Computer Science • Vol. 12 • 2011

Sławomir Zieliński, Tomasz Handzlik, Tomasz Lewicki[*]

UPNP-BASED DISCOVERY AND MANAGEMENT OF HYPERVISORS AND VIRTUAL MACHINES

The paper introduces a Universal Plug and Play based discovery and management toolkit that facilitates collaboration between cloud infrastructure providers and users. The presented tools construct a unified hierarchy of devices and their management-related services, that represents the current deployment of users' (virtual) infrastructures in the provider's (physical) infrastructure as well as the management interfaces of respective devices. The hierarchy can be used to enhance the capabilities of the provider's infrastructure management system. To maintain user independence, the set of management operations exposed by a particular device is always defined by the device owner (either the provider or user).

Keywords: *discovery, management, hypervisor, virtual machine, UPnP*

WYSZUKIWANIE ZARZĄDCÓW WIRTUALIZACJI I MASZYN WIRTUALNYCH ORAZ ZARZĄDZANIE NIMI ZA POMOCĄ UPNP

Artykuł opisuje zestaw narzędzi opartych na technologii Universal Plug and Play, wspomagających współpracę pomiędzy użytkownikami i dostawcą usług w zakresie wykrywania zarządców wirtualizacji (ang. hypervisor) i maszyn wirtualnych oraz zarządzania nimi. Prezentowane narzędzia konstruują spójną, hierarchiczną reprezentację rozmieszczenia maszyn wirtualnych w infrastrukturze fizycznej, zawierającą opisy fizycznych i wirtualnych urządzeń oraz usług służących do zarządzania nimi. Struktura ta może być wykorzystana przez dostawcę usług np. w celu optymalizacji wykorzystania posiadanych mocy obliczeniowych. W celu zachowania niezależności pomiędzy użytkownikami i dostawcami infrastruktury przyjęto założenie, że zbiór usług oferowanych przez poszczególne urządzenia jest określany przez jego posiadacza.

Słowa kluczowe: *zarządzanie, maszyna wirtualna, UPnP*

1. Introduction

Computational infrastructures are continuously getting more and more dynamic, and the numbers of machines present in datacenters continue to grow. Meanwhile, due

[*] AGH University of Science and Technology, Faculty of Electrical Engineering, Automatics, IT and Electronics, Department of Computer Science, al. A. Mickiewicza 30, 30-059 Krakow, e-mail: `slawek@agh.edu.pl`, {`thandzli,tlewicki`}`@student.agh.edu.pl`

to the potential utilization increase [1], the virtualization technologies are becoming increasingly popular and the infrastructures become increasingly complex. That makes the infrastructure management a truly demanding task, which is perceived as a significant barrier for adopting server virtualization [2].

In virtualized environments, there are two clearly distinguished views upon infrastructure: provider's (resource-oriented) and users' (application-oriented). The infrastructure management suites (IMSs) that are in use by the infrastructure providers (e.g. OpenNebula [3], Hitachi Unified Compute Platform [4]), are obviously provider-oriented and use the management operations implemented by hypervisors. Similarly, the IMSs designed for the users (e.g. enStratus [5]) are designed to support user operations; they are focused on unifying the management of user applications run upon infrastructures provided by various vendors.

It seems that the reason for the separation between the user's and provider's perspectives is keeping the user operations private and secure. However, in order to maintain the agreed service level, the providers' IMSs need to collect runtime information about users' environments and therefore monitor the virtual machines to measure the resource consumption, etc. [6] Note that being unaware of the virtual machine specificity, the provider can influence its behavior only in a graceless way, e.g., by limiting the resources or stopping the whole VM. The prototype toolkit described in this article can be used to enhance the set of management operations available to the provider. It does so by implementing means for exposing user-defined services to a provider's management suite while maintaining the independence between the parties.

The rest of the article is organized as follows. Section 2 gives an overview of the target environment the toolkit is expected to run on and draws a few requirements for its design. Section 3 discusses the two distinct views of the (already instantiated) virtualized infrastructure. The following section 4 presents the key functions of the toolkit, as well as its support for the runtime environment operations. The last section summarizes the article.

