Next-generation HPC workflows for natural hazards

Overview of eflows4HPC Pillar “Urgent Computing” workflows

Marisol Monterrubio (BSC-CNS) and Louise Cordrie (INGV)

eFlows4HPC workshop – Barcelona, September 13th 2023

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 955558. The JU receives support from the European Union’s Horizon 2020 research and innovation programme and Spain, Germany, France, Italy, Poland, Switzerland, Norway.

URGENT COMPUTING

eFlows4HPC

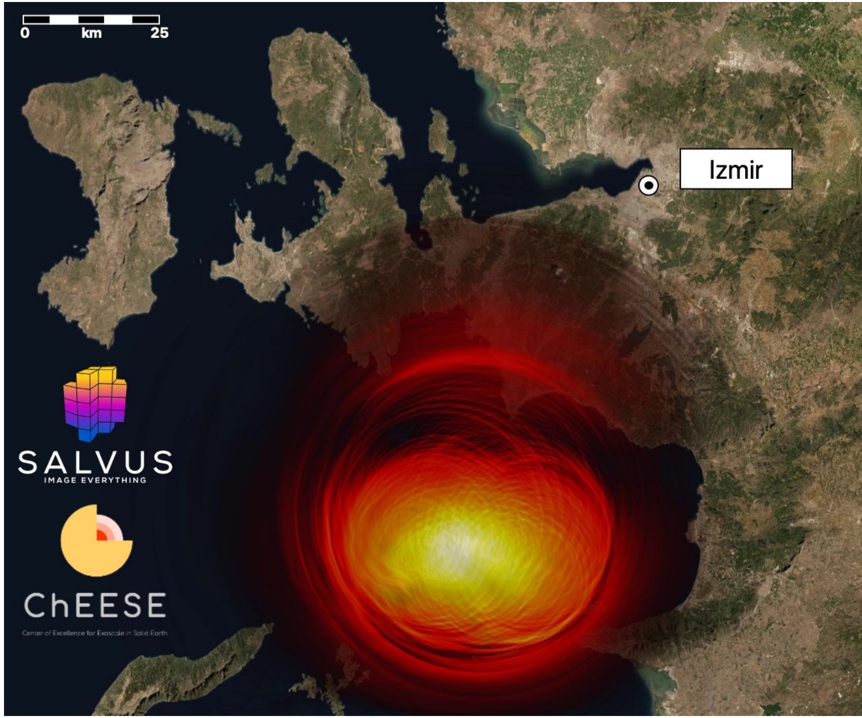
Links the advantage on:

- COMPUTING CAPACITY
- OPTIMIZED SIMULATION CODES
- DATA AVAILABILITY
- HPDA, ML

1. The computation operates under a strict deadline after which the computational results may have little practical value.

2. The onset of the **events** that need the computation is mostly **unpredictable.**

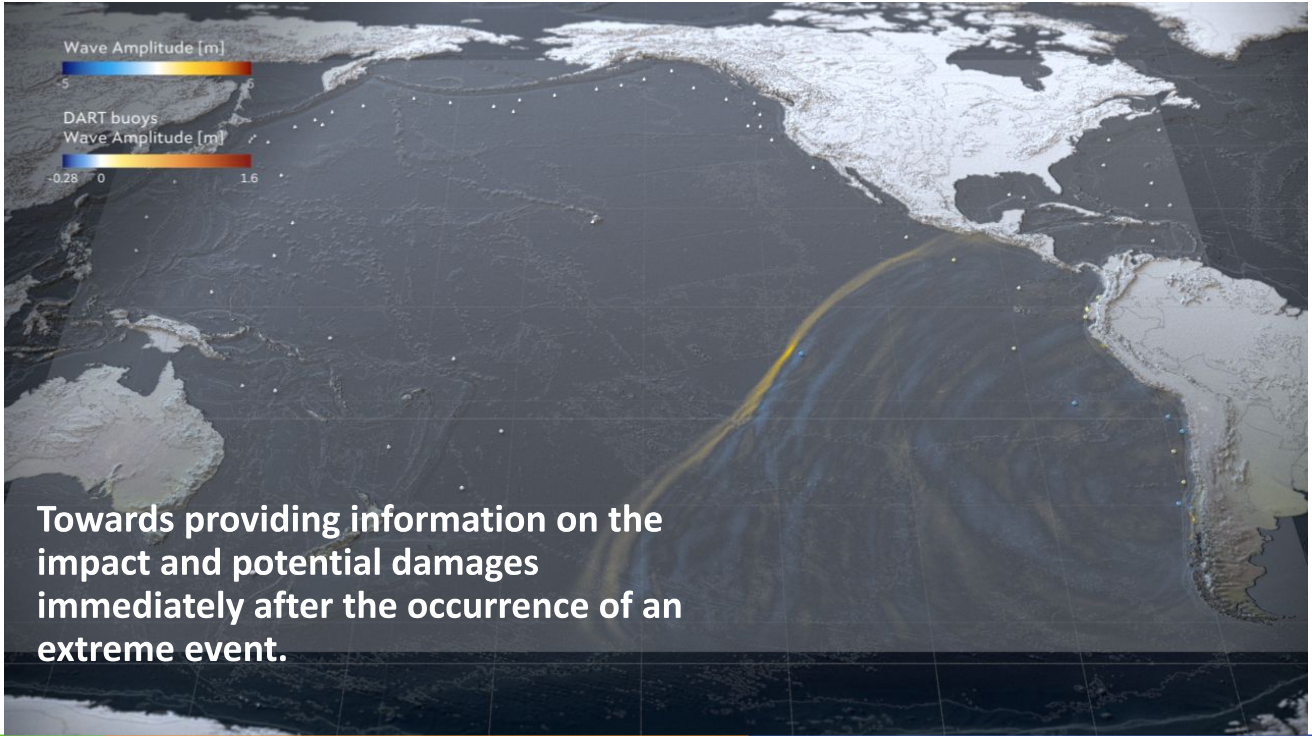
3. The computation requires **significant** computational resource usage.



URGENT COMPUTING FOR NATURAL HAZARDS

eFlows4HPC

Earthquakes and tsunamis are unpredictable and devastating events that can have catastrophic socioeconomic impacts.



Pillar III: Urgent computing for natural hazards

Earthquakes and Tsunamis

The development of UC workflows for earthquakes and tsunamis involves the deployments of advanced tools and developments of complex tasks to ultimately bring them to an operational level.

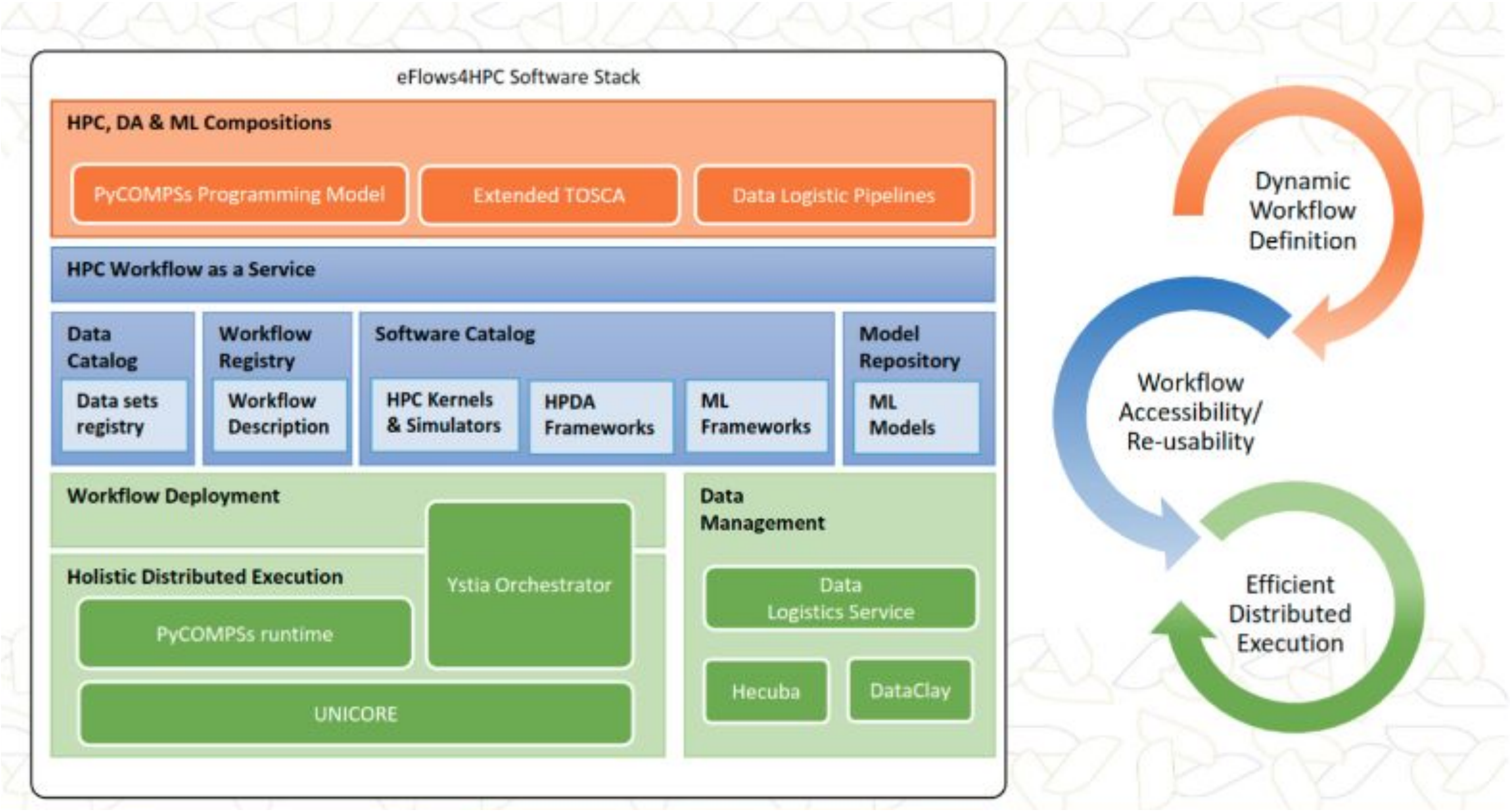
Challenges

eFlows4HPC

- Obtaining **high-resolution Earth models** (velocity models that define the properties of the subsurface).
- Rapidly **constraining source parameters** and accurately estimating the impact of parameter variations in the outcome of simulations, i.e. sensitivity to parameter uncertainties.
- Ensuring fast and reliable results with **urgent access to computational resources** and smart management of all workflow components.

Software Stack

eFlows4HPC



Natural hazard workflows and components

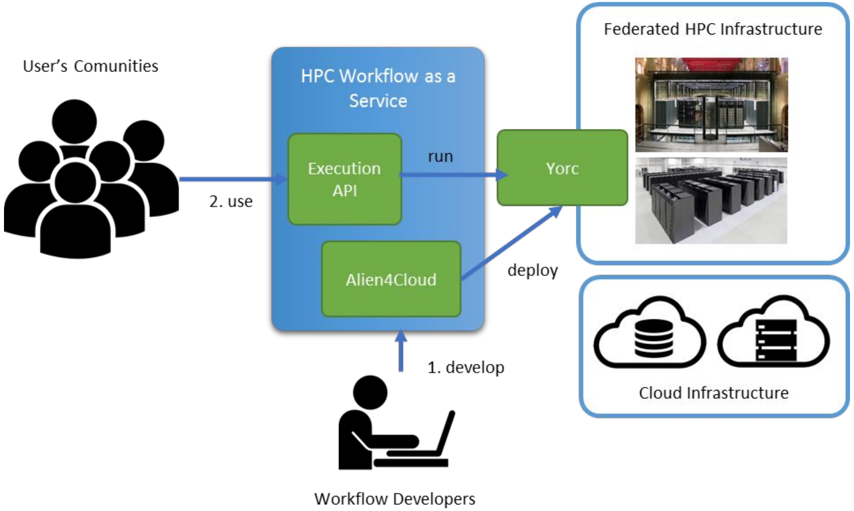
- **Workflows involved:**
  - Tsunami: PTF
  - Earthquakes: UCIS4EQ, MLESmap
- **Software Stack Components used:**
  - DA and ML: Dislib, EDDL, Ophidia
  - HPC Kernels: Salvus, HySEA
  - HPC, DA & ML Compositions: PyCOMPSs

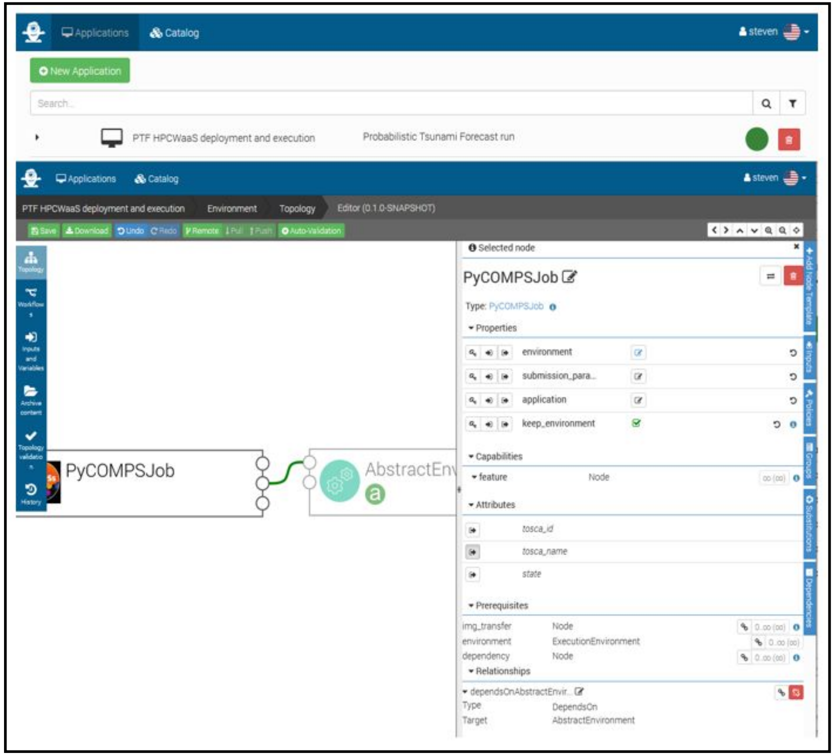
Software stack components	UCIS4EQ Workflow	MLESmap	PTF/FTRT Workflow
HPC Workflows	**PyCOMPSs**		
ML / AI	**dislib**	**dislib** **EDDL**	**EDDL**
Data Analysis			**Ophidia**

Deployment with HPC Workflows as a Service (HPCWaaS)

HPCWaaS:

- TOSCA: description of the workflow
- Alien4Cloud: development interface





Earthquake workflows

UCIS4EQ

MLESmap

Urgent Computing Integrated Services for Earthquakes

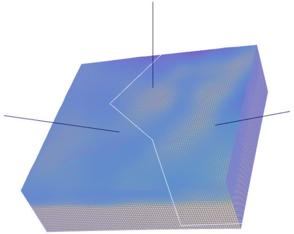
UCIS4EQ workflow

Urgent Computing for Seismic simulation

eFlows4HPC

Resilience Workflow: to provide fast outcomes using a fully automatic workflow

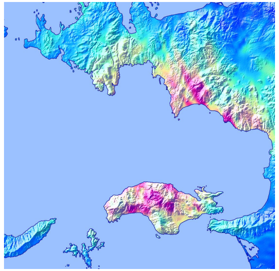
analysis



pre-processing

SALVUS
IMAGE EVERYTHING

HPC simulations



post-processing

Source location
Magnitude estimation
Grid generation
Ask for resources

Launch in supercomputer
Filter data
Build maps
Uncertainty quantification
Reach conclusions

minutes / hours

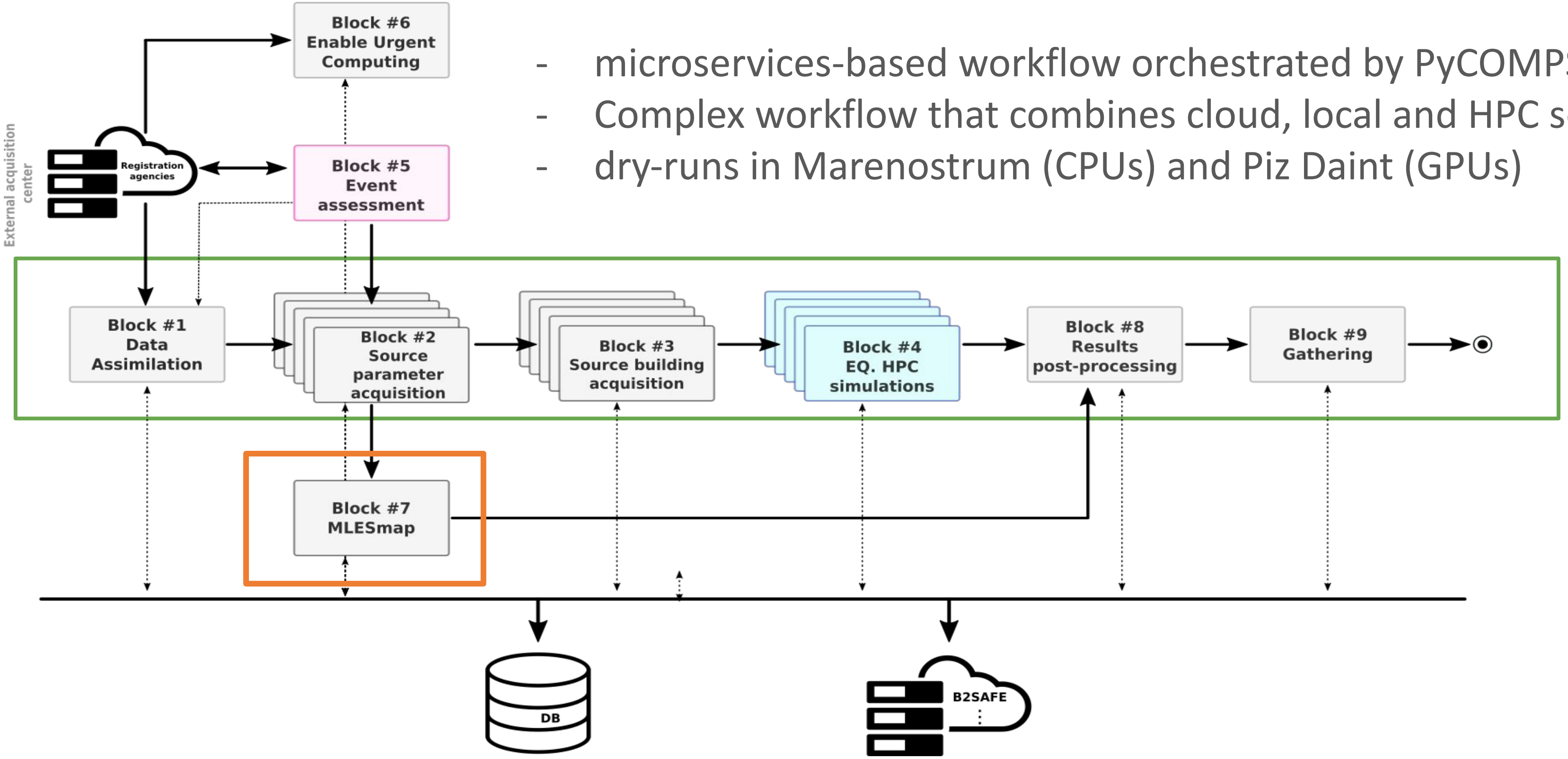
3D-physics based seismic simulations:

- Full time-histories
- Uniform sampling in space
- Sensitive in different ways to uncertainties than current approaches

The high resolution of this approach can complement the information of the GMM.

