2013 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing

The HPC testbed of the Italian Grid Infrastructure

R. Alfieri[*], S. Arezzini[†], G. B. Barone[‡], U. Becciani[§], M. Bencivenni[¶], V. Boccia[‡], D. Bottalico[‡], L. Carracciuolo[‖],
D. Cesini[¶], A. Ciampa[†], A. Costantini[**], S. Cozzini[††], R. De Pietri[*], M. Drudi[‡‡], A. Ghiselli[¶], E. Mazzoni[†],
S. Ottani[x], A. Venturini[x] and P. Veronesi[¶]

[*] *University of Parma and INFN Parma,*
viale G.P. Usberti n.7/A, 43124 PARMA, Italy (roberto.alfieri@unipr.it)

[†] *INFN Pisa, Polo Fibonacci Largo B. Pontecorvo 3, Pisa, Italy*

[‡] *INFN IGI and University of Naples Federico II,*
Complesso Universitario di Monte Sant'Angelo, via Cintia, 80126 Napoli, Italy

[§] *INAF, via S.Sofia 78, 95123 Catania, Italy*

[¶] *INFN IGI, via Ranzani 13/2, Bologna, Italy (daniele.cesini@cnaf.infn.it)*

[‖] *INFN IGI and CNR, Complesso Universitario di Monte Sant'Angelo, via Cintia, 80126 Napoli, Italy*

[**] *INFN IGI and University of Perugia, Perugia Italy*

[††] *CNR-IOM UOS Democritos, via Bonomea 265, 34136 Trieste, Italy*

[‡‡] *INGV, via Donato Creti 12, 40128 Bologna, Italy*

[x] *ISOF CNR, via P. Gobetti 101, 40129 Bologna, Italy*

***Abstract*—Even though the Italian Grid Infrastructure (IGI) is a general purpose distributed platform, in the past it has been used mainly for serial computations. Parallel applications have been typically executed on supercomputer facilities or, in case of "not high-end" HPC applications, on local commodity parallel clusters. Nowadays, with the availability of multiple cores processors, Grid computing is becoming very attractive also for parallel applications but some problems exist in supporting of HPC applications on Grid environment.**

Here we describe the work made to set up a HPC testbed for "not high-end" HPC applications, based on IGI Grid technologies, to find solutions to those problems. Participating sites have been selected among the ones running HPC clusters in Grid environment. Each of them contributed with their specific HPC experience and their available resources to the present test, which encompasses an unprecedented large set of applications from different disciplines in the fields of astronomy, astrophysics, chemistry, climatology, material science and oceanography. In addition to computing resources sharing, the main contribution of each participant was the identification of the real requirements of his application also related to the current middleware limitations and then the realization of a test platform enhanced with additional HPC solutions and configurations developed in a tight collaboration between HPC administrators, users and IGI managers. The main work was on computational resources selection, data management and the definition, the deployment and the documentation of the software execution environment. The outcoming results of the testbed represent the basis of the HPC support in the IGI production infrastructure.

***Keywords*-High Performance Computing; GRID computing; Parallel applications; Distributed computing infrastructures;**

I. Introduction

The Italian Grid Infrastructure (see section III-A) is one of the 40 National Grid Initiatives (NGIs) composing the European Grid Infrastructure (EGI) [1], which provides computational and storage resources to scientific communities in Europe. Grid technologies have been introduced in Europe since early 2000s pushed by big science experiments and projects, starting from LHC (Large Hadron Collider) at CERN mainly focused on High Throughput Computing (HTC) exploitation of distributed resources.

Only recently, with the availability of multiple cores processors, parallel computing on the Grid is becoming very attractive. Many Grid clusters have been upgraded with the new multicore processors, becoming very interesting for different types of parallel applications (MPI, SMP and hybrid). An important step towards a better support for HPC in Grid has been the recent introduction of specific features enabling the use and exploitation of the cores granularity (see section III-B).

Despite the dedicated support that a number of NGIs provides to the European Grid community through the EGI "Heavy User Communities" activity, until now there was still significant issues in uptake and satisfaction of MPI services amongst various user communities. We considered however that in order to start a HPC production level support in IGI, a further step is needed in order to verify and test the actual middleware features, including the new *"Granularity"* functionalities, and to propose, verify and deploy new functionalities (i.e. the definition of a suitable *"Applicative Middleware"*) that, by an "high level" approach, may enable a flexible and easy porting of HPC applications to the Grid. For these reasons a specific testbed on the IGI infrastructure has been deployed, where selected sites contributed with their specific HPC experience and their available infrastructures experiencing one or more solutions to issues related to the support of HPC applications.

