Virtual Working Nodes in gLite middleware

A. Kretsis, P. Kokkinos and E. A. Varvarigos
Computer Engineering and Informatics Department, University of Patras, Greece
Research Academic Computer Technology Institute (RACTI), Patras, Greece

Introduction

We extend the gLite middleware so as to create and efficiently manage *virtual Working Nodes* (WNs) that run in actual nodes in the Grid Network.

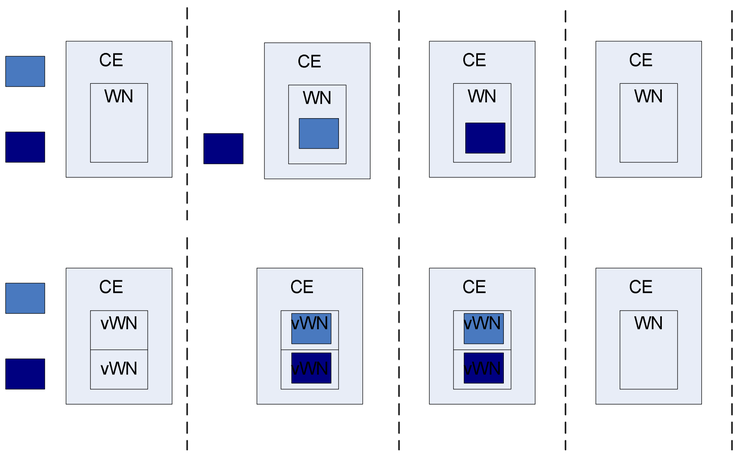
The basic idea is the dynamic provision of additional Working Nodes through a process controlled by dedicated middleware services, transparently from the rest of the middleware and the end users. The additional WNs are created on the fly as *virtual WNs,* using the virtualization technology.

Based on our architecture *a virtual WN* can be hosted at any node that has enough resources. In this way we can better exploit the existing Grid infrastructure.

Description and Motivation

Virtual WNs are initiated:

- **Dynamically**, by middleware services, based on Grid status and several policies.
- **Statically**, when user requests from the Workload Management System (WMS) advanced services such as an execution environment or guaranteed percentage of the computational capacity.



Apply advanced scheduling policies using virtual WNs

- Specify the percentage of CPU capacity a virtual WN is using
- Create homogeneous virtual WNs

Each *virtual WN* is configured for a particular community, providing specific services.

- In general if a pre-configured WN is not used by the targeted community, then it cannot be used from other communities and for other purposes
- Virtual WN with different configurations may co-exist in the same physical host and initiated only when needed
- Create on the fly execution environments with specific requirements on libraries, software, configuration etc

Using a number of concurrent virtual WNs becomes possible to execute more jobs at the same time in one node.

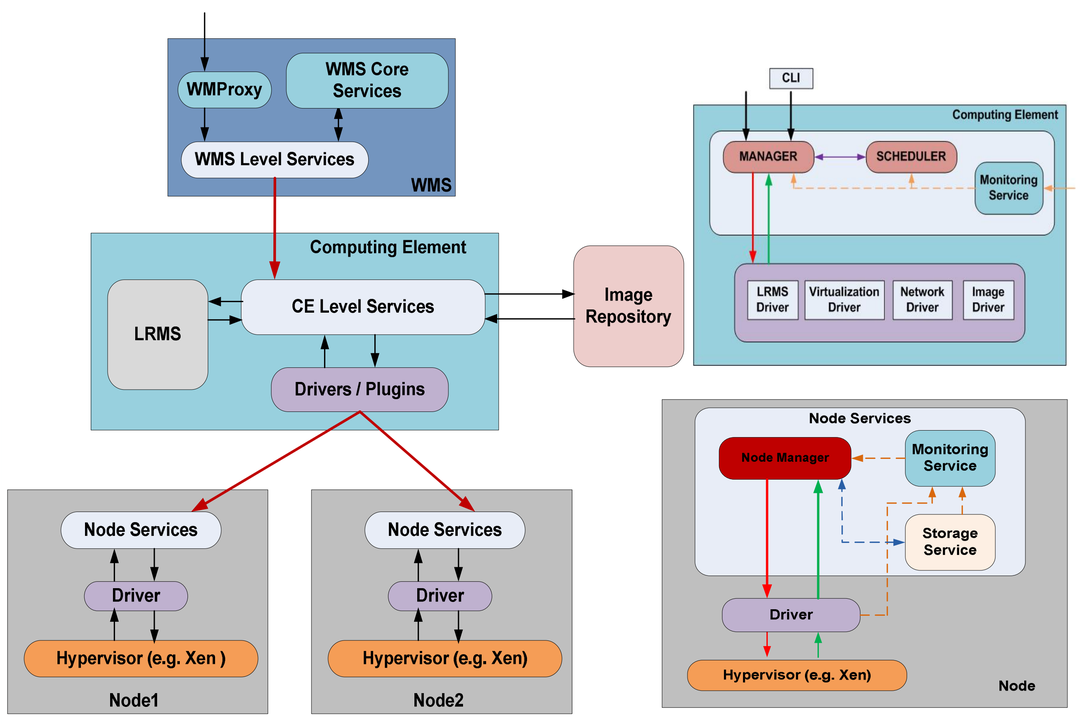
- Better utilization of available resources, without any extra hardware requirement
- Less capital and operational expenditures
- On demand scaling of the available resources in the Grid
- Use dynamically as virtual WNs nodes that normally run services with low load

Advanced fault tolerance services.

