PROCEEDINGS OF SCIENCE

Resource Provisioning through Cloud and Grid Interfaces by means of the Standard CREAM CE and the WNoDeS Cloud Solution

Elisabetta Ronchieri[*†]
INFN CNAF
E-mail: elisabetta.ronchieri@cnaf.infn.it

Giacinto Donvito
INFN, Sezione di Bari
E-mail: giacinto.donvito@ba.infn.it

Paolo Veronesi
INFN CNAF
E-mail: paolo.veronesi@cnaf.infn.it

Davide Salomoni
INFN CNAF
E-mail: davide.salomoni@cnaf.infn.it

Alessandro Italiano
INFN CNAF
E-mail: alessandro.italiano@cnaf.infn.it

Gianni Dalla Torre
INFN CNAF
E-mail: gianni.dallatorre@cnaf.infn.it

Daniele Andreotti
INFN CNAF
E-mail: daniele.andreotti@cnaf.infn.it

Alessandro Paolini
INFN CNAF
E-mail: alessandro.paolini@cnaf.infn.it

PoS(EGICF12-EMITC2)124

Resource provisioning using both cloud and grid interfaces offers flexible solutions for existing infrastructure providers and end users alike. In this paper, we will present the work done to build a cloud infrastructure on top of an EGI grid farm by means of the standard CREAM CE and of the WNoDeS cloud solution. The ultimate goal of this work is to provide the end user with a rich set of interfaces to access computing resources, such as standard grid job submission, grid job submission to specific virtual machine images, a Web interface for self-allocating virtual machines, and interactive provisioning and usage of virtual images. The solution is based on open source software like CREAM CE, WNoDeS, Torque/Maui and Lustre. It is intended to fulfill the computing requirements of several and largely different community of researchers, starting from HEP and other communities that have been maturing valuable experiences in grid computing for many years, but providing also simple and user friend interfaces for users or communities that do not have the needed expertise to exploit complex grid infrastructures.

EGI Community Forum 2012 / EMI Second Technical Conference,
26-30 March, 2012
Munich, Germany

[*]Speaker.

[†]Corresponding author.

1. Introduction

Typically, large computing centers used for a wide range of applications by multiple user communities need to define proper and timely configuration of compute resources without sacrificing efficiency, flexibility and security. User communities also need to be allowed access to resources using a number of different interfaces, authenticating via several methods, exploiting existing local and distributed infrastructures. Therefore, a multi-layer framework requires to be defined in order to optimize the usage of computing center resources for multiple communities.

WNoDeS[1] is a software framework to integrate grid and cloud provisioning through virtualization that has been defined and developed starting from the needs of the INFN CNAF computing center (located in Bologna, Italy). This center currently hosts about 10,000 computing cores, 9 PB of disk space and 10 PB of tape space; about 80,000 jobs are run each day at CNAF, which supports several international astro-particle physics experiments (e.g., AMS2, Argo, Auger, Fermi/Glast, Magic, Pamela, Virgo). The center is the Italian Tier-1 for high energy physics CERN-based LHC experiments (ATLAS, CMS, LHCb, Alice), and is Tier-0 or Tier-1 for several others. CNAF adopts Platform LSF as batch system and IBM GPFS as shared File System. WNoDeS has proved to be scalable and reliable: it is in production at several Italian centers since November 2009. Currently WNoDeS is managing at CNAF about 2000 on-demand Virtual Machines (VMs). Since last year, WNoDeS is a joint effort between INFN and IGI[2]; in December 2011, WNoDeS was accepted in EMI and will be part of the upcoming EMI-2 release.

In this paper the installation and configuration of WNoDeS over a standard EGI grid farm where a CREAM CE is running will be shown. This activity starts with a farm that has been already set up by using standard EMI software releases and YAIM procedures. All the changes needed in supporting cloud functionalities can be executed without any service interruption or disruption. The described solution exploits specific functionalities that are part of a standard Torque and Maui installation with a few simple customizations. From the point of view of the system administrator all computing requests, for both real and virtual resources, are always managed by the batch system, and typical configurations are needed in order to deal with resource sharing, priorities and limits about resource usage. Some details of the Web application functionalities will be shown that are offered from the Web interface installed on the testbed. This integrates the Open Cloud Computing Interface (OCCI)[3] that is used to instantiate virtual images; the options available to the end user will also be described.

