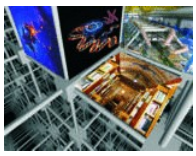


BiG Grid
the dutch e-science grid

BiG Grid

Virtualization of worker nodes

Working group progress report

Completion of phase 1

Document identifier:	BiGGrid-VMWG-M1-v1.0
Date:	02/02/10
Activity:	Virtual Machine working group
Authors:	P. van Beek, M. van Driel, S. Klous, R. Starink, R. Trompert
Document status:	Final
Document link:	https://wiki.nbic.nl/index.php/BigGrid_virtualisatie

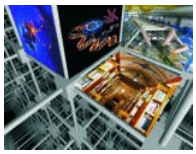
Abstract

The Virtual Machine (VM) working group summarizes the results of their feasibility study to provide virtualized worker nodes on BiG Grid resources. This documents marks the completion of phase 1, as specified in the working group charge.

The first part of the document contains a description of the collected Use Cases for virtualized worker nodes from different user communities. From these Use Cases and the Big Grid operational constraints, a set of requirements has been extracted. These requirements should be respected by the Big Grid VM management system as will be designed in phase 2.

In the second part of the document, an overview is presented of different ongoing efforts outside BiG Grid that aim to integrate grids and clouds. We discuss if and how these external efforts fit the needs of BiG Grid. A dedicated section discusses the foreseen limitations and suggests an approach for providing a maintainability and scalable solution. We conclude this progress report with a set of recommendations and a proposed test-plan on a proof-of-concept (POC) system. The tests should demonstrate the opportunities as well as reveal the technical constraints of an environment with virtualized worker nodes. A design document for the POC system will be available in February.

Table of Contents



Executive summary

The main purpose of this progress report is to inform the executive team about the status of the VM working group research. In phase 1 we investigated the feasibility of a VM management system to provide virtualized worker nodes to the BiG Grid user communities.

The working group concludes that there is a broad interest from the BiG Grid user communities for virtualized worker nodes. Additional flexibility in operating system and applications as well as protection from the complex grid environment have been identified as the main benefits from end-user perception. Site administration expressed serious concerns about VMs deployed within the trusted network fabric, or even in a demilitarized zone, by *e.g.* end-users or VO managers. The working group aims to provide a balanced design for a VM management system with sufficient end-user flexibility and acceptable risk on site integrity.

The working group proposes a road-map in order to design a VM management system. The road-map is based on the deployment and implementation of the relevant tools, technologies and mechanisms in a small POC environment. The POC environment can be used for performance testing and to investigate a number of issues that might hamper scalability. A detailed list of studies as planned in the proof-of-concept environment, is provided in Appendix A. A design document for the proof-of-concept environment will follow in February. The POC design document can serve as input for the design of a production quality VM management system in phase 2.

In this report, a set of recommendations is already provided to overcome the most obvious scalability problems. A complete list of recommendations can be found in the conclusions and recommendations section of this document. The working group recommends the following sub-projects to the executive team to facilitate development of a VM management system for BiG Grid:

- Design and implement a POC system based on the OpenNebula cloud computing project, this will be part of “phase 2” of the VM working group activities.
- Design and implement a tool that allows end-users to supply VM images, preferably in a joined project of middleware developers and application domain analysts.
- Initiate an effort to investigate, implement and enforce security mechanisms and policies in the POC environment.

A number of risks related to negligence, security and software licenses have been identified by the working group. Although the working group questions the effectiveness of transferring liability to end-users, it feels that investigation of legally binding contracts and formalization of responsibilities are out of its scope. The executive team is requested to take a stand on a number of liability issues as formulated in the conclusions and recommendations section of this document.

1. Working group charge

Objectives

- Phase 1: Provide a design and Proof-of-Concept for virtualized worker nodes that fulfill requirements of different user communities (Q2 and Q3 2009). Take into account the different efforts already under way in this respect.
- Phase 2: focus on attainable first implementation in Q4 of 2009

Resources (working group members)

- Sander Klous (NIKHEF, chair)
- Pieter van Beek, Ron Trompert (SARA),
- Marc van Driel (NBIC)
- Ronald Starink (NIKHEF)

Other users and communities must be part of this effort but not necessarily part of the working group

'Opdrachtgevers'

- BiG Grid Executive team

Boundary conditions

- System must function on the type of BiG Grid resources available now

Planning

- start: March 1, 2009
- end Phase 1: June 1, 2009
- end Phase 2: December 31,2009

Activities

- Phase 1:
  - Attain requirements from different user communities and Use Cases.
  - Take into account different efforts on worker node virtualization in Europe and check their applicability with respect to the above Use Cases.
  - Write up of first design
- Phase 2:
  - Implementation of Proof-of-Concept on a dedicated resource at NIKHEF, SARA or Life Science Grid.
  - Build customized VM images for one or two selected application domains.

Deliverables

- Phase 1: design for a Proof-of-Concept environment
- Phase 2: implementation of Proof-of-Concept environment

2. VMs: motivation and Use Cases

There is a broad interest from the BiG Grid user communities for virtualized worker nodes. Additional flexibility in operating system and applications as well as protection from the complex grid environment have been identified as the main benefits from end-user perception.

The majority of grid users find the grid environment too complex. Users might be unfamiliar with the operating system installed on the grid, or the user rights to install specific software, or they might lack technical skills. VMs provide an isolation layer between user application and this complex environment. Applications running inside a VM can be deployed without modification on a local desktop, a batch cluster, a grid site, or multiple grid sites. Consequently, an end-user starting from scratch on a VM significantly reduces the overhead of these different scaling steps. Workflows might even change due to this increased transparency, *e.g.* a user might decide to perform a number of interactive tasks in a VM on the local desktop. Subsequently, this VM can be replicated to the grid for a computational intensive step. The resulting VM might be started locally to evaluate the results. Such workflows, combining interactive and batch processing, are difficult to achieve without VMs.

In the following sections a number of Use Cases from the BiG Grid communities are described. First we discuss two High Energy Physics (HEP) Use Cases, followed by two Use Cases from Bioinformatics.

2.1 High Energy Physics Use Cases

Analysis on VMs

Most of the high energy physicists run their analysis on remote hosts with pre-installed experiment software. Interactive clusters of machines are available at CERN and Nikhef with the latest versions of the experiment software and grid middleware installed. The physicist logs into the cluster, runs a few configuration scripts and has access to both the experiment software and the grid user interface. An experiment specific grid access application (Ganga for ATLAS and LHCb, Alien for Alice) facilitates the conversion from a minimal local analysis to a grid submission.

In order to compete with this scenario, VM images can be prepared with the latest experiment software and grid middleware. These images should be available both for deployment on a desktop and on a grid site. Typical release cycles of experiment software are in the order of 3 to 6 months. In between only security updates and application patches are (centrally) applied to the VM image.

The physicist runs a minimal analysis on the local VM, which provides access to the grid services, like raw data storage, file catalogs and configuration databases. Once the physicist is satisfied with the results of the minimal analysis, a full analysis (*e.g.* over a larger data set, or a parameter scan over 1000 different combinations) is run on VMs deployed on grid resources. These VMs provide identical access to the grid services and

preferably allow for interactive access to do debugging.

The typical high energy physicist has no intention to become a system administrator. Just a few personally developed libraries and scripts are needed in addition to the standard experiment software and grid middleware. These libraries can be installed in user space, so no privileged access is required to the VM. Note that it must be possible for the physicist to store user space image modifications persistently, *e.g.* on a user mountable disk.

Example 1: Muon trigger efficiency analysis for the ATLAS experiment.
Datasets are analyzed with the latest ATLAS production software to extract the efficiency of the trigger system for events with muons. The analysis will follow the procedure as outlined above.

Large scale distributed productions
The HEP experiments produce millions of simulated events. These events contain information about the physics interactions and the detector responses. Many jobs are submitted to the Tier-1 centers via a grid interface to acquire the compute power. Management of these large scale productions is a complex task. Hence, the workload is not submitted directly with the jobs. Instead, highly specialized place holder jobs are submitted. Once these jobs run, they register with the experiment workload management systems and receive their assignments. Constructions like this are known as “pilot job frameworks”. All of the HEP experiments run their large scale distributed productions like this.

Virtualized worker nodes can significantly simplify the integration with pilot job frameworks. Experiments no longer need to tune their pilot jobs for all the different sites to set up correct software configurations. They simply create a single VM, containing all the software needed for the large scale production and the integration with the pilot job framework. Instances of the VM can be deployed on all participating grid sites, improving the cross-site consistency.

2.2 Bioinformatics Use Cases

Bioinformatics Use Cases contain a few crucial differences with respect to HEP Use Cases from site operations point of view. The user groups are much smaller and generally diverse. The diversity makes it difficult to maintain one or a few standard images suitable for all users. Smaller groups imply that there will be less or no dedicated specialists to maintain the images.

