

D-SA2.4: VERCE platform integration: Third release report of integrated services and tools

15/10/2014

Project acronym: VERCE
Project n°: 283543
Funding Scheme: Combination of CP & CSA
Call Identifier: FP7-INFRASTRUCTURES-2011-2
WP: WP6/SA2, Integration and evaluation of the platform services
Filename: D-SA2.4.pdf
Authors: S.H. Leong (BADW-LRZ)
Location: <http://www.verce.eu/Repository/Deliverables/RP4>
Type of document: Deliverable
Status: Final
Due date of delivery: 15/10/2014
Reviewer: M. Atkinson
Keywords: Evaluation, Integration, Testing, Tools, Services, Application codes, Software components, Fifth release, Report

<i>History</i>	<i>Author</i>	<i>Date</i>	<i>Comments</i>
1	Siew Hoon Leong	28/08/2014	Initial draft for comments
2	Siew Hoon Leong	11/09/2014	Corrections
3	Siew Hoon Leong	07/10/2014	Update: Internal reviewer's comments
4	Siew Hoon Leong	29/10/2015	Minor polish

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. Fifth Release Report	5
1.1. Release Management Schedule	5
1.2. Requested component	5
1.3. Assignments of Evaluators/Testers and Resources	7
1.4. Evaluation and Testing	7
1.4.1. Component Specific Tests	7
1.5. Results and Recommendations	8
1.6. Documentations	9
2. Lesson Learned	9
3. Conclusion and Future Plans	10
3.1. Schedule for next cycle	10
Appendix A: Release Management Schedule	12
References	13
hyperref links	
List of Figures	
1. Release Management Schedule (Oct 2013 – Mar 2014)	5
List of Tables	
1. Requested Component for Evaluation and Testing	6
2. Assigned Resources and Testers of each component	7

Executive Summary

One of the objectives of the VERCE project is to provide a service-oriented architecture and framework that wraps the data-infrastructure resources and services with a set of distributed data-aware Grid and HPC resources provided by the European e-Infrastructure and community. To this end, the tools, services and application codes, i.e. software components, which are particularly relevant to the seismologists and the Earth Science community, are selected for integration on the VERCE platform.

The main aim of this report is to report on the fifth release of the integrated tools and services. The Plan-Do-Check-Act (PDCA) cycle is used to manage the release process. This period corresponds to the fifth completed PDCA cycle and thus the fifth release.

The fifth release successfully completed within the schedule timeframe. In this release, five components were evaluated and four were approved as shown in the list below.

Approved

- ObsPy 0.9.2
- MPI for Python 1.3.1
- NetworkX 1.8.1
- iRODS 3.3.1

Delayed

- MSNoise 1.2

The main focus of all work-packages in this reporting period was to continue to support the provision of a working version of the VERCE Science Gateway for user evaluation. Additionally, following the advice to focus on supporting the forward modelling use case via the Science Gateway, only components that are beneficial for this specific purpose were approved.

The lessons learned in the earlier releases have resulted in now the usual smooth and efficient evaluation cycle. Members of the JRAs were as usual providing strong support to enable the on-time completion of this evaluation cycle.

1. Fifth Release Report

The fifth PDCA cycle, corresponding to the fifth release of integrated services and tools, was completed on 31 July 2014 of the project. As in the previous releases, a release management schedule was prepared to ensure that the process was clear to each involved work package. JRA1 and JRA2 submitted two and three tools respectively for evaluation. Four tools were approved in this release. The details of this release are described in the following subsections.

1.1. Release Management Schedule

The work performed in this six months period is depicted in Figure 1. Learning from the experience, the JRAs and SAs were once again encouraged to submit their request form as early as possible (before the official deadline on 30th April 2014).

