D-SA2.4.1: VERCE platform integration: updated release report of integrated services and tools

31/03/2015

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:	31/03/2015
Reviewer:	M. Atkinson
Keywords:	Evaluation, Integration, Testing, Tools, Services, Application codes, Software components, Sixth release, Report

---

History	*Author*	*Date*	*Comments*
1	Siew Hoon Leong	02/03/2015	Initial draft for comments
2	Siew Hoon Leong	12/03/2015	Corrections
3	Siew Hoon Leong	13/03/2015	Corrections by A. Gemuend
4	Siew Hoon Leong	01/04/2015	Corrections from S. Hachinger
5	Siew Hoon Leong	02/04/2015	Update: Internal reviewer's comments
6	Siew Hoon Leong	29/10/2015	Minor polish

Copyright notice

Contents

List of Figures

List of Tables

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 sixth 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 sixth completed PDCA cycle and thus the sixth release.

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

Approved

- SPECFEM3D 2.0.2
- h5Py 2.3.1
- dispel4py 1.0
- ParaView 4.2.0 with Python Support
- FFmpeg 2.4.3

The main focus of 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 and to update the application code, SPECFEM3D, such that the latest features and bug fixes were integrated.

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. Sixth Release Report

The sixth PDCA cycle of the project, corresponding to the sixth release of integrated services and tools, was completed on 31 January 2015. As in the previous releases, a release management schedule was prepared to ensure that the process was clear to each involved work package. JRA1 submitted a tool request and an application request while JRA2 submitted three tool requests for evaluation. Four tools and one application code 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 in view of the Christmas vacation.

