D-NA2.2.2: Training applications: first selection and documentation (in collaboration with WP3)

30/03/2013

Project acronym:	VERCE
Project n°:	283543
Funding Scheme:	Combination of CP & CSA
Call Identifier:	FP7-INFRASTRUCTURES-2011-2
WP:	WP2/NA2, Pilot applications and use cases
Filename:	D-NA2.2.2.pdf
Author(s):	A. Michelini
Location:	http://www.verce.eu/Repository/Deliverables/RP2/
Type of document:	Deliverable
Dissemination level:	Public
Status:	Final
Due date of delivery:	01/04/ 2013
Reviewer:	Giovanni Erbacci
Keywords:	data-intensive, cpu-intensive, HPC, earthquake, seismology, data infrastructure, forward modeling, inversion, metrics, evaluation

---

Version	*Author*	*Date*	*Comments*
1	A. Michelini (INGV)	17/03/2013	Initial draft for comments
2	G. Erbacci (CINECA)	23/03/2013	Reviewer's comments
3	A. Rietbrock (ULIV)	25/03/2013	Comments
3	J.-P. Vilotte (CNRS-IPGP)	26/03/2013	Extended executive summary, use case implementation, training applications update

Copyright notice

Contents

Executive Summary

The main objectives of WP2/NA2 are: (1) select existing pilot data-intensive applications and design sound use case scenarios; (2) analyze and define a use case implementation strategy during the project with WP8, WP7 and WP9; (3) support and evaluate the "productising" transition of the methods and their implementation performed by WP8; (4) support and evaluate the deployment and the efficiency of the pilot applications and their use case scenarios on the VERCE platform; (5) define in collaboration with NA3 documentation and tailored training session material; (6) provide requirements and support to WP7 and WP9 for tailored interfaces of the scientific gateways targeted to the developers and the users.

This document presents an update of the implementation of the data-intensive and cpu-intensive use cases detailed in D-NA2.1, and based on this analysis provides a list of identified core software components and their documentation. Together with WP3, this report identifies in relation with the use cases implementation progress and the identified software a number of tailored training sessions around the use case implementation and those software components.

VERCE's primary objective consists of "enabling" existing data- and HPC-intensive software applications through the development of processing elements (PEs) within dedicated workflows. It follows that the applications to be enabled or under enablement are all well developed and already have their own line of development. This also implies that they have already chosen their dissemination strategies through tutorials, web portals, etc.

In this report, we refer to an initial suite of applications enabled on the VERCE platform. The documentation provided in this report is necessarily concise since all these codes and module libraries are well documented on their own developer portals. The main identified applications are

- ObsPy [1] consisting of a Python Toolbox for seismology/seismological observatories
- Whisper [2] for large data sets seismic ambient noise analysis
- SPECFEM3D [3] for seismic wave simulation and its environment suite (meshing and graph partitioning)

With the next reports, this list will be extended with other applications as they are progressively enabled. All these codes are open beside the WHISPER suite which is provided to VERCE for implementation within the VERCE environment, the result of which being fully open to the community.

[1] https://github.com/obspy/obspy/wiki
[2] http://code-whisper.isterre.fr/html/
[3] http://www.geodynamics.org/cig/software/specfem3d

1 Use cases implementation and strategy

In the following sections, we provide a brief overview on the work performed for the actual implementation of the first VERCE use cases within the DISPEL environment and its implication for the first selection of the training applications and documentation. We highlight the main implementation strategy both for the data-intensive and cpu-intensive selected use cases, as well as the main components' contribution and the specification of the workflow requirements.

1.1 User code integration framework

In coordination with SA3, JRA1 and SA2, an integration framework has been provided to facilitate and coordinate the contribution of the seismology researchers in the implementation of the first selected applications and use cases within the DISPEL environment. This integration framework is detailed in the D-SA3.2.1 report, and has simplified the development effort of the seismology researchers directing their focus on reshaping and testing the selected software components.

Pair-programming and user's feedback: This activity has been organized through pair-programming sessions and on-line support. The pair-programming sessions were realized typically with frequent Skype Calls providing development support from SA3, JRA1 and JRA2, and with off-line code reviews. These involve pairs of seismologists and developers interested in similar analysis procedures. The coding activity was extremely important to better identify, analyze and take into account the inherent differences among some of the methods and approaches of the seismology partners. The identification and the acknowledgement of these differences suggested the need for their evaluation and for Processing Elements (PEs) that implement the same functionalities adopting different combinations of libraries and algorithms.