2. Target runtime environment

The presented toolkit is designed to work as a part of the PL-Grid [7] experimental infrastructure. The experimental infrastructure is a small subset of the whole PL-Grid infrastructure, dedicated for experiments with cloud services provisioning. The characteristics of the planned runtime environment was taken into account at the toolkit design phase. This section points out the most important guidelines that resulted from the analysis.

Experimental infrastructure elements

The key elements of the experimental infrastructure (see Fig. 1) are as follows:

- *access gateway*, which is a host providing network admission control,

- *boot services provider*, which is a host supporting additions of hardware and virtual nodes,
- *repositories container*, which is a host providing access to configuration-related repositories,
- *JIMS gateway*, which integrates the environment with the JMX-based Infrastructure Monitoring System [8],
- *storage server*, which stores the images of hypervisors' and virtual machines' operating systems as well as the users' data,
- computational nodes, which provide XenSource hypervisor services,
- switched local area network (LAN) with IEEE 802.1q virtual LAN (VLAN) support.

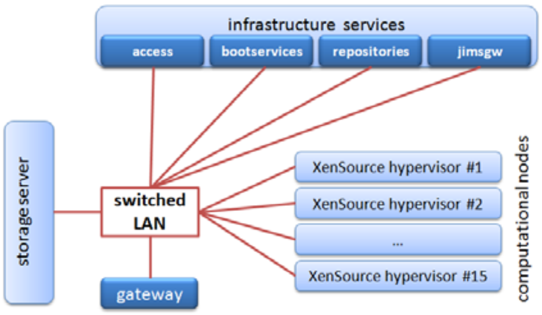


Fig. 1. Key elements of the PL-Grid experimental infrastructure

Technically, the access gateway, boot services provider, repositories container, and JIMS gateway are run as Oracle Solaris zones on a single server. However, they are visible as logically separate hosts connected to the provider's network.

Because all the components are connected to a single LAN, no inter-provider links can be simulated, and no application-specific components are in place, the infrastructure is suitable mostly for experimenting with provisioning Infrastructure-as-a-Service (IaaS) clouds upon a single-provider infrastructure. Therefore, the presented toolkit prototype is focused on that scenario. The prototype tools are capable of discovering, keeping track of, and managing key virtualized computational infrastructure entities, i.e. hypervisors and virtual machines, but designed to accommodate more granular (e.g. services) or specific (e.g. dedicated devices) entities.

Provisioning-related repositories, network traffic isolation

The system is not intended to implement all provisioning-related functions from scratch. Rather, it is designed to cooperate with the existing infrastructure components. As introduced in [9], the process of provisioning the virtualized infrastructures in the PL-Grid experimental infrastructure is supported by three repositories:

- *hardware configuration repository*, which keeps track of the hardware computational nodes possessed by the provider,
- *virtual appliance repository*, which holds the images of virtual appliance operating systems,
- *virtual infrastructure configuration repository*, which tracks the configuration of the providers infrastructure.

The infrastructure is built with isolation of provider and user networks traffic in mind. Separate virtual networks (VLANs) are constructed for provider and user operations. The repositories are organized accordingly – while all the repositories are accessible via the provider's (management) VLAN, only parts of the virtual infrastructure configuration repository are accessible via respective users' VLANs.

The presented tools collect data that can be reused to keep the repositories up to date; therefore appropriate modules for updating the repositories were designed and implemented.

Design guidelines

This section gave an overview of the tools' target runtime environment. To sum up, three main guidelines for the system design were drawn:

- the key scenario for the system operation is managing a set of single-vendor IaaS clouds,
- the toolkit needs to operate in both user and provider networks,
- the toolkit needs to update the configuration-related repositories.

The next section sketches the process of implementing user (virtual) infrastructure upon the PL-Grid experimental infrastructure and points out the features that can be used by the toolkit to combine the information possessed by the users and the provider.

