing. The idea is similar but the realization of utility computing is significantly different. In a nutshell, grid computing assumes that the application executes on nodes that are predefined, in fact all nodes in a grid are set by an administrator. On the other hand, in cloud computing nodes are provisioned on user's demand and fully utilized. In grids the user has access to entire computers, while in clouds access is limited to VMs, applications and services.

The current evolution from grid systems towards cloud systems is a result of progress in virtualization technologies covering three main resource types: computing power, network and storage [12]. The emergence of Cloud systems does not mean that Grid systems will lose their importance. For high-performance applications in scientific research they remain an effective way of exposing computational resources. However, virtualization technologies introduce great resource management mechanisms, introducing the ability to:

- flexibly share physical resources by solely allocating them as virtual resources while guarantying QoS parameters;
- create (at runtime, on the application's or user's demand) proper virtual resources and allocate them;
- isolate virtualized resources allocated for a given user from other users' resources. This isolation preserves SLA guarantees for allocating specific physical resources and eliminates crosstalk of applications belonging to different users;
- ensure security in the sense of data confidentiality and reliability of operations.

The first of these aspects is a consequence of isolation provided by virtualization technologies, while the second one refers to the ability to dynamically create virtualized infrastructures on top of physical resources. The aforementioned features of virtualization technologies concern both server virtualization provided by environments such as Xen or xVM and operating system virtualization ensured e.g. by Solaris or OpenVZ.

Isolation is, in turn, a crucial prerequisite for achieving the guaranteed level of virtual resources. Thus, resource allocation can be realized similarly to classic grid installations where worker nodes are allocated to tasks on an exclusive basis. Virtualization can effectively solve another problem, namely assurance of infrastructure protection through creation of an environment which enables secure execution of untrusted application code or user services.

These features and properties of virtualization techniques make it possible to construct a system enabling resource sharing in a coordinated way by applying the cloud computing model.

This article proposes a flexible architecture of repositories for provisioning cloud infrastructures. This feature is achieved by fulfilling the following requirements:

Runtime configurability – in the proposed system, the configuration of components can be modified during runtime, after the infrastructure is created.

Autoconfiguration – system components are automatically discovered, simplifying initial configuration and resource reallocation.

Reliability – no single point of failure: crucial components can be duplicated while discovery mechanisms facilitate adjustments of service state, guaranteeing continued system operation in case of failures of individual components.

Automatic backup – the system automatically creates copies of repositories and data collected during its operation, at regular intervals.

Technology independence – the system does not make any assumptions regarding the technologies used (including virtualization technologies); hence it is easy to incorporate new technologies.

These assumptions come from recommendations for systems belonging to a similar class [2].

3 ARCHITECTURE OF CLOUD RUNTIME INFRASTRUCTURE

The development of a system exposing virtual grid resources is a multistage task with regard to automation of creation of these resources and their autonomic management. The starting point for our project is the implementation of basic mechanisms enabling creation of virtual grids and their exposure to the user.

The management process of virtualized PL-Grid infrastructures requires a monitoring and management system (Figure 1). This system distinguishes three types of agents:

Monitoring & management physical node manager – responsible for coordinating VMM on each physical node, e.g. creating new VM, resource consumption by each VM, VM migration among different nodes.

Monitoring & management agent – monitoring user applications in a given VM.

Monitoring & management system manager – runs on a Global Node and acts as a coordinator responsible for managing M & M physical node manager and M & M agents.

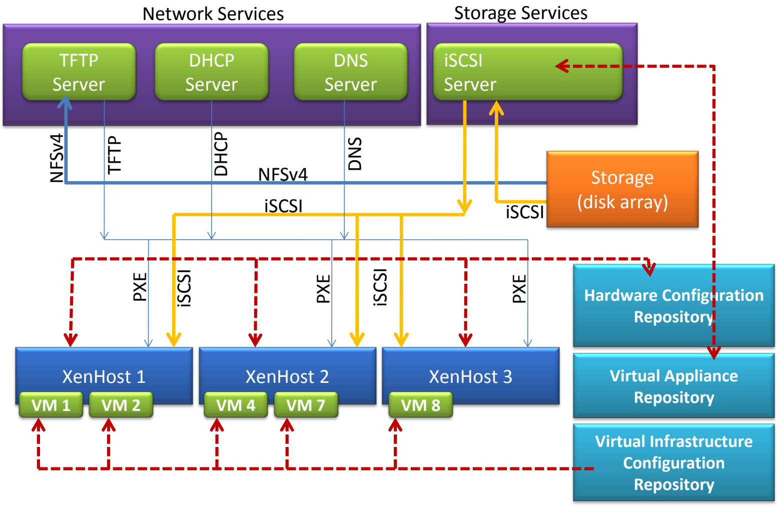

Fig. 1. Architecture of monitoring and management services in the PL-Grid infrastructure

The presented architecture also contains repositories of information related to infrastructure lifecycle (creation of new VMSet, repository of system images, etc.) Each repository exposes an interface that is utilized by the monitoring & management system manager.

