D-SA1.3 – VERCE platform: Second operation and monitoring report including application deployment

29/10/2013

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-SA1.3.pdf
Authors[1]*:*	S. Claus (SCAI), A. Gemünd (SCAI), G. Moguilny (IPGP), V. Muraleedharan (IPGP), M. Schnell (SCAI), D. Waldmann (LRZ) and D. Weissenbach (IPGP)
Location:	http://www.verce.eu/Repository/Deliverables/RP3
Type of document:	Deliverable
Dissemination level:	Public
Status:	Draft
Due date of delivery:	30/10/2013
Reviewer:	I. Klampanos (UEDIN/JRA2)
Keywords:	Unified access, iRODS, Testbed, Cloud, Inca, Authentication and Authorisation, Modules

History	*Author*		*Date*	*Comments (and sections)*
1	G. Moguilny	(IPGP)	05/09	Initial skeleton
2	V. Muraleedharan	(IPGP)	13/09	iRODS data platform (section 3)
3	M. Schnell and S. Claus	(SCAI)	17/09	Cloud resources (section 5)
4	A. Gemünd	(SCAI)	23/09	New sites requirements (2.1) Authentication and Authorisation (2.3)
5	D. Waldmann	(LRZ)	27/09	Inca monitoring (section 4)
6	D. Weissenbach	(IPGP)	27/09	Modules environment (2.2)

[1] Alphabetical order

Copyright notice

COPYRIGHT © VERCE PROJECT, 2011-2015. SEE www.verce.eu FOR DETAILS ON VERCE.

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.

This work is licensed under the Creative Commons Attribution-Noncommercial 3.0 License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc/3.0 or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, and USA.

The work must be attributed by attaching the following reference to the copied elements:
COPYRIGHT © VERCE PROJECT, 2011-2015. SEE www.verce.eu FOR DETAILS ON VERCE. Using this document in a way and/or for purposes not foreseen in the license requires the prior written permission of the copyright holders. The information contained in this document represents the views of the copyright holders as of the date such views are published.

Contents

Executive summary **4**

1 VERCE testbed **5**

2 Unified Access **5**
2.1 New sites requirements 5
2.2 Modules 5
2.3 Authentication and Authorisation 7

3 Data platform with iRODS **8**
3.1 Current configuration 8
3.2 Evaluation and integration of this data platform 8

4 Testbed monitoring with Inca **9**
4.1 Monitoring updates 9
4.2 Monitoring metrics 10

5 Cloud resources **11**
5.1 EGI Federated Cloud Task Force 11
5.2 SCAI Cloud Environment 11

6 Conclusion and perspectives **12**

A Computing resources available to VERCE **13**

B Software components **14**

List of Figures

1 `obspy`, `basemap` and `h5py` software dependencies. 6
2 Dependencies resolution with a single module. 6
3 iRODS site details in VERCE platform. 8

List of Tables

1 Derivation and calculation of metrics for SA2. 11
2 Hardware resources. 13
3 Data and Software resources. 14

COPYRIGHT © VERCE PROJECT, 2011-2015

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.

During this last months, SA1 focused on the standardisation of the VERCE platform, in order to simplify the operation of each site and the use of this platform. First, to allow system administrators of new sites to join VERCE rapidly, a best practice document describing a suggested minimum set of components is in preparation. Secondly, to simplify the use of the platform, SA1 proposes to use the *Modules* system that simplify the load of the right environment (software versions and dependencies) for final users. The Authentication and Authorisation question is being studied, in particular by discussions with the Eduroam project who develops an world-wide roaming access service for the international research and education community.

To enable researchers to easily access and share data, an IRODS data platform has been setup and six VERCE partners are already federated.

To monitor the testbed, Inca is being deployed and already installed in HPC and Grid resources. Inca can monitor a large types of services, as data transfers, deployments (version numbers), job submissions... In coordination with SA2, Key Performance Indicators have been defined.