Scientists of non-HEP Virtual Organizations can often not avoid to do (part of) the system administration themselves. Assisted by application domain analysts, they have to setup the specific tools and applications needed to run their individual analysis on grid resources. Hence, end-users would like to be able to install a wide variety of distributions (Ubuntu, SuSE, Fedora, Windows, etc.), general purpose software packages and libraries (Java, Perl, Python, Jython, Ruby, JRuby, etc.) and application specific software (BLAST,

IPRScan, MAQ). Often, a specific combination of versions (including glibc, libstdc++ and libxml versions) is required, *e.g.* to reproduce previously published results. Some of the Use Cases require multi-core VMs or specific (legacy or exotic) platforms.

Example 2: Identification and classification of G protein-coupled receptors (GPCRs).
First build a high-GPCR content sequence file by blasting the NR database with ~1600 GPCR family consensus sequences. All hits with E-value below 1 will be extracted. Next, classify these sequences with a library of ~1600 GPCR family Hidden Markov Models (HMMs) and extract all significant family members.

Software requirements: BLAST 2.2.16, HMMER 2.3.2, BioPython 1.50.
Hardware requirements: 10 GB disk space, 2GB ram and 15 minutes of CPU time.

Example 3: Run Sparc Solaris binary of bioinformatics tool
Run a Sparc Solaris binary to reproduce previously published results. The source code is no longer available nor is the hardware. The Sparc Solaris distribution can run in an emulator.

2.3 User VMs

Template based VMs
A graphical representation of our current view on the way end-users want to interact with their VM images is shown in Fig. 1. This picture shows the possibility to select from a number of template images. The template image is extensible with user-defined software specific for the VM task. Two modes of operation are available. In single mode the end-user can make persistent modifications to the image. In multi mode, more than one instance of the image runs simultaneously to perform the tasks in parallel. Note that license issues, related to *e.g.* Windows and Mac/OS-X templates shown in the picture, should be addressed.

Non-template based VMs
Besides centrally predefined images, user-defined images aid to the flexibility of the VM/grid setup. These images can be user-specific, but can also be shared with others. Shared images take away the burden of installing specific software and tuning the configuration. Furthermore, user-defined images provide the opportunity to use legacy

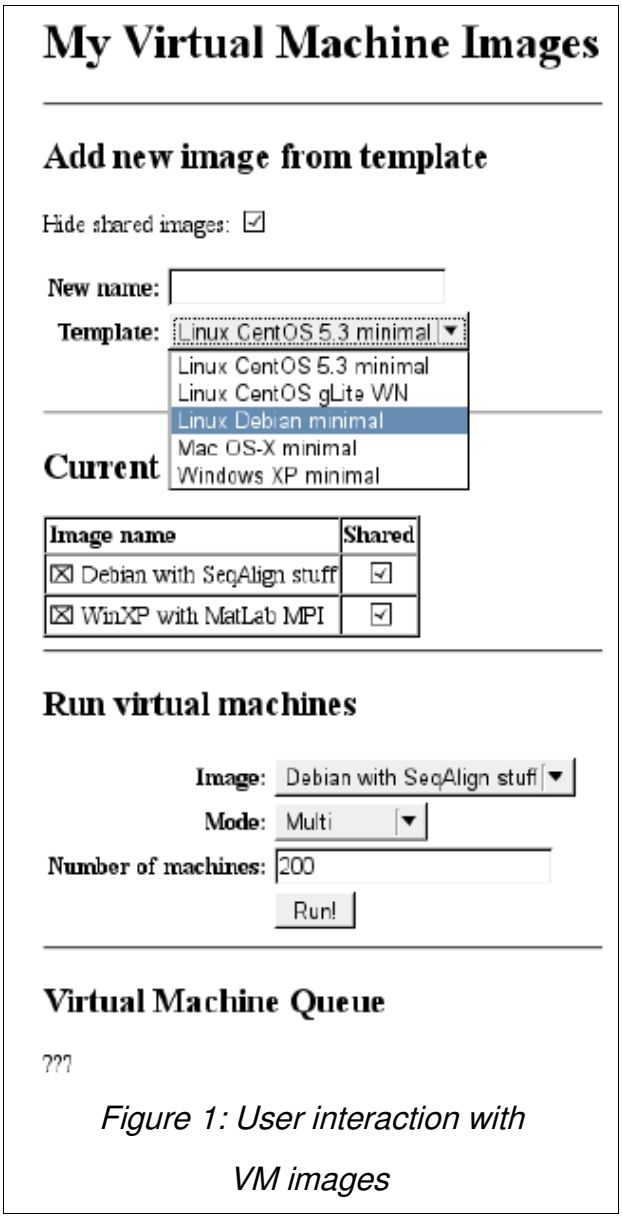


Figure 1: User interaction with VM images

operating systems or software packages in a flexible and scalable fashion.

2.4 System Administration Use Cases

A number of grid services are already running in VMs at BiG Grid sites to improve the resource efficiency and flexibility. Virtualization of worker nodes is expected to be beneficial for site administration as well. Virtualization would *e.g.* improve sand-boxing of resources like CPU, RAM and disk, providing the flexibility needed for multi-core applications. On top of that, VMs allow for better scheduling (suspend, migrate). As a (very) long term goal, virtualization could significantly simplify the handling of multi-core jobs.

Even though there are many potential advantages in VMs from system administration point of view, the working group decided that this is part of site internal strategy and organization. Only minimal coordination is required between operations, user communities and BiG Grid management to establish such site infrastructure upgrades. Hence, such upgrades are outside the scope of the VM working group.

3. Boundary conditions for VM support

Introducing VMs provided by virtual organisations or end-users into the existing grid infrastructure brings new challenges for the grid operations and security teams. This chapter gives a brief summary of these issues, which results in a set of requirements for the implementation of such VMs on the BiG Grid infrastructure. Those security and operational requirements are presented in the next chapter.

It is the responsibility of the grid operations teams to provide a stable and functional grid infrastructure to legitimate national and international grid users, and to ensure the privacy and integrity of data on grid storage systems.

In the current situation, *i.e.* without VMs provided by users or virtual organisations, the operations teams have full control over the software running on all machines. They make sure that no software with (in their eyes) unacceptable problems or known risks is running, which could threaten the stability of the infrastructure or the integrity of the data. The currently used grid site security models rely on the assumption that machines in the infrastructure can be trusted. Introduction of VMs implies that the operations teams no longer have full control over the installed software and that they will have to reconsider the existing security models to accommodate the new and untrusted components in the infrastructure.

At present, the site administrators can easily find out which user processes are executed from a grid job. When VMs are used, this possibility is greatly reduced or even impossible. It is therefore practically impossible to investigate problems on a virtual computing node. That includes the investigation of potential security problems as well as the investigation needed to support user-reported problems.

The sites participating in BiG Grid are used by national and international users. The High-Energy Physics community, one of the large user groups of the BiG Grid resources, predominantly consist of international users. They must also be able to use the resources at the BiG Grid sites after support for VMs has been implemented. Therefore, it is essential that the solution for VM support fits in the existing grid infrastructure or can at least coexist with it in a non-disruptive manner.

More details about the issues described above are given in Appendix B, which also contains possible measures to address the risks that are introduced by supporting VMs. In addition to these measures, policies can be defined to regulate what may and may not be done with VMs. Appendix C discusses some existing policies that are in use today and which can serve as an example for VM policies.

4. Requirements

The Use Cases listed in Chapter 2 conflict in a number of cases with the boundary conditions listed in Chapter 3 and Appendix B. Rather than trying to reach consensus about the validity of each individual statement, the working group decided to first convert the boundary conditions into a set of requirements. The listed requirements are divided over different areas (users, security and operations) and are not expected to be internally consistent. Potential conflicts are identified in section 4.4 and possible solutions for these conflicts are presented in Chapter 6.

4.1 User requirements

The following set of end-user requirements can be extracted from the Use Cases listed in Chapter 2:

The user should be provided with the following functionality:

1. Image management
   a) Customize a VM image (operating system, distribution, installed software packages, user software, provided services in user space).
   b) Upload, download, store (before or after deployment), modify, rename and/or delete VM images.
2. VM management
   a) Deploy, redeploy and/or abort a specifiable number VMs on BiG Grid resources
   b) Specify virtual resources, such as the number of virtual processors, the VM image(s), outbound networking whitelist.

4.2 Security requirements

Security requirements for scenarios with virtualized worker nodes within the trusted network fabric are very different from security requirements for virtualized worker nodes in a DMZ, as discussed in Appendix B and C. Hence, we provide two requirement lists (one for each case) on BiG Grid operations and security. Later in this document, these two cases are mapped to different VM types (Class 2 and Class 3, see Chapter 6).

Virtualized worker nodes part of the trusted network fabric.

1. All policies governing conventional worker nodes apply automatically
2. Virtual worker node images have to be provided by a trustworthy source[1] and have to fulfill the following guidelines:
   a) Stored images must not contain end-user credentials
   b) Images should be provided without packages or applications containing known vulnerabilities that would put the site infrastructure at elevated risk.
   c) Deployment of VMs based on available images with known vulnerabilities that would put the site infrastructure at elevated risk is prohibited.