The architecture presented in Figure 1 contains network services which support deployment and operation of the proposed system. The main components that are used during environment initialization are:

- components responsible for bootstrapping the operating system on physical nodes. This mechanism uses PXE and the services are DHCP and TFTP servers, running in a dedicated Solaris system zone;
- the infrastructure for serving VM disk images, which consists of a dedicated service sharing disk space through the iSCSI protocol;
- a set of repositories (discussed in the following sections).

The proposed architecture takes into consideration the availability of a (dynamically changing) resource pool represented by physical machines (computers installed at computing centers) connected via a computer network. This concept assumes exposing computing and communication resources through a virtualized computing infrastructure. The creation of a virtual computing infrastructure results in instantiation of virtual resources (virtual machines) connected via a virtual computer network. This procedure is performed according to user specification and executed on system administrator's demand.

The procedure of creating a virtual infrastructure presents the user with the ability to express expectations related to types, capacity and configuration of the required resources. The user interface can be organized as a set of forms available on the PL-Grid website.

4 ORGANIZATION OF REPOSITORIES FOR CLOUD PROVISIONING

This section focuses on the key repositories of the proposed system and the way they support the process of deploying user-specified logical topologies. Moreover, it introduces some guidelines for selecting repository implementation technologies.

There are three main repositories that are expected to play an important role in the operations performed by the proposed environment, which may be characterized as follows:

1. hardware configuration repository, which stores information about available physical resources
2. VM appliance repository, which stores images of virtual machines, operating systems and preinstalled applications
3. virtual infrastructure configuration repository, which stores information about mapping between resources and VMs, QoS guarantees, planned resource usage, etc.

The repositories were designed not only according to requirements specified in Section 2 but also with regard to the three layers of the cloud computing open architecture [15], namely:

- virtualization – hardware and software – thorough management and sharing physical nodes as VMs. Software virtualization is realized as OS image management and software sharing;
- service-orientation – common reusable and composite services – services that constitute cloud offerings such as iaas and paas;
- cloud core – provisioning and subscription services – the system provisions clients with required resource types and notifies them when such resources become available.

The structure and roles of the proposed repositories are described in the following subsections.

4.1 Hardware Configuration Repository

The goal for introducing a hardware configuration repository is to dynamically discover changes to the hardware runtime environment, e.g. to keep track of dynamic addition or removal of hardware dedicated to the cloud computing infrastructure. Addition of a new hardware node results not only in its registration in the repository, but also in a micro-benchmark test in order to obtain an estimate of the node's capabilities. Removal of a node following a failure (or on an administrator's demand) would also be reflected in the repository, making it possible to migrate all VMs that consume the node's resources. Obviously, such events involve addition and removal of hardware elements performed by the infrastructure provider's staff.

From the infrastructure provider's point of view several use cases and sets of requirements can be distinguished:

- installation of a new hypervisor node,
- making the node available to a production network,
- notifying about a node leaving the network due to its shutdown or failure.

