



D-SA2.2: VERCE platform integration: First release report of integrated services and tools

01/10/2012

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-SA2.2.pdf
Author(s): S.H. Leong (BADW-LRZ)
Location: <http://www.verce.eu/Repository/Deliverables/RP2/>
Type of document: Deliverable
Dissemination level: Public
Status: Final
Due date of delivery: 01/10/2012
Reviewer: G. Moguilny (CNRS-IPGP) and M. Atkinson (UEDIN)
Keywords : Evaluation, Integration, Testing, Tools, Services, Application codes, Software components, First release, Report

Version	Author	Date	Comments
0.1	S.H. Leong (BADW-LRZ)	08/08/2012	Initial draft for comments
0.2	S.H. Leong (BADW-LRZ)	20/09/2012	Final version
1.0	S.H. Leong (BADW-LRZ)	24/09/2012 - 25/09/2012	To update reviewer's comments

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.

Table of Content

1. First Release Report.....	5
1.1. Release Management Schedule	5
1.2. Requested components	6
1.3. Assignments of Evaluators/Testers and Resources	8
1.4. Evaluation and Testing	9
1.4.1. Component Generic Tests	9
1.4.2. Component Specific Tests	9
1.4.3. Jenkins Integration	10
1.5. Results & Recommendations	12
1.6. Documentations	15
2. Lessons Learned	16
2.1. Questionnaire	16
2.2. Improvements	16
3. Conclusion and Future Plan.....	17
3.1. Schedule of next cycle	17
Appendix A: RP2 - Release Management Schedule.....	19
Appendix B: Obspy - Generic Acceptance Test	20
Appendix C: Obspy - Specific Tests.....	24
Appendix D: Jenkins CI – Robot Framework Test Results	25
Appendix E: Release Management Process Review Questionnaire	26
References.....	32
Glossary and Links.....	33

List of figures

Figure 1. Release Management Schedule	5
Figure 2. Jenkin CI webpage.....	11
Figure 3. Jenkin CI Test Results	11

List of tables

Table 1. Requested components for evaluation and testing	8
Table 2. Assigned Resources and Testers of each component.....	9
Table 3. Transfer speed comparison (SCP and GridFTP).....	13

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 to be integrated on the VERCE Platform.

The main aim of this reporting period is to report on the first release of the integrated tools and services. The Plan-Do-Check-Act (PDCA) cycle is used to manage the release process. Each cycle requires approximately one year with two overlapping cycles that occur contemporaneously to facilitate a six-monthly release of the platform. This reporting period corresponds to the first completed PDCA cycle.

The first release successfully completed with the schedule timeframe. In this release, eight components are evaluated. They are:

Application codes

- SeisSol
- AxiSEM
- SPECFEM3D
- Ses3D
- Sec3D

Tools

- ObsPy
- GridFTP
- DISPEL Gateway
- OGSA-DAI

Three of these components, SeisSol, ObsPy and GridFTP, passed the generic, i.e. quality, and specific, i.e. functionality and performance, tests and were thus approved for release in this reporting period. The remaining five components were delayed until the issues found are resolved. Delayed components are planned to be re-evaluated in the upcoming cycles.

The Jenkins Continuous Integration (CI) Platform is installed in this cycle and the GridFTP test suite provided by IGE as per the MoU in the last reporting period is integrated. More component tests are expected to be integrated on the Jenkins CI platform in upcoming cycles.

A questionnaire to review the release management procedure was prepared and feedbacks were gathered. The summarised result is shared in this report. Improvements are planned or are already in place for the next cycle to address the observed issues. Finally, the schedule for the second release is circulated to prepare for the next release cycle.

1. First Release Report

The first PDCA cycle, corresponding to the first release of integrated services and tools, was completed at month 12, September 2012, of the project. In order to coordinate this release, a release management schedule was prepared to ensure that the process was clear to each of the work packages that were involved. A total of eight application codes and four tools were submitted by the JRAs for evaluation. Three of the application codes were prioritised by JRA1, as per requested by SA2, to be evaluated first. In total, five application codes and four tools were evaluated in this reporting period. One application code, SeisSol, and two tools, ObsPy and GridFTP, passed the generic, i.e. quality, and specific, i.e. functionality and performance, tests and were thus approved for release in this reporting period. The remaining application codes and tools were returned to the developers for improvements that require significant time to implement. They are planned to be included in the next release. The details of this release were further elaborated in the following subsections.

1.1. Release Management Schedule

The work performed in these six months can be summarised in Figure 1. Due to the fact that this is the first release, two additional weeks in May 2012 were given to the JRAs for them to get used to the process and to learn to fill up the form to request for a component to be evaluated. All evaluation requests were received on time by 15th May as per the schedule.

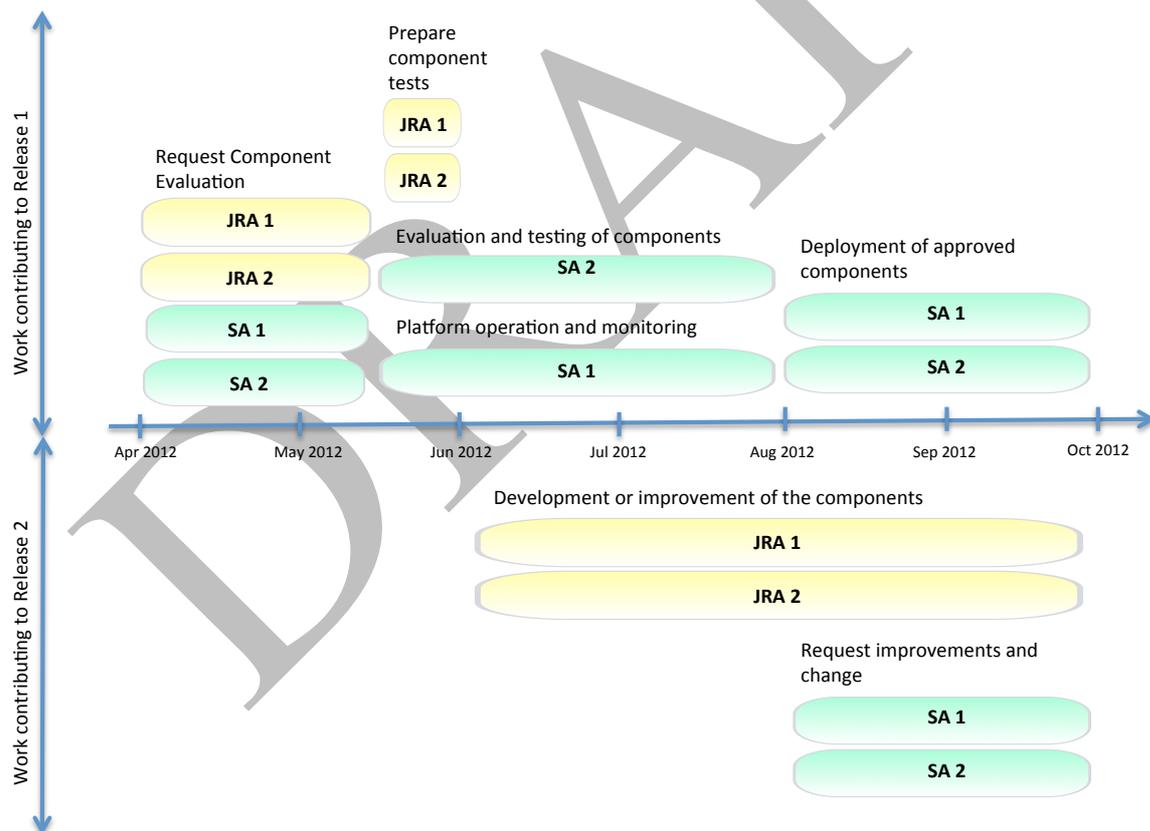


Figure 1. Release Management Schedule

An ad-hoc meeting for SA2 was set up on 16th May to assign the components to the members of SA2. The evaluation and testing of the components officially commenced on the 17th of May until the end of July. During this period, SA1 could operate and monitor the VERCE testbed while the JRAs were given 2 weeks

to prepare specific tests for their responsible components. The remaining time was assigned for the development and improvement of their responsible components for additional features and fixes for the next release. During this time, SA2 might contact the JRAs for minor fixes on the current proposed release. The evaluation and the testing of the components completed at the end of July.

In the beginning of August, SA1 was contacted by SA2 via email to share the information of this release. Information included which components were approved and which were not and technical documentations written by SA2 members during their evaluation. Information on the evaluated components that were not approved in this release was also shared. From August until the end of September, SA1 coordinated the deployment of the approved components on the VERCE testbed. During this time, SA2 will provide support when necessary. Simultaneously, SA2 will feedback the issues on the components that are not approved to the respective JRAs and the developers. Recommendations on how to improve the application are shared with the JRAs. The developers in the JRAs continue to work on improving their codes during this period. The deployment of the approved components should complete at the end of September.