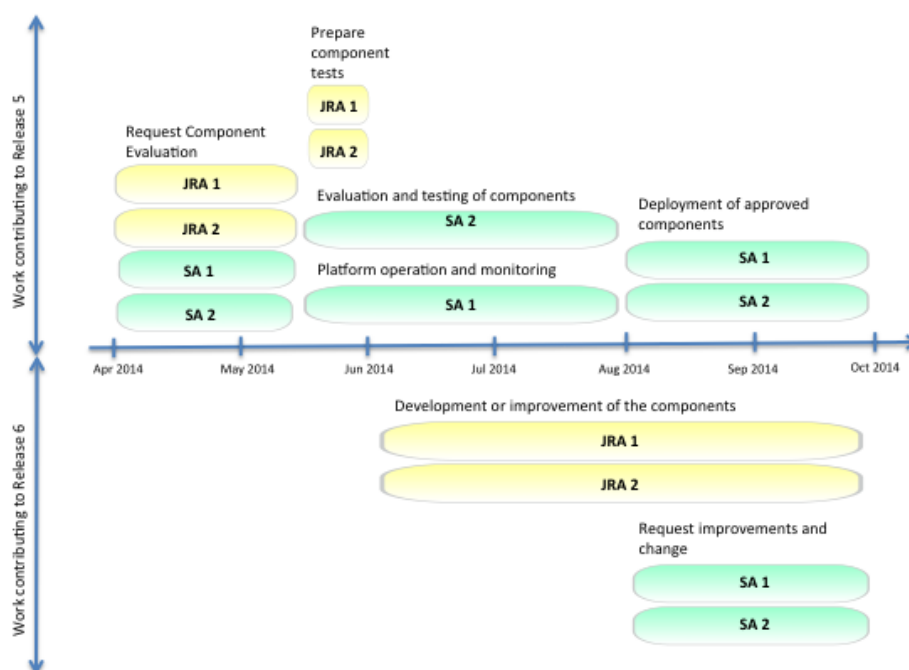


Figure 1 – Release Management Schedule (Oct 2013 – Mar 2014)

1.2. Requested component

Five tools evaluation requests were received from the JRAs as shown in the Table 1. Detailed information about the component is available at http://www.verce-project.eu/projects/verce1/wiki/RP4a_Evaluation_and_Tests.

Component	Version	Type	Purpose	Submitted by
MSNoise[4]	1.2	Tool	Complete software package for computing and monitoring relative velocity variations using ambient seismic noise	JRA1
ObsPy[5]	0.9.2 with Python 2.7.6	Tool	Python framework for processing seismological data	JRA1
MPI for Python (mpi4Py)[1]	1.3.1	Tool	Provides bindings of the Message Passing Interface (MPI) standard for the Python programming language, allowing any Python program to exploit multiple processors	JRA2
NetworkX[2]	1.8.1	Tool	Python language software package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks	JRA2
Integrated Rule-Oriented Data System (iRODS)[3]	3.3.1	Tool	An open-source data management software	JRA2

Table 1 – Requested Component for Evaluation and Testing

1.3. Assignments of Evaluators/Testers and Resources

The partners from other work packages were once again requested to assist in the evaluation and testing. The assigned resources and testers for each component is shown in Table 2

Component	Assigned Resource (SA1 Definition)	Assigned Tester
MSNoise	SuperMUC (HPC-LRZ-01)	LRZ: Siew Hoon Leong
	EGI Cluster (GRI-LRZ-02)	LRZ: Siew Hoon Leong
	Departmental Resource (DEP-INGV-01)	KNMI: Alessandro Spinuso
	Departmental Resource (DEP-ULIV-01)	LRZ: Siew Hoon Leong ULIV: Thomas Garth (supporting)
ObsPy	SuperMUC (HPC-LRZ-01)	LRZ: Siew Hoon Leong
	EGI Cluster (GRI-IPGP-03)	IPGP: Visakh Muraleedharan IPGP: David Weissenbach (supporting)
	EGI Cluster (GRI-SCAI-01)	SCAI: Andre Gemünd
	EGI Cluster (GRI-LRZ-02)	LRZ: Siew Hoon Leong
	Institutional Resource (DEP-UEDIN-01)	UEDIN: Amy Krause UEDIN: Iraklis Klampanos
mpi4Py	SuperMUC (HPC-LRZ-01)	LRZ: Siew Hoon Leong
	EGI Cluster (GRI-IPGP-03)	IPGP: Visakh Muraleedharan IPGP: David Weissenbach (supporting)
	EGI Cluster (GRI-SCAI-01)	SCAI: Andre Gemünd
	EGI Cluster (GRI-LRZ-02)	LRZ: Siew Hoon Leong
NetworkX	SuperMUC (HPC-LRZ-01)	LRZ: Siew Hoon Leong
	EGI Cluster (GRI-SCAI-01)	SCAI: Andre Gemünd
	EGI Cluster (GRI-IPGP-03)	IPGP: Visakh Muraleedharan IPGP: David Weissenbach (supporting)
iRODS	LRZ VMWare virtual machine	IPGP: Visakh Muraleedharan LRZ: Matteo Lanati (supporting)
	Institutional Resource (DEP-UEDIN-01)	IPGP: Visakh Muraleedharan