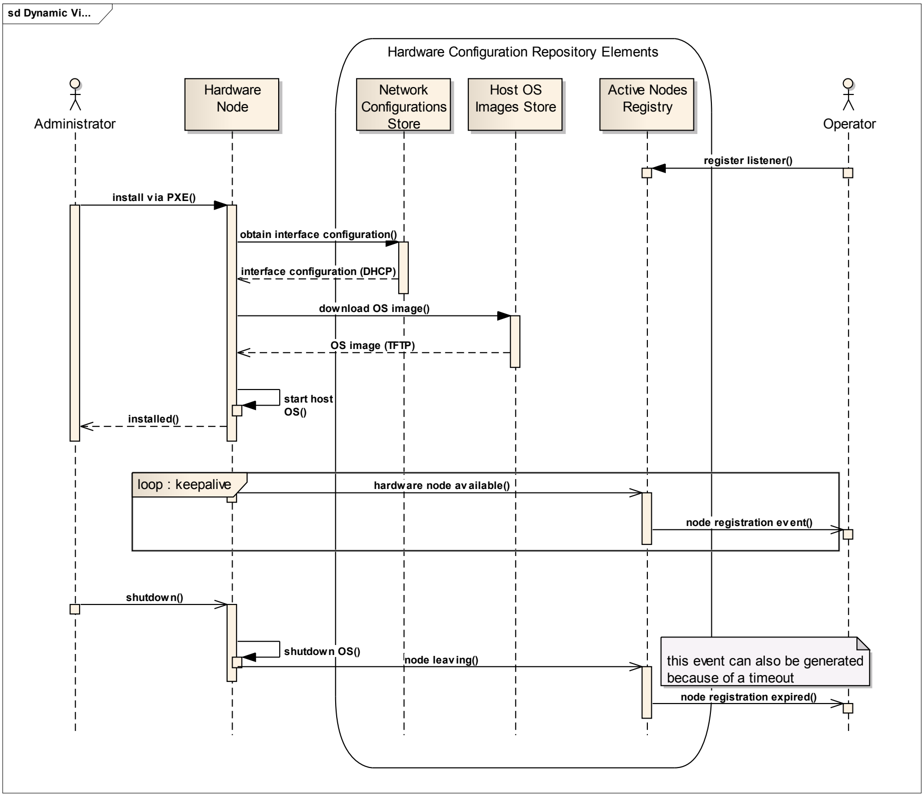

Fig. 2. Hardware configuration repository update process

The hardware configuration repository update process is illustrated in Figure 2 In the installation use case a provider installs the host operating system, pre-configured to operate properly in the provider's environment. The installation process is supported by the Pre-boot eXecution Environment (PXE); thus a suitable set of functions should be provided for this package. In light of this fact, the hardware configuration repository should maintain pre-configured and tested images of host operating systems in order to download them during machine startup. Although simple and unreliable, the TFTP protocol can be used to download a system image, especially if the server is located inside or close to the LAN containing the newly installed machine. DHCP as the leading dynamic configuration protocol can also be leveraged for basic machine setup.

Following initial setup, the node needs to be exposed to the production network. However, because it is hard to assess the computational capabilities of the node based on a static data record (e.g. processor architecture, number of cores, etc.) describing its architecture (especially in a heterogeneous environment), a set of micro-benchmarks might be executed during reboot. Their results should be stored along with static data for later use, e.g. for a scheduler planning deployment of a VM set with specific SLA contract requirements.

In order to propagate the changes made to the hardware environment at a reasonably robust level, a dynamic discovery scheme needs to be adopted. Although many such schemes exist, simple multicast-based discovery of JMX software agents installed in the host's operating system should be sufficient. The same agents could be used both for obtaining machine descriptions and for performing management actions.

The results of the dynamic discovery need to be stored in a database whose structure would support the administrator's console with various perspectives regarding the current resource allocation. Although it is not a strict constraint, a centralized database supporting the provider's perspective on resource usage seems to be a reasonable approach. However, depending on the size and geographical distribution of the installations, a single server might prove insufficient. Therefore, in order to support various scenarios, constructing the database on top of a standard directory service, such as LDAP, would be advisable as this approach offers access to a variety of tools that facilitate deployment, replication, referrals, etc.

4.2 VM Appliance Repository

The goal of the VM appliance repository is to store and distribute the disk images of the VM appliances, which are customized operating systems running on virtual machines. The main idea behind the repository concept is to physically separate the disk used by the VM from the hosting machine the hypervisor operates upon. This enables us to mitigate the consequences of a hardware node crash (though it remains a difficult task). Moreover, because the disks are linked using the provider's network rather than physically present on the host machine, the process of migrating virtual machines could be made more straightforward and less time-consuming.

The VM appliance repository will maintain not only system images but the descriptions of their configurations and profiles as well. Our solution proposes components for managing the process of sharing disk images through a network protocol, e.g. iSCSI or NBD.

In order to support failure recovery, the VM appliance repository is required to perform snapshots of the VM disk state. Moreover, the ability to roll back disk operations to a previous snapshot would help in failure recovery. Such requirements could be satisfied by a modern filesystem. Because stable release of BTRFS has not been made as of August 2010, the proposed environment will utilize ZFS, which supports the required features, at least until its main rival, BTRFS, becomes mature. Also, that does not prevent other FS from being used in the applications run on the PL-Grid project infrastructure. For example, Lustre FS [?] can still be used for computational data distribution, and that usage not affected by the configuration of the repositories.