3. User and provider networks

This section focuses on the networks that connect the elements of either provider (physical) or user (virtual) infrastructures. The presented tools are going to process information collected from all user and provider networks. Therefore, this section outlines the relationship between the networks, analyzes what data can be collected from both types of networks, and points out the desired toolkit capabilities.

In case of a single-vendor cloud scenario, which was chosen as the main one due to the reasons explained in section 2, the user requests the provider to create and expose a set of virtual machines or virtual appliances connected by a – presumably virtual – network. Moreover, the network needs to be connected to the user premises.

An important non-functional requirement – which can be easily omitted in formal specifications of requirements, maybe because it is so obvious – is the simplicity of abstractions presented to the end users. Therefore, as a basis for our toolkit we chose a very plain, Ethernet LAN-like, logical network topology.

The task of the provider is to implement the logical infrastructure requested by the user upon the provider's infrastructure (see Fig. 2).

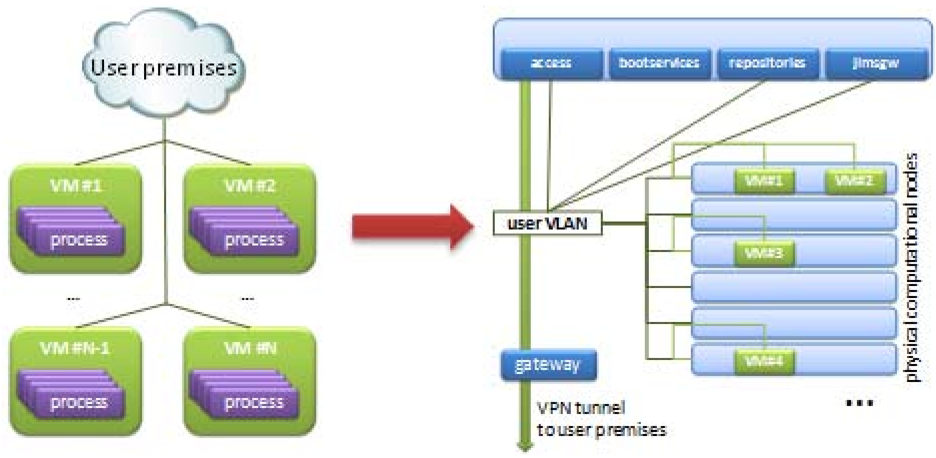


Fig. 2. An overview of implementation of a user-defined abstraction upon the provider's infrastructure

Roughly speaking, the instantiation of the requested set of virtual machines consists of the following steps:

- creation of virtual machines, based on plain or customized operating system images,
- configuration of a dedicated virtual LAN (VLAN) for the user,
- configuration of a virtual private network (VPN) connection to the user's site.

After completing the instantiation process, the user – using the VPN connection – is able to use the created virtual machines as if they were available locally.

User network

In order to manage the processes run on virtual machines, the user's IMS needs to be able to discover the managed entities, which can include virtual machines and processes executed by them. In complex cases it seems reasonable to introduce even more granular management (e.g. in case of an application server that hosts multiple applications, each of which has a management interface). Thus, extendibility of the tools for cloud users seems to be a desirable feature.

The user network is expected to forward advertisements or queries and responses for the user virtual machines and their subcomponents. Note that because the advertisements or responses are issued by the virtual machines, which are controlled by the users, their contents can also be controlled by the users. In case the user wants to feed the provider's IMS with a management interface, appropriate advertisement can be formed and passed to the provider's IMS. However, due to security constraints, the user will have no direct connectivity with the management network. Instead, the provider part of the toolkit should contain tools for tracking the user advertisements.

Provider network

To facilitate creation and configuration of virtual machines, the provider needs an isolated management network interconnecting hypervisors and other entities that participate in the provisioning process. In the case of the experimental infrastructure, the network is simply a dedicated VLAN.

The management network is expected to forward advertisements or queries and responses for the hypervisors. Although the hypervisors are aware of the virtual machines run in the system, they have no insight into the VMs and their services. Therefore, the information published by the hypervisors can be only used to locate particular VMs. In order to feed provider's IMS with management interfaces of user virtual machines and their processes, tools for matching the deployment information with virtual machines' advertisements need to be implemented.