1.2. Requested components

Eight application code requests and one tool request, and three tool requests were received from JRA1 and JRA2 respectively. A summary of each requested component is shown in Table 1. Detailed information about each component is available at http://www.verce-project.eu/projects/verce1/wiki/RP2_Evaluation_and_Tests.

Component	Version	Type	Purpose	Submitted by
SeisSol ¹	1.0	Application code	Calculation of wavefield propagation with ADER-EG scheme	JRA1
AxiSEM ²	-	Application code	Calculation of 3D wave propagation in spherically symmetric 2D lateral heterogeneities	JRA1
SPECFEM3D ³	2.0.1	Application code	Simulate forward and adjoint coupled acoustic-(an)elastic seismic wave propagation on arbitrary unstructured hexahedral meshes	JRA1
SPECFEMGLOBE ⁴	5.1.3	Application code	Calculation of wavefield propagation with ADER-DG scheme	JRA1
COMCOT ⁵	1.7	Application code	Tsunami modelling	JRA1

¹ <http://www.geophysik.uni-muenchen.de/~kaeser/SeisSol/>

² <http://www.seg.ethz.ch/software/axisem>

³ <http://www.geodynamics.org/cig/software/specfem3d>

⁴ <http://www.geodynamics.org/cig/software/specfem3d-globe>

⁵ <http://ceeserver.cce.cornell.edu/pll-group/comcot.htm>

RegSEM⁶	4.2	Application code	Elastic wave propagation on regular hexahedral mesh with possible 3D interfaces such as Moho and free surfaces	JRA1
Ses3D⁷	-	Application code	Calculation of wavefield propagation with spectral element method in spherical coordinates	JRA1
Sec3D⁸	-	Application code	A programme package for the simulation of elastic wave propagation in 3D Cartesian earth models	JRA1
ObsPy⁹	core 0.7.1, mseed 0.7.0, imaging 0.7.0, sac 0.7.0, gse2 0.7.0, signal 0.7.0, neries 0.7.0, seisan 0.5.1, arclink 0.7.1, datamark 0.1.0, db 0.7.0, earthworm 0.1.0, fissures 0.4.7, iris 0.7.0, realtime 0.1.0, seg2 0.7.0, segy 0.5.2, seishub 0.5.1, sh 0.5.2, taup 0.7.0, wav 0.5.1, xseed 0.7.0, seedlink 0.0.4	Tool	Handle and process seismological data	JRA1
GridFTP¹⁰	6.14 from GT5.2.2	Tool	Efficient transfer of large data files between remote locations	JRA2
Dispel Gateway¹¹	-	Tool	Delegation and implementation of logical workflows onto distributed compute resources	JRA2

⁶ <http://www.ipgp.fr/~paulcup/RegSEM.html>

⁷ <http://www.geophysik.uni-muenchen.de/Members/fichtner/ses3d>

⁸ <https://svn.geophysik.uni-muenchen.de/trac/sectrev>

⁹ <http://obspy.org/>

¹⁰ <http://www.globus.org/toolkit/docs/latest-stable/gridftp/>

¹¹ <http://sourceforge.net/apps/trac/admire/wiki/GatewayInstallation>

OGSA-DAI¹²	4.2	Tool	Execution of workflows accessing distributed data resources	JRA2
------------------------------	-----	------	---	------

Table 1. Requested components for evaluation and testing

Due to the personnel restrictions in SA2, SA2 had requested JRA1 to prioritise the application codes that had to be evaluated. Five out of eight of the application codes, SeisSol, AxiSEM, SPECFEM3D, Ses3D and Sec3D, and all four tools were evaluated in this reporting period.

1.3. Assignments of Evaluators/Testers and Resources

The assignment of evaluators is made to leverage on the expertise of each member of SA2 while the assignment of the resources to evaluate each component is made mainly based on the characteristics of the components and where the components are expected to be installed as per the scientific use cases that are provided by NA2 and the JRAs.

Component	Assigned Resource	Assigned Tester
SeisSol	SuperMIG (LRZ)	LRZ: Siew Hoon Leong and Gilbert Brietzke
	PLX (CINECA)	SCAI: Andre Gemünd
AxiSEM	EGI Cluster (SCAI)	SCAI: Andre Gemünd and Michael Schnell
	Personal Resource (SL6)	SCAI: Michael Schnell
SPECFEM3D	SuperMIG (LRZ)	IPGP: David Weissenbach LRZ: Siew Hoon Leong (supporting)
	PLX (CINECA)	LRZ: Siew Hoon Leong CINECA: Graziella Ferini (supporting)
	SuperMUC (LRZ)	LRZ: Siew Hoon Leong
Ses3D	SuperMIG (LRZ)	LRZ: Siew Hoon Leong
	EGI Cluster (SCAI)	SCAI: Andre Gemünd
Sec3D	SuperMIG (LRZ)	LRZ: Gilbert Brietzke
	EGI Cluster (SCAI)	SCAI: Michael Schnell
ObsPy	EGI Cluster (SCAI)	SCAI: Andre Gemünd
	EGI Cluster (IPGP)	IPGP: David Weissenbach
	SuperMIG (LRZ)	LRZ: Siew Hoon Leong
	Virtual Machine (LRZ)	LRZ: Siew Hoon Leong
GridFTP	SuperMIG (LRZ)	LRZ: Siew Hoon Leong
	Linux Cluster (LRZ)	LRZ: Siew Hoon Leong

¹² <http://sourceforge.net/apps/trac/ogsa-dai/>

Dispel Gateway	EDIM1 (EDIN)	EDIN: Paul Martin
	EDIM1 (EDIN)	KNMI: Luca Trani and Alessandro Spinuso
	Institutional Resource (ULIV)	KNMI: Alessandro Spinuso ULIV: Xiao Wang (supporting)
OGSA-DAI	EDIM1 (EDIN)	EDIN: Iraklis Klampanos
	Institutional Resource (ULIV)	KNMI: Alessandro Spinuso ULIV: Xiao Wang (supporting)

Table 2. Assigned Resources and Testers of each component

Table 2 provides a summarised view of the assigned resources and testers for each component. For each component, at least two resources and testers were assigned. Partners who were not a member of SA2 were also contacted for support, e.g. ULIV, when their resources were assigned for the evaluation and testing of a component.

1.4. Evaluation and Testing

The evaluation and testing phase commenced on the 17th of May until the end of July. To ensure that evaluation and testing could take place efficiently and effectively, a JRA1 representative was invited to attend SA2 biweekly Skype call when required.

1.4.1. Component Generic Tests

The evaluation and testing phase began with the generic acceptance tests that were meant to check the quality of each component. An example of the generic tests conducted for the component tool, ObsPy, is shown in Appendix B. Each SA2 member was in close contact with the request submitter to communicate issues and difficulties. Many components faced issues resulting in failed tests in this stage. The request submitters were contacted immediately. SA2 provided recommendation to the request submitters on how to come up with a quick resolution to the failed tests. Common issues found were missing documentations, no release version numbers or missing license or copyright notices. Most of the failed tests in this stage could be quickly resolved.

1.4.2. Component Specific Tests

Once the generic acceptance tests were completed, the request submitters were contacted for the test examples required for the specific tests of each component. The assigned tester then installed the component on the assigned resource. If the assigned tester was not a staff member from the assigned resource provider, a staff member from the assigned resource provider is contacted and support was requested.

While installing each component, SA2 members recorded their specific installation procedures, specific resource dependencies, installation recommendations, etc. A copy of the particular version of approved evaluated component and its test examples were kept in the VERCE's repository¹³ for future retrieval. The specific tests conducted for each approved components are summarised below.

¹³ <http://www.verce-project.eu/projects/vercel/repository/show/verce/All/SA/SA2/RP2Components>

SeisSol

In order to verify the correctness of the installation, a small test example was provided by JRA1 that could be run with e.g. 32 cores for less than an hour. After the small test run, bigger and longer runs were also set up. The correctness of the run and performance of the application was checked. In addition, the suitability of providing SeisSol as an application module (service) was evaluated.

ObsPy

To verify the correctness of the installation of ObsPy, the test suites provided by SciPy, NumPy and ObsPy were used. On all test installations, the test suites must complete without any unexpected issues. Additionally, two python scripts that use the Matplotlib and ObsPy modules were ran to verify the installation. An example of the specific tests documentation in the request form for the component tool, ObsPy, is shown in Appendix C.

GridFTP

GridFTP was verified by using the test suite that is provided by IGE as per the MoU that was signed in the last reporting period. The tests in the test suite were based on the tests defined by SA2 in the last reporting period. IGE uses the Robot Framework¹⁴ from Google code to implement their test suites. SA2 integrated the framework on our Jenkins Continuous Integration¹⁵ (CI) application. More information about our Jenkins integration is elaborated in the next section.