UCIS4EQ: Urgent Computing Integrated Services for Earthquakes

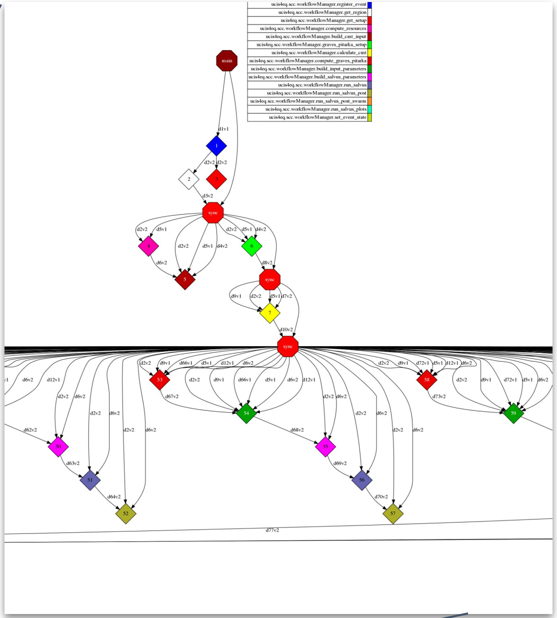
- microservices-based workflow orchestrated by PyCOMPSs
- Complex workflow that combines cloud, local and HPC services
- dry-runs in Marenostrum (CPUs) and Piz Daint (GPUs)

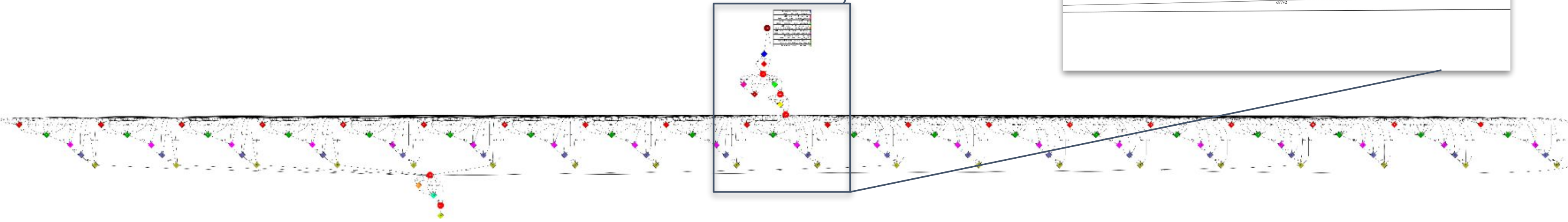


PyCOMPSs orchestration of microservices

- PyCOMPSs adapted to the micro-services design structure and integrated into UCIS4EQ.
- PyCOMPSs has been extended with the **@http decorator.** It allows developers to define a task that performs an HTTP request

```
@on_failure(management='CANCEL_SUCCESSORS')
@http(request="POST", resource="Graves-Pitarka", service_name="slipgen",
      payload='{ "event" : {{alert}}, "id" : {{event_id}}, "CMT" : {{cmt}}, \
                 "trial" : "{{path}}", "region": {{region}}, "setup" : {{setup}}, \
                 "resources" : {{resources}} , "ensemble" : "{{ensemble}}" }',
      produces='{"result" : "{{return_0}}"}')
@task(returns=1)
def compute_graves_pitarka(event_id, alert, path, cmt, region, setup, resources, ensemble):
    pass
```





HPC Workflow implemented in PyCOMPSs

eFlows4HPC

PyCOMPSs workflows to orchestrate different HPC executions

Before:

→ every execution in the HPC system was performed in a separate service call with its corresponding overhead

→ every system has its own job scheduler, the original UCIS4EQ workflow implements a set of adaptors to submit the job in the HPC schedulers of every machine

Now:

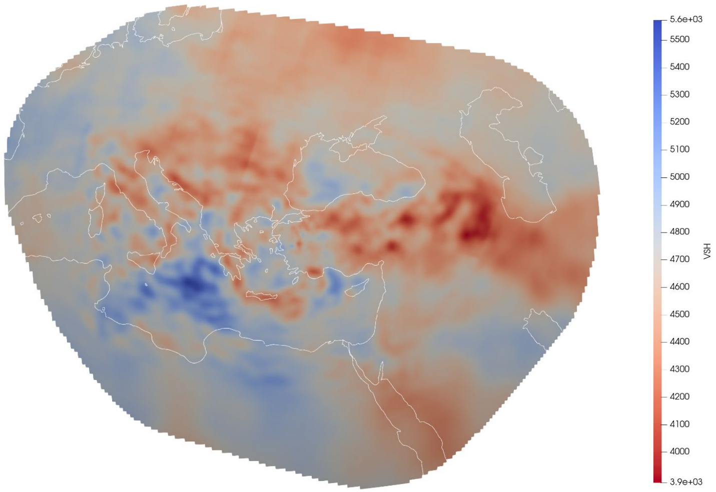
⟶ This Workflow is called from the microservices workflow which submits the HPC Workflow using the **PyCOMPSs queuing scripts** which already **supports different schedulers that has the same execution interface**

- *slipgen* which runs the slip generation using a singularity image
- *salvus_prepare* and *salvus_post* which executes the Salvus preprocessing and postprocessing as normal python tasks,
- and *salvus_run* which performs the simulation with Salvus defined as an MPI application

```
@container(engine="SINGULARITY", image="$SLIPGEN_IMAGE", options="-e --bind {{workingdir}}:/workspace/ --pwd /workspace")
@binary(binary="/opt/scripts/launcher.sh", args="-o rupture --dt {{dt}} -v {{fk_file}} -s {{input_src}}" , working_dir="{{workingdir}}")
@task(input_src=FILE_IN, fk_file=FILE_IN, workingdir=DIRECTORY_INOUT)
def slipgen(input_src, dt, fk_file, workingdir):
    pass

@task(input_data=FILE_IN, rupture=DIRECTORY_IN, salvus_setup=FILE_IN, working_dir=DIRECTORY_INOUT)
def salvus_prepare(input_data, rupture, salvus_setup, working_dir):
    rupture_file = rupture + "/scratch/outdata/rupture/rupture.srf"
    os.chdir(working_dir)
    pre_process(input_data, rupture_file, salvus_setup, working_dir)

@mpi(runner="mpirun", binary="$SALVUS_BINARY", args="compute {{prepare_path}}/salvus_input_rupture.toml", processes="$SALVUS_PROCESSES" ,
processes_per_node= "$SALVUS_PPN", working_dir="{{working_dir}}")
@task(prepare_path=DIRECTORY_IN, working_dir= DIRECTORY_INOUT)
def salvus_run(prepare_path, working_dir):
    pass

@task(UC_input = FILE_IN, salvus_setup= FILE_IN, grid_coordinates= DIRECTORY_IN, simu_folder=DIRECTORY_IN, output_path=DIRECTORY_INOUT)
def salvus_post(UC_input, salvus_setup, grid_coordinates, simu_folder, output_path):
    process_outputs_grid( UC_input, salvus_setup,  grid_coordinates, simu_folder, output_path=output_path)

if __name__ == "__main__":

    slip_id, input_path, salvus_setup, slip_input_src, region_fk1d, dt = parse_arguments()
    slipgen_dir, salvus_wrapper_dir, salvus_dir, salvus_post_dir = create_output_dirs()

    slipgen(slip_input_src, dt, region_fk1d, slipgen_dir)
    salvus_prepare(input_path, slipgen_dir, salvus_setup, salvus_wrapper_dir)
    salvus_run(salvus_wrapper_dir, salvus_dir)
    salvus_post(input_path, salvus_setup, salvus_wrapper_dir, salvus_dir, salvus_post_dir)
```

UCIS4EQ Inputs

Simulations are sensitive to model inputs



- Earth models (HPC or remote repository)

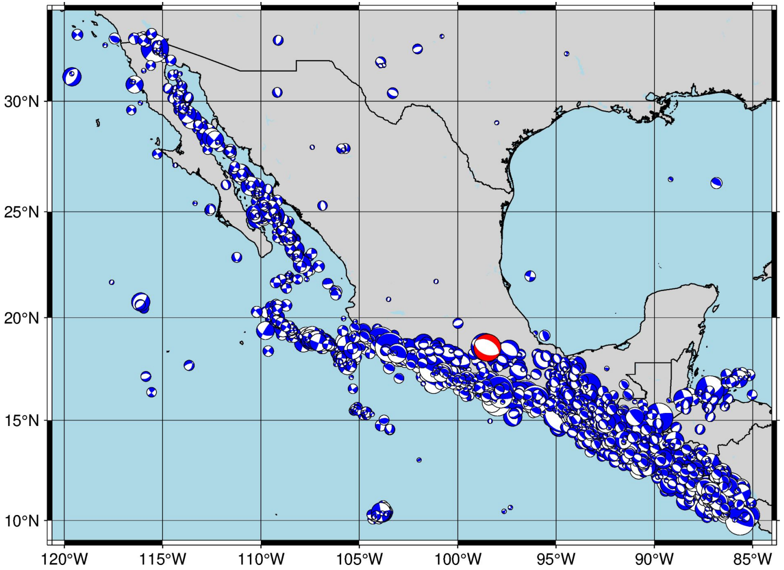
UCIS4EQ requires reliable Earth models for the forward modelling of ground motions.

The second generation Collaborative Seismic Earth Model (CSEM) – a multiscale global tomographic Earth model that incorporates a range of local-, regional- and global-scale updates – has been integrated into the UCIS4EQ workflow

UCIS4EQ Inputs

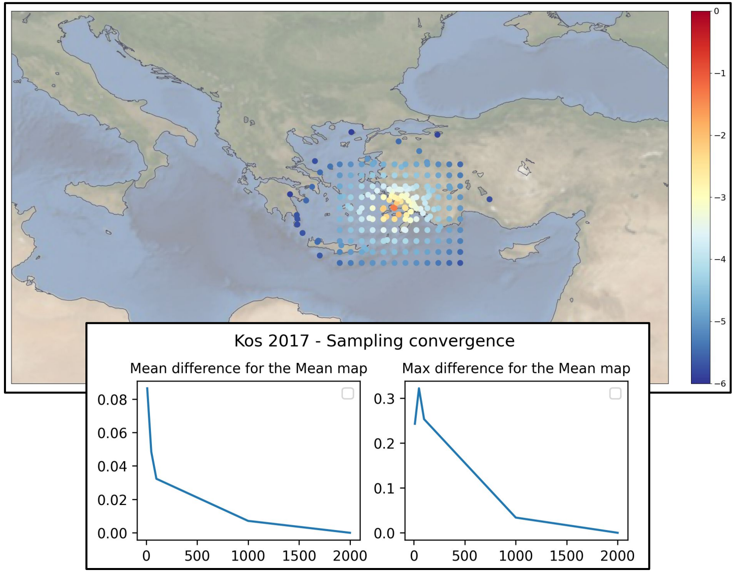
- Ensemble methodologies

Statistical based on historical events



Monterrubio-Velasco, M., et al. (2022). *Frontiers in Earth Science*, 339.

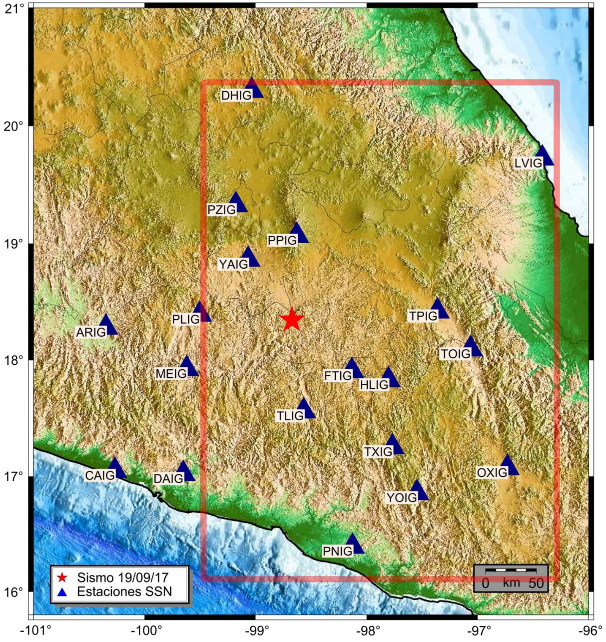
Probabilistic approach: SeisEnsMan



Stallone, A., et al., *International Union of Geophysics and Geodesy General Assembly* 2023

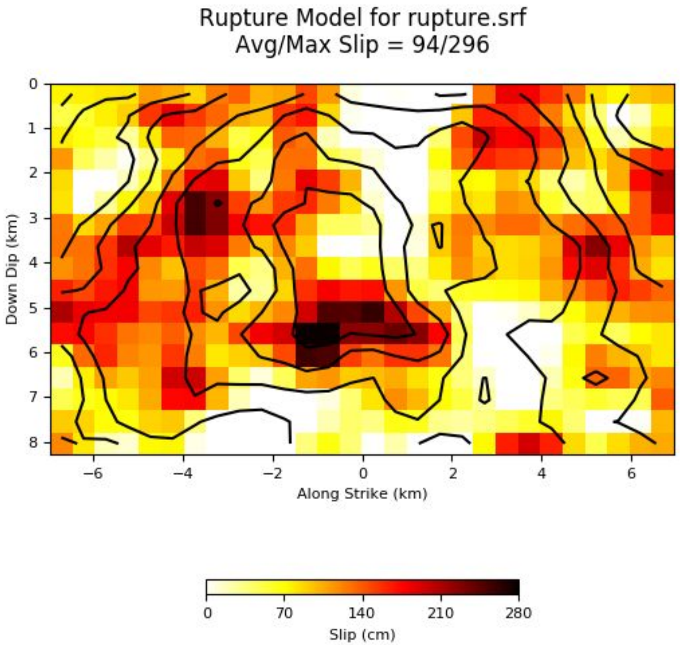
UCIS4EQ Inputs

- Receivers



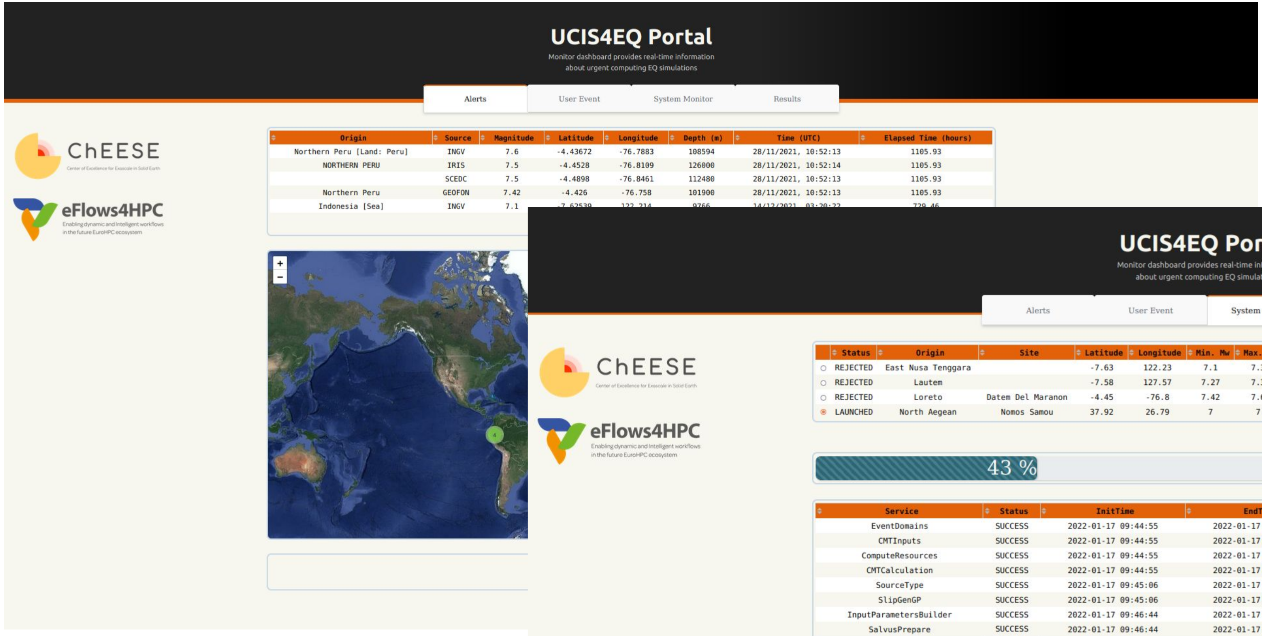
Selecting the stations and the receivers on the simulation domain.