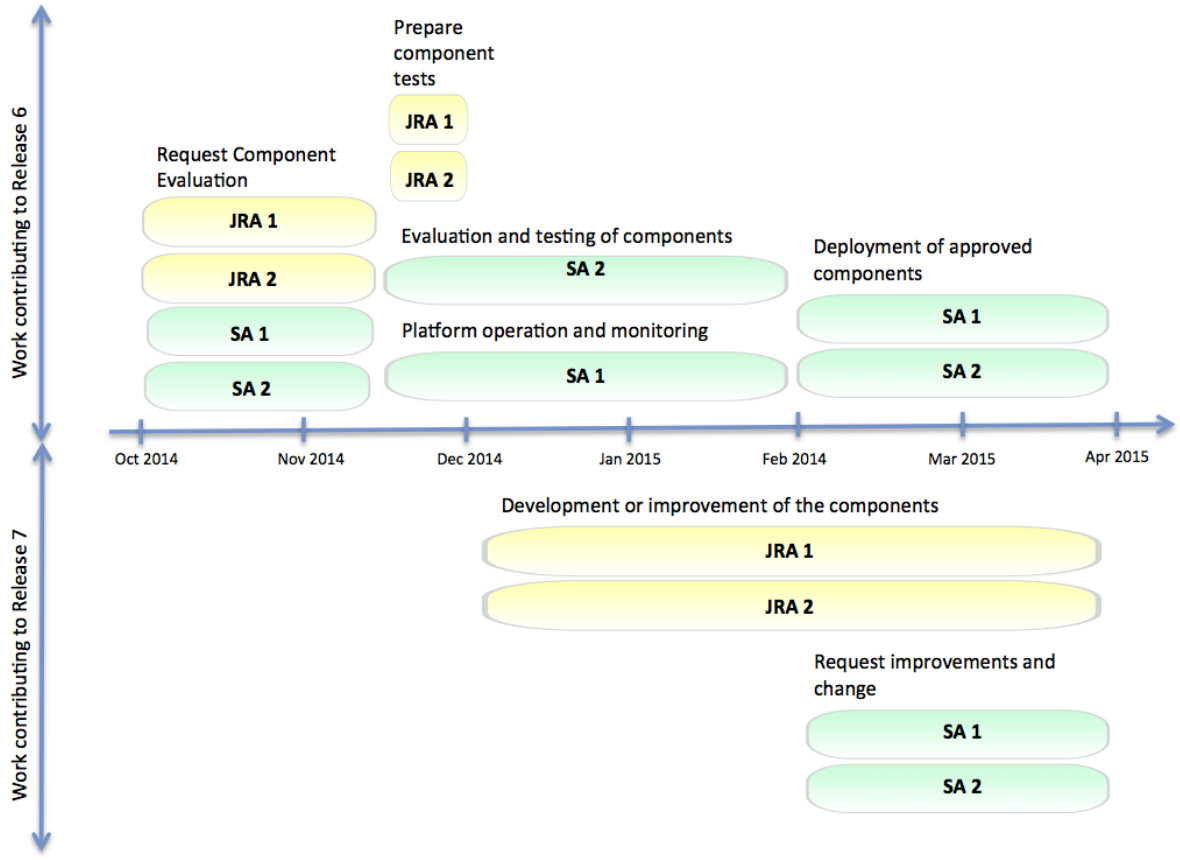

Figure 1 – Release Management Schedule (Oct 2014 – Mar 2015)

1.2. Requested components

Four tools and one application evaluation requests were received from the JRAs as shown in the Table 1. Detailed information about each component is available internally on Redmine.

Component	Version	Type	Purpose	Submitted by
SPECFEM3D[6]	2.0.2	Application Code	Simulates acoustic (fluid), elastic (solid), coupled acoustic/elastic, poroelastic or seismic wave propagation in any type of conforming mesh of hexahedra (structured or not)	JRA1
h5Py[1]	2.3.1 with serial/parallel hdf5	Tool	A Python interface to the HDF5 binary data format	JRA1
dispel4py [9]	1.0	Tool	A Python library to describe abstract workflows for distributed data-intensive applications	JRA2
ParaView[8]	4.2.0	Tool	An open-source, multi-platform data analysis and visualization application	JRA2
FFmpeg[4]	2.4.3	Tool	A complete, cross-platform solution to record, convert and stream audio and video.	JRA2

Table 1 – Requested Component for Evaluation and Testing

1.3. Assignment 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

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.

1.4. Evaluation and Testing

The evaluation and testing phase commenced in early Nov 2014 and was completed by the end of Jan 2015. In anticipation of the winter 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

SPECFEM3D

SPECFEM3D was recommended for an upgrade in this evaluation cycle. Some bug fixes that are significant to VERCE would be in this upgraded version. Unfortunately, due to an internal SPECFEM3D organisational issue after the move to GitHub, the developers were hesitating to make a new release in spite of the stability of the code. The recommendation was to utilise the latest code from the master branch. As such, SA2 cooperated with JRA1 to find a suitable copy of SPECFEM3D, i.e. with a specific hash. The experts from JRA1 finally recommended to use the copy on the development branch with hash 9e2aad47d52f974e465a1ceee9488faf515fc350. Since no release version is available for this copy, the last release version number, 2.0.2, is used for convenience. A change in the format of the configuration files and job batch script implied new tests had to be set up. An integration test was performed

Component	Assigned Resource (SA1 Definition)	Assigned Tester
SPECFEM3D	SuperMUC (HPC-LRZ-01)	LRZ: Siew Hoon Leong
	EGI Cluster (GRI-LRZ-02)	LRZ: Siew Hoon Leong
	Departmental Resource (DEP-SCAI-01)	SCAI: Andre Gemünd
h5Py	EGI Cluster (GRI-LRZ-02)	LRZ: Siew Hoon Leong
	SuperMUC (HPC-LRZ-01)	LRZ: Siew Hoon Leong
	Departmental Resource (DEP-SCAI-01)	SCAI: Andre Gemünd
dispel4py	SuperMUC (HPC-LRZ-01)	UEDIN: Amy Krause
	Departmental Resource (DEP-SCAI-01)	SCAI: Andre Gemünd
ParaView	SuperMUC (HPC-LRZ-01)	LRZ: Siew Hoon Leong
	EGI Cluster (GRI-LRZ-02)	LRZ: Siew Hoon Leong
	Departmental Resource (DEP-SCAI-01)	SCAI: Andre Gemünd
FFmpeg	SuperMUC (HPC-LRZ-01)	LRZ: Siew Hoon Leong
	EGI Cluster (GRI-LRZ-02)	LRZ: Siew Hoon Leong
	Departmental Resource (DEP-SCAI-01)	UEDIN: Iraklis Klampanos SCAI: Andre Gemünd (Support)

