A GRID APPROACH FOR GRAVITATIONAL WAVES SIGNAL ANALYSIS WITH A MULTI-STANDARD FARM PROTOTYPE.

F. Acernese[1,2], F. Barone[2,3], R. De Rosa[1,2], A. Eleuteri[1,2], R. Esposito[2], L. Giordano[2,4], P. Mastroserio[2], L. Milano[1,2], S.Pardi[2,4], G. Russo[1], F.M. Taurino[5], G. Tortone[2],

[1]*Università di Napoli "Federico II", Dipartimento di Scienze Fisiche, 80126 Napoli, Italy*

[2]*INFN – Sezione di Napoli, 80126 Napoli, Italy*

[3]*Università di Salerno, Dipartimento di Scienze Farmaceutiche, 84084 Salerno, Italy*

[4]*Università di Napoli "Federico II", Dipartimento di Matematica ed Applicazioni "R.Caccioppoli", 80126 Napoli, Italy*

[5]*INFM – Unità di Napoli, 80126 Napoli, Italy*

Abstract

The standard procedures for the extraction of gravitational wave signals coming from coalescing binaries provided by the output signal of an interferometric antenna may require computing powers generally not available in a single computing center or laboratory. A way to overcome this problem consists in using the computing power available in different places as a single geographically distributed computing system. This solution is now effective within the GRID environment, that allows distributing the required computing effort for specific data analysis procedure among different sites according to the available computing power.

Within this environment we developed a system prototype with application software for the experimental tests of a geographically distributed computing system for the analysis of gravitational wave signal from coalescing binary systems. The facility has been developed as a general purpose system that uses only standard hardware and software components, so that it can be easily upgraded and configured. In fact, it can be partially or totally configured as a GRID farm, as MOSIX farm or as MPI farm. All these three configurations may coexist since the facility can be split into configuration subsets. A full description of this farm is reported, together with the results of the performance tests and planned developments

INTRODUCTION

The detection of gravitational waves (GW) is one of the most interesting fields of the modern physics: it will provide a strong proof of the general relativity theory, opening in this way a completely new channel of information on the dynamics and evolution of astrophysical objects [1].

Within this framework, the large scale terrestrial interferometric detectors like VIRGO [2], LIGO [3], GEO [4] and TAMA [5], will have a prominent role. In fact, these detectors will operate with large detection bands, typically spanning from 10 up to 10 kHz, with a sensitivity of about 10^{-21} h/(Hz)$^{1/2}$ at 100 Hz, where h is the gravitational strain. In addition the very long baseline space interferometer LISA [6] will explore the detection frequency band from 10^{-5} up to 1 Hz.

For all these detectors, but especially for earth-based antennas, the main problem to solve in data analysis is the expected low signal-to-noise ratio (SNR). The low value of this quantity is due to the intrinsic weakness of gravitational signals with respect to the instrumental noise.

To overcome this problem many efforts are being done in the development of suitable data analysis techniques. When the expected signals shape is known, the most promising technique seems to be the matched filtering, i.e. the correlation of the detector output with a set of theoretical waveform templates.

In the case of the VIRGO antenna the required computing power for detecting gravitational waves generated by a coalescing binary systems is about 300 GFlops for masses ranging from 1.4 to 50 solar masses, assuming a signal-to-noise recovery of 90% [7][8]. These performances can be obtained using computer farms composed by several nodes connected each other through the network. This technical solution represents the only present possible way for an on-line data analysis. Of course a more accurate analysis can be performed off-line. In this case there are no time constraints and therefore the number of used templates can be easily increased and other parameters can be also included in the model. For this reason the off-line phase usually requires a very large amount of computing power that can be obtained only by adding more computing resources. The direct consequence of this approach is that a problem of optimum algorithms development must take into account the farm architecture and configuration. For this reason, we implemented a very versatile and modular computing power tool in Napoli, whose configuration can be dynamically changed according data analysis tests, becoming a MPI farm, a MOSIX farm or a GRID farm. In particular, the GRID solution is based on the idea of using the existing GRID environment as a platform to connect several geographically distributed computing and storing resources [9]. In this way the tasks needed for the off-line data analysis can be performed by dividing them in subtasks to be executed on remote computers. The GRID architecture also provides all the tools needed to collect back the results of the analysis.