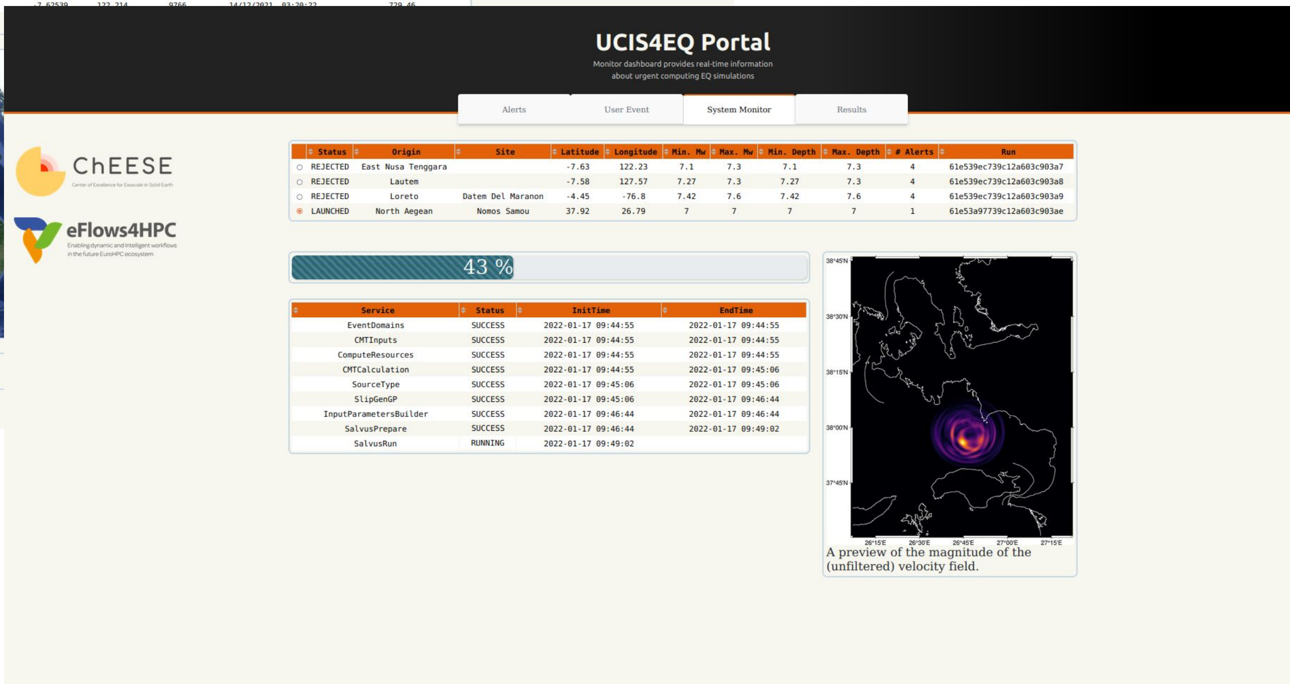
- Kinematic finite-fault



Generating the kinematic finite-fault history using the Graves-Pitarka rupture code

UCIS4EQ Front-end -- GUI





UCIS4EQ -- Use cases

eFlows4HPC

- **Mediterranean Sea:**
  - 2017 M6.6 Kos-Bodrum earthquake, 120 km x 100 km
  - **2020 M7.0 Samos-Izmir earthquake, 140 km x 110 km**

- **Iceland:**
  - **2000 (June 17) M6.4 SISZ earthquake**, 135 km x 85 km domain
  - 2000 (June 21) M6.5 SISZ earthquake, 135 km x 85 km domain

- **México:**
  - 2017 M7.1 Puebla earthquake, 200 km x 150 km

Mediterranean Sea

Mw 7.0 Samos-Izmir, 2020

- Off-shore the North coast of Samos Island in the eastern Aegean Sea
- 2020-10-30 11:51:27 (UTC)
- 118 fatalities, ~ 100 injuries, collapse of structures
- Local high-intensity effects, Tsunami run-up

Source: https://earthquake.usgs.gov/earthquakes/eventpage/us7000c7y0/shakemap/pga

Mw 7.0 Samos-Izmir, 2020

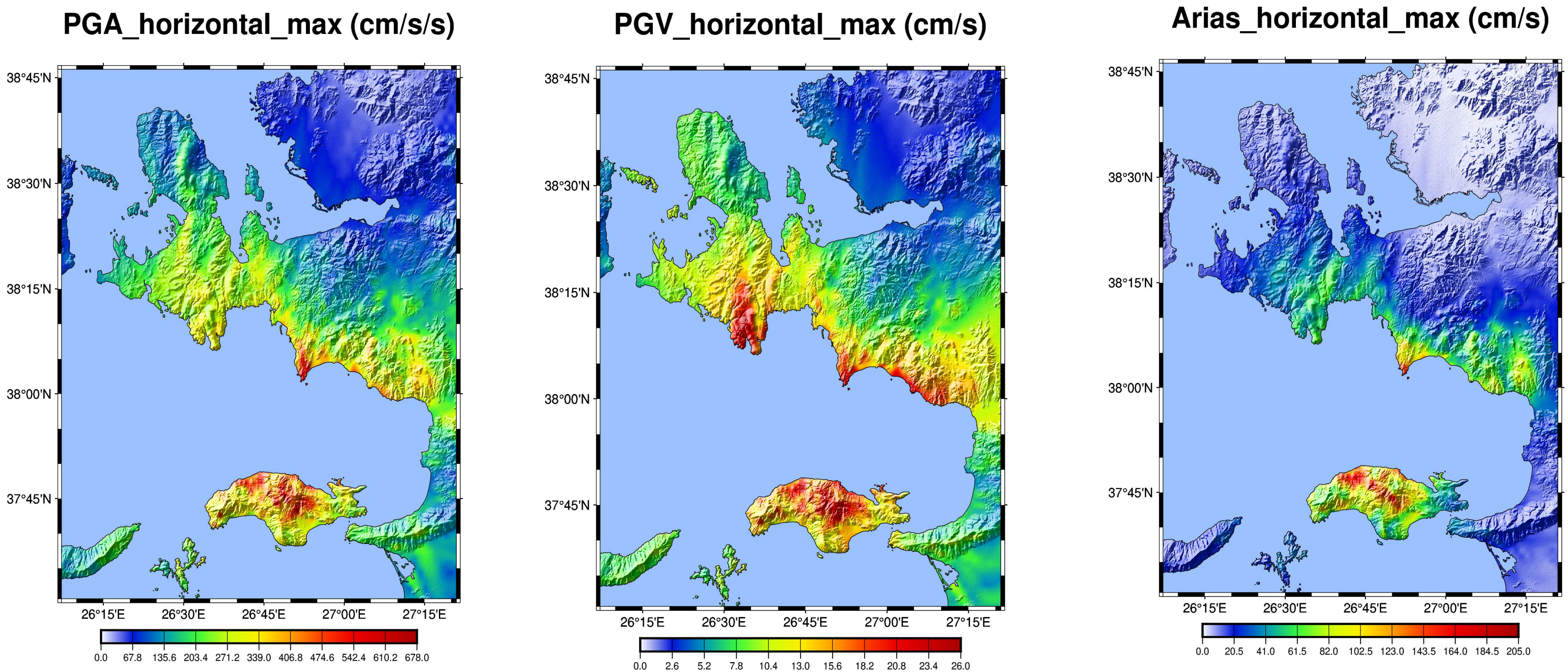
eFlows4HPC

UCIS4EQ configuration

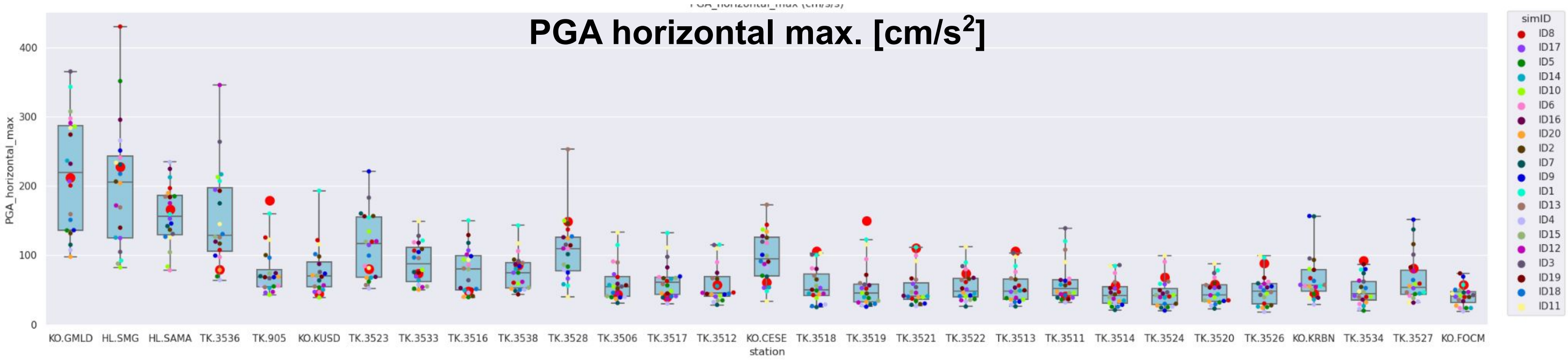
- 4,012,250 number of mesh elements
- Domain: 110km in longitude, 140km in latitude, and 35km in depth
- **Up to 5 Hz**
- 22 simulations in the ensemble
- 90 GPUs (Piz Daint) per WF execution

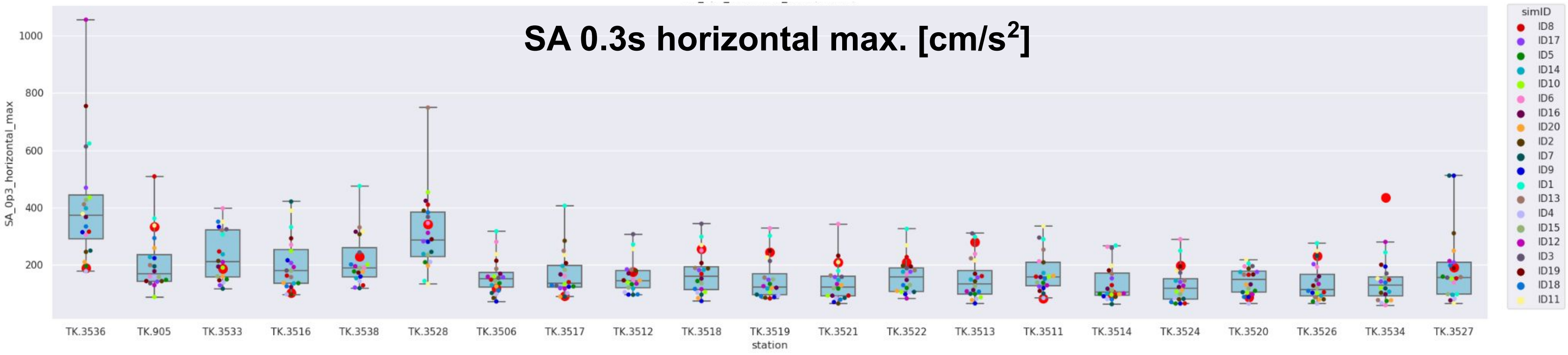
Use cases: Mw 7.0 Samos-Izmir, 2020

- **1h20m** wallclock per WF execution



Use cases: Mw 7.0 Samos-Izmir, 2020





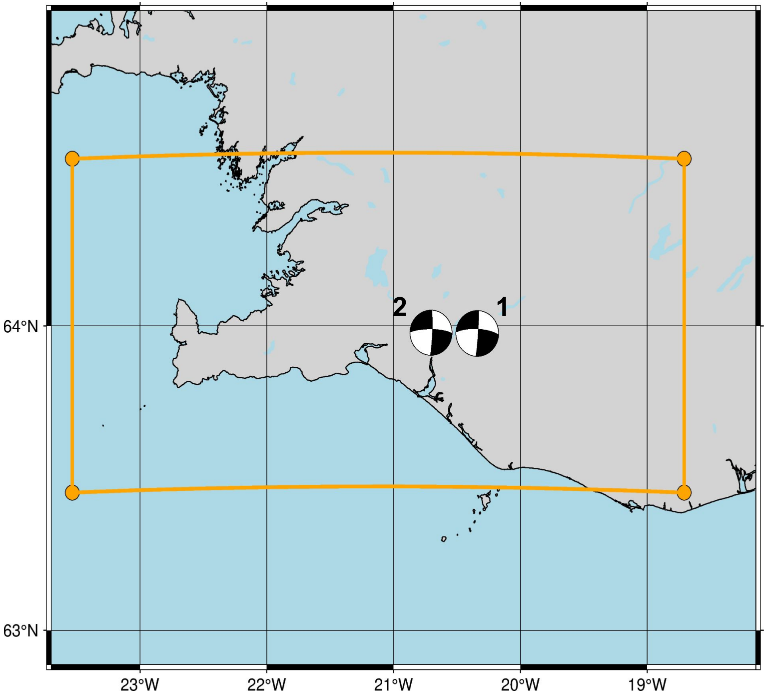
Use cases: 2000 doublet SISZ earthquakes

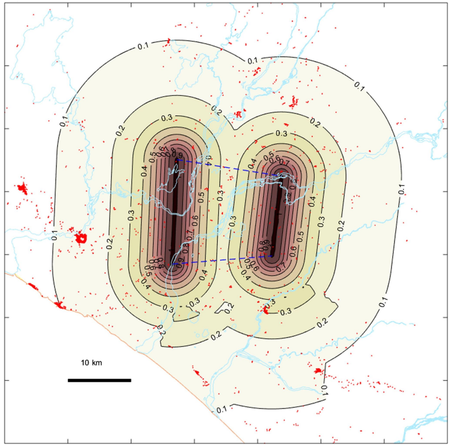
1. M_w 6.5, 17/06/2000

   Lat: 63.98º
   Lon: -20.34º
   Depth: 6.3 km
   Focal mechanism:
   {strike:273 , dip:74, rake:-3}

2. M_w 6.4, 21/06/2000

   Lat: 63.98
   Lon: -20.70º
   Depth: 5.1 km
   Focal mechanism: :
   {strike:271 , dip:77, rake:-5}



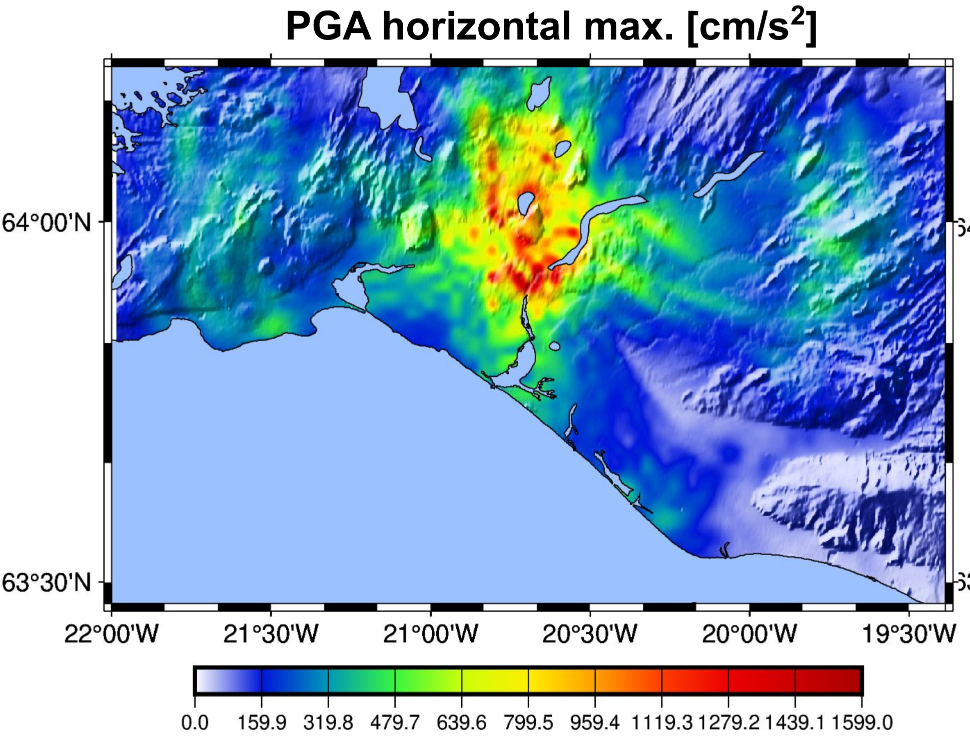


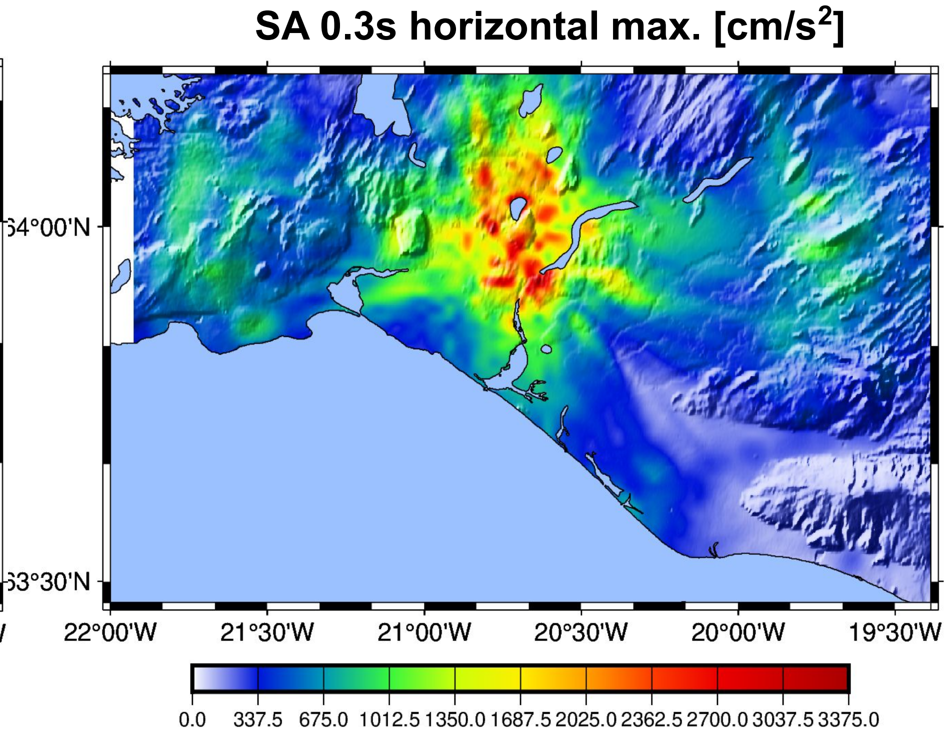
Source: **Shake map for the two June 2000 earthquakes in South Iceland in Bessason, B., Bjarnason, J. Ö., & Rupakhety, R. (2020)**..