The first selected cpu-intensive use case is a forward modeling use case that aims at generating synthetic seismograms for various earthquake scenarios and earth models. The synthetic seismograms have to be compared with actual observations in order to assess the quality of the different earth models and to compare different wave simulation codes currently used by the seismology community.

1.2 CPU-intensive applications and use cases

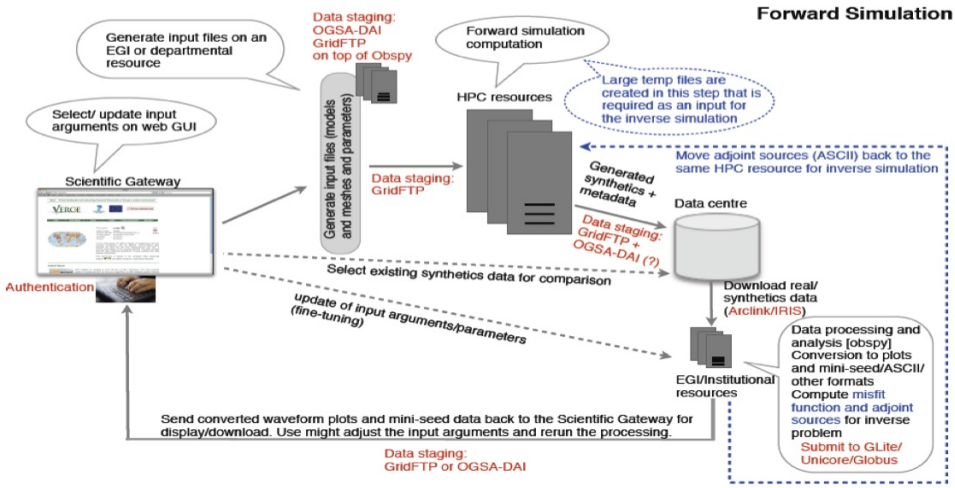

Figure 1 – Components diagram of the forward modeling use case

In Figure 1, the main identified components of the forward modeling use case are illustrated, see also the D-SA3.2.1 report for more explanation. Starting from an initial set of input data: earthquake events,

parameters and earth models, wave simulations are launched on different computing infrastructures in order to prove the versatility of the VERCE platform. The synthetic seismograms predicted by those simulations will have to be compared with the actual observation on existing seismology networks in order to assess and explore the data space, the earth models and the simulation codes.

Supported by regular meetings and discussions between NA2, JRA1, SA3, JRA2 and SA2 a roadmap has been identified. The workflow will create an input file to submit to HPC infrastructures (LRZ, CINECA) and possibly on a local cluster (INGV). The creation of the input file will be generic and compliant with different types of solvers already deployed and available in the current platform. After the submission of the job the workflow will take care of the retrieval and shipment of the results to the assigned resource. Functionalities like monitoring, exception and error handling will be implemented in a minimal version. The automation of the overall process achieved by means of the VERCE workflow engine can already be seen as an added value compared to the previous common practices. In the initial prototype a basic user interface in order to kick off the simulations and eventually visualize the results will be provided. In the later cycles, in close interaction with scientists and users, processing and visualization tools and interfaces will be tailored to the scientific requirements of applications.

At this stage, NA2 has selected some initial wave simulation codes and their software environment based on the community practice. Considerable efforts, involving close collaboration between NA2, JRA1 and SA2, have been focused on the selection and the integration of existing wave simulation codes and their software environment (meshing, parallel partitioning). These have involved a close collaboration between seismologists and the HPC computing infrastructures within SA2 in order to validate those components before implementation in the VERCE platform.

One of the main important software suites identified by NA2 for the forward modeling use case is based on:

SPECFEM3D : The spectral element code SPECFEM3D[4] is currently the most developed and most used code globally within the seismology research community, in more than 100 institutions worldwide. SPECFEM3Dallows the simulation of acoustic (fluid), elastic and viscoelastic (solid) and coupled acoustic/elastic seismic wave propagation using structured or unstructured conforming Cartesian mesh (hexahedra). This software has been validated on PRACE HPC infrastructures within SA2 (LRZ and CINECA) and is also available at the TGCC.

