Delivering easy-to-use frameworks to empower data-driven research

Dr. Rosa Filgueira,
rosa.filgueira@ed.ac.uk

DARE

CCS-EPCC Workshop Tsukuba 2019

epcc

Background

PhD Computer Science – University Carlos III of Madrid, Spain

- *Dynamic optimization techniques (reducing overhead in IO and communication systems) to enhance performance of MPI-based applications*
- Compute-Intensive Applications

Research Fellow – EPCC, University of Edinburgh, UK

- Scalability and performance of applications executed on HPC and Cloud
- Workflows, data-frameworks, containers and reproducibility tools, etc.
- Data-Intensive Computing Applications

Big Data Science Era, Data Intensive Computing Applications

Scientific fields

Data-driven

Astronomy, Geosciences, Meteorology, Bioinformatics

Common points

Big complex data sets

Need to be analysed

Numerous software tools

Data transformation and visualisation

Big Data Science Era, Data Intensive Computing Applications

Scientific Workflows and Computing environments

Automatically handle large amount of data

Computational steps:

- Scalability
- Flexibility
- Robustly

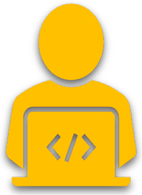
Big Data Science Era, Data Intensive Computing Applications

Seismology

Scientific Workflows:

dispel4py

CWL



Computing environments:

Local

Cloud

HPC

API

DARE

Delivering Agile Research Excellence on European e-Infrastructures

Aim: To empower domain experts to invent and improve their methods and models

How: By providing a new platform and working environment

Outcome:
* Tools/frameworks/APIs for data-driven experiments
* Rapid prototyping
* Run applications at scale on heterogenous systems.

Domains: **Seismology (INGV)** and Climate (CERFACS)

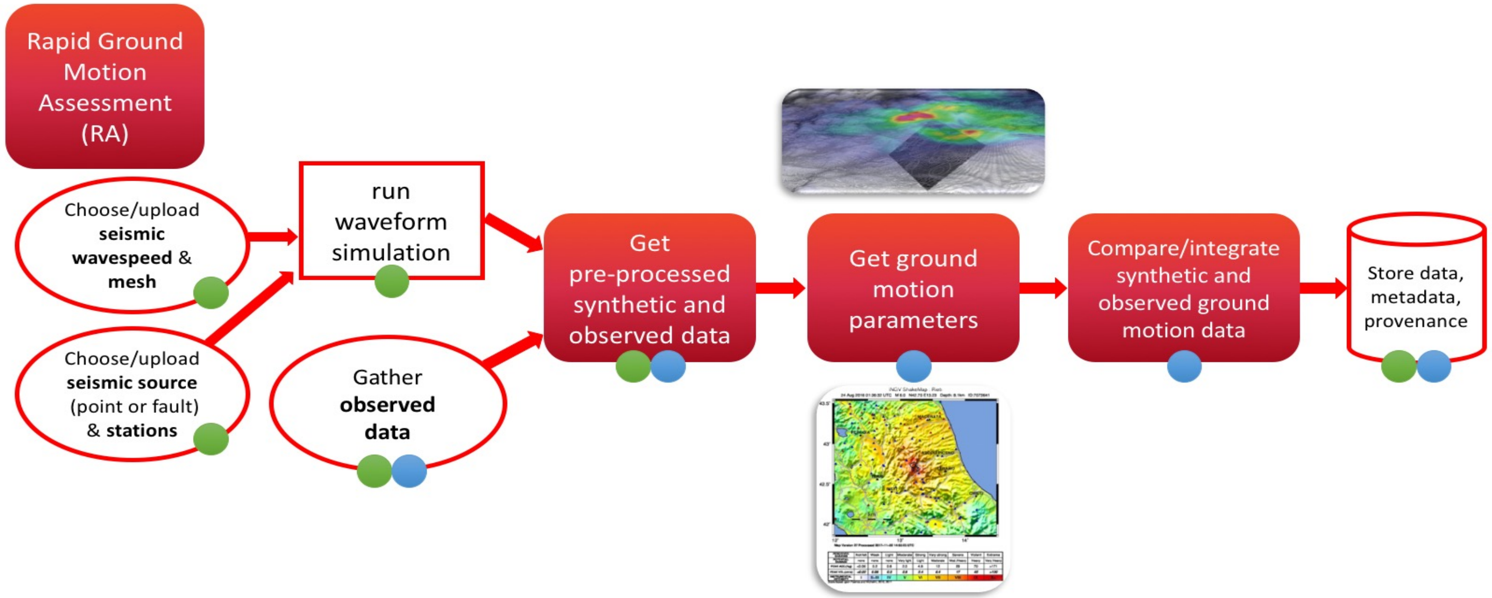
ISTITUTO NAZIONALE
DI GEOFISICA E VULCANOLOGIA

http://project-dare.eu/

Rapid Ground Motion Assessment (RA)

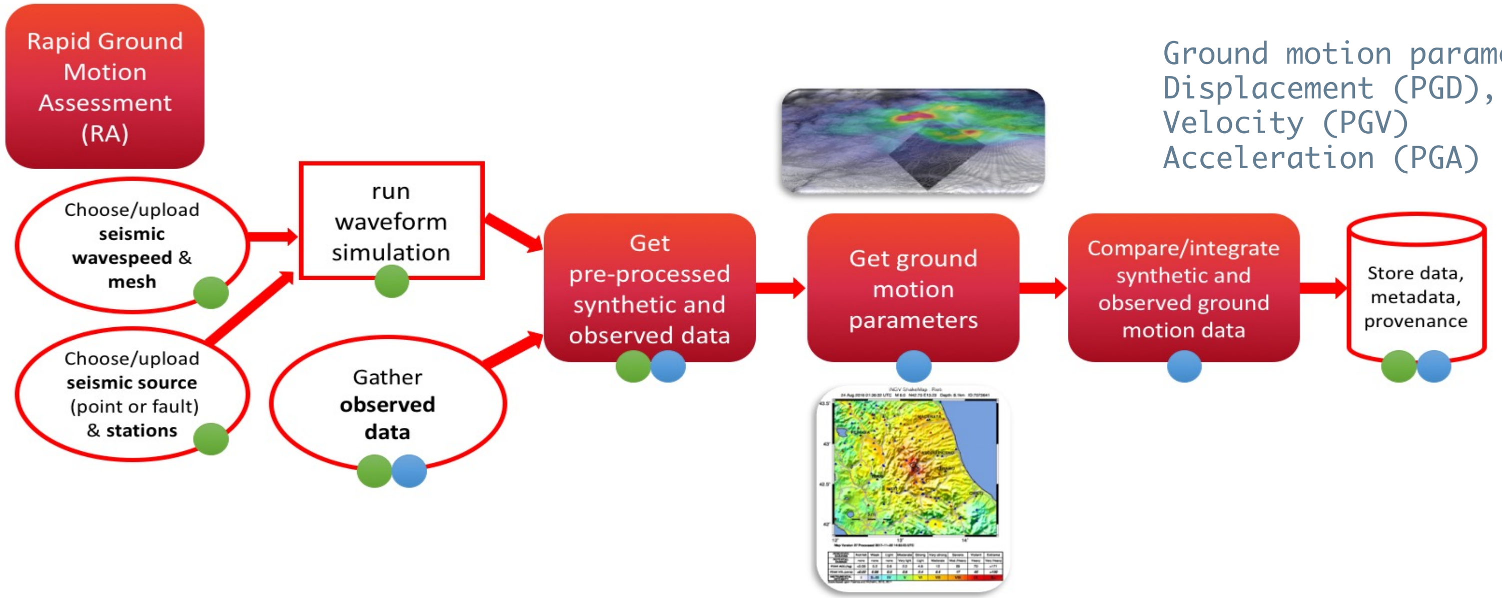
ISTITUTO NAZIONALE DI GEOFISICA E VULCANOLOGIA

* Quickly analyse earthquakes
* Model the ground motion after earthquakes
* Rapid assessment of earthquakes' impact, and emergency response



RA – Five main phases

(1) Select an earthquake gathering the real **observed** seismic **waveforms**
(2) Simulate **synthetic** seismic **waveforms** corresponding to the same earthquake
(3) Pre-process both synthetic and real data;
(4) Calculate the ground motion parameters for synthetic and real data
(5) Compare them with each other (two types of normalization: mean | max)
- shake maps and json files



Ground motion parameters:
Displacement (PGD),
Velocity (PGV)
Acceleration (PGA)

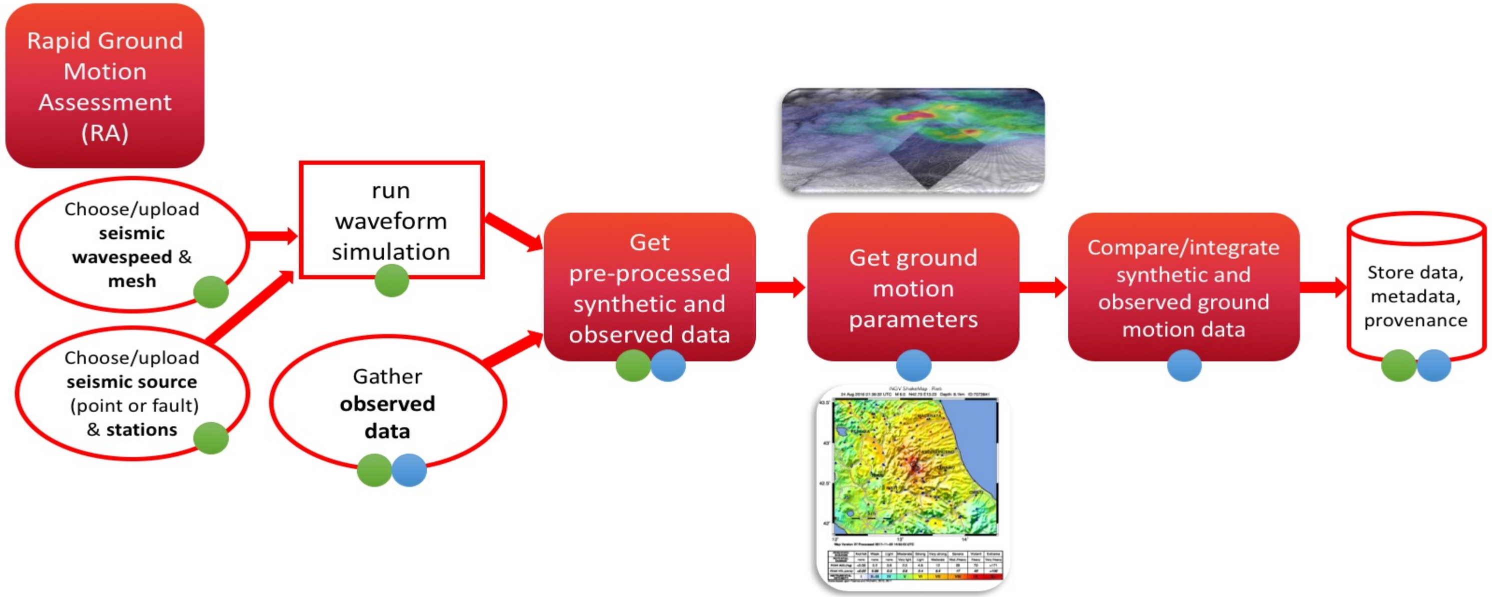
RA – Five main phases

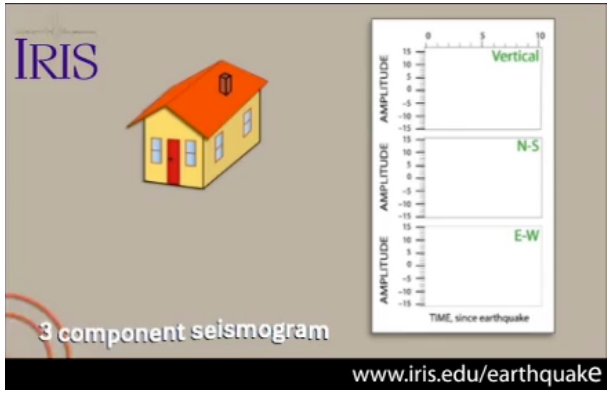
Earthquake Sora: Southern Italy (Lazio - Frosinone) – 16 Feb 2013
Magnitude : 4.9