Mw 6.4, 21/06/2000

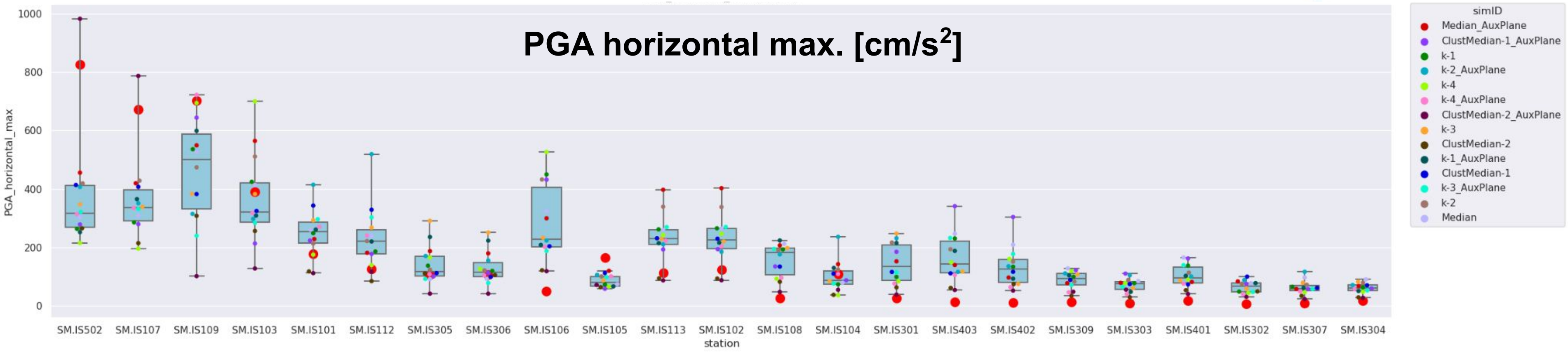
- 4,400,001 number of mesh elements
- Domain: 127km in longitude, 84km in latitude, and 25km in depth
- **Up to 5 Hz**

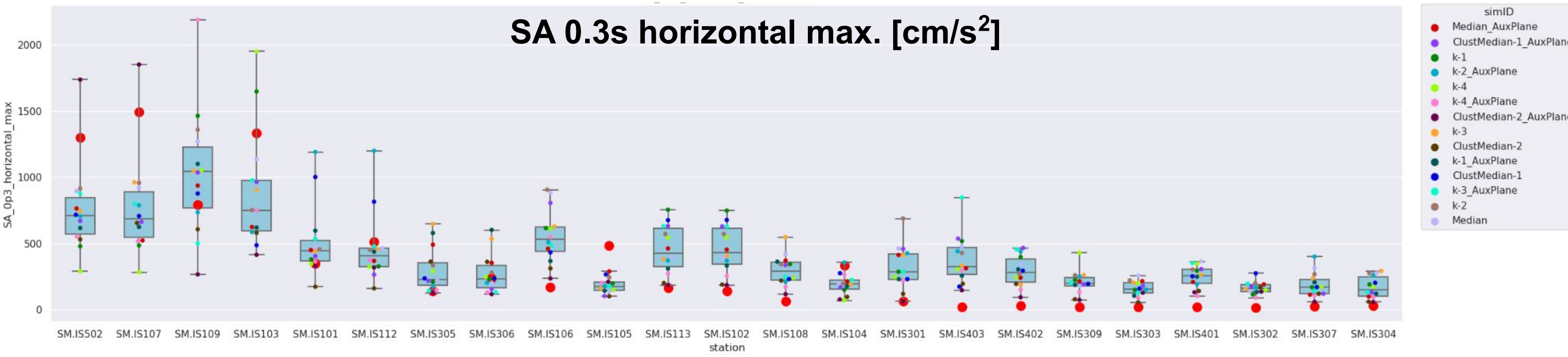
- 14 simulations
- **44m** wallclock per execution of WF
- 90 GPUs (Piz Daint) per execution





Mw 6.4, 21/06/2000





UCIS4EQ conclusions

- Successful end-to-end executions of the UCIS4EQ using PyCOMPSs workflow manager
- The results are encouraging, with synthetics reproducing the right orders of magnitude observed in the recorded data.
- When well-calibrated, our results could complement or replace GMPEs for rapid hazard assessment.

Machine Learning based Estimator for ground Shaking maps

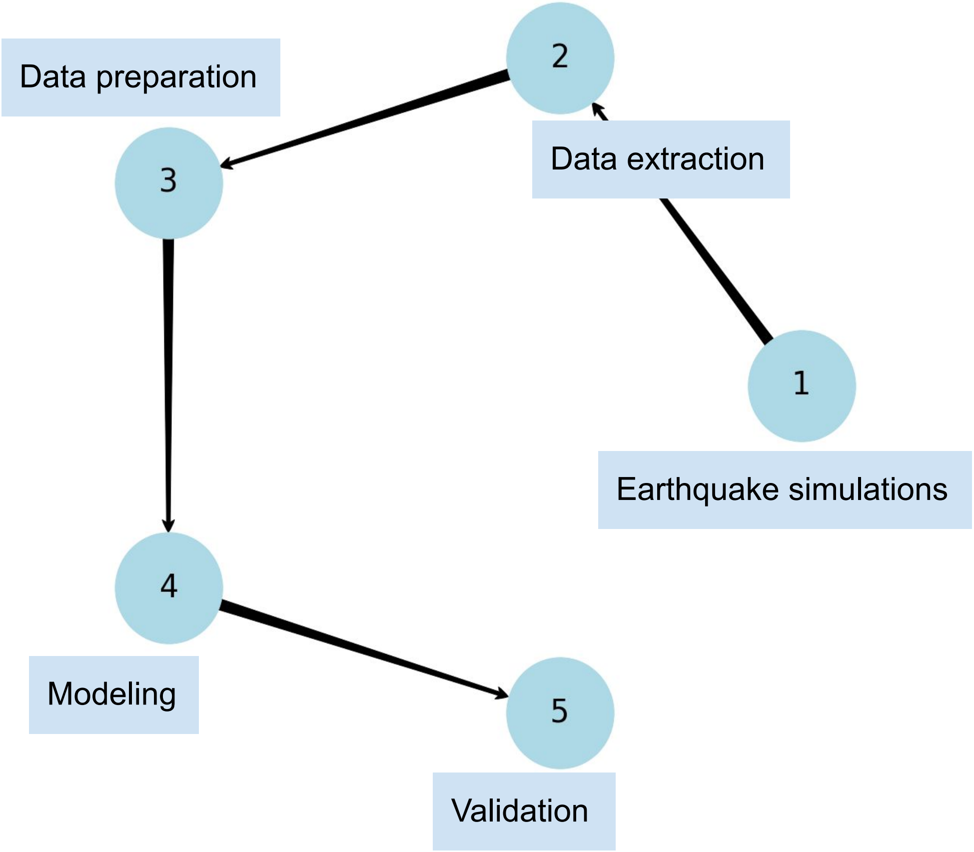
MLESmap workflow prototype

Machine Learning based Estimator for ground Shaking maps (MLESmap)

eFlows4HPC

Developing a novel methodology based on **analogous ML models** trained by a **large data set of physics-based** seismic simulations to fast-generate intensity maps in a given region **few seconds after an earthquake occurs**.

ML Methodology



features: {
"Site Lat", “Site Lon",
"Magnitude",
"EQ Lat","EQ Lon","EQ Depth",
"Distance EQ-Sta", "Azim_EQ-Sta" }

target: "PSA"

Regression

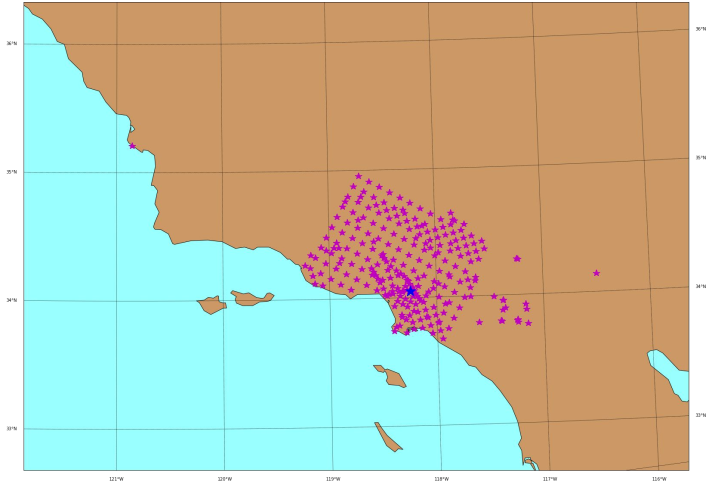
Random Forest → dislib

Neural Network → keras

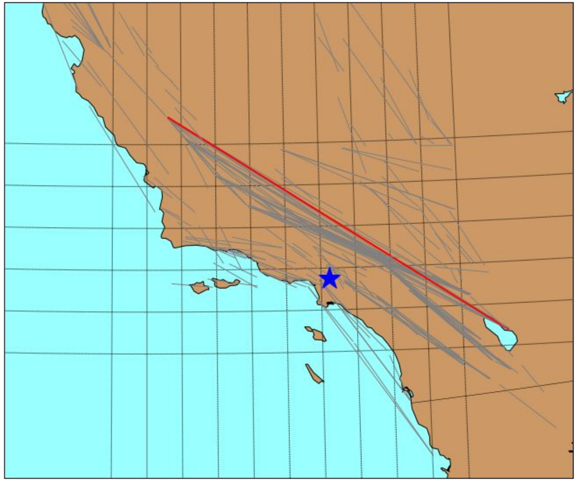
Physics-based dataset

Los Angeles basin, Southern California (EEUU)

Recording stations (sites)

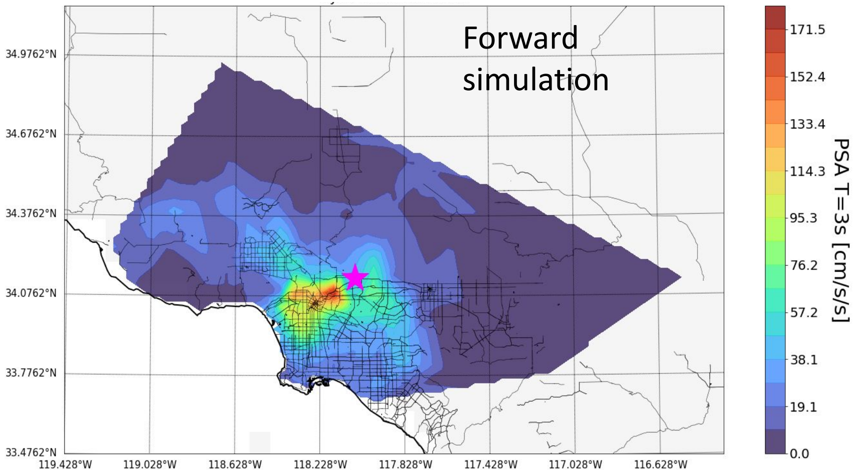


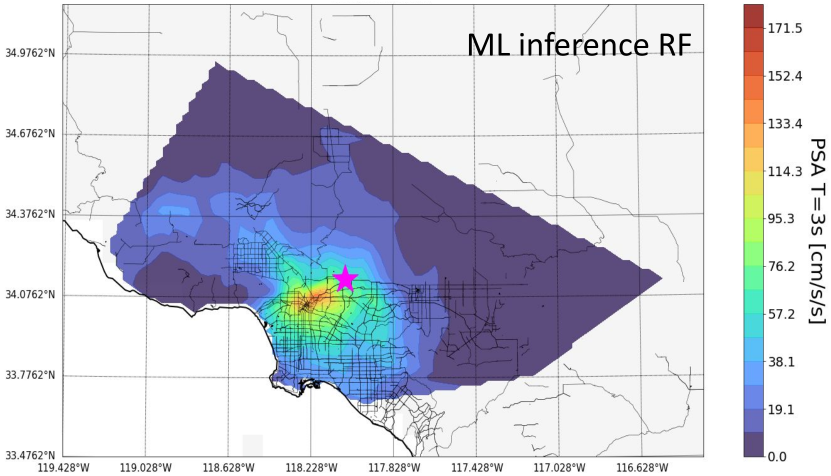
Fault systems



- CyberShake 15_4
- 253 Sites
- 225 Sources (faults or faults segments)
- 2.857.860 observations (seismic scenarios) per site
- Total of **721.687.578** events

