ISTITUTO NAZIONALE DI FISICA NUCLEARE

Sezione di Perugia

INFN/TC-05/10
July 4, 2005

DESIGN, IMPLEMENTATION AND CONFIGURATION OF A GRID SITE WITH A PRIVATE NETWORK ARCHITECTURE

Leonello Servoli[1,2], Mirko Mariotti[2], Francesco Truncellito[1]

[1] INFN, SezionediPerugia, I 06123 Perugia, Italy
[2] Dip. Fisica, Univ. di Perugia, I-06123 Perugia, Italy

Abstract

We present a possible solution for the configuration and implementation of a GRID site with the Worker Nodes belonging to a private network, using LCFGng as a tool for managing the configuration, and PBS as batch queueing system, in the framework of the INFN GRID.

PACS: 07.05.Bx

*Published by **SIS–Pubblicazioni***
Laboratori Nazionali di Frascati

1 Introduction

1.1 What is the GRID

A computing GRID system (GRID) is a way to share computing power and data storage across a geographically distributed environment; hence the GRID defines an infrastructure that enables the integration of several kind of resources:

1. high-end computers;
2. fast networks;
3. databases;
4. scientific instruments.

GRID applications often involve large amounts of data and/or computing power and also require also secure resource-sharing across organizational boundaries, and are not easily handled by today's internet and web infrastructure. All the software needed to build a GRID is generally called *middleware* and it shields the user from the complexities of the underlying technology, offering an interface to one large virtual computer.

1.2 How does it work

There is no single standard architecture to implement a GRID. Many organizations and companies (such as IBM, Apple, etc.) have created their own GRID structure. Usually all the architectures are hierarchical. At the top there are the decisional nodes hosting several high level services, while at the bottom there are the worker nodes and the storage nodes that perform the very basic operations required to the GRID by the users. Figure 1 shows a generic grid architecture with 3 different kind of nodes:

- **Resource Broker:** it is the managing node that keeps a database about the status information of every GRID site; it receives the job submit demands that must be distributed to the GRID sites.

- **Provider:** it is the interface node among the Resource Broker and the Agents. Its principal tasks are: to collect the information about the status of the entire GRID site and to forward it to the Resource Broker, to accept from the latter the job submit requests, to distribute them to the local Agents and to return the output data to the node who initiated the job request.

- **Agent:** it is a terminal node that executes the jobs which are assigned to it.