This challenging endeavour encompasses an unprece-

1066-6192/12 $26.00 © 2012 IEEE
DOI 10.1109/PDP.2013.42

dented large set of applications from different disciplines. HPC users in the fields of astronomy, astrophysics, chemistry, climatology, material science and oceanography provided test cases and support. Grid middleware was adjusted to meet the requirements of CPU and data storage intensive applications. The deployment of such a wide range of computational procedures has been realized within a tight multidisciplinary collaboration between HPC site administrators, HPC users and IGI experts, as detailed in the following sections.

This document is organized as follow: in section II we report two activities, made in european context, to improve the support for "not high-end" HPC applications, in Grid environments. In section III we describe the state of the art of HPC support in IGI. In section IV is reported a brief description of testbed infrastructure and in sections V, VI and VII the focus is on the enhancements introduced to solve some middleware issues related respectively to computing resource selection, software documentation and discovery and data management. In section VIII, for each one of the applications participating to the activity, are reported a brief description of the problem solved, the application needs and some results obtained from the distributed infrastructure use. Finally in section IX we summarize the activity main results and we provide an overview on future works.

II. Related works

Relevant works on HPC support in the Grid environment have been carried on at different levels. A deep work was performed to create UNICORE [2], a middleware specifically created for HPC applications, used at several supercomputer centers worldwide. UNICORE supports resource specifications through a Job description file, written in JSDL (the OGF standard language for job description) with the addition of extensions for parallel applications and the support for the *"Execution Environments"*. The Execution Environments, specified by the administrator by means of the IDB (Incarnation Database) file, provides to the users a high level of abstraction, hiding resources and applications details and providing simple execution environments to the user.

At community level, the European Grid Initiative (EGI) established has established a Virtual Team (VT-MPI) [3] of six-month duration in order to collect and to address the issues in uptake and satisfaction of MPI services amongst different user communities and make EGI a more attractive platform for MPI jobs. The VT-MPI work spanned across a number of "low level" technical areas as: (i) the definition of more detailed guidelines to configure and to use MPI resources, (ii) the definition of more accurate monitoring probes for the EGI Service Availability Monitor (SAM), (iii) the collection of issues related to parallel applications that have been reported to responsible technology developers and providers with request for addressing.

III. The IGI infrastructure

A. IGI Status

The Italian Grid Infrastructure (IGI) [4] started its activity years ago and it is presently the result of the participation to various Italian and European projects such as DataGrid [5], EGEE [6], Grid.it [7], INFNGRID [8]. It is now satisfying the computing and storage demand of various user communities - high energy physics, bioinformatics, astronomy and astrophysics, earth science. Nowadays, Italy is actively participating to the EGI-InSPIRE project [1] and is one of the biggest National Grid Initiatives of EGI with a recognised leadership both in the Grid technology development and in the distributed computing infrastructures operation for the scientific research communities.

The Italian Grid Infrastructure currently comprises more than 50 geographically distributed sites providing about 33000 computing cores and 30PB of storage capacity, 50 Virtual Organisations (VOs) with thousands of active users supported by the infrastructure. The infrastructure is run using a customised version of the gLite [9].

B. HPC support in IGI

As stated in the previous section, the actual Grid services supported in IGI are based on gLite components distributed by the EMI middleware [10]. The user can specify the needed execution environment through a Job Description Language (JDL) file. The basic support for the selection of the hardware environment relies on the `CPUnumber` attribute, which specifies the needed number of processors. This support has been recently improved [11] through the introduction of the *"Granularity attributes"*. These attributes allow users to specify:

(i) `SMPGranularity`: Minimum number of cores that should be allocated on any host.
(ii) `WholeNodes`: Whether whole nodes should be exclusively allocated for the job.
(iii) `HostNumber`: How many nodes should be used.

The software environment setup is based on the `MPI-Start` tool. MPI-Start [12] is an abstraction layer, located between the middleware and the underlying Local Resource Management System (i.e. Torque or LSF) and *"MPI flavors"*, that offers a unique interface to start parallel jobs with different implementations of the execution environment. The user has only to specify which MPI flavor will be used and to provide any pre-run and post-run scripts. Moreover, the tool allows a flexible management of the execution environment for mixed MPI/openMP jobs, through the following command line options: (i) `-pcore`, one MPI process per CPU core; (ii) `-psocket`, one MPI process per CPU socket; (iii) `-pnode`, one MPI process per node.