1.4.3. Jenkins Integration

Jenkins CI is installed as a web application on a virtual machine hosted at LRZ to support the building and testing of components to be integrated in the VERCE platform. The web frontend allows access only to registered EUGridPMA certificate owners.

The robot framework test suites provided by IGE are used to evaluate most of the VERCE resources that provide the GridFTP services as can be seen in Figure 2. Jenkins provides a convenient summary of the test results, as shown in Figure 3 and maintains a copy of the original robot framework test results and logs, as shown in Appendix E.

¹⁴ <http://code.google.com/p/robotframework/>

¹⁵ <http://jenkins-ci.org/>

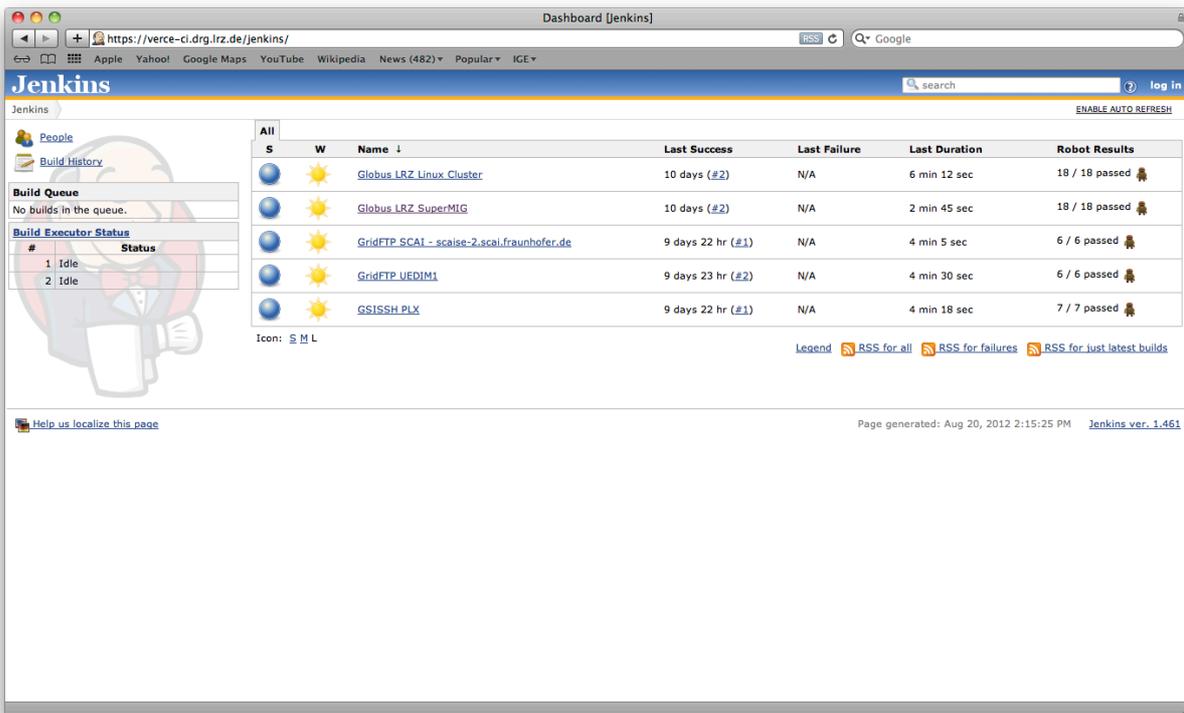


Figure 2. Jenkin CI webpage

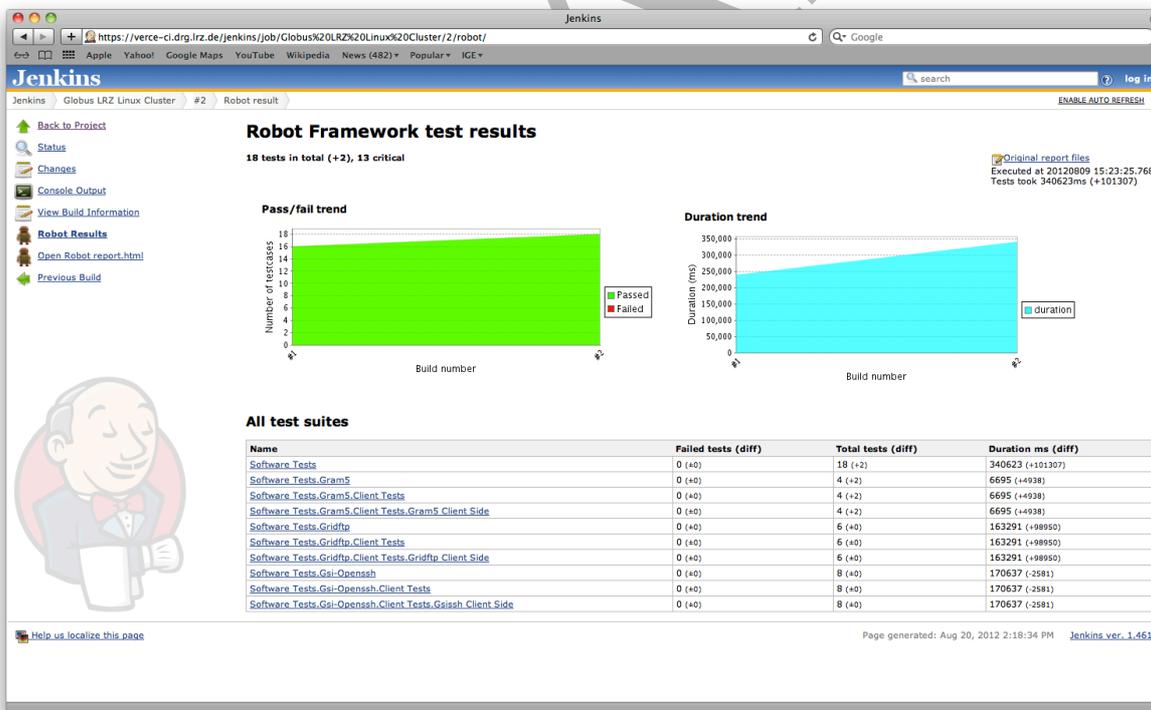


Figure 3. Jenkin CI Test Results

More components integration, e.g. GSI-SSH, GRAM and ObsPy, on the Jenkins framework is planned in the future.

1.5. Results & Recommendations

Approved components

Three out of the nine evaluated components were approved for release after the evaluation and test. The three components that were approved are:

1) SeisSol

Both HPC resources available on the VERCE testbed were selected as test resources for this component. During the generic tests, a number of improvement requests were sent to the SeisSol development team to improve the quality of the software, e.g. to provide a license or copyright notice, to include a user documentation and to provide a release note and version number. The developers managed to provide a quick fix to these issues. As such, the evaluation and test progressed without any significant delay. SeisSol's performance check using a bigger number of cores, >1000 cores, on SuperMIG showed that the code was within the acceptable performance range, >3% peak performance. As such, the application code was approved by all of its assigned testers to be released. Since SeisSol is a HPC code, it is recommended to be installed on all HPC resources as an application module. SeisSol can also be installed on any resource that supports Message Passing Interface (MPI).

2) ObsPy

ObsPy was assigned to be evaluated on one virtual machine acting as a personal resource, e.g. laptop, of a user, two EGI and one HPC resources to ensure that most of the possible resources that are expected to support ObsPy are evaluated. ObsPy is dependent on python and a number of python libraries, e.g. SciPy¹⁶ and NumPy¹⁷, which are in turn dependent on Mathematical libraries like LAPACK¹⁸ and BLAS¹⁹, and an interface like ATLAS²⁰.

On the virtual machine, a SLES 11 Linux machine, the installation is straightforward since the standard installation procedure is applicable. The installation procedure on most departmental and institutional resources, with Linux operating systems, should be performed in a similar manner. As such, it is highly recommended for such resources to install ObsPy where applicable.

On EGI resources, the required version of Python that ObsPy uses is different from the system version in the widely used SL5. Many EGI systems are still using SL5 or its variants and an upgrade to SL6 will require some time before it can be coordinated. As such, ObsPy cannot be easily installed on both EGI's testbed resources even though the standard installation procedures are applicable. The recommendation for the SL5 resource providers is thus to install ObsPy and its dependencies in the EGI's VO Software area. This allows ObsPy's required Python version to be installed without removing the system version.

