Virtual Earthquake and seismology Research Community e-science environment in Europe
Project 283543 – FP7-INFRASTRUCTURES-2011-2 – www.verce.eu – info@verce.eu

Combining HPC with Data-Intensive Research via UNICORE for Seismological Applications

Michele Carpené (CINECA)

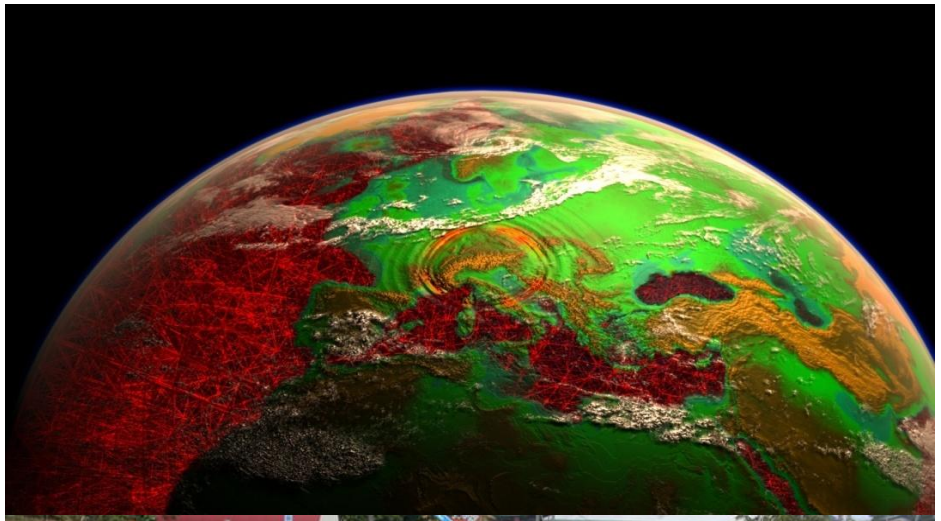
Leipzig, 18 June 2013

Outline

- Introduction – Overview of Verce
- Background
- Forward Modelling and Misfit Calculation
- Technology Description
- UNICORE-based Architecture
- The Processing Elements
- Testing
- Conclusions and future work
- References

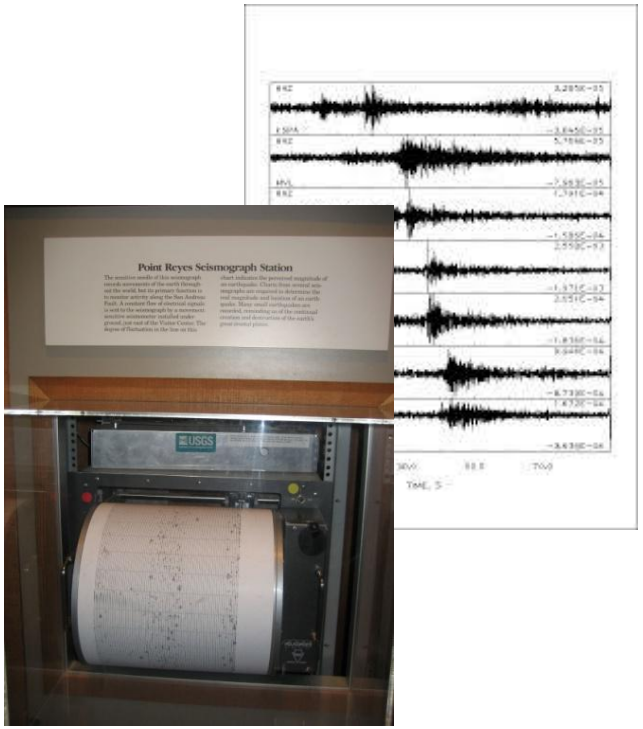
Introduction

- Today's advanced seismology research aims at analysing Earth's structure and composition and predict consequences of seismic phenomena mainly for national security purposes;
- Seismology communities are focused on obtaining information about seismic events from stations and build consistent geophysical wave propagation models from such data



Introduction (2)

- More than 3000 seismic stations sparse around the globe;
- Huge amount of data to be stored, retrieved and processed
- Very complex mathematical Earth's models to estimate effects of future seismic events



Introduction (3)

Two main problems:

- The first problem pertains to storage and data-access issues, EU funded projects like EUDAT(www.eudat.eu) are currently addressing some of these;
- The second problem involves solving complex differential equations over large data sets comparing output of such models with real measurement from stations.

The second one would be intractable without relying on high-performance facilities. VERCE (www.verce.eu) aims to address this last scenario.