SA1 is also evaluating the possibility to integrate Cloud resources in the testbed and two environments and projects are being investigated: the *EGI Federated Cloud Task Force* which offers a kind of "Grid of Clouds" and the *SCAI Cloud Environment* based on OpenNebula.

COPYRIGHT © VERCE PROJECT, 2011-2015

1 VERCE testbed

The updated list of the computing resources available to VERCE can be found in the Appendix A pages 13 and 14 and the updated list of the components per site is described in the Appendix B page 14.

2 Unified Access

2.1 New sites requirements

Documentation and suggestions of the VERCE service activities regarding setup and maintenance of software on the sites of the testbed has previously been confined to components to be deployed additionally on existing sites, already providing productive services. The reason for this viewpoint is simply that sites of the testbed were either already part of infrastructures such as PRACE and EGI, or were already providing services for institutional users. The components of the VERCE software stack admit different underlying Linux distributions and software stacks, which allows site administrators to easily fulfil requirements of other users in parallel. Additionally, the utilized middleware should not restrict the choice of underlying resource managers, schedulers, storage systems, etc. This is inevitable in cases of supercomputers and Grid sites for example. On the other hand, this degree of freedom leads to ambiguities for sites that are not experienced in the set up of a site for e-infrastructures. Recently, a request for an enhanced description of the whole software setup emerged, which is being adressed by SA1.

Besides components that are approved for the VERCE platform, some additional software, mostly prerequisites for, resp. dependencies of, the installed middleware, have to be deployed to join the testbed. A best practice document describing a suggested minimum set of components including alternatives where possible is currently in preparation, as discussed in the last SA1 meetings. The document will offer a summary of the requirements and suggestions for deployments.

2.2 Modules

The 'environment module' software is a comprehensive tool to enable computing resources to expose different versions of the same software in a coherent environment: it handles dependencies and mutual exclusions, allows libraries, APIs and applications to be built with different parameters or compilers and to be instantiated separately. In a classical installation, all software stand in the same place (typically under `/usr` or `/usr/local`) and thus need different designations to allow users to make the distinction between versions or parametrisations. When using modules, the user simply loads *modulefiles* that will adapt her/his UNIX environment to the requested software. For instance, one could access several gcc versions and still use the same `gcc` generic command in scripts or Makefiles.

'Environment module' offers an enriched `tcl` interpreter to execute modulefiles. Besides the main role of setting appropriate environment variables (typically `PATH`, `LD_LIBRARY_PATH` and so on) these short programs can as well manipulate other modules or define aliases for better user awareness; it is available for all recent Linux distributions.

While already adopted by PRACE HPC resources and on some VERCE departmental resources, the SA1 recommendation is to deploy it on every computing resource in order to expose to the final VERCE user a coherent and similar interface. By carefully defining a single and/or few VERCE modules, most often available in different versions, the user will straightfully trigger the use of the complete VERCE software stack, rather than namely loading separate modules. Such a module would of course need to be maintained locally by the resource administrators, with SA1 taking care of Grid resources as these are designed to allow well identified external users (called software managers) to install and maintain a Virtual Organisation's own software stack.