This is the first time that WNoDeS is used in a production environment using Torque/Maui as a batch system. Information on how users can exploit such an infrastructure by means of the standard interfaces provided by the EGI grid, like the WMS-based or direct CE job submission, will be shown; this will be a first example of usage of a federation of cloud infrastructures.

The rest of the paper is organized as follows. Section 2 details some of the WNoDeS features that are part of its workplan, while Section 3 describes the WNoDeS architecture. Section 4 shows the WNoDeS testbed built on top of the Torque/Maui batch system. Section 5 lists the WNoDeS capabilities. Finally, Section 6 draws some conclusions.

[1] WNoDeS, `http://web.infn.it/wnodes/`

[2] Italian Grid Infrastructure, `http://www.italiangrid.it/`

[3] Open Cloud Computing Interface (OCCI), `http://occi-wg.org/`

2. WNoDeS Features

In this paper the most important features of WNoDeS are described [1].

Batch Job Execution - Local batch jobs can be run on both real and virtual execution hosts (Worker Nodes). A virtual Worker Node is instantiated on-demand to provide computing resources for the execution of batch jobs just when needed. WNoDeS allows customized execution environments based on virtual machines, built and instantiated to match user requirements. This feature will be delivered in EMI-2.

Grid Integration - WNoDeS provides a seamless integration with Grid Computing EMI gLite middleware for transparent Grid job execution on Virtual Machines. Dynamic selection of resources (virtual machine types, parameters) is done through standard EMI gLite Grid submission tools. Resource selection is based on a Glue 1.x schema attribute (SoftwareRunTimeEnvironment). This interoperates with current gLite WMS and CREAM CE middleware. Authentication is based on VOMS (Virtual Organization Membership Service). This feature will be delivered in EMI-2.

Mixed Mode - WNoDeS mixed mode is a configuration mode that lets WNoDeS to send batch jobs on a physical system and, at the same time, to instantiate virtual machines (VMs) on the same system. These VMs can be used to also run batch jobs or to support Cloud services. Table 1 sums up the pros and cons of enabling the mixed mode feature. This feature will be delivered in EMI-2.

Pros	**Cons**
Increase the flexibility of the configuration of a farm	In case of LSF, increase the number of LSF licenses needed
Allow system administrators to progressively integrate WNoDeS into existing farms without first having to decide which nodes will support virtualization and which not	Add one more configuration step to execute standard real job in the system where WNoDeS is installed
Optimize resource usage	Put hypervisors in public address space
Add support e.g. for Cloud computing, interactive usage on custom VMs etc. in a traditional farm	
Direct jobs to VMs or to real hardware using LRMS policies	
differentiate real vs. virtual requests/jobs e.g. based on queues, users, requirements, Grid VOs, etc	
Allow jobs that require full performance to be executed on physical systems	

Allow jobs that require hardware that is not easily virtualized to be executed on physical systems	

Table 1: WNoDeS Mixed Mode, Pros and Cons

Cloud Computing - WNoDeS has a Cloud Computing interface based on the OCCI API. This API is either accessible via CLI, or via a Web-based Application. This provides on-demand computing resources allocated to users out of a common pool. Authentication is internally based on Short-Lived X.509 Certificates obtained through an online CA. However, users can authenticate through several methods, like Kerberos, Shibboleth, X.509/VOMS or Username/Password.

Virtual Interfactive Pool - WNoDeS VIP [2] is an on-demand provisioning of virtual user interfaces for local users out of a common resource pool. Based on a CLI, it can be easily used by computing center users seeking for additional, unused resources. Provides customized computing resources for interactive usage like, for example, software development or physics analysis. The provisioned virtual resource matches user requirements for RAM, CPU, bandwidth and shared file systems to be mounted. A caching mechanism is available, to reduce system provisioning time.

3. WNoDeS Architecture

The WNoDeS architecture is composed of several components that are shown in Figure 1. Looking at Figure 1 from top to bottom:

Web/CLI are the WNoDeS OCCI-compliant interfaces [3] that end users can use to request cloud computing resources. Figure 2 and Figure 3 show two screen shots of the WNoDeS Web application related to the creation of a virtual machine and to the status of the virtual machines that a user requested.