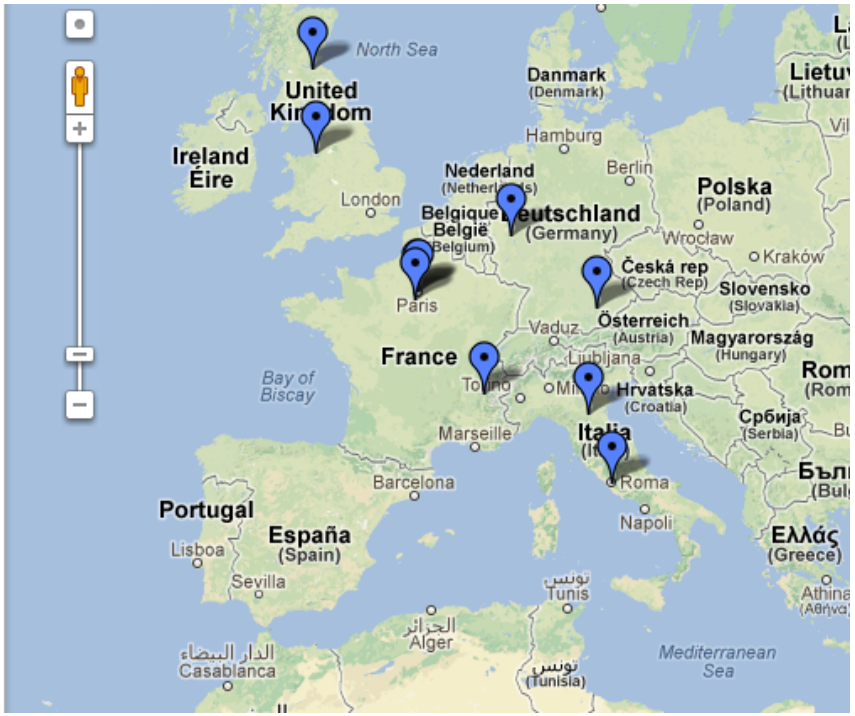
Introduction – Overview of Verce

- VERCE EU-funded project counts 10 partners
- VERCE's strategy is to build up a service oriented architecture+data-intensive platform in order to enable innovative data analysis and data modelling methods

VERCE

Privata · 101 visualizzazioni
Creata il 2 apr · Di · Aggiornata il 2 apr
Valuta la mappa · Scrivi un commento · KML ·

- Institut de Physique du Globe de Paris
- National Center for Scientific Research
- School of Informatics, The University of Edinburgh
- EMSC, France
  France
- INGV, Italy
  00143 Roma, Italy
- University of Liverpool
- LRZ Leibniz-Rechenzentrum der Bayerischen Akadem der Wissenschaften
- Fraunhofer Inst. f. Algorithm. u. Wissenschaftliches Rechnen SCAI
- Cineca, Itaky
- Université Joseph Fourier



Here we present VERCE project, goals and challenges, showing the proposed architecture and focusing on the HPC related scenario.

Background

- VERCE EU-funded project addresses requirements collected from communities integrating HPC, data-intensive and storage resources through a workflow-enabled software platform based on Dispel, a powerful streaming workflow specification language;
- The Dispel language can be used to develop complex workflows based on processing elements, logical units that can be linked together to implement multi-step simulations.

Here we present how a part of the software components in the VERCE architecture can be reused to implement UNICORE GridBeans as Processing Elements.

Forward modelling and misfit calculation

- Generation of synthetics+inversion are compute intensive tasks (heavy MPI communication);
- Comparisons between synthetic data and real data using a selected misfit function;
- The misfit calculation is data-intensive (can be split up in a large number of independent Processes);
- The misfit calculation is completely parallel
  - No MPI communication
  - multiple FFT's required (computationally expensive for each waveform).
  - During computation each data stream can be treated independently.