Merging the user and provider-originated information

The key concept of the presented toolkit is to combine the information available in the provider's and users' networks and to create a unified logical representation of the whole computing environment for the provider's IMS. To accomplish that, the provider needs to be able to introduce IMS components into user networks. In order not to break users' privacy, a decoupled approach is proposed. The provider is equipped with components listening for advertisements issued by VMs, while the user is in complete control of the VM advertisements' contents. Using the proposed toolkit the provider's IMS can augment the information coming from the management network with VM management services selected by the user. In that way, the provider's IMS can learn the hypervisor-VM-services hierarchy.

The following section discusses the integration of user and provider-originated data in more detail.

4. Toolkit functions and components

The Universal Plug and Play (UPnP) [10] architecture defines a standard for advertising devices and their services. What makes the UPnP kind of advertisements noteworthy, is that they allow for advertising nested devices, effectively allowing for creating a device hierarchy. That naturally fits the hypervisor-VM hierarchy observed by cloud service providers.

The most straightforward approach to describe the environment could be based on constructing a software "bridge" between the hypervisor OS and the hosted virtual machines. Such a bridge would be run on the hypervisor (or other dedicated) operating system and used to query the virtual machines using a standardized interface. Then, the information collected from the virtual machines could be used to advertise the hypervisor "contents" in the provider management network. However, the approach suffers from at least two drawbacks:

- lack of flexibility – in case of a change in a single VM state, the whole hypervisor advertisement would need to be canceled and re-sent,
- introduction of a potential security risk – it is at least discussable whether introducing the possibility to expose (presumably simple) management operations to the provider is worth constructing additional communication channels on each of the hypervisors.

Due to the listed drawbacks, a slightly more complicated, but less invasive and more flexible approach was chosen. Instead of proposing an interface for hypervisor-VM communication, the toolkit requires all parties to issue their advertisements using the same protocol using the networks they are connected to. The advertisements are then processed by dedicated entities that are responsible for:

- feeding the infrastructure-related repositories with data,
- integrating the user-defined services with provider management tools by combining the contents of the advertisements issued by the virtual machines and hypervisors.

Operating upon a simple abstraction, the toolkit leverages a direct, multicast based discovery based on the Simple Service Discovery Protocol [11]. We assumed the whole user network including both the user private premises and the cloud provisioned by the provider to form a single IP multicast domain. Similarly, we also assumed the provider management network to form a single IP multicast domain.

The following subsections discuss briefly main functions of the toolkit and introduce the elements responsible for implementing the requested functionality.

Updating repositories

In the presented approach, two distinct kinds of devices (in the UPnP sense) are defined and advertised. One of them represents a hypervisor that announces the list of hosted virtual machines' identifiers as well as its services (starting/stopping/suspending a VM, etc.).Hypervisor advertisements are published in the provider's management network. The other kind of UPnP device represents a VM and is advertised inside users' networks. The services provided by the virtual machines are known in advance neither to the hypervisor OS, nor to the management system, because that would negate the assumed independence between the two kinds of devices. Therefore, in order to have a complete view of management operations, the interested parties need to merge advertisements coming from two different sources.

Both kinds of advertisements are of interest to configuration-related repositories. Therefore, in order to gather the data necessary to keep the repositories up to date, the elements of the toolkit must be present not only in all user networks, but in the management network as well. The components present in the management network feed the hardware configuration repository as well as the virtual infrastructure configuration repository with data that reflect the placement of virtual machines on hypervisors. Moreover, they track the changes caused by additions or removals of hardware as well as additions, removals or migrations of virtual machines. These goals are accomplished by:

- augmenting the hypervisor operating system with a service that advertises the devices (hypervisors) along with their sub-devices (virtual machines), and
- augmenting the repositories with advertisement trackers plugged into the management network to gather the advertisements (see Fig. 3).