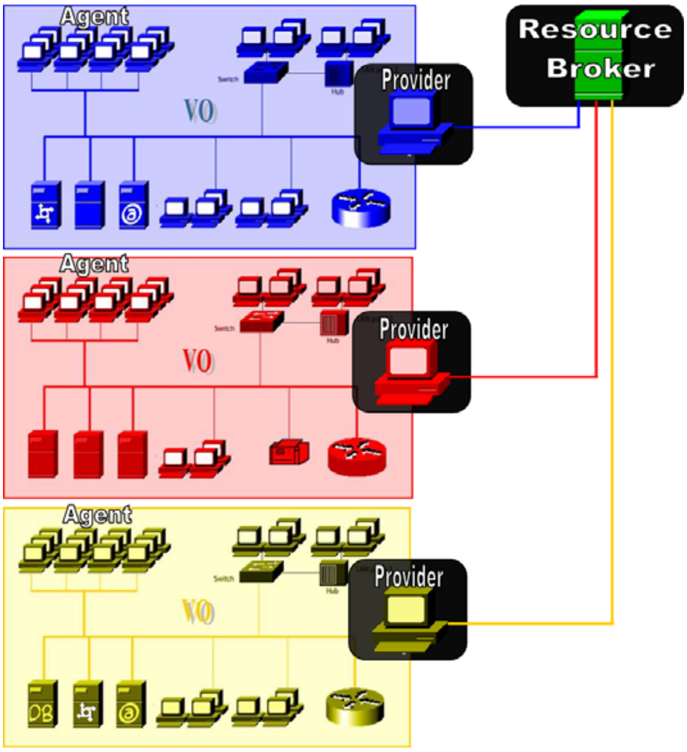

Figure 1: Generic Grid architecture

1.3 INFN and the GRID

The interest of the National Institute for Nuclear Physics (INFN) in the GRID concept is due to the inherently international effort needed to design, build and operate large experiments at the particle colliders such as the Tevatron at FNAL and LHC at CERN. These collaborations, that rely on thousands of scientists, need to share the computing resources available and to use them as efficiently as possible, and all as a single tool. As of 2007, the *LHC* will host four experiments: **ALICE** (A Large Ion Collider Experiment), **ATLAS** (A Toroidal LHC ApparatuS), **LHCB** (Large Hadron Collider Beauty experiment), **CMS** (Compact Muon Solenoid)[1].

These detectors will enable the detection and recognition of thousands of particles which are generated when proton beams collide at rates of forty million times per second; the foreseen data recording rate is 1 PByte/year. This huge amount of data, that should be analysed by thousands of scientists, residing in dozens of countries, has led to a general interest for GRID technologies, and to an effort for developing, together with other scientific communities (biologists, weather forecasts, etc.), a new GRID infrastructure, driven

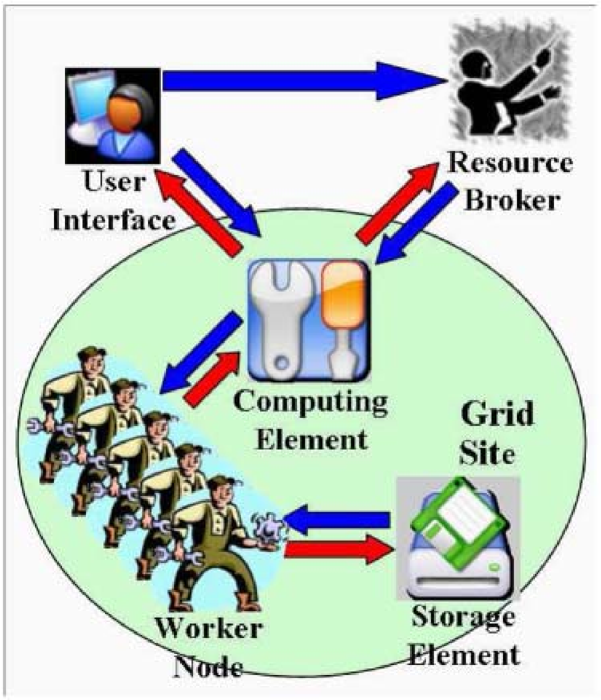

Figure 2: LCG Architecture

by data handling and efficient CPU sharing.

INFN has been one of the main institutions leading this software development, both participating to EU funded efforts, as well as promoting national initiatives. The outcome of the first EU funded project, DATAGRID, and of the related work, has been the middleware called LHC Computing GRID (LCG), which is currently managed by the LCG project at CERN [2]. This is the baseline of the deployed GRID, while its developments are carried on by the new EU initiative EGEE (European Grid Enabling E-Science).

INFN-GRID is the INFN national infrastructure [3] , based upon LCG version 2.3.X, that is linked to the international GRID, and is currently built upon 20 or so GRID sites, managed by more than one hundred scientist and engineers.

In figure 2 the Architecture of the LCG middleware is shown and it is possible to notice the presence of five different kinds of nodes:

- **Resource Broker**: it is the managing node and it is responsible for keeping information on the status of the entire GRID, for collecting the requests coming from the users, and to distribute them to the GRID, performing matchmaking with the available resources.

- **Computing Element**: it is the interface betweeen one GRID site and the rest of the GRID. It is responsible for keeping detailed knowledge of the GRID site and for distributing locally the requests sent by the Resource Broker.
- **Storage Element**: it is responsible for the management of large amounts of disk space.
- **User Interface**: it is an Agent that interfaces the user and the GRID. It sends the user's requests to the Resource Broker and it receives the output of the jobs.
- **Worker Node**: it is an Agent that offers the CPU server functions.