CUBIT : The meshing CUBIT [5] tool suite is used by SPECFEM3Dand is developed at Sandia National Labs. This software is available for academic research without restriction. This meshing software allow to build structured and unstructured conforming mesh using hexahedra elements on which SPECFEM3D is built. CUBIT benefits from a large user community, among which the seismology community, and from continuous ongoing efforts to improve and add new parallel mesh generation algorithms and functionalities such as geometry cleanup and simplification, complex pre-processing tasks for wave simulations in complex geological media.

SCOTCH **and** METIS : The SCOTCH[6] and METIS[7] packages are widely used graph and hypergraph parallel partitionning tool suites both for simulation and meshing software. These tool suites are used by SPECFEM3D, and allow efficient parallel mapping of domain decomposition strategy.

1.3 Data-intensive applications and use case

The generic data-intensive use case includes the processing and the statical analysis of continuous waveforms data sets. This consist of the following steps: data ingestion, data pre-processing (validation and

[4] http://www.geodynamics.org/cig/software/specfen3D
[5] http://cubit.sandia.gov
[6] http://www.labri.fr/perso/pelegrin/scotch/
[7] http://glaros.dtc.umn.edu/gkhome/views/metis

quality control), seismic noise cross-correlation analysis. The final cross-correlations are a secondary data product that can be used for different subsequent analysis such as tomography and/or seismic property time variations.

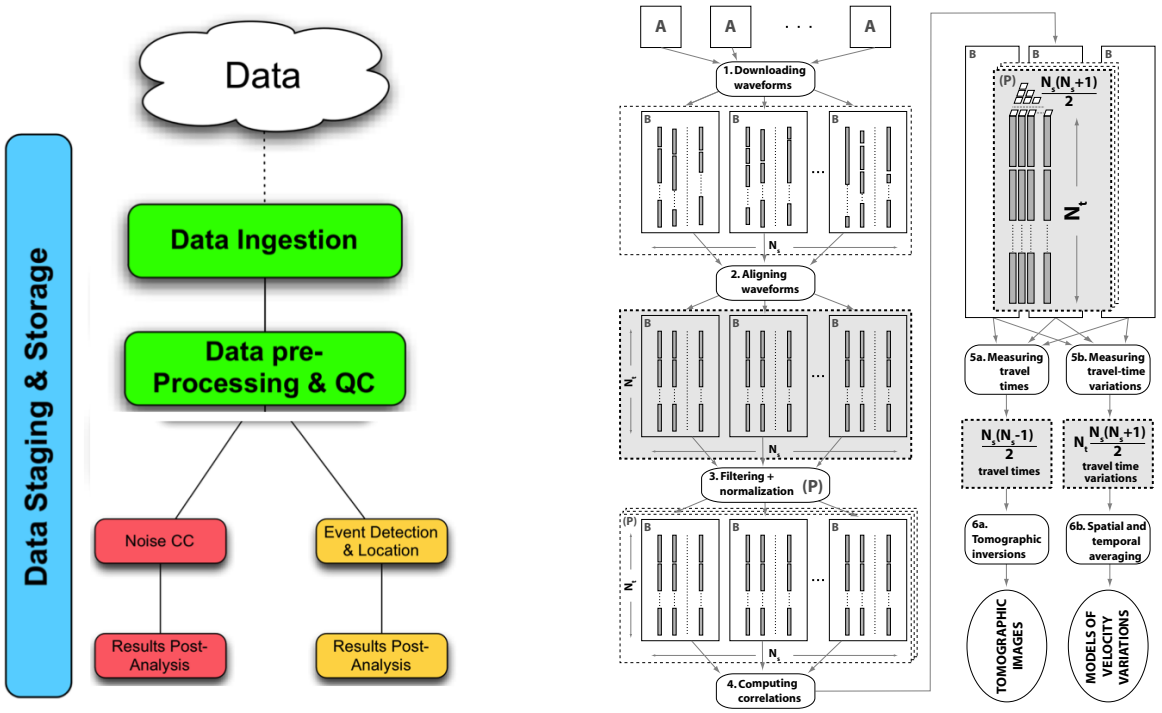

Figure 2 – components diagram of the data-intensive use case

In Figure 2, the main identified components of the noise correlation use case are illustrated, see D-NA2.1 report for more explanation. At the present stage, the work flow implementation starts after the data ingestion stage. After extracting the data, the workflow implements an initial version of pre-processing and noise-correlation processing stages. The extracted noise correlations are then either visualized or stored locally for further use locally.