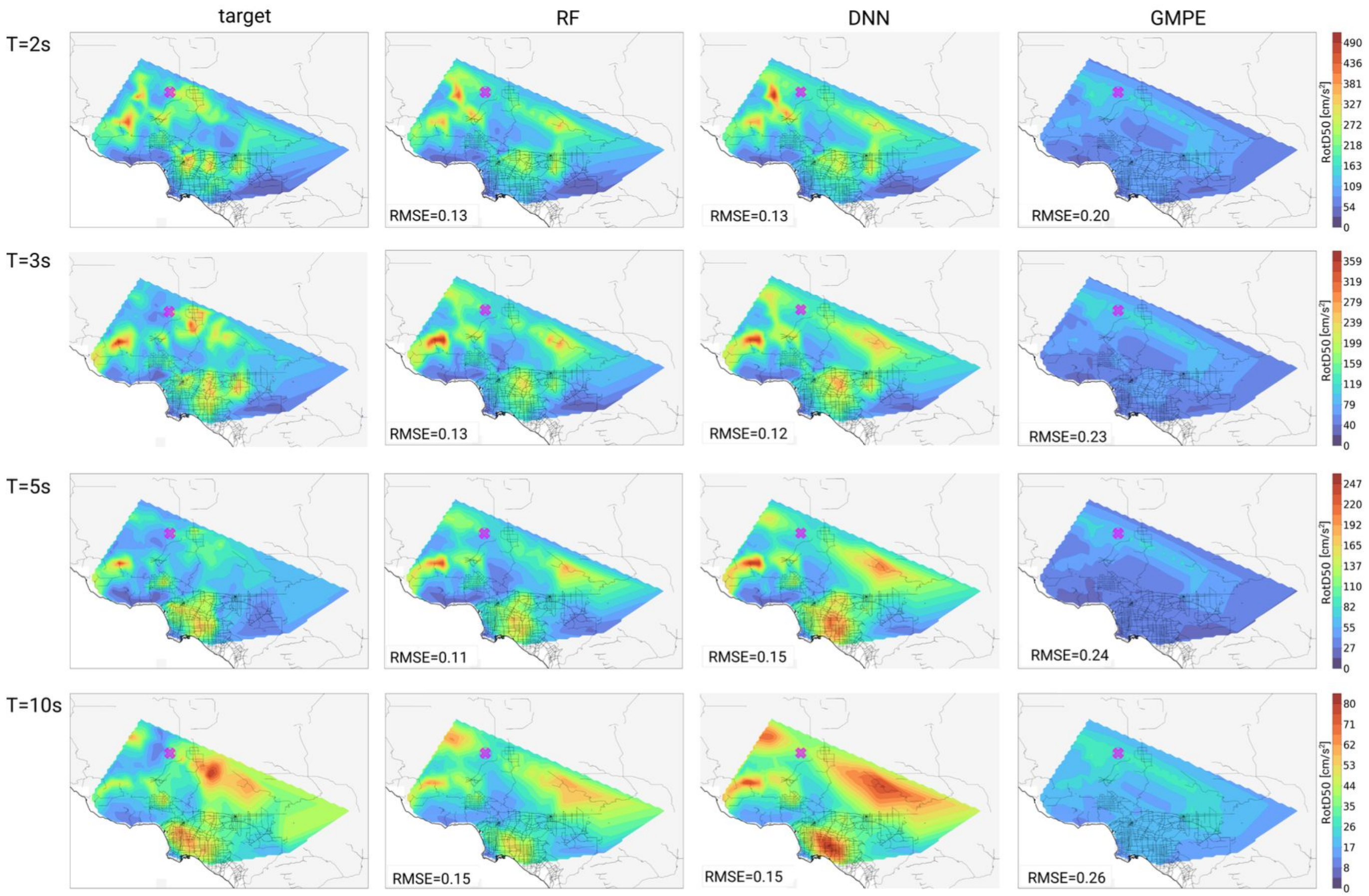
MLESmap on synthetic unseen example





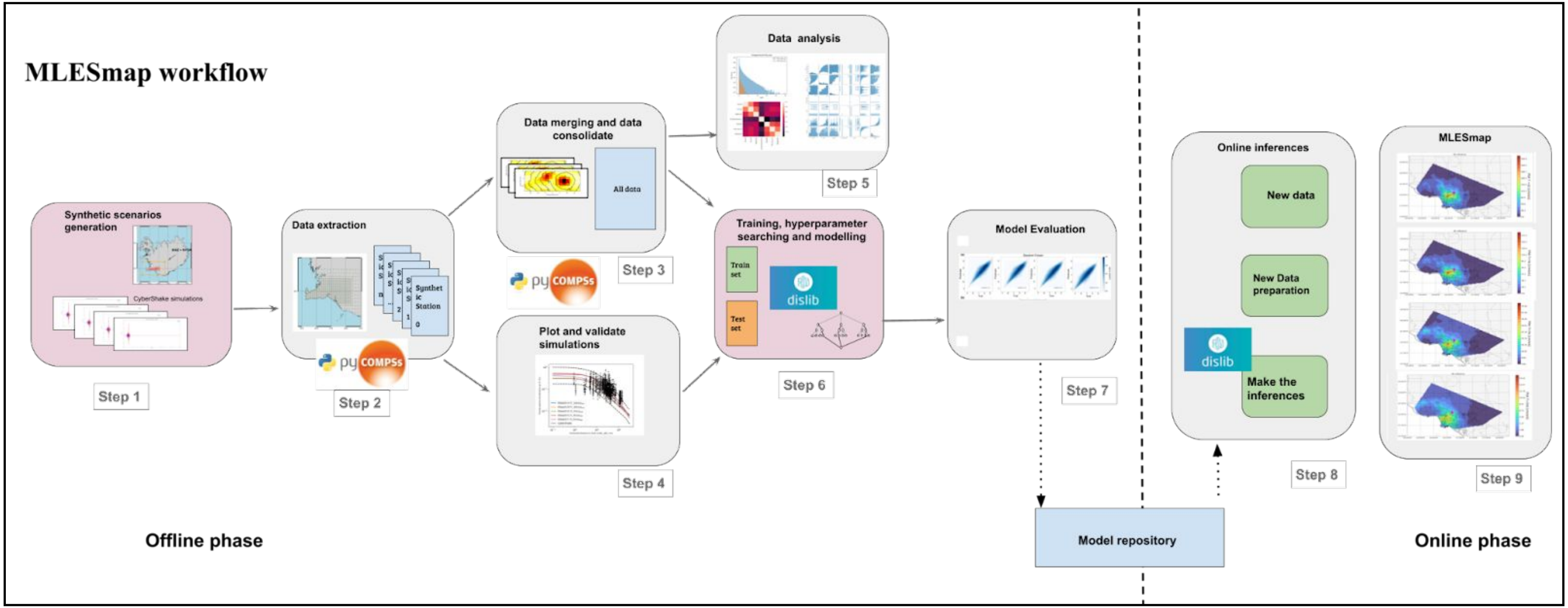
eFlows4HPC

Test results on synthetic EQ

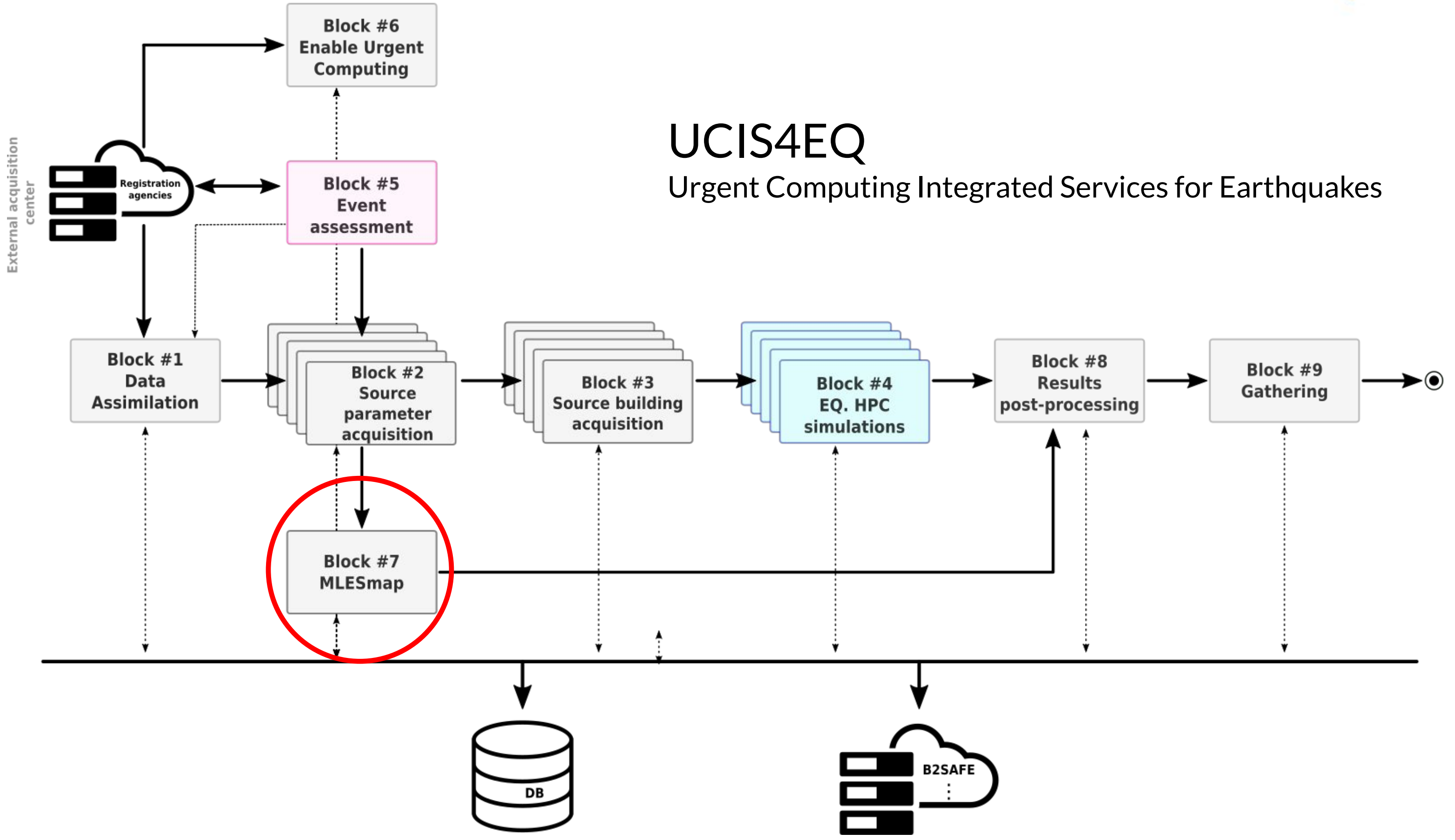


Synthetic EQ of magnitude 8.05

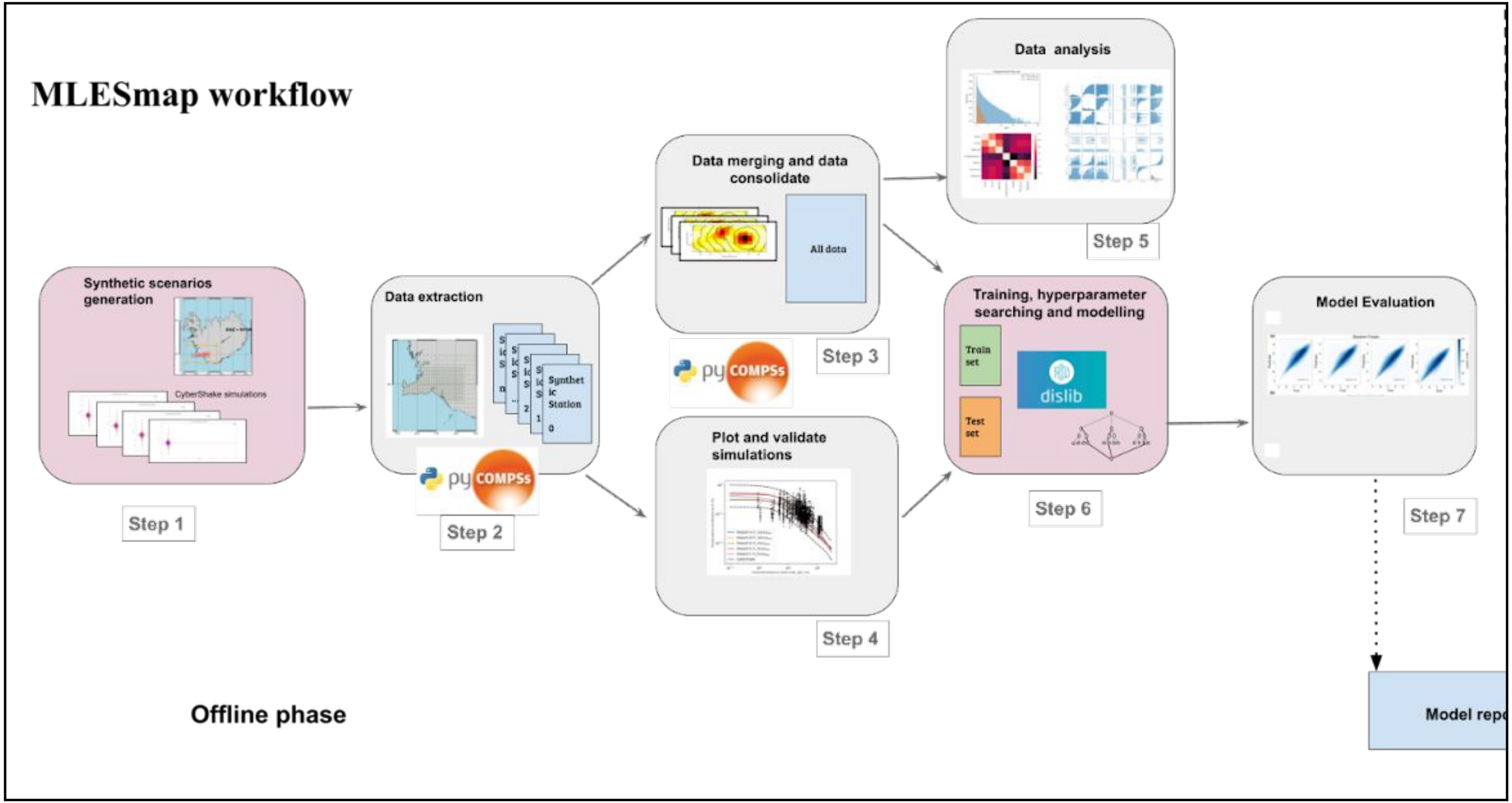
MLESmap workflow



MLESmap models integrated into UCIS4EQ



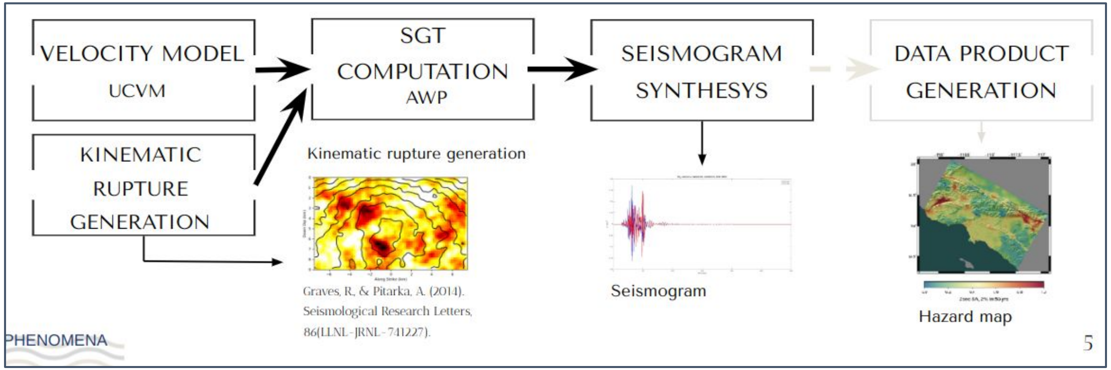
MLESmap workflow offline phase



MLESmap workflow offline phase

CyberShake WORKFLOW

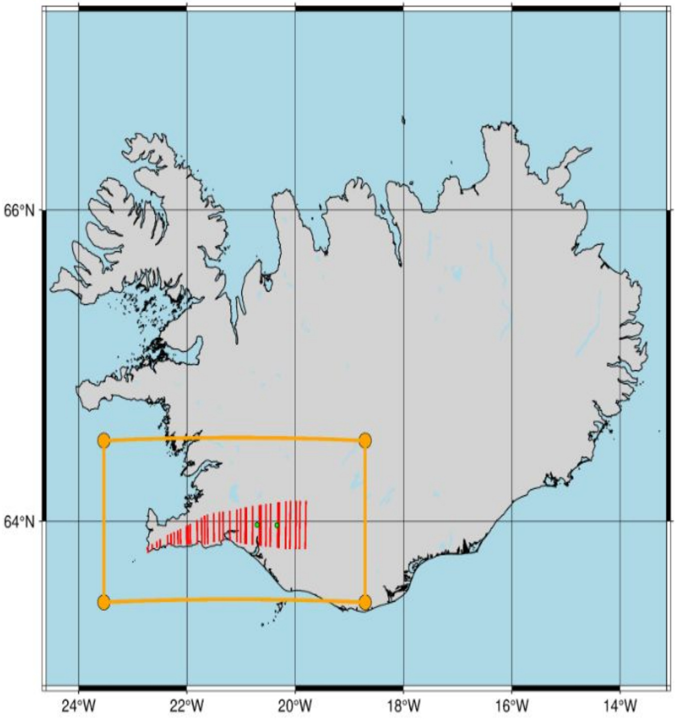
CyberShake generates the database from physics-based seismic scenarios. The number of synthetic seismograms depend on the number of stations and the number of faults to be simulated.



Computer resources per each station

Stages	CPU's	Node	Tasks	Runtime
Pre-SGT	48	1	1	1 min
pre-AWP	48	1	1	15 s
AWP_X	576	12	576	20 min
AWP_Y	576	12	576	20 min
post-X	48	1	1	10 min
post-Y	48	1	1	10 min
run_DS	576	12	288	5 min

MLESmap -- Study area



Southwest Iceland bookshelf transform zone

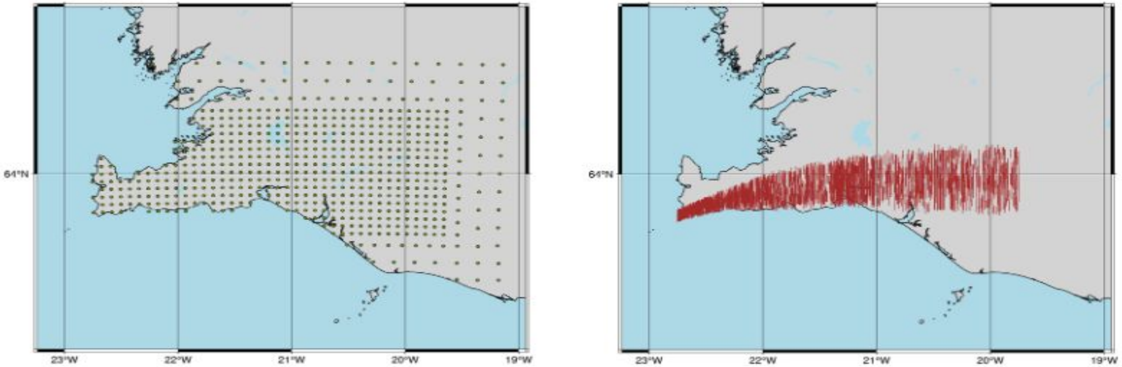
- **Iceland is the most seismically active region in northern Europe**, due to its location on the Mid-Atlantic Ridge, which along with the Icelandic hot spot, is responsible for the tectonics and its active seismicity and volcanism

- **The largest earthquakes** in Iceland occur within the two transform fault zones in the country, the South Iceland Seismic Zone (SISZ) and Reykjanes Peninsula Oblique Rift (RPOR)

- The SISZ is characterized by the bookshelf faulting model containing seismogenic **strike-slip N-S striking faults**

MLESmap data generation

CyberShake generates the database from physics-based seismic scenarios. The number of synthetic seismograms depend on the number of stations and the number of faults to be simulated.



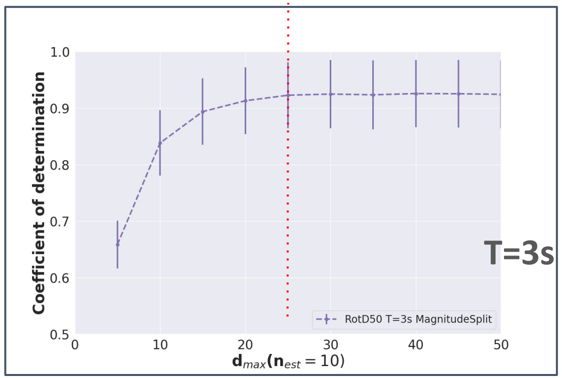
Location map of the synthetic seismic stations and the location of the faults 593 Synthetic Stations and 16633 events

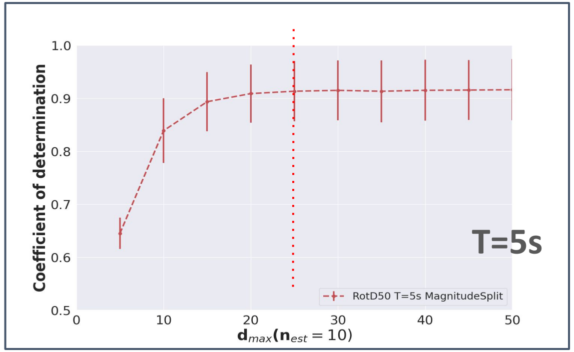
Computer resources per each station

Stages	CPU's	Node	Tasks	Runtime
Pre-SGT	48	1	1	1 min
pre-AWP	48	1	1	15 s
AWP_X	576	12	576	20 min
AWP_Y	576	12	576	20 min
post-X	48	1	1	10 min
post-Y	48	1	1	10 min
run_DS	576	12	288	5 min

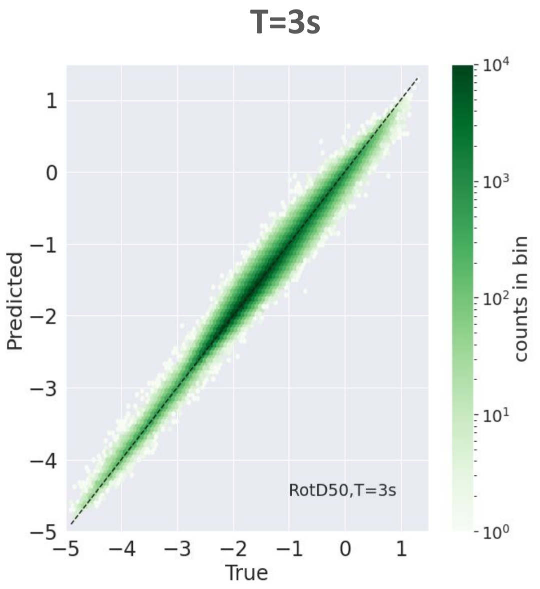
Preliminary results on SISZ region

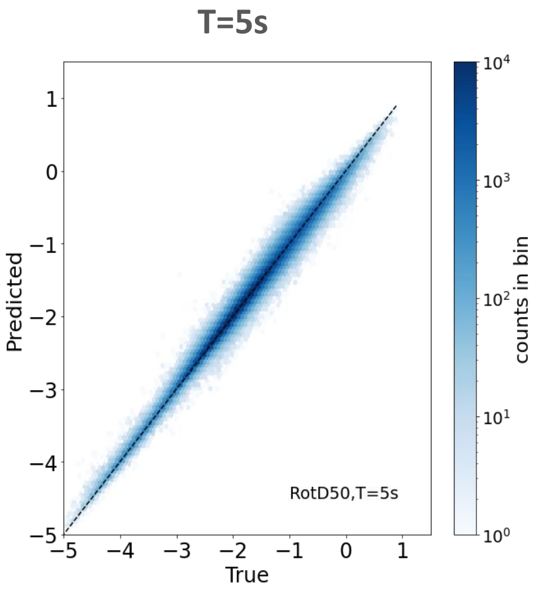
Hyperparameters RF dislib





Results on validation set





MLESmap conclusions

eFlows4HPC

- MLESmap: towards the combination of physics-based data and ML engine to fast estimate the ground shaking intensity using EQ information available shortly after the event