ZFS volumes could be served as disks to the virtual machines using the standard iSCSI protocol (Figure 3).

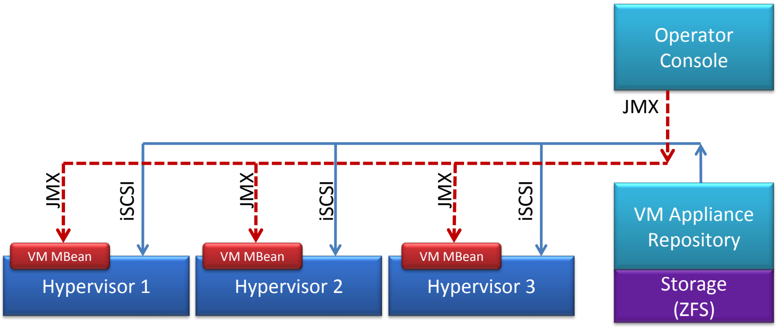

Fig. 3. Storage space sharing with the VM appliance repository

By introducing the VM appliance repository, our solution provides the cloud administrators with a way to create and maintain a growing set of typical OS configurations while keeping the overall disk space usage at a reasonable level. Thanks to utilizing ZFS features, the organization of VM appliance images is not flat. In most cases, the amount of physical disk space consumed by a new image is much lower than the size of the image itself, provided that the underlying OS is used by another image.

The incremental storage of images is performed at the filesystem level and these operations are hidden from the instances of virtual machines. Therefore, the process of installing any software, including user libraries or applications, does not require any additional steps, i.e. no modifications of the installed elements are required.

Leveraging ZFS could also help in multiplying VM appliances, if needed, thanks to the filesystem's volume cloning ability. Since cloning does not require replication of all data, the overall disk space consumption could be kept at a low level: most clones will differ only with regard to user data (and thus not in the system area).

4.3 Virtual Infrastructure Configuration Repository

virtual infrastructure configuration repository will complement the description of the runtime environment by keeping track of the configurations created as a result of processing the requirements specified by cloud users (Figure 4).

The repository will keep track of the current allocation of hardware resources to the VM sets and individual VMs, that result from the user-defined specifications of VM sets and interconnecting networks. The main records stored inside the repository refer to:

- the users, user groups, their roles and privileges;
- the relationships between user groups and VM sets;
- the configuration of individual virtual machines, i.e. their network addressing and links to external services such as remote disk storage;
- configuration changes (to be) made to networking appliances in order to interconnect the virtual machines and expose them to the user console;
- quality of service (QoS) guarantees;
- plans for resources usage.

The records need to be coupled with virtual infrastructure elements. For example, in order to support automation of virtual machine and hardware node configuration, user-related data should be stored in a format applicable for processing by operating systems. Network-related configuration, such as locations of iSCSI targets, IP addressing etc. could also be automated using data from this repository, thereby making the task of the scheduler (either an automated process or an operator interacting with a console) much easier. Section 5 introduces a use case that describes the usage of the virtual infrastructure configuration repository in more detail.

5 ON-DEMAND CLOUD PROVISIONING SCENARIOS

Cloud provisioning scenarios should take end user experience and system knowledge into account. From this point of view, two categories of cloud infrastructure can be distinguished:

- VM set, called IaaS. It is a collection of virtual machines connected via a virtual network. These VMs contain only an operating system, on top of which any user application can be installed. This model is suitable for more experienced users.

- VM appliance set, similar to PaaS. In this case VMs contain preinstalled applications, so no provisioning activity is required from the user's point of view.

Advanced users, possessing OS configuration knowledge, can flexibly configure the system executing on a VM. On the other hand, they cannot directly define machine parameters. To provide the advanced user with flexibility in configuring the entire system, it is reasonable to expose VMs with shell access. Additionaly, the user can configure network parameters which are used to link virtual machines together.

$$IaaS \begin{array}{l} \nearrow \{VM\} \\ \searrow \{VM, VNet\} = \text{VMSet} \end{array} \tag{1}$$

Users without proper knowledge (or those not interested in configuration details of infrastructure elements) should be equipped with predifined VMs, specifically the VM Appliance Set. A typical configuration comprises not only the operating system with its parameters, but also software packages frequently accessed by users.

$$PaaS \rightarrow \{VMAppliance\} = \text{VM Appliance Set} \tag{2}$$

The greatest simplification from the user's point of view is the reduction in the amount of specific parameters that are provided while creating the infrastructure. The set of parameters corresponding to predefined configurations is stored in a dedicated repository. Configuring these machines would also store parameters used for allocating their resources. This case refers to a situation where the user requires grid middleware support for a custom application (for instance Globus or GLite should be available).