Supported by regular meetings and discussions between NA2, JRA1, SA3, JRA2 and SA2 a roadmap has been identified. The main effort in NA2 and JRA1 has been to identify the first PEs and to provide software implementation of these PEs involving the seismology developers and collaboration with other projects like ObsPy [8] and WHISPER [9]. The coding activity revealed several differences among the methods and approaches pursued by the scientific partners, provoking interesting discussions. The existence and acknowledgement of these differences suggested the need to implement PEs adopting different combination of libraries (ObsPy, WHISPER, NumPy) and algorithms for the same functionalities. This in order to evaluate those different implementations and to provide a flexible library of PEs depending of the application and the research practice. These PEs could be used within the same DISPEL workflow. The PEs will therefore be able to make use of the libraries accepted and provided within VERCE (SA2).

Today, the contribution of the NA2 seismology researchers has allowed to implement and to test a number of PEs within DISPEL. The notion of parameters reflect the scientists' practice where the parameters are considered as separated from the input data. Even though PEs in the DISPEL environment treat those parameters as any other input, it was decided to keep the separation, in order to have things consistent with the scientists' development approach. This useful compromise has been achieved thanks to the adoption of DISPEL functions, producing composite and configurable high level PEs.The data type,

[8] https://github.com/obspy/obspy/wiki
[9] http://whisper.obs.ujf-grenoble.fr

which is a language-specific entity in DISPEL is further detailed in the report D-SA3.2.1.

Table 1: List of available PEs. The names currently used are temporary and will change as we move towards defining relevant naming conventions within VERCE.

Name	Author	Description
`Synchronization`	INGV	Time series syncronization using the Fast Fourier Transform.
`DataTrimming`	LMU	Cut data stream to given starttime and endtime. Parameters: starttime, endtime.
`SplitStream`	INGV	Cutting of data stream into chunks of given length. Input parameter: length of chunks (in sec).
`FillGaps`	INGV	Fills the gaps in the signal using linear interpolation or adding zeros, and merges the overlapping traces. If the gaps are less than the allowed percentage the function will interpolate, otherwise it will put 0. In case `pergap` equals 0, no interpolation is performed, only zeroes are added. Parameter: `pergap` (percentage of gap allowed), `time` (duration of the data on which pergap is computed).
`TemporalNormalization`	LMU	Implements 4 different methods of performing time normalisation on the traces. `clipping`: signal is clipped to `clip_factor` times the signal's standard deviation, `std`. `clipping_iter`: the signal is clipped iteratively: values above `clip_factor * std` are divided by `clip_weight`, until the whole signal is below `clip_factor * std`. `ramn`: running absolute mean normalisation; a sliding window runs along the signal. The values within the window are used to calculate a weighting factor, and the centre of the window is scaled by this factor. `[weight factor: w = np.mean(np.abs(tr.data[win]))/(2 * norm_win + 1) ]` finally, the signal is tapered with a tukey window, `alpha = 0.2`. `1bit`: only the sign of the signal is conserved. Parameters: `norm_method` (specifies which of the above mentioned methods to apply), `norm_win` (window length used), `clip_factor`, `clip_weight`
`ResamplingINGV`	INGV	Resample the original time series at a different sampling rate (uses: `obspy.core.trace.Trace.resample`). Parameters: `samplingrate`.
`ResamplingWHISPER`	CNRS	Resamples the signal trace with the new frequency. (uses: `scipy.signal.decimate`). Iterates decimation for appropriate small number in 8,7,6,5,4,3,2 and linear interpolation if necessary with the `numpy.interp` function. Parameters: `frequencyTrace`, `newFrequency`.
`Whitening`	INGV	Spectral whitening of the signal. Input parameters: minimum and maximum frequency range
`InstumentCorrection`	KNMI	Removes the instrument response from the signal. It uses poles and zeros provided by a repository of stations RESP files