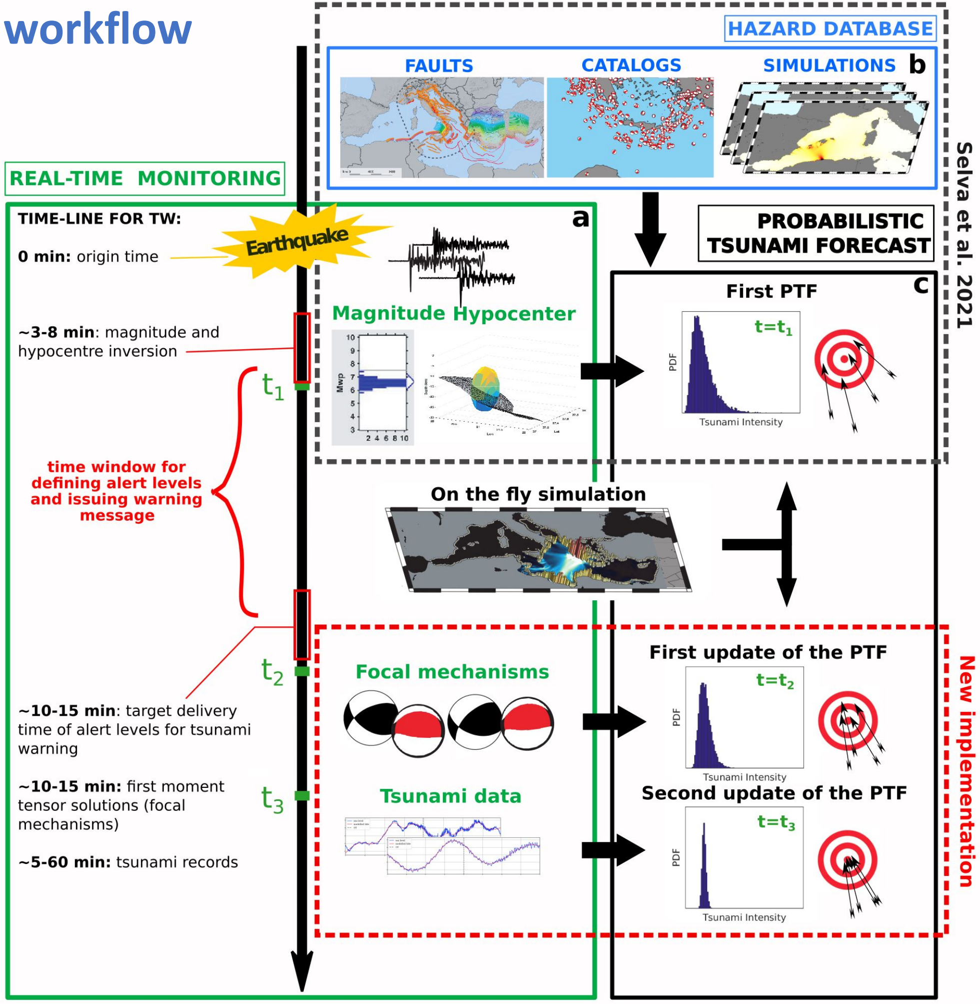
Tsunami workflow

PTF (Probabilistic Tsunami Forecast)

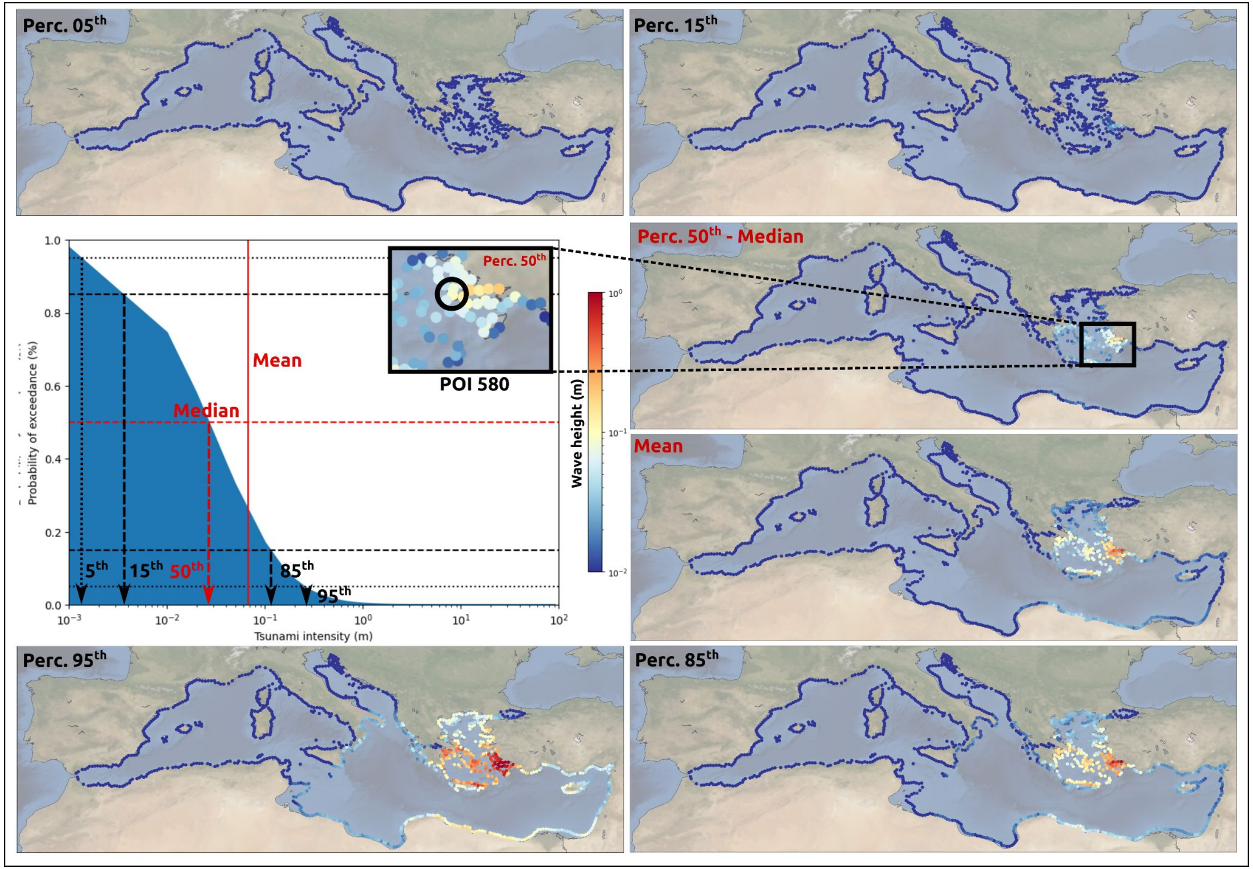
Content

- introduction and motivation
- workflow description: technical and scientific improvements
- further developments

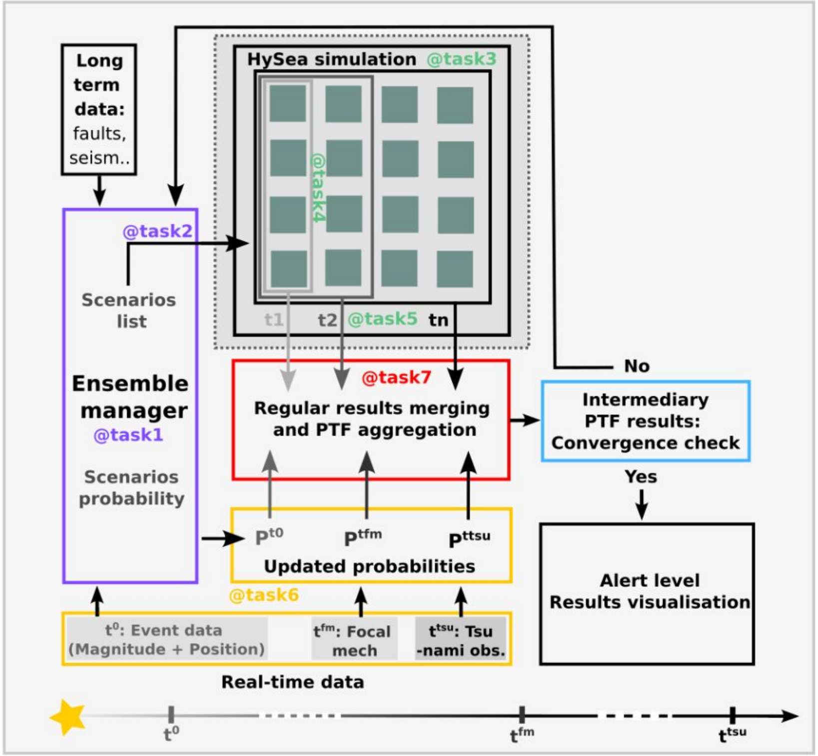
Introduction of the PTF workflow

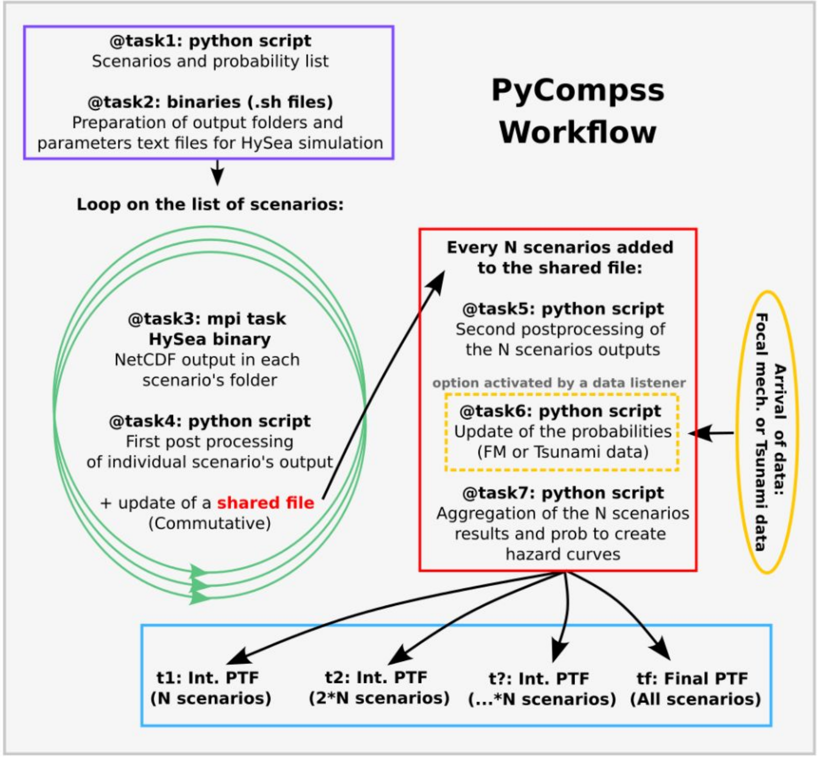


Introduction of the PTF workflow

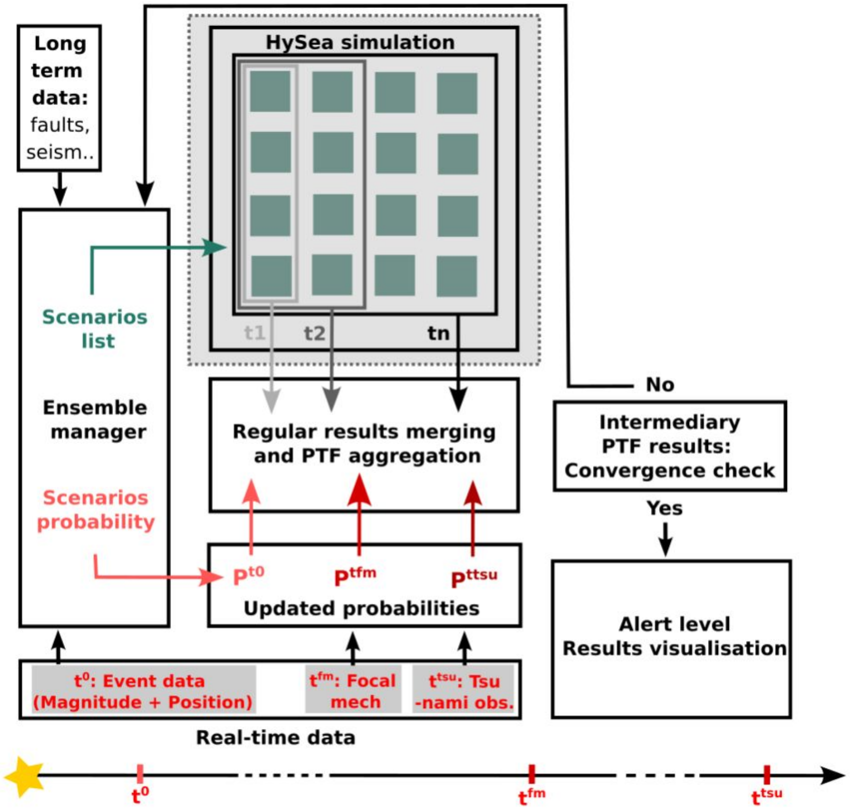


First end-to-end version of PTF orchestrated with PyCOMPSs

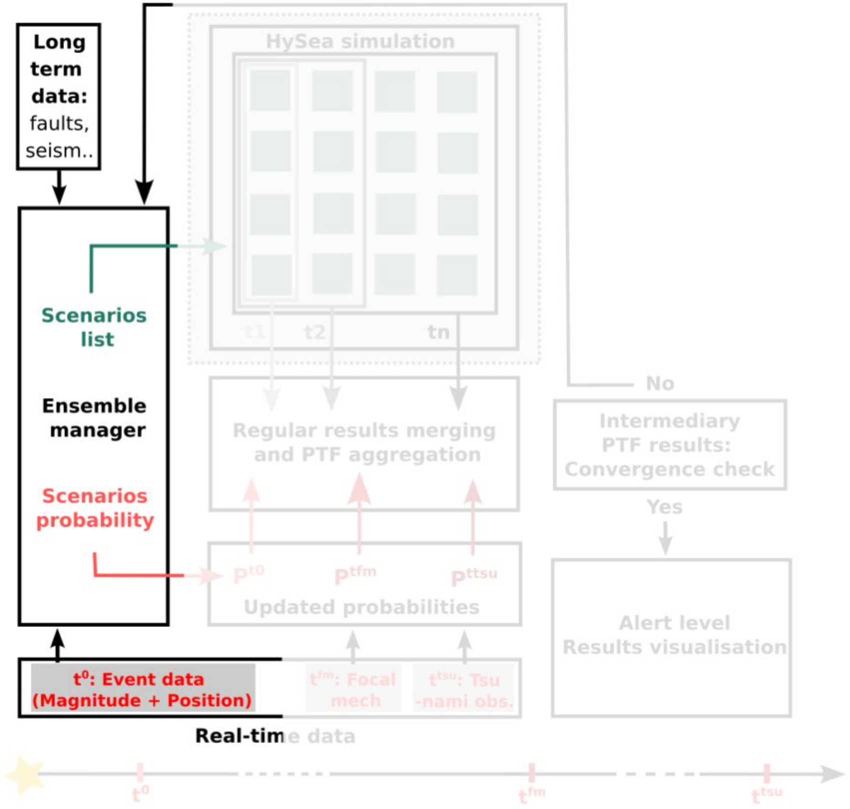


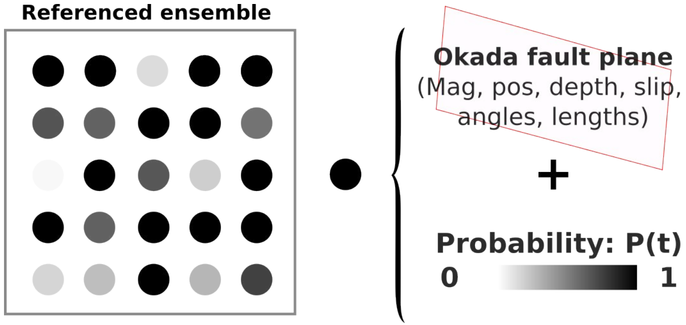


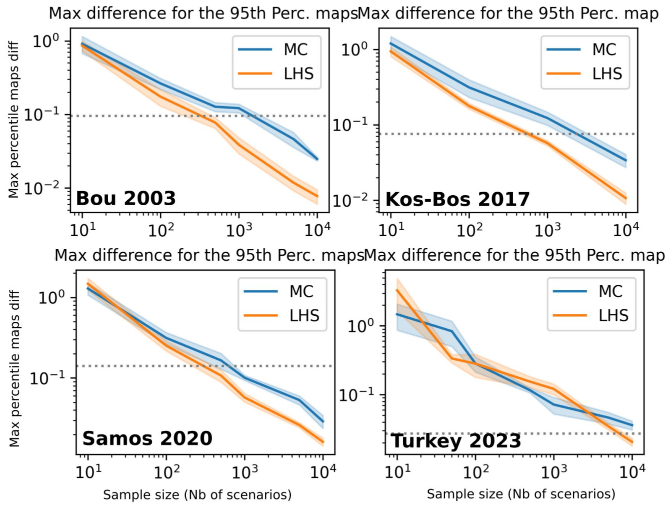
PyCOMPSs workflow



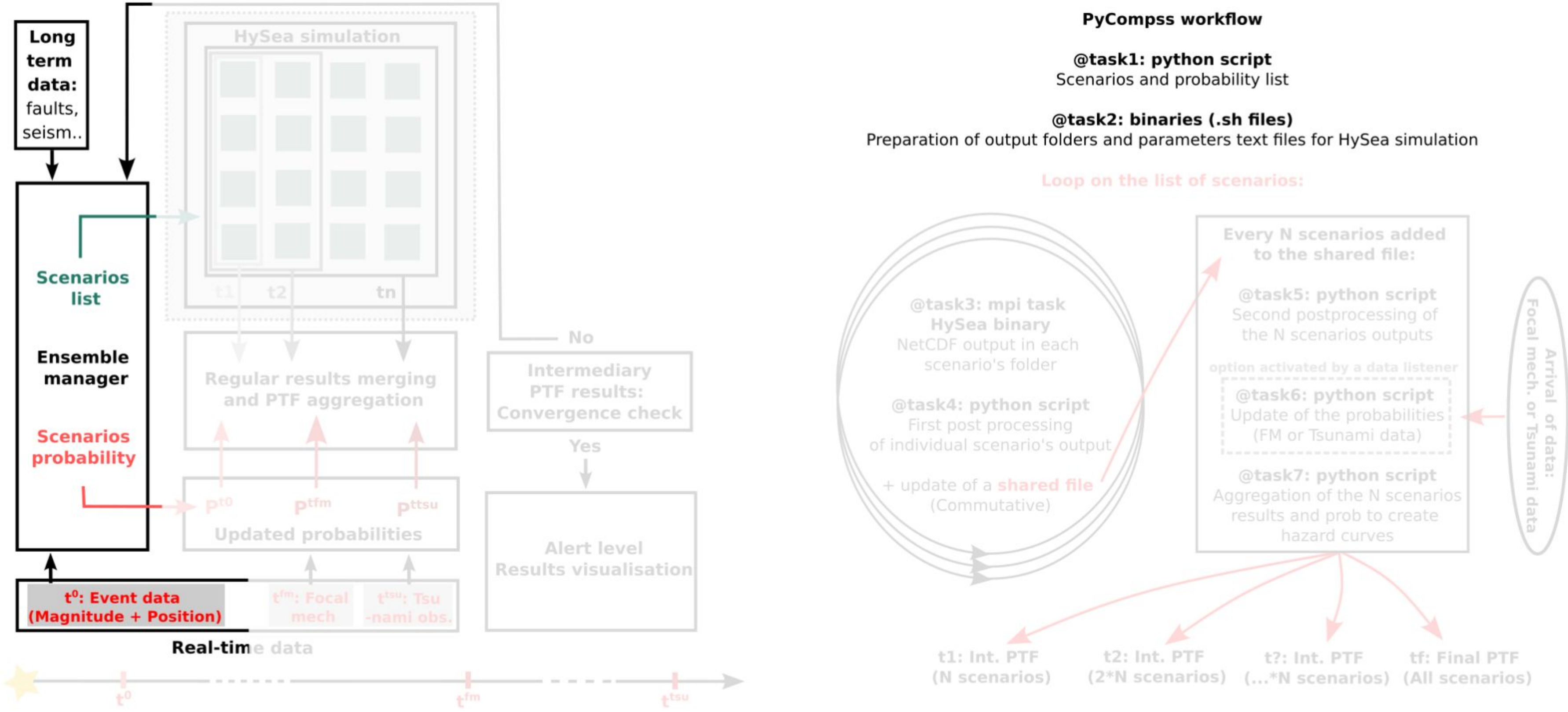
PyCOMPSs workflow: Step1, ensemble manager