COPYRIGHT © VERCE PROJECT, 2011-2015

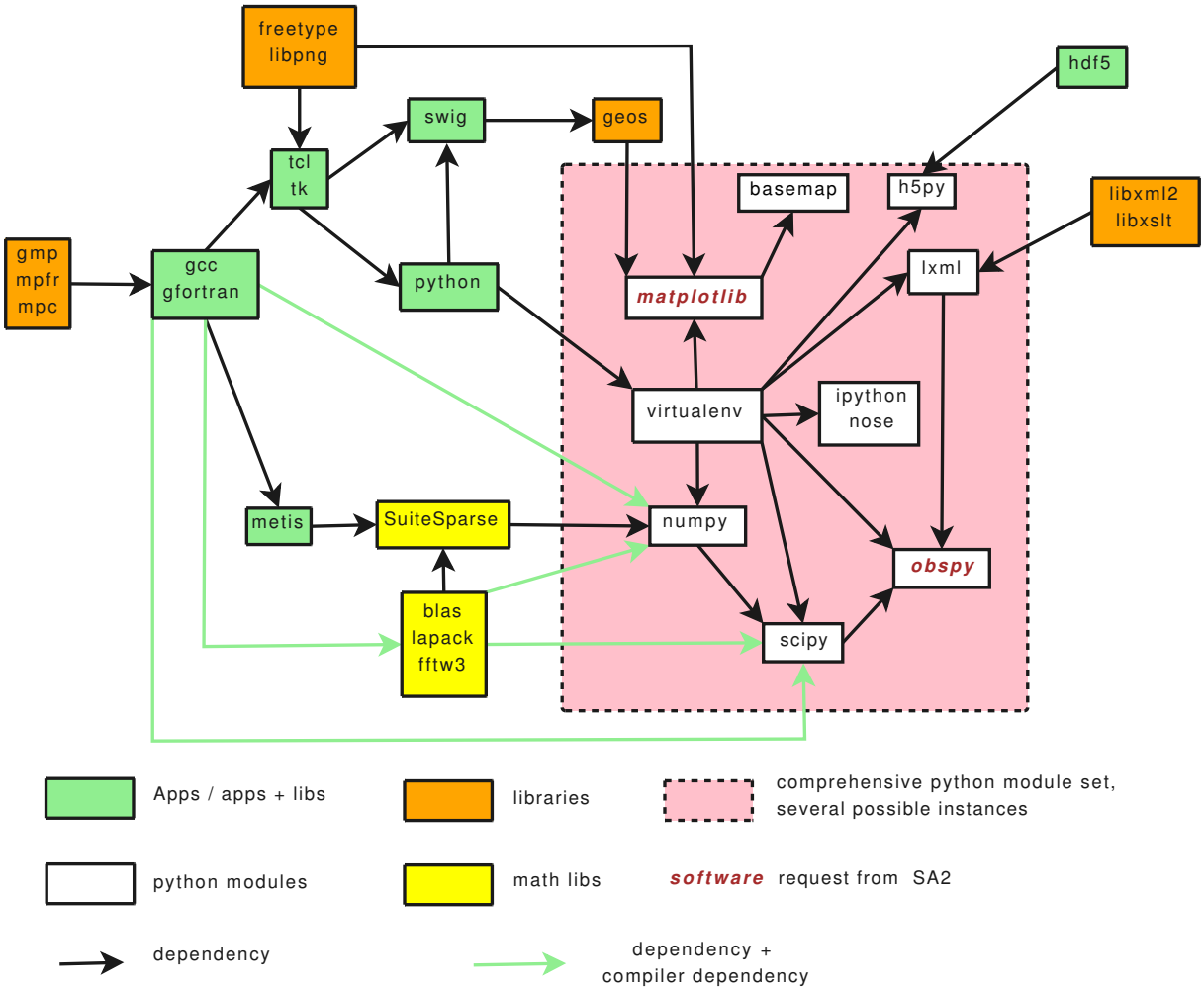


Figure 1 – obspy, basemap and h5py software dependencies.

```
$ module list
No Modulefiles Currently Loaded.
$ module load python/gcc/2.7.3_mod
$ module list
Currently Loaded Modulefiles:
  1) gcc/4.8.1                  7) SuiteSparse/gcc/4.0.2
  2) atlas/gcc/3.10.1           8) swig/gcc/2.0.10
  3) fftw3/scalar/gcc/3.3.2     9) geos/gcc/3.3.7
  4) tcltk/gcc/8.5.9           10) hdf5_18/1.8.11
  5) python/gcc/2.7.3          11) python/gcc/2.7.3_mod
  6) metis/gcc/4.0.3
$
```

Figure 2 – Dependencies resolution with a single module.

COPYRIGHT © VERCE PROJECT, 2011-2015

Illustration: complex python applications for VERCE

An example of how well defined environment modules help the user to instantiate his applications is shown below: the case of a `python` script using python modules `obspy`, `basemap`, and `h5py`. The full amount of required software is high, much more than ten components as shown by Fig. 1 page 6.