Table 1 – continued from previous page

Name	Author	Description
Detrend	LMU	Linear trends in the signal are removed (uses the Obspy method: detrend.)
Filter	LMU	Applies a Butterworth filter to the signal for given frequencies, number of corners and filter type (bandpass, highpass, lowpass). Uses the Obspy method filter. Parameters: frequency_min, frequency_max, corners, zerophase, filtertype.
StreamToSeedFile	KNMI	Creates a mini-SEED file containing the input data and stores it to a local resource. Parameters: filedestination.
CorrelateNoiseLMU	LMU	Cross-correlation of two data streams (uses: obspy.signal.xcorr)
CorrelateNoiseWHISPER	CNRS	Computes the correlation with the time-shift maxlag of the two signal trace with the same length. (uses: scipy.fftpack). Parameters: maxlag, goodNumber (improves the performance of the computation by setting an appropriate combination of small prime).
Spectrogram	INGV	Plot seismogram using the python function spectrogram. Parameters: log, per_lap, wlen, filedestination
WaveformPlot	INGV	Plots waveform. Parameters: filedestination.
EventsReader	KNMI	Reads event catalogues from external services.
InventoryReader	KNMI	Reads station inventory information from external services (ArcLink).
PAZReader	KNMI	Reads instrument response information from external services (ArcLink).

Today a number of software and libraries have been identified in relation to the different stages of the data-intensive use case.

Data ingestion and access : As detailed in the D-NA2.1 report, this stage involves data storage, indexing and discovery within the VERCE data-intensive platform. This involves the ingestion and the extraction from the raw data of metadata that is then stored via a dedicated metadata repository or database. It will also involve the distribution and synchronization of the raw data on several nodes, adopting virtual file system technologies. The internal data access and distribution system will be crucial for the management of the raw data, as well as for the intermediate results, that can be re-used by the workflow or downloaded. The software and libraries identified for this stage at the moment are: ObsPy, which provides a reading, processing and visualization tool suite dealing with a number of seismological raw data and metadata formats, and which is widely used in the international seismological community; iRODS [10] for a distributed and virtual file system, which is widely used and is part of the EUDAT components; MonetDB [11], a scientific database for the metadata extraction and storage.

Data pre-processing and noise-correlation processing : As detailed in the D-NA2.1 report, these stages involve a number of signal processing elements which must scale out when dealing with large data

[10] http://www.irods.org
[11] http://www.monetdb.org

sets to be analyzed. The optimization of those processing elements and their efficient parallel instantiation on different infrastructures will be crucial and must take into account the perfectly parallel nature of the pre-processing phase while the noise-correlation phase involves a quadratic complexity that implies a domain decomposition based parallelism. The software libraries identified for this stage at the moment are: ObsPy, which provides a number of signal processing tools but the scalability of those need to be carefully evaluated; the WHISPER tool suite that is implemented in the WHISPER ERC project and that has been specifically designed for actual seismic noise cross-correlation analysis of very large data sets. The latter is provided by WHISPER for a VERCE implementation through the ongoing collaboration with VERCE.

2 First training applications and documentation

As a result of this ongoing work and analysis, NA2 and NA3 have identified a number of training applications in relation with the development of the VERCE environment. At this stage those are divided according to the use cases' implementation strategy. It will focus on the software and libraries identified for the VERCE use case implementation strategy and the two demo cases that have been designed through the collaboration between NA2, JRA1, JRA2 and SA2. After a first training session focussed on the DISPEL environment, the next training sessions to be organized before the end of 2013 will focus on a number of well documented libraries and software that are integrated within the VERCE platform.

The first training applications have been identified and the training sessions will have to be finalized in collaboration with NA3, SA3 and SA1, together with a number of projects VERCE is collaborating with.

2.1 CPU-intensive training applications and documentation

The CPU-intensive training application will be based on the implementation of the use case described earlier Figure 1 and will focus on the software suite identified in the previous section.

In terms of software this will include:

SPECFEM3D : The spectral element code SPECFEM3D[12] has already set-up specific tutorials and various set-ups[13]. This code has already been integrated by SA2 and is available on a number of HPC infrastructures available to VERCE (LRZ and CINECA) as well as on local clusters (INGV, IPGP, ULIV).

