A Web-based Portal to Access and Manage WNoDeS Virtualized Cloud Resources

Davide Salomoni\textsuperscript{1}, Daniele Andreotti\textsuperscript{1}, Luca Cestari\textsuperscript{2}, Guido Potena\textsuperscript{2}, Peter Solagna\textsuperscript{3}

\textsuperscript{1}INFN-CNAF, Bologna, Italy
\textsuperscript{2}University of Ferrara, Italy
\textsuperscript{3}INFN-Padova, Italy (now at EGI.eu, Amsterdam)

Outline

- Introduction to WNoDeS
- WNoDeS Cloud Provisioning
- Conclusions
Introduction to WNoDeS

- The INFN WNoDeS (Worker Nodes on Demand Service) is a virtualization architecture targeted at Grid/Cloud integration
  - Providing transparent user interfaces for Grid, Cloud and local access to resources
  - Re-using several existing and proven software components, e.g. Grid AuthN/AuthZ, KVM-based virtualization, local workflows, data center schedulers
  - See http://web.infn.it/wnodes for details
- In production at the INFN Tier-1, Bologna, Italy since November 2009
  - Several million production jobs processed by WNoDeS (including those submitted by experiments running at the LHC)
  - Currently, about 2,000 dynamically created VMs
  - Integration with the INFN Tier-1 storage system (8 PB of disk, 10 PB of tape storage)
  - Also running at an Italian WLCG Tier-2 site, with other sites considering its adoption
Key WNoDeS Characteristics

- Uses Linux KVM to virtualize resources *on-demand*; the resources are available and customized for:
  - direct job submissions by local users
  - Grid job submissions (with direct support for the EMI CREAM-CE and WMS components)
  - instantiation of Cloud resources
  - instantiation of Virtual Interactive Pools (VIP)
    - See e.g. the WNoDeS talk on VIP at CHEP 2010, October 2010
- VM scheduling is handled by a LRMS (a “batch system software”)
  - No need to develop special (and possibly unscalable, inefficient) resource brokering systems
  - The LRMS is totally invisible to users for e.g. Cloud instantiations
- No concept of “Cloud over Grid” or “Grid over Cloud”
  - WNoDeS simply uses *all* resources and dynamically presents them to users as users want to see and access them
- At this conference, see also:
  - Grids and Clouds Integration and Interoperability: an Overview
  - Performance Improvements in a Large-Scale Virtualization System
WNoDeS Release Schedule

- WNoDeS 1 released in May 2010
- WNoDeS 2 “Harvest” public release scheduled for September 2011
  - More flexibility in VLAN usage - supports VLAN confinement to certain hypervisors only
    - Used at CNAF to implement a “Tier-3” infrastructure alongside the main Tier-1
  - `libvirt` now used to manage and monitor VMs
    - Either locally or via a Web app
  - Improved handling of VM images
    - Automatic purge of “old” VM images on hypervisors
    - Image tagging now supported
    - Download of VM images to hypervisors via either `http` or Posix I/O
  - Hooks for porting WNoDeS to LRMS other than Platform LSF
  - Internal changes
    - Improved handling of Cloud resources
    - New plug-in architecture
  - Performance, management and usability improvements
    - Direct support for LVM partitioning, significant performance increase with local I/O
    - Support for local `sshfs` or `nfs` gateways to a large distributed file system
    - New web application for Cloud provisioning and monitoring, improved command line tools
Outline

- Introduction to WNoDeS
- WNoDeS Cloud Provisioning
- Conclusions
VM Provisioning

- **WNoDeS per-se allows full customization of VMs**
  - I.e., the parameters to define VMs are all available to the system
  - *Realistically*, though, at the INFN Tier-1 we have decided to characterize and provision VMs according to a fixed set of parameters
    - This should answer most if not all the practical requests of our users, while at the same time limiting entropy
    - The *billing model* is set-up accordingly
      - Since WNoDeS is built on top of a LRMS (batch system) for the provisioning of resources, accounting and billing are typically based on the LRMS itself

- **There is more to this**
  - Definition of “custom images”
    - Through modification of pre-defined VM image sandboxes and subsequent saving and retrieval
  - Virtualized Storage: “dropbox-like” (easy; testing of virtual containers on top of GPFS ongoing at the INFN Tier-1), QoS-constrained (less easy)
  - Support for these features will likely be in WNoDeS 2+x (i.e. not in the next release)
VM Characterization