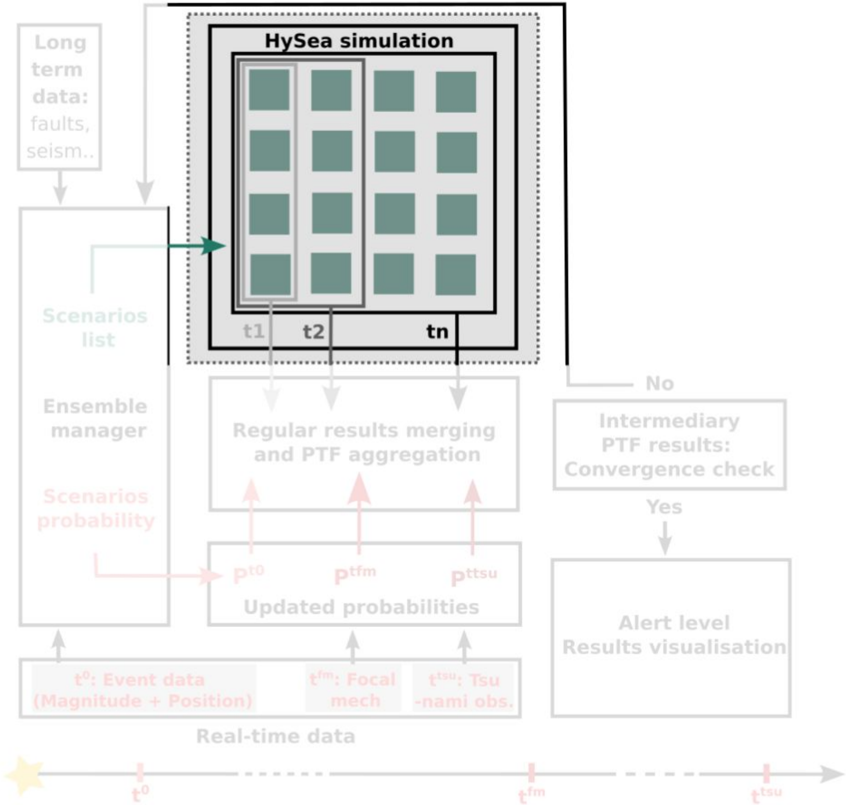




PyCOMPSs workflow: Step1, ensemble manager

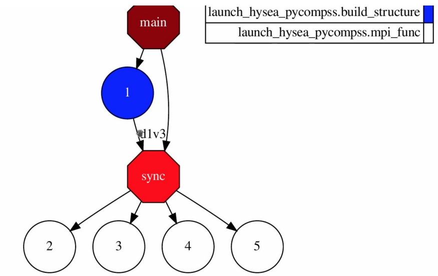


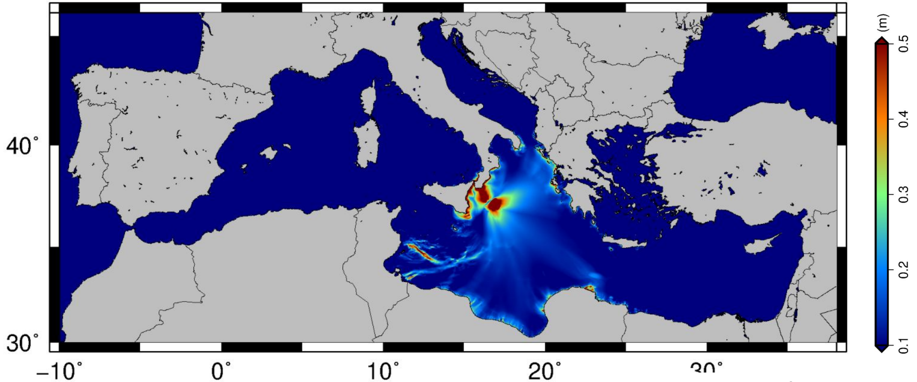
PyCOMPSs workflow: Step2, HySea simulations



Integration of HySEA in the workflow and mpi-mc parallelisation

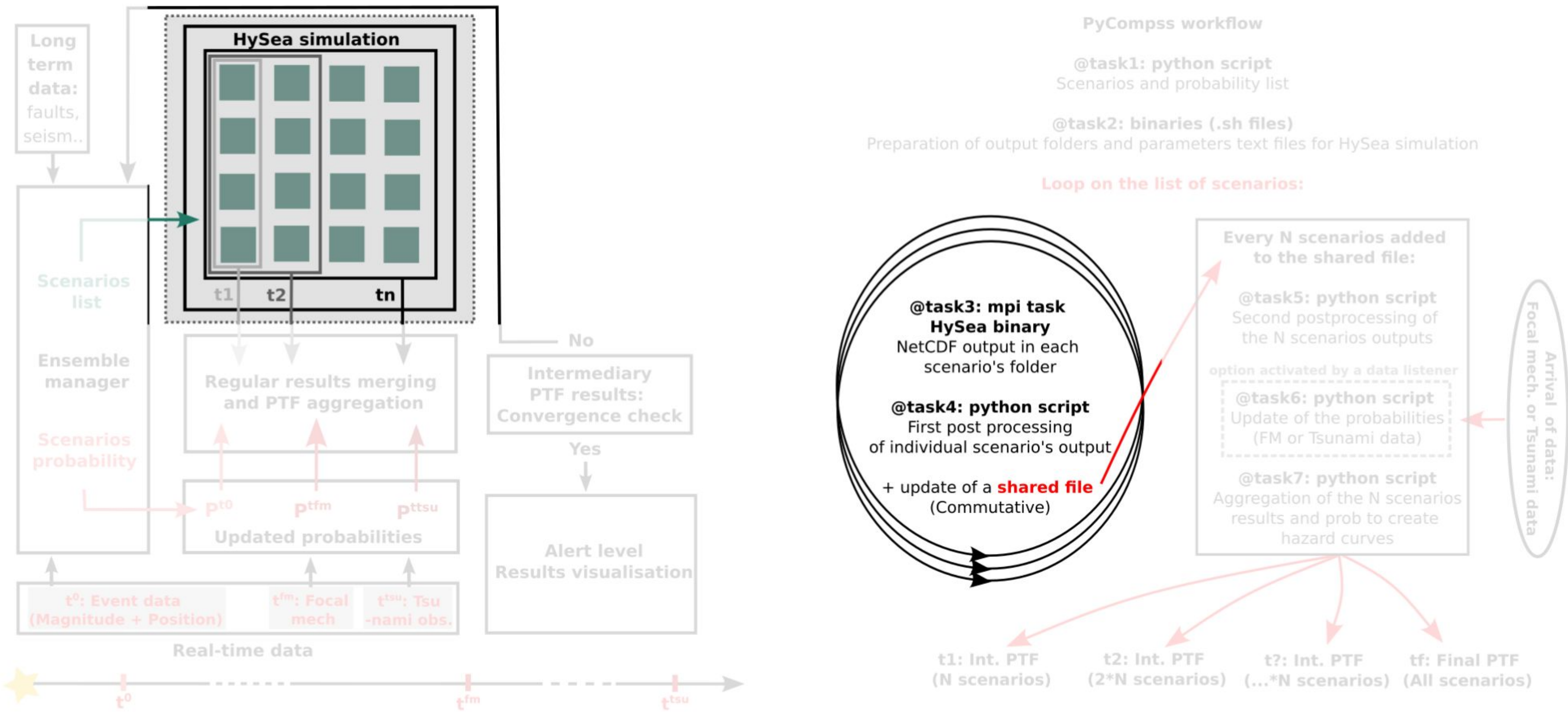
PyCompss calls the HySEA binary task in several nodes in parallel. Each task runs a single job to the queue system that implicitly carries out the parallel execution of a predefined number of simulations dividing into internal jobs, and allowing the traceability of each processes involved to be observed.



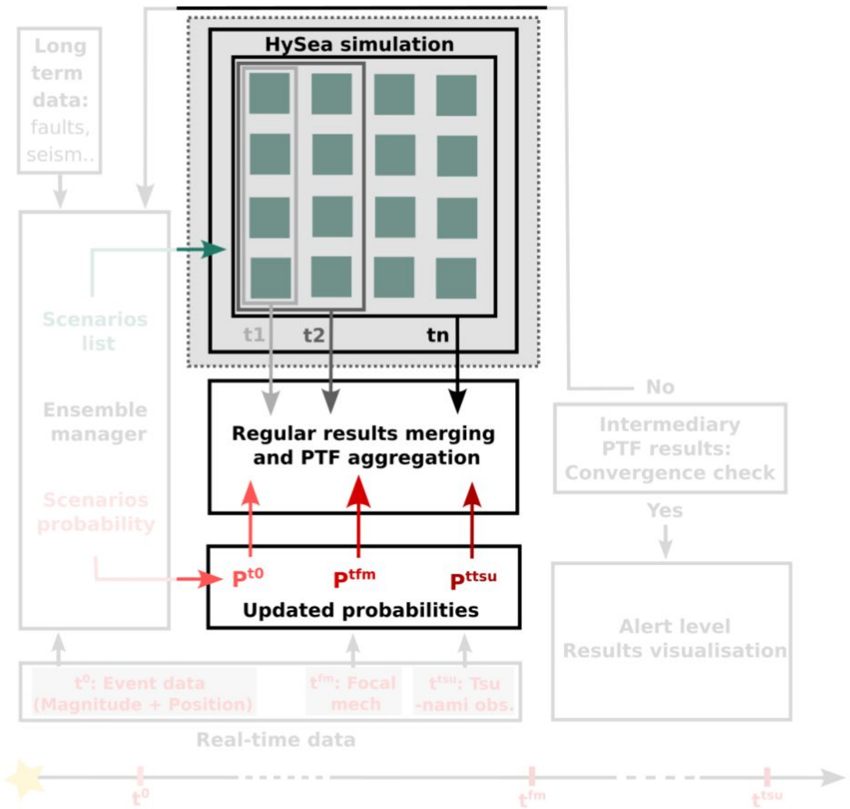


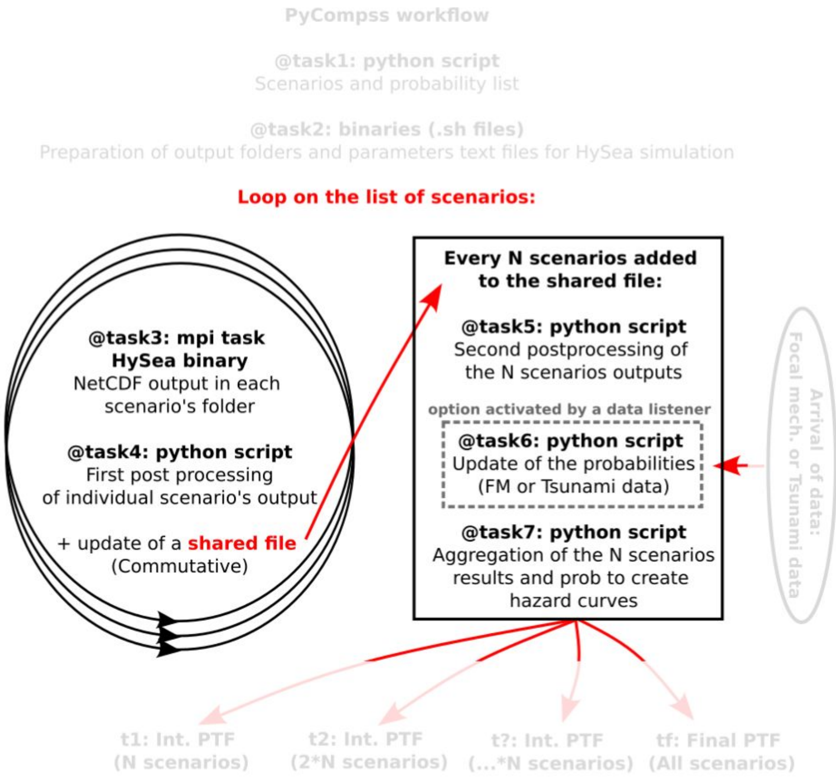
Lorito et. al, 2021

PyCOMPSs workflow: Step2, HySea simulations

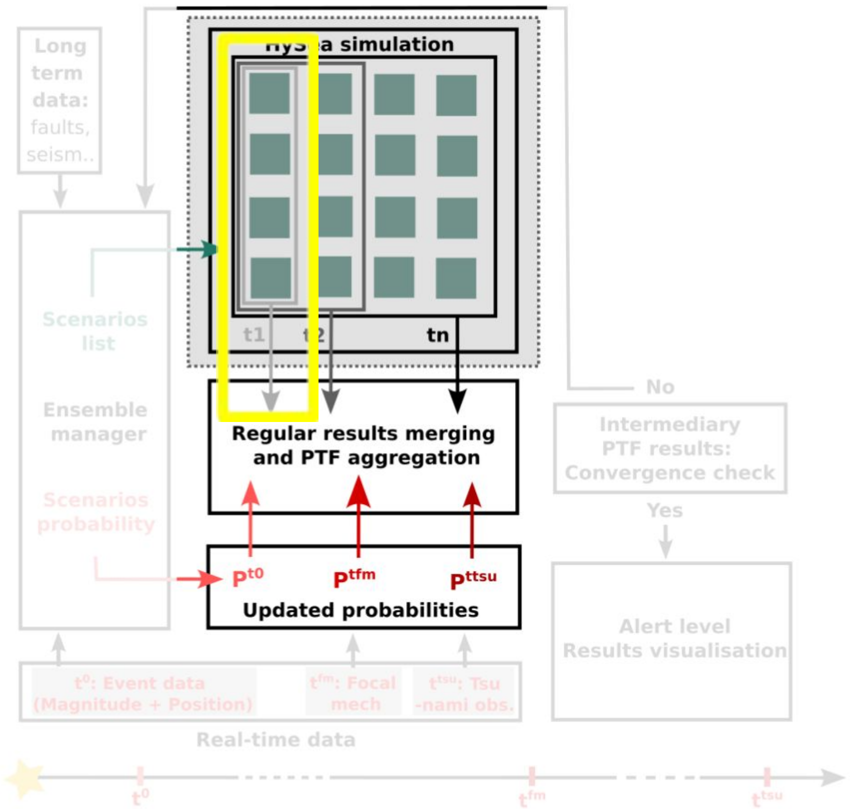


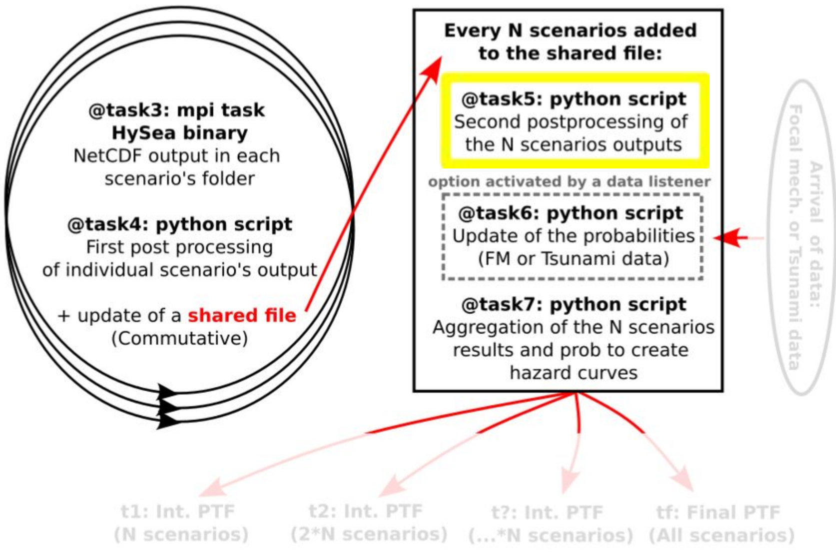
PyCOMPSs workflow: Step3, post-processing of the output



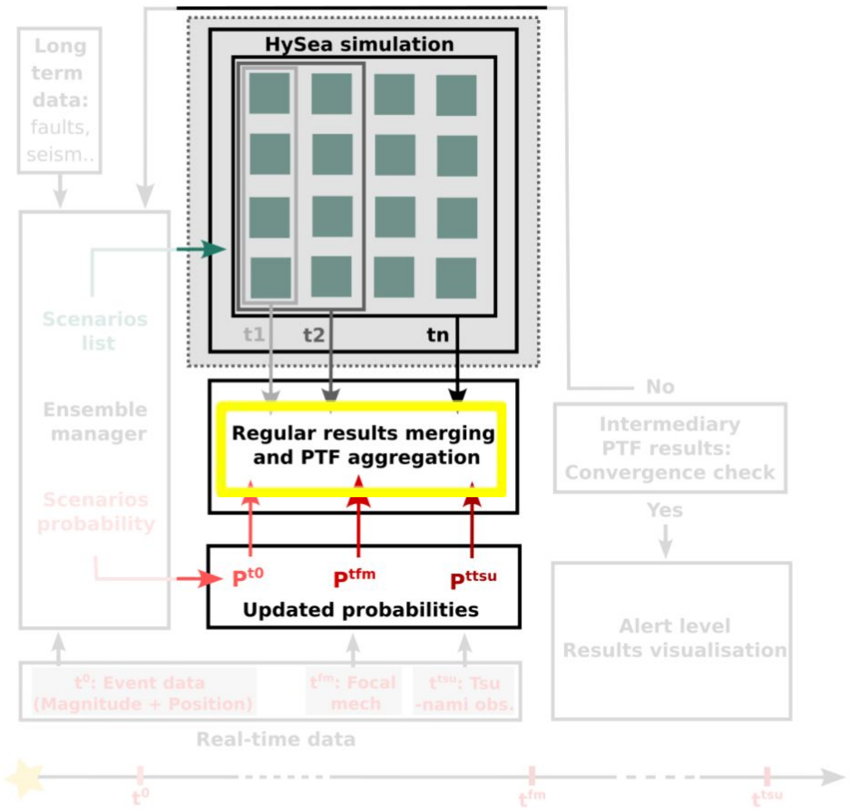


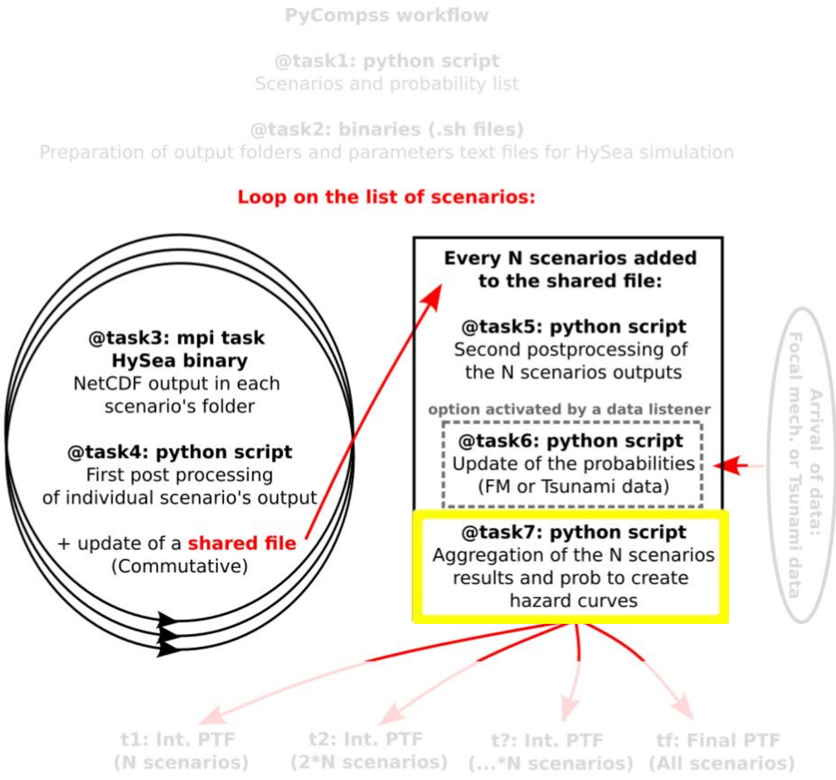
PyCOMPSs workflow: Step3, post-processing of the output