CUBIT : The meshing CUBIT[14] tool suite used by SPECFEM3D, and developed at Sandia National Labs, already has specific documentation[15]. This software has already been integrated by SA2 and is available on a number of HPC infrastructures and local clusters (INGV, IPGP, ULIV).

SCOTCH **and** METIS : The SCOTCH[17] and METIS[18] packages are graph and hypergraph parallel partitioning tool suites used by SPECFEM3D. They are integrated by SA2 on the VERCE platform and available on a number of local cluster resources (INGV, IPGP, ULIV).

The aims of this training use case are:

[12] http://www.geodynamics.org/cig/software/specfen3D
[13] http://www.geodynamics.org/cig/software/specfem3d/tutorials
[14] http://cubit.sandia.gov
[15] http://cubit.sandia.gov/documentation.html and tutorials [16]
[17] http://www.labri.fr/perso/pelegrin/scotch/
[18] http://glaros.dtc.umn.edu/gkhome/views/metis

- To provide a training on the use of this wave simulation tool suite through examples including the setup of earth model meshes and analysis of the synthetic waveforms generated by the simulations. This will include mesh generation and parallel graph partitioning in support of the domain decomposition. Another aspect will concern graph partitioning of the input mesh.

- To provide a training on the VERCE implementation starting from an initial set of input data: earthquake events, parameters and earth models, and launching simulations on different computing infrastructures in order to prove the versatility of the VERCE platform. The workflow will create an input file to submit to HPC infrastructures (LRZ, CINECA) and possibly a local cluster (INGV). The creation of the input file will be generic and compliant with different types of solvers already deployed and available in the current platform. After the submission of the job the workflow will take care of the retrieval and shipment of the results to the assigned resource. Functionalities like monitoring, exception and error handling will be implemented in a minimal version. The automation of the overall process achieved by means of the VERCE workflow engine can be already seen as an added value compared to the previous common practices.

Those training use cases will involve the SPECFEM3D developers team, and advanced users within the VERCE consortium. VERCE will also foster collaboration with the ITN QUEST [19] and the US CIG [20] projects with whom VERCE is collaborating. This will broaden the impact of these training sessions and promote the VERCE workflow environment.

2.2 Data-intensive training applications and documentation

The training applications will be organized around the two main stages of the data-intensive use case: data ingestion and data access; data pre-processing and noise correlation analysis. It will be in synergy with the workflow implementation progress of the data-intensive use case as detailed in the D-SA3.2.1 report.

2.2.1 Data ingestion and data access

This stage involves data storage, indexing and discovery across the VERCE data-intensive platform is still in development. This includes raw data ingestion and extraction from the raw data of metadata to be stored through a dedicated metadata repository database, as well as the distribution and the synchronization of the raw data on several nodes adopting virtual file system technologies. However a number of software components, still in discussion with JRA2, SA3 and SA2, have been identified at this stage:

ObsPy : ObsPy is a tool suite widely used by the international seismology community, e.g. a "swiss army knife" for seismological data. This tool suite provides reading, processing and visualization components dealing with most seismological raw data and metadata formats. ObsPy provides already very complete documentation and tutorials[21]. Therefore the ObsPy portal already provides a mine of information and documentation. These libraries are used by an enormous number of users and on the software portals a very thorough documentation with cook-book examples and tutorials can be found on their web portals.

iRODS : iRODS provides services for distributed and virtual file system. iRODS already provide documentation and tutorials[22]. It has been analyzed by JRA2 and SA3 that will be organizing a technical presentation on 26 April 2013 at IPGP to the VERCE consortium. iRODS has already been

[19] http://www.quest-itn.org
[20] http://www.geodynamics.org
[21] http://docs.obspy.org/tutorial/
[22] https://www.irods.org/index.php/Documentation

installed at a number of the VERCE nodes (UEDIN, IPGP, INGV, CINECA) and is a component of the EUDAT project.

MonetDB : MonetDB is a column-based SQL scientific database system well adapted for metadata extraction and storage. MonetDB already provides documentation and tutorials[23]. MonetDB is already installed at a number of the VERCE nodes (KNMI, UEDIN). It has been analyzed by JRA2 and SA3 that will be organizing a technical presentation on 26 April 2013 at IPGP for the VERCE consortium.