IV. Testbed infrastructure

Participating sites have been selected among the ones running HPC clusters in Grid environment. These sites contributed with their specific HPC experience and their available infrastructure, including special HPC components such as high speed networks, high performance shared file systems, libraries and tools. The selected sites provided clusters installed with the new EMI middleware, which supports the Granularity attributes (see section III-B). In particular all sites provide Cream Computing Element (CE) endpoints enabling users to submit directly their jobs, but they are all inserted into IGI production Grid by means of a set of core services [9]: the Virtual Organisation Management System (VOMS) for the *gridit* VO, the IGI information system based on the gLite Berkeley Database Information Index (BDII) and a series of dedicated Workload Management Systems (WMS) used to handle jobs submission and tracking. We chose to use dedicated instances of the WMS service in order to be able to apply and test all the needed patches from new EMI releases to better support HPC applications.

Table I summarizes main hardware, network and software characteristics of the participating sites.

V. Computational resources management

The management of the computational resources can be seen from different points of view:

(i) users need a flexible, detailed and transparent way to select the hardware environment for their jobs;
(ii) site administrators need allocation policies to maximize the exploitation of the computing resources;
(iii) points (i) and (ii) are even more complicated by the fact that the IGI is a general purpose platform shared among serial and parallel jobs. Since serial and parallel jobs coexist in the same environment, there is the need to have a way to manage job types, avoiding standard serial jobs being dispatched to parallel queues by the WMS matchmaking process.

In order to solve point (i), IGI already provides to the users a method to select the granularity of the allocated processors by means of some dedicated JDL attributes (see section III-B). The problem of the job type management (iii) has been faced in our activity and we explored two different solutions that are described in the next subsection.

A. *The* "parallel" *role*

The solution proposed at the Pisa site [13] was based on the use of VOMS services, which provides support for users Roles and Groups membership within a VO. In order to access HPC resources the ``Role=Parallel'' was introduced and has to be assigned to HPC users by the VO manager. Thus, the command to start a parallel session is obtained by adding the option ``-voms gridit:/gridit/Role=parallel'', to the ``voms-proxy-init'' command:

```
voms-proxy-init
     -voms gridit:/gridit/Role=parallel
```

This method is not user friendly since users have to generate different kind of VOMS credentials depending on the kind of jobs they need to execute, but the use of VOMS extensions (Role or Group) was the simplest way to implement this feature in the current Grid architecture. The problem of keeping generic jobs out of a given resource is common to other special resources, such as nodes equipped with GP-GPU cards or large amount of memory.

B. Scheduling Policies for HPC systems in a distributed environment

The deployment, management and TCO (Total Cost of Ownership) of large computing environments able to satisfy the needs of HPC applications always involve huge investments. The solution proposed at the University of Naples is based on the assumption that only an efficient and effective use of these systems can repay the investment made. However computational resources, if included in a distributed environment, have to meet the needs of all the users belonging to heterogeneous communities (HPC and not). In this scenario, where on the same resources run both the traditional jobs and some parallel applications, HPC communities are usually penalized by general-purpose scheduler configurations. For this reason, system administrators experimented a combination of some mechanisms provided by the Local Resources Manager/Scheduling System with the aims to gain a balanced, efficient and effective use of the computing resources by heterogeneous communities without neglecting the users satisfaction [14].

A combination of fairshare, reservation, preemption and backfill mechanisms has been used in a context where HPC applications have to be treated as *"privileged applications"*. In particular: (i) preemption helps concurrent (es. MPI based) applications in allocating at the same time a large number of resources (ii) fairshare guarantees a fair and balanced access to resources (iii) backfill helps to "maximize" the use of resources and finally (iv) reservation ensures resources availability (i.e. for certification job).

Moreover, the implemented solution automatically identifies, by means of some integrations of the middleware components, the type of application (sequential or parallel) without any burden to the users.

VI. Software environment: the Applicative Middleware

Scientific software applications are often based on an underlying layer of software tools, that we call *"applicative middleware"* (libraries, applications, problem solving environment, etc.) that provides the needed environment for applications effective and efficient execution on computational resources. The communities can use both local and distributed resources to run their applications. The use of local resources ensures the chance of a more direct control of software configuration to obtain the most suitable