PyCOMPSs workflow: Step3, post-processing of the output

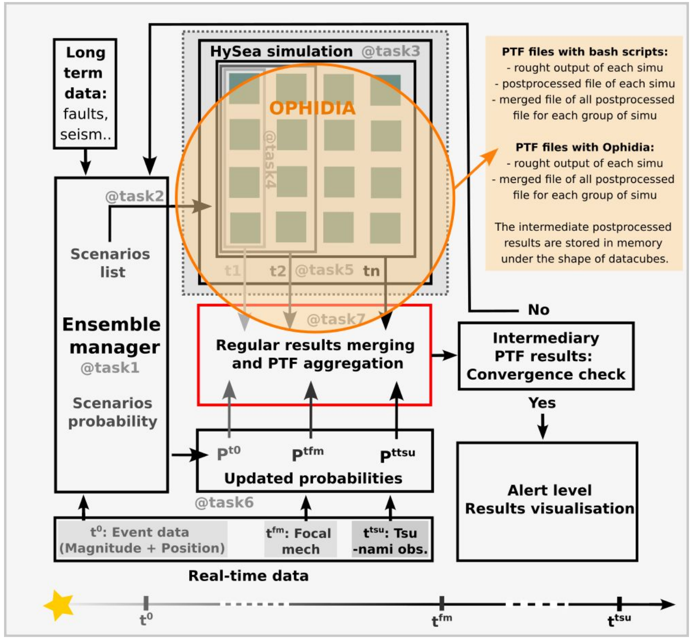


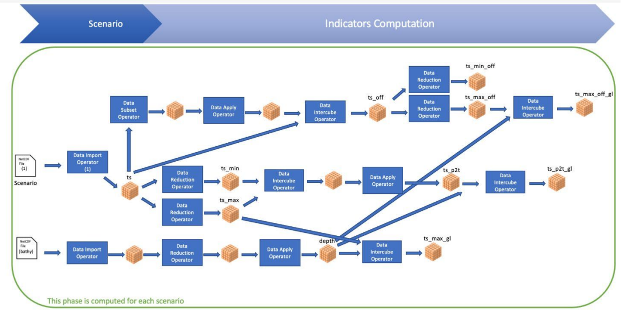


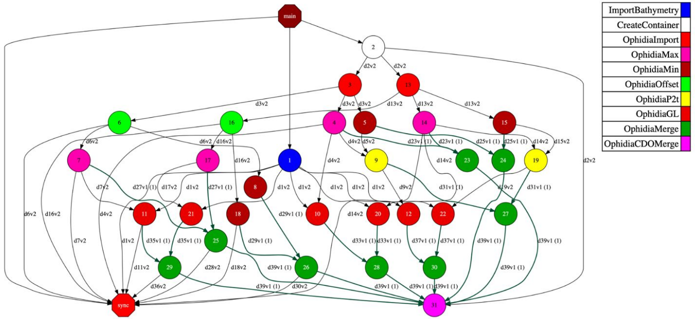
PyCOMPSs workflow: Step3, post-processing of the output

Post-Processing in 2 steps with python scripts or using Ophidia

- STEP3 is implemented with **2 python scripts**, one running after each simulation, and one running when all simulations are completed. The use of the **Ophidia** framework avoid generating required continuous I/O operations from disk to save and then retrieve the outputs for the final merging phase.

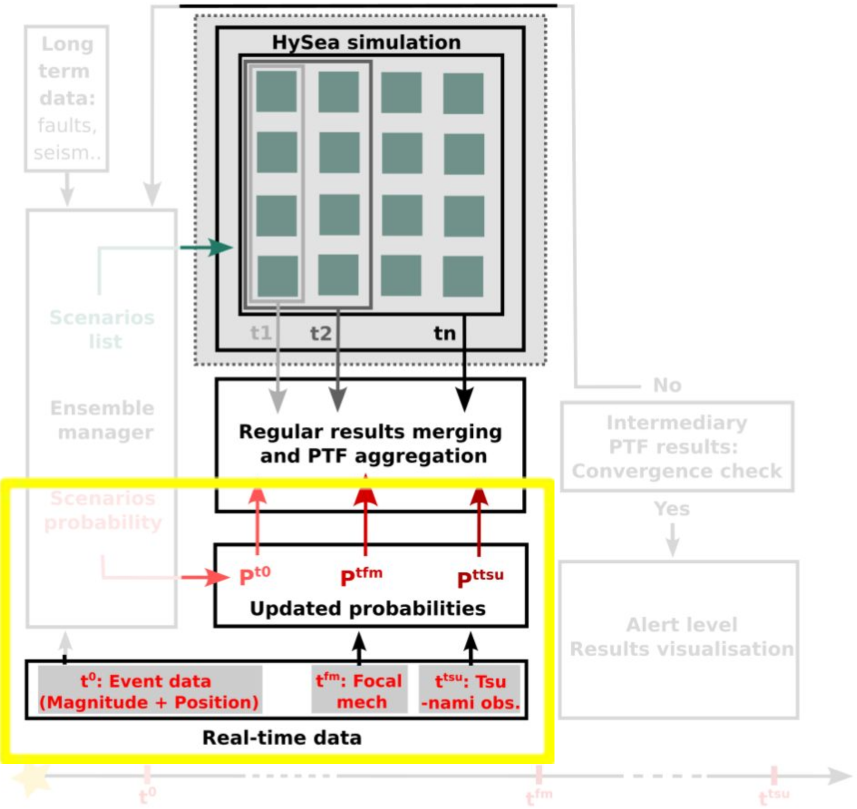


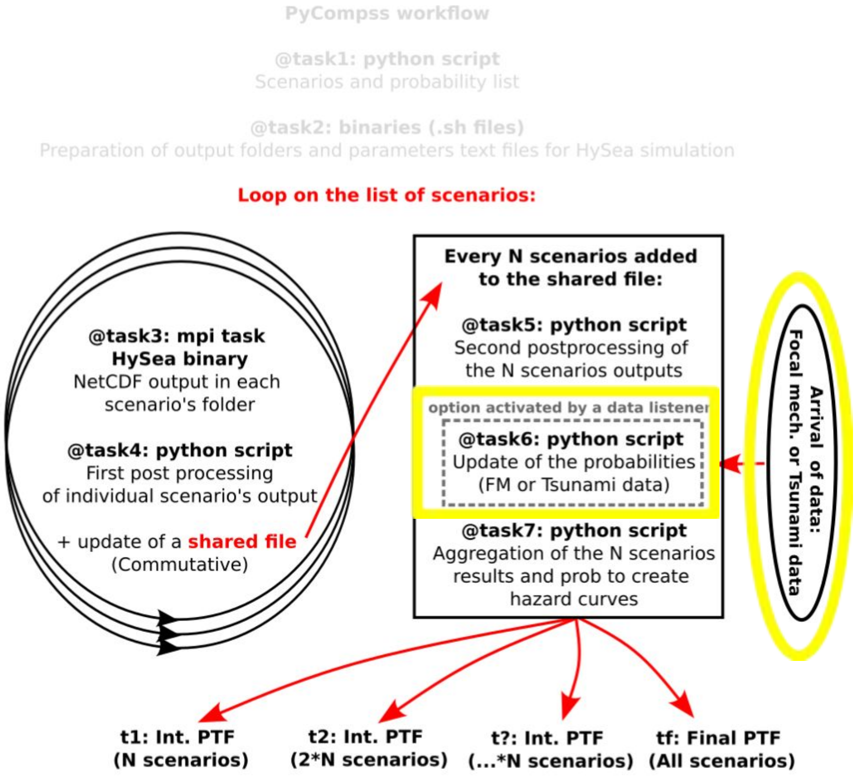




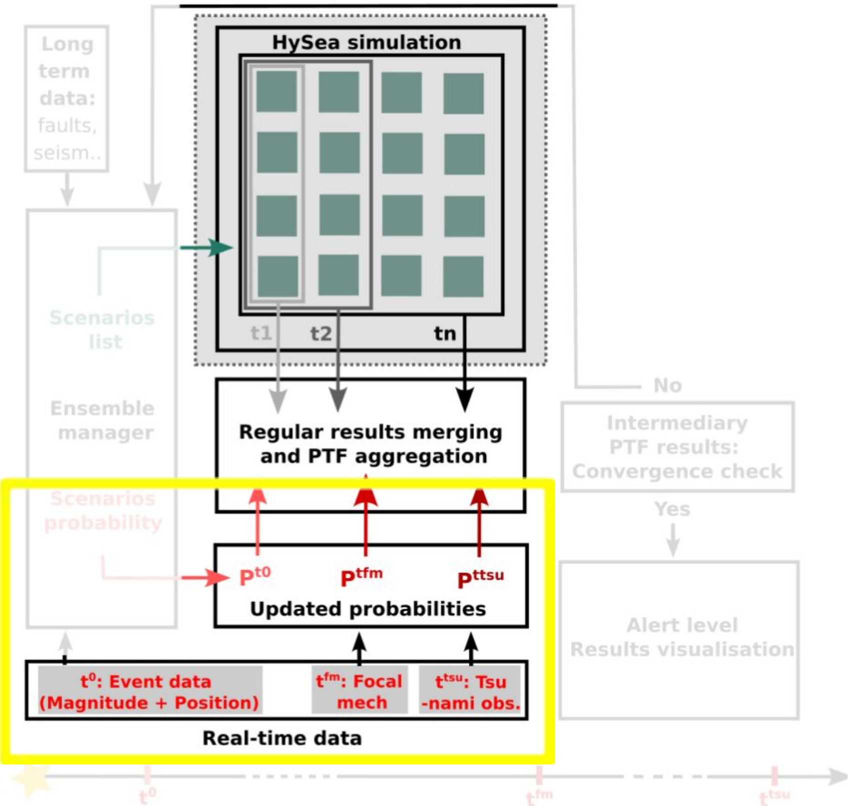
This figure shows the PyCOMPSs tasks graph generated at the end of the workflow.

PyCOMPSs workflow: Data-driven update of the PTF





PyCOMPSs workflow: Data-driven update of the PTF

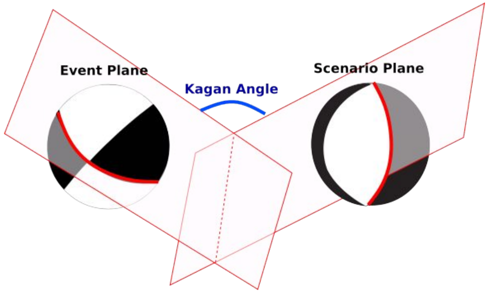


PyCOMPSs workflow: Data-driven update of the PTF

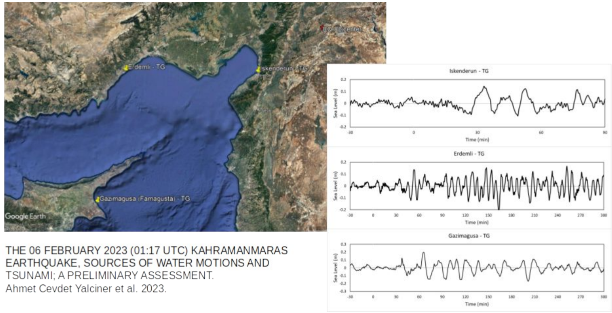
Use of new incoming information to update the scenarios' probabilities

- After the post-processing step and before the aggregation step, two tasks can be optionally (listener/parameter?) activated and allow a re-weighting of the probabilities based on information on the earthquake or the tsunami
- One task takes into input data on the earthquake focal-mechanism and the second one takes into input the tsunami observations (tide-gage records)

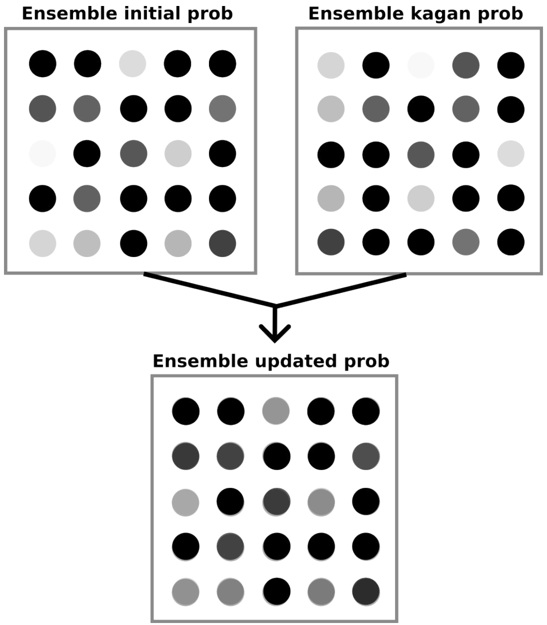
Focal mechanism (Kagan angle)



Tsunami data

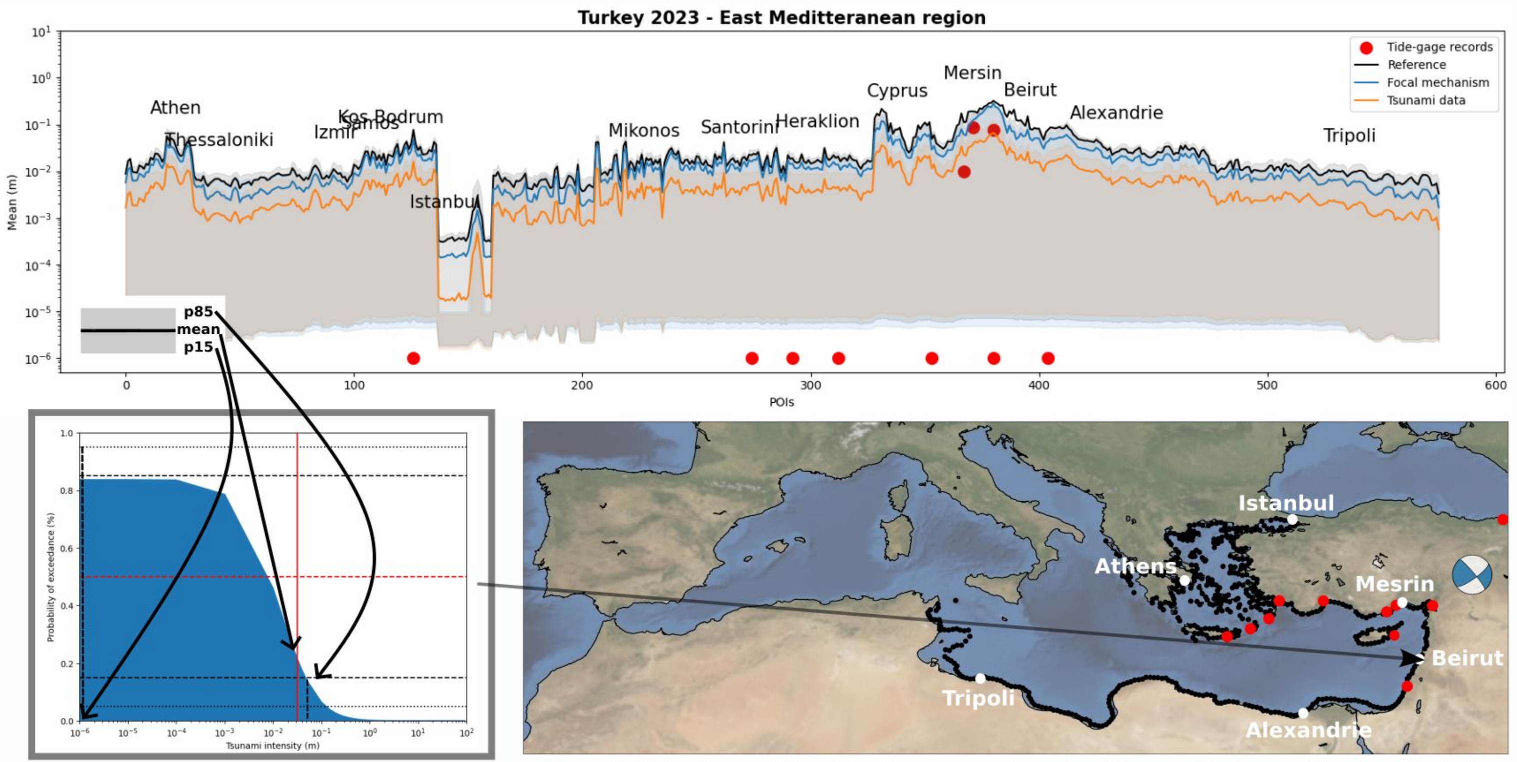


THE 06 FEBRUARY 2023 (01:17 UTC) KAHRAMANMARAS EARTHQUAKE, SOURCES OF WATER MOTIONS AND TSUNAMI; A PRELIMINARY ASSESSMENT.
Ahmet Cevdet Yalciner et al. 2023.



PyCOMPSs workflow: Data-driven update of the PTF