Table 2 – Assigned Resources and Testers of each component

with smaller number of cores before proceeding to use scientifically relevant examples from the portal. Features that were not properly supported in the previous release were specially tested in addition to the tests to compare results of this release to those from a previously installed version.

h5Py

H5Py was recommended for evaluation in this cycle. Both ObsPy and dispel4py, another component that was being evaluated in this cycle, can take advantage of its functionality and potentially improve the input/output performance. This component is dependent on Python and its many libraries, e.g. NumPy and SciPy. The included tests were used for the first verification. Additional tests from sample codes that perform both read and write were used to verify the installation.

dispel4py

dispel4py [5] is an in-house Python development by our partner, UEDIN, for data intensive applications. It enables the description of misfit calculations as workflows. It has a number of dependencies, e.g. ObsPy[10], NetworkX [3] and mpi4Py[2]. mpi4Py is a Python language binding for the Message Passing Interface (MPI) and is utilised to distribute the tasks in parallel on compute resources. dispel4py scripts will be installed on the Science Gateway and staged in to respective compute resources supported by the Gateway. The scripts were first tested by manually transferring them to the selected compute resources. Once the scripts were verified, the scripts were included in the Gateway and moved accordingly to the selected evaluation resources.

ParaView

ParaView was evaluated in this period for the visualisation feature of the Science Gateway. Both HPC and Grid resources were included in the evaluation. ParaView python support, i.e. pvpython, was tested to visualise and convert the output files of SPECFEM3D. The tests also included scripts that utilise mpi4Py to speed up the conversion process.

FFmpeg

FFmpeg was evaluated in this period since it is a crucial component to generate videos via the Science Gateway. First tests included simple commands to generate videos from png files. Finally, the production parallel (mpi4Py) script is utilised by the Gateway to generate the video on compute resources was ran to verify the correctness of the integration.

1.5. Results and Recommendations

Approved components

All evaluated components are approved for release after the evaluation and testing phase. The approved components in the sixth release are:

1) SPECFEM3D

SPECFEM3D was evaluated on one Grid, one HPC and one institutional resources. Installation was in general simpler than the previous release. When the latest Intel MPI version 5 was used, the recommended flags in the documentation were sufficient. However when Intel MPI version 4 was utilised, there were compilation errors that require setting additional MPI flags for the icc compiler. Tests that were used in the earlier release of SPECFEM3D were no longer valid in this new release due to a format change in the configuration. New tests were setup by the scientists in JRA1. The tests were then ran and checked for integrity among different resources. All tests were successfully completed. SPECFEM3D upgrade is thus accepted and recommended on all HPC resources and any Grid/Institutional resources that can support it.

2) h5Py

h5Py was evaluted on one HPC, one Grid and one institutional resources. Installation was simple if the HDF5 [7] technology suite is already installed on most compute resources. HDF5, in particular the parallel version, is a difficult to install and manage software. As such, the serial version is recommended on the VERCE setup since the pre- and post- processing work are embarrassingly parallel. Sample tests to check read and write functionality were used to test the setup. All tests were successfully completed. h5Py is thus accepted and recommended for all resource types that can support it.