1. In this context end-users and Virtual Organizations are not considered trustworthy.

d) End-users will not have privileged access to the VM image at any time.
e) End-users will not have inbound access to the VM image at run-time
f) The virtualized worker node will not run services on privileged ports.
g) The virtualized worker node will not run privileged services on any port.

Virtualized worker nodes in a DMZ

The following requirements might be sufficient for virtualized worker nodes in a DMZ.

1. The Grid acceptable-use-policy (AUP) applies automatically.
2. A maintenance plan should be provided with a patch/update procedure for each VM image.
3. Virtualized worker nodes are deployed with minimal network connectivity, *i.e.* white-listed minimal outbound access.
4. Appropriate security monitoring tools have to be in place.
   a) Network traffic has to be monitored for malicious content.
   b) Packages on the VM images have to be monitored for vulnerabilities[2].
   c) Host machines must contain an intrusion detection system.
5. Appropriate measures for guest and host system separation have to be in place.
6. A zero tolerance policy toward users deploying vulnerable VMs should be in place.
7. Minimalistic 24/7 incident response should be available.
8. The DMZ must be part of a separate TLD.

4.3 Operational requirements

The following guidelines will help to set up a reliable, scalable and manageable VM management system on BiG Grid resources:

1. The system must comply with all applicable (BiG) Grid and site local policies and procedures.
2. Tools for managing (transferring, scheduling, starting, pausing, stopping, etc.) VMs and VM images have to be integrated in the existing local resource management and fabric management systems.
   a) This system should handle accounting, respect fair shares and quotas, assign VM credentials and integrate grid authentication and authorization mechanisms.
   b) The system should be reliable[3] and non-interfering[4]
   c) The system and the VMs logging and monitoring tools must comply to and be compatible with site central logging and monitoring tools.
3. Site administration must have transparent access to VM images and the content must be traceable. This includes provenance of experiment data.
4. The number of public IP addresses for worker nodes can not significantly increase.

2. Automated package inspection is difficult to accomplish. Tools exist to do this for RPM based distributions, but a single exploit doesn't provide insight in the vulnerability of an entire system. On top of that, most operating systems are not RPM based.

3. The site down-time should not increase significantly

4. Workload on site administration should not increase significantly and existing site performance should not significantly degrade.

4.4 Potential conflicts in the listed requirements

The most obvious conflicts arise from clashes between user and security requirements. The user requirement to have customizable VM images will most likely conflict with the security requirements to only allow images from trustworthy sources and without known vulnerabilities (in case the VM is part of the trusted network fabric). Also, it is not clear if the security restrictions on privileged access are acceptable for the BiG Grid user communities. These conflicts can be circumvented by running the VMs in a DMZ, which would reduce the impact of incidents on the site infrastructure (but not on the rest of the Internet). However, the security requirements for the DMZ might interfere with the scalability and maintainability requirements from operations. Furthermore, it is not clear how the operational requirements on logging and transparent access can be enforced if VMs are fully customizable.

BiG Grid requirements are compared to the features of existing external products for handling VMs in section 6. In that comparison the working group specifically focused on solutions offered by the different products to mediate between conflicts of user, security and operational requirements as listed in this paragraph.

5. External efforts to integrate grids and clouds

Developers of a number of existing cloud computing products expressed interest to work on the integration of grid and cloud technologies. Most notably Nimbus and OpenNebula developers presented ongoing activities [1],[2] to provide interfaces between grid middleware and their VM scheduling, authorization and authentication frameworks. On the opposite side of the spectrum, a number of large grid sites (CERN, INFN, DESY/FZK) are integrating the deployment of VMs with their job submission frameworks [3],[4],[5].

5.1 Cloud computing products

Besides Nimbus [6] and OpenNebula [7], the working group considered Eucalyptus [8] (Open Source) and investigated the feasibility of deploying VMWare [9] (Commercial). This section contains the evaluation of all of these products.

Note that many other cloud computing products are available, like Enomaly [10] or AbiCloud [11]. These products seem to have smaller scope and are primarily focused on commercialization of cloud computing technology. Furthermore, we briefly looked into Ganeti [12], a Google backed solution for virtual server management. Ganeti provides a strong and reliable back-end for cluster management of VMs and VM images. Unfortunately, no front-end development is foreseen (*i.e.* no scheduling, authentication and authorization implementation).

Nimbus

Nimbus is an open source toolkit that turns a cluster into an infrastructure as a service (IaaS) cloud. The Nimbus cloud client allows provisioning of customized compute nodes with a leasing model based on the Amazon EC2 service. Nimbus (formerly known as the virtual workspace service) is developed in part within the Globus Toolkit 4 framework [13].

Feature list

- User interfaces.
  - Web Services Resource Framework (WSRF) interface [14].
  - Amazon EC2 Web Service Description Language (WSDL) interface [15].
  - Protocols supported through Apache Axis based Globus Toolkit Java Container.
- Remote deployment and lifecycle management of VMs.
  - Deploy, pause, restart and shutdown VMs.
  - Specify image, network configuration, resource allocation on deployment.
  - Request VM status (lifecycle state, IP address, etc.).
- Scheduling features
  - Support for VM groups (image groups, resource allocation groups) and ensembles (groups of groups), to allow for co-scheduling (even on best effort basis).
  - Support for accounting and Fair Share scheduling.
  - Support for site scheduler integration.
    - Nimbus allocates nodes from the site scheduler (Torque/PBS).

        - Nodes are returned to the site scheduler after VM completion.
- Authentication and Authorization features.
    - Based on the Grid Security Infrastructure (GSI).
    - Authorization based on VOMS information.
    - Authorization policies can be applied to image and network access.
        - Network isolation provided with bridging tables (ebtables).
    - Support for group policies (image access, quota and reservation limits).
- Just in time configuration.
    - Contextualization via the context broker.
        - Supports insertion of credentials into the VM.
            - Avoiding the need to store credentials in the VM image.
        - Customize files in the VM on a per launch basis.
        - Shared, secure information context for all VMs of a group.
- Support for Xen (preferred) and KVM (under development).

OpenNebula

OpenNebula is the virtual infrastructure manager of the RESERVOIR virtualization project [16]. RESERVOIR is part of the FP7 program [17] and works together with EGEE [18] to explore how EGEE associated institutes could benefit from cloud technology [19].

OpenNebula is developed by UCM [20] and orchestrates storage, network and virtualization technologies to enable the dynamic placement of (groups of interconnected) VMs on distributed infrastructures. OpenNebula provides internal and external cloud administration user interfaces for the full management of the VM infrastructure.

Feature list

- User interfaces.
    - Unix-like command line interface.
    - OGF Open Cloud Computing Interface (OCCI) API [21].
    - OpenNebula Cloud API (OCA) through XML-RPC [22].
    - Ruby bindings to OCA.
    - Support for the libvirt virtualization API.
- Remote deployment and lifecycle management of VMs.
    - Deploy, pause, migrate, restart and shutdown VMs.
    - Specify image, network configuration, resource allocation on deployment.
    - Request VM status (lifecycle state, IP address, etc.).
    - Support for hybrid systems, only unidirectional ssh access is required.
    - Support for LVM block devices and copy-on-write (under development).
- Scheduling features
    - Support for workload and resource-aware matchmaking policies.
    - Support for lease based scheduling with the optional Haizea scheduler.
        - Leases can expire after a specified wall clock time.
        - Haizea has advance reservation capabilities.
    - Support for VM groups (multi-tier services).
- Authentication and Authorization features.
    - User management, but only a placeholder implementation.

- Network isolation provided with bridging tables (ebtables).
- Just in time configuration.
  - Support for auto-configuration at boot time (advanced contextualization).
  - Supports insertion of credentials into the VM.
  - Customize files in the VM on a per launch basis.
- Support for Xen, KVM and VMWare.

Comparison of OpenNebula and Eucalyptus

A detailed comparison between the OpenNebula and Eucalyptus functionality is provided by the lead developer of OpenNebula, Ruben Montero, on the OpenNebula list [23]. The information is fairly recent (dated July 2009).

Advantages of OpenNebula over Eucalyptus

- OpenNebula provides a superior administration interface. VMs can be migrated, suspended and so on. Eucalyptus only provides the functionality offered by EC2 (*i.e.* no suspend or migration of any kind).
- OpenNebula provides a flexible physical host interface to monitor, and manage the physical resources of the cloud. This management interface is missing from the current Eucalyptus releases.
- OpenNebula provides better placement policies, either with its default matchmaking algorithm that can be tuned with user-driver consolidation hints or with the optional "Haizea" scheduler. Eucalyptus uses a round-robin approach.
- OpenNebula allows management of sets of VMs and the network configurations (VLANs) between them (public and/or private). Network isolation is provided through bridging tables (ebtables). Eucalyptus does not allow definition of virtual networks.
- OpenNebula has the ability to push any context data to a VM, so it can auto-configure at boot time (*e.g.* software licenses, credentials, VM environment data, etc.). Eucalyptus does not provide this advanced contextualization.
- OpenNebula provides a powerful API to extend its capabilities. Either to build applications on top or to integrate any storage, virtualization, or network technology. Eucalyptus only provides the EC2-soap interface to interact with it.
- OpenNebula can build hybrid clouds. So you can either deploy your VMs locally or in another cloud. This can not be done with Eucalyptus.
- OpenNebula provides a libvirt interface. The infrastructure can be controlled with the libvirt API or using its related tools (*e.g.* virsh).