Table 2 – Assigned Resources and Testers of each component

The assigned resources are based on the profile of the components. Components that can potentially utilise HPC and/or GRID resources are assigned to at least one of such resources for evaluation. Unique component like iRODS that requires special data resources is evaluated with any available data resources that partners can provide.

1.4. Evaluation and Testing

The evaluation and testing phase commenced in early May 2014 and was completed by the end of July 2014. In anticipation of the summer vacation, the team was encouraged to begin the evaluation process as early as possible. The detail of each specific component test is described in the following sub-section.

1.4.1. Component Specific Tests

MSNoise

MSNoise was evaluated in this cycle as a possible tool to be integrated with the Science Gateway. A compatibility evaluation was performed on a HPC and Grid resource. The conclusion was that it is unsuitable for resources that utilise a batch job system. MSNoise requires interactive access via XSession since one of its main features is to provide an easy to use client interface to other Python libraries like ObsPy. This is unfortunately incompatible with the usage model of HPC and Grid resources from PRACE and EGI. As such, departmental resources were added to the evaluation. JRA1 and JRA2 experts were also consulted to check the possibility of integrating the client interface with the Science Gateway. However, there is currently no support for this functionality.

ObsPy

ObsPy was recommended for an upgrade in this evaluation cycle. Significant number of new features were implemented in version 0.9.2. This new version of ObsPy is compatible with any Python 2.7.* version. The evaluation also included a recommended upgrade Python to 2.7.6 (non-compulsory). All required python modules, e.g. SciPy, NumPy and Matplotlib, and ObsPy test suites were run to verify the correctness of the installation and its integration with the existing VERCE platform. Additional tests, simplified version of the core python scripts used by the science portal, were also requested from JRA2 and SA3. The scripts were run on updated test resources to ensure the integrity of the platform after the upgrade.

MPI for Python (mpi4Py)

mpi4Py was evaluated in this period since it is an important library for dispel4py (a VERCE development) and the visualisation service that is provided via the Science Gateway. Both HPC and Grid resources were evaluated. The included python scripts were used to check the installation. dispel4py scripts were also requested and used to verify the installation.

NetworkX

NetworkX was evaluated in this period since it is another dependency of dispel4py. Both HPC and Grid resources were included in the evaluation. The python tests that were included in the distribution was used to verify the installation. Additional dispel4py scripts were requested from JRA2 to ensure that there are no integration issues with the platform.

Integrated Rule-Oriented Data System (iRODS)

The iRODS system was evaluated in this period since it is a crucial data component in VERCE. It is used for data storage after a job is completed. Data search and retrieval from the Science Gateway are managed via iRODS. Two iRODS resources were evaluated. Varying configurations (or rules), different zone setups, etc., were evaluated to assess stability and usability.

1.5. Results and Recommendations

Approved components

Four out of five evaluated components are approved for release after the evaluation and testing phase. The approved components in the fifth release are:

1) ObsPy

ObsPy was evaluated on three Grid, one HPC and one institutional resources. Installation was in general simple. Typically, the most complex step with regards to the installation of ObsPy was the installation of its dependency python libraries, in particular SciPy and NumPy. On the HPC resource at LRZ, these modules were installed separately as an external Python library during the last deployment. As such, most modules did not have to be reinstalled (can be reused) since the recommended Python upgrade was a minor version update, from 2.7.5 to 2.7.6. On the EGI resource at SCAI, all modules had to be reinstalled

since they were installed into the previous Python version. As such, the upgrade on this resource was more complex. On the institutional resource at UEDIN, the default setup as per recommended by Python and ObsPy could be used. Thus the installation was simple. All tests were successful completed. The ObsPy upgrade is thus accepted and recommended for all resource types.

