D-SA1.2 – VERCE platform: First operation and monitoring report

01/10/2012

Project acronym:	VERCE
Project n°:	283543
Funding Scheme:	Combination of CP & CSA
Call Identifier:	FP7-INFRASTRUCTURES-2011-2
WP:	WP5/SA1, Management and operation of the research platform
Filename:	D-SA1.2.pdf
Authors[1]*:*	G. Ferini, A. Gemuend, G. Moguilny, H. Schwichtenberg and D. Weissenbach
Location:	http://www.verce.eu/Repository/Deliverables/RP2/
Type of document:	Deliverable
Dissemination level:	Public
Status:	Draft
Due date of delivery:	01/10/2012
Reviewers:	S. H. Leong (LRZ)
Keywords:	VERCE resources, HPC, Grid, Community management, Virtual Organisation

History	*Author*		*Date*	*Comments*
1	G. Moguilny	(IPGP)	11/09/2012	Initial Draft
2	A. Gemünd	(SCAI)	12/09/2012	VO verce.eu
3	G. Ferini	(CINECA)	13/09/2012	Deployment description
4	G. Ferini	(CINECA)	19/09/2012	Scalasca monitoring tool
5	D. Weissenbach	(IPGP)	19/09/2012	GridFTP
6	G. Ferini	(CINECA)	19/09/2012	SeisSol and ObsPy deployment
7	A. Gemünd	(SCAI)	20/09/2012	Inca monitoring tool
8	A. Gemünd	(SCAI)	20/09/2012	More about ObsPy deployment
9	A. Gemünd	(SCAI)	20/09/2012	Modifications after internal review

[1] Alphabetical order

Copyright notice

VERCE, *Virtual Earthquake and seismology Research Community e-science environment in Europe*, is a project co-funded by the European Commission as an Integrated Infrastructure Initiative within the 7th Framework Programme. VERCE began in October 2011 and will run for 4 years.

Contents

List of Figures

List of Tables

Executive summary

The main objectives of the WP5/SA1 work package are to:

- Operate and manage in coordination with SA3 a Virtual Organisation.
- Provide tools and services for a unified access to the VERCE research platform combining Data, Grid and HPC resources.
- Provide and manage the distributed research platform.
- Provide and manage a set of application-tailored workbenches and enactment gateways, in coordination with SA2, to support specific use-case scenarios.
- Provide and manage a development testbed on which SA2 and JRA2 develop, integrate and evaluate the next releases of the VERCE platform.

In the first reporting period [Month 1 - Month 6] information about the resources that the VERCE consortium provided for the first testbed was collected and processed.

The midterm period [Month 6 - Month 12] allowed SA1 to move forward on mainly three important tasks: resources documentation, software deployment and creation of a virtual organisation.

For the resource documentation, to have a clear view on what is acessible on each resource of the testbed, ID cards have been prepared and are now available on the collaborative platform of the project. They contain at least the responsible contact person, the description and how to access the resource (section1).

Concerning the software deployment, SA2 performed evaluation results and published their results at the beginning of August: one seismological Code (SeisSol), and two libraries/tools (ObsPy, GridFTP) have been validated. For each of these components, a wiki page has been created to gather useful information (Contacts, History, Installation documentation, User documentation..) and installations are being performed (subsection 1.3).

Usage of GridFTP for data transfer between VERCE resources has been validated on a subset of resources by a Data movement campaign. The results of this campaign are described in 1.3.3.

SA1 has set up the Virtual Organisation `verce.eu` in the framework of EGI and the first registered users are using it to test seismologic tools on the Grid resources (section 2).

Concerning monitoring, this deliverable presents two tools, Inca (section 3) for user-level and Scalasca (subsection 1.3.1) for performance monitoring.

1 Platform

1.1 Updated testbed

The first computing resources available to the VERCE platform are described page 24 in the Deliverable[2] of the first Reporting Period.

Main changes are:

- a new Grid resource, GRI-LRZ-02, is being integrated to the VERCE platform. It differs from the other Grid clusters in that it will be provding access through the Globus middleware instead of through gLite.
- DEP-UEDIN-05 is now part of DEP-UEDIN-01.

The updated testbed is described in the Appendix page 17.

1.2 Contacts

The list of task leaders and the contacts for each resource have been established.