For Cloud requests, we define the following instance types:

- **Small**: 1 core, 1.7 GB RAM, 50 GB HD
- **Medium**: 2 cores, 3.5 GB RAM, 100 GB HD
- **Large**: 4 core, 7 GB RAM, 200 GB HD
- **Extra-large**: 8 cores, 14 GB RAM, 400 GB HD

Current Grid VM images normally fall into the “Small” instance type

Two further options are foreseen, initially for Grid and VIP resources:

- **Whole-node, hard**: all hardware cores, \((1.7 \times \text{num. cores})\) GB RAM, \((50 \times \text{num. cores})\) GB HD
- **Whole-node, soft**: all available cores \((\text{with a minimum})\), \((1.7 \times \text{num. cores})\) GB RAM, \((50 \times \text{num. cores})\) GB HD

Note: network and virtual distributed storage not considered above

The instance type is then applied to a chosen Operating System

- Linux only, for the time being (preliminary testing of instances running Microsoft Server 2008 w/ HPC pack ongoing)
Cloud Provisioning

- The hard way: API-based
  - WNoDeS supports and implements the OGF OCCI (Open Computing Cloud Interface)
  - Not really meant for direct human consumption and essentially not directly used, at least at the INFN Tier-1

- More practically, via a web-based application
  - We don’t need yet another “submission portal”, though
  - Want to converge around the general concept of “resource allocation & utilization”
    - Grid, Cloud, or else. In other words, a single Grid or Cloud submission/allocation portal
    - With integrated authentication, authorization, accounting
      - We strongly believe in component re-use. For example, for what regards AuthN and AuthZ, we (internally) use VOMS and the gLite Argus Authorization Service
Process Flow for the WNoDeS Web-Based Cloud Management

1. Submit
2. Client-side validation
3. Create Empty Compute
4. Retrieve and manage parameters
5. Server-side validation
6. Persist SSH key pair generation
7. Token-UID-status generation
8. Persist Compute
9. My Virtual Machines List
10. Virtual image selection
11. Job Wrapper
12. Wrap Sub Exec
13. Wrap Sub Command
14. Execute

D. Salomoni, ISGC 2011

19-25 March, 2011
The Cloud Web Interface

- Integrated VOMS / gLite Argus support
  - VO selection validate through VOMS servers
  - Resource usage (wallclock time) billed to the VO
  - Access authorization based on policies defined on an Argus server
Resource Selection

Create a New Virtual Machine

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 GB HD, 800 Mb/s throughput

Resource selection (note network bandwidth throttling)

X.509 Digital Certificate
Either upload an existing public key, or ask the system to generate a public/private key pair (*).

(*) security disclaimers apply
Public/Private \texttt{ssh} keys

Single sign-on (reusing an ssh key pair)

The ssh public key is put by WNoDeS on the created VMs using \texttt{libguestfs}
VM Creation, Deletion

The WNoDeS OCCI interface will be contacted to eventually create a single VM, or a pool of machines with the same hardware characteristics (physical names will of course be different).

VMs will continue to exist until explicitly destroyed, or until maximum wallclock time for the user is reached.
Besides command line tools, it is also possible for administrators to check the status of WNoDeS VMs via a Web portal

- Grouping VMs e.g. by name, memory used, owner, group via a treemap-based representation
- Simple monitoring of single VMs for what regards CPU usage, local I/O, network
WNoDeS Web-based Administration (2)
Next: the Integrated Grid/Cloud Portal

- Key goals:
  - Web-based access for the submission of Grid jobs (w/ JDL customizations)
  - Allowing “Grid users” to instantiate Cloud resources
  - Allowing “Cloud users” to exploit Grid resources
  - Integration of accounting visualization (HLRmon portlet)
Outline

- Introduction to WNoDeS
- WNoDeS Cloud Provisioning
- Conclusions
Conclusions

- The WNoDeS Cloud Portal:
  - For users, it offers a simple way to self-provision Cloud systems
    - “Grid” users do not need to acquire additional credentials
  - For site administrators, it provides a graphical view of the state of the local WNoDeS installation
  - For both, it exploits the ability of WNoDeS to dynamically configure resources (polymorphism) out of a common pool
- The first public release of the WNoDeS Cloud Portal will be in WNoDeS 2 “Harvest”
- Future developments will integrate the existing portal in a comprehensive Grid/Cloud submission/resource management system