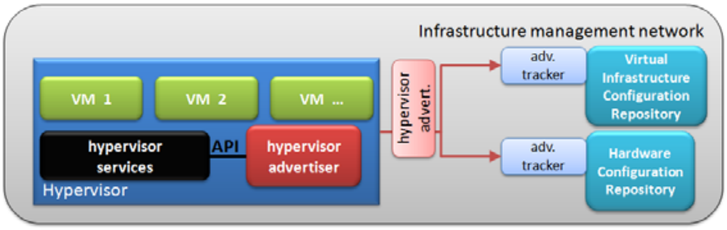

Infrastructure management network
VM 1
VM 2
VM ...
hypervisor services
API
hypervisor advertiser
Hypervisor
hypervisor advert.
adv. tracker
Virtual Infrastructure Configuration Repository
adv. tracker
Hardware Configuration Repository

Fig. 3. Hypervisors' advertisements flow in the management network

The virtual infrastructure configuration repository supports not only the provider, but also the user infrastructures by providing a central point of virtual machines' operating systems configuration. The presented toolkit is capable of feeding the repository with information regarding particular VMs. Technically, it augments the repository with a tracker that processes the advertisements sent by virtual machine advertisers, which are run inside VM operating systems (see Fig. 4).

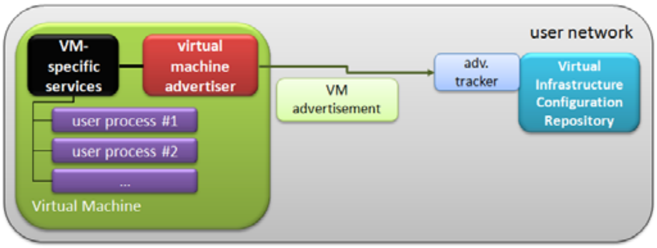


Fig. 4. Virtual machine advertisements flow in a user's network

Combining hypervisor and virtual machine advertisements

UPnP advertisements carry not only device descriptions, but also descriptions of services. In order to make a full use of the services, the functionality of the aforementioned hypervisor and virtual machine advertisers was enhanced to bridge SOAP calls originated by management applications.

The hypervisor advertisements carry descriptions of virtual machine lifecycle management functions, which can be called using SOAP by the provider's IMS. SOAP calls are then converted by hypervisor-specific UPnP bridge to local Xen Management API [12] calls.

The VM advertisement by default describes no operations. However, it can be modified by the user. To make such operation straightforward, the VM-specific UPnP bridge is implemented as an OSGi [13] service, which is deployed in an OSGi container run by the VM. In case the user wants to expose an interface, it can be easily accomplished by registering another OSGi service in the same container. Such a service is noticed by the bridge, which in turn takes care of updating the VM advertisement and then converts the incoming SOAP calls to local OSGi calls appropriately.

As depicted in Figure 5, there are only two entities that process both hypervisor and virtual machine kinds of advertisements — the virtual infrastructure configuration repository and management application. Note however, that both entities do not forward the advertisements, but consume them. Moreover, by keeping the hypervisor advertisements decoupled from virtual machine ones, it is feasible even to track an event of virtual machine "de-virtualization", i.e. running the VM operating system image on a dedicated hardware host.