On HPC resources, optimised high performance computing libraries like MKL²¹ and ESSL²² are typically used. As such, ObsPy and its dependencies have to be installed in a non-standard manner with these

¹⁶ <http://www.scipy.org/>

¹⁷ <http://numpy.scipy.org/>

¹⁸ <http://www.netlib.org/lapack/>

¹⁹ <http://www.netlib.org/blas/>

²⁰ <http://math-atlas.sourceforge.net/>

²¹ <http://software.intel.com/en-us/articles/intel-mkl/>

optimised libraries. On the HPC resource, SuperMIG, MKL was used to compile SciPy and NumPy. When Intel compilers, `icc` and `ifort`, were used, the compilation completed without any issues. However, a check on the installation shows many numerical accuracy issues within NumPy and SciPy. A second installation using a combination of `gcc`, `gfortran` and `ifort` finally resulted in a version that passed the tests provided by the SciPy and NumPy developers. The procedure on how to install ObsPy with MKL is documented in the SA2's results and documentation wiki. Although Python is not expected to take full advantage of highly optimised resources like HPC resources, the fact that it is widely used by many researchers, especially the seismologists, leads to SA2's recommendation for the HPC resource providers to install Python and ObsPy as tool modules on their HPC resources.

3) GridFTP

GridFTP was evaluated on an EGI resource, a HPC resource and a departmental resource. GridFTP is picked as one of the initial portfolio of software components in the last reporting period due to its availability on most of the existing e-Infrastructure. Detailed specific tests are carried out in this reporting period on the above selected resources and other resources in VERCE that provide GridFTP as a service. The specific tests are carried out using the test suite provided by IGE as per the MoU that was signed in the last reporting period. During this period, SA2 shared the specific tests we had defined in the last reporting period with IGE. The defined tests are now a part of the GridFTP test suite provided by IGE.

Jenkins is configured to run the GridFTP test suite. The resources that were selected to evaluate GridFTP were added to the configuration and the specific GridFTP tests were performed. In addition, other known resources on the testbed that provide GridFTP service are also added. All resources pass the tests. One interesting observation from the tests is the varying transfers speed of the resources on testbed. In order to get more information about this, an ad-hoc test was conducted to transfer a 1GB file from a test machine at LRZ to each resource with secure copy (SCP) and GridFTP (with single and multiple streams). The results of the test are shown in Table 3. In the case of multiple GridFTP streams, only the best transfer speed and its corresponding number of streams is shared.

Machines	SCP	GridFTP with single stream	GridFTP with multiple streams (no. of streams)
SuperMUC	54 MB/s	200 MB/s	238 MB/s (3)
SuperMIG	60 MB/s	189 MB/s	217 MB/s (4)
LRZ Linux Cluster (lx64ia3)	49 MB/s	77 MB/s	55 MB/s (3)
LRZ Linux Cluster (lxgt2)	43 MB/s	92 MB/s	152 MB/s (4)
PLX (CINECA)	28 MB/s	11.87 MB/s	22.46 MB/s (3)
SCAI	-	19.43 MB/s	22.51 MB/s (2)
UEDIM1	9.2 MB/s	17.44 MB/s	16.84 MB/s (2)

Table 3. Transfer speed comparison (SCP and GridFTP)

The test results show that GridFTP is definitely one of the data strategies VERCE should adopt since it can significantly improve transfer rates. As such, SA2 recommends all resources on VERCE testbed to install a GridFTP server. It is especially recommended for the data centres to switch from using anonymous FTP servers to anonymous GridFTP servers.

²² <http://www-03.ibm.com/systems/software/essl/index.html>

Delayed components

In spite of the effort to attempt to fix all issues as quick as possible, five components did not meet the criteria required to be approved for release in this cycle. Most of the application code components require re-implementing parts of their code, which would require more time than is available until release.

1) AxiSEM

AxiSEM was evaluated only on Grid resources since it is not expected to scale. Compilation of the code fails on the current operating system, Redhat Enterprise Linux 5 (RHEL5) – a variant of SL5, at SCAI due to the system default gFortran version, 4.1.2. As such, an additional local resource is included in the evaluation to check and confirm that the newer version of SL6, which includes a newer version of gFortran, is free of this limitation. The compilation issue is fed back to the AxiSEM developers at the end of the evaluation period. A post-processing part of the code is still work in progress and has a known issue that the developers are currently fixing. As such, SA2 decided to delay the acceptance of this component until the post-processing code work is completed. In event that the upgrade to SL5 or RHEL6 on EGI resources are not completed by the next cycle in February 2013, a backup plan to recommend the installation of additional gFortran compilers >4.1.2 in the EGI's VO Software area will be proposed.

2) SPECFEM3D

SPECFEM3D, one of the most widely used computation application codes by seismologists, was evaluated on both HPC resources on the VERCE platform. A number of issues were found by the testers during the evaluation. The issues are

- Input parameters in *.h file that requires a recompilation
- Memory issues that require system stack size to be set to unlimited
- Output at the last stage of the code could be serial instead of parallel

Since the SPECFEM3D developers are planning to release of a significantly changed version to exploit GPGPU early next year, they are unable to fix the issues or problems we found on the current released version.

As such, SA2 worked very closely with the JRA1 members to attempt to find a workaround to above issues. The JRA1 members conducted tests to check if the default value set in the code is usable for most simulation scenarios. The findings show that only the amplitude and phases are affected. The phase effect disappears when a low frequency filter is applied. As such, supporting the default version should be sufficient and acceptable for most users.

The memory issue occurs in one particular example provided with the application code. As per documented, if the stack size is set to unlimited, the simulation run can complete without any issues. However, this is not a recommended setup and additional tests are planned with the compiler option, `-heap-array [size]`, to use the heap memory instead of the stack memory for bigger arrays. This should solve the problem. More tests need to be conducted to confirm this and to ensure that performance is not significantly compromised.

One of the testers observed that the final output performed seem to be serial instead of parallel. As such, additional tests are planned to check this.

Finally, SPECFEM3D supports both single and double precision versions. Typically only the single precision version is used since the double precision version uses more resources without any significant precision benefits. However, for the purpose of completeness, tests are also planned to ensure that the double precision version produces similar output as the single precision version.

Due to the issues mentioned above, the release of SPEC3D is delayed until we complete the above tests.

3) Ses3D (Spherical)

Ses3D was evaluated on a HPC resource and an EGI resource. A quick test on the HPC resource shows that the application has to be recompiled for different problem sizes. This implies that it cannot be offered as a HPC/EGI service. As such, a ticket is submitted to the developer to support dynamic allocation to avoid recompilation. The developer is currently working on this issue and the evaluation is delayed until the new version is available.

4) Sec3D (Cartesian)

Sec3D was selected to be evaluated on a HPC resource and an EGI resource. On the EGI resource, the tester observed that the code is always compute intensive, using almost 100% of the CPU resource. When the computation requires a large number of iterations, the application runs into a floating point exception. On the HPC resource, the application did not manage to scale to 1000 cores. The assigned tester found several areas to improve the performance of the code. The recommendations are recorded and shared with JRA1. The release of this code will be delayed until the above issues are addressed.

5) Dispel Gateway

The Dispel gateway was selected to be evaluated at two department resources. However due to the missing documentation, both testers were unable to complete their evaluations. The Dispel team has been informed. The evaluation of the Dispel gateway will continue in the next cycle.

6) OGSA-DAI

OGSA-DAI was selected to be evaluated at two departmental resources. However, the testers fed back that they were unable to evaluate the component without valid client tests. The OGSA-DAI developers have been informed and will ensure that valid client tests are included in the next release.

1.6. Documentations

All issues faced and solutions the SA2 team used during this reporting period were documented in the SA2's wiki and 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/RP2_Evaluation_Results.

2. Lessons Learned

2.1. Questionnaire

In order to improve the process of the release, a questionnaire to gather feedback from various working packages involve in the Management Release Process is prepared. The questionnaire can be found in Appendix E.

The result of the questionnaire shows that most members in the involved work packages know the process. The request form and defined tests in the release management process is considered as sufficient for its purpose and most know who to contact or have someone to contact in event of issues. Communication between the JRAs and SA2 is adequate in the last release while communication between SA1 and SA2 has room for improvement. Some respondents want more time for their work in each phase while others believe that they have adequate time. One respondent requests for flexibility in handling component that is still in beta stages.

2.2. Improvements

To address the issues raised in the questionnaire, improvements are planned in the next cycle. To better improve communication between SA1 and SA2, SA2 will attempt to coordinate with SA1 to organise common meetings. An earlier start or preparation should help to ease the time issue for all involved. SA2 has prepared the release schedule one month in advance for the next reporting period so that some members can have an earlier start. The flexibility issue can be handled on a case-by-case basis. An example would be the SPEC3D evaluation, where additional support and work took place in non-defined phases between the involved members. In the case of components in beta stages, test machines can be used to provide a test environment for components that need to be integrated but are not ready for production machines. As per required, SA2 can provide additional support to handle such situation.

Vacation and holiday season can also delay the evaluation process as we observed in this cycle. As such, all members are now encouraged to share their vacation plans as early as possible so that it can be included in the work plan.

At the end of the next evaluation period, a similar questionnaire will be prepared to check if the situation has improved.