- By executing tasks in virtual WN is possible to pause their execution or transfer a copy of them in an other virtual WN and run both tasks concurrently.

Framework Architecture

Figure show the overall architecture of the proposal framework, the basic components, their elementary interaction and internal services.



WMS level services, interact with the WMS services providing new services to users, efficiently manage the virtual WNs and resources while the same time the complexity of the operations is hidden from the users.

- Efficiently scheduling of the virtual WN creation
- Ability to adapt to the changes in the heterogeneous Grid environment
- New services to the end users (create execution environments, guaranteed CPU allocation, fault tolerance etc)

CE level services, manage the life-cycle of the virtual WNs that belong to a particular Computing Element.

- Appropriate actions for executing commands by the WMS level services
- Orchestrate the interaction between framework services such as LRMS, hypervisors, image repository and network configuration
- Local scheduling of the virtual WNs
- Accessed directly by the administrators through an command line tool

Node services, offer hypervisor-enabled hosts that provide the resources needed by the Virtual WNs.

- Receive operational commands by the CE level
- Interface to local hypervisor through the usage of the appropriate driver
- Monitoring, publishing information for the status of the node and each virtual WN

Image Repository, any storage medium that holds the base images of the available virtual machines. The framework requires the existence of some images in some known location.

Drivers, programs and scripts used to interface with services that their types and characteristics can vary between infrastructures. Their usage provides to the framework the ability to operate transparently and independently from the various configurations.

- LRMS that uses a CE and its WNs
- Hypervisor technology that a resource provides to the framework
- Storage file system for the virtual machine images
- Network configuration of the nodes

Grid Testbed

The proposed framework was evaluated in a small Grid testbed, using the gLite 3.1 middleware. The testbed consisted of 12 machines: 1 glite-SE/LFC, 1 glite-BDII, 1 glite-MON, 1 glite-UI, 1 glite-VOMS , 1 glite-WMS/LB , 3 lcg-CE and 3 glite-WN.

The SE and the three CEs have the following characteristics:

SE capacity	**60 GB**
CE/WN 1	Ram 512 MB, Pentium 4 3.0 GHz , Node Services
CE/WN 2	Ram 256 MB Pentium 4 1.8 GHz
CE/WN 3	Ram 512 MB Intel Core 2 DUO 3GHz, Node Services

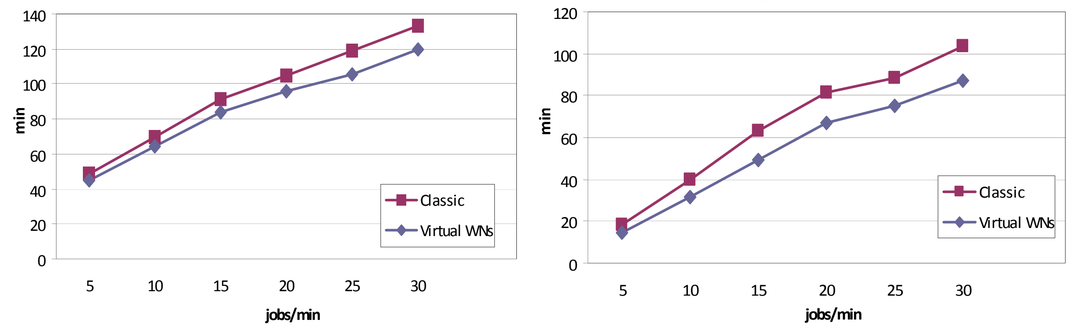
In our experiments we used five users for job submission belonging to the same Virtual Organization, having the same rights at the resources. We installed the Node Services of the framework, using the Xen Virtualization, in two WNs.

Job submission rate follows an exponential distribution, with mean values: 5, 10, 15, 20, 25 and 30 jobs/min. We used five kinds of jobs with different durations: 3, 5, 7 ,9 and 11 each job created is selected probabilistically from a uniform distribution. All users use the same ranking expression, preferring the CE with the fewer assigned jobs.

Experimental Results

The algorithms are evaluated using the following metrics:

- Average job execution time
- Average job queue time
- Max job queue time
- CPU utilization at the three CEs



Conclusions & Open Questions

The proposed idea is exciting however there several open questions that need further investigation.

- Implement sophisticated policies for scheduling tasks to virtual WNs
- Need for real-time monitoring information regarding the status of nodes and virtual WNs
- Policies for the execution environments' creation
- A more flexible way to configure the virtual WNs (related to the yaim)