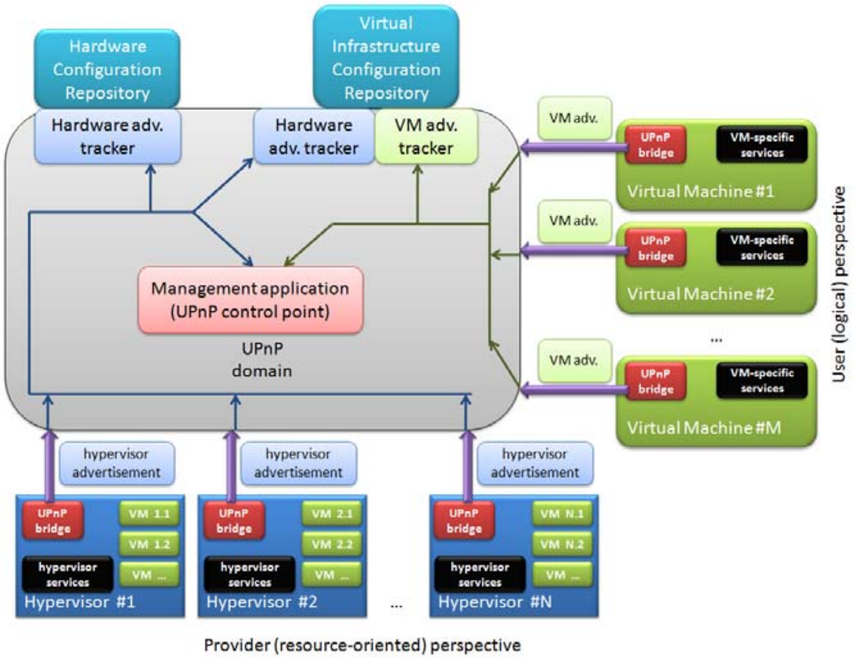
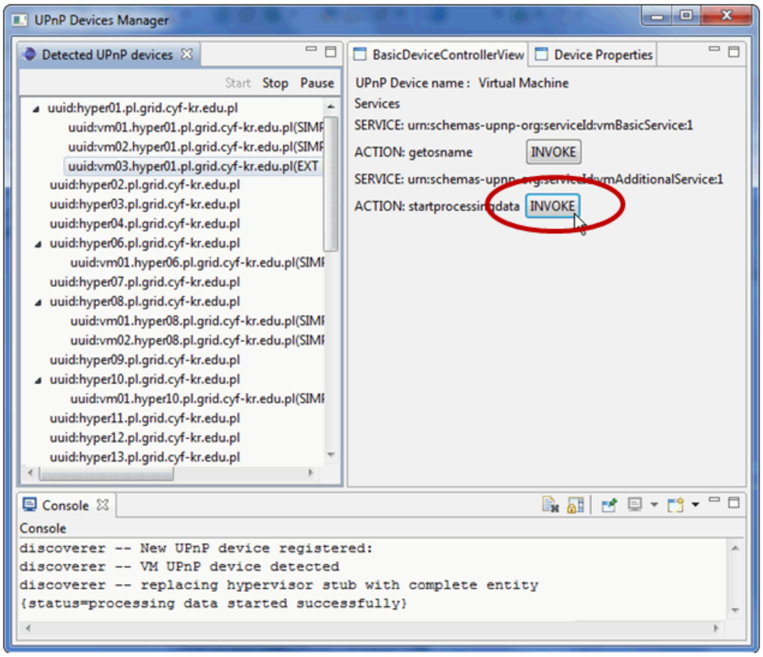


Fig. 5. The flow of all UPnP advertisements in the proposed environment

Both kinds of device advertisements can be received by a provider's IMS, which can aggregate the data by matching the virtual machine identifiers and in that way build up the complete, structured representation of the infrastructure state at runtime. The following subsection briefly describes a simple management application that demonstrates the capability.

Simple management application

To illustrate its capabilities, the presented prototype of the UPnP-based virtual machine discovery and management toolkit is equipped with a simple management application (UPnP control point) that is able to track both hypervisor and virtual machine kinds of advertisements, match them, and display the operations exposed by VM administrators. Figure 6 contains a screenshot of the application.



uuid:hyper01.pl.grid.cyf-kr.edu.pl
- uuid:vm01.hyper01.pl.grid.cyf-kr.edu.pl(SIMF
- uuid:vm02.hyper01.pl.grid.cyf-kr.edu.pl(SIMF
- uuid:vm03.hyper01.pl.grid.cyf-kr.edu.pl(EXT

uuid:hyper02.pl.grid.cyf-kr.edu.pl
uuid:hyper03.pl.grid.cyf-kr.edu.pl
uuid:hyper04.pl.grid.cyf-kr.edu.pl
uuid:hyper06.pl.grid.cyf-kr.edu.pl
- uuid:vm01.hyper06.pl.grid.cyf-kr.edu.pl(SIMI