3. Conclusion and Future Plan

The first release of the VERCE software components successfully completed within the scheduled timeframe. Three out of eight selected components were approved for release as a part of the VERCE platform. The remaining five components are expected to be included in the upcoming releases once the issues that were fed back are resolved.

A questionnaire was circulated among relevant work packages to evaluate the release management procedures. In general, the procedures are effective for managing the release. There is some room for improvement as observed by various partners. Plans are in placed, based on the feedback, to improve the procedure for the next cycle.

The release management schedule for the next release, refer to Section 3.1, is prepared and shared with involved work packages.

3.1. Schedule of next cycle

The schedule for the next cycle, 1st October 2012 to 31st March 2013, for the respective work packages is summarised below. This schedule was shared with the SAs and JRAs on 20th August 2012.

JRA2

- [1st Oct – 31st Oct 2012]: 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.
- Until 31st Oct to define tests to perform (in particular, functionalities that will be used) for external tools and in-house development components
- [1st Nov 2012 – 31st Mar 2013] To work on developing the required workflow related tools

JRA1

- [1st Oct – 31st Oct 2012]: 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.
- Until 31st Oct to provide the tests (including input files)
- [1st Nov 2012 – 31st Mar 2013] To work on developing the application codes

SA2

- [1st Oct – 31st Oct 2012] To identify missing features and functionalities and thus suggest addition tools to be evaluated. Administrative work to prepare for the next release.
- [1st Nov – 6th Nov 2012] To select the team for each component and begin installation on the selected resources
- [7th Nov 2012 – 31st Jan 2013] To work on evaluating and integrating the selected components.
- [1st Feb – 31st Mar 2013] 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 cleared the evaluation and tests) to SA1 and assist SA1 in deploying the approved components.

SA1

- [1st Oct – 31st Oct 2012] To identify missing features and functionalities and thus suggest addition tools to be evaluated. Administrative work to prepare for the next release.
- [1st Nov – 28th Feb 2013] Operate and monitor the scientific platform and the tools and services running on it. To collect service information

- [1st Mar – 31st Mar 2013] To work on deploying the released components provided by SA2. To check the service information collected and provide feedback/new requirements to the JRAs.

DRAFT

Appendix A: RP2 - Release Management Schedule

Reporting Period 2: 1st April 2012 - 30th September 2012

JRA2

- [1st April - 15th May 2012]: 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.
- Until 31st May to define tests to perform (in particular, functionalities that will be used) for external tools
- Until 31st May to provide functionality tests for self developed tools
- [1st June - 30th September 2012] To work on developing the required workflow related tools

JRA1

- [1st April - 15th May 2012]: 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.
- Until 31st May to provide the tests (including input files)
- [1st June - 30th September 2012] To work on developing the application codes

Please kindly note that you are only supposed to fill up the "Codes and Tools Form" sheet of the Request Form.

SA2

- [1st April - 15th May 2012] To identify missing features and functionalities and thus suggest addition tools to be evaluated. For this reporting period, it is a good opportunity to work on the automatic test framework.
- [15th May - 31st May 2012] To select the team for each component and begin installation on the selected resources
- [1st June - 31st July 2012] To work on evaluating and integrating the selected components.
- [1st August - 30th September] 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 cleared the evaluation and tests) to SA1 and assist SA1 in deploying the approved components.

SA1

- [1st April - 15th May 2012] To identify missing features and functionalities and thus suggest addition tools to be evaluated.
- [15th May - 31st July 2012] Operate and monitor the scientific platform and the tools and services running on it. To collect service information.
- [1st August - 30th September] To work on deploying the released components provided by SA2. To check the service information collected and provide feedback/new requirements to the JRAs.

[Source from : http://www.verce-project.eu/projects/verce1/wiki/Release_Management_Schedule#Reporting-Period-2-1st-April-2012-30th-September-2012]

Appendix B: Obspy - Generic Acceptance Test

Integration Process Checklist (EGI)	Description	Applicability	Pass/Fail	Comments
1 Download/Receive code/software		All	Pass	It is rather complicated to download obspy libraries if one is not using easy_install. Will recommend to the developers to perhaps have obspy as one zip and one can install it with one setup.py script.
2 Source code availability		Open source software	Pass	
3 Installation/Administration Documentation	Administration guide describing installation, configuration and operation	All	Pass	There is room for improvement especially when one has to install via source.
4 Functional Description	Code/Tool should provide a functional document describing its functionality	All	Pass	There is a readme.txt that in the source.
5 Release Notes	Include all changes in release: bug fixes and new features	All	Pass	E.g. obspy.core-0.7.1 Installed module directory: lib/python2.6/site-packages/obspy.core-0.7.1-py2.6.egg/obspy/core/CHANGELOG.txt or in the source directory: ./obspy/core/CHANGELOG.txt
6 User Documentation	A document/readme describing how to use it.	All	Pass	Yes, there is a online documentation that is very comprehensive. E.g. http://docs.obspy.org/packages/obspy.core.html
7 Online help (man pages)	End user command line tools must include man pages or online help	End user command line tools	Pass	online help
8 Software License	Code/Service should have a license to allow usage on shared resources (EGI/PRACE infrastructure) For Open Source products, compatible licenses are those accepted by the Open Source Initiative and categorised as "Popular and widely used or with strong communities": <ul style="list-style-type: none"> - Apache License, 2.0 (Apache-2.0) - BSD 3-Clause "New" or "Revised" license (BSD-3-Clause) - BSD 3-Clause "Simplified" or "FreeBSD" license (BSD-2-Clause) - GNU General Public License (GPL) - GNU Library or "Lesser" General Public License (LGPL) - MIT license (MIT) - Mozilla Public License 1.1 (MPL-1.1) - Common Development and Distribution License (CDDL-1.0) 	All	Pass	GNU Lesser General Public License, Version 3 (LGPLv3)

		<ul style="list-style-type: none"> - Eclipse Public License (EPL-1.0) Other licenses accepted by the Open Source Initiative and listed as “Special Purpose” are compatible with the infrastructure (when applicable): - Educational Community License - IPA Font License (IPA) - NASA Open Source Agreement 1.3 (NASA-1.3) - Open Font License 1.1 (OFL-1.1) <p>Any other license, and non Open Source products will have to be evaluated.</p>			
9	Service reference card	<p>For each of the services that a product runs, document its characteristics with a reference card. The document must exist and contain the following information for each service:</p> <p>ServiceName</p> <ul style="list-style-type: none"> - Description: Description of the service - Init scripts: List of init scripts for the service, expected run levels - Daemons: List of daemons needed for the service - Configuration: List of configuration files used by the service - Logs: List of log files used by the service - Open ports: List of ports the service uses - Cron: List of crons used by the service <p>Other information: Any other relevant information about the service.</p>	Services	Not Applicable	
10	API Documentation	Public API of product/appliances must be documented.	Open source software	Pass	E.g. http://docs.obspsy.org/packages/obspsy.core.html#classes-functions

	Main Tests	Additional Info		Pass/Fail	Comments
1	Release changes testing	All the changes in a release should be tested, especially bug fixes	All	Not Applicable	Can't test since there is no release notes
2	Integration and Functionality Test	Interaction with other software modules should be tested.		Pass	ObsPyTest in specific tests
3	Regression Tests	Ensure that old bugs are still resolved and if new bugs appear.		Not Applicable	First version to install and no unit tests from developers

4	Backwards compatibility	Minor/Revision releases of a product must be backwards compatible.	All	Not Applicable	
5	Service Control and Status	Services run must provide a mechanism for starting, stopping and quering the services following the OS init scripts conventions	Service	Not Applicable	
6	Service logs	All services should create log files where the service administrator can trace most relevant actions taken.	Service	Not Applicable	

	Service tests	Additional Info		Pass/Fail	Comments
1	Service Reliability	Services must maintain a good performance and reliability over long periods of time with normal operation. Service must not show performance degradation during a 3-day period. The most important parameters to check are: - stable memory usage - throughput and/or response times remain stable during the period of activity (they should be as good or better than at the beginning of the test for similar requests)	Service	Not Applicable	
2	Service Robustness	Services should not produce unexpected results or become uncontrollable when taxed beyond normal capacity. Services taxed beyond normal capacity: - should not become unresponsive to normal start/stop operations - must be able to start after a forceful stop - must not expose (potentially sensitive) memory contents to other processes - must not leave sensitive data in world-readable files must not accept connections that would be refused under normal operating conditions	Service	Not Applicable	