At IPGP, this python code would require to load 10 modules, including gcc, mathematical libraries (atlas, fftw and SuiteSparse), hdf5, geos, python and python extensions. The actual python extensions module (an adaptation of the `virtualenv` python module) does it all by itself, loading the immediately needed 7 dependencies modules, which in turn load the according `swig` and `metis` modules. All complex dependencies resolution are solved by the use of a single command shown in Fig. 2 page 6.

A VERCE module would load the full python environment, as well as modules for other applications selected by SA2 (e.g. SpecFEM3D, GMT...), when available on the resource.

An additional outcome of the usage of 'environment modules' is that it also greatly helps solving problems of monitoring the availability of the VERCE applications on VERCE sites by INCA, as INCA sensors are already module aware.

2.3 Authentication and Authorisation

Due to the heterogeneity of the resources in the VERCE testbed, different authentication and authorization mechanisms are used currently. Resources belonging to existing infrastructures like EGI obviously inherit their AA methods and policies. As a final harmonization of these mechanisms across infrastructures in Europe is a task that is out of scope for VERCE, SA1 is in contact with European initiatives attempting to solve this challenge in a general way. E.g. members of SA1 have contributed to the Earth Science use case in the paper "Federated Identity Management for Research Collaborations". The further progress of this FIM4R initiative are closely followed.

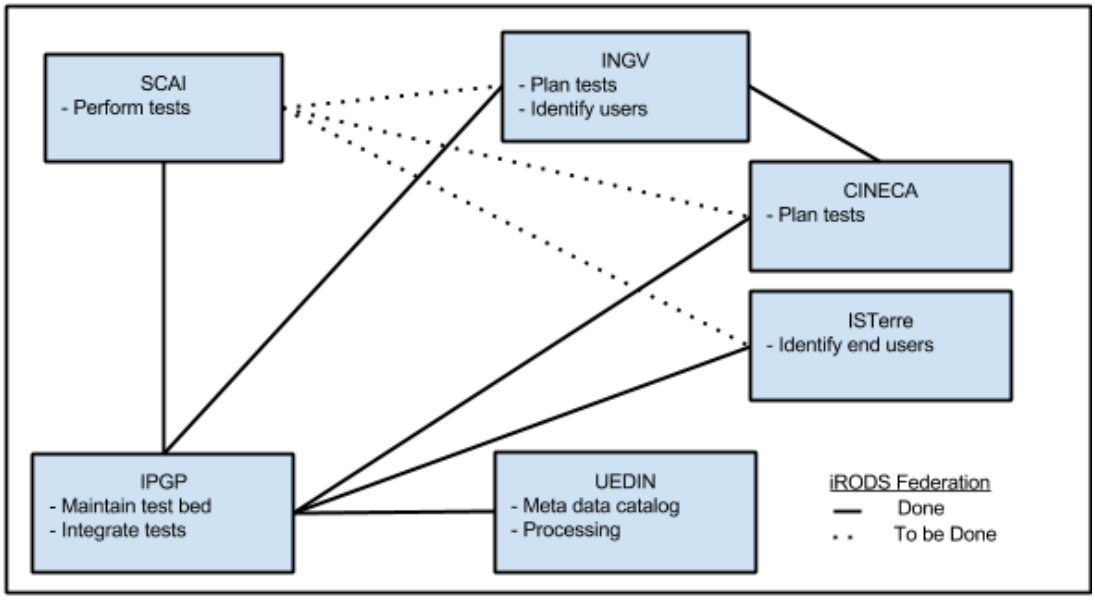
Additionally, SA1 submitted a proposal of a VERCE use case to the GEANT3plus/REFEDS Use-Case Submission for assisting communities in integrating their AAI services into eduGAIN.