Synthetics:
43 stations * (3 components) simulated data == 129 synth waveforms

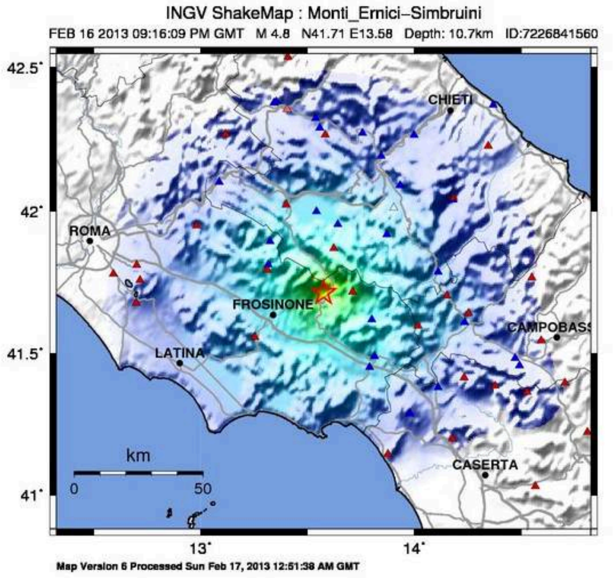
Real – Download from **EIDA archive**
31 stations * (3 components) observed data == 93 real waveforms

Outputs: 2 maps – one per normalization type == 62 files





3-Component Seismogram Records Seismic-wave Motion

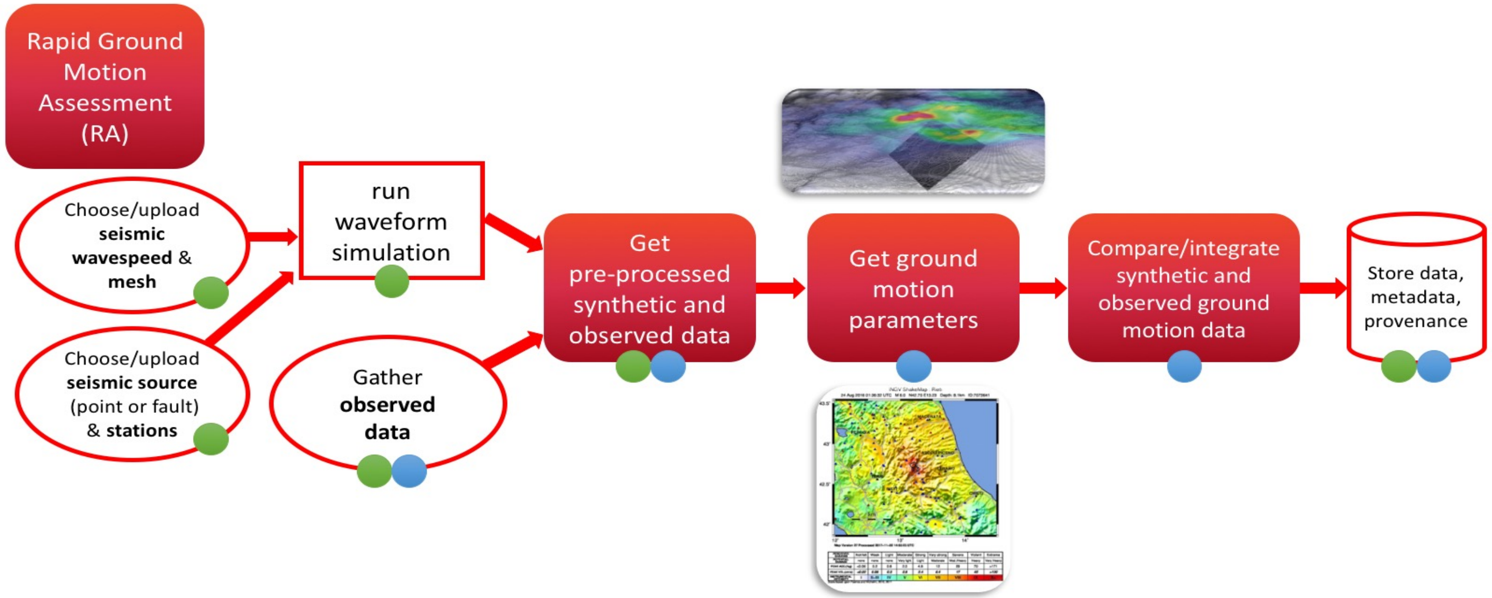


PERCEIVED SHAKING	Not felt	Weak	Light	Moderate	Strong	Very strong	Severe	Violent	Extreme
POTENTIAL DAMAGE	none	none	none	Very light	Light	Moderate	Mod./Heavy	Heavy	Very Heavy
PEAK ACC.(%g)	<0.1	0.5	2.4	6.7	13	24	44	83	>156
PEAK VEL.(cm/s)	<0.07	0.4	1.9	5.8	11	22	43	83	>160
INSTRUMENTAL INTENSITY	I	II–III	IV	V	VI	VII	VIII	IX	X+

Scale based upon Wald, et al.; 1999

RA – Five main phases

* Rapid data analyses and transfer between co-working environments
* Multiple data formats (ascii, xml, obspy, jpg, binary, geojson)
* Multiple data sources (public databases)
* Combination of numerous outputs from multiple workflows/software → Provenance
* Computing and storage resources on demand
* Flexibility and abstraction of workflow pipelines



RA – User Stories – Agile Methodology

ISTITUTO NAZIONALE DI GEOFISICA E VULCANOLOGIA

Closed 20 0

As a seismologist I want to upload to an archive simulation configuration parameters (waveform model, mesh and seismic source)
C - API | C - DataStore | T - Rapid Assessment | T - Seismic Source Characterisation
#29

As a seismologist I want to search and select from an archive of configuration parameters (waveform model, mesh and seismic source) to setup my simulation
C - API | C - DataStore | T - Rapid Assessment | T - Seismic Source Characterisation
#28

As seismologist I want to run a waveform simulator with my selected configuration parameters
C - API | C - Provenance | C - Workflow Language | C- Processing | T - Rapid Assessment | T - Seismic Source Characterisation
#31

Closed 20 0

As a seismologist I want to access observed data from external archives for further processing
C - API | C - DataStore | C - Provenance | C - Workflow Language | C- Processing | T - Ensemble Simulations | T - Rapid Assessment | T - Seismic Source Characterisation
#32

As seismologist I want to preprocess waveforms for further analyses
C - API | C - PE Catalogue | C - Provenance | C - Workflow Language | C- Processing | T - Ensemble Simulations | T - Rapid Assessment | T - Seismic Source Characterisation
#41

As a seismologist I want extract selected ground motion parameters from waveforms
C - API | C - PE Catalogue | C - Provenance | C - Workflow Language | C- Processing | T - Ensemble Simulations | T - Rapid Assessment
#12

Closed 20 0

As a seismologist I want to compare simulation outputs (e.g. ground motion parameters) with observed data
C - API | C - PE Catalogue | C - Provenance | C - Workflow Language | C- Processing | T - Ensemble Simulations | T - Rapid Assessment
#5

As a seismology I want to store output data and visual products produced by any workflow with provenance and metadata
C - DataStore | C - Provenance | C - Workflow Language | T - Ensemble Simulations | T - Rapid Assessment | T - Seismic Source Characterisation
#19

DARE

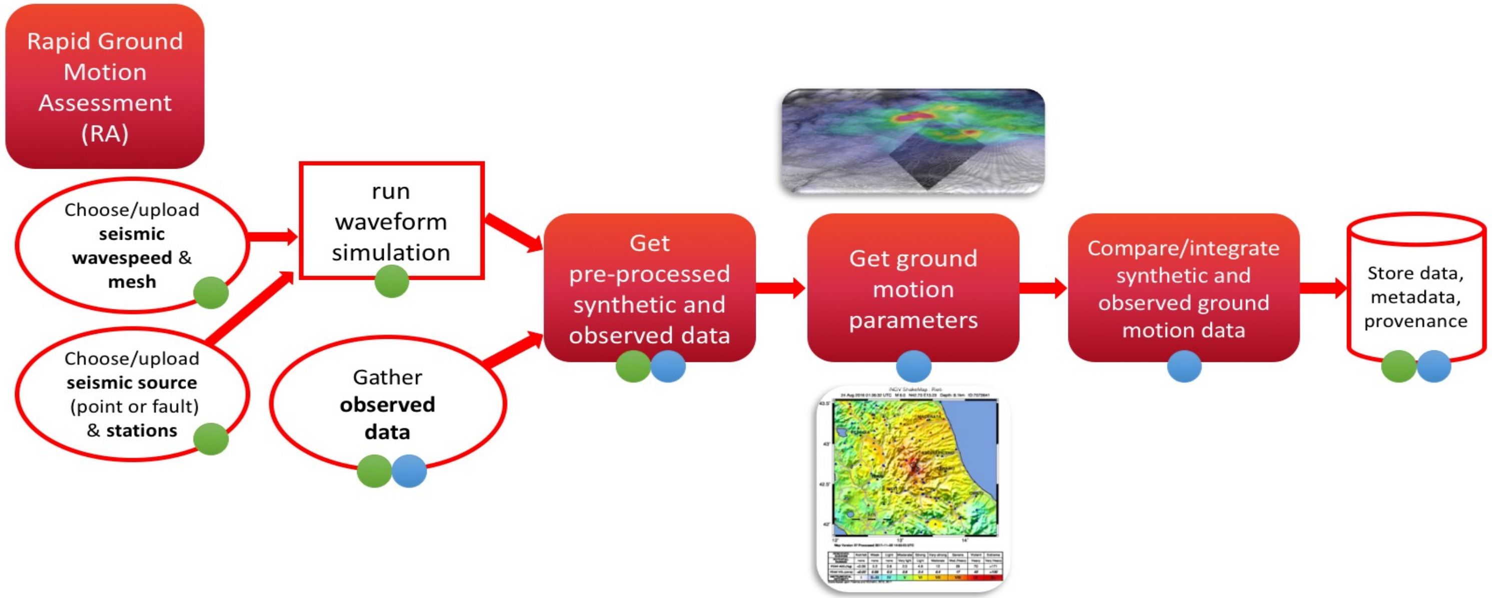
Current status (at M18 of the project) of the stories (all closed) for the RA use case

epcc

RA – Portable and Reproducible

To run it on different computing environments
- without making any (or little) changes

Technologies:
* Scientific workflows (CWL), stream-based data-flow systems (dispel4py),
* Containers (Docker), Infrastructure orchestrations (Kubernetes),
* Notebooks (Jupyter), and Cloud platforms.



RA – Summary Steps (I)

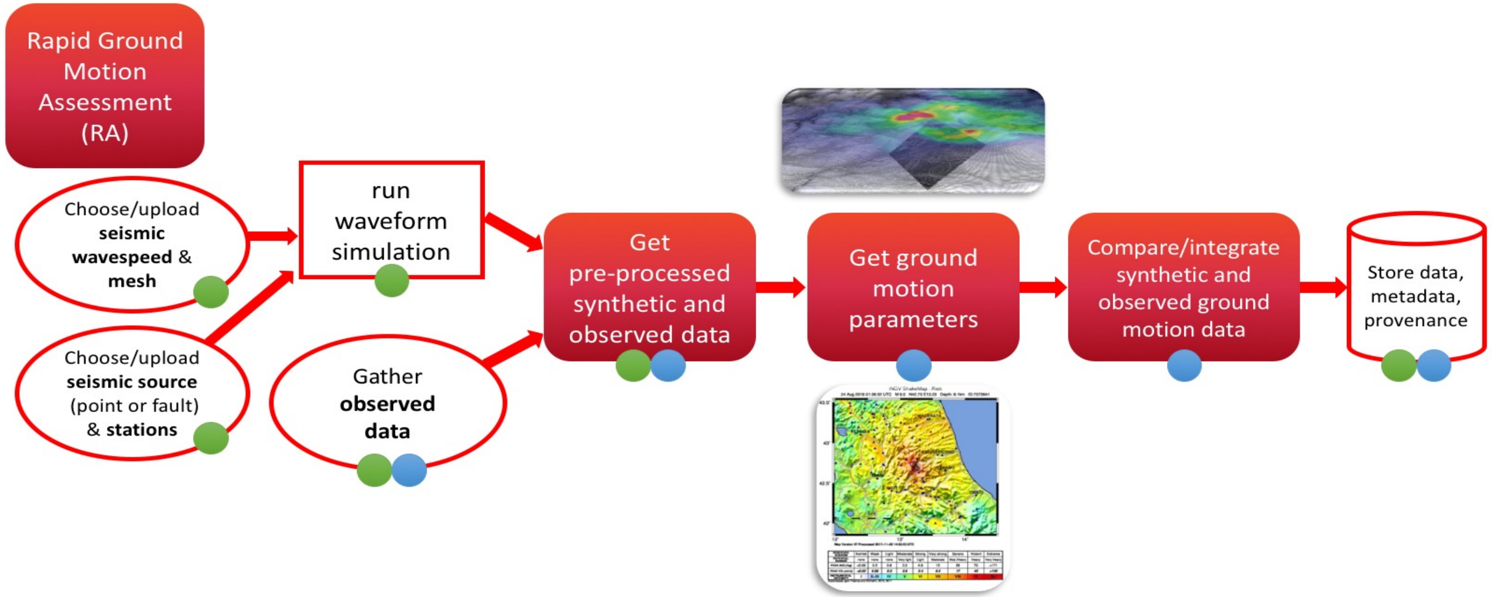
1. **Dockerize Specfem3D**

   Build a **CWL** workflow for generating synthetic data

2. Build **dispel4py workflows** to represent each part of the RA (**)

   (**) Except for the generation of the synthetic data

3. Use **CWL** to **connect RA** dispel4py workflows



RA – Summary Steps (I)