If a user wants to submit a job to the GRID, it must have an account on a User Interface, which supplies all the tools needed to submit and monitor the status of a job. The User Interface forward the user's request to the Resource Broker who decides which GRID site has the resources necessary to satisfy the job, and then informs the relevant Computing Element that there is a new job to run. The CE sends the job for the execution to a Worker Node that may use a Storage Element as cache area. Finally the Worker Node returns the output to the CE from which it is forwarded to the user.

The current version of the middleware is LCG 2.3.1 and is based on Linux platforms RedHat 7.3 and Scientific Linux 3; with the next release only the Scientific Linux platform will be supported.

The official tool used to create and maintain the *LCG* GRID site is **LCFGng**.

2 LCFGng Tools

2.1 History

The **Local Configuration system** (LCFG) [4], is designed to manage and configure many workstations and servers within a single framework. The original version, developed in 1993, ran under *Solaris*, and later it was ported to *Linux*. Today, the versions supported are *Red Hat* 7 and 9. The latest version of LCFG is called LCFGng and from now on, for simplicity, it will be referred to as LCFG.

2.2 The environment

LCFG is based on a central database where the definitions of every node and of the entire site are stored. All nodes will be automatically installed following the configuration stored in the central database. To change the settings of a single node, the corresponding configuration files in the LCFG server have to be modified.

The LCFG architecture is designed to satisfy multiple requests:

1. Declarative configuration specification: every node has a configuration specification defined through files.

2. Devolved management: the responsibility for the installation of a site could be shared among several people, each one in charge of the configuration of a single aspect, because LCFG tools are capable of putting together all these pieces to create a complete profile for an individual node.

3. Variety of nodes: LCFG is capable of managing either nodes that share exactly the same configuration, or ones that have no aspect in common.

4. Update: LCFG contains tools capable of managing the entire site or single nodes when there is a change in the configuration.

5. Automation: all the steps (installation, configuration and management), are executed automatically.

6. Correctness: the configuration system should be able to configure systems completely from the declarative specifications with no manual intervention. The automation permits a high degree of confidence on the correctness of the configuration.

2.3 The Architetture

The figure 3 show the architecture of the LCFG Framework:

- The configuration of the entire site is created in many declarative files maintained on the central LCFG server. A node description is normally contained in a small number of configuration files, and usually, each file describes a single aspect of the machine. An include system, similar to the C++ language, forms a tree structure that describes all the node aspects.

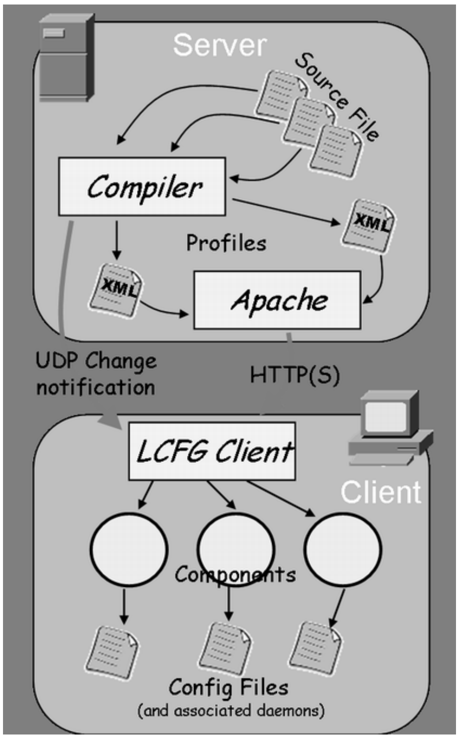

Figure 3: LCFG architecture

- The LCFG compiler compiles these source files, and creates a single XML profile for each machine. The profile contains all the configuration parameters, including variables and values for all aspects needed for the installation step.
- A standard web server (as Apache), publishes all the profiles. The profile of each node is used, at startup time, to install the node.
- If a node is already installed and a profile changes, the server sends to the client an UDP notification. The node then downloads this profile via HTTP and applies the changes.
- Periodically the client sends an UDP acknowledgement to the server; this notification is stored and used to create an HTML web page, published by the server, that shows many informations about the client. Looking at this page, it is possible to monitor the status of all clients.