> The eduGAIN service is intended to enable the trustworthy exchange of information related to identity, authentication and authorisation between the GÃL'ANT (GN3plus) Partners' federations. The eduGAIN service will deliver this through co-ordinating elements of the federations' technical infrastructure and a policy framework controlling the exchange of this information. [2]

The project, comparable to the Eduroam initiative, will establish a federation for access to e-infrastructure resources. In the current phase, the project intends to concentrate on browser-based access and SAML. For VERCE, however, X.509 certificate authentication is mandatory to access HPC and Grid resources. To integrate this with other resources, e.g. web-based portals, a kind of credential translation mechanism (e.g. SAML $\leftrightarrow$ X.509) is additionally required. Regarding the integration of third-party projects, the GN3Plus service thus deems it too early for a direct collaboration, i.e. integration of VERCE as a use-case. However, it might be possible to be included in a later phase of the project. As some of the use-cases of the GN3Plus Moonshot pilot task involve non-browser access, there might also be other interesting possibilities for VERCE to profit from the results achieved in this initiative. A larger federation like eduGAIN would be a good solution for VERCE, as it is more sustainable like specific custom solutions.

[2] eduGAIN service

COPYRIGHT © VERCE PROJECT, 2011-2015



Site id	iCAT Server	port	ZoneName	Test User
IPGP	ui1.ipgp.fr	1247	IPGP05Zone	irods
EDIM1	dir.epcc.ed.ac.uk	1247	UEDINZone	visakh
UJF	quath.ujf-grenoble.fr	1247	cigri	briandx
CINECA	irods-dev.cineca.it	1248	CINECA	mcarpene
INGV	verceirods.rm.ingv.it	1247	INGV1211	pdanecek
SCAI	irods.d-grid.scai.fraunhofer.de	1247	verce-scai	verce-scai

Figure 3 – iRODS site details in VERCE platform.

3 Data platform with iRODS

3.1 Current configuration

VERCE data platform is using iRODS as the widely distributed data management system. This enables researchers to easily access and share data with other researchers. This data platform allows each site to preserve their data policies and provide data and storage access to federated partner sites who have different set of policies. Now, the sites participating in current evaluation are CNRS-IPGP, CNRS-ISTerre (UJF), UEDIN, CINECA, INGV and SCAI.

Seismic data stored in each site will be cataloged based on meta-data and stored in their own servers. The goal is to make data accessible by querying meta-data information like array, station name, component and time, so the researchers do no have to be worried of the file details and format. This data platform also serves as the data store for other VERCE services for processing data.

As shown in Fig.3, all sites listed above are federated with IPGP and test file transfers were performed using IPGP test user. Since SCAI will be performing the evaluation, the SCAI federation with other partner sites is also planned.

3.2 Evaluation and integration of this data platform

The identified tasks to evaluate this VERCE data platform are:

- define and perform platform evaluation tests,

- catalog meta-data asynchronously in each sites using iRODS micro-services,
- services to query data based on meta-data,
- enable Unified Access to services provided by data platform...

Joined efforts of SA1 and SA2 members will be required for these tasks.

4 Testbed monitoring with Inca

4.1 Monitoring updates

During the last reporting period, the Inca monitoring solution has been further extended and adapted to the VERCE testbed infrastructure. The list of currently monitored sites could be extended and now includes:

- HPC: HPC-CINECA-02, HPC-LRZ-01,
- GRI: GRI-IPGP-03, GRI-LRZ-02, GRI-SCAI-01.

The list of currently monitored services, respectively components, could be extended to:

- *Middleware/Data Transfer* (from central Inca instance to the sites as well as cross tests from site-to-site; hourly interval).
  Available tests: GSISSH, GridFTP.

- *Deployment* (comparison of actual software version vs. required version; daily interval). GRAM, GridFTP and GSISSH provide a command-line based way of querying the version number. In order to retrieve the versions of the middlewares Unicore and gLite (CE and WMS) basically a registry has to be queried, which is currently ongoing work.
  Available tests: GRAM, GridFTP, GSISSH.

- *Deployment* (comparison of module version vs. required version; daily interval). The module version tests extract the component version from the module incorporating the component. For the following components this approach is necessary, since the software version number cannot be queried directly (e.g. SeisSol does not have a internal version number). Thus for proper working of these tests an installation of the module system and the setup of the modules is required. The installation of the module system within the VERCE testbed is currently discussed in SA1 (cf. section 2.2).
  Available tests: GMT, Matplotlib, Metis, Obspy, SeisSol, SpecFEM3D.