Task leaders

1	Integrate and operate the distributed data and computational resources	CNRS
2	Operate and manage the platform	SCAI
3	Support the deployment of applications and use cases	CINECA
4	Supply technical documentation	CNRS

Tesbed representatives / site

- CINECA: g.ferini@cineca.it
  HPC-CINECA-01
  HPC-CINECA-02
- INGV: daniele.melini@ingv.it
  DEP-INGV-01
- IPGP: weissenb@ipgp.fr
  DEP-IPGP-01
  DEP-IPGP-02
  GRI-IPGP-03
  DAT-IPGP-04
- LRZ: Siew-Hoon.Leong@lrz.de
  HPC-LRZ-01
  GRI-LRZ-02 (new but not operational)
- UEDIN: iraklis.klampanos@ed.ac.uk
  DEP-UEDIN-01
  GAT-UEDIN-02
  GAT-UEDIN-03
  GAT-UEDIN-04

[2] http://www.verce.eu/Repository/Deliverables/RP1/D-SA1.1.pdf

- ULIV: Xiao.Wang@liverpool.ac.uk
  DEP-ULIV-01
- SCAI: horst.schwichtenberg@scai.fraunhofer.de
  GRI-SCAI-01
- KNMI/ORFEUS: luca.trani@knmi.nl, spinuso@knmi.nl
  DAT-KNMI-01

1.3 Software deployment

SA1 coordinates the deployment of the first software release, consisting of the components recommended after validation by SA2. The results of the evaluation performed by SA2 for the second reporting period and published at the beginning of August are illustrated in Table 1.

Table 1 – RP2 software evaluation.

HPC application codes	
Accepted	SeisSol
Not Ready	AxiSEM, SpecFEM3D, Ses3D, Sec3D
Libraries/Tools	
Accepted	ObsPy, GridFTP
Not Ready	Admire/Dispel, OGSA-DAI

According to the technical features of the resources available in the VERCE testbed, the expected role within the platform and intrinsic characteristics/requirements of the software to be deployed, each component evaluated by SA2 can be grouped in the following categories:

Recommended: it means that a component performs the best on a given part of the platform (e.g. HPC sites);

Compatible: the component can work on other systems, but with limitations (e.g. in performance or in the size of physical problem);

Incompatible: the component is not suitable for the given resource and/or it is not expected to be provided on it.

For instance, application codes are incompatible with resources in the testbed which are not computing facilities but only data repositories (e.g. DAT-KNMI-01, DAT-IPGP-04). If exhibiting good scalability and performance on several hundreds/thousands of cores (e.g. SeisSol, SpecFEM3D), application codes are recommended on HPC sites. Additionally, such application codes can also be compatible with Grid and departmental resources. On the other hand, frameworks and processing tools such as ObsPy, are more suitable for departmental resources as compared to HPC resources. Poorly scalable applications are incompatible with HPC systems and are thus recommended only on Grid and departmental resources.

SA1 deploys each accepted component on both recommended and compatible resources in the testbed. The procedure of deployment aims at providing the users with tested and documented components available on the VERCE platform. As a first step, each component is "installed", which means that the component is available for specific users (e.g. in a local path, or only accessible with specific permissions). Installation is performed by following indications provided by SA2 derived from testing and evaluation. In case of specific issues related to the system, these are fixed and documented. After installation is completed, the component is made available to general users. This also implies that documentation is provided on both the installation/deployment procedure (technical doc) and on the component features/capabilities (scientific doc).

A component in such a status is hence labelled as "operational", i.e. ready to use.

The state of the art is shown in Table 2. Notice that, according to what explained above, possible statuses for each component are: To install, Being installed, Installed, Operational.

As explained in the DSA2.2, three components (SeisSol, ObsPy and GridFTP) have been approved for deployment. The release of remaining components is delayed due to several issues. In particular, some application codes (AxiSEM, Ses3D) need re-compilation when changing the scientific problem size and/or parameters and/or the number of processors to be used, i.e. the computational configuration. Such a feature clearly prevents the deployment of these applications as services that can be used for generic use cases. Nevertheless, SA1 deployed "demo-versions" of a couple of codes on resources of the testbed in order to:

- collaborate with SA2 in testing these applications;
- provide viable solutions to fix known issues;
- allow preliminary checks and comparison of the results on different systems.

The situation is slightly different for SpecFEM3D. As explained in detail in DSA2.2 and as outlined during discussion with NA2, this application code does not suffer from the limitations as AxiSEM and Ses3D, in the sense that many usual scenarios can be simulated without changing some "hard-coded" input parameters. As a consequence, the code fulfils basic requirements to be deployed as a "service for general use cases" and is already available on some of the systems in the testbed. However, some other issues mainly concerning memory management need to be fixed, in close collaboration with both the SpecFEM3D development team and SA2, before completing deployment on other systems.

For components in operational status, test users are identified and will allow a first step in evaluating and assessing reliability of the procedure and clarity/completeness of documentation.

For each of components recommended by SA2, a wiki page has been created to gather useful information such as:

- Contacts
- History (version number)
- Installation documentation
- User documentation
- Test user(s)

1.3.1 SeisSol

SeisSol application code has been deployed on all HPC resources (HPC-LRZ-01, HPC-CINECA-02) in the testbed and it is currently available to users as a "module". Simulations can be run by authorised users to by following a few simple steps:

1. login on the system;
2. move to the working area and copy there all the input files;
3. submit a job (example scripts are provided through the "help" of the module).