Technology Description

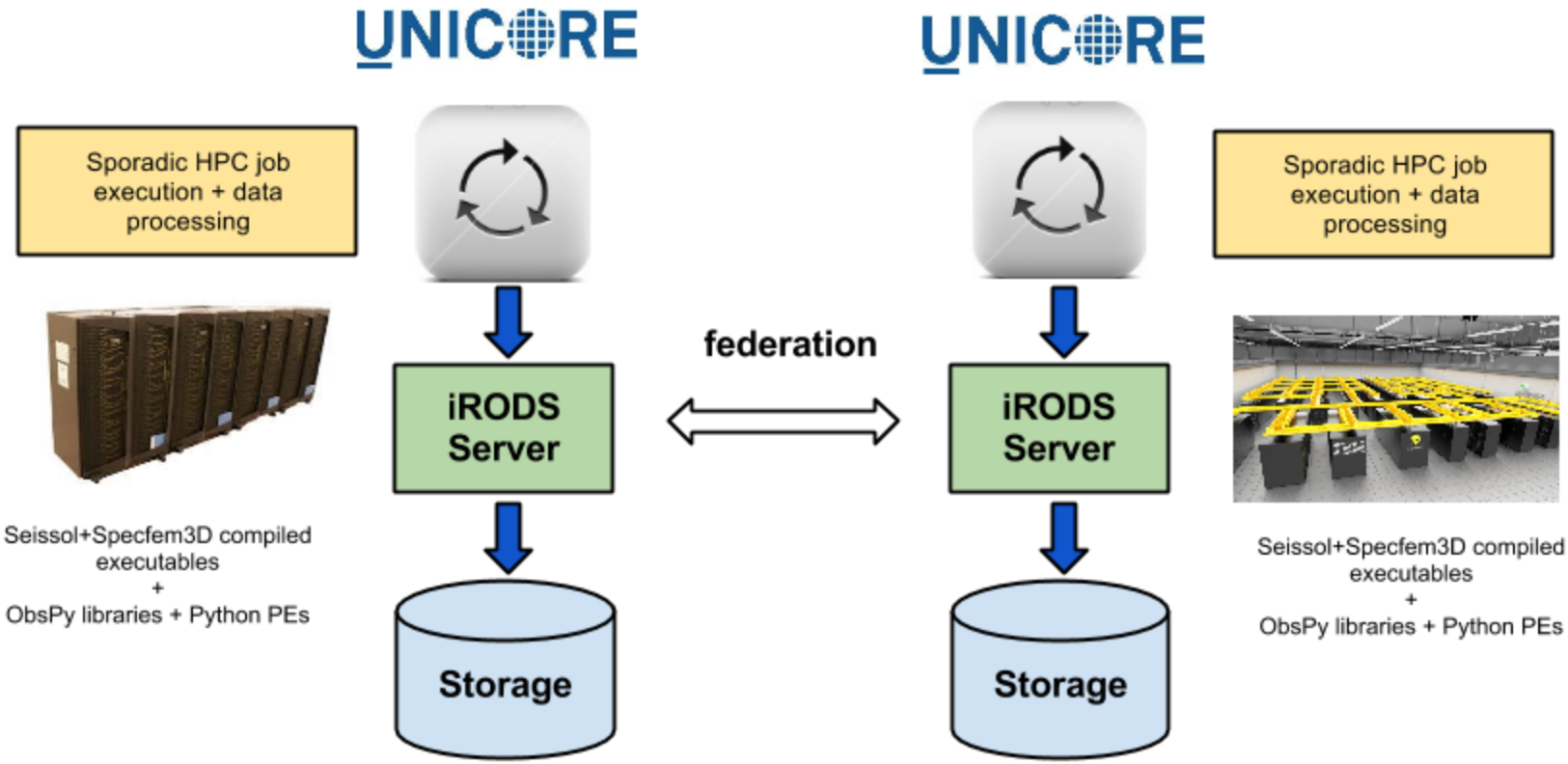
These applications have been installed both on CINECA and LRZ HPC facilities:

- Specfem3D is one of the solvers used in forward and inverse simulations to compute appropriate wavefields in two or three dimensions;
- SeisSol simulation software mimics seismic wave propagation in realistic media with complex geometry;
- ObsPy is an open-source Python framework for processing seismological data.

UNICORE SO Architecture

- The service-oriented architecture we propose is completely based on UNICORE, ObsPy libraries as well as application codes and executables are installed on the HPC machines where the UNICORE Server is deployed.
- Data-intensive operations are executed directly on HPC resources, thus giving two possible advantages:
  - Output files from simulations are already stored very close to the place where the post-processing begins (gridftp transfer is not mandatory for output files)
  - The data-intensive part can take advantage of the HPC resource capabilities (useful when processing million of files simultaneously) and pre/post-processing scripts can be submitted as computational jobs.

UNICORE SO Architecture (2)



The Processing Elements (PEs)

- In VERCE the Processing Element (PE) is intended as a conceptual entity able to perform general tasks on data.
- Behind these elements a script takes one/more files in input and gives back an output result.
- We exploit capabilities of the "data-intensive" PEs, a set of Python scripts able to perform data-intensive tasks on a base64 encoded string.

The Processing Elements (PEs)

We make an assumption: all the Python scripts used for processing data import the same Python module verce.py

```
from admire.verce import *
```

and they receive input and return output in the form of JSON strings

```
cat <inputfile> | python <yourscript> <VERCE JSON> <PARAMETERS JSON (accordingly to PE params)> > <outputfile>
```

For example the pySeisolFileReadTest.in contains the .pkl file path to map synthetic ouput files to seismic stations and the local path to the synthetics output folder.