Cache Manager is responsible for pre-allocating resources in order to reduce the latency between the request from the users and the start of the virtual machine. It guarantees that there are at least some virtual machines available to the users, re-allocating or destroying the machines according to the dynamic need of the users. It handles cloud requests coming from both the CLI and the Web Application.

Site-Specific can be considered as the site configuration resolver. It is the component that sends the resource request to the `wnodes_bait` service and checks the job status;

Bait is the host resource manager responsible for verifying that there are resources to execute both real jobs and virtual jobs, requiring the instantiation of virtual machines when necessary, and executing the job on the allocated resource;

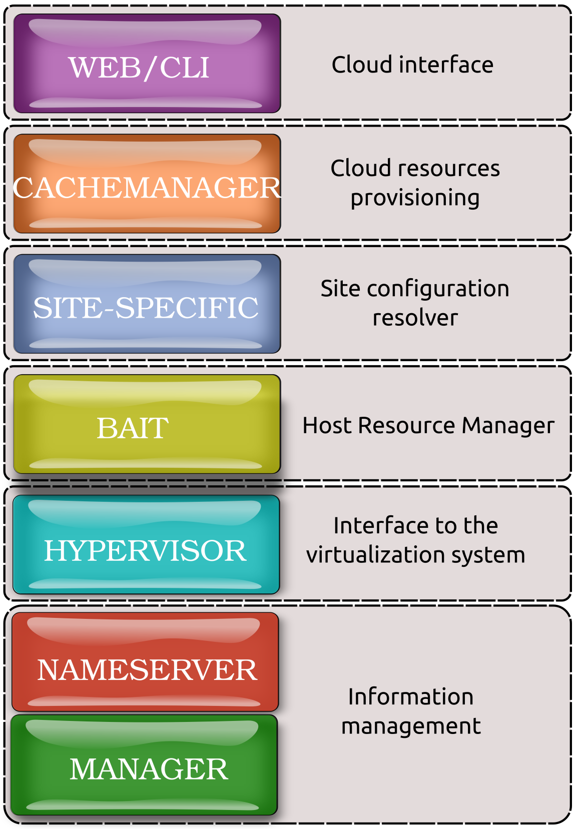


Figure 1: WNoDeS Full Stack. All the components shown will be included in EMI-2 with the exclusion of the Web interface, the CLI and the Cache Manager; these will be provided in further EMI updates.

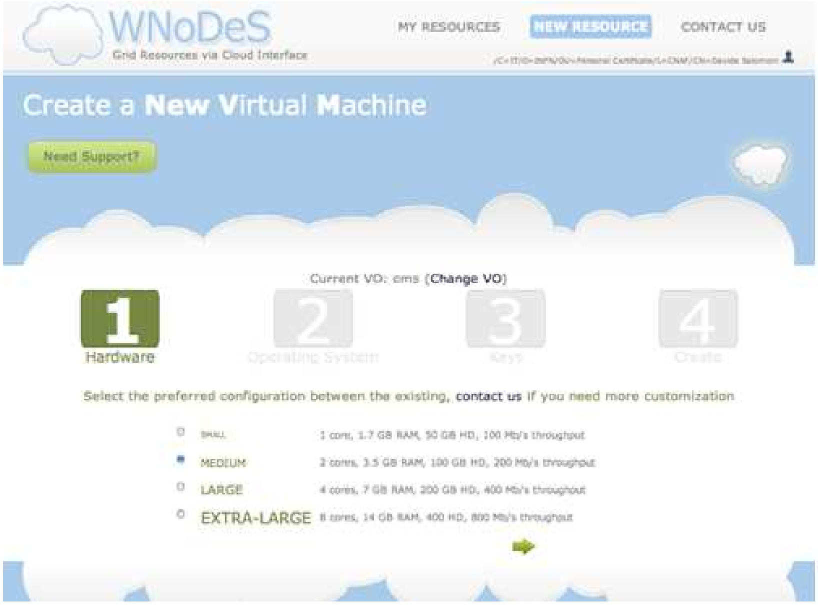
Hypervisor is mainly the interface to the virtualization system responsible for instantiating virtual machines where a virtual job will be executed. However, if the mixed mode feature is enabled, it is also able to run real jobs;

Nameserver, Manager is the WNoDeS information management component. The nameserver is a catalogue responsible for keeping trace of all the virtual machines currently running on each hypervisor, and of all the virtual machine images stored in the configured repository. The `Manager` is a Command Line Interface (CLI) that is responsible for the configuration of the repository of the virtual machine images. It provides a set of options to handle images, VLANs, hostnames, bait and hypervisor configuration files. Furthermore, it supports a set of options to manage bait and hypervisor status.