In addition, an automatic procedure to extract basic profiling information has been implemented on HPC-CINECA-02. Though it relies on Scalasca profiler to retrieve the relevant metrics, the whole procedure will be transparent to the final users, who will simply obtain textual reports in their working directory.

Performance monitoring: Scalasca

Scalasca (SCalable performance Analysis of LArge SCale Applications) is a toolset for performance analysis of parallel applications, like SeisSol. It is developed by Julich Supercomputer Centre, can be

Table 2 – Software deployment on the VERCE platform.

		TestBed											
		HPC-LRZ-01	HPC-CINECA-02	DEP-UEDIN-01	DEP-ULIV-01	DEP-IPGP-02	DEP-IPGP-01	DEP-INGV-01	GRI-SCAI-01	GRI-IPGP-03	GRI-LRZ-02	DAT-KNMI-01	DAT-IPGP-04
	Codes												
►	SeisSol	✓	✓					◒	◒		◒		
	SpecFEM3D	●	●										
	Sec3D	●	●						●				
	Ses3D		●										
	AxiSEM	●	●						◒				
	Scientific libraries												
►	ObsPy	●	✓	✓	○	◒	●	◒	✓	◒	◒		
	Whisper suite												
	Environment / Tools												
►	GridFTP	✓	✓	●	○	○	○	●	✓	✓	●	●	○
	Admire/Dispel			◒	◒								
	OGSA-DAI			◒	◒								

► Validated/recommended by SA2,
○ To be installed,
◒ Currently being installed,
● Installed,
✓ Available for VERCE users
An empty cell means not applicable or undecided as of yet

```
[cin0753a@node342 ~]$ module help scalasca-utils
----------------------------------------------------------------------
Module Specific Help for /cineca/prod/modulefiles/advanced/tools/scalasca-utils/0:

modulefile "scalasca-utils/0"

scalasca-utils-0
A set of wrappers to scalasca commands to analyse applications and extract summarized information
from the scalasca report.

-------------------------------------------------------------------------------
License type: ---
Web site:     ---
Download url: ---
-------------------------------------------------------------------------------

The package includes the following commands:

a) mpirun-scalasca   : replaces mpirun for applications instrumented for scalasca measurements;
   Syntax: mpirun-scalasca -np <NPROC> <exe>

b) scalasca-report-full : creates a text file with information about time (in seconds) spent
                          in each subroutine/function of the application.
                          All application function call are labelled according to type:
                          MPI - refers to function calls to MPI library
                          OMP - refers to OpenMP regions or calls to the OpenMP API
                          COM - refers to user-program routines on paths that directly or
                                indirectly call MPI or OpenMP
                          USR - refers to user-program routines involved with purely local computation
   Syntax: scalasca-report-full <path-to-the-"epik*"dir>

c) scalasca-report-mpi  : creates a text file with information about time (in seconds) spent
                          in MPI type function calls
   Syntax: scalasca-report-mpi <path-to-the-"epik*"dir>

d) scalasca-report-omp  : creates a text file with information about time (in seconds) spent
                          in OMP type function calls
   Syntax: scalasca-report-omp <path-to-the-"epik*"dir>

e) scalasca-report-usr  : creates a text file with information about time (in seconds) spent
                          in USR type function calls
   Syntax: scalasca-report-usr <path-to-the-"epik*"dir>

f) scalasca-report-all  : creates two text files with information about time (in seconds) spent
                          in USR- and in MPI- type function calls respectively.
   Syntax: scalasca-report-all <path-to-the-"epik*"dir>
```

Figure 1 – `module help scalasca-utils` output

easily installed on a wide variety of systems and is able to manage MPI, OpenMP, MPI+OpenMP application codes. Scalasca provides detailed information not only about the time spent by each part of the code in a given subroutine (computational profiling), but also about the communication patterns and timing and the load balancing, generating reports that can be interactively explored in a graphical browser. It is therefore extremely useful in spotting bottlenecks in the code and improving its performance, especially when increasing the number of tasks and/or threads and, as a consequence, the weight of communication on the total computing time. Such an instrument can be therefore profitably used before large seismological simulations, in some explorative test runs, to exploit the best computational configuration for a given use case. It is indeed known for seismic application codes and for geophysics ones in general that performance can be markedly affected, for a given use case, by the domain decomposition and hence by the chosen number of tasks.