HARDWARE ARCHITECTURE

The Beowulf cluster is a not homogenous farm composed from 20 nodes that can be divided into two main homogeneous hardware subsets. Furthermore another 5 machines are used as basic elements for the GRID infrastructurefarm management and for geographycal computing, as we will explain later in the paper. The two subset are:

Super Micro 6010H subset: 12 nodes each equipped with two intel Pentium III 1 GHz, 512 MByte RAM, a local 18 GByte SCSI disk and 2 integrated network connections on boards (Fast Ethernet 10/100). The first node is equipped with a further Giga Ethernet board, since it may act as farm master.

APPRO 2114Xi subset: 8 nodes each equipped with two Intel Xeon 2.4 GHz processors, 1024 MByte RAM, a local 60 GByte IDE hard disk, mother board TYAN, and two integrated network connections on board (Fast Ethernet and Giga Ethernet).

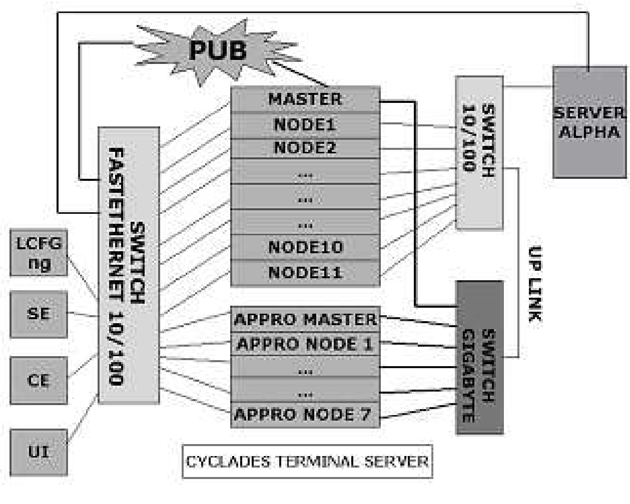

Fig.1
Hardware architecture of the Farm

The farm has two independent networks: a private network and a public network. The private network, used to optimize the data transfer speed among the farm nodes, is split into two subsections, consisting in a Fast Ethernet switch and a Giga Ethernet switch, respectively. An uplink connects these two switches to implement an equivalent single private network. The two private network subsections are necessary because the Super Micro nodes have Fast Ethernet links, while the APPRO ones have Giga Ethernet links. The public network is used for public access to the farm and for its management. It consists of a Fast Ethernet switch to which all the farm nodes are linked, together with an Alpha Server and all the basic GRID elements (the LCFGng, the Storage Element, the Computing Element and the User Interface). A Terminal Server (Cyclades S2000) is also connected to the public network for the farm remote management, while all the nodes are also connected to the Terminal Server through serial links. A scheme of the Farm Architecture is shown in Figure 1.

SOFTWARE ARCHITECTURE

All the farm nodes run the operating system Linux Red Hat 7.3 with OpenMosix kernel 2.4.20, being this operating system a *de facto* standard for high performance parallel machines, characterized by a high stability and effectiveness of the libraries for distributed computing. Moreover, as we already underlined, according to data analysis requirement our main goal was the implementation of a single nodes like blocks in a grid and to use a diskless architecture for the cluster nodes, defining two possible operational modes for the farm, that are

FARM MPI/OpenMosix
GRID CONFIGURATION

Excluding two nodes, that play the role of sub-farm masters (i.e. the first SuperMicro node (MASTER) and the first APPRO node named MASTERAP), all the nodes can be remotely and independently configured in one of the two modes, through simple scripts.

If a node is configured in Mode 1 (MPI-OpenMosix), then at the boot stage it asks for the kernel through TFTP from node MASTER, where the kernel is installed mounting the *root* directory and all the packages, via NFS. The local disk of the node is used as swap area. The APPRO nodes have a further possibility. In fact, each APPRO can mount the {\it root} directory also from the node MASTERAP, setting up, in this way, two independent but homogeneous farms, configuration often necessary during the test phase of data analysis algorithms. This solution gives a large flexibility to the whole infrastructure, allowing the execution of homogenous performances tests for the data analysis algorithms. As outlined above, all the nodes working like MPI farm constitute a mini-cluster, that may coincide with the whole farm. Furthermore, the farm can also work with OpenMosix, a patch of the kernel of Linux allowing the dynamic balance of the computing charge of the nodes, a very useful so

Software tool for scheduling jobs also in not homogeneous farms. This is also another good reason for having two separated networks (private and public). In fact, being the cluster diskless, the system calls of the nodes are dispatched through the network to the node MASTER (or MASTERAP), and a large data transfer with OpenMosix or MPI on the network may cause the failure of some node. The most direct solution is to have two networks, a public network to mount the *root*, the disks, to access the machine from outside, etc., and a private network for the parallel computing data transfer. In this way the farm becomes very stable and performant.