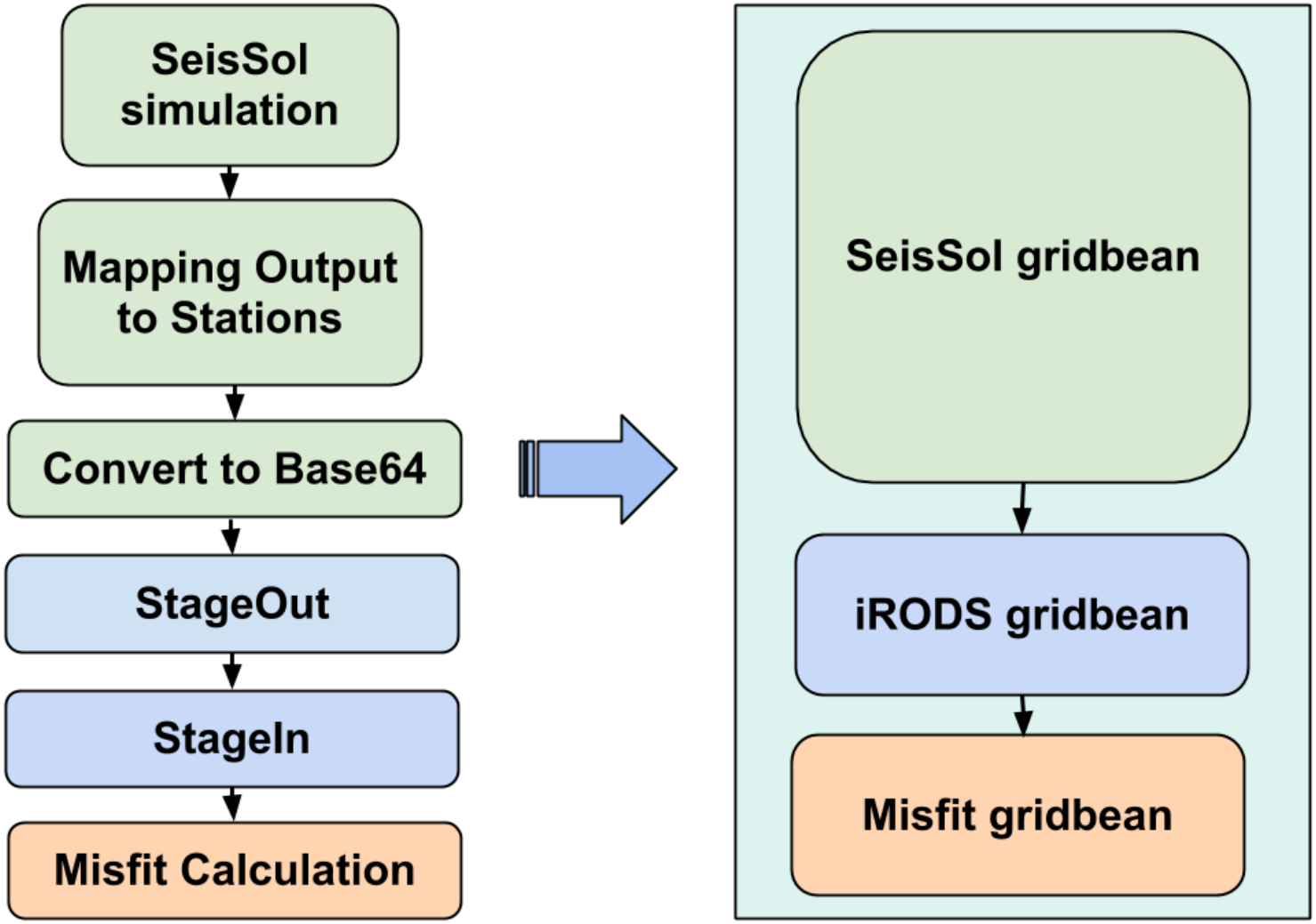
```
{"streams":[{"data": "./used_stations.pkl"},{"data": "../out/"}]}
```

The Processing Elements (PEs)