Advantages of Eucalyptus over OpenNebula

- Eucalyptus provides an EC2 like interface. OpenNebula 1.2 does not provides a simplified cloud interface. OpenNebula 1.4 includes a cloud API to implement any cloud interface, but only a subset of the EC2 interface is implemented as an example.
- Eucalyptus contains an S3 implementations. There is no such functionality in OpenNebula 1.2. OpenNebula 1.4 includes a simple image management tool.

On top of above comparison, Eucalyptus cooperation with educational and research

institutes appears to be affected negatively by the recent establishment of an associated company to deliver extensions and commercial support for their Open Source product. This was reported by representatives from *e.g.* the Baltic Grid during the EGEE'09 conference in Barcelona. Based on our current information, Eucalyptus is clearly outperformed and is no longer taken into account in our evaluation.

VMWare

A first estimate of licensing costs for VMWARE is in the order of 1000 Euro per box and an additional 5000 Euro per site. Scaling up to a reasonable amount of resources (50 to 100 hosts) would result in an investment of about 100 kEuro. Vendor lock-in becomes a non-negligible risk with investments of this size. Although the VMWare management tools are very smooth, problems related to integration of authentication, authorization and batch scheduling with an existing grid infrastructure are not resolved. Hence the working group concludes that a VMWare implementation is not a realistic solution for BiG Grid.

5.2 Grid site developments

The requirements presented in this document are more demanding than the solutions developed by other grid sites can currently provide. The large grid sites aimed their developments at introduction of VMs with minimal modification to the existing infrastructure. Three different approaches are identified and discussed in this chapter. INFN has done extensive scripting to allow deployment of virtual worker nodes on demand. CERN introduced a queue monitoring system to adapt the number of VMs automatically. DESY and FZK tried to keep the VM management as simple as possible. They deploy VMs with the (already existing) preprocessing functionality of their batch scheduler and clean up in the post-processing step.

INFN: Worker nodes on demand

At INFN jobs can be submitted to classical worker nodes or to virtual worker nodes, as shown in Fig. 2. The Job Submission Description Language (JSDL) contains a few keywords that allow specification of the VM image on job submission. Specialized “bait” VMs satisfy the VM image criteria specified in JSDL and attract jobs. The job is launched by the bait VM inside a newly deployed VM with the correct image. New images will be installed by site operations on request of the user communities, work is ongoing to allow installation of new images via HTTP.

CERN: The VM Orchestrator

CERN has worked together with “Platform” (the company behind the LSF batch system) on a tool called the 'VM Orchestrator' (VMO). VMO monitors the requested VM images as specified in the JSDL descriptions in the job queue. An increasing number of requests for a certain image type will result in additional deployment of VMs with this image type (at the cost of VMs with other image types). In other words, VMO dynamically balances the VM resources with the number of job requests. This approach works well when the number of different VM images is limited (*e.g.* it was tested with SLC4 versus SLC5 images) and when the VMs are tightly integrated with the batch scheduling system. It doesn't support the flexibility BiG Grid requires to schedule customized images provided by users or Virtual

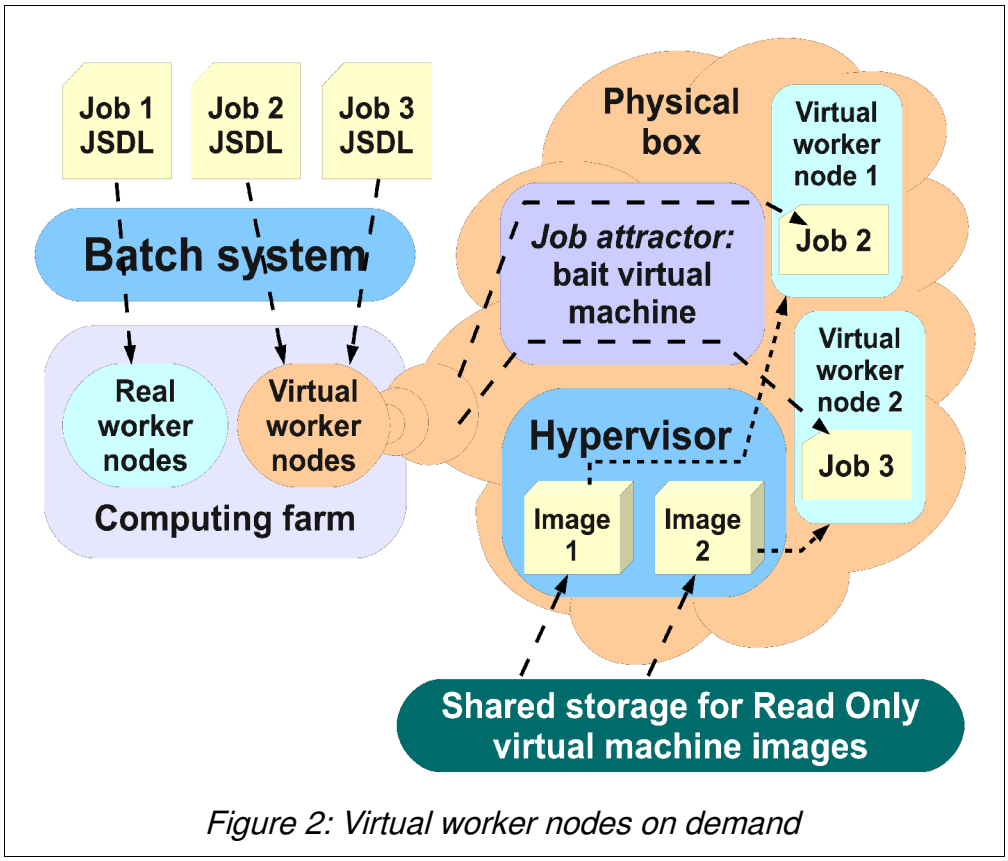


Figure 2: Virtual worker nodes on demand

Organizations. CERN is currently investigating the replacement of VMO with OpenNebula, which might result in a more flexible solution.

DESY and FZK: Dynamic Partitioned Cluster

The workload management system provides a wrapper script to the batch server. The wrapper script exploits the Prologue and Epilogue hooks of the batch server to deploy and shutdown the VM respectively. Furthermore, the wrapper script monitors the status of the VM and starts the job inside the VM. This is a very simple and efficient approach, but it doesn't provide any image management infrastructure nor does it offer the required flexibility of the BiG Grid user communities.

6. Matching BiG Grid requirements with external efforts

All of the external efforts presented in chapter 5. are providing parts of the functionality needed by BiG Grid, but none of these products are production ready or provide a plug and play solution for BiG Grid. In this chapter, the features of the different products are compared to the potentially conflicting requirements as listed in section 4.4 to evaluate of their suitability for the BiG Grid proof-of-concept environment. In summary, we are looking for products able to:

- Deploy different kinds of customized virtual images.
  - Site provided, similar to current worker nodes (Class 1).
  - From trustworthy sources, running as part of the trusted network fabric (Class 2).
  - Generated by end-users running in a DMZ (Class 3).
- Configure appropriate (network) security restrictions.
  - Preferably for a hybrid network infrastructure (in part trusted and in part DMZ).
- Provide a maintainable and scalable solution.
  - Integrated with the existing scheduling and accounting systems.
  - With appropriate logging and monitoring for all VMs.

6.1 Deployment of customizable images

The cloud computing products natively support the provisioning of customized VM images by end-users. Unfortunately, their implementations do not support a direct solution for the provisioning of images from trustworthy sources. They do contain extensive APIs, which enables BiG Grid to develop a component that respects such requirements.

The products developed by the grid sites allow to select customized VM images as well. However, their implementation is quite static and requires human intervention. BiG Grid will have to make significant investments in order to meet the flexibility requirements of their user communities with these products.

6.2 Configuration of appropriate security restrictions

The security related features of the cloud computing products include “advanced contextualization” and “network isolation”. These features allow (at least partial) implementation of the required security limitations and restrictions for VMs in a DMZ. OpenNebula provides “support for hybrid systems”, which is especially relevant when a fraction of the VMs is deployed within the trusted network fabric.

The products developed by the grid sites do not provide dynamic security related features. Of course, static configuration of IP tables and bridging tables remains possible, but compared to the cloud computing products, additional effort should be expected to meet the security requirements put forward by BiG Grid operations.

6.3 Providing a maintainable and scalable solution

All of the listed products are still in a development stage. Hence, maintainability and scalability are serious concerns. In this chapter, two expected areas of development are discussed, related to maintainability and scalability:

- Integration with the BiG Grid accounting and fair share scheduling system.
- Scalability of VM image management and image distribution.