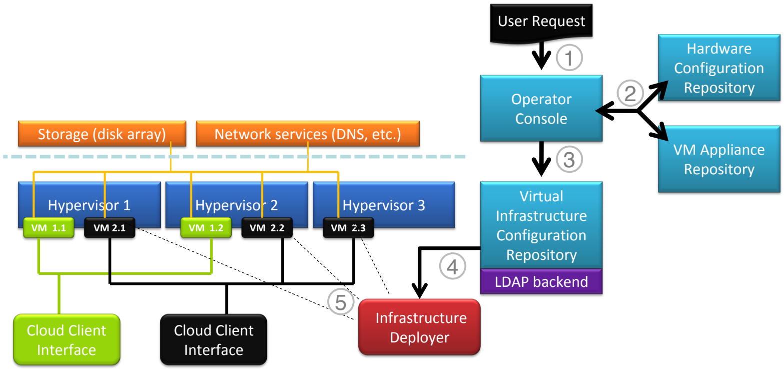

Fig. 4. Processing a user request through the virtual infrastructure configuration repository

Processing of end-user requests cointaing specifications of required cloud infrastructures is accomplished according to the following scenario (Figure 4):

1. The end user (step 1) specifies the requirements regarding VM set or VM appliance set. This specification contains all information necessary to create the requested infrastructures. In particular, in this specification the user chooses whether the IaaS or PaaS model should be applied.
2. The end-user specification is processed by the cloud infrastructure manager using the operator console. The manager has access to the hardware configuration repository and VM appliance repository and is able to allocate requested resources to satisfy end-user demands (step 2). This activity yields a configuration specification of VM set.
3. Once accepted, the VM set specification (step 3) created for the end user is stored in the virtual infrastructure configuration repository in order to allocate the necessary resources;
4. The accepted VM set specification is passed to (step 5) the infrastructure deployer. This component is responsible for constructing the specified cloud infrastructure.
5. The provisioned VM set is accessible through the CloudClient interface enabling remote access and usage of allocated resources. Access to the VM set is exclusive to the selected end user owing to isolation provided by virtualization technologies.

This scenario assumes cloud manager activity. In a more advanced case, the role played by the operator can be performed by the end user. This advanced scenario would offer a full range of on-demand cloud infrastructure benefits; however, it calls for further development of the proposed system architecture.

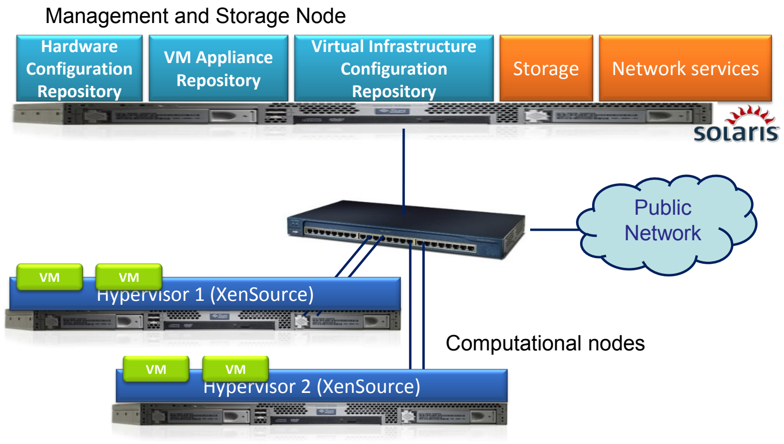

Fig. 5. Repositories evaluation infrastuctre configuration

Figure 5 shows hardware infrastructure used for presented architecture evaluation. Our infrastructure uses nodes running Oracle Solaris OS (repositories instances) and additional two computational nodes running Linux with Xen Hypervisor (for deployment of VMs). PL-Grid users can obtain a VMSet that consists of at least one virtual computational node. Each of the nodes is represented by a VM which is managed by a hypervisor. The VM is assigned pre-configured computing resources (CPU, RAM, network) and a fixed operating system which runs applications belonging to the VMSet user. Further (larger scale) deployment is planned.

6 CONCLUSIONS

Dynamic on-demand provisioning of cloud computing infrastructures is a sophisticated and challenging issue. This process requires suitable tools whose activity is very much related to data specifying the configuration of underlying hardware and virtual resources. Resource-specific data should be kept up to date using dedicated discovery and monitoring protocols. The configuration parameters of the virtualized infrastructure should be carefully selected by a system administrator in the context of end-user functional and nonfunctional requirements. A collection of commonly used configurations should be pre-configured and stored, ready for deployment without additional effort.