On one HPC resource in the testbed, PLX system (HPC-CINECA-02), a procedure has been implemented that allows users to get a textual version of the Scalasca reports without requiring any knowledge of the toolset itself. In particular, for accepted application codes (currently SeisSol), two different versions have been deployed: the standard one and a second one already instrumented for Scalasca measurement. Users can choose between these two versions by loading the related module, e.g. by typing `module load SeisSol` or `module load SeisSol-scalasca`. In the latter case, the `module help` command provides them an example script for job submission with ad-hoc commands: `mpirun-scalasca` and `scalasca-report-option`. `mpirun-scalasca` is a command that

replaces standard launching command for parallel applications (mpirun) in order to activate performance analysis with Scalasca. `Scalasca-report-`*option* is the command that generates textual reports from the profiling and -*option* (whose value can be `full, usr, mpi, omp`) allows user to choose among different reports (e.g. for each type of function call in the application code, for computational parts only, for mpi functions only, and so on). Both these commands are part of a specifically created module, called `scalasca-utils` and their meaning and usage is shown through the `help` of the module (the corresponding output is shown in the figure 1).

Anyway, also the complete Scalasca report is kept and can be graphically explored by more skilled users.

As a side note, we mention that this procedure is completely independent on the application code and can therefore be extended to other ones. Currently, it has been already applied and tested for the latest release of SpecFEM3D, which, as SeisSol, is available as a module on PLX (HPC-CINECA-02) both in the standard version and in the version instrumented for Scalasca measurement (i.e. module SpecFEM3d and module SpecFEM3d-scalasca, respectively).

1.3.2 ObsPy

The ObsPy library is a Python framework for processing seismological data that provides typical processing routines, parsers for common file formats and other tools for seismologists.

Following suggestions from SA2, all ObsPy modules (with related dependencies, as listed in DSA2.2) are being installed on all the resources within the testbed and are presently "operational" in two systems (DEP-UEDIN-01 and HPC-CINECA-02).

The full ObsPy toolkit has a variety of binary dependencies, such as C libraries for data formats or maths, which makes deployment on the heterogeneous resources nontrivial. The EGI Grid resources often use RHEL5 compatible distributions, Scientific Linux. In these distributions, the system Python is version 2.4.x, which is too old for ObsPy to function correctly. This means that on such systems, an additional Python installation is required. Using the `virtualenv` system, however, it is possible to maintain arbitrary user-controlled Python environments without system permissions. Beyond that, ObsPy depends on a BLAS and LAPACK implementation and various other C libraries. To keep a consistent environment on the different Grid resources while still providing acceptable performance, GotoBLAS is also compiled and installed in this user tree. SA1 has started to create scripts for this procedure which are currently being tested and extended for different resources.

On HPC resources, ObsPy is recommended to be installed as a module. This is what has been done on HPC-CINECA-02, where ObsPy and all its dependencies as well (including Python) have been installed as modules. In such a way, ObsPy can be bound, during installation, to a given version of the dependencies (currently the following versions are used: Python2.7.2, numpy1.6.2, scipy0.10.1, matplotlib1.1.0, lxml2.3.4, ipython0.13, distribute0.6.28, with numpy and scipy relying on the optimised libraries MKL and ESSL).

1.3.3 GridFTP

The widely used GridFTP protocol has been chosen as the primary tool for data transfer in VERCE. It is a mandatory component of the VERCE infrastructure in the sense that a well integrated GridFTP server should be operational at most of the facilities. Due to its popularity and wide acceptance, it was already available on most resources of the testbed. However, regarding VERCE, additional requirements exist. To be fully optimal for VERCE, such a server must be accessible from the resource service, compute and/or data nodes with at least one of the following methods:

1. direct local access to client commands and APIs;

2. sending commands to a local third party interface implementing previously enabled single sign on access and shared filesystem.

To be able to effectively exchange data between the different resource centers in VERCE, the GridFTP server should also accept transfers incoming from and outgoing to the other VERCE servers. Although this is an issue of network configuration, it will be accounted for in the deployment state of the GridFTP component. In case a partner provides multiple GridFTP servers, for simplicity it is suggested to assign and name one instance to be used by VERCE.

Data movement campaign

To validate the availability and functionality of the essential aspects of GridFTP for VERCE, a data movement campaign is being carried out. The first step taken in this direction has actually been server integration and preliminary testing. The challenge involved five VERCE resources, 2 supercomputer (HPC-LRZ-01, HPC-CINECA-02), 2 GridFTP servers part of the EGI Grid (GRI-SCAI-01 and GRI-IPGP-03), and 1 departmental resource (DEP-UEDIN-01).

Table 3 shows an overview of the results of the first preliminary tests. Transfers were mostly executed with a single GridFTP data channel, and transfer rate eventually increase when several data channels are used. The file size was only between 20 and 100 MB size. However, the determination of a maximum transfer rate was not a goal of these tests. Also many factors make this value highly volatile due to server and network concurrent activities. As such effective figures are better described by mean and extreme values.