	Security tests	Additional Info		Pass/Fail	Comments
1	World writable files	Products must not create world-writable files or directories. - Start the service under test and initiate a standard client to use the service. Then using lsof check for open files by the service and test each one for world writable files. - Check for the files installed by the package and check each file/folder for world writable permissions	All	Pass	
2	System initialisation Files	System initialisation files (/etc/rc* , /etc/init.d/*, /etc/rc?.d) of the service should be protected such that only the root/service owner can write to	Service	Not Applicable	

	them.		
--	-------	--	--

	Performance tests	Additional Info		Pass/Fail	Comments
1	Ease of installation	How easy is it to install this component. Rating: 1-5 (1 is very easy and 5 is very difficult)	All		4 It is really tough on superMUC since we need to install as a module using intel compilers and mkl and cannot do it as root.
2	Performance of application codes	How many percent of peak performance. <u>Rating 1-5</u> 0-1%: 5 1-2.9%: 4 3-5%: 3 5%-10%: 2 >10%: 1	application codes	Not Applicable	

DRAFT

Appendix C: Obspy - Specific Tests

	Obspy Tests	Description	Pass/Fail	Comments
1	python -c "import numpy; numpy.test(verbose=2)"	Unit tests of numpy	Pass	Passed at SCAI. 2 failures on SuperMUC when compiled with icc. More information: http://www.verce-project.eu/projects/verce1/wiki/Results_Issues_and_Documentations#Obspy Passed at SuperMUC when compiled with gcc and gfortran
2	python -c "import scipy; scipy.test(verbose=2)"	Unit tests of scipy	Pass	6/7 runtime failures on superMUC when compiled with icc/fort. More information: http://www.verce-project.eu/projects/verce1/wiki/Results_Issues_and_Documentations#Obspy Segmentation fault at superMUC when compiled with gcc/gfortran. Passed at SuperMUC when compiled with gcc/fort
3	ObsPyTest	A test provided by Marek python MAIN.py http://www.verce-project.eu/projects/verce1/repository/show/verce/All/SA/SA2/RP2Components/PythonTests	Pass	EGI cluster at SCAI: Successful SuperMUC: Failed with socket.timeout: timed out during from Stations import Stations Firewall problem
4	matplotlib Test	A simple test to be runned with X11 python matplotlibTest.py http://www.verce-project.eu/projects/verce1/repository/show/verce/All/SA/SA2/RP2Components/PythonTests	Pass	SuperMUC: ok
5	python -c "import obspy.core; obspy.core.runTests()"	http://docs.obspy.org/packages/autogen/obsypy.core.scripts.runtests.html#module-obsypy.core.scripts.runtests	Pass	SuperMUC: All green with gnu version. The only tests that failed are those that require external access and were thus blocked by our firewall obsypy.mseed 0.7.* at EDIM1 and macbook - Seem to be a global lock problem. This is an implementation problem and not a obspy one.

Appendix D: Jenkins CI – Robot Framework Test Results

Test Details :

Name	Documentation	Tags	Crit.	Status	Message	Start / Elapsed
Software Tests - Grand - Client Tests - Grand Client Side - Access testcase	Simple authentication test	basic, client, critical, gram	yes	PASS		20120809 15:23:25.863 / 00:00:00.204
Software Tests - Grand - Client Tests - Grand Client Side - Simple Job Execution	Sends a simple /bin/date job on server using globusurn to all job managers	basic, client, critical, gram	yes	PASS		20120809 15:23:26.068 / 00:00:01.549
Software Tests - Grand - Client Tests - Grand Client Side - Simple job with file stage in	Sends a simple job on server with a file stage-in using globusurn to all job managers	basic, client, critical, filetransfer, gram	yes	PASS		20120809 15:23:27.616 / 00:00:02.994
Software Tests - Gridftp - Client Tests - Gridftp Client Side - Simple job with file stage out	Sends a simple job on server with a file stage-out using globusurn to all job managers	basic, client, critical, filetransfer, gridftp, guc	yes	PASS		20120809 15:23:30.613 / 00:00:01.948
Software Tests - Gridftp - Client Tests - Gridftp Client Side - GRIDFTP Test small file local2local	File transfer local to local of 10MB	basic, client, critical, filetransfer, gridftp, guc	yes	PASS		20120809 15:23:32.584 / 00:00:01.664
Software Tests - Gridftp - Client Tests - Gridftp Client Side - GRIDFTP Test small file local2remote	File transfer local to remote of 10MB	basic, client, critical, filetransfer, gridftp, guc, remote	yes	PASS		20120809 15:23:34.248 / 00:00:12.499
Software Tests - Gridftp - Client Tests - Gridftp Client Side - GRIDFTP Test small file remote2remote	File transfer remote to remote of 10MB	basic, client, critical, filetransfer, gridftp, guc, thirdparty	yes	PASS		20120809 15:23:37.173 / 00:00:23.145
Software Tests - Gridftp - Client Tests - Gridftp Client Side - GRIDFTP Test file restart	File transfer to remote with restart, 100MB	client, critical, filetransfer, gridftp, guc, restart	yes	PASS		20120809 15:24:09.893 / 00:00:28.165
Software Tests - Gridftp - Client Tests - Gridftp Client Side - GRIDFTP Test large file restart	File transfer remote to remote with restart, 100MB	client, filetransfer, gridftp, guc, restart, thirdparty	no	PASS		20120809 15:24:38.056 / 00:00:40.321
Software Tests - Gridftp - Client Tests - Gridftp Client Side - GRIDFTP Test large file local2remote, with parallel streams	File transfer local to remote of 200MB	client, filetransfer, gridftp, guc, large, parallel	no	PASS		20120809 15:25:18.381 / 00:00:57.497
Software Tests - Gsi-Openssh - Client Tests - Gsish Client Side - Access testcase	Runs a /bin/date on the server	basic, client, critical, gsi-openssh, gsish	yes	PASS		20120809 15:26:15.902 / 00:00:10.192
Software Tests - Gsi-Openssh - Client Tests - Gsish Client Side - GSISCP Test small file	Simple file copy 1M	client, critical, filetransfer, gsi-openssh, gsish	yes	PASS		20120809 15:26:26.095 / 00:00:30.903
Software Tests - Gsi-Openssh - Client Tests - Gsish Client Side - GSISCP Test large file	Simple file copy 100MB	client, filetransfer, gsi-openssh, gsish, long	no	PASS		20120809 15:26:38.056 / 00:00:47.804
Software Tests - Gsi-Openssh - Client Tests - Gsish Client Side - GSISSH Robustness	Start 1000 connections and compare the load before and after	client, gsi-openssh, gsish, long, robustness	no	PASS		20120809 15:27:44.803 / 00:00:41.048
Software Tests - Gsi-Openssh - Client Tests - Gsish Client Side - Login without certificate	Runs a /bin/date on the server without a certificate	client, critical, gsi-openssh, gsish, security	yes	PASS		20120809 15:28:25.653 / 00:00:10.116
Software Tests - Gsi-Openssh - Client Tests - Gsish Client Side - Login with different username	Tries to login as a different user (non default)	client, extra, gsi-openssh, gsish, multuser	no	PASS		20120809 15:28:35.972 / 00:00:10.188
Software Tests - Gsi-Openssh - Client Tests - Gsish Client Side - Login with non existent username	Runs a /bin/date on the server asking for a nonexistent username	client, critical, extra, gsi-openssh, gsish, security	yes	PASS		20120809 15:28:46.161 / 00:00:10.172
Software Tests - Gsi-Openssh - Client Tests - Gsish Client Side - Login with privileged username	Checks for login as root. Test succeeds if the login is unsuccessful	client, critical, gsi-openssh, gsish, security	yes	PASS		20120809 15:28:56.333 / 00:00:10.212