NA2 and NA3 will work with JRA2, JRA1 and SA3 in order to organize a training session on those software components, taking advantage of existing collaborations with MonetDB and EUDAT, together with concrete use cases developed at INGV, CINECA and KNMI.

2.2.2 Data pre-processing and noise correlation analysis

The training applications will be based on the first implementation of the data-intensive use case within the VERCE DISPEL workflow environment as detailed in the previous sections and in the D-SA3.2.1 report. This first implementation, and the PEs developed so far, makes use of a number of software components.

OBSPY : ObsPy provides a number of signal processing tools and as decribed in the previous section comes with an extensive documentation and tutorial material. ObsPy makes large use of the standard libraries NumPy and SciPy[24] and of matplotlib[25]. These libraries are used by a wide number of users and on the software portals a very thorough documentation with cook-book examples and tutorials can be found.

WHISPER : The ERC project WHISPER has developed an extensive tool suite for actual noise-correlation analysis of large data sets with particular attention on the performance and on the parallelization. This tool suite comes with a very extensive documentation[26]. This tool suite makes use of some functionalities provided by ObsPy as well as by the standard libraries NumPy and SciPy. The WHISPER suite has been ported on Grid infrastructure and has explored iRODS functionalities. Through the collaboration between WHISPER and VERCE, the WHISPER suite has been provided to VERCE for implementation within the VERCE DISPEL environment.

The aims of the training use cases are:

- To provide a training session on the use of the ObsPy libraries and functionalities for the pre-processing stage of the data-intensive application. NA2 and NA3 will work closely with SA3, as well as with the ObsPy project to which VERCE is actually contributing, in order to organize such a training session that will serve also as an introduction to Python programming.
- To provide a training session on the use of the WHISPER suite, based on actual scientific use cases of the WHISPER project. NA2 and NA3 will work closely with SA3 and the WHISPER project in order to organize a joint VERCE-WHISPER training session.
- To provide a training session based on the present data-intensive use case implementation within the VERCE DISPEL environment, integrating a number of PEs built in particular upon ObsPy and WHISPER functionalities. This session will be organized around the demo case that has been set-up in collaboration with SA3 and SA2.

[23] http://www.monetdb.org/Documentation
[24] http://www.scipy.org
[25] http://matplotlib.org
[26] http://code-whisper.isterre.fr/html/

3 Conclusions and next steps

All the software components identified in this report are being incorporated in the VERCE Knowledge base managed by NA3. The aim of the Knowledge base is to collect information required by the VERCE project and users of the VERCE platform. The collected knowledge can be accessed by linking to the relevant information websites. All the links are set up on the project page http://www.verce.eu/Training/KnowledgeBase.php.

The next report on M30 will include additional software enabled on the VERCE platform. It will also include an extensive review of the scientific requirements for the VERCE enabled applications. Those requirements are driven by the need to replicate and improve actual research practices so that the VERCE environment can be efficiently used by, and improve, these research practices when dealing with large data-sets or complex HPC use cases. A first requirement is the need for rearranging with a certain level of freedom some parts of the workflow. For instance, the definition of a processing pipeline has to be completely configurable, allowing to test different data processing strategies. The role of the scientific gateway is to provide an interactive GUI or a declarative interface that allows the selection of the components of the pipeline and their configuration. Moreover, it will provide a way to access the results to evaluate intermediate results after different stages. This should be achievable either via the gateway interface itself or by downloading some of the processed data. We note that this report is necessarily concise since it would have been inappropriate to report on software developed elsewhere and featuring already its own portal. A second requirement concerns the possibility of creating personalized PEs. These PEs can make use only of the libraries accepted and provided within the VERCE platform (SA2) and can perform any sort of computation. The scientific gateway should provide a sort of small development environment where the scientist can edit/upload his script and eventually test it with real data. This functionality opens several challenges in terms of security, performance and deployment of the new PEs within the infrastructure. The feasibility of this feature within the current project will have to be evaluated and discussed in detail. We should note also that the participants of VERCE, within NA2, have been main developers in the developments of both ObsPy and SPECFEM.