2) MPI for Python (mpi4Py)

mpi4Py was evaluated on three Grid and one HPC resources. Installation was simple if the mpi libraries, e.g. Intel MPI or Open MPI, are already installed. The EGI cluster at IPGP was unable to support this module since they do not have the physical hardware on the cluster to support inter-nodes/processors communication. For all other resources, the recommended tests were successfully completed. As such, mpi4Py is accepted and recommended for all resource types that can support it.

3) NetworkX

NetworkX was evaluated on two Grid and one HPC resources. Installation was simple and the recommended tests were successfully completed. As such, NetworkX is accepted and recommended for all resource types.

4) Integrated Rule-Oriented Data System (iRODS)

The iRODS system was evaluated on two data resources. Both data resources were on-demand resources. The iRODS installation was easy. The actual challenge is to figure out the best configuration that suits VERCE's needs. A GridFTP interface is included with the iRODS installation to ensure a good data transfer performance. iRODS zones and backup were tested. Backup across zones has a significant overhead and thus live synchronisation across zones is not implemented. iRODS is accepted and recommended to all resource providers who have the data resources for VERCE usage.

Delayed components

1) MSNoise

MSNoise is unsuitable for the VERCE platform due to incompatibilities with HPC and Grid resources and integration issues with the Science Gateway/VERCE Platform. MSNoise, a client side tool, is providing a subset of services that the Science Gateway is providing. However, the client front-end user interface cannot be integrated onto the Science Gateway. To utilise MSNoise, one has to login to the resource where MSNoise is installed and start a XSession. The job will then run interactively on the login node. This usage violates both the VERCE policy (to use the Science Gateway as a user front-end) and also the resource providers, EGI and PRACE, policies. Running jobs directly on the login nodes are unacceptable on such resources. As such, despite of the usefulness of MSNoise as a tool, it is deemed as unsuitable for the VERCE platform and is thus delayed.

1.6. Documentations

All issues faced and solutions that the SA2 team encountered during this release cycle were documented in the SA2 wiki and in the request forms for each component. These documents contribute to the technical documentation that SA1 will use to coordinate the deployment of the approved components on the VERCE testbed. The documentation can be found at http://www.verce-project.eu/projects/verce1/wiki/RP4a_Results_Issues_and_Documentations.

2. Lesson Learned

The lessons learned in the previous release cycle were compiled and proposed as improvements and are carried out in this release. Clearer recommendations introduced previously continued to provide a

guideline to SA1 on the best matched or preferred resources. The constant communication with the work package members who had requested for component evaluation throughout the evaluation period greatly helped in the evaluation process.

3. Conclusion and Future Plans

The fifth release of the VERCE software components successfully completed within the scheduled time frame. Four out of five components were approved in the fifth release as a part of the VERCE platform. Currently, there are a total of eighteen unique approved components, including all previous releases, on the platform. In general, the procedures are effective for managing the release. Improvements that were recommended in the previous releases have helped SA2 to carry out its evaluation smoothly and effectively. SA2 will be on a constant lookout for ways to further improve its existing procedures. The release management schedule for the sixth release, refer to Section 3.1, is prepared and shared with work packages involved.

3.1. Schedule for next cycle

The schedule for the next cycle, 1st October 2014 to 31st March 2015, for the respective work packages is summarised below.

JRA2

- [1st October - 31st October 2014]: To provide requirements (tools and services to evaluate) via the Request Form. Simultaneously, JRA2 should be receiving new requirements from NA2 and JRA1 (and SAs) as per the roadmap to plan for the work in this reporting period.
- Before 31st October: To define tests to perform (in particular, functionalities that will be used) for external tools and in-house developed components.
- [1st November - 31st March 2015]: To work on developing the required workflow related tools.