- *Middleware/Job Submission* (from central Inca instance to the sites; hourly interval). This can only be done from Inca-to-site, since the clients for job submission need to be installed on the testing machine. The clients have been installed on the central Inca instance.
  Available tests:

  - gLite CE, gLite WMS: since the gLite clients are intended to be run under Scientific Linux (whereas Inca is running on SuSE Linux) it is extremely difficult to port them on other OS. The CE test is working but the WMS test does not provide full functionality;
  - GRAM;
  - Unicore: the test is working but in order to check all Unicore resources in the testbed a Unicore registry is required. The setup of a registry for VERCE is currently discussed in SA1.

COPYRIGHT © VERCE PROJECT, 2011-2015

4.2 Monitoring metrics

In order to support the Key Performance Indicators (KPI) required by SA2, SA1 investigated the applicability of Inca for providing the KPIs related to monitoring. For a complete list and detailed description of the KPIs please refer to D-SA2.3.0. Altogether 5 KPIs (see 4.2.2) from that list have been identified which can be either directly or indirectly derived from Inca metrics, which will be explained in this section.

4.2.1 Definitions

Status – The following definitions apply for the status of running services as well as availability of software installed on the VERCE testbed:

1. *service/component available (OK)*. The service/component is available (i.e. working properly, respectively at the requested version number) at the time the test is being executed;
2. *service/component Not available (NOK)*. The service/component is not available (e.g. due to failure or downtime respectively the requested software version number is not installed) at the time the test is being executed;
3. *service/component status Unknown (UK)*. The status of the service/component is unknown (e.g. due to failures or maintenance of the monitoring system) at the time the test is being executed.

Tests – The following definitions refer to the tests executed by the Inca reporters:

1. *number of OK (nOK).* Number of tests with status OK in the time period considered;
2. *number of NOK (nNOK).* Number of tests with status NOK in the time period considered;
3. *number of Total Tests executed (nTT).* Total number of tests executed in the time period considered;
4. *number of relevant Tests (nT)*. Number of tests executed in the time period considered *excluding* UK (i.e. nTT - nUK), i.e. tests relevant for calculation of metrics;
5. *Test time Interval (tTI).* Interval between service tests (in hours).

Note that these tests are executed in different time intervals, e.g. version checks are made usually only once a day whereas service checks normally run once per hour.

Metrics – The status and test descriptions above allow the definition of the following metrics:

1. *Service Availability (SA).* Percentage of time the service is running during the time period considered;
2. *number of Service Interruptions (nSI).* Number of service interruptions in the time period considered;
3. *duration of Service Interruptions (tSI).* Duration of service interruptions in the time period considered;
4. *Total number of Services/components (nST).* Total number of services/components which are provided in the VERCE testbed during the time period considered;
5. *number of Services/components Monitored (nSM).* Number of services/components monitored during the time period considered;
6. *Availability Monitoring (SM).* Percentage of services monitored during the time period considered;
7. *Completed Deployment within scheduled timeframe (DC).* Percentage of components which are deployed on the VERCE testbed within the scheduled time frame.

4.2.2 Calculation of metrics

[3]Currently there is no possibility to schedule downtimes of services of the VERCE testbed, therefore the service availability is calculated as (nOK / nT) $\times$ 100 instead of (nOK / nTT) $\times$ 100.

COPYRIGHT © VERCE PROJECT, 2011-2015

Table 1 – Derivation and calculation of metrics for SA2.

Metric	Formula		Derivation
SA[3]	$(nOK/nTT) \times 100$	[%]	continuous probing by Inca
nSI	$nNOK$	[-]	continuous probing by Inca
tSI	$nNOK \times tTI$	[h]	continuous probing by Inca
AM	$(nSM/nST) \times 100$	[%]	manually
DC	$(nOK/nST) \times 100$	[%]	querying Inca at end of deployment phase

The defined metrics can be derived and calculated from Inca as shown in table 1 page 11. Due to the ongoing extensions of the monitoring system and tests as mentioned in section 4.1, the metrics cannot yet be delivered for all sites and services. Further adaptations of the system might also render some service- or site-specific metrics obsolete.