Implementation and maintenance of these features requires close collaboration with the developers of the products. Here we evaluate the possibilities to establish joined projects with the developers on the aforementioned topics.

- The close ties between RESERVOIR and EGEE indicate possible interest from OpenNebula to combine efforts. This indication is confirmed by a joined project on authentication, authorization and scheduling by Nikhef grid middleware developers and OpenNebula, which started as a follow up of EGEE'09.
- Collaboration with Nimbus, part of the Globus Toolkit and the US based Open Science Grid (OSG), is expected to be more difficult. It is not clear if the approach taken by the Nimbus developers with the scheduler and the user interface is different from the preferences of BiG Grid. Before we pursue a proof-of-concept based on this product, we need to understand the implications on maintainability and scalability.
- Smooth collaboration is expected with the other grid sites if BiG Grid decides to deploy one of their products. Some of the developers are well known at Nikhef. In contrast to the more generic cloud computing products, the grid sites products are specifically developed with a high throughput computing infrastructure in mind, aimed at HEP. Hence, high quality support for problems related to typical performance bottlenecks in large computing centers is expected. On the other hand, support for problems related to the broad spectrum of user communities in BiG Grid will be worse.

Here the two expected areas of development are discussed in a bit more detail, before we make a final recommendation about the product to select for the BiG Grid POC environment.

Accounting and Scheduling integration

The grid site products provide native integration of VM management with their accounting and fair share scheduling[5] system. Clearly, more efforts are needed to integrate one of the cloud computing products.

- Nimbus already provides support for fair share scheduling and site scheduler integration. Their mechanism to allocate nodes from the site local scheduler effectively turns VM requests into job requests. This is a nice intermediate solution for VM scheduling, although more advanced scheduling options with VMs would not be available.

5 Fair-share scheduling is a scheduling strategy for computer operating systems in which the CPU usage is equally distributed among system users or groups, as opposed to equal distribution among processes.

- Native integration of the OpenNebula scheduler will become available over the next 6 months through plugins in the associated grid middleware components: the Execution Environment Service (EES) and the Batch Local ASCII Helper (BLAH). Fair Share scheduling is not yet planned. This is a crucial missing feature for large scale deployment of VMs.

Note that the cloud computing products introduce a parallel VM submission system next to the existing batch system. This additional system has to be maintained, at least on the short to medium term. On the long term, a single scheduler might be able to provide scheduling for both jobs and VMs.

Closely related to the scheduler integration is the development of a grid aware 'Amazon like' User Interface. In fact, Nimbus provides such a user interface, which might (almost) be compatible with the requirements of the non-HEP BiG Grid communities. The other products provide a command line interface, compatible with the existing grid job submission interface. A significant amount of work is expected in order to provide an 'Amazon like' User Interface for these products.

Scalability of image distribution

Typical grid workloads (like *e.g.* the parameter scan Use Case mentioned in section 2.1) are characterized by peaks and long tails in the number of submitted jobs[24]. A VM image is expected to be deployed on at most a few nodes during preparation. Once the preparation is over and modifications are made, the VM will run on many nodes in parallel for a relatively short period of time. After that, usage will slowly fade out. In such scenarios, distribution of VM images has to be optimized to avoid network congestion during peak loads.

As an example assume the following Use Case, typical for *e.g.* a parameter scan:

- Submit 1024 single core VMs, running on a 128 core cluster.
- Each VM runs for 1 hour.
- A VM image size of 10 GB.
- Each node connected via 1 Gbps Ethernet to the image server.

Transferring the image will take (10 GB x 8) / (0.8 x 1 Gbps) = 100 seconds, with 0.8 the efficiency of the TCP/IP connection. In the most optimistic scenario 36 images can be transferred in series per hour over a 1 Gbps link. The image server should be able to continuously provide over 3 images in parallel to occupy all 128 cores with 1 hour jobs. Hence, with brute force image transfers a dedicated network will be required with a 10 Gbps switch, a 10 Gbps connection to the image server and 1 Gbps connections to each node.

OpenNebula provides a scalable solution for VM image distribution through the support of LVM block devices and copy-on-write clones [25]. With copy-on-write, many VMs can boot from the same image, without the need for multiple copies. The feasibility to implement similar features in the other products should be investigated. The benefits of copy-on-write, combined with (device level) caching are significant, as is shown when the example is

revisited.

Suppose that the 128 core cluster contains 16 nodes with 8 cores each. The image will be transferred once to each node (*i.e.* only 16 times in total, instead of 1024 times). After the transfer, all VMs deployed on the node can be booted from the cache with copy-on-write clones. The overhead caused by the image transfers reduces with a factor 1024/16 = 64, *i.e.* almost 2 orders of magnitude. This most likely eliminates the need for a dedicated image distribution network. Storage space on the nodes would typically allow for caching of 5-10 images. So, the advantage of caching disappears if many different kinds of image are deployed at the same time.

Quota management will be crucial for image storage. It is impossible to store a new image for every modification made to the original. Copy-on-write makes storage of modifications more efficient (only differences with respect to the original can be stored). However, this will not scale well and proliferation of images should be avoided. The availability of a small user disk is in many cases a reasonable alternative for persistent storage.

Scalable image management

Image management is a highly debated topic in the VM working group. The effort needed to implement an image management solution strongly depends on the type of images supported by BiG Grid. We will focus on the implementation of image management solutions for Class 2 and Class 3 VMs. As explained in section 2.4, site provided VMs (Class 1) for the benefit of system administration are not a goal of this working group. However, when the technical solutions for Class 2 and Class 3 VMs have been developed, Class 1 VMs can be supported without additional effort. In any case, the image management solution should be able to limit access to VM images based on standard grid authentication and authorization.

Image management for Class 3 VMs
Image building for Class 3 VMs is a user responsibility and therefore not part of the image management infrastructure. The image management infrastructure has to be able to accept user supplied VM images from authenticated and authorized users. These images should be made available on the BiG Grid infrastructure for the appropriate user or user community. A point of concern is the storage management: quota must be assigned to user communities, or even individual users, to avoid exhaustion of image storage capacity by a few individuals. Furthermore, the supplied images should be accessible for site administrators and meet the requirements on logging and monitoring.

Nimbus already provides a trust and image management infrastructure required for user supplied VM images. Unfortunately, this infrastructure is missing features like quota management and a scalable solution for image distribution. None of the other products provide solutions for user supplied VM images, although OpenNebula has a development version of an 'Amazon like' solution based on their cloud APIs.

Image management for Class 2 VMs
Several possibilities have been discussed to supply VM images from trustworthy sources.

The most straightforward solution is the inspection and certification of each and every VM image supplied to BiG Grid. An authority should be established, trusted by the sites and the user communities, with sufficient (human) resources to fulfill such a task. No problems are expected in the POC phase, when only few VM images will be deployed. However, it is not clear how the number of supplied images can be maintained at acceptable levels in a production environment. This is most likely doable for the HEP community, assuming they are able to provide standard experiment images. However, the non-HEP communities are expected to supply a much larger diversity of VM images, which would not scale.

As an alternative for certification of individual images, BiG Grid could certify a package repository. A few standard templates can be made available to cover the most popular distributions (CentOS, Ubuntu, etc.). These templates can be extended with different (versions of) packages available in the repository. Once the image is assembled, end-users can only make modifications in user space (this solution would meet the security requirements). The image can be downloaded to a desktop or laptop, or supplied to BiG Grid for deployment on their infrastructure. Although this solution would be acceptable for a significant fraction of the non-HEP communities (if they are able to request new templates), it certainly doesn't cover all of the Use Cases. Furthermore, implementation of an image management and trust infrastructure is not trivial for this scenario. It is *e.g.* not at all obvious who should carry the responsibility for the certified package repository (BiG Grid or a third party).

6.4 Weighted decision matrix

A weighted decision matrix is a tool to make a decision in cases with many available alternatives and many criteria to consider. The most important criteria for the VM management system as discussed in chapter 5 are listed in the weighted decision matrix. A weight is assigned to each criterion as justified below and a scoring definition is introduced to make sure that all alternatives are treated consistently.

	Handling customized images	Handling trusted images	Security features	Developer community interaction	Scheduler integration
Weight	1	1	3	3	2
Nimbus	2	0	2	0	1
OpenNebula	1	0	2	1	1
Grid site products	0	0	1	1	2

	Accounting and	User	Scalable	Image	Total

	Fair Share	interfaces	image distribution	management	
Weight	2	1	2	3	17
Nimbus	1	1	0	1	16
OpenNebula	0	0	1	1	17
Grid site products	2	0	0	0	14

Justification of weights:

- The standard weight is 2.
- Security features and image management are considered basic functionality. Major efforts will be required when (part of) this functionality missing. Hence, these two items contain a weight of 3.
- Due to specific requirements and a tight integration with the BiG Grid infrastructure, significant efforts should be foreseen in handling of customized and trusted images, as well as in development or modification of the user interface. Hence, a reduced weight of 1 is introduced on already present functionality in these areas.
- All products are in early stages of development. Hence, smooth interaction with the developers will be essential and receives additional weight (weight = 3).