The level of automation of the provisioning process can be gradually increased, starting with a purely manual system and evolving in the direction of a system with adaptability mechanisms in place, guaranteeing requested QoS of cloud infrastructure configuration under changing load and hardware availability conditions. Clustering technology should be taken into account as a highly atractive solution as far as high availability of virtualized infrastructures is concerned. This issue is going to be the subject of further interesting research.

Acknowledgement

This work was made possible thanks to the PL-Grid project (contract number: POIG.02.03.00-00-007/08-00, website: `www.plgrid.pl`). The project is co-funded by the European Regional Development Fund as part of the Innovative Economy program.

REFERENCES

[1] Amazon Elastic Compute Cloud. `http://aws.amazon.com/ec2/`. Accessed in April 2010.

[2] Armbrust, M.—Fox, A.—Griffith, R.—Joseph, A. D.—Katz, R. H.—Konwinski, A.–Lee, G.–Patterson, D. A.—Rabkin, A.—Stoica, I.—Zaharia, M.: Above the Clouds: A Berkeley View of Cloud Computing. Technical report, EECS Department, University of California, Berkeley, February 2009.

[3] Brock, M.—Goscinski, A.: A Technology to Expose a Cluster As a Service in a Cloud. In AusPDC 2010: Proceedings of 8th Australian Symposium on Parallel and Distributed Computing, Australia 2010.

[4] Erl, T.: Service-Oriented Architecture (SOA): Concepts, Technology, and Design. SOA Systems Inc., 2009.

[5] Google App Engine. `http://code.google.com/intl/pl/appengine/`. Accessed in April 2010.

[6] Zieliński, K.—Kosińska, J.—Kosinski, J.: Virtual Grid Resource Management System with Virtualization Technology. In KU KDM '09: Proceedings of the Second Conference of the High Performance Computers' Users, 2009.

[7] Microsoft Azure. `http://www.microsoft.com/windowsazure/`. Accessed in April 2010.

[8] National Institute of Standards and Technology (NIST). `http://www.nist.gov/index.html`. Accessed on April 2010.

[9] Parkhill, D.: The Challenge of the Computer Utility. Addison-Wesley Educational Publishers Inc, 1966.

[10] PL Grid Project. `http://www.plgrid.pl/`. Accessed in August 2009.

[11] Salesforce. `http://www.salesforce.com/`. Accessed on April 2010.

[12] Słota, R.—Nikolow, D.—Polak, S.—Kuta, M.—Kapanowski, M.—Skalkowski, M.—Pogoda, M.—Kitowski, J.: Prediction and Load Balancing System for Distributed Storage. Scalable Computing Practice and Experience, Special Issue: Grid and Cloud Computing and their Application, Vol. 11, 2010, No. 2, pp. 121–130.

[13] Sun Cloud Computing. `www.sun.com/cloud_computing`. Accessed in February 2009.

[14] von Hagen, W.: Professional Xen Virtualization. Wiley-Publishing Inc. 2008.

[15] Zhang, L.—Zhou, Q: CCOA: Cloud Computing Open Architecture. In ICWS 2009: Proceedings of 7th IEEE International Conference on Web Services, Los Angeles, CA, USA 2009.

Joanna Kosińska is a research assistant at the Department of Computer Science, AGH University of Science and Technology, Krakow (since 2001). Her main interests focus on distributed environments and portal technologies based on Java. She has participated in several national and international research projects, mainly EU-funded.

Jacek Kosiński is an assistant professor at the Department of Computer Science, AGH University of Science Technology, Krakow (since 2000). His main interests focus on computer networks and operating systems. He is a CCNP instructor. He has participated in several national and international research projects, mainly EU-funded.

Sławomir Zieliński is an assistant professor at the Department of Computer Science, AGH University of Science and Technology. His main areas of interest include computer networking and distributed systems. He has participated in several national and international research projects, some of which were EU-funded.

Krzysztof Zieliński is a full professor and head of the Institute of Computer Science at AGH-UST. His interests focus on networking, distributed computing, object-oriented distributed system engineering and SOA-related technologies. He is the author/co-author of over 150 papers in the above-mentioned areas. He served as co-chair and PC member of many conferences and workshops worldwide. He is a member of the IEEE Computer Society, ACM, and the Polish Academy of Sciences (Computer Science Chapter).