5 Cloud resources

VERCE has been encouraged to evaluate Cloud resources in the testbed, or as an option to run services and applications. SA1 has started a task for this purpose, which will evaluate the execution of a workflow on Cloud resources.

Two major cloud environments and projects for integration into the VERCE platform are investigated in the following sections. First, the EGI Federated Cloud Task Force, an EGI project to accumulate and share local virtualised environments with external collaborators and secondly the secure Cloud4Health infrastructure provided by SCAI. Both projects share the goal to establish a secure and easy-to-use cloud environment with a unified virtual machine, image storage, data and user management to further exploit and utilise unused computing resources. In order to fullfill these tasks, certain aspects needs to be solved and defined. Among them are:

- Interface for the end user to start and stop his job/Virtual Machine (VM),
- Unified accounting and authentication of VERCE users in different Cloud environments,
- Simulation data access across data center boundaries.

5.1 EGI Federated Cloud Task Force

The EGI Federated Cloud was set up by a TaskForce of EGI with the goal to offer a “Grid of Clouds” to the scientific community, which combines the ease of use and flexibility of Cloud approaches, with the openness and sharing of Grid environments. It is a federation of academical private clouds of different resource providers, for which the operations are provided centrally. More information can be found in the web presentation at EGI. More general approach with different cloud software stacks, e.g. OpenNebula, OpenStack, Stratuslab, are ongoing projects.

5.2 SCAI Cloud Environment

The Cloud environment at SCAI is based on OpenNebula with the Open Cloud Computing Interface (OCCI)[4]. Its main characteristics are:

- Isolated infrastructure, only accessible over secure VPN.
- Architecture can be adapted with a more comfortable access to the VERCE community, e.g. user requests a simulation with application and the corresponding VM is deployed.
- Java interface to control the VM over OCCI used for web applications

[4] http://opennebula.org/documentation:rel4.2:occiug

COPYRIGHT © VERCE PROJECT, 2011-2015

Originally, the SCAI Cloud environment was designed to process sensitive medical data in a secure and isolated environment for the BMWI (Federal Ministry of Economics and Technology) founded *cloud4health* project. Many tools and techniques developed for *cloud4health* can be transferred to host a VERCE Cloud infrastructure. SCAIs' Cloud environment is based on the open source data center virtualization software OpenNebula. It is capable of managing images and files in so called data stores, it supports separated virtual clusters and is equipped with a VM scheduler. In addition, it provides various interfaces to enable third-party application access to the Cloud management features. Exemplary interfaces are OCCI, the Amazon EC2 as well as a Java API, which can be used to start virtual machines from Portlet applications.

The following tasks were performed to incorporate VERCE into the Cloud environment at SCAI:

- Templates with virtual machines created for different use cases (ObsPy and GMT...).
- Context template to setup network and use case specific informations like which use case to execute, data location and to load the users certificate into the VM, so the user is able to log into the virtual machine, e.g. for debugging purposes.
- Separate VLAN tags for isolating network traffics from different users and projects.
- Portlet gateway to directly start use case with selected files and specified number of VMs.

Future plans to further improve usability includes:

- Integration of the Cloud environment into the verce.eu virtual organisation
- iRODS integration into the VM templates for simulation data transfer
- Add Cloud resources to the VERCE Inca monitoring system.

6 Conclusion and perspectives

The main achievements for this period are:

- the enrichment of the "User documentation" and "Deployment" wiki pages;
- the creation of the iRODS data platform for VERCE, including today six partners;
- a simplified procedure to access VERCE software with modules;
- the deployment of Inca for monitoring the testbed and the definition of KPIs in coordination with SA2.