Log Report:

```

TEST CASE: GSISCP Test large file (non-critical)
Full Name: Software Tests.Gsi-Openssh.Client Tests.Gsish Client Side.GSISCP Test large file
Documentation: Simple file copy 100MB
Tags: client, filetransfer, gsi-openssh, gsish, long
Timeout: 1 minute 40 seconds
Start / End / Elapsed: 20120809 15:26:56.999 / 20120809 15:27:44.803 / 00:00:47.804
Status: PASS (non-critical)

KEYWORD: ige-library.Create Test File $(TMP_FILE), $(GSISCP_LARGE_FILE_SIZE)
Documentation: Creating a temp file for transfer
Start / End / Elapsed: 20120809 15:26:57.001 / 20120809 15:27:11.298 / 00:00:14.297

KEYWORD: $[output] = OperatingSystem.Run /bin/dd if=/dev/urandom of=${FILE} ibs=1M count=${FILESIZE}
Documentation: Runs the given command in the system and returns the output.
Start / End / Elapsed: 20120809 15:26:57.002 / 20120809 15:27:11.291 / 00:00:14.289
15:26:57.012 INFO Running command '/bin/dd if=/dev/urandom of=/tmp/IGF_TS_tomcat_gftp_test_file_1 ibs=1M count=100 2>&1'
15:27:11.292 INFO $(output) = 100+0 records in
204800+0 records out
104857600 bytes (105 MB) copied, 14.265 s, 7.4 MB/s

KEYWORD: OperatingSystem.File Should Exist $(FILE)
Documentation: Fails unless the given 'path' points to an existing file.
Start / End / Elapsed: 20120809 15:27:11.293 / 20120809 15:27:11.295 / 00:00:00.002
15:27:11.294 INFO File '/tmp/IGF_TS_tomcat_gftp_test_file_1' exists

KEYWORD: $(checkfilesize) = OperatingSystem.Get File Size $(FILE)
Documentation: Returns and logs file size as an integer in bytes
Start / End / Elapsed: 20120809 15:27:11.295 / 20120809 15:27:11.296 / 00:00:00.001
15:27:11.295 INFO Size of file '/tmp/IGF_TS_tomcat_gftp_test_file_1' is 104857600 bytes
15:27:11.296 INFO $(checkfilesize) = 104857600

KEYWORD: $(filesize_in_mb) = BuiltIn.Evaluate $(checkfilesize)/1048576
Documentation: Evaluates the given expression in Python and returns the results.
Start / End / Elapsed: 20120809 15:27:11.296 / 20120809 15:27:11.297 / 00:00:00.001
15:27:11.297 INFO $(filesize_in_mb) = 100

KEYWORD: BuiltIn.Should Be Equal As Numbers $(filesize_in_mb), $(FILESIZE)
Documentation: Fails if objects are unequal after converting them to real numbers.
Start / End / Elapsed: 20120809 15:27:11.297 / 20120809 15:27:11.298 / 00:00:00.001
15:27:11.298 INFO Argument types are:
<type 'int'>
<type 'unicode'>

KEYWORD: ige-library.Run Command $(GSISCP_CMD) $(TMP_FILE) $(DEFAULT_USERNAME_WITH_AT)$(HOST)$(TMP_FILE_REMOTE)
Documentation: Runs the specified command and checks if it failed
Start / End / Elapsed: 20120809 15:27:11.299 / 20120809 15:27:23.787 / 00:00:12.488

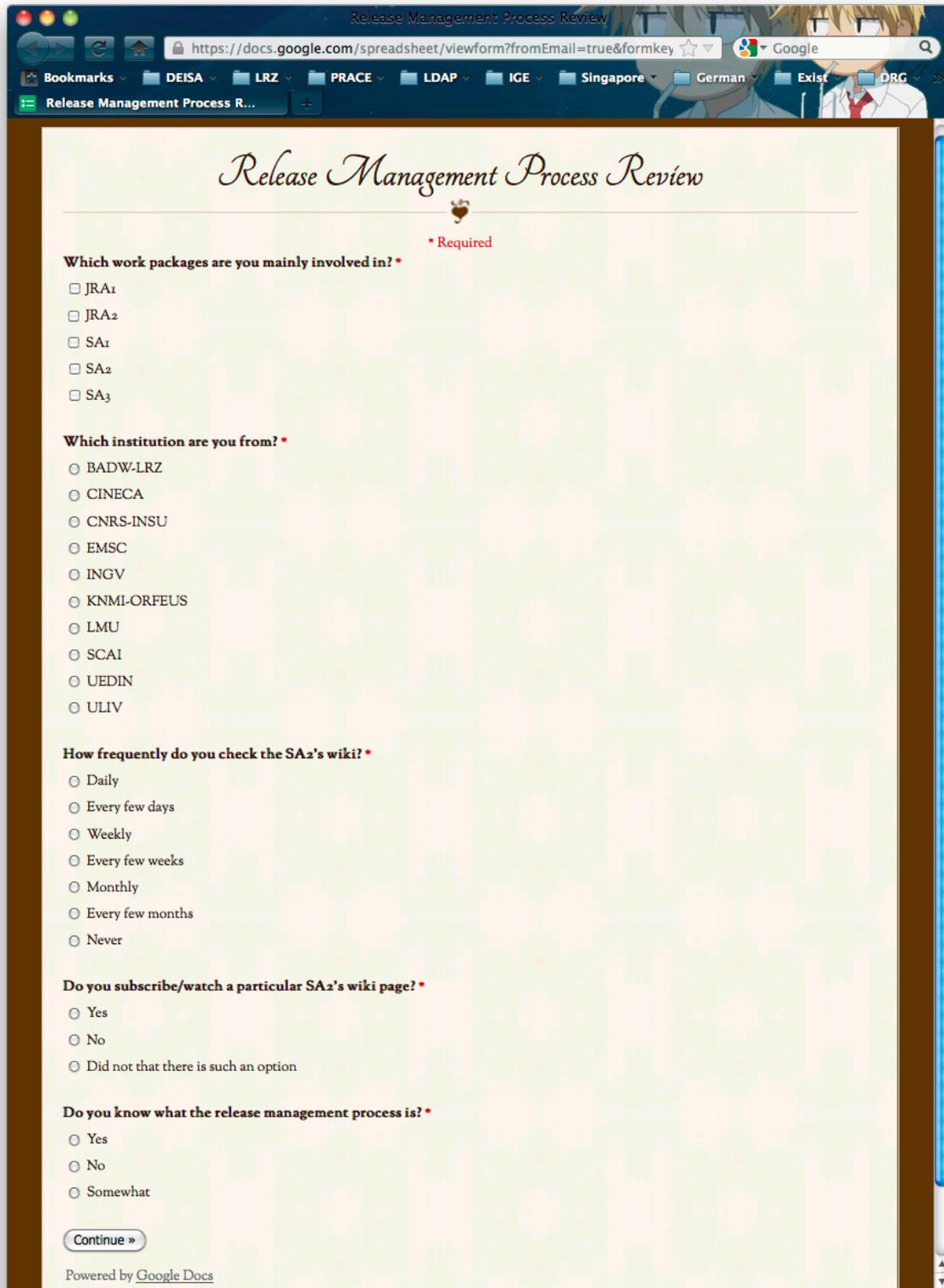
KEYWORD: $(rc_exec), $(cmd_out) = OperatingSystem.Run And Return Rc And Output $(CMD_STRING)
Documentation: Runs the given command in the system and returns the RC and output.
Start / End / Elapsed: 20120809 15:27:11.301 / 20120809 15:27:23.781 / 00:00:12.480
15:27:11.311 INFO Running command '/opt/globus-5.2.1/bin/gaiscp -P 2222 -o StrictHostKeyChecking=no -o PasswordAuthentication=no /tmp/IGF_TS_tomcat_gftp_test_file_1 104857600 /tmp/IGF_TS_tomcat_gftp_test_file_r 2>&1'
15:27:23.782 INFO $(rc_exec) = 0
15:27:23.782 INFO $(cmd_out) =

KEYWORD: BuiltIn.Log $(cmd_out)
Documentation: Logs the given message with the given level.
Start / End / Elapsed: 20120809 15:27:23.783 / 20120809 15:27:23.785 / 00:00:00.002
15:27:23.784 INFO

KEYWORD: BuiltIn.Should Be Equal As Integers $(rc_exec), 0
Documentation: Fails if objects are unequal after converting them to integers.
Start / End / Elapsed: 20120809 15:27:23.785 / 20120809 15:27:23.786 / 00:00:00.001
15:27:23.786 INFO Argument types are:

```

Appendix E: Release Management Process Review Questionnaire



The image shows a screenshot of a Google Docs spreadsheet form titled "Release Management Process Review". The browser address bar shows the URL: <https://docs.google.com/spreadsheet/viewform?fromEmail=true&formkey>. The form is displayed within a window with a blue header bar containing various bookmarks like "DEISA", "LRZ", "PRACE", "LDAP", "IGE", "Singapore", "German", "Exist", and "DRG".

The form content is as follows:

Release Management Process Review

*** Required**

Which work packages are you mainly involved in? *

- JRA₁
- JRA₂
- SA₁
- SA₂
- SA₃

Which institution are you from? *

- BADW-LRZ
- CINECA
- CNRS-INSU
- EMSC
- INGV
- KNML-ORFEUS
- LMU
- SCAI
- UEDIN
- ULIV

How frequently do you check the SA₂'s wiki? *

- Daily
- Every few days
- Weekly
- Every few weeks
- Monthly
- Every few months
- Never

Do you subscribe/watch a particular SA₂'s wiki page? *

- Yes
- No
- Did not that there is such an option

Do you know what the release management process is? *

- Yes
- No
- Somewhat

Powered by [Google Docs](#)

Release Management Process Review

https://docs.google.com/spreadsheet/formResponse?formkey=dFUtV2wy

Bookmarks DEISA LRZ PRACE LDAP IGE Singapore German Exist DRG

Release Management Process R...

Release Management Process Review

• Required

Release Management Process Review

Do you know about the release management schedule? *
E.g. where you can find it and how it affects your work.

Yes
 No
 Know about it but do not know where to find it

Do you have sufficient time to perform your work based on the schedule?