3 Standard and modified GRID site

3.1 Standard architecture

A typical LCG 2.3.X installation is shown in figure 4. It consists of a single CE, one or more SEs, several WNs and optionally one or more UIs. In the standard architecture, all the GRID elements have only one network interface connected to the public LAN.

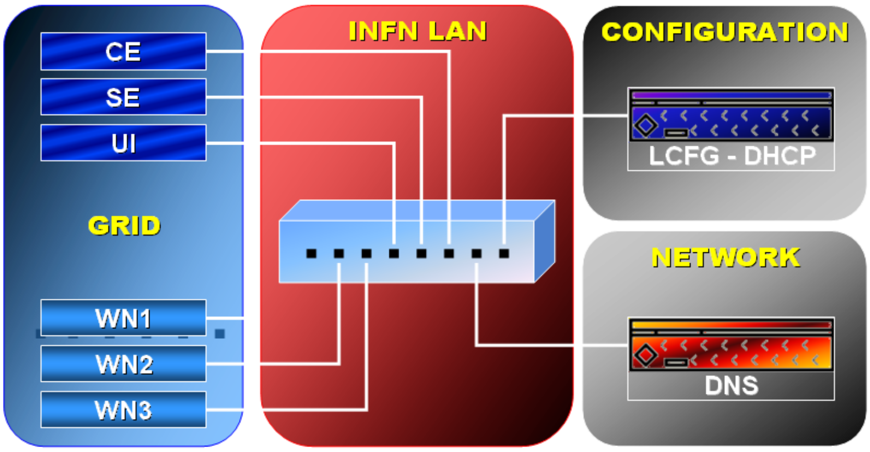

Figure 4: Standard GRID site architecture

On the LAN there is also an LCFG server that manages the installation of the GRID elements. The services necessary to assure the functionality of the GRID site are the following:

- DNS (Hosted on the external LAN): every node should be registered on it.
- DHCP (Hosted on the LCFG server): it is a non authoritative DHCP and it is used for netbooting the nodes. It assigns network parameters to the nodes and provides to every LCFG client the location of its profile.
- Batch System and Scheduler (Hosted on the CE): it provides the resource management and the job control for the site, every WNs is a client of the Batch System.
- Monitoring System (Hosted on CE and SE): it collects all the relevant informations from the WNs to monitor the use of the resources.

In this architecture, an external job comes from the RB to the CE and is then assigned to a WN. In order to achieve this, between CE and WNs it must be possible to exchange files in HostbasedAuthentication mode, i.e. among all the nodes of the site, the user password is not needed.

3.2 Modified architecture

In several sites it is not possible to assign public IP addresses to all the nodes, due to the saturation of the assigned IP class. Hence, it is important to define a new configuration where all the worker nodes are on a private network. This will also simplify the security issues.

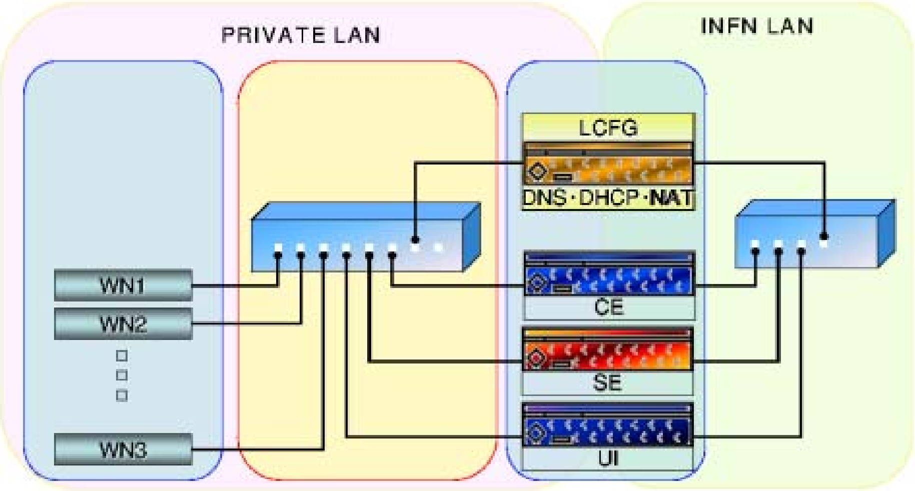

Figure 5: Modified GRID site architecture

The proposed architecture is shown in figure 5. All the GRID elements, other than the WNs, need to have two network interfaces. The first one, with a public IP address, to be used to communicate with the external world and the second one with a private IP address, to be used for the traffic inside the GRID site. A dedicated switch for the private network is also needed.