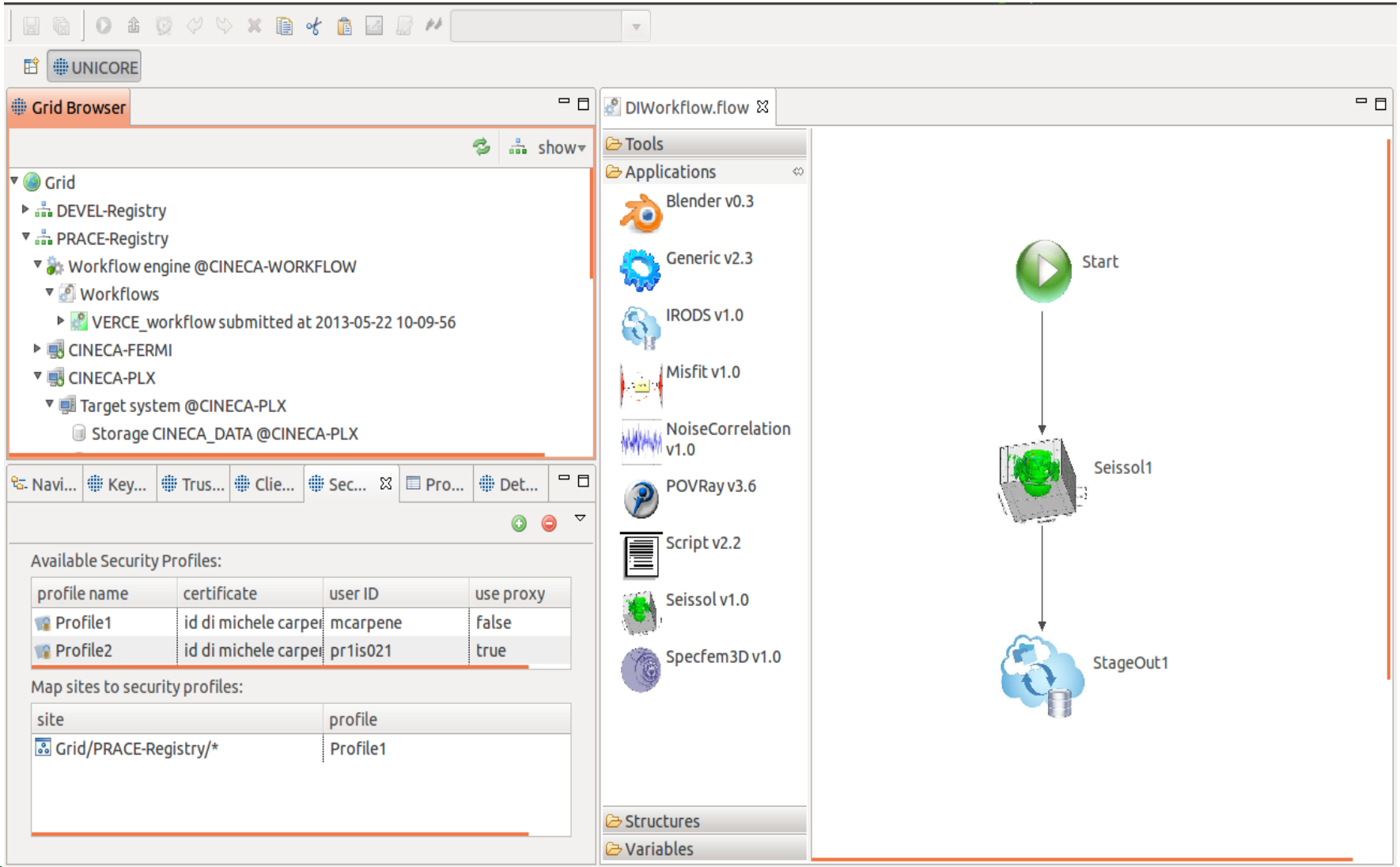
The main Python scripts used in our tests for post-processing tasks from the VERCE repository:

- MapSeissolOutputToStations.py - this Python PE takes as input the path of the SeisSol output directory and the used_stations.pkl configuration file to map the SeiSol output files to seismic stations. A file with correct station code/file name mapping is produced;
- ConvertSeissolOutput.py - this script takes as input a tuple from the output of the above PE and converts a SeisSol output .dat file in a base64 string;
- SingleValuedPhaseMisfit MS.py - performs misfit calculation comparing synthetic output against real data. Takes as input two different base64 strings.