3) dispel4py

dispel4py was evaluated on one HPC resource and one institutional resource. Installation was simple since it consists only of python scripts. It uses other python libraries like mpi4Py and h5Py. All tests were successfully completed. Dispel4py is thus accepted. It will reside on the Science Gateway and will be moved to the appropriate resources during the execution of the workflow. No installations on HPC/Grid/institutional resources, other than the Science Gateway, are required.

4) ParaView

ParaView was evaluated on one Grid, one HPC and one institutional resources. ParaView is a difficult software to install due to its many dependencies. As such, the precompiled version of ParaView is recommended for VERCE's purposes. Python support, i.e. pvpython, is ensured by setting additional environment variables and exchanging MPI python libraries. All tests were successfully completed. ParaView is thus accepted and recommended on all resources that will be supporting the visualisation requirements of the Science Gateway.

5) FFmpeg

FFmpeg was evaluated on one Grid, one HPC resource and one institutional resource. Installation and usage were both extremely easy. All tests were successfully completed. FFmpeg is recommended on all resources that will be supporting the visualisation requirements of the Science Gateway.

1.6. Documentation

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 on our internal Redmine under the heading "RP4b Results Issues and Documentations".

2. Lesson Learned

The lessons learned in the previous release cycles were compiled and proposed as improvements to be carried out in every new 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 that had requested for component evaluation throughout the evaluation period greatly helped in the evaluation process.

3. Conclusion and Future Plans

The sixth release of the VERCE software components successfully completed within the scheduled time frame. All components were approved in this release as a part of the VERCE platform. Currently, there are a total of twenty-two 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 and last cycle, 1st April 2015 - 30th September 2015, of the project for the respective work packages is summarised below.

JRA2

- [1st April - 30th April 2015]: To provide requirements (tools and services to evaluate) via the Request Form.
- 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 2015]: To work on supporting SA3 wtih the required workflow tools.

JRA1

- [1st April - 30th April 2015]: To provide requirements (application codes to evaluate) via the Request Form.
- Before 31st May]: To provide the tests (including input files).
- [1st June - 30th September 2015]: To work on supporting SA3 on the application codes.

SA2

- [1st April - 30th April 2015]: 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 2015]: To select the team for each component and begin installation on the selected resources.
- [8th May - 31st July 2015]: To work on evaluating and integrating the selected components.
- [1st August - 31st September 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 April - 30th April 2015]: 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 2015]: Operate and monitor the scientific platform and the tools and services running on it. To collect service information.
- [1st August - 31st September 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 and ensure the sustainability of the platform after September.

References

[1] A. Collette and contributers, *HDF5 for Python*, March 2015, http://www.h5py.org/.

[2] L. Dalcin, *MPI for Python*, August 2014, http://mpi4py.scipy.org.

[3] NetworkX developer team, *High-productivity software for complex networks*, August 2014, http://networkx.github.io.

[4] ffmeg.org, *FFmpeg*, March 2015, https://www.ffmpeg.org/.

[5] R. Filguiera, I. Klampanos, A. Krause, M. David, A. Moreno, and M. Atkinson, *dispel4py:A Python Framework for Data-Intensive Scientific Computing*, International Journal of High Performance Computing Applications (2015).

[6] Computational Infrastructure for GeoDynamics, *SPECFEM3D Cartesian*, March 2015, http://geodynamics.org/cig/software/specfem3d/.

[7] The HDF Group, *WELCOME TO THE HDF5 HOME PAGE!*, March 2015, http://www.hdfgroup.org/HDF5/.

[8] Kitware, *ParaView*, March 2015, http://www.paraview.org/.

[9] The University of Edinburgh, *Welcome to dispel4pyâĂŹs documentation!*, March 2015, http://dispel4py.org/.

[10] ObsPy Wiki, *ObsPy: A Python Toolbox for seimolgy/seimological observatories*, March 2014, https://github.com/obspy/obspy/wiki.