uuid:hyper07.pl.grid.cyf-kr.edu.pl
uuid:hyper08.pl.grid.cyf-kr.edu.pl
- uuid:vm01.hyper08.pl.grid.cyf-kr.edu.pl(SIMI
- uuid:vm02.hyper08.pl.grid.cyf-kr.edu.pl(SIMI

uuid:hyper09.pl.grid.cyf-kr.edu.pl
uuid:hyper10.pl.grid.cyf-kr.edu.pl
- uuid:vm01.hyper10.pl.grid.cyf-kr.edu.pl(SIMI

uuid:hyper11.pl.grid.cyf-kr.edu.pl
uuid:hyper12.pl.grid.cyf-kr.edu.pl
uuid:hyper13.pl.grid.cyf-kr.edu.pl

UPnP Device name : Virtual Machine
Services
SERVICE: urn:schemas-upnp-org:serviceId:vmBasicService:1
ACTION: getosname
SERVICE: urn:schemas-upnp-org:serviceId:vmAdditionalService:1
ACTION: startprocessingdata

```
discoverer -- New UPnP device registered:
discoverer -- VM UPnP device detected
discoverer -- replacing hypervisor stub with complete entity
{status=processing data started successfully}
```

Fig. 6. A management console displaying a service exposed by a virtual machine

The management application presents a two-level tree view of the runtime environment. At the first level, the hypervisors are presented. The second-level entries contain descriptions of the virtual machines deployed on specific hypervisors. At first, until a VM-specific advertisement is discovered, the entries could contain only an ID of the virtual machine. Later, the entry is extended to represent the list of advertised, VM-specific services.

5. Summary

The presented software leverages the UPnP mechanisms to construct a logical device hierarchy that represents the state of a provider's infrastructure and the current VM deployment state. The hierarchy can be used to update provider's repositories that collect information regarding the current configuration of the infrastructure entities.

The presented solution provides for user-provider collaboration while preserving the independence between respective parties, by implementing general-purpose UPnP bridges that advertise user services and convert remote calls to local ones. The services can be called either from the users' or provider's management applications. Moreover, thanks to the capabilities of UPnP, the hierarchy can be extended to contain logical subdevices of virtual machines.

The tools described in the article need to be developed further to provide granular access control mechanisms based on the provider repositories' contents. Moreover, to facilitate automatic processing, annotations regarding operation semantics could be desirable.

Acknowledgements

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] *Virtualization: A foundation of as-a-Service & Cloud.* Technical report, Capgemini, 2011.

[2] *Trends in Enterprise Virtualization Technologies.* Technical report, F5 Networks, Inc., 2009.

[3] OpenNebula.org. *OpenNebula 2.2 Documentation*, 2011.

[4] *Hitachi Unified Compute Platform Solution Profile.* Technical report, Hitachi Data Systems, 2011.

[5] *Manage your private/hybrid clouds from your own data center.* Technical report, enStratus Networks, 2011.

[6] *Cloud Computing Use Cases Whitepaper.* Technical report, Cloud Computing Use Case Discussion Group, 2010.

[7] *Polish Infrastructure for Supporting Computational Science in the European Research Space (PL-Grid).* http://www.plgrid.pl.

[8] Radziszowski D., Zieliński K., Zieliński S.: *Monitoring GRID resources: JMX in action.* TASK Quarterly (scientific bulletin of Academic Computer Centre in Gdansk), 8(4), 2004, pp. 487–501.

[9] Kosińska J., Kosiński J., Zieliński K., Zieliński S.: *Flexible organization of repositories for provisioning cloud infrastructures.* [in:] Proc. of KU KDM 2010, Zakopane, Poland, March 2010.

[10] UPnP Forum. *UPnP^{TM}Device Architecture 1.1*, 2008.

[11] Goland Y. Y., Cai T., Leach P., Gu Y., Albright S.:. *Simple service discovery protocol/1.0.* Internet Engineering Task Force (IETF) Internet Draft, 1999.

[12] Mellor E., Sharp R., Scott D.: *Xen Management API.* XenSource Inc., 2008.

[13] OSGi Alliance. *OSGi Service Platform Core Specification, Release 4, version 4.3*, 2011.