For this modified installation, some services have to be reconfigured and other features have to be added:

- DHCP: It runs on the same machine as the LCFG server, but now it must serve two different networks because the public network elements (CE and SE) are configured as before while the private ones (WNs) need a different configuration. From now on, this machine will be called the Site Gateway (SG).
- DNS: It is necessary to install a dedicated DNS server to resolve the addresses inside the private network. It can be installed on the SG.
- WNs outbound connectivity: The SG has another important role. It has to be configured as the network gateway for the WNs so that they may have outbound connectivity needed for the data transfer operation requested by a job.
- LCFG modifications: There are many changes to be made to the standard LCFG configuration files that come from LCG, mainly due to the configuration of the private network.

4 Implementation

To implement the new GRID site structure, three sets of modifications have been made to the standard configuration. The first one is needed to build a functional private network infrastructure. The second one contains the changes related to the GRID elements configuration within the LCFG. Finally some post-installation tasks have also to be performed.

4.1 Private network infrastructure

In a standard GRID site certain services, such as DNS, are provided by the external network, but in order to put the WNs on a private LAN, the site has to provide itself with such service, because the external network does not know about the internal one.

The best place to put the DNS server is the SG because it should serve both the internal and the external network and it is the only node that can be manually configured. In fact, all the other nodes are managed via LCFG and it is quite difficult to manage the DNS this way. Every machine in the site has a NIC on the private LAN so on the DNS server a zone is configured mapping them, in order to allow the resolution of the names.

This server is the primary DNS for all the GRID site elements because it is capable of resolving also the external names by querying the external DNS.

The configuration of the DHCP server is divided in two sections: the first one defining the parameters for the bootstrap of the CE, SE and UI via the external NIC (same as in the standard configuration), and the second one related to the bootstrap of the WNs via the internal NIC belonging to the private domain just defined.

In order to allow outbound connectivity to the WNs, the *Netfilter*'s rules have been inserted in the SG kernel using *iptables*. The same technique, with different rules, has to be used to allow the correct functioning of AFS on the WNs.

4.2 Site configuration via LCFGng

The standard installation on LCG was not including the support for WNs on private LAN, and hence some modifications to the configuration have been made. The internal elements (i.e. the WNs) and the external ones (the others) have different names and parameters so the first thing needed is an LCFG variable (PRIVATE-NETWORK), defined in the site-cfg.h file, to choose between a standard installation and a modified one. When this variable is set to 1, several options became available and the most important ones are:

- The configuration of two NIC instead of one, on the elements where it is needed.
- The configuration of the batch system, to reflect the fact that the WNs are on the private network while the SE and CE are on the public one. In fact the *torque mom* on the WN identifies its server using the hostname and this implies, that in this configuration, we have to allow both names of the CE to be used for identification.

4.3 Post-installation tasks

In some cases, depending on the services, need may arise to define some static routes on WNs so as to avoid that the traffic going to the SE, CE or UI uses the external NIC. The batch system may need some adjustement too, such as changing the server ACLs.

5 Conclusion

We have installed and configured a GRID site with the WNs on a private network. All the necessary modifications have been included in the INFN-GRID release of the LCG middleware and in the related documentation, starting from version 2.1.0. We do plan to port this configuration in the future version of the LCG middleware.

6 Acknowledgment

We would like to acknowledge E.Ferro for his kind advice.

References

[1] http://cms.cern.ch/, http://atlas.cern.ch/, http://alice.cern.ch/, http://lhcb.cern.ch/

[2] http://lcg.web.cern.ch/LCG/

[3] http://grid-it.cnaf.infn.it/

[4] http://www.lcfg.org/