Table I
SITES CHARACTERISTICS

SITE	Nodes	Core-RAM per node	OS -MW	Network	MPI flavour	Shared storage	Batch system
INFN PISA	128	8 - 8 GB	SL 5.X - EMI-1	Infiniband	OpenMPI	GPFS	LSF
UNI NAPOLI	8	8 - 8 GB	SL 5.X - EMI-1	Infiniband	OpenMPI	Lustre	PBS - Maui
INFN PARMA	8	8 - 12 GB	SL 5.X - EMI-1	GbEth	OpenMPI	NFS	PBS - Maui
UNI PERUGIA	8	2 - 2 GB	SL 5.X - EMI-1	2 x GbEth	OpenMPI/mpich2	NFS	PBS - Maui
IGI BOLOGNA	8	24 - 80 GB	SL 6.X - EMI-2	2 x GbEth	OpenMPI	glusterFS	PBS - Maui

environment to run. However local resources availability can be often limited. On the contrary users can choose to use distributed environments to have a greater amount of resources but they dont get the same level of control on such software layer. In particular they cannot easily:

- verify the presence of software on computing resources belonging to distributed infrastructures,
- get information about the available software (i.e. version, compiler used for software installation, which MPI implementation has been used to compile a certain MPI-based library, etc.),
- get information on software usage (documentation and examples of use).

In EGI context, the supported method to publish the presence of a specific software run-time environment on a *"grid site"* is based on the use of the `GlueHostApplicationSoftwareRunTimeEnvironment` TAGs. Users, specifying their requests in the JDL file, can select the most suitable site to run their applications in the "right" environment. For example, if a site offers the tool Quantum-Espresso v 4.3.1, the site can decide to publish a TAG like the following:

```
GlueHostApplicationSoftwareRunTimeEnvironment =
                                          espresso-4.3.1
```

The corresponding users JDL file needed to select this software is

```
Requirements = member (espresso-4.3.1,
 other.GlueHostApplicationSoftwareRunTimeEnvironment);
```

If no `Requirements` is specified no WMS matching will be performed and it will not be guaranteed that Quantum-Espresso is installed nor that it is available on the selected site. However we note that this is still a poor method due to the fact that nothing is said about the way used to install the package.

To provide further information to the user about the software environment we propose a combination of two approaches:

(i) the first method is to define a basic and common software environment, within all the sites belonging to a specific HPC community, through the CernVM File System (CernVM-FS) [15]. This is a network file system that aims to remove the need for local installation of software and, nevertheless to have a distributed infrastructure where all sites have the same common software environment, selected on the basis of the virtual organization to which the users belong to.

(ii) the second method is based on the definition of a standard to publish information about customized and optimized software layer available on a site. The proposed standard aims to improve the use of the `GlueHostApplicationSoftwareRunTimeEnvironment` TAGs to publish more details on the available software. In particular, the TAGs can publish information about interconnection topology, some hardware and software details (i.e. compiler version used to generate the tool, MPI distribution, etc.) and the URL where users can find documentation and use cases. Any features TAGs may be expressed by a TAG like the following:

```
GlueHostApplicationSoftwareRunTimeEnvironment
          = FEATURENAME_FEATUREVALUE
```

For instance, if the site offers several optimized installations of the tool Quantum-Espresso v 4.3.1, each of them generated by different compilers (i.e. Intel and GNU C Compilers), and documentations about the tool and its usage, the following TAGs may be published:

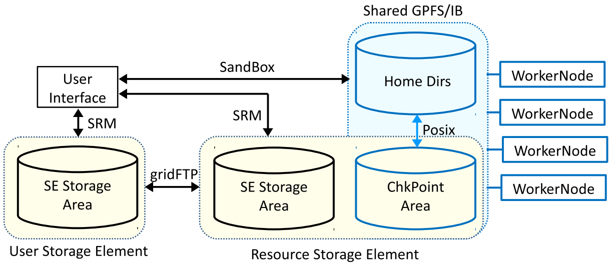
```
GlueHostApplicationSoftwareRunTimeEnvironment =
   SITESOFTWARE1_espresso-4.3.1
GlueHostApplicationSoftwareRunTimeEnvironment =
   SITESOFTWARE2_espresso-4.3.1
GlueHostApplicationSoftwareRunTimeEnvironment =
   SITESOFTWARE1COMPILER_icc-12.0
GlueHostApplicationSoftwareRunTimeEnvironment =
   SITESOFTWARE2COMPILER_gcc-4.1
GlueHostApplicationSoftwareRunTimeEnvironment =
   SITESOFTWARE1DOCURL_http://www.scope.unina.it/
   servizioutenti/docs/SIS_guida_librerie_v6.pdf
```

VII. DATA MANAGEMENT

In general, data management for HPC activities in the Grid environment has some important differences with respect to the model in use for serial jobs [16]:

(i) the storage architecture may be shared between Worker Nodes (WNs) and Storage Elements (SEs). In many cases SEs and WNs have a fast access (such as Infiniband technologies) to an enterprise class Storage Area Networks (SAN). This organization can

Figure 1. Cluster storage architecture



be particularly useful to ease the access to programs and common data. Beside this sharing, clusters may provide file systems local to the nodes, needed by data intensive applications. All our sites provide shared file-systems, with different solutions (and performance): GPGS/Infiniband (Pisa), Lustre (Napoli), NFS/GigabitEthernet (Perugia and Parma) and glusterfs/GigabitEthernet (Bologna).