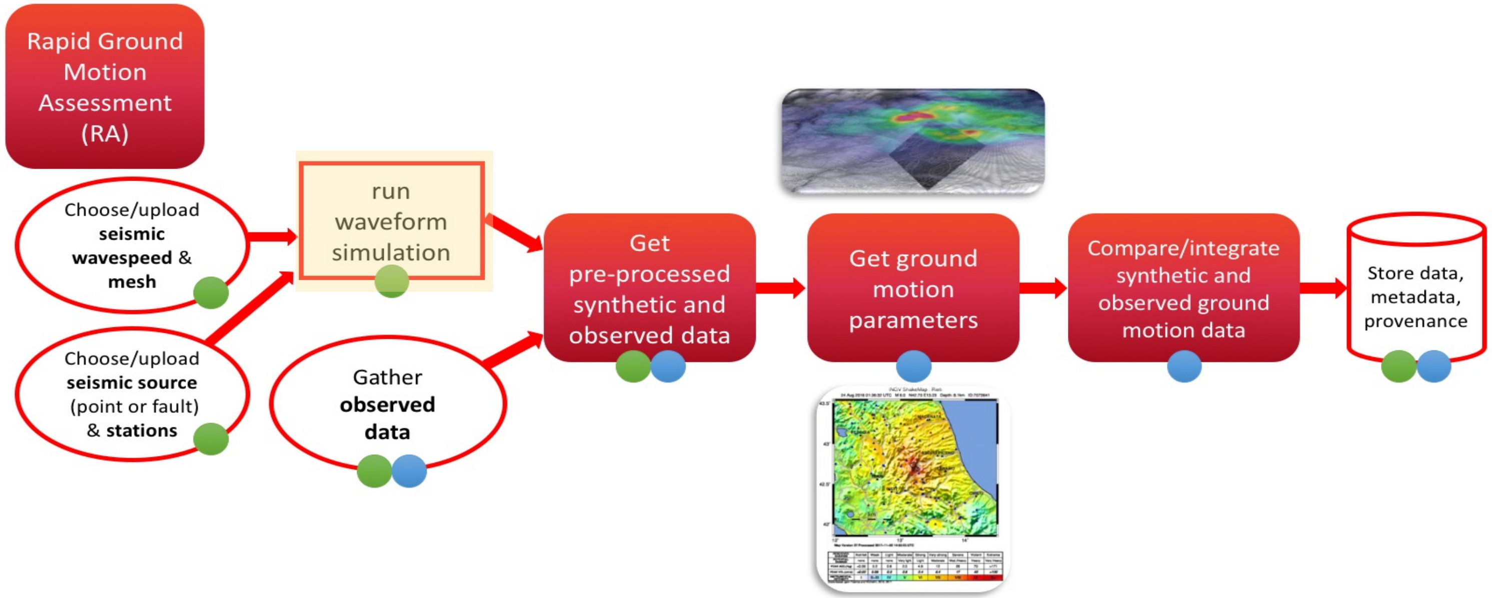
1. Dockerize Specfem3D

   Build a CWL workflow for generating synthetic data

2. Build **dispel4py** workflows to represent each part of the RA (**)

   (**) Except for the generation of the synthetic data

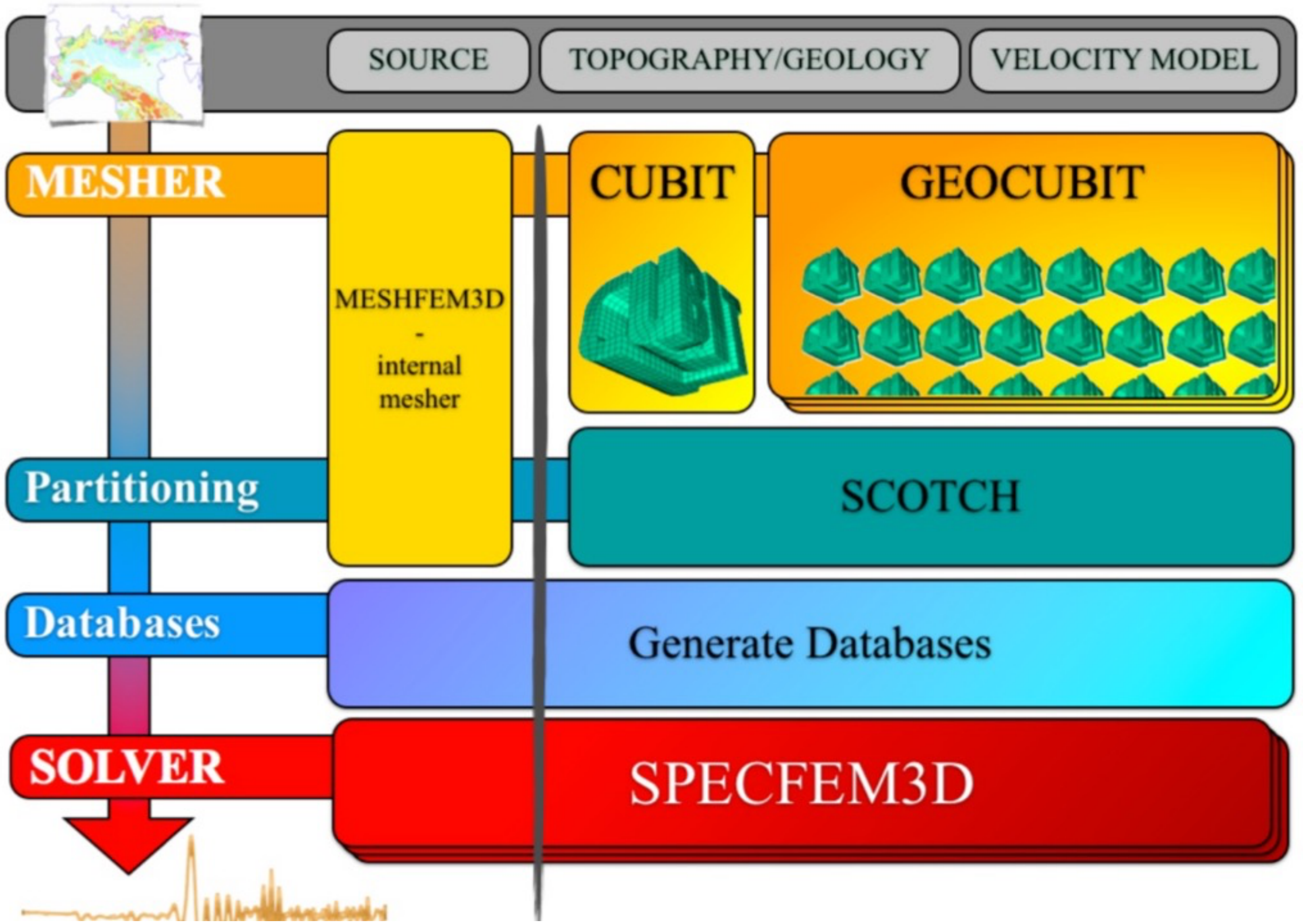
3. Use **CWL** to **connect RA** dispel4py workflows



Seismic Waveform Simulation: SPECFEM3D

Rapid Ground Motion Assessment (RA)

run waveform simulation

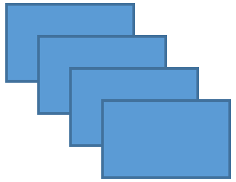


Specfem3d– MPI application that creates the synthetic waveforms

Seismic Waveform Simulation: SPECFEM3D

ISTITUTO NAZIONALE DI GEOFISICA E VULCANOLOGIA

We run it before
In SuperMUC -
HPC–cluster -
256 cores



MPI Cluster

>> cd specfem3D_input_test
>> ./run_specfem3d.sh

specfem3D

specfem3D_input_test

DATA

- CMTSOLUTION : earthquake source parameter file
- mesh_homogeneous : high-quality mesh for the region
- Par_file : the main parameter file
- STATIONS : list of stations

```
#!/bin/bash

echo "running example: `date`"
currentdir=`pwd`

# sets up directory structure in current example directory
echo
echo "   setting up example..."
echo

# cleans output files
mkdir -p OUTPUT_FILES
rm -rf OUTPUT_FILES/*

# links executables
mkdir -p bin
cd bin/
rm -f *
ln -s ../../specfem3d/bin/xdecompose_mesh
ln -s ../../specfem3d/bin/xgenerate_databases
ln -s ../../specfem3d/bin/xspecfem3D
cd ../

# stores setup
cp DATA/Par_file OUTPUT_FILES/
cp DATA/CMTSOLUTION OUTPUT_FILES/
cp DATA/STATIONS OUTPUT_FILES/

# get the number of processors, ignoring comments in the Par_fi
NPROC=`grep ^NPROC DATA/Par_file | grep -v -E '^[[:space:]]*#'

BASEMPIDIR=`grep ^LOCAL_PATH DATA/Par_file | cut -d = -f 2 `
echo $BASEMPIDIR
mkdir -p $BASEMPIDIR

# decomposes mesh using the pre-saved mesh files in MESH-defaul
echo
echo "  decomposing mesh..."
echo
./bin/xdecompose_mesh $NPROC ./DATA/mesh_homogeneous $BASEMPIDI
# checks exit code
if [[ $? -ne 0 ]]; then exit 1; fi

# runs database generation
if [ "$NPROC" -eq 1 ]; then
  # This is a serial simulation
  echo
  echo "  running database generation..."
  echo
  ./bin/xgenerate_databases
else
  # This is a MPI simulation
  echo
  echo "  running database generation on $NPROC processors..."
  echo
  mpirun --allow-run-as-root -np $NPROC ./bin/xgenerate_databases
fi
# checks exit code
if [[ $? -ne 0 ]]; then exit 1; fi

# runs simulation
"run_test.sh" 84L, 1838C
```

```
AQT1 IV 42.77383 13.29350 770.0 0.0
AQU IV 42.35400 13.40500 710.0 0.0
ARRO IV 42.57917 12.76567 253.0 0.0
CAFR IV 42.22730 14.34700 250.0 0.0
CAMP IV 42.53578 13.40900 1283.0 0.0
CERA IV 41.59780 14.01830 800.0 0.0
CERT IV 41.94903 12.98176 773.0 0.0
CESX IV 42.60849 12.58676 380.0 0.0
FAGN IV 42.26573 13.58379 761.0 0.0
FIAM IV 42.26802 13.11718 1070.0 0.0
GIUL IV 41.55827 13.25458 566.0 0.0
GUAR IV 41.79450 13.31229 741.0 0.0
INTR IV 42.01154 13.90460 924.0 0.0
LATB IV 41.49390 12.96239 0.0 0.0
LAV9 IV 41.67778 12.69888 300.0 0.0
LNSS IV 42.60286 13.04032 1155.0 0.0
LPEL IV 42.04680 14.18320 760.0 0.0
MA9 IV 41.76976 12.65907 340.0 0.0
MIDA IV 41.64188 14.25402 950.0 0.0
MNS IV 42.38546 12.68106 706.0 0.0
MODR IV 41.14590 13.87790 345.0 0.0
MOMA IV 42.80387 12.57007 1040.0 0.0
MTCE IV 42.02280 12.74222 388.0 0.0
NRCA IV 42.83355 13.11427 927.0 0.0
PIGN IV 41.20000 14.17989 398.0 0.0
POFI IV 41.71743 13.71202 878.0 0.0
PTQR IV 42.02193 13.40057 957.0 0.0
RDP IV 41.76039 12.71030 760.0 0.0
RM32 IV 42.57024 13.29322 1362.0 0.0
RM33 IV 42.50898 13.21452 1097.0 0.0
RMP IV 41.81112 12.70222 380.0 0.0
RNI2 IV 41.70328 14.15240 950.0 0.0
ROM9 IV 41.82842 12.51553 110.0 0.0
SAMA IV 41.78050 12.59230 119.0 0.0
SGG IV 41.38667 14.37917 880.0 0.0
SMA1 IV 42.63050 13.33530 1150.0 0.0
SRES IV 42.23696 12.50993 410.0 0.0
T0104 IV 42.35990 13.33820 754.0 0.0
```