Figure 4 shows the interactions among all the WNoDeS components to handle a request of cloud resource provisioning.

4. WNoDeS Testbed

In the following it is described how the presented solution is deployed at INFN-CNAF and how it can be used by an end user.


WNoDeS
Grid Resources via Cloud Interface
MY RESOURCES
NEW RESOURCE
CONTACT US
/C=IT/O=INFN/OU=Personal Certificate/L=CNAF/CN=Davide Salomoni
Create a New Virtual Machine
Need Support?
1 Hardware
2 Operating System
3 Keys
4 Create

Current VO: cms (Change VO)

Select the preferred configuration between the existing, contact us if you need more customization

○ SMALL 1 core, 1.7 GB RAM, 50 GB HD, 100 Mb/s throughput
● MEDIUM 2 cores, 3.5 GB RAM, 100 GB HD, 200 Mb/s throughput
○ LARGE 4 cores, 7 GB RAM, 200 GB HD, 400 Mb/s throughput
○ EXTRA-LARGE 8 cores, 14 GB RAM, 400 HD, 800 Mb/s throughput

Figure 2: Create a virtual machine.

4.1 Deployment

The testbed is characterized by 1 shared CREAM-CE; the Torque and MAUI batch scheduler; 1 nameserver machine (i.e., a WNoDeS dedicated service) that hosts the WNoDeS manager CLI; 1 cache manager host; 1 Web application host; 1 hypervisor machine that could host up to 24 concurrent virtual nodes; 2 virtual machine images with Scientific Linux SL release 5.7 (Boron) as Operating System, 1 Core per machine, and 1.5 GB of memory. Different images, e.g. one per each VO/user are possible. In our case, each image is shared across all VOs (i.e., DTEAM, enmr.org). Typically, the creation of customized images needs to be agreed upon with a resource center.

4.2 Use Cases

Functional and stress tests have been performed daily on the testbed for several months. Functional tests are executed considering local account, grid and cloud submission, while stress tests consisted of submissions of 2000 jobs in one shot. All of the tests were completed without errors.

Four use cases are shown below.

4.2.1 Running jobs on a VM through direct CE submission

There are two prerequisites:

1. users must belong to a VO (e.g., DTEAM)
2. users must have a x.509 digital certificate

Once created the executable script and the jdl file, glite command lines need to be run to submit user's jobs directly to a CE [4].

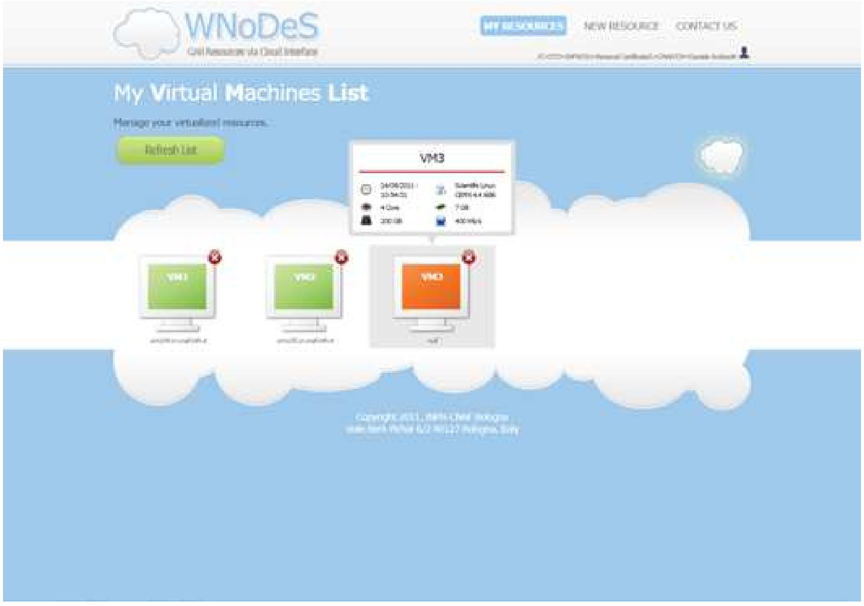


Figure 3: Get the status of the virtual machines.