(ii) Parallel jobs have typically a long execution time (several days), while elaborating large amount of datasets (more than 1GB). In the standard Grid architecture such jobs need to spend considerable amount of time uploading the input data-set from the users storage area to the WNs and, at the end of the calculation, from the WNs back to the users storage area. The data transfer involves typically a single process, while other allocated cores remain unused.

(iii) Moreover, parallel jobs often require a CPU time larger than the maximum time allowed by the queue, so the user, in principle, has to move checkpoint data-sets forward and backward several times.

To ease up these problems, at the Pisa cluster we have experimented the possibility that the same physical storage can be shared between WNs and Storage Elements. On the cluster a storage space (the *"chkPoint"* space) is shared for reading and writing between WNs and the Storage Element. The job work-flow on the cluster is the following:

- Input data-sets are uploaded from the users SE to the resource SE via GridFTP or, in case of small amount of data, they are shipped along with the InputSandbox. The running job writes outputs (standard output, standard error and output data) to the ChkPoint Area.
- At the job completion the user can retrieve the output files via the Storage Resource Mamagement (SRM) interface. In case of long term jobs the execution can continue on the same data-set through a subsequent submission performed on the same cluster (direct posix access) or on any other HPC resources since the dataset can be accessed via SRM [17].

VIII. The Applications suite

The Italian and European Grid infrastructures were initially created to satisfy the computational and storage requirements of the HEP community applications. Those applications are embarrassingly parallel codes, with the parallelism guaranteed by the independence of the particle events generated inside the experiments or simulated through Monte Carlo runs. The Grid evolution was driven by this use case but making the infrastructure available to non-HEP communities requires to support more general parallel applications. In particular support to "not high-end" HPC applications has to be added to the infrastructure, "high-end" HPC applications are still prerogative of other distributed infrastructure (i.e. PRACE [18]).

Parallel applications used by different communities have very different requirements in terms of the parallel paradigm used, i.e. shared vs distributed memory, type of parallel programming used, input and output data size, memory footprint and duration of a typical production run. Given the broad range of different requirements, we selected some use cases from different disciplines (Astronomy, Relativistic Astrophysics, Earth Science, Quantum Chemistry, Molecular Dynamics) with the aim to build a suite of applications, representative of resources requirements by new and old IGI HPC users, useful to validate and evaluate all the solutions implemented by the testbed: computational resources management, data management, software environment, etc..

The suite is composed by the following applications: Gaia Mission data analysis (§VIII-A), the Einstein toolkit (§VIII-B), the Nemo Ocean Modelling Framework (§VIII-C), NAMD (§VIII-D), Quantum Chemistry by Quantum ESPRESSO code (§VIII-E), Regional Climate model RegCM4 (§VIII-F). The applications suite covers different requirements: NAMD and RegCM require very long runs and checkpointing; NEMO and GAIA require high availability of RAM; the Einstein toolkit implements a hybrid (MPI/OpenMP) parallelisation paradigm and requires big storage capacity for the output data (this also apply to the RegCM package); shared storage among the processors is required by the NEMO application that also need large amount of RAM; Quantum Espresso requires the availability of an optimized software environment in order to be efficiently run.

A. Astronomy: The Gaia Mission

The parallel application is for the development and test of the core part of the AVU-GSR (Astrometric Verification Unit - Global Sphere Reconstruction) software developed for the ESA Gaia Mission [19]. The main goal of this mission is the production of a microarcsecond-level 5 parameters astrometric catalog - i.e. including positions, parallaxes and the two components of the proper motions - of about 1 billion stars of our Galaxy, by means of high-precision astrometric measurements conducted by a satellite sweeping continuously the celestial sphere during its 5-years mission.

The memory request to solve the AVU-GSR module depends on the number of stars, the number of observations and the number of computing nodes available in the system. During the mission, the code will be used in a range of 300,000 to 50 million stars at most. The estimated memory requirements are between 5 GB up to 8 TByte of RAM. The parallel code uses MPI and openMP (where available) is characterized by an extremely low communication level between the processes, so that preliminary speed-up tests show a behavior close to the theoretical speed-up. A complete code description is given in [20].