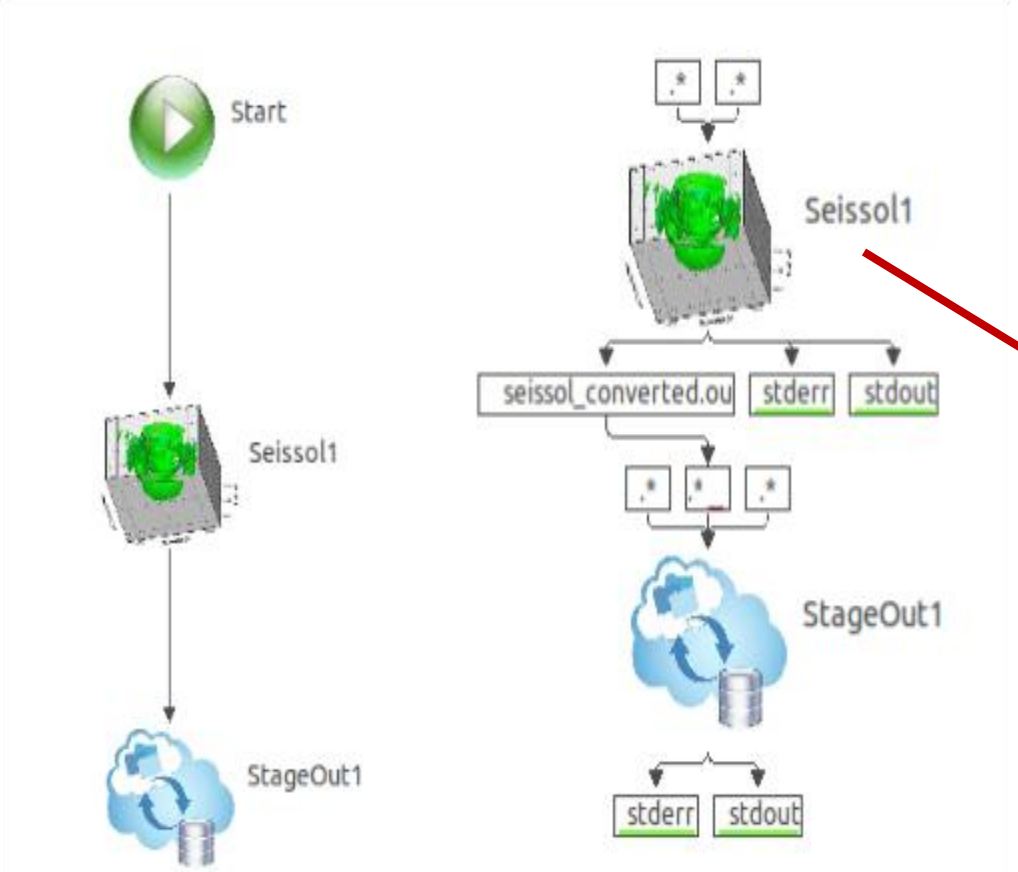
Gridbeans + UNICORE workflow

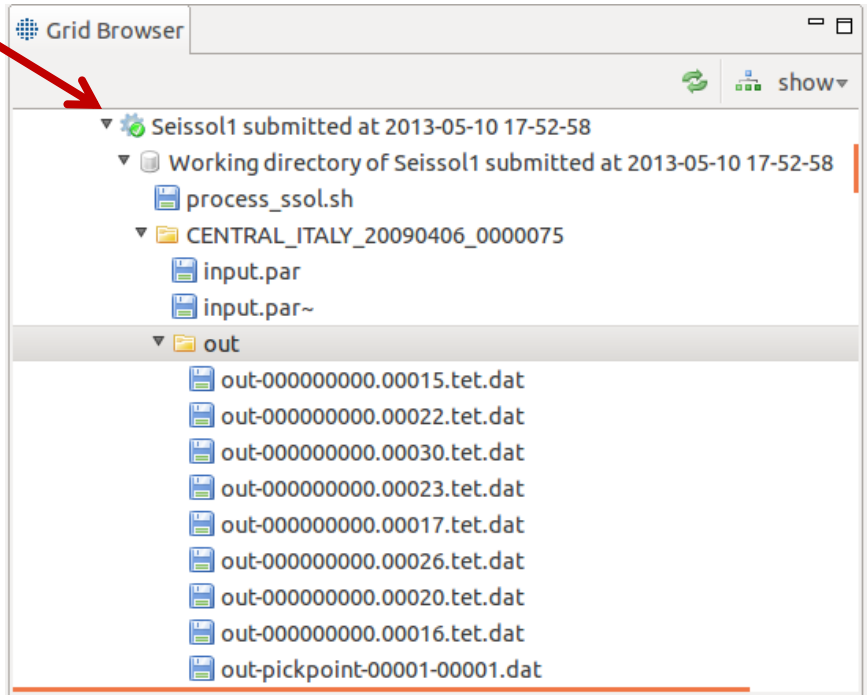


Application Flow example

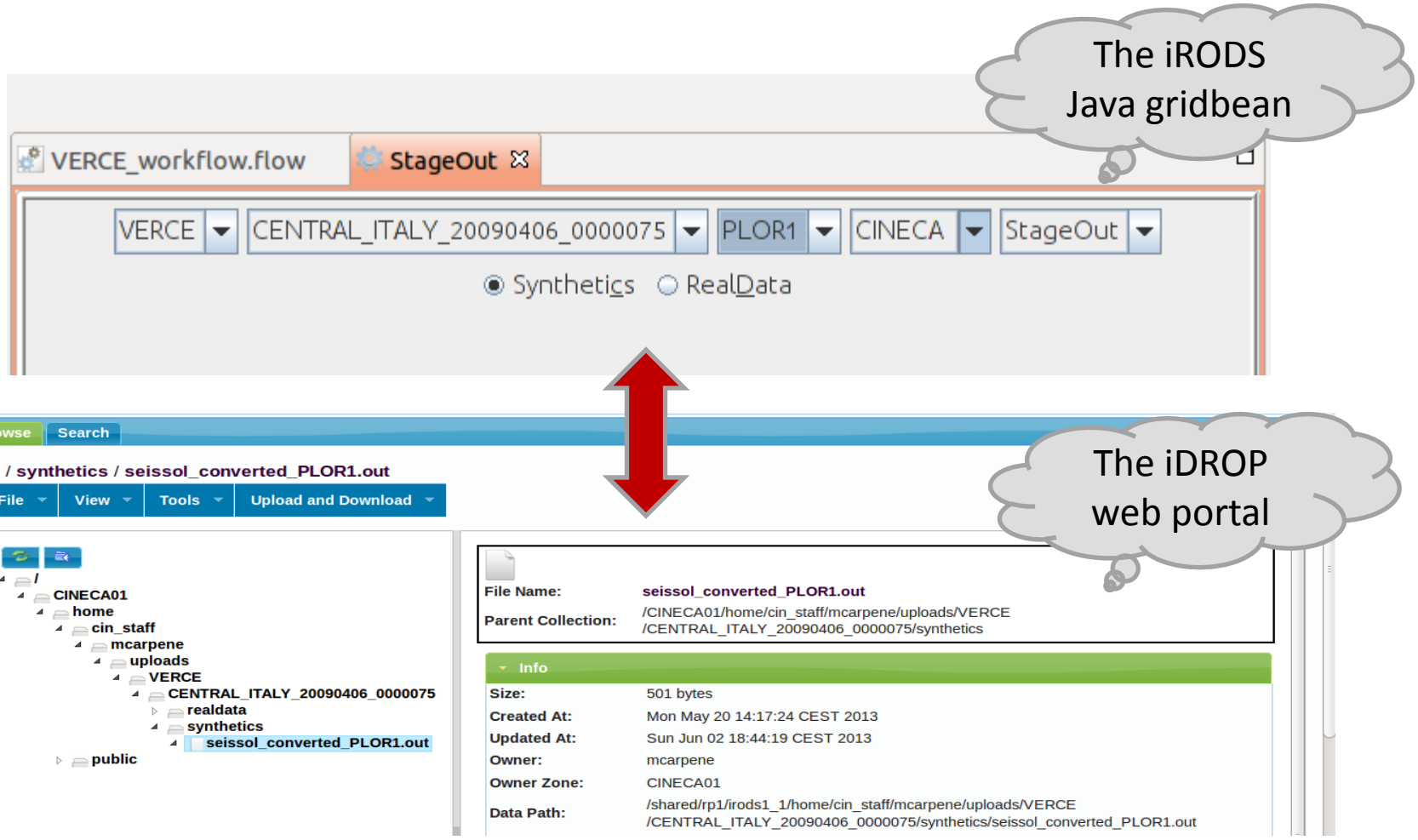


A first simple use-case



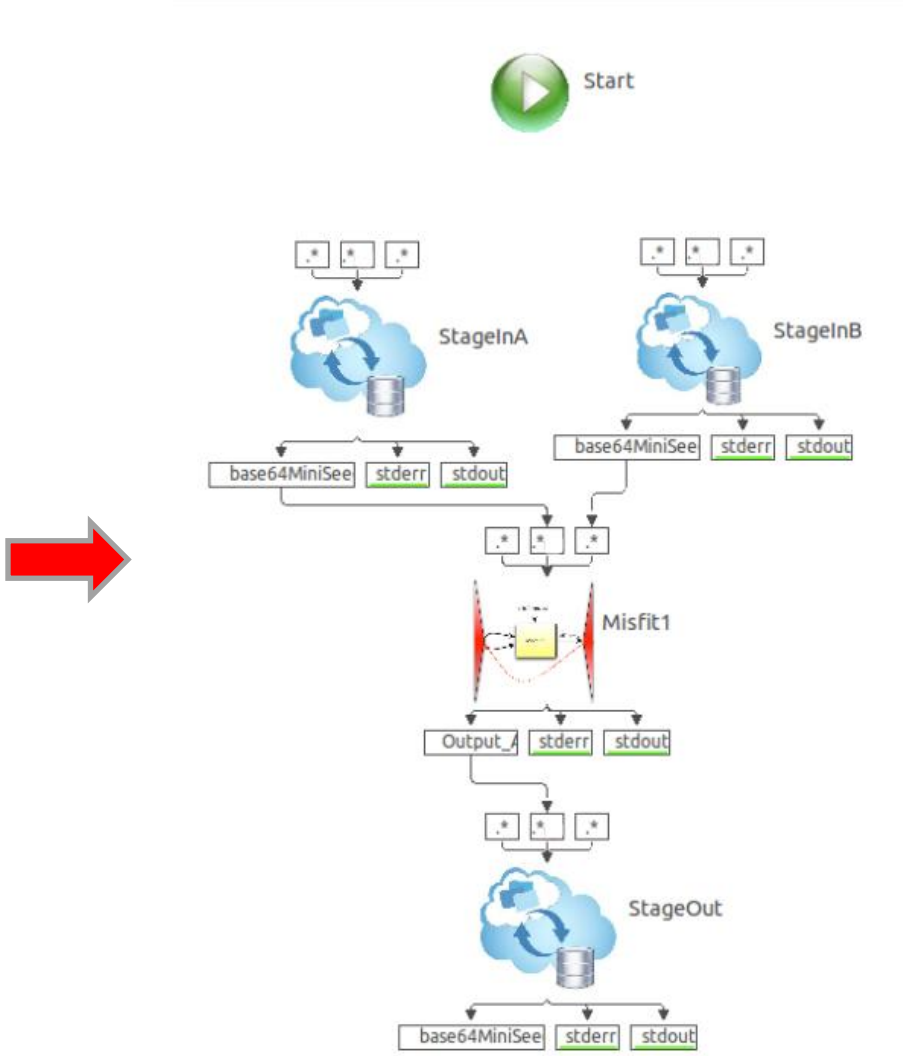


The iDROP Interface



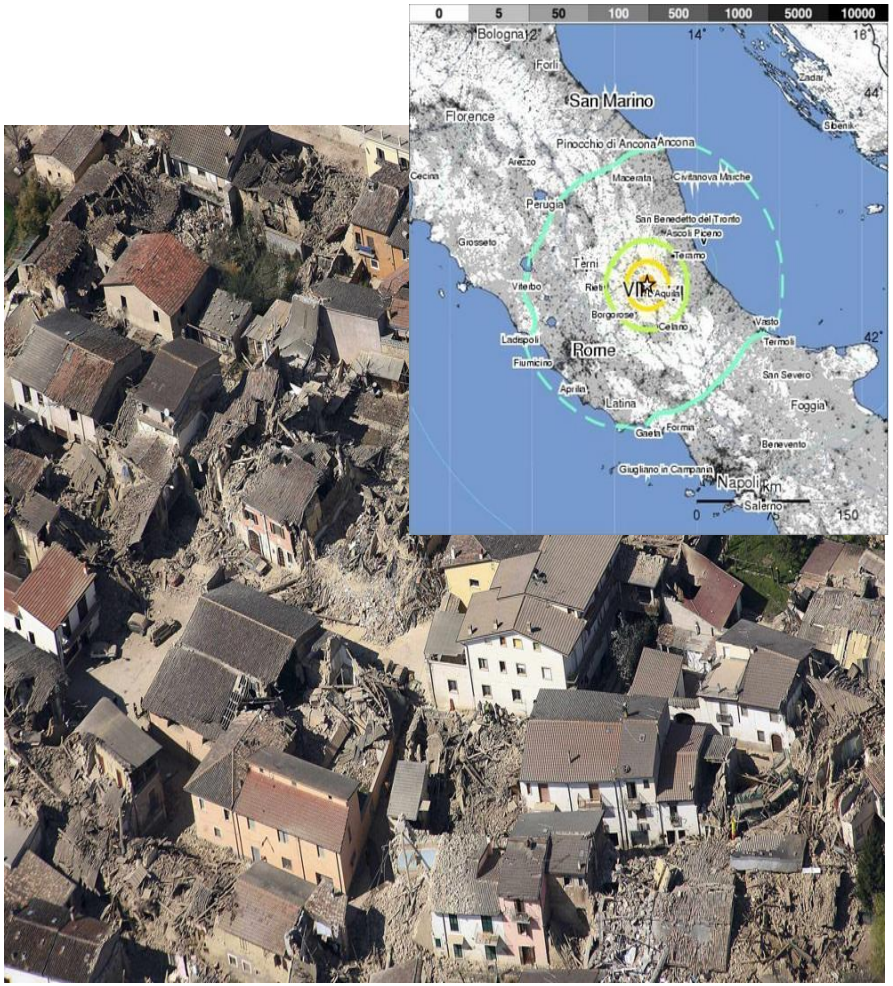
Second use-case: the Misfit

- Two input files are staged in using the iRODS gridbean;
- Input files are passed as UNICORE workflow files to the Misfit gridbean for computation;
- Finally the Misfit result can be stored again on iRODS with an additional StageOut step.



Testing

- The first functionality tests have been performed submitting SeisSol and Specfem simulation jobs to UNICORE on the CINECA IBM PLX;
- The SeisSol simulation has been executed starting from the input data related to the Central Italy 2009 (Aquila) event.



Conclusions and future work

- We have presented how the VERCE Python framework for data-intensive operations can be utilised also outside the Dispel context;
- Python scripts created from the VERCE Data-Intensive Task Force can be re-used to implement UNICORE Java gridbeans to modelize UNICORE workflows;
- Prouved the flexibility and reusability of VERCE's products;
- The UNICORE-based approach could simplify data-transfer between sites and improve the waveform inversion use-case, where million of files must be accessed and analyzed thus overloading the file system;
- In the future more gridbeans can be created to implement different usecases (noise-correlation and pre-processing);
- Finally could be very interesting to define metrics and compare performances obtained submitting same workflows through different service-oriented architectures.

Questions?