Prelir

Harvard CMT solution

PD
eve
tim
hal
lat
lon
dep
Mrr
Mtt
Mpp
Mrt
Mrp
Mtp

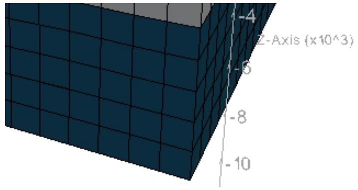
body-wave magnitude
surface-wave magnitude

longitude	depth	mb	Ms	PDE event name
-118.3792	6.4	4.2	4.2	HOLLYWOOD

$$\begin{bmatrix} M_{rr} & M_{r\theta} & M_{r\phi} \\ M_{r\theta} & M_{\theta\theta} & M_{\theta\phi} \\ M_{r\phi} & M_{\theta\phi} & M_{\phi\phi} \end{bmatrix}$$

$$\frac{1}{\sqrt{2}}(\mathbf{M}:\mathbf{M})^{1/2} \approx 2.18 \times 10^{22} \text{ dyne cm}$$

$$\frac{2}{3}(\log_{10} M_0 - 16.1) \approx 4.19$$



NGNOD
The nu
elemen

exahedra. We use either 8-node mesh
rnal mesher, the only option is 8-node

Test Case: RA

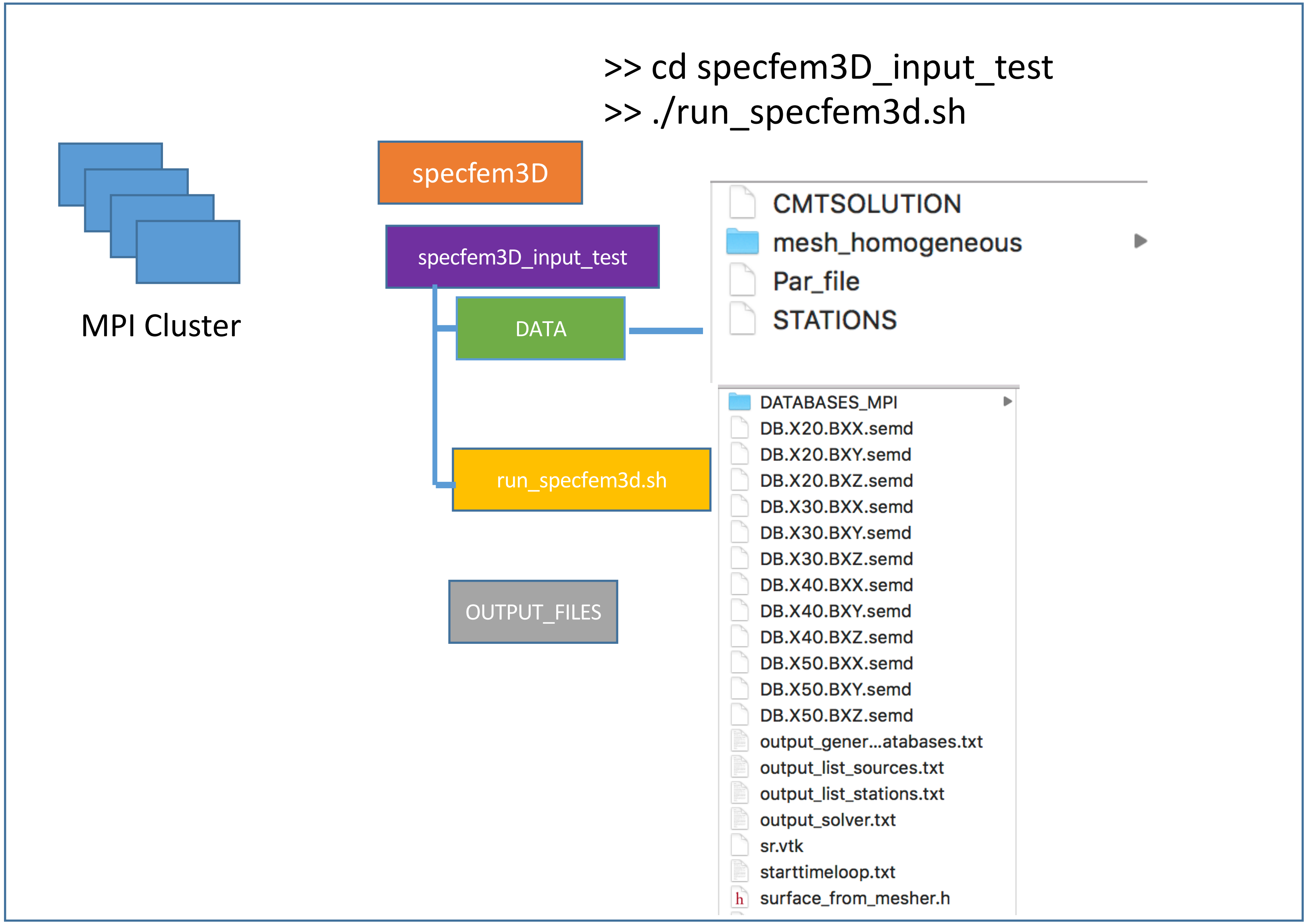
run waveform simulation

DARE

Seismic Waveform Simulation: SPECFEM3D

ISTITUTO NAZIONALE
DI GEOFISICA E VULCANOLOGIA

We run it before
In SuperMUC -
HPC–cluster -
256 cores



Test Case: RA

run waveform simulation

DARE

epcc

Common Workflow Language - CWL

COMMON WORKFLOW LANGUAGE

Open standard for describing

- workflows and tools
- platform-independent

Designed to meet the needs of data-intensive science

1st-tool.cwl

```
#!/usr/bin/env cwl-runner

cwlVersion: v1.0
class: CommandLineTool
baseCommand: echo
inputs:
  message:
    type: string
    inputBinding:
      position: 1
outputs: []
```

```
$ cwl-runner 1st-tool.cwl echo-job.yml
[job 1st-tool.cwl] /tmp/tmpmM5S_1$ echo \
    'Hello world!'
Hello world!
[job 1st-tool.cwl] completed success
{}
Final process status is success
```

Next, create a file called `echo-job.yml`, containing the following boxed text, which will describe the input of a run:

echo-job.yml

```
message: Hello world!
```

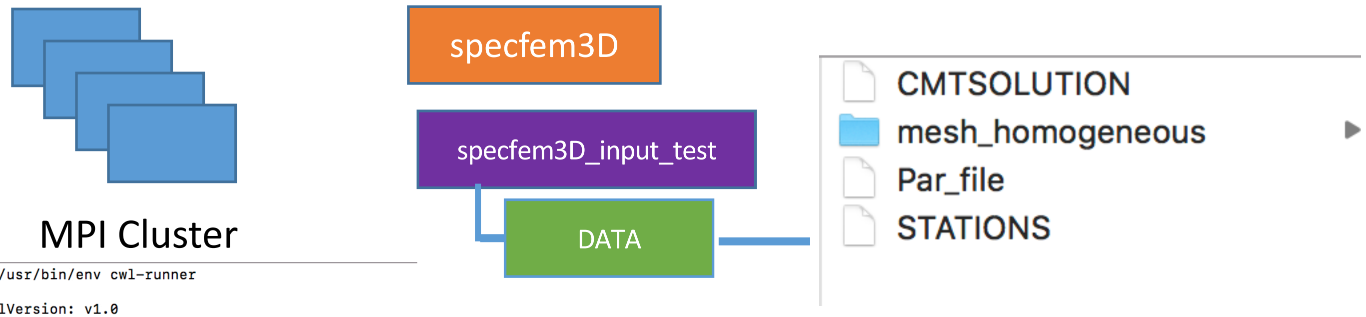
Rules to describe each command line tool and its parameters

https://www.commonwl.org/user_guide/02-1st-example/index.html

epcc

Seismic Waveform Simulation: SPECFEM3D

ISTITUTO NAZIONALE
DI GEOFISICA E VULCANOLOGIA




```
#!/usr/bin/env cwl-runner

cwlVersion: v1.0
class: Workflow

inputs:
   script_environment: File
   script_decompose: File
   script_database: File
   script_simulation: File
outputs:
    results:
        type: Directory
        outputSource: simulation/output
steps:
  create_environment:
    run: env_preparation.cwl
    in:
      script: script_environment
    out: [output]
  decompose_mesh:
    run: decompose_mesh.cwl
    in:
      script: script_decompose
      results: create_environment/output
    out: [output]

  database_generation:
    run: database_generation.cwl
    in:
      script: script_database
      results: decompose_mesh/output
    out: [output]

  simulation:
    run: simulation.cwl
    in:
      script: script_simulation
      results: database_generation/output
    out: [output]

~
```

run_specfem3d.cwl

```
>> cwltool run_specfem3d.cwl run_specf3m3d.yml
```

```
script_environment:
  class: File
  path: env_preparation.sh
  secondaryFiles:
    - class: Directory
      location: specfem3d_test_input

script_decompose:
  class: File
  path: decompose_mesh.sh

script_database:
  class: File
  path: database_generation.sh

script_simulation:
  class: File
  path: simulation.sh
~
~
```

run_specfem3d.yml

Test Case: RA

run waveform simulation

DARE

epcc

RA – Summary Steps (I)

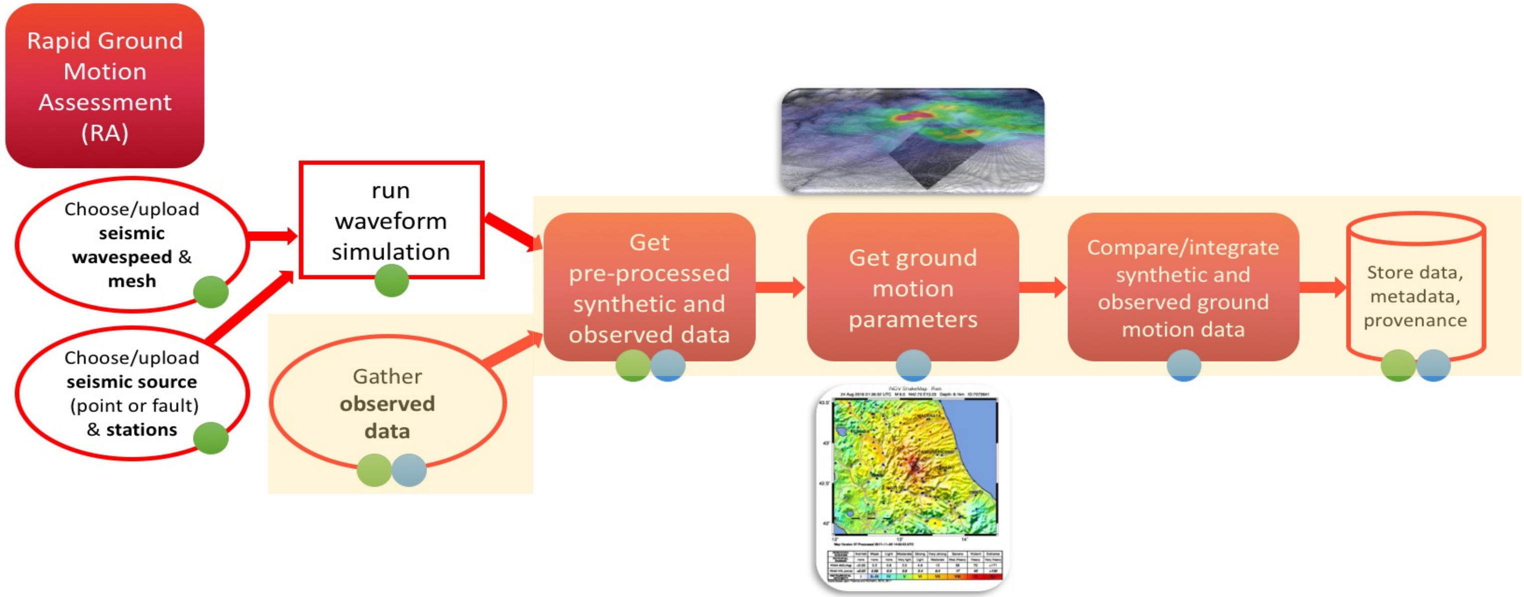
1. Dockerize Specfem3D

   Build a CWL workflow for generating synthetic data

2. Build **dispel4py** workflows to represent each part of the RA (**)

   (**) Except for the generation of the synthetic data

3. Use **CWL** to **connect RA** dispel4py workflows



dispel4py parallel stream-based dataflow system

Automation

Automates pipeline executions
Concurrent, distributed computations
Stream-based mode
Provenance