Scoring definitions:
0 = Functionality absent or unknown.
1 = Functionality (partially) present or under development, but missing significant features.
2 = Functionality present, only minor modifications might be needed.

The cloud computing products score slightly better in the evaluation than the grid site products. In general terms this is caused by the more ambitious requirements of BiG Grid on Use Case and image handling flexibility. The difference between the cloud computing products is minimal. Nimbus has better handling of customized VM images, a better accounting and fair share system and a better user interfaces. OpenNebula has a scalable image distribution system and a better accessible developer community. The latter is crucial for the plans of BiG Grid, since the working group identified a significant amount of development work.

7. Conclusions and recommendations

The working group concludes that there is a broad interest in virtualized worker nodes from the different user communities in BiG Grid. The Use Cases can roughly be divided in two categories:

1. Class 2 VMs, tightly integrated with grid services (file catalogs, etc.). They perform data intensive operations, so it would be beneficial if these VMs can run within the trusted network fabric of the sites. The VM images are generated by specialists in the Virtual Organizations. Only a small rate of new images is expected, which might make it possible to certify them with a limited amount of human resources.
2. Flexible class 3 VMs. Operating System, library versions, etc. should be in exact accordance with user specifications. This inherently leads to clashes with the site security and operations requirements for the trusted network fabric. Hence, these VMs should be deployed in a DMZ, with appropriate security measures.

Even with appropriate security measures, class 3 VMs will pose additional risks to the BiG Grid infrastructure. Although end-users can be educated and are liable for their actions, negligence during the preparation of VM images can cause a non-negligible increase of exploits. The BiG Grid executive team should decide if they are willing to accept these additional risks associated with Class 3 VMs and the costs associated with mitigating these risks.

A number of external products for the management of VMs have been evaluated. The working group concludes that the two leading open source cloud computing products, OpenNebula and Nimbus, match best with the BiG Grid requirements. This is a direct consequence of their image management flexibility and security features. Existing solutions of other grid sites are less suitable for BiG Grid. These solutions are mainly designed to handle class 2 VM scenarios, but are not aimed at the support of diverse e-science communities like BiG Grid.

The working group did not study licensing issues. Clearly these issues have to be sorted out, especially in the context of Microsoft Windows related products. If possible, liability for license violation should be transferred to the end-users. The BiG Grid executive team has to decide what to do with Microsoft Windows VMs (in fact, with any non-free software) as long as these liability issues are not sorted out. There is no technical limitation to either accept or deny Windows VMs. (Note one technical detail: under Xen these VMs can only be fully virtualized, which has significant impact on file I/O and network I/O performance).

Recommendations

The working group recommends the following actions to the executive team:

1 Design and implement a POC infrastructure based on OpenNebula, this will be part of “phase 2” of the VMs working group activities. If time and human resources permit, an alternative implementation based on Nimbus should be investigated.
   1.1 The initial POC environment should be as simple as possible, with just a few VM images from reliable sources. Submission of VMs should be integrated with

existing grid middleware.

1.2 The POC environment should be deployed to test site security and operational procedures, reliability and scalability.

2 Design and implement a tool to supply VM images, preferably in a joined project of middleware developers and application domain analysts (estimated effort 6-12 person months). The tool should be able to handle several kinds of images (standard image templates, user supplied images, certified images, third party images). Standard grid authentication and authorization should be respected. Quota management is an essential feature in the supply chain of VMs. The tool should be tested in the POC environment.

3 Initiate an effort to investigate, implement and enforce security mechanisms and policies in the POC environment.

4 Initiate an effort to create documentation and training material for BiG Grid user communities on the proper way to create and maintain VM images.

5 Investigate the feasibility of minimalistic 24/7 incident response for the VM infrastructure (and the rest of BiG Grid).

6 Investigate the feasibility to make (a fraction of) the VM infrastructure part of a separate TLD.

7 Investigate the feasibility to transfer the liability for the management of the VM infrastructure to a separate legal entity.

8 Investigate the feasibility to transfer the liability of license violation to the end-user.

Ongoing activities

A number of efforts related to the integration of grids and clouds started already:

- SARA has initiated Claudia [26], a small scale experiment to investigate the use of Cloud Computing in the e-science community. This experiment will provide valuable experience on scalability, especially related to image distribution. The working group suggests to collaborate with this promising project.
- A master student joined the Nikhef Physics Data Processing group to implement the proposed grid middleware plugin for the OpenNebula scheduler (see section 6.3).
- CERNVM is a relatively mature project to provide a baseline Virtual Software Appliance for the LHC experiments. This project might be able provide a base for standardized Class 2 VMs as described in this report.
- A HEPiX virtualization working group has been established to investigate the feasibility of VM deployment on grid infrastructures in a European context. Nikhef will actively participate in this working group. The security concerns raised in this report will probably be addressed in the HEPiX efforts as well. Since this is a HEP based effort we expect less emphasis on flexibility than put forward by the BiG Grid VM working group.

Based on the above list of ongoing activities the design and implementation of a tool to supply VM images (recommendation 2) is expected to be the most critical for successful implementation of virtualized worker nodes on BiG Grid resources. This is the most time consuming and complex sub project foreseen. Furthermore, it will most likely not be covered by any of the known ongoing efforts in the area of cloud and grid integration.

Bibliography

1: Kate Keahey, Nimbus: Open Source IaaS Cloud Computing Software, CERN Workshop, 2009, http://indico.cern.ch/materialDisplay.py?contribId=5&sessionId=4&materialId=slides&confId=56353

2: Ruben Montero, Grids and Cloud Computing: Perspectives and Early Experiences, EGEE, 2009, http://indico.cern.ch/sessionDisplay.py?sessionId=74&slotId=0&confId=55893#2009-09-21

3: Ulrich Swickerath, The batch virtualization project at CERN, EGEE, 2009, http://indico.cern.ch/materialDisplay.py?sessionId=74&materialId=1&confId=55893

4: Davide Salomoni et al., Enabling Distributed Job Submission in Dynamic Virtual Execution Environments for EGEE Users, EGEE, 2009, http://indico.cern.ch/materialDisplay.py?sessionId=74&materialId=0&confId=55893

5: Yves Kemp, Dynamic Cluster Partitioning for Worker Nodes, CERN Workshop, 2009, http://indico.cern.ch/materialDisplay.py?contribId=23&sessionId=3&materialId=slides&confId=56353

6: Nimbus, http://www.nimbusproject.org

7: OpenNebula, http://www.opennebula.org

8: Eucalyptus, http://www.eucalyptus.com

9: VMWare, http://www.vmware.com

10: Enomaly, http://www.enomaly.com

11: AbiCloud, http://www.abiquo.com

12: Ganeti, http://code.google.com/p/ganeti

13: Globus, http://www.globus.org

14: Globus, WSRF, http://www.globus.org/wsrf

15: Amazon, WSDL, http://aws.amazon.com/ec2

16: RESERVOIR, http://www.reservoir-fp7.eu

17: FP7 Program, http://cordis.europe.eu/fp7/home.html

18: EGEE, http://public.eu-egee.org

19: EGEE, EGEE/RESERVOIR Collaboration, EGEE News, 2009, http://www.eu-egee.org/fileadmin/documents/newsletter/summer-2009/EGEE-newsletter-summer-2009.html

20: UCM, http://dsa-research.org/doku.php

21: OCCI, OCCI API, OGF, 2009, http://www.occi-wg.org/doku.php

22: OCA, OpenNebula, http://www.opennebula.org/doku.php?id=documentation:rel1.4:api

23: Ruben Montero, OpenNebula/Eucalyptus comparison, OpenNebula mailing list, 2009, http://lists.opennebula.org/pipermail/users-opennebula.org/2009-July/000551.html

24: Alexei Vazquez et al., Modeling bursts and heavy tails in human dynamics, Phys.Rev.E73 036127, ,

25: Tino Vazquez, Block Device support, OpenNebula, http://dev.opennebula.org/issues/32

26: SARA, Claudia, 2009, https://grid.sara.nl/wiki/index.php/Claudia;_cloud_experiment

27: Tripwire, http://www.tripwire.com

28: WatchGuard, http://www.watchguard.com

29: Cisco FWSM, http://www.cisco.com/en/US/products/hw/modules/ps2706/ps4452/index.html

30: Various, Policies, BiG Grid, http://www.biggrid.nl/big-grid-infrastructure/policies

Appendix A: test plan for a POC environment

Steps 1 and 2 of the test plan below will be performed on a small cluster with 4 dual CPU quad core machines and an image server with 2 TB storage. The testbed will run a Cream-CE and a Torque batch system with a Maui scheduler. Phase 2 of the working group covers all tasks up to the Use Case tests in Step 2. The estimated effort for these tasks is about 12 person months. Some assistance is needed if the working group has to complete these tasks within a period of half a year (the current assignment assumes 3 months).