If a node is configured in Mode 2 (GRID), then it executes the boot from the local disc where the Linux RedHat GRID distribution is installed together with all the packages necessary for GRID applications. In this case the private network is not used and the nodes work stand-alone.

GRID INFRASTRUCTURE

The Virgo Laboratory is also a VO (Virtual Organization) registered at Test-Bad of project GRID-it managed from the CNAF of INFN; so the Grid infrastructure is conformed to the requirement dictated from the plan (see Fig.2), and use the LCG software based on globus (today the machine works with version 2.0.0).

We have 4 machines dedicated for the GRID architecture named GRID-Element, that is:

CE (Computing Element): that represents the front-end between the grid and the Worker Node of the farm.

SE (Storage Element):a grid machine that we used for the storage of the remote data.

UI (User Interface): a grid machine that works as grid interface for all users to submit our jobs.

LCFGng Server: the server that installs and manages the machine of GRID infrastructure.

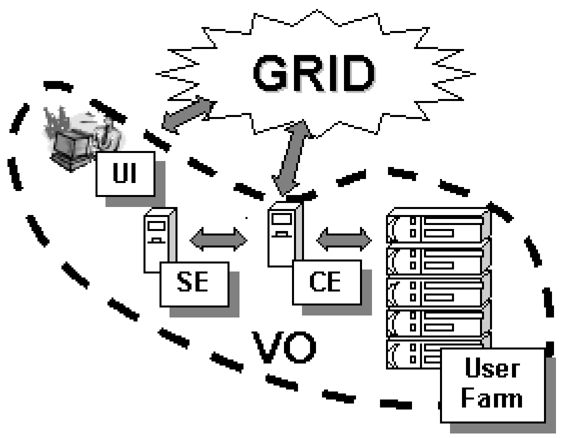

Fig.2
Grid Infrastructure

FARM MANAGEMENT

Every node of farm can be remotely managed through scripts using a terminal server. For this task, they were configured at the BIOS level to execute the boot from the LAN using PXE, a small bootloader present on the net card, executed before asking the DHCP server the IP address, and then using TFTP to obtain the kernel with the instructions for {\it root} mounting. LCFGng is the machine that manages the cluster, acting as DHCP server, receiving and serving the IP address requests of the nodes and providing them with all the information for boot execution. The management scripts work on the configuration file dhcp.conf, in which it is possible to associate an IP address and a next server to every NIC. This is the next machine to address for reading the PXE configuration file with the kernel name, the machine to which addressing TFTP and the information for *root* mounting. The boot sequence is described in Figure 3 and 4

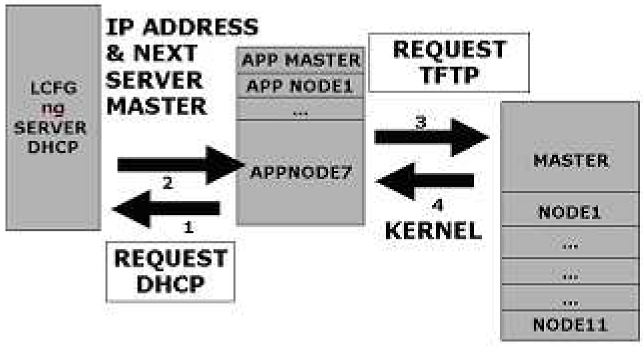

Fig.3
Boot sequence for the MPI/OpenMosix mode

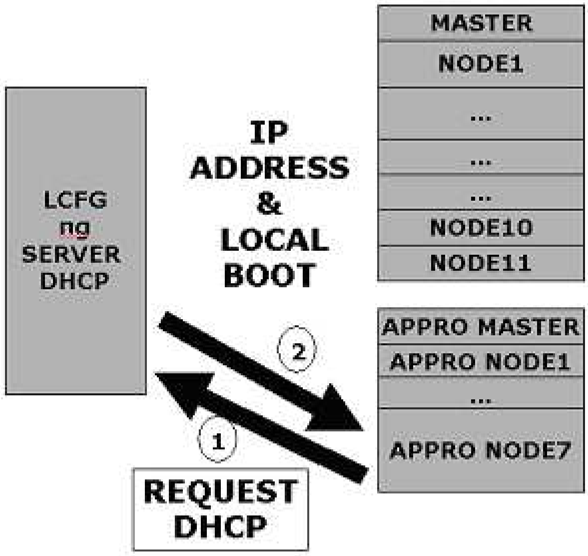

Fig.4
Boot sequence for the MPI/OpenMosix mode

To set-up a machine like a MPI/OpenMosix farm, the script sets up node MASTER like next server and then sends it a {\it shutdown} command. Once loaded the kernel, the node completes its configuration mounting the root directory via NFS from the same MASTER machine.