Table 3 – GridFTP tests (MB/s), FwD: Firewall blocking gridftp Data ports, Auth: problematic authentication/authorization.

From \ **To**	HPC-LRZ-01	DEP-UEDIN-01	GRI-IPGP-03	GRI-SCAI-01	HPC-CINECA-02
HPC-LRZ-01		FwD	FwD	3.6	Auth
DEP-UEDIN-01	5		5.8	2.5	Auth
GRI-IPGP-03	1.6	4		1.5	3.2
GRI-SCAI-01	5	2.8	3.5		8.7
HPC-CINECA-02	Auth	Auth	3.5	9.8	

Some challenges were faced during the integration and tests:

- restrictive security policies (esp. regarding network connectivity) require additional preparation (e.g. requests for acceptance of firewall rules);
- different configuration settings of GridFTP servers and clients; including a.o. specification of paths that are available for external access, paths that can be used for sharing files in groups of users on the same resource, etc.;

- more generally, evaluating and fine-tuning different configuration settings as well as debugging issues remotely in coordination with the local system operators is costly in terms of time, so SA1 intends to set up clearer procedures and communication channels for this.

The tests did not involve the aspect of user sign up, authentication and authorization. As such, valid X.509 credentials were presupposed. The integration with the `verce.eu` Virtual Organisation on EGI resources is an additional aspect for the future.

The tests showed that precise documentation of the systems is highly important. Information such as the network preconditions, used TCP ports and filesystem structure should be easily and centrally available for all sites. The set of scripts used for the campaign can be used with small files as a good way of monitoring the validity of the known platform state.

2 VO verce.eu

Users of EGI are organised into Virtual Organisations (VO). A VO is a group of people (typically application scientists and application developers) who share similar interests and have similar goals and who need to work collaboratively and/or need to share resources (e.g. data, software, expertise, CPU, storage space) through a grid infrastructure regardless of their geographical location.[1]

As planned, VERCE has created its own Virtual Organisation for EGI `verce.eu`.

Detailed information about the requirements and course of actions of the VO registration in EGI can be found in the official document: VO Registration Process in EGI[1]. The project hosts its own gLite VOMS server on `verce-voms.scai.fraunhofer.de`. The registration process has been passed and the VO is operational since 1st August 2012.

- VO Name: `verce.eu`
- Supported middleware: starting with gLite
- Contacts (Manager, Operations, Security, User Support, VO Users): vomembers@verce.eu and vomgt@verce.eu
- ID-card: http://operations-portal.egi.eu/vo/view/voname/verce.eu

GRI-SCAI-01 and GRI-IPGP-03 are now accepting `verce.eu` members.

The VO will be gradually augmented e.g. eventually through a dedicated dashboard, VO monitoring, or additional deployed software.

2.1 User Registration

Interested users in possession of a personal Grid certificate that is accepted by EUGridPMA can already join the `verce.eu` VO. To get a certificate, users have to contact their local grid Certification Authority, that can be found on the EUGridPMA homepage. Users from non-European countries can find their local authority through the International Grid Trust Federation webpage. For VERCE users, a list with additional guidelines has been made available on the project's internal website (Redmine).

With a valid EUGridPMA certificate imported in the browser, users have to visit the VOMS web interface and register with their personal data. After acceptance of the Acceptable Use Policy and an opt-in by email, the SA1 mailing list will be notified and accept the request.

[1] https://documents.egi.eu/public/RetrieveFile?docid=278&version=5&filename=EGI-VO_registration_process_v4.pdf