Workflow Composition

Python Library
Groupings
Jupyter Notebooks

Rosa Filgueira
Amrey Krause
Malcolm Atkinson

DAG execution engine

High-level stream-based data pipeline
~ Apache Beam

Mapping

Sequential
Multiprocessing
MPI
Apache Storm
ZeroMQ (prototype)

Optimisation

Multiple streams
Avoids I/O operations

Key-features: Automatic mappings to different engines, concurrent & stream-based

Embarrassing parallel data-instensive applications

https://github.com/dispel4py/dispel4py

dispel4py parallel stream-based dataflow system

Graph

- Connections among PES
- Abstract workflow

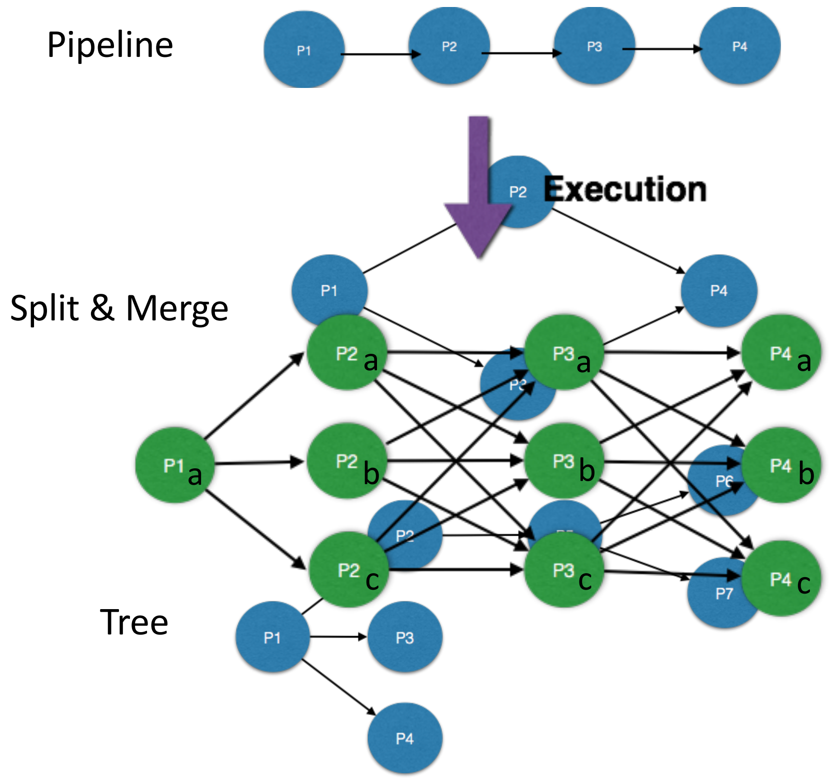
Instance

- Each PE is translated into one or more instances in run-time
- Each instance runs in a process
- dispel4py does it for you
- Concrete workflow

Mappings

- Sequential, multiprocessing, MPI

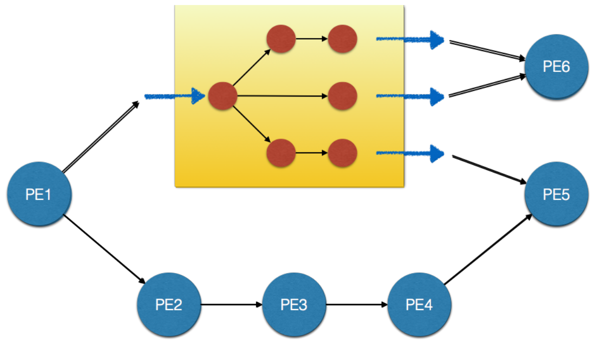
+ Example of graphs



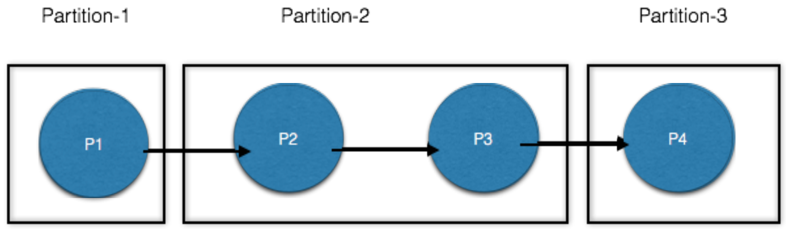
4 PEs & 10 processes

dispel4py parallel stream-based dataflow system

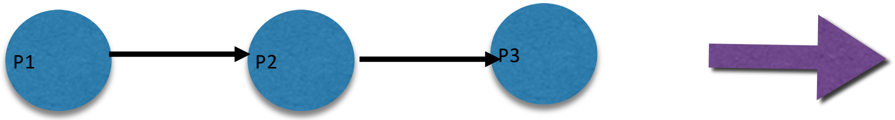
- Example of Composite PE

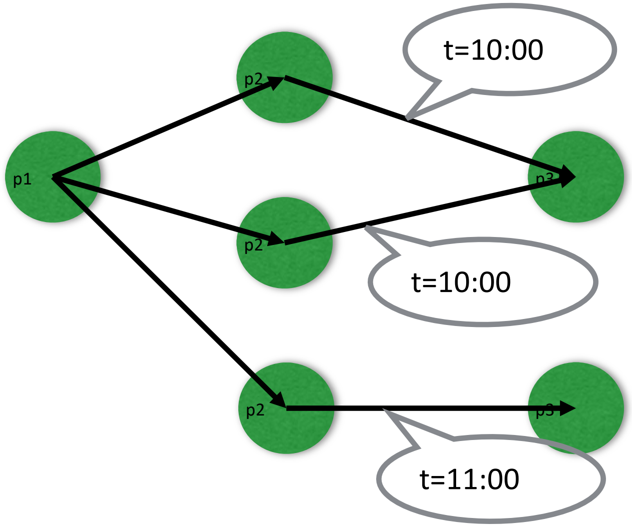


- Example of Partition



- Example of Grouping By a feature (MapReduce)

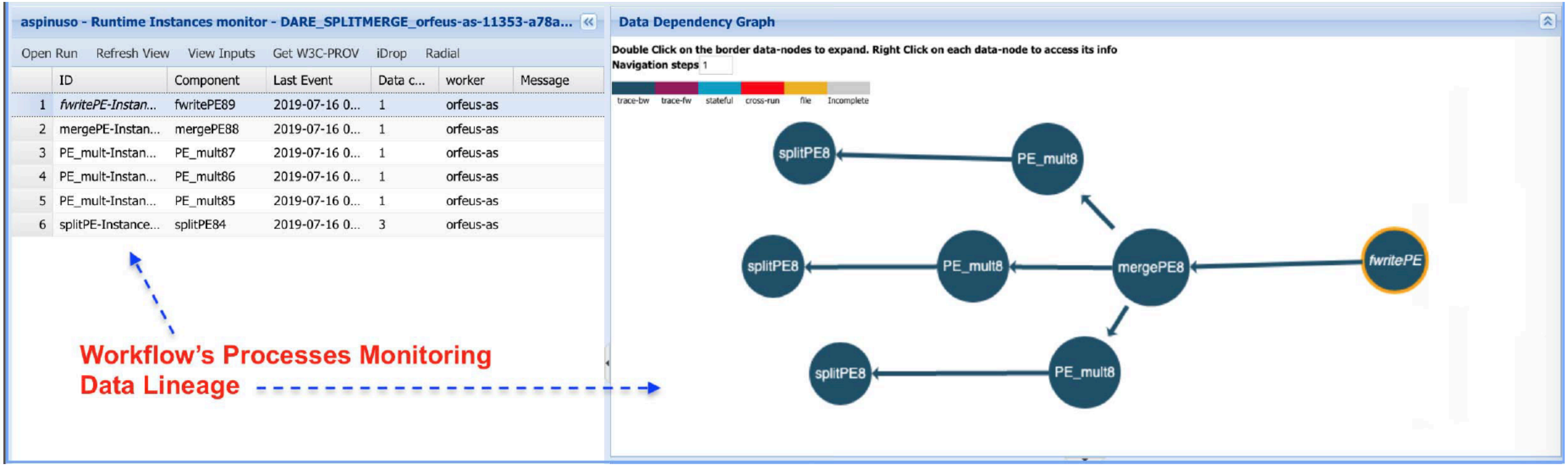




- All data items that satisfy the same feature are guaranteed to be delivered to the same instance of a PE

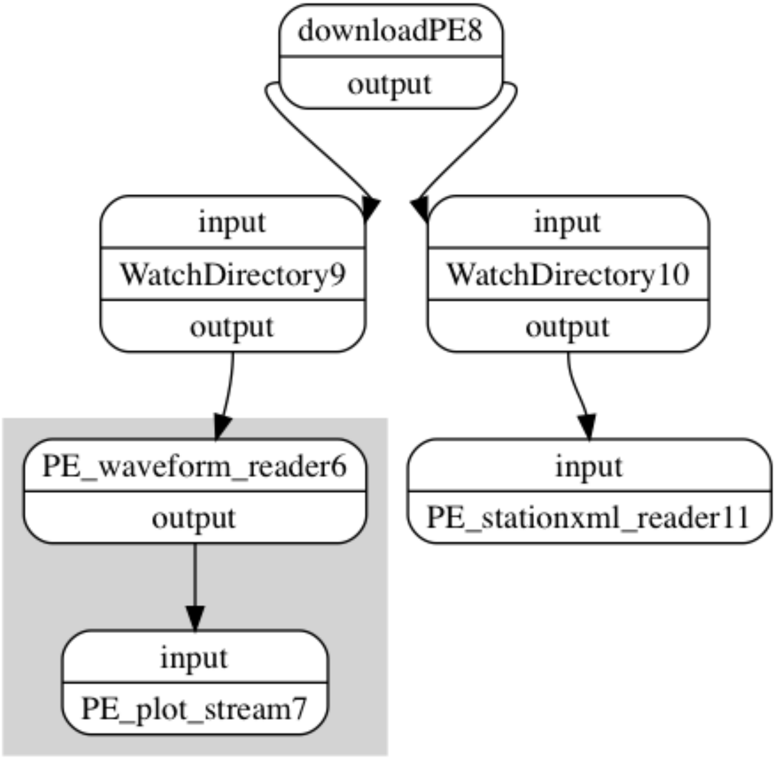
dispel4py parallel stream-based dataflow system

Provenance *Alessandro Spinuso*



Runtime provenance collection with selective paths and user-defined domain information.

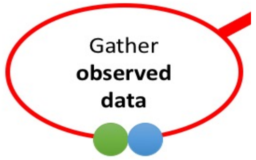
RA – Download observed data



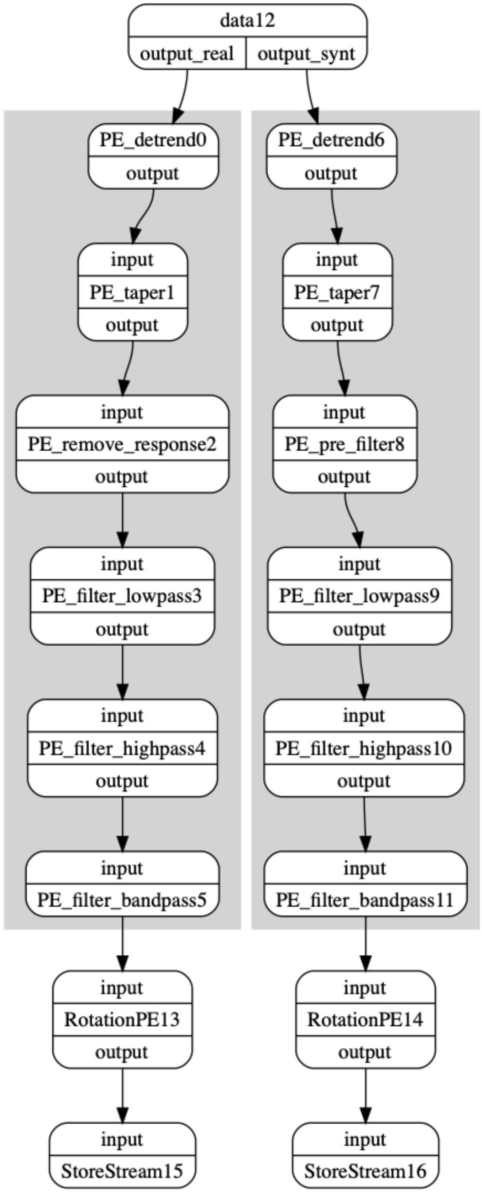
The workflow downloads real waveforms corresponding to the same earthquake.