If a node has to boot in GRID mode, its corresponding line on

the dhcp.conf is commented, so that the machine boots using the file of the PXE configuration present on the LCFGng, that commands the execution of the local boot.

TEST OF PERFORMANCE

After the implementation of the farm, we have ran many benchmarks (Bonnie, Glibench, Netpipe) to estimate the CPU, file system and network performance of machine. The result of the tests are presented in the Figures 5, 6 and 7 respectively.

CPU (glibench 0.2.5)		
CPU (SMP)	**Dhrystones** (MIPS)	**Whetstone** (MFLOPS)
Pentium III 1 ghz	4106	1207
Xeon 2.4 ghz	6265	1402

Fig. 5
CPU performance

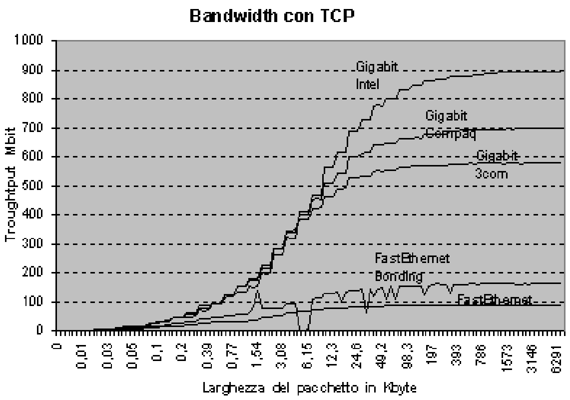

Fig. 6
Network throughput

DISK (bonnie++ 1.02a)		
DISK	**Block read** (Kbyte/s)	**Block write** (Kbyte/s)
SCSI	28566	32325
IDE	10434	1087

Fig. 7
Disk performance

CONCLUSION

We have implemented a not homogenous Beowulf Cluster flexible with a very easy management thanks to one web interface very user-friend. The possibility of

being able to change blots them of it in two configurations GRID and local FARM allows to a laboratory with resources limited to participate to the new GRID plans without to renounce every time to a farm local that it is necessary.

REFERENCES

[1] C. W. Misner, K. S. Thorne, J. A. Wheeler, Gravitation (Freeman & Co., San Francisco, 1973).

[2] C. Bradaschia et al., “The VIRGO Project, Final Design of the Italian-French large base interferometric antenna of gravitational wave detection”, Proposal to INFN Italy and CNRS France, 1989, 1992, 1995.

[3] R.E. Vogt, R.W. Drever, F.J. Raab, K.S. Thorne, “Proposal for the construction of a large interferometric detector of gravitational waves”, Proposal to the National Science Foundation, California Institute of Technology, Pasadena, California, USA, 1989.

[4] Hough et al., “Proposal for a joint german-british interferometric gravitational wave detector”, MPQ 147, Max Planck Institut für Quantenoptik, Munich, Germany, 1989

[5] unpublished, see http://tamago.mtk.nao.ac.jp (1996)

[6] P. Bender, et al. “LISA: Laser Interferometer Space Antenna for the detection and the observation of gravitational waves},MPQ 208, Max Planck Institut für Quantenoptik, Munich, Germany, 1996.

[7] B. Owen, 1996, Phys. Rev. D, 53, 6749.

[8] P. Canitrot, L. Milano, A, Vicere', VIR-NOT-PIS-1390-149, 2000

[9] I. Foster, 2002, Physics Today, 55, 42

[10] L. Blanchet, T. Damour, B. R. Iyer, 1995, Phys Rev. D, 51, 5360

[11] C. W. Helstrom, Statistical Theory of Signal Detection, 2nd ed. (Pergamon Press, London, England, 1968)

[12] Chervenak, I. Foster, C. Kesselman, C. Salisbury, S. Tuecke, 2001, Journal of Network and Computer Applications, 23, 187