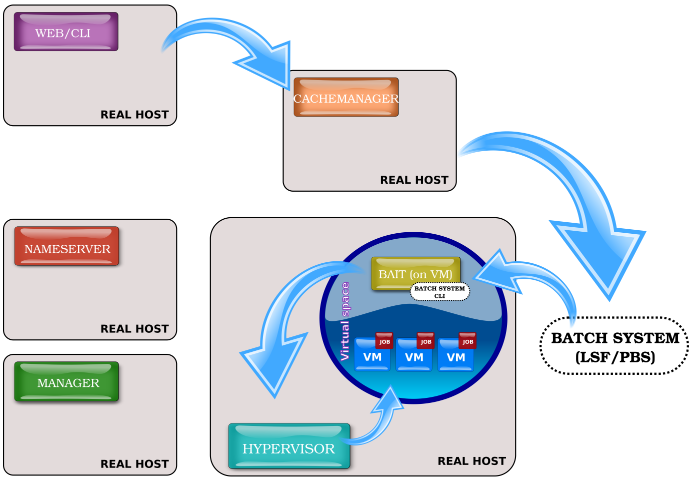


Figure 4: Cloud resource provisioning by using WNoDeS.

4.2.2 Running jobs on a VM through the use of a WMS

The prerequisites have already listed before. Once created the executable script and the jdl file, glite command lines need to be run to submit user's jobs to a WMS [4].

4.2.3 Using the Cloud CLI to get a VM

The prerequisites have already listed before. Then to submit user's request to instantiate a virtual machine:

- set the Cloud CLI configuration file:

```
[server]
endpoint = test-wnodes-web01.cnaf.infn.it
port = 8443
[infrastructure]
category = compute
[content]
type = text/plain
[nameserver]
ns_host = wn-104-03-01-01-a.cr.cnaf.infn.it
ns_port = 8219
[user]
user_key = /home/joda/.globus/userkey.pem
user_cert = /home/joda/.globus/usercert.pem
pub_key = /home/joda/.ssh/id_rsa.pub
vo = dteam
ca = /home/joda/ssl/INFNCA.crt
```

- run the following commands in order to require the instantiation of an existing image in the WNoDeS repository and to get the state of the submitted request:

```
> wnodes_list_images_tags
...
Name            arch
vwn_sl5_emi     x86_64
wn_sl53_T3      x86_64
wnodes_sl5_bait x86_64
img_VIP_sl53    x86_64
> wnodes-create-image -t  img_VIP_sl53
...
https://test-wnodes-web01.cnaf.infn.it:8443/resource/compute/
1315b95b-fcee-46c6-bcdd-04bd883a3686
> wnodes_list_image -l
https://test-wnodes-web01.cnaf.infn.it:8443/resource/compute/
1315b95b-fcee-46c6-bcdd-04bd883a3686
...
hostname  status      arch        memory  cores  speed
null            PENDING  x86_64  1.7          1        0.0
> wnodes-list-images
...
https://test-wnodes-web01.cnaf.infn.it:8443/resource/compute/
1315b95b-fcee-46c6-bcdd-04bd883a3686
```

- once the VM status is ACTIVE, connect to the VM via ssh

- destroy the VM image when finished by running the following command:

```
> wnodes_delete_image -l
https://test-wnodes-web01.cnaf.infn.it:8443/resource/compute/
1315b95b-fcee-46c6-bcdd-04bd883a3686
```

4.2.4 Using the Cloud Web application to get a VM

There are two prerequisites:

1. users must belong to a given VO (e.g., DTEAM)
2. users must have a x.509 digital certificate installed in their browser.

Then to submit user's request to instantiate a virtual machine through the Cloud Web application the following steps need to be performed:

- select the VO the user belongs to. The Web application will validate the user authentication by using VOMS
- select a computing resource, specifying such as number of CPU cores, RAM size, and Operating System (see Figure 2)
- submit the VM allocation request; the VM will be subsequently accessed via passwordless ssh
- check the VM status whenever necessary (see Figure 3).