Already completed tests are in *italic.*

Preparation (Step 1)

Installation tests (CentOS 5 based)

- *Xen kernel*
- *OpenNebula*
- *LVM over iSCSI*

Validation tests

- *OpenNebula submission of Xen VMs*
  - *Paravirtualized machines*
    - *Standard images*
    - *Images on LVM*
    - *Images on LVM, copy-on-write (single and multiple instances)*
  - *Fully virtualized machines on LVM*
    - *Standard images*
    - *Images on LVM*
    - *Images on LVM, copy-on-write (single and multiple instances)*
- OpenNebula submission of KVM VMs
  - Paravirtualized machines
    - Standard images
    - Images on LVM
    - Images on LVM with copy-on-write (single and multiple instances)
  - Fully virtualized machines on LVM
    - Standard images
    - Images on LVM
    - Images on LVM with copy-on-write (single and multiple instances)

Performance tests

- File I/O performance (sequential read/write and random read/write)
  - Ext3
  - Ext3 on LVM
  - Ext3 on LVM with copy-on-write (single and multiple instances)
  - Ext3 on LVM with copy-on-write and device level cache
  - Ext3 on LVM with copy-on-write and device level cache inside VM
    - Paravirtualized machine (Xen and KVM)
    - Fully virtualized machine (Xen and KVM)
- Network I/O performance

- Bridged
  - Paravirtualized (Xen and KVM)
  - Fully virtualized (Xen and KVM)
- NAT – measure CPU overhead
  - Paravirtualized (Xen and KVM)
  - Fully virtualized (Xen and KVM)

Integration (Step 2)

Batch scheduler tests

- Submission of VMs via grid middleware
- Submission of jobs and VMs on a combined system
- Quasi static separation of job and VM hosts

Use Case tests

- Run HEP and Bio analysis Use Cases

Security tests

- Network isolation of VMs
- Run VMs unprivileged, SE Linuxed and chrooted
  - Does this reduce the number of VM exploits?
- Monitor netflows and Honeypots
  - Detection of suspicious network traffic
  - Detection of malicious VMs
- Port scanners
  - Detection of unacceptably open VMs

Operations tests

- Create infrastructure management profiles (Quator, etc.)

Scalability and maintainability (Step 3)

- A dedicated Storage Area Network for image distribution
- Device level caching of VM images
- Fair share scheduling and accounting integration
- Multi-core and advanced scheduling (migration, advanced reservations)
- Separation of DMZ and trusted domain VMs
- Policy implementation for image certification
- Policy implementation for images on the DMZ
- Network isolation of groups of VMs
- User interface for the supply of VM images
- (User interface for the submission of VMs)
- Move from testbed to production system
- Scale up
  - Number of Use Cases
  - Number of users
  - Number of resources

Appendix B: operational and security considerations

It is the responsibility of BiG Grid operations to avoid abuse of resources by unauthorized users and to guarantee access to services, privacy and data integrity for all legitimate users. A number of operational procedures have been implemented on the BiG Grid sites to meet these responsibilities. One of the corner stones in these procedures is the non-repudiation of user actions, *i.e.* site administrators have to be able to tie user actions to individuals (via certificates[6]). Although this allows sites to gain trust in user intentions (because they are able to trace malicious users through the certificate authorities), it does not improve trust in the users' implementations.

Virtualized worker nodes pose additional challenges on the proper implementation of user applications. Hence, BiG Grid operations will only deploy virtualized worker nodes if procedures are in place that do not put the fulfillment of their responsibilities at risk. In this section a number of aspects is discussed related to the additional risk introduced by deployment of virtualized worker nodes and ways to handle these risks.

Vulnerabilities

As shown in Fig. 3, a virtualization layer inevitably introduces additional vulnerabilities (*i.e.* because more software means additional critical bugs), resulting in a so-called increased attack surface. These vulnerabilities can lead to compromised VMs in several ways:

- Online, when the VM is deployed by BiG Grid, through open ports with insecure services.
- Online, but before the VM was deployed on the BiG Grid infrastructure, *e.g.* during preparation.
- By processing of corrupted data, *e.g.* through buffer overflows.
- Introduction of malicious executables in the VM image before the end-user got control over it (*e.g.* because of image exchange between users, or with third parties).

On the other hand, virtualized worker nodes provide the opportunity to run a lightweight host operating system. As an example of the advantages we mention a recent incident with a kernel-vulnerability on non-virtualized worker nodes. In the current environment, this posed a direct threat to the site infrastructures at large. In case of a virtualized environment this would not have been the case. Only VMs with a susceptible kernel could have been removed, without affecting the entire grid infrastructure.

The operations teams have no control over the software installed in VMs provided by users or Virtual Organizations. The lack of this control could result in VMs running with *e.g.* unknown OS patch levels, or with unwanted privileges in a site trusted network[7]. Most of

6. The impact of identity theft has not been considered by the working group. The consequences of identity theft will not significantly differ for systems with or without VMs.

7. The trusted network fabric represents a composition of trusted components, where the trusted components are supporting mandatory access controls, discretionary access controls, audit, identification and authentication of site administration [Trusted Network Interpretation, NCSC-TG-005, NSA].

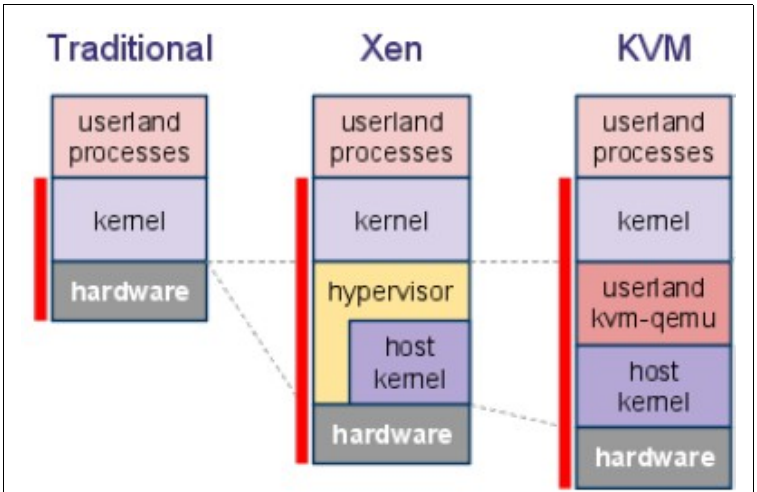


Figure 3: Increasing attack surface

these issues are caused by negligence during the construction of a VM image. On non-virtualized worker nodes, site administrators mitigate risks of these threats with the distinction between user and privileged accounts. Such a distinction is not possible for virtualized worker nodes, unless site administration can trust the privileged account(s) on a VM. In most (if not all) scenarios, this trust relation can not be established.

Once a VM is compromised, it might be used for a number of illegal activities. Examples are: distribution of spam, distributed denial of service attacks (or other kind of attacks against services or hosts on the Internet), hosting of illegal material, copyrighted data or porn, etc. Several breach levels can be distinguished for VMs:

- The VM can be compromised in user space (lowest breach level), or in kernel space.
- The host machine can be compromised in user space (this is known as a break-out), or even in kernel space (highest breach level).

The lowest breach level is already sufficient to do serious damage (all of the illegal activities mentioned above can be performed from this level), but only in the highest breach level VMs of other end-users might be at risk (this risk is reduced when only a single VM runs on a physical host). A severe exploit can directly lead to a break-out in kernel space, *i.e.* there is not necessarily an "escalation path" from lowest breach-level to highest breach-level.

So far, there have only been reports about break-outs with VMWare, none are reported for Xen or KVM. This is not necessarily the result of a difference in intrinsic security, it might just be due to market share. Note that the consequences of a break-out are much more severe than the consequences of a compromised VM. In the former case, the whole site infrastructure should be considered compromised until proven otherwise, leading to serious impact on resource availability for all BiG Grid user communities. In the latter case, counter measures and forensics will be limited to the VM itself (and all possibly associated VMs).

Counter measures

Several counter measures are available to bring the additional risk caused by virtualized worker nodes back to acceptable levels. Some of these measures require a significant effort, either to setup, to maintain or both. Counter measures have been divided in eight different categories to investigate the feasibility and the impact on resources. First these categories will be discussed one by one, then a summary table is shown with the estimated investments per category.

Internal constraints

The VM can be obliged to run an approved distribution, with approved software packages. The user can be denied access to a privileged account inside the VM. Any user software, not available as an approved software package, can be enforced to run in user-space. Before deployment, the VM image should be inspected and certified by a BiG Grid accepted authority. In this scenario operations delegates the responsibility for the image content to the certifying authority.

External constraints

The privileges of the VM manager account (the VM process runs under this account in the host machine) can be reduced to an absolute minimum. It might be possible to run VMs from an unprivileged account in the host machine, although especially the impact on network and file I/O performance is unclear. Several tools are available (chroot, SELinux), to further increase the barrier between VMs and the host operating system.