Since AVU-GSR is very demanding on hardware resources, the typical execution environment is provided by Supercomputers, but the resources provided by IGI are very attractive for debugging purpose and to explore the simulation behaviour for a limited number of stars.

The porting on the EGI is in progress in the framework of the IGI HPC testbed in which we select resources with a large amount of global memory and a high speed network, such as the one provided by the INFN-PISA and UNI NAPOLI sites.

B. *Relativistic Astrophysics: The Einstein toolkit*

The Einstein Toolkit [21] is an open software that provide the core computational tools needed by relativistic astrophysics, i.e., to solve the Einstein's equations coupled to matter and magnetic fields. In practice, the toolkit solves time-dependent partial differential equations on mesh refined three-dimensional grids. The code has been parallelized using MPI/OpenMP and is actually in production with simulation involving up to 256 cores on the PISA site.

Another special characteristic of the application is that it needs checkpoints and it is often used writing large data-output (order 10 GBytes). The production usage of this application in the Grid environment depends on the presence of the data management infrastructure described in section 7 at the target site and, as a consequence, of targeted submissions. However it has been possible to test the submission and execution process of small jobs on the whole testbed. The tests involved the evolution of a stable general relativistic TOV-Star model on a cubic multi-grid mesh with five levels of refinement (each of local size 40x40x40) and performing 800 time evolution steps.

C. *Oceanography: The Nemo project*

NEMO [22] is an ocean modelling framework which is composed of "engines" nested in an "environment". The "engines" provide numerical solutions of ocean, sea-ice, tracers and biochemistry equations and their related physics. The "environment" consists of the pre- and post-processing tools, the interface to the other components of the Earth System, the user interface, the computer dependent functions and the documentation of the system.

Table II
TOTAL EXECUTION TIME (IN MINUTES) OF THE SAME SIMULATION ON THE DIFFERENT SITES OF THE TESTBED AND DIFFERENT GRANULARITY SETTINGS (SEE SECTION III-B).

SITE	#nodes	1	2	4	8	16
UNI Napoli	psocket	131,4	104,9	68,1		
	pcore	124,8	117,5	82,3		
	pnode	148,8	124,2	108,1		
INFN Parma	psocket	115,1	104,3	104,3		
	pcore	129,6	221,1	174,9		
	pnode	120,6	108,2	85,8		
INFN Pisa	psocket	129,3	78,9	59,8	46,6	43,3
	pcore	127,2	79,6	57,6	49,1	47,2
	pnode	172,5	123,5	81,1	66,3	52,9

A specific NEMO based implementation on the Mediterranean Sea, carried out from the latest development activities [23] has been ported in the Grid environment in order to run parallel calculations. Such implementation requires a distributed memory multiprocessor architecture with a shared file system, and has significant CPU and memory demand (from our calculations we estimated 1GB/core for a 8 cores simulation), hence its porting was made possible thanks to the increased memory size and CPU multi core architecture recently available on the production Grid infrastructure. After the porting, the application can be used for production as well as for testing purposes by modifying the model (this means recompiling the source code) or varying the input parameters. The user community was interested in evaluate possible benefits in the specific case of the operational data production service (first use case), and in exploiting the Grid for testing and tuning the model (second use case) which implies that the package must be executed several times in a parameter sweeping approach.

D. *Molecular Dynamics: NAMD*

NAMD [24] is a powerful parallel Molecular Mechanics(MM)/Molecular Dynamics(MD) code particularly suited for the study of large biomolecules. However, it is also compatible with different force fields, making possible the simulation of systems of quite different characteristics. NAMD can be efficiently used on large multi-core platforms and clusters. The NAMD use case was a simulation of a 36000 atoms lipid provided by a CNR-ISOF group located in Bologna. To have a real-life use case the simulation had to be run for at least 25 nanoseconds of simulated time resulting on a wallclock time of about 40 days if run on a 8 cores machine. The use case is described in greater details in [25].

To run NAMD in a Grid environment, the whole application was rebuilt on Scientific Linux 5 with OpenMPI libraries linked dynamically. Sites supporting the MPI-Start framework and OpenMPI were selected to run the jobs through JDL requirements. The porting was challenging for two main reasons:

(i) a data management strategy was needed because we had to make available the output files to the ISOF researchers and the size of the output could not be easily handled via the WMS. This was obtained through "pre-run" and "post-run" scripts, both enabled via MPI-Start.