1 2 3 4 5

Insufficient More than sufficient

Comments (if any):
E.g. Why do you have insufficient or more than sufficient time? What do you recommend to improve the situation?

Do you find the tasks you need to perform in the release management process sufficiently illustrated in the release management schedule?

Yes
 No
 Somewhat

Do you know about the release request form? *
E.g. where you can find it and how it affects your work?

Yes
 No
 Know about it but don't know where to find it

Do you find the release request form sufficient for its purpose?
E.g. where you can find it and how it affects your work?

1 2 3 4 5

Insufficient More than sufficient

Comments (if any):
E.g. If insufficient, please elaborate and recommend. If form is too complex, please recommend.

Release Management Process Review

https://docs.google.com/spreadsheet/formResponse?form

Bookmarks DEISA LRZ PRACE LDAP IGE Singapore German

Release Management Process R...

Do you know what evaluation tests are being carried out to check for the acceptance of a recommended software or service component? *
These are the tests that are carried out by SA2 during the Check phase of the release management process

Yes
 No

Do you find the tests sufficient to check if a component is suitable to be released as a VERCE software component?
E.g. with a reasonable quality and stability, and is able to perform the required functionality that it is supposed to partake in the VERCE software platform.

1 2 3 4 5

Insufficient More than sufficient

Comments (if any)
E.g. Recommendations on how to improve the tests

Do you have sufficient time to perform your work based on the release management process? *

Yes
 No
 Not applicable

If no, please elaborate
Please indicate which work package you are referring to and if possible the particular activity you refer to.

Do you know which components are recommended and finally selected for this release? *
E.g. in this reporting period?

Yes
 No
 Does not concern me

Do you know what it implies when a component is selected for a release? *

Yes
 No
 Somewhat

Done

Release Management Process Review

https://docs.google.com/spreadsheet/formResponse?form

Bookmarks DEISA LRZ PRACE LDAP IGE Singapore German

Release Management Process R...

Do you think that communication between the JRAs and SA2 is sufficient to communicate the release management process in the last release cycle? *
Applicable to members of JRA1, JRA2 or SA2

Yes
 No
 Not applicable

If no, please elaborate.
Recommendations are very welcome

Do you know where to find the recommendations from SA2 to your component if it is not accepted? *

Yes
 No
 Not applicable

Do you think communication between SA1 and SA2 is sufficient in the last release cycle? *
Applicable to members of SA1 or SA2

Yes
 No
 Not applicable

If no, please elaborate.
Recommendations are very welcome

Do you know where to find the documentation from SA2 with regards to a particular accepted component? *
How to install or deploy the service or tool and recommendations

Yes
 No

Who would you contact if you have a problem? *
If you have a different person in mind that you will contact as per the problem, please kindly elaborate.

Done

Release Management Process Review

https://docs.google.com/spreadsheet/formResponse?form

Bookmarks DEISA LRZ PRACE LDAP IGE Singapore German

Release Management Process R...

Do you know what evaluation tests are being carried out to check for the acceptance of a recommended software or service component? *

These are the tests that are carried out by SA2 during the Check phase of the release management process

Yes

No

Do you find the tests sufficient to check if a component is suitable to be released as a VERCE software component?

E.g. with a reasonable quality and stability, and is able to perform the required functionality that it is supposed to partake in the VERCE software platform.

1 2 3 4 5

Insufficient More than sufficient

Comments (if any)

E.g. Recommendations on how to improve the tests

Do you have sufficient time to perform your work based on the release management process? *

Yes

No

Not applicable

If no, please elaborate

Please indicate which work package you are referring to and if possible the particular activity you refer to.

Do you know which components are recommended and finally selected for this release? *

E.g. in this reporting period?

Yes

No

Does not concern me

Do you know what it implies when a component is selected for a release? *

Yes

No

Somewhat

Done

The image shows a screenshot of a web browser displaying a Google Docs form titled "Release Management Process Review". The browser's address bar shows the URL "https://docs.google.com/spreadsheet/formResponse?foi". The browser's bookmark bar includes folders for "DEISA", "LRZ", "PRACE", "LDAP", "IGE", "Singapore", and "German". The form itself has a light green background and a brown border. At the top, the title "Release Management Process Review" is written in a cursive font. Below the title is a red asterisk and the word "Required". The main heading of the form is "Release Management Process", also in cursive. The first question is "Do you want to know more about the release mangement process?" with three radio button options: "Yes", "No", and "I know enough". Below this is a text input field for "Any additional recommendations". The second question is "Thank you for your time! Please do share who you are so that we can contact you! Thanks!" followed by another text input field. At the bottom of the form are two buttons: "« Back" and "Submit". Below the buttons, it says "Powered by Google Docs" and provides links for "Report Abuse", "Terms of Service", and "Additional Terms". The browser's status bar at the bottom left shows the word "Done".

Release Management Process Review

https://docs.google.com/spreadsheet/formResponse?foi

Bookmarks DEISA LRZ PRACE LDAP IGE Singapore German

Release Management Process R...

Release Management Process Review

* Required

Release Management Process

Do you want to know more about the release mangement process? *

Yes

No

I know enough

Any additional recommendations

Thank you for your time! Please do share who you are so that we can contact you! Thanks! *

« Back Submit

Powered by [Google Docs](#)

[Report Abuse](#) - [Terms of Service](#) - [Additional Terms](#)

Done

References

GT 5.2.2 GridFTP. (n.d.). Retrieved September 20, 2012 from <http://www.globus.org/toolkit/docs/latest-stable/gridftp/>.

Jenkins. (n.d.). Retrieved September 20, 2012 from <http://jenkins-ci.org/>

ObsPy. (n.d.). Retrieved September 20, 2012 from <http://obspy.org/>

Robotframework - A generic test automation framework - Google Project Hosting. (n.d.). Retrieved September 20, 2012 from <http://code.google.com/p/robotframework/>

SeisSol. (n.d.). Retrieved September 2012, 2012 from <http://www.geophysik.uni-muenchen.de/~kaeser/SeisSol/>

DRAFT

Glossary and Links

ATLAS	Automatically Tuned Library Algebra Software
AXISEM	A parallel spectral-element method http://www.seg.ethz.ch/software/axisem
BADW-LRZ	The Bavarian Academy of Sciences and Humanities - Leibniz Supercomputing Centre http://www.lrz.de/english/
BLAS	Basic Linear Algebra Subprograms
Dispel	Data-Intensive Systems Process Engineering Language, a workflow composition language for data-intensive applications
Dispel gateway	The new name of “ADMIRE gateway”
DoW	Description of Work
EGI	European Grid Infrastructure http://www.egi.eu/
ESSL	IBM’s Engineering and Scientific Subroutine Library
Globus	Open source Grid software http://www.globus.org/
Globus Toolkit	Open source software toolkit used for building grids http://www.globus.org/toolkit/
GPGPU	General-purpose computing on graphics processing units
GT	Globus Toolkit
ITU	International Telecommunication Union
JRA1	Equivalent to Work Package 8 (WP8)
JRA2	Equivalent to Work Package 9 (WP9)
LAPACK	Linear Algebra PACKage
LDAP	Lightweight Directory Access Protocol
MKL	Intel Math Kernel Library
MoU	Memorandum of Understanding
NA2	Equivalent to Work Package 2 (WP2)
NCSA	National Center for Supercomputing Applications

	http://www.ncsa.illinois.edu/
NumPy	Numerical Python
ObsPy	A Python Toolbox for seismology/seismological observatories http://obspy.org/
OGSA-DAI	An innovative solution of distributed data access and management http://www.ogsadai.org.uk/
PRACE	Partnership for Advanced Computing in Europe http://www.prace-project.eu/
RegSEM	A Spectral Element Method code to compute seismic wave propagation http://www.ipgp.fr/~paulcup/RegSEM.html
SA1	Equivalent to Work Package 5 (WP5)
SA2	Equivalent to Work Package 6 (WP6)
SA3	Equivalent to Work Package 7 (WP7)
SciPy	Scientific Tools for Python
SEC3D	Programme package for simulation of elastic wave propagation in 3D Cartesian earth models
SeisSol	A simulation software based on the Discontinuous Galerkin Finite Element Method http://www.geophysik.uni-muenchen.de/~kaeser/SeisSol/
SES3D	Programme package for simulation of elastic wave propagation in a spherical section and the computation of Frechet kernels http://www.geophysik.uni-muenchen.de/Members/fichtner/ses3d
SL5	Scientific Linux 5
SL6	Scientific Linux 6
SLES	SUSE Linux Enterprise Server
SPECFEM3D	A simulation software code based on the spectral-element method for 3D seismic wave propagation http://www.seg.ethz.ch/software/specfem3D
The PDCA cycle	The Plan-Do-Check-Act cycle http://labspace.open.ac.uk/mod/resource/view.php?id=346003