5. WNoDeS Capabilities

The WNoDeS capabilities refer to infrastructure monitoring, resources usage accounting, publishing of virtual machines information, and VM image policies. Their implementations do not change in relation to the adoption of grid and cloud interfaces with the exclusion of monitoring, for which there are two distinct NAGIOS probes to consider in order to monitor the WNoDeS availability: the CE CREAM probe that is already integrated in Grid sites and can be used when Grid jobs are executed using the current CE; the OCCI probe that will be developed within the EGI Federated Cloud Task Force[4] and will be used when cloud jobs are executed using the OCCI interface.

For what regards accounting, users, both using grid and cloud interfaces, can be authenticated by using X.509 certificates and validating their inclusion into a given VO. Jobs and cloud instantiations running on WNoDeS resources will be accounted like standard resource requests mediated by a batch system.

For regards publishing of WNoDeS resources, both grid and cloud interfaces are published by the CREAM resource BDII by specifying a dedicated queue. This queue can be characterized by parameters like available virtual images, supported VOs, max wall time and so on.

[4] EGI Federated Cloud Task Force, http://www.egi.eu/news-and-media/newsfeed/news_0078_cloud_taskforce.html

For what regards virtual image policies, WNoDeS currently supports a scenario where virtual machine images requested by users or Virtual Organizations should be agreed with a local site / resource provider. Currently, this type of policy is required for security reasons. Further WNoDeS releases will provide features able to relax this requirement.

The WNoDeS cloud solution relies very much on the capabilities of the batch system used natively in the farm in order to obtain good performance, scalability and reliability while scheduling requests for virtual machine on available nodes. Indeed, for each of the supported batch scheduler (Torque/MAUI and LSF) WNoDeS is exploiting different capabilities, for example: the capability of moving job among nodes in the case of LSF, while the procedure foresees that the job is executed on the bait machine and an ad-hoc pre-execution script make it fails if the job should be executed on a virtual machine in the case of Torque. The script itself makes the virtual machine start and executes the job on the started machine.

6. Conclusions

Resource provisioning by means of CREAM CE's and WNoDeS can provide site administrators and end users with several benefits. Site administrators can use the same set of services (such as the batch system, CREAM CE, NAGIOS and so on) to manage the same pool of resources that are allocated dynamically to satisfy user requests. Resource centers can provide their users with a cloud infrastructure that is built with simple and non-invasive changes in their production computing infrastructure. Conversely, end users can exploit grid and cloud interfaces based on their applications and requirements, and enjoy potentially considerably more available resources through easy and intuitive user interfaces. Each end user could ask a site / resource providers to offer a dedicated virtual host image able to fulfill his/her own specific requirements in terms of memory, installed packages and services. End users could also exploit interactive usage of one or more self-provisioned custom virtual machines.

In addition, thank to WNoDeS mixed mode, a farm is not required to be fully converted to WNoDeS, which can co-exist and share resources with a traditional computing cluster that does not use virtual machines. All the supplied resources could be used in order to easily deploy specific services that could not be offered using the standard tools provided by the EGI grid infrastructure.

In the future, WNoDeS upgrades, included in EMI releases, will address features like OCCI integration and local virtual machine provisioning (Virtual Interactive Pools, or VIP).

References

[1] Davide Salomoni, Alessandro Italiano and Elisabetta Ronchieri, "WNoDeS, a Tool for Integrated Grid/Cloud Aaccess and Computing Farm Virtualization," CHEP 2010, Taipei, 2011 J. Phys.: Conf. Ser. 331 052017.

[2] Claudio Grandi, Alessandro Italiano, Davide Salomoni, and Anna Karen Calabrese Melcarne, "Virtual Pools for Interactive Analysis and Software Development through an Integrated Cloud Environment," CHEP 2010, 2011 J. Phys.: Conf. Ser. 331 072017.

[3] Davide Salomoni, Daniele Andreotti, Luca Cestari, Guido Potena, and Peter Solagna, "A Web-based portal to access and manage WNoDeS Virtualized Cloud resources," in PoS(ISGC 2011 & OGF 31) 054.

[4] Job Submission Tutorial, refhttp://iag.iucc.ac.il/workshop/job_submit.htm