(ii) the length of the simulation implied many computation checkpoints given the time limits on the batch system queues of the sites matching the requirements. We decided to split the simulation in 50 steps each 500 ps of simulated time long, allowing to complete each step without reaching the queues time limits.

The HPC testbed is a highly controlled environment dedicated to this tests so the failure rates and the "waiting on queue" times due to concurrency with other users were very limited. However, we showed that also even long parallel NAMD simulations can be run on Grid exploiting a checkpoint strategy if resources similar to those available on the HPC testbed are available in the infrastructure.

E. Quantum Chemistry: Quantum ESPRESSO

QUANTUM ESPRESSO (Q/E) is an integrated suite of computer codes for electronic-structure calculations and materials modeling, based on density-functional theory [26].

The suite contains several heterogeneous codes with a wide range of simulation techniques in the area of quantum simulation for material science. Typical CPU and memory requirements for Q/E vary by orders of magnitude depending on the type of system and on the calculated physical property, but in general, both CPU and memory usage quickly increase with the number of atoms simulated. Only tightly-coupled MPI parallelization with memory distribution across processors allows to solve large problems, i.e. systems requiring a large number of atoms. The resulting MPI programs which composes the Q/E suite need fast communications and low latency requires the need to access via Grid HPC cluster resources. Our goal here is to check and evaluate which kind of highly intensive parallel production runs can be done on the top of the IGI MPI infrastructure.

The porting procedure of the HPC examples in the HPC Grid testbed was done on Pisa site where all the needed software stack was manually installed. Once this process was completed users were able to run the code and scientific production can be started. The presence of the check-point area available on the cluster (as discussed in section VII) allow to run long runs without problems and start real production by users.

F. RegCM

RegCM is the first limited area model developed for long term regional climate simulation currently being developed at the Abdus Salam International Centre for Theoretical Physics (ICTP), Trieste, Italy and version 4 is described in [27]. RegCM4 is a regional climate model based on the concept of one-way nesting, in which large scale meteorological fields from a Global Circulation Model (GCM) run provides initial and time-dependent meteorological boundary conditions for high resolution simulations on a specific region. The RegCM4 computational engine is CPU intensive and based on MPI.

Standard climate RegCM simulations require large dataset (ranging from a few gigabyte for small region up to hundreds of Gigabytes for the largest ones) to be downloaded on Grid and transferred back and forth several times during the model execution. There are however other kinds of computational experiments that can be conducted in a Grid environment: validation runs. This experiment requires to run many different short simulations with different initial conditions. This mixed HTC/HPC approach could be efficiently done on multiple SMP resources made available by Grid resources.

We therefore provide the possibility to run RegCM (or any other MPI parallel application actually) through a "relocatable package" approach. With this approach all the software needed, starting from an essential OpenMPI distribution is moved to computational resources by the job. All the libraries needed by the program have to be precompiled elsewhere and packaged for easy deployability on any architecture the job will land on. The main advantage of this solution is that it will run on almost every machine available on the Grid and the user will not even need to know what the GRID will have assigned to him. The code itself will need to be compiled with the same "relocatable" libraries and shipped to the computational resources by the job.

This alternative approach allows a user to run a small RegCM simulation on any kind of SMP resource available to her, quite widely available nowadays. The main drawback of this solution is that a precompiled MPI distribution will not take advantage of any high speed network available and will not be generally able to use more than one computing node.

IX. Conclusion and future works

In this paper we presented the work performed in the IGI context to deploy a grid testbed to validate and improve the support for "not high-end" HPC applications. A suite of applications from several disciplines, and representative of different requirements, has been built to test the readiness to run parallel applications of both the infrastructure and the middleware. These applications have been used to validate and evaluate all the solutions implemented on the testbed.

During the validation phase, challenges for users, site administrators and grid managers have been analysed and addressed in particular for what concerns the jobs scheduling policies, data management and the availability of a suitable software environment. If the validation phase is almost completed, the evaluation stage has just begun.

Specific resource allocation and data management policies permit now all the applications to benefit from the availability of a distributed infrastructure of HPC resources instead of a local cluster. However, moving from a dedicated testbed to the production environment could introduce issues that were not faced at validation time.

The first evaluation phase showed that all the applications have been satisfactorily executed on the testbed. However efficiency and portability may be increased by a more conscious use of the underlying Applicative Middleware layer.

HPC support on IGI/EGI Grid for more complex use cases (i.e. workflows-based applications) is not always straightforward, thus the consultancy of Grid and applications experts is often needed to exploit all middleware and software features.

To address the difficulties for end users, IGI is studying and developing a customizable web portal that will act as a Grid front-end also to run complex applications; this activity will be presented in future publications.