downloadPE:
0:
input:
minimum_interstation_distance_in_m: 100
channel_priorities:
0: "BH[E,N,Z]"
1: "EH[E,N,Z]"
location_priorities:
0: ""
1: "00"
2: "10"
mseed_path: "./data"
stationxml_path: "./stations"
RECORD_LENGTH_IN_MINUTES: 1
ORIGIN_TIME: "2013-02-16T21:16:09.29"
minlatitude: 41.10007459633125
maxlatitude: 42.89777970948071
minlongitude: 12.041644551237324
maxlongitude: 14.439665626790928

Test Case: RA



RA – Pre-processing observed and synthetic data

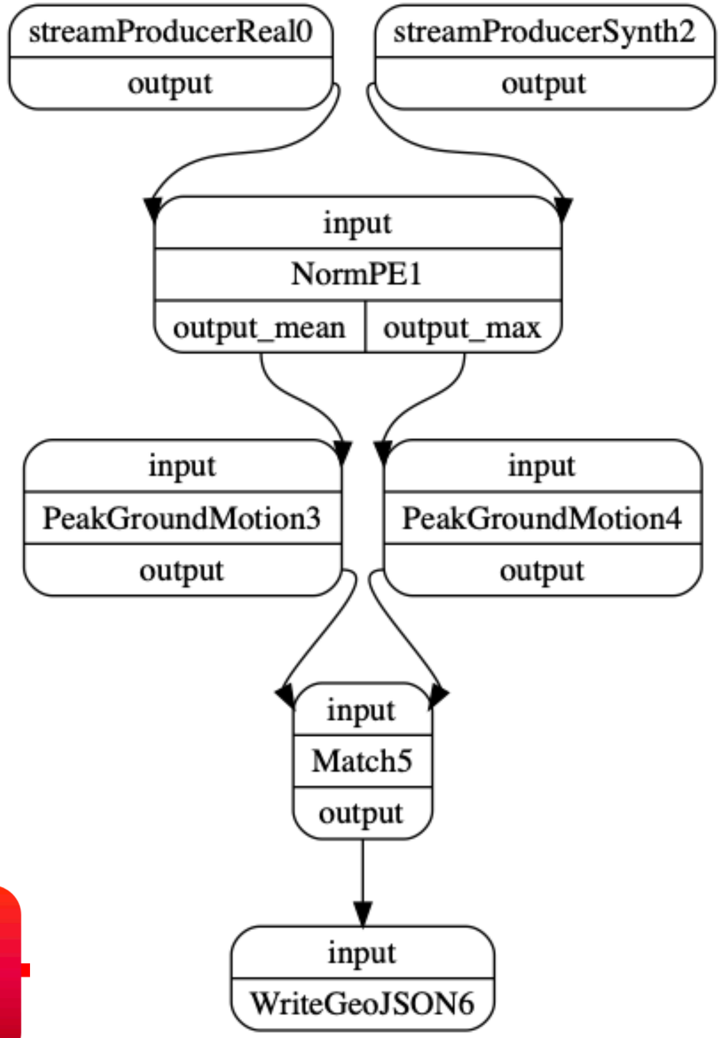


Similar preprocessing steps
in synthetic and observed data

Test Case: RA

Get
pre-processed
synt and data

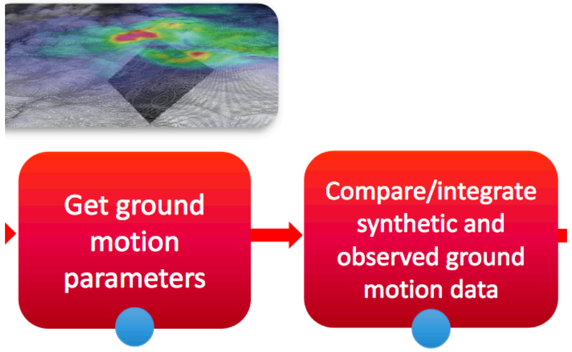
RA – Ground motion parameters



Ground motion parameters:
Peak ground values of **displacement**, **velocity** and **acceleration**.

Two types of normalisation – Mean & Max
Two set of PGM outputs – Max & Mean

Test Case: RA



RA – Ground motion parameters

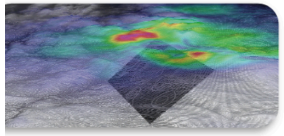
Peak ground values of Displacement (PGD), Velocity (PGV) Acceleration (PGA)

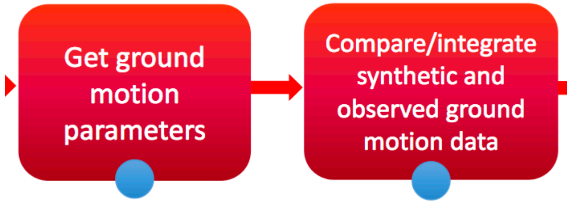
output

type: "Feature"
properties:
  station: "ARRO"
  data:
    PGD: 0.000030589700119155I5
    PGV: 0.0001538736268399262 7
    PGA: 0.0008035731527687157
    p_norm: "max"
    PSA_0.3Hz: 0.000171880777011629
    PSA_1.0Hz: 0.0020140831084336057
    PSA_3.0Hz: 0.0009812436987400703
  synt:
    PGD: 0.00003889927280500509 6
    PGV: 0.00011736045694475592
    PGA: 0.0003549366539822615 5
    p_norm: "max"
    PSA_0.3Hz: 0.0003942182089108563
    PSA_1.0Hz: 0.0007176179285863899
    PSA_3.0Hz: 0.0003707452407479844
  difference:
    PGD: -0.000008309572685849946
    PGV: 0.000036513169895170355
    PGA: 0.0004486364987864541
    PSA_0.3Hz: -0.0002223374318992273
    PSA_1.0Hz: 0.0012964651798472158
    PSA_3.0Hz: 0.0006104984579920859
  relative_difference:
    PGD: -0.2716460983102781
    PGV: 0.23729322980834633
    PGA: 0.5583020005592205
    PSA_0.3Hz: -1.2935561251517065
    PSA_1.0Hz: 0.6436999418834826
    PSA_3.0Hz: 0.6221680289778919
  geometry:
    type: "Point"
    coordinates: []

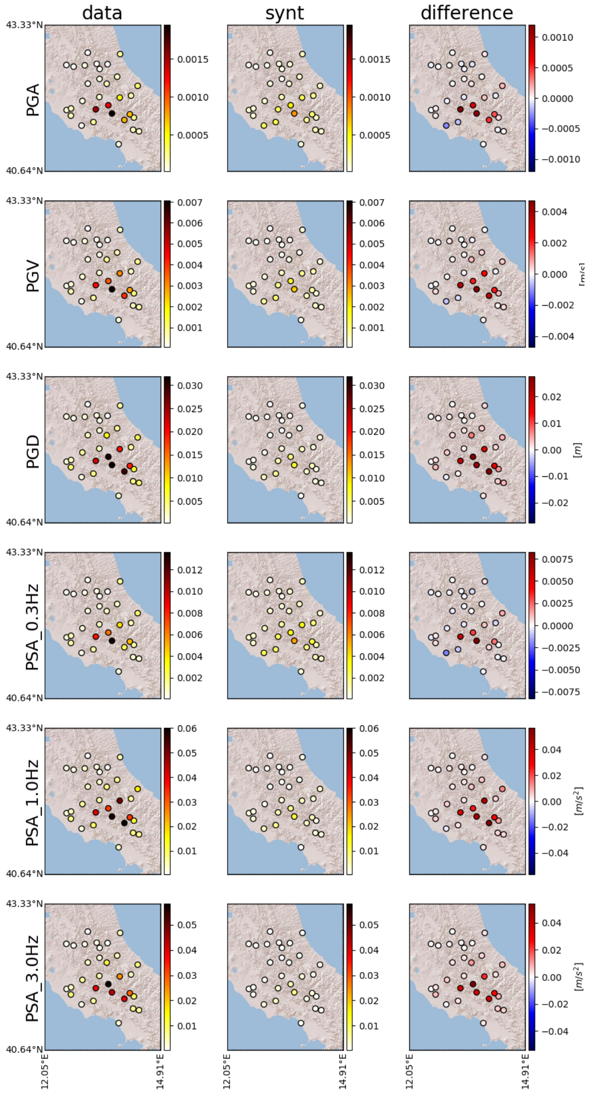
type: "Feature"
properties:
  station: "ARRO"
  data:
    PGD: 0.000030589700119155 15
    PGV: 0.0001538736268399262 7
    PGA: 0.0008035731527687157
    p_norm: "mean"
    PSA_0.3Hz: 0.000171880777011629
    PSA_1.0Hz: 0.0020140831084336057
    PSA_3.0Hz: 0.0009812436987400703
  synt:
    PGD: 0.000038899272805005096
    PGV: 0.00011736045694475592
    PGA: 0.0003549366539822615 5
    p_norm: "mean"
    PSA_0.3Hz: 0.0003942182089108563
    PSA_1.0Hz: 0.0007176179285863899
    PSA_3.0Hz: 0.0003707452407479844
  difference:
    PGD: -0.000008309572685849946
    PGV: 0.000036513169895170355
    PGA: 0.0004486364987864541
    PSA_0.3Hz: -0.0002223374318992273
    PSA_1.0Hz: 0.0012964651798472158
    PSA_3.0Hz: 0.0006104984579920859
  relative_difference:
    PGD: -0.2716460983102781
    PGV: 0.23729322980834633
    PGA: 0.5583020005592205
    PSA_0.3Hz: -1.2935561251517065
    PSA_1.0Hz: 0.6436999418834826
    PSA_3.0Hz: 0.6221680289778919
  geometry:
    type: "Point"
    coordinates: []

Test Case: RA

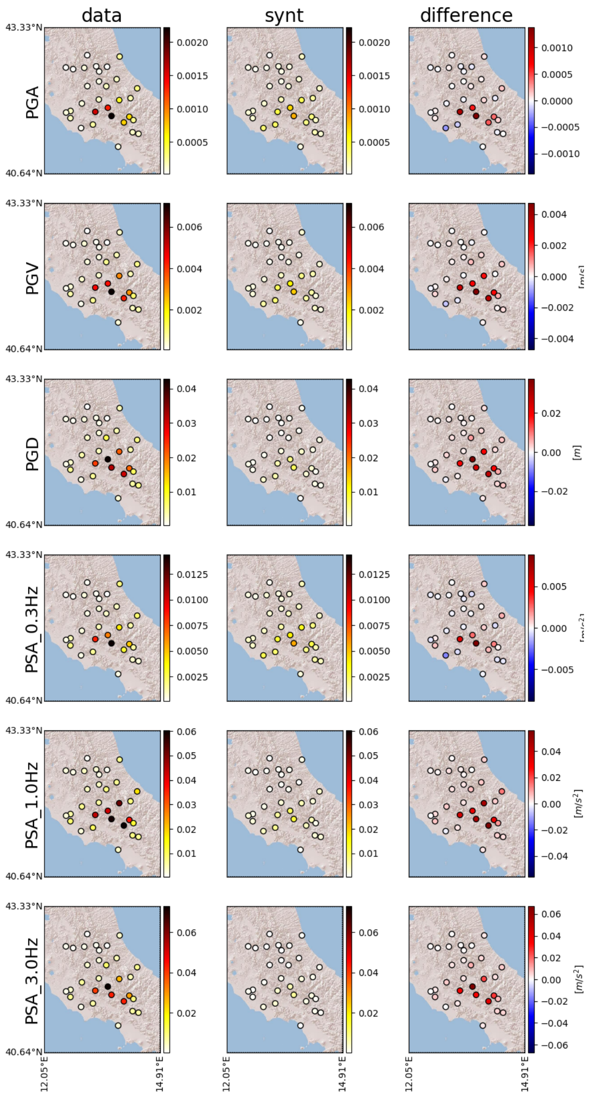




RA – Ground motion parameters maps



MAX



MEAN

Waveform propagation snapshots and **maps of ground motion parameters** are fundamental for a visual representation of the earthquake

ISTITUTO NAZIONALE DI GEOFISICA E VULCANOLOGIA

DARE

epcc

RA – Summary Steps (I)