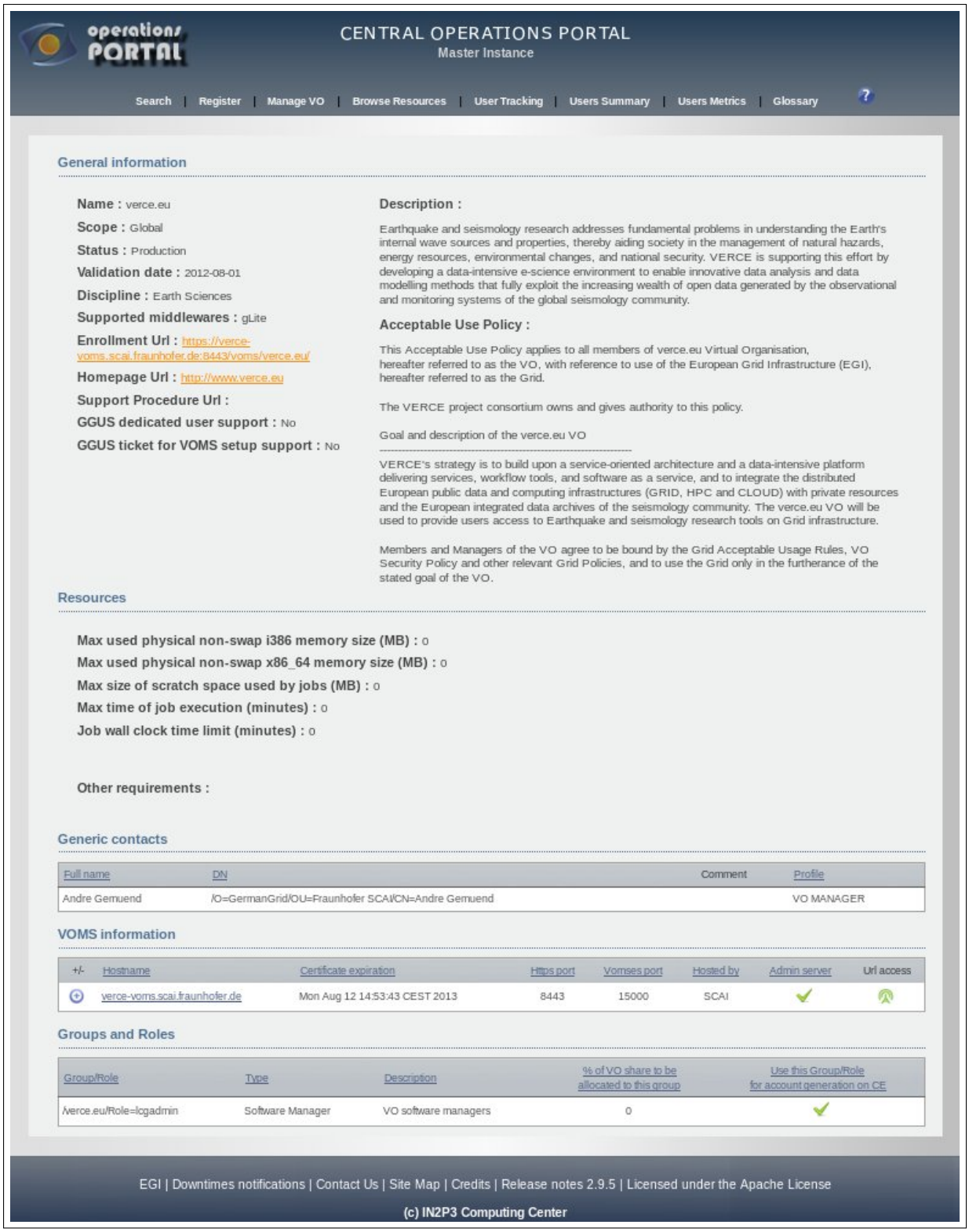

Figure 2 – `verce.eu` ID card

2.2 NGI/Site support

VERCE now has to find National Grid Infrastructures (NGIs) or individual sites that are willing to share resources with the VERCE VO. It is planned to contact the NGIs through their official communication routes and to publish the creation of the VO in the EGI and VERCE newsletters to encourage support.

3 Monitoring

Two tools are being used to monitor jobs on HPC centres: Inca for user-level monitoring and Scalasca (see 1.3.1) for performance monitoring.

3.1 User-level monitoring: Inca

To facilitate accessing information from the typically quite restricted Supercomputers, VERCE will re-use the monitoring data that is already collected in the PRACE infrastructure. In PRACE, *Inca* is used, which has initially been developed as a monitoring tool for the TeraGrid project but is now also the central monitoring software used by PRACE. It implements a client-server model where clients called *reporter managers* are testing components and sending the collected monitoring data to the central Inca server for processing, archival and presentation. Inca detects infrastructure problems by executing periodic, automated, user-level testing of software and services. In PRACE, it monitors the state of selected user environment software components, including applications, compilers, shells and tools, as well as various user level tests of provided services. System administrators and users can check if all services provided by the systems are correctly implemented and available.