Next steps are:

- evaluation of the iRODS data platform,
- completion of the best practice document for the integration of new sites;
- concrete run of complete workflows in a Cloud infrastructure;
- pursuance of the Inca deployment;
- possible use of portlets for Authentication and Authorisation, maybe with SCI-BUS;
- implementation of a VERCE registry in collaboration with JRA2.
- investigation to find a complete and independant Ticketing System.

COPYRIGHT © VERCE PROJECT, 2011-2015

A Computing resources available to VERCE

Table 2 – Hardware resources.

Designation	Resource Type	Short Name	Brief Description	Access Protocols	Accounting	Person in charge
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)	ssh to login node	yes	Graziella
HPC-CINECA-03	Computation	BlueGene/Q	Architecture: 10 BGQ Frame with 2 MidPlanes each Front-end Nodes OS: Red-Hat EL 6.2 Compute Node Kernel: lightweight Linux-like kernel Processor Type: IBM PowerA2, 1.6 GHz Computing Nodes: 10.240 with 16 cores each Computing Cores: 163.840 RAM: 16GB / node; 1GB/core Internal Network: Network interface with 11 links ->5D Torus Disk Space: more than 2PB of scratch space Peak Performance: 2.1 PFlops	ssh to login node	no	Graziella
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.	yes INCA, Dmon	Cerlane
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		Daniele
DEP-IPGP-06	Data	S-CAPAD Dell cluster	96 CPU nodes,16 data intensive nodes. 4 GPGPU nodes. 1 SMP node, parallel storage : GPFS on 576 effective TB, a high-performance network with InfiniBand QDR switches.	ssh to login node	yes	Geneviève
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	yes Ganglia	Paul
DEP-ULIV-01	Data	Linux Storage	CPU Intel Xeon dual core 2.8 GHz, 2GB RAM, 2.8TB storage		no	Andreas
GRI-IPGP-03	Computation	EGI Cluster	6 IBM x3550 (2x4 core Xeon 5420 2.5 Ghz, 8 GB RAM) with more nodes currently added, 1TB storage to be soon expanded to 9TB 2 user interfaces (IBM 325 single-core twin CPU)	gLite job/file operations, ssh to login on User Interface	yes	David
GRI-LRZ-02	Computation	Linux Cluster	> 2800 cores		yes (but only for internal purpose)	Cerlane
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)	yes	André

COPYRIGHT © VERCE PROJECT, 2011-2015

Table 3 – Data and Software resources.

Designation	Resource Type	Short Name	Brief Description of Resource	Access Protocols	Installed Software	Accounting	Person in charge
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

B Software components

	TestBed									
	HPC-LRZ-01	HPC-CINECA-02	HPC-CINECA-03	DEP-UEDIN-01	DEP-ULIV-01	DEP-IPGP-06	DEP-INGV-01	GRI-SCAI-01	GRI-IPGP-03	GRI-LRZ-02
Codes										
SeisSol (1.0)	✓	✓	✓				◒	●		✓
SpecFEM3D (2.0)	✓	✓	✓			◒	●			✓
GMT (4.5.9)	●	◒			●	✓	●	●		●
Scientific libraries										
Matpotlib (1.0.5)	●	✓		●		●	●	✓	●	●
ObsPy (0.8.3)	●	✓		●		✓	●	✓	●	●
Metis (4.0.3 or 5.0.2)	●	✓	●			●		✓	●	●
Environment / Tools										
Admire/Dispel (1.0)				●	●				◒	
OGSA-DAI (4.2)				●	●				◒	
GSI-SSH (GT 5.2.x)	✓							✓	◒	✓
GridFTP (6.14)	✓	✓	✓	●			✓	✓	●	✓
GRAM (5)							◒	✓		✓
JSAGA (0.9.15)				●				✓	●	
Unicore (6)	●	✓	✓				●			
gLite CREAM (1.14.0)								✓	●	
gLite WMS (3.4)								✓		

◒ Currently being installed

● Installed

✓ Available for VERCE users

An empty cell means not applicable or undecided as of yet

COPYRIGHT © VERCE PROJECT, 2011-2015