Acknowledgment

The research leading to the results presented in this paper has been possible thanks to the Grid resources and services provided by the European Grid Infrastructure (EGI) and the Italian Grid Infrastructure (IGI).

References

[1] *European Grid Infrastructure - An Integrated Sustainable Pan-European Infrastructure for Researchers in Europe (EGI-InSPIRE)*, White Paper, 18 April 2011, https://documents.egi.eu/document/201

[2] Streit, A., et al., *UNICORE 6 - Recent and Future Advancements*, Annales des Tlcommunications 65(11-12), 757-762 (2010).

[3] *The EGI MPI Virtual Team*, https://wiki.egi.eu/wiki/VT_MPI_within_EGI

[4] *The Italian Grid Infrastructure*, http://www.italiangrid.it

[5] *The DATAGRID Project*, http://eu-datagrid.web.cern.ch/eu-datagrid/

[6] Ferrari T., Gaido, L., *Resources and Services of the EGEE Production Infrastructure*, Journal of Grid computing, Volume 9, Number 2, pp. 119-133 (2011).

[7] Vanneschi M., *The Grid.it Project*, Workshop on ERA and GridResearch, Brussels, 17 July 2003, ftp://ftp.cordis.europa.eu/pub/ist/docs/grids_era_workshop_it1.pdf

[8] *The INFNGRID Project*, http://server11.infn.it/grid/doc/

[9] Laure, E. at al., *Programming the Grid with gLite*, Computational Methods In Science and Technology 12(1), pp. 33-45 (2006).

[10] Fuhrmann, P., *EMI, the Introduction*, in Proceeding of CHEP 2010, Taipei, 18 October 2010.

[11] Engelberts, J., *Towards a robust and user friendly MPI functionality on the EGEE Grid*, EGEE User Forum 2010, Uppsala, 12-15 April 2010.

[12] Dichev, K., Stork, S., Keller R., Fernandez, E., *MPI Support on the Grid*, Computing and Informatics, vol 27, No 2 (2008).

[13] Alfieri, R., Arezzini, S., Ciampa, A., De Pietri, R., Mazzoni, E., *HPC on the Grid: The Theophys Experience*, J. Grid Computing (online first), DOI 10.1007/s10723-012-9223-6 (2012).

[14] Barone, G.B., Boccia, V., Bottalico, D., Carracciuolo, L., Doria, A., Laccetti, G., *Modelling the behaviour of an Adaptive Scheduling Controller*, Proceedings of the Sixth International Conference on Complex, Intelligent, and Software Intensive Systems, pp. 438-442 (2012).

[15] Blomer. J., Buncic, P., *CernVM-FS: delivering scientific software to globally distributed computing resources*, Proceedings of the First International Workshop on Network-aware Data Management, pp 49-56 (2011).

[16] Frohner, A., et al., *Data Management in EGEE*, Journal of Physics, Conference Series (2010).

[17] *The Storage Resource Manager Interface Specification*, https://sdm.lbl.gov/srm-wg/doc/SRM.v2.2.html

[18] Berg, A., *PRACE Distributed Infrastructure Services and Evolution* EGI Community Forum 2012 (2012).

[19] Perryman, M. A. C. et al., *GAIA: Composition, formation and evolution of the Galaxy* , Astronomy and Astrophysics, v.369, p.339-363 (2001).

[20] Bandieramonte, M., et al., *AVU-GSR Gaia Mission. An hybrid solution for HPC and Grid-MPI infrastructures* 21th IEEE International Workshop on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE), (2012).

[21] Loffler, F., et al., *The Einstein Toolkit: A Community Computational Infrastructure for Relativistic Astrophysicsxi*. Classical and Quantum Gravity, 29(11), (2012).

[22] Madec, G., *NEMO ocean engine*. Note du Pole de modlisation, Institut Pierre-Simon Laplace (IPSL), France, No 27 ISSN No 1288-1619, (2008).

[23] MyOcean MFS http://gnoo.bo.ingv.it/mfs/myocean

[24] *The NAMD Package*, http://www.ks.uiuc.edu/Research/namd/

[25] Venturini, A., Zerbetto, F., *Dynamics of a lipid bilayer induced by electric fields*; Phys. Chem. Chem. Phys., 13. (2011).

[26] Giannozzi, P., et al., *QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials*, J. Phys. Condens. Matter (2009).

[27] Giorgi, F., et al., *RegCM4: Model description and preliminary tests over multiple CORDEX domains*, Climate Research, 52, pp. 7-29. (2012).