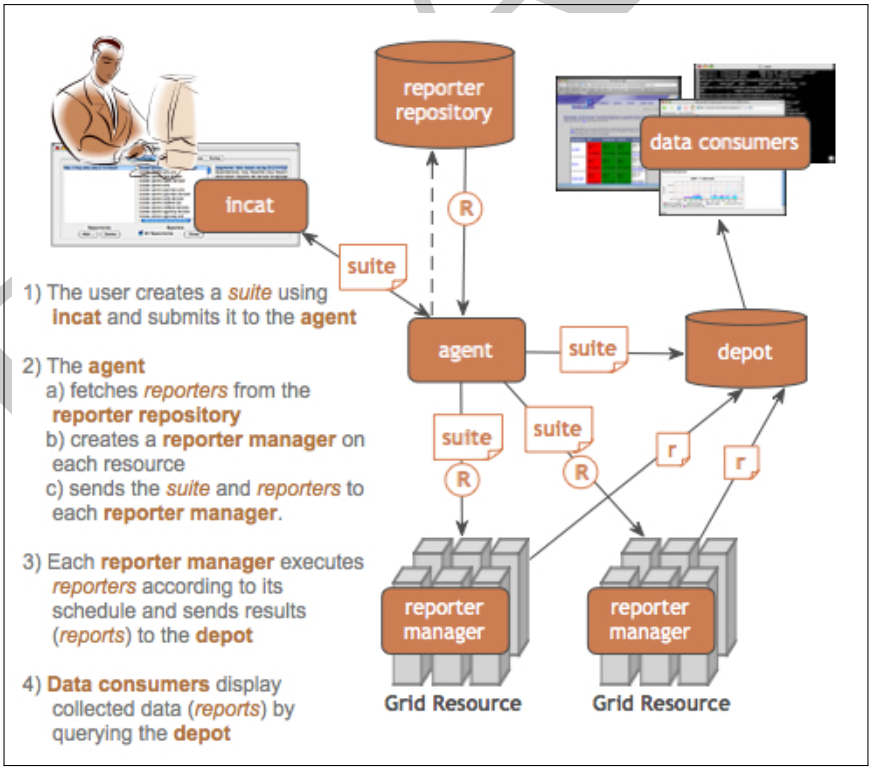

Figure 3 – Inca monitoring architecture

As LRZ has the most experience in maintaining Inca, they will be hosting and service the central installations of Inca for VERCE. The installation of the reporter managers on the resources is foreseen in the next period.

4 Conclusion, perspectives, issues

During these 6 last months, SA1 paid attention first of all on the organization of the working group, started to deploy components recommended by SA2, and gather documentation.

As explained in the *Risk Mangement report*, manpower shortage is still an issue, but this is expected to be resolved soon.

Next period will focus on the definition of a clear procedure for the deployment of the components in order to make them easily installable, fully operational and documented. The monitoring part has to be more clearly defined, for each type of resource.

Glossary and Links

EGI	European Grid Infrastructure www.egi.eu
EUDAT	European Data Infrastructure http://www.eudat.eu
GridFTP	High-performance, secure, reliable data transfer protocol optimized for high-bandwidth wide-area networks (tool) http://www.globus.org/toolkit/docs/latest-stable/gridftp
INCA	Monitoring tool (used in PRACE) which detects Grid infrastructure problems by executing periodic, automated, user-level testing of Grid software and services http://inca.sdsc.edu
ObsPy	Python Toolbox for seismology/seismological observatories (library) http://obspy.org
PRACE	Partnership for Advanced Computing in Europe http://www.prace-project.eu
Scalasca	Software tool that supports the performance optimization of parallel programs by measuring and analyzing their runtime behavior http://www.scalasca.org
SeisSol	High Resolution Simulation of Seismic Wave Propagation in Realistic Media with Complex Geometry (code) http://www.geophysik.uni-muenchen.de/~kaeser/SeisSol
VirtualEnv	Python tool used to create isolated environments for Python in which you can install packages without interfering with the other virtualenvs nor with the system Python's packages.
VO	(Virtual Organisation) Dynamic set of individuals defined around a set of resource-sharing rules and conditions.

Appendix A: Computing resources available to VERCE

Designation semantics

In the Designation of a resource, we find

1. the first 3 letters indicate the type of resource:
   - DAT: Data Archive or other data provider,
   - DEP: Department/Institute resource,
   - GAT: Gateway resource,
   - GRI: Grid resource,
   - HPC: High Performance Computing resource;
2. the name of the partner;
3. a 2-digit number unique to the partner.