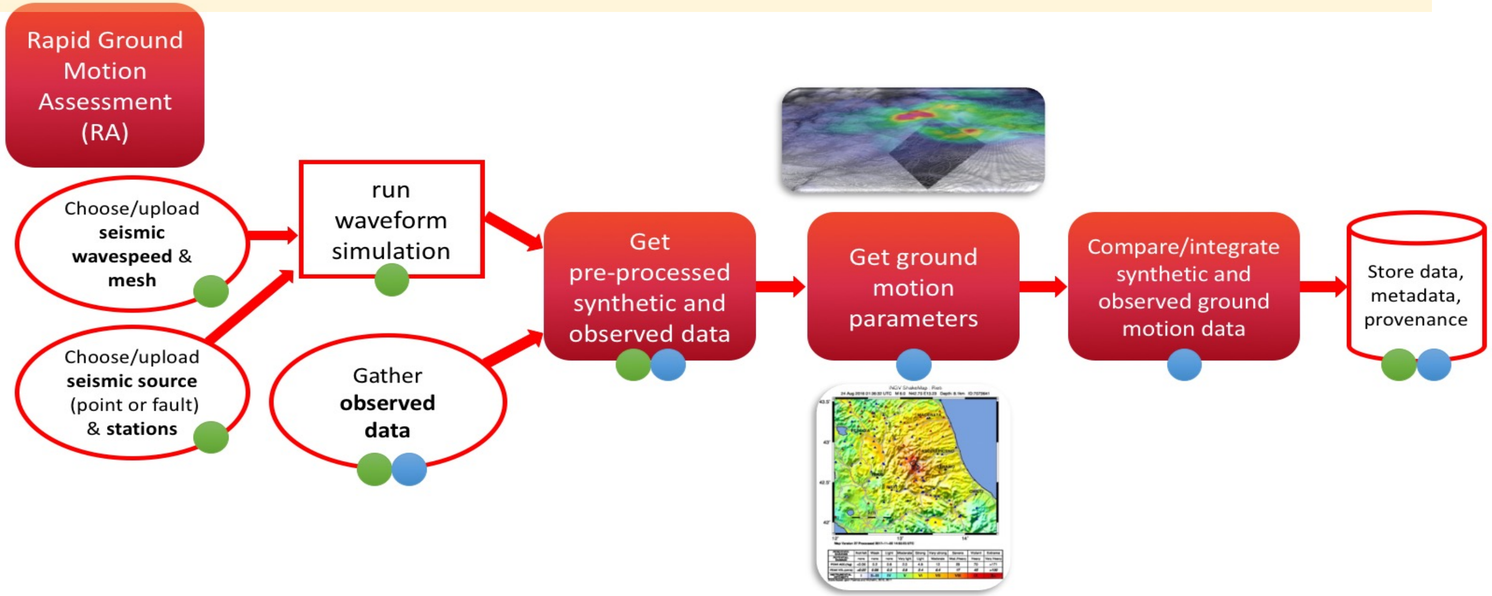
1. Dockerize Specfem3D

    Build a CWL workflow for generating synthetic data

2. Build **dispel4py** workflows to represent each part of the RA (**)

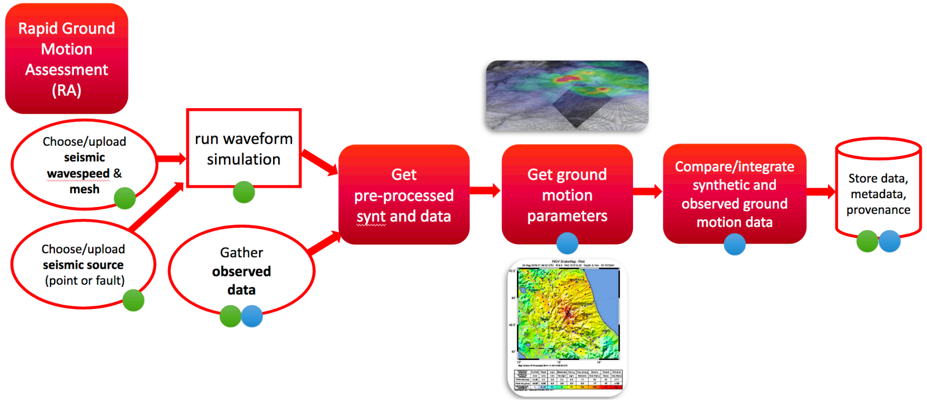
    (**) Except for the generation of the synthetic data

3. Use **CWL** to **connect RA** dispel4py workflows



RA – dispel4py + CWL

ISTITUTO NAZIONALE
DI GEOFISICA E VULCANOLOGIA



dispel4py + COMMON WORKFLOW LANGUAGE → semantics and descriptions

```
>> cwltool --provenance run-ra/ --full-name "Rosa Filgueira”
run_ra.cwl run_ra.yml
```

```
>> cwlprov --directory run-ra/ validate
Valid CWLProv RO: run-ra

>> cd run-ra/
>> cwlprov info
Research Object of CWL workflow run
Research Object ID: arcp://uuid,7b810637-40bb-467a-9c0a-a89865bf3c09/
Profile: https://w3id.org/cwl/prov/0.6.0
Workflow run ID: urn:uuid:7b810637-40bb-467a-9c0a-a89865bf3c09
Packaged: 2019-09-17

>> cwlprov who
Packaged By: cwltool 1.0.20190228155703 <urn:uuid:bbfe3471-2b87-4b11-975a-b96b2f24f4f8>
Executed By: Rosa Filgueira <urn:uuid:7d32cf62-d9c8-467e-8f58-5a74b70091e5>
```

```
#!/usr/bin/env cwl-runner

cwlVersion: v1.0
class: Workflow

inputs:
   create_env_script: File
   download_workflow: File
   download_argument_f: File
   preprocess_workflow: File
   ra_workflow: File
   ra_argument_d: string
outputs:
    misfit_data:
        type: Directory
        outputSource: preprocess_data/output
    pgm_data:
        type: Directory
        outputSource: rapid_assessment/output
steps:
  create_env:
    run: create_env.cwl
    in:
      script: create_env_script
    out: [output]
  download_data:
    run: dispel4py_download.cwl
    in:
      workflow:  download_workflow
      argument_f: download_argument_f
      misfit_data: create_env/output
    out: [output]
  preprocess_data:
    run: dispel4py_preprocess.cwl
    in:
      workflow: preprocess_workflow
      misfit_data: download_data/output
    out: [output]
  rapid_assessment:
    run: dispel4py-RA-pgm_story.cwl
    in:
      workflow: ra_workflow
      argument_d: ra_argument_d
      misfit_data: preprocess_data/output
    out: [output]
~
```

epcc

RA – Summary Steps (I)

1. **Dockerize Specfem3D** + CWL

2. **RA dispel4py** workflows

3. **CWL** to **connect RA** dispel4py workflows

Experiment I:

- Run all the steps of the RA in our laptops, small dataset, sequential mapping

Experiment II:

- Run the same codes using NSF-Chameleon cloud, MPI docker cluster, larger dataset, MPI mapping

CWL is in charge to execute and connect each part of the RA application.

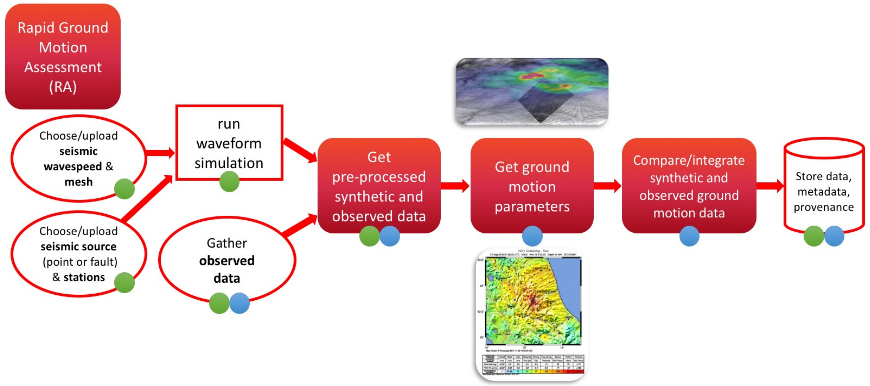
RA – Summary Steps (II)

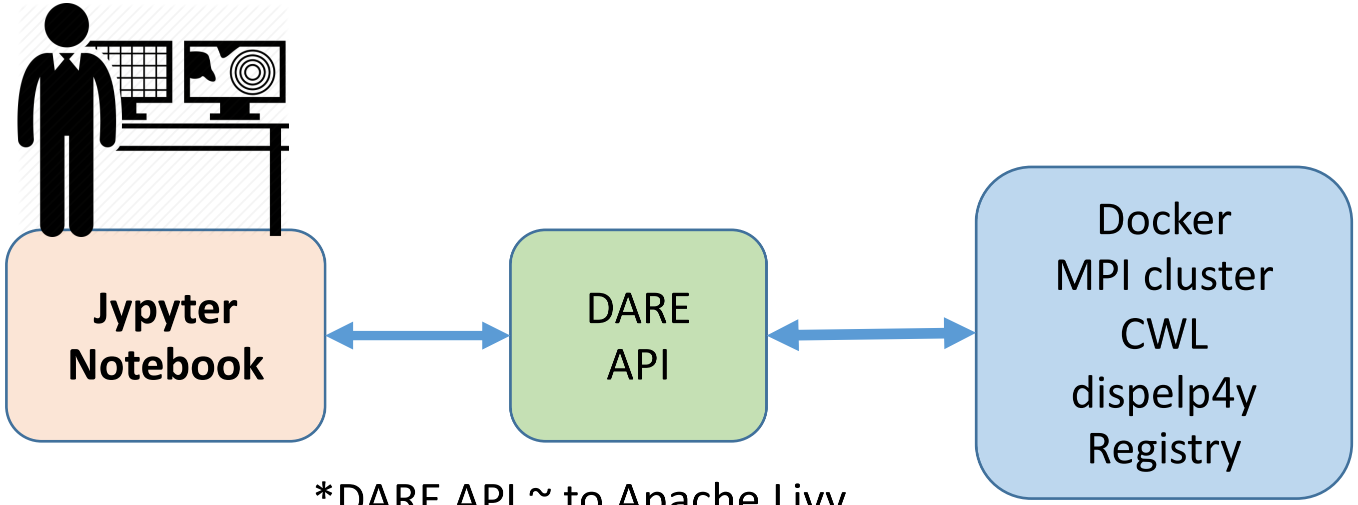
4. DARE API – Workflows as a Services

5. Orchestration with Kubernetes: MPI cluster, dispel4py, CWL, SPECFEM3D, Registry

6. Jupyter Notebooks to submit applications/workflows to the working environment:

    * Talk with the DARE API





*DARE API ~ to Apache Livy

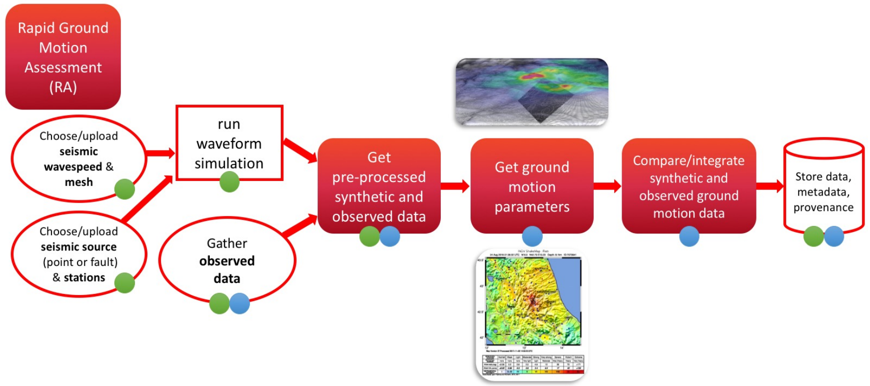
RA – Summary Steps (II)