Black-box

The VM can be encapsulated, white-listing only minimal outbound IP connectivity of known services on selected ports. Inbound connectivity should initially be blocked completely. In a later stage, more dynamic firewall configurations to selectively open inbound ports might be possible[8]. Network Address Translation (NAT), a firewall (IP tables) and bridging tables (ebtables) can be installed on host level. The VMs can be deployed on statically assigned resources in a demilitarized zone (DMZ), with no access to grid services other than offered to any machine on the public Internet. This will only be an acceptable solution for a fraction of the Use Cases.

Central logging

Central logging is a first requirement to monitor the behavior of VMs. Nevertheless, logs collected from VMs do not necessarily contain correct information, so logging does not establish a trust relation between site administration and VMs.

Intrusion detection

The software stack on the hosts will be relatively simple, only a few well defined processes will be running. Any deviation from standard operation can be detected and trigger an alarm. Tools like Tripwire [27] can be installed on the host to monitor the integrity of the system and to provide an intrusion detection mechanism.

8. Dynamic configuration of a firewall is non-trivial and requires additional elements in the user-interface of the VM management system.

Network monitoring

Network traffic inspection (netflows, honeypots, etc.) allows identification of malicious content. The host network can be isolated from the VM network in a separate VLAN. Traffic inspection should be done for both the host and the VM network. Proprietary devices like WatchGuard [28], or Cisco FWSM [29], can perform traffic pattern analysis or even deep packet inspection to detect suspicious behavior. Extensive package inspection could also be integrated in a more affordable way with open source solutions, combined with host level NAT. The disadvantage of host level NAT is the significant (maybe even unacceptable) CPU overhead in case the VM runs network intensive applications.

Education

BiG Grid should offer education for end-users and managers of Virtual Organizations on subjects like: risks of exchanging data, risks of running VMs in an insecure environment, the importance of security patches, etc. The education can be mandatory, or more lightweight in the form of a brochure with a list of "do's and dont's" on security.

Liability

End-users and managers of Virtual Organizations should be aware of their liability in case of negligence. The latter can clearly be stated in the end-user license agreement (EULA), which has to be signed before BiG Grid resources can be accessed. Composing a legally binding EULA to formalize the responsibilities toward VM security should be left to the appropriate lawyers.

The measures presented above have different benefits. Applying internal and external constraints, and running VMs in a black box aim to reduce the vulnerabilities and actively contribute to site security. Central logging, intrusion detection and network monitoring enable the detection of abnormal usage patterns, but have no preventive value; they facilitate a timely and proper response by the administration teams and may help in restricting damage. User education and liability may raise awareness with the users, although the effect of such measures is not guaranteed.

Investments (BiG Grid total)

Counter measure category	*Human resources (setup)*	*Human resources (maintenance)*	*Other (setup)*	*Other (maintenance)*
Internal constraints	1 month	3 days/image	0	0
External constraints	Negligible	Negligible	0	0
Black-box	Negligible	Negligible	0	0
Central logging	1-4 weeks per year	Negligible	0	0
Intrusion detection	Negligible	1-4 weeks per year	0	0
Network monitoring	Proprietary: Days Open Source: 2 Months	A few days per year 2 days per month	O(? kEuro) 0	O(10kEuro) 0

Education*	1-2 weeks	1-2 weeks per year	O(100 Euro)	0
Liability*	1-3 months	A few days per year	O(kEuro)	0

Forensics

In case of incidents, forensics is required to investigate legal consequences, to find the vulnerability and the size of the incident and to make sure that vulnerabilities are fixed before the systems come back on line.

Forensics can be facilitated when “start-states” of VMs are stored for a limited period before they are disposed. The number of start-states is relatively small, because multiple VMs will be booted from the same VM image. As a result, only limited investments in human resources (a few weeks to setup and a few days per year to maintain) and hardware (estimated in the order of a few kEuro) are required to store start-states for a few days, up till a week. Even better, from forensics point of view, is the storage of all VM end-states. However, the working group concluded that a significant investment is required to provide sufficient storage capacity. The limited benefits compared to the storage of start-states do not justify such an investment. Another possibility to be investigated, is tracking of differences between start-states and end-states. This could well be a feasible mechanism to implement with limited storage capacity, especially if copy-on-write techniques are applied for VM deployment.

Forensics on a VM is a complicated and time consuming process, especially when the operations team does not have control over the internals of the VMs. It might be difficult to get access to the VM (*e.g.* when the image has been deleted when the job expired). On top of that, even the most basic information and applications inside the VM can not be trusted. Even though the BiG Grid sites have a Security Incident Response Team (SIRT), an incident could lead to multiple days or even weeks of unexpected manpower investments and downtime of (part of) the infrastructure. Note that most SIRT teams only provide office hour support, which might be insufficient for some of the virtualized worker node scenarios. Minimalistic 24/7 support along the following lines could be investigated: in case monitoring tools discover irregularities outside office hours, on-call support employees will receive an automated message. They can decide to manually[9] interrupt or block the misbehaving services until the next working day. Note that 24/7 support is a matter between the BiG Grid operational partners and their employees. As a rule of thumb, the vast majority of the security efforts should be focused on prevention rather than on forensics.

9. Automatic interruption is undesirable, it could lead to unnecessary interruptions or unexpected changes in *e.g.* switch or router configurations.

Appendix C: Policies

One of the key ingredients for successful deployment of virtualized worker nodes on the BiG Grid infrastructure is the development of a set of policies (defining responsibilities and possibilities) and procedures for deployment of virtualized worker nodes. These policies and procedures should address the concerns raised by BiG Grid operations side, without hampering the functionality requested by the BiG Grid user communities. In this chapter, existing policies are discussed that are (or might be) relevant for an infrastructure with virtualized worker nodes.

Existing policies

Some of the BiG Grid sites participate in a number of national and international grid computing collaborations. Here, the relevant sections of the policy documents of these collaborations are listed, as well as the relevant parts of our own BiG Grid policies. Note that besides this list, site local policies and practices may apply as well.

Applicable policies [30]

- Grid Security Policy, version 5.7a
- VO Portal Policy, version 1.0 (draft)
- BiG Grid Security Policy, version 2009-025
  - Grid Acceptable Use Policy, version 3.1
  - Grid Site Operations Policy, version 1.4a
  - LCG/EGEE Incident Handling and Response Guide, version 2.1
  - Grid Security Traceability and Logging Policy, version 2.0

No conflicts between these policies and the deployment of virtualized worker nodes have been identified. In fact, some policies may even serve as guidelines for a new policy that helps in making virtualized worker nodes an acceptable extension of the BiG Grid infrastructure.

First of all, “users may be held responsible for all actions taken with their credentials”. So, they carry the responsibility to deploy trustworthy VMs. However, “the provisioning of resources to the Grid is at your own risk”, which indicates that sites have to accept the risk of incidents caused by running these VMs. Two handles are provided for sites to reduce the risk associated with the deployment of VMs:

Logging — “In order to satisfy traceability requirements, software deployed in the Grid must include the ability to produce sufficient and relevant logging, and to collect logs centrally at a site.”. Sites can therefore require specific logging functionality to be implemented in the VMs, which enables them to monitor proper operation of the VM. As discussed, it does not enable them to trust the VM. Defining requirements for logging by VMs may be feasible for images produced within BiG Grid, but that will not

be the case for images provided by other users and VOs.

Networking — "All the requirements for the networking security of resources are expected to be adequately covered by each site's local security policies and practices.". Hence, every site can decide to limit network connectivity for virtualized worker nodes up to the level it deems necessary from its security perspective. This is not necessarily the same for all BiG Grid sites.

An interesting policy, that could serve as an example for a virtualized worker node policy, is the "VO-Box Security Recommendations and Questionnaire", version 0.6 (draft, not ratified). A VO-Box is in part similar to virtualized worker nodes because it contains experiment specific software outside the control of BiG Grid operations. The policy states: "The VO Box is part of the trusted network fabric of the site. […] therefore privileged (root) access to the system must be limited to the Resource Administrators.".

One of the of the crucial decisions to make in the design of a virtualized worker node infrastructure is: are virtualized worker nodes going to be part of the trusted network fabric of the site?

- If the answer to this question is yes, then the virtualized worker node policy can be very similar to the VO-box policy, so end-users will not have privileged access to virtualized worker nodes after the image has been certified.
- If the answer is no, then network policies and practices have to be investigated for the network segment with virtualized worker nodes. In that case, a virtualized worker node policy will probably be very similar to policies of so-called "guest networks" or Internet Service Providers. Note that even though the VM is not part of the trusted network, it can still do damage to site operations.

Hybrid solutions, with part of the resources in the trusted network and part of the resources in a DMZ (either statically or dynamically assigned), are possible but more complicated to implement and maintain. Whatever the policies for virtualized worker nodes will be, it is crucial that these policies and practices are approved by the OST. Networking is covered by site local policies, so only BiG Grid wide support from operations will allow for cross-site solutions.