Hardware resources

Designation	Resource Type	Short Name	Brief Description of Resource (connectivity, storage, job mngnt, ...)	Access Protocols	Installed Software	Accounting	Person in charge
HPC-LRZ-01	Computation	SuperMIG/ SuperMUC	PRACE Tier 0 machine. SuperMUC (Summer 2012) Intel Xeon-Architecture, more than 110,000 cores 3PFlop/s, 16cores/node with 2GB/core memory and 40cores/node with 6.4 GB/core memory 10 TB GPFS (aggregate BW 200 GB/s) linked via fully non-blocking Infiniband network 1GB Ethernet to archive and backup	ssh to login nodes behind firewall, only accessible from IPs entered in firewall; then port 22 and globus ports range opened.	Globus, UNICORE	yes INCA, Dmon	Cerlane
GRI-LRZ-02	Computation	Linux Cluster	> 2800 cores		Globus, UNICORE, gLite, dCache	yes (but only for internal purpose)	Cerlane
DEP-UEDIN-01	Computation	EDIM1	Data intensive cluster. 120 nodes, each with 6.24TB of storage CPU atom dual core, 4GB ram	SSH to login node behind firewall only from IPs within Edin. Informatics or EPCC.	Managed through ROCKS (CentOS)	yes Ganglia	Paul
GAT-UEDIN-04	Enactment	ADMIRE	Framework for running data intensive workflows, expressed in DISPEL, and implemented on OGSA-DAI	deployment onto Apache Tomcat.		no	Amy
HPC-CINECA-01	Computation	IBM SP Power6	Model: IBM pSeries 575 IBM P6-575 Infiniband Cluster, Processor IBM Power6, 4.7 GHz 168 Nodes, 5376 Cores: 5376, RAM: 21 TB (128 GB/node) Internal Network: Infiniband x4 DDR Disk Space: 1.2 PB Operating System: AIX 6 Peak Perf: 101 TFlop/s		GridFTP, GSI-SSH, UNICORE	yes INCA	Graziella
HPC-CINECA-02	Computation	PLX GPU Linux Cluster	Model: IBM PLX (iDataPlex DX360M3) Architecture: Linux Infiniband Cluster, 274 IBM iDataPlex M3 Nodes, Processors: 2 six-cores Intel Westmere 2.40 GHz (548 processors, 3288 cores) GPU: 2 NVIDIA Tesla M2070/node for 264 nodes + 2 NVIDIA Tesla M2070Q/node (for 10 nodes), RAM: 48 Gb/node Internal Network: Infiniband with 4x QDR switches Disk Space: 100 TB Operating System: Red Hat RHEL 5.6, Peak Perf: 300 TFlop/s (142 TFlops sustained - Linpack benchmark)		Local Batch Scheduler (PBS), UNICORE if needed	yes	Graziella
GRI-SCAI-01	Computation	EGI Cluster	32x Sun Fire X2200 M2 (Dual Opteron 2218, 2x2 cores), 8GB RAM, 250GB HDD 1 x Sun Fire X4600 M2 (Dual Opteron 8220, 4x2 cores), 78GB RAM, 2x140GB SAS HDD 8.5 TB NFS4 on ZFS shared filesystem (homes), 4.5 TB gLite DPM Storage Element (Grid) Connectivity: 2x Gbit Ethernet, Mellanox MT25204 10Gbps InfiniBand.	gLite, Unicore, Globus job/data access. GSISSH login to UI and dev-node (1, 2).	gLite CREAM, Unicore 6, Globus 4.2, VOMS, LFC, DPM, UI, WMS	yes	André
DEP-IPGP-01	Data	COHER-C	110TB storage, 4 IBM x3755 (2x2 cores Opteron 8218 @ 2.6 Ghz, 4 GB RAM)	local login, firewalled	LSF	yes	Geneviève
DEP-IPGP-02	Computation	Parallel Cluster	64 IBM x3550 (2x4 core Xeon 5420 2.5 Ghz, 8 GB RAM), myrinet 2000	local login, firewalled	LSF	yes	Geneviève
GRI-IPGP-03	Computation	EGI Cluster	32 heterogeneous single core CPU (≥2MHz) RAM: 2GB/CPU STORAGE: 1TB extensible CONNECTIVITY: 100Mb/s ethernet (compute nodes), 1Gb/s ethernet (services)	gLite job/file operations, ssh to login on User Interface	gLite 3.2	yes	David
DEP-ULIV-01	Data	Linux Storage	CPU Intel Xeon dual core 2.8 GHz, 2GB RAM, 2.8TB storage			no	Andreas
DEP-INGV-01	Computation		48 dual-proc AMD 6136 2.4GHz 8-core 64GB RAM (total 768 cores) 64 dual proc AMD 2378 2.4GHz 4-core 16GB RAM (total 512 cores) 36 TB scratch-grade storage	ssh through relay server	PBSPro	no	Daniele

notes:
(1) same soft- and hardware as workernode
(2) Local accounts for gsissh and local jobs possible (myproxy based).

Data and Software resources

Designation	Resource Type	Short Name	Brief Description of Resource	Access Protocols	Installed Software	Accounting	Person in charge
GAT-UEDIN-02	Enactment	DISPEL	Workflow language, with dataflow perspective for data intensive processes			no	Amy, Paul
GAT-UEDIN-03	Enactment	OGSA-DAI	Standard interface to data and services.	deployment onto Apache Tomcat.		no	Amy
GRI-EGI-01	Computation	EGI Virt. Org.	EGI Earth Science Research VO	gLite operations	gLite software stack	yes (EGI accounting)	André, David
GRI-EGI-02	Computation	EGI Virt. Org.	VERCE.EU VO	gLite operations	gLite software stack	yes (EGI accounting)	André
DAT-KNMI-01	Data	ORFEUS Data Center	Virtual European Broadband Seismograph Network, European Integrated Data Archive	Web services, interactive sessions	ArcLink		
DAT-IPGP-04	Data	COHER-D	IPGP Data Archive, NIED (Japan)	direct access from IPGP network		no	Geneviève