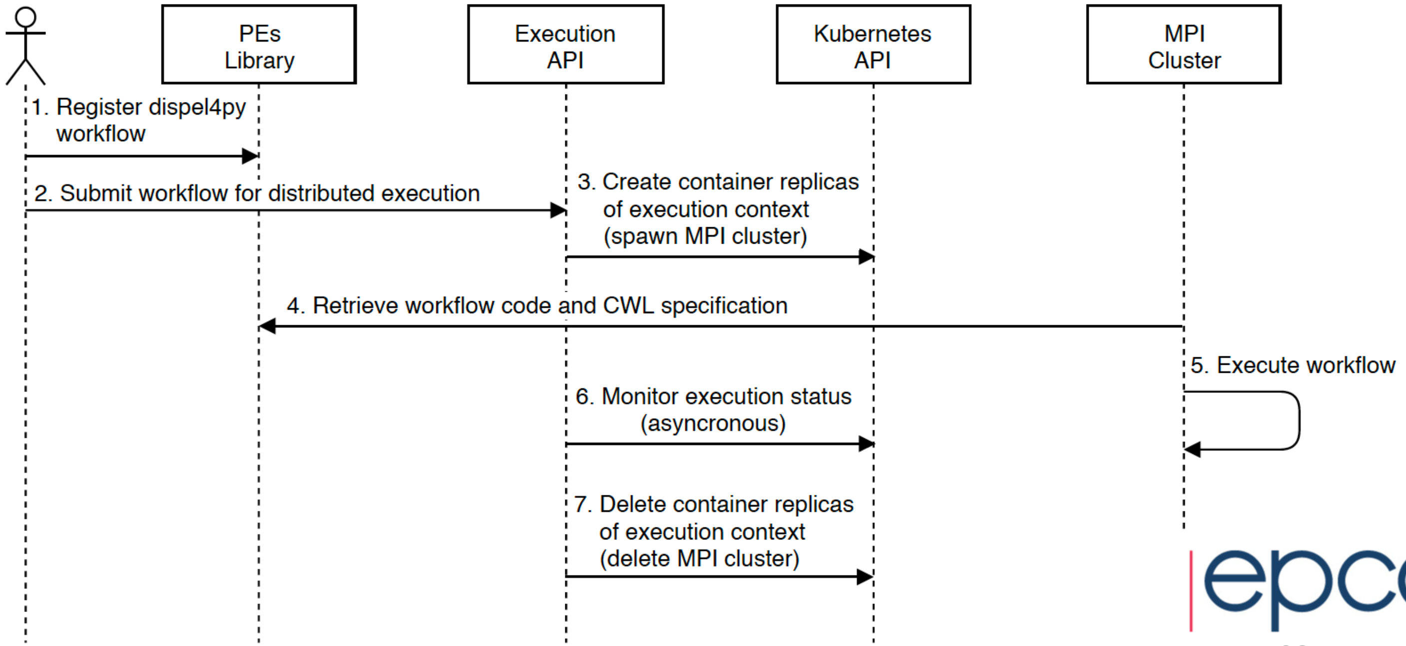
4. DARE API – Workflows as a Services

5. Orchestration with Kubernetes: MPI cluster, dispel4py, CWL, SPECFEM3D, Registry

6. Jupyter Notebooks to submit applications/workflows to the working environment:

    * Talk with the DARE API

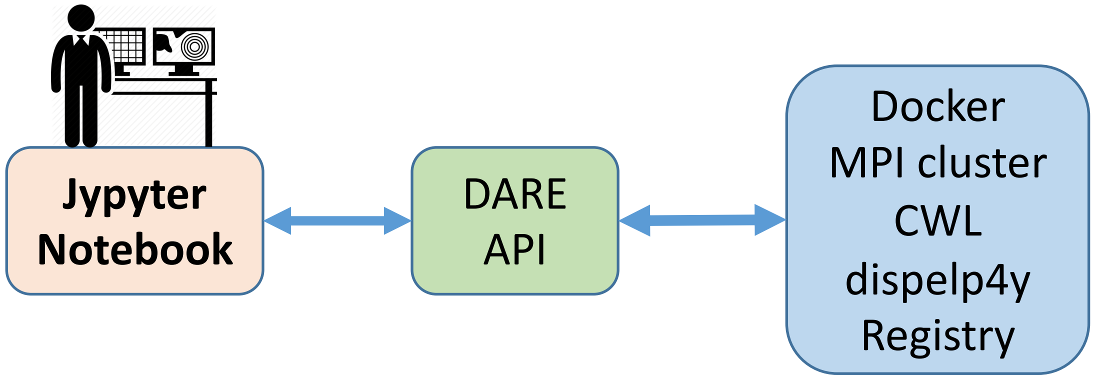


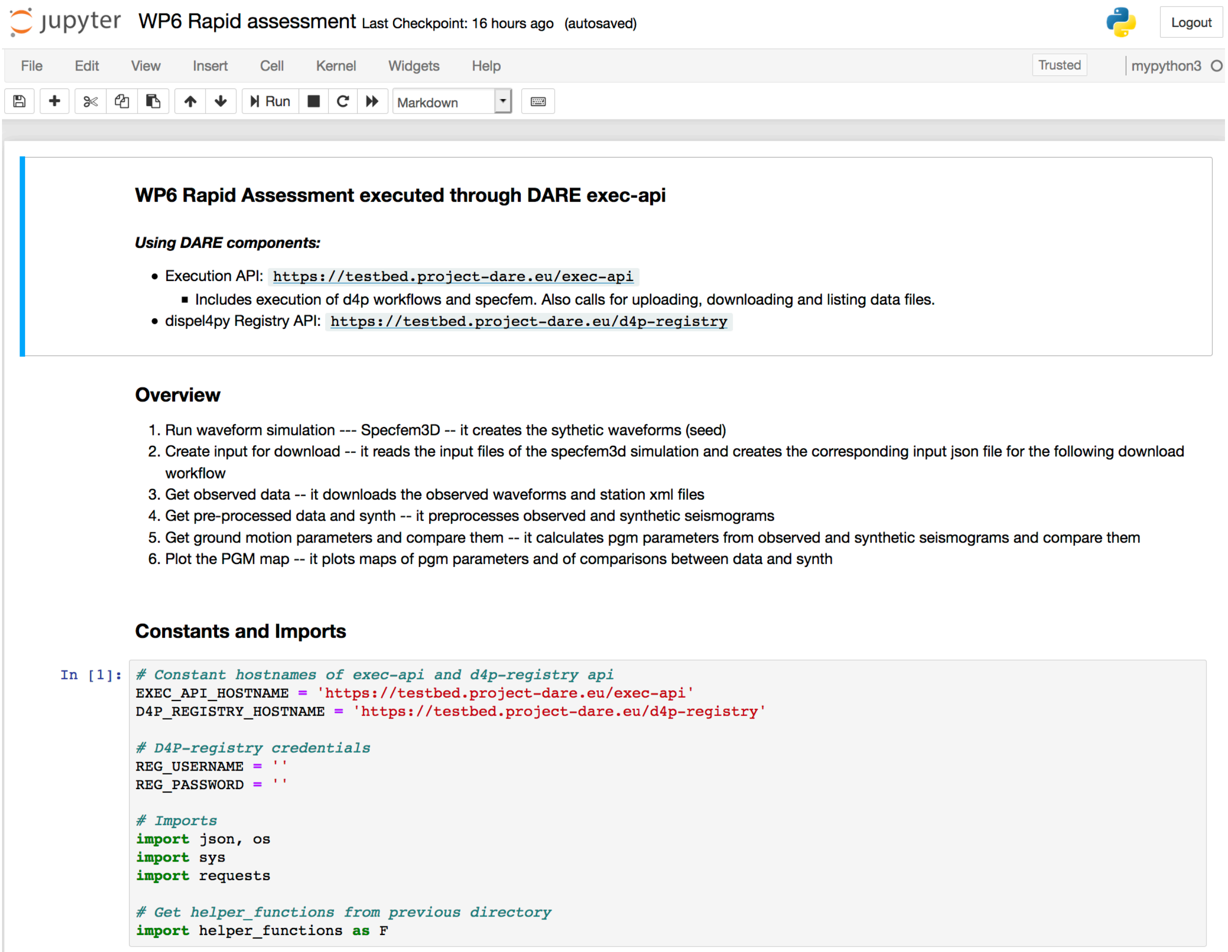


DARE API

ISTITUTO NAZIONALE
DI GEOFISICA E VULCANOLOGIA

- Web service
- Acts as an intermediary between:
    - users' applications
    - the underlying computing resources
- Provisions a computing environment
    - Docker MPI cluster spawned on demand / Kubernetes
- Runs and monitors an application
- Collect its provenance and results

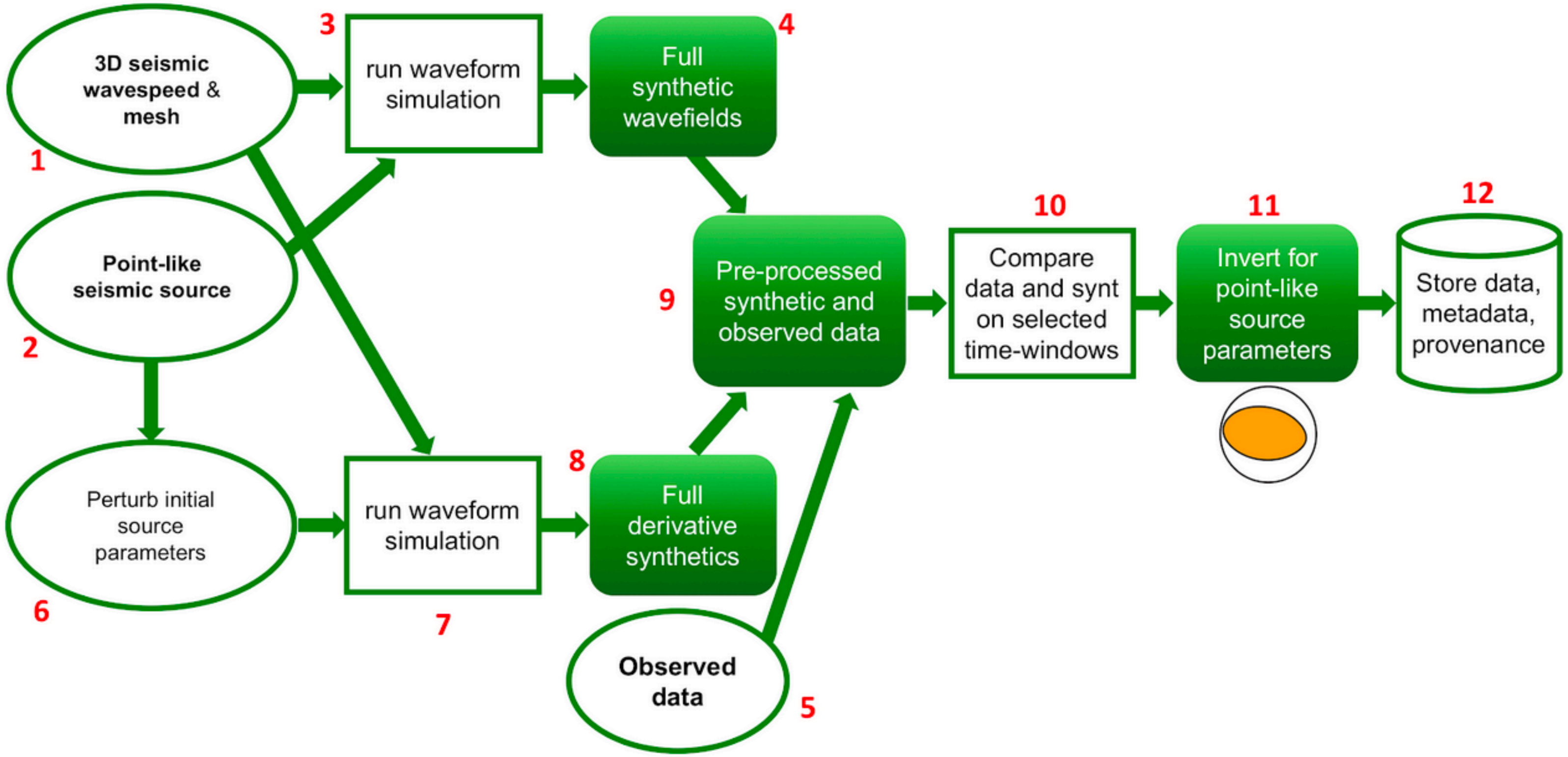




epcc

Seismic Source Analysis (SS) – New Use Case

Characterises the source parameters of an earthquake: **location**, **magnitude** and **rupture** mechanism.



Many simulations by perturbing the source parameters.
Analysis of the impact caused by **each simulation** on the ground motion.

Conclusions

New interfaces provide a fluent path from prototyping to production.

Data-intensive applications are not locked to platforms

* can be moved to suitable new platforms
* without human intervention

Abstract and semantic descriptions to allow reproducibility and portability.

Workflows-as-a-services (Waas): End-users do not need to set up any environment

Future Work

Workflows can be optimised intelligently without the user needing to do that

- New dispel4py mappings – dynamic deployment – *ZeroMQ*
- Handling errors - Recovery from failure(s)
- Automatic optimizations - exploiting data parallelism

CWLProv + dispel4py Provenance – integrate different levels of provenance

Delivering easy-to-use frameworks to empower data-driven research

Questions ?

DARE: A Reflective Platform Designed to Enable Agile Data-Driven Research on the Cloud, IEEE eScience 2019
Comprehensible control for researchers and developers facing data challenges, IEEEE eScience 2019

Thanks!

Dr. Rosa Filgueira
rosa.filgueira@ed.ac.uk

DARE

epcc