JRA1

- [1st October - 31st October 2014]: To provide requirements (application codes to evaluate) via the Request Form. Simultaneously, JRA1 is receiving new requirements as per the roadmap from NA2 (and SAs) to plan for their development work in this reporting period.
- Before 31st October: To provide the tests (including input files).
- [1st November - 31st March 2015]: To work on developing the application codes.

SA2

- [1st October - 31st October 2014]: To identify missing features and functionalities as per the roadmap and thus suggest additional tools to be evaluated. Administrative work to prepare for the next release.
- [1st November - 6th November 2014]: To select the team for each component and begin installation on the selected resources.
- [7th November - 31st January 2015]: To work on evaluating and integrating the selected components.

- [1st February - 31st March 2015]: To check the service information collected by SA1 and provide feedback/new requirements to the JRAs. Simultaneously, SA2 will provide the accepted list of components (components that have passed the evaluation and tests) to SA1 and assist SA1 in deploying the approved components.

SA1

- [1st October - 31st October 2015]: To identify missing features and functionalities as per the roadmap and thus suggest additional tools to be evaluated. Administrative work to prepare for the next release.
- [1st November - 31st January 2015]: Operate and monitor the scientific platform and the tools and services running on it. To collect service information.
- [1st February - 31st March 2015]: To work on deploying the released components provided by SA2. To check the service information collected and provide feedback/new requirements to the JRAs. Continue to operate the scientific platform.

Appendix A: Release Management Schedule

Reporting Period: 1st April 2014 - 30th September 2014

JRA2

- [1st April - 30th April 2014]: To provide requirements (tools and services to evaluate) via the Request Form. Simultaneously, JRA2 should be receiving new requirements from NA2 and JRA1 (and SAs) to plan for the work in this reporting period.
- Before 31st May]: To define tests to perform (in particular, functionalities that will be used) for external tools and in-house developed components.
- [1st June - 30th September 2014]: To work on developing the required workflow related tools.

JRA1

- [1st April - 30th April 2014]: To provide requirements (application codes to evaluate) via the Request Form. Simultaneously, JRA1 is receiving new requirements from NA2 (and SAs) to plan for their development work in this reporting period.
- Before 31st May]: To provide the tests (including input files).
- [1st June - 30th September 2014]: To work on developing the application codes.

SA2

- [1st April - 30th April 2014]: To identify missing features and functionalities and thus suggest additional tools to be evaluated. Administrative work to prepare for the next release.
- [1st May - 7th May 2014]: To select the team for each component and begin installation on the selected resources.
- [8th May - 31st July 2014]: To work on evaluating and integrating the selected components.
- [1st August - 31st September 2014]: To check the service information collected by SA1 and provide feedback/new requirements to the JRAs. Simultaneously, SA2 will provide the accepted list of components (components that have passed the evaluation and tests) to SA1 and assist SA1 in deploying the approved components.

SA1

- [1st April - 30th April 2014]: To identify missing features and functionalities and thus suggest additional tools to be evaluated. Administrative work to prepare for the next release.
- [1st May - - 31st July 2014]: Operate and monitor the scientific platform and the tools and services running on it. To collect service information.
- [1st August - 31st September 2014]: To work on deploying the released components provided by SA2. To check the service information collected and provide feedback/new requirements to the JRAs. Continue to operate the scientific platform.

[Source: http://www.verce-project.eu/projects/verce1/wiki/Release_Management_Schedule#Reporting-Period-4a-1st-April-2014-30th-September-2014]

References

- [1] L. Dalcin, *MPI for Python*, August 2014, <http://mpi4py.scipy.org>.
- [2] NetworkX developer team, *High-productivity software for complex networks*, August 2014, <http://networkx.github.io>.
- [3] iRODS, *IRODS*, August 2014, <https://irods.org>.
- [4] T. Lecocq and C. Caudron, *MSNoise: A Python Package for Monitoring Seismic Velocity Changes using Ambient Seismic Noise*, August 2014, <http://www.msnoise.org>.
- [5] ObsPy Wiki, *ObsPy: A Python Toolbox for seimolgy/seimological observatories*, March 2014, <https://github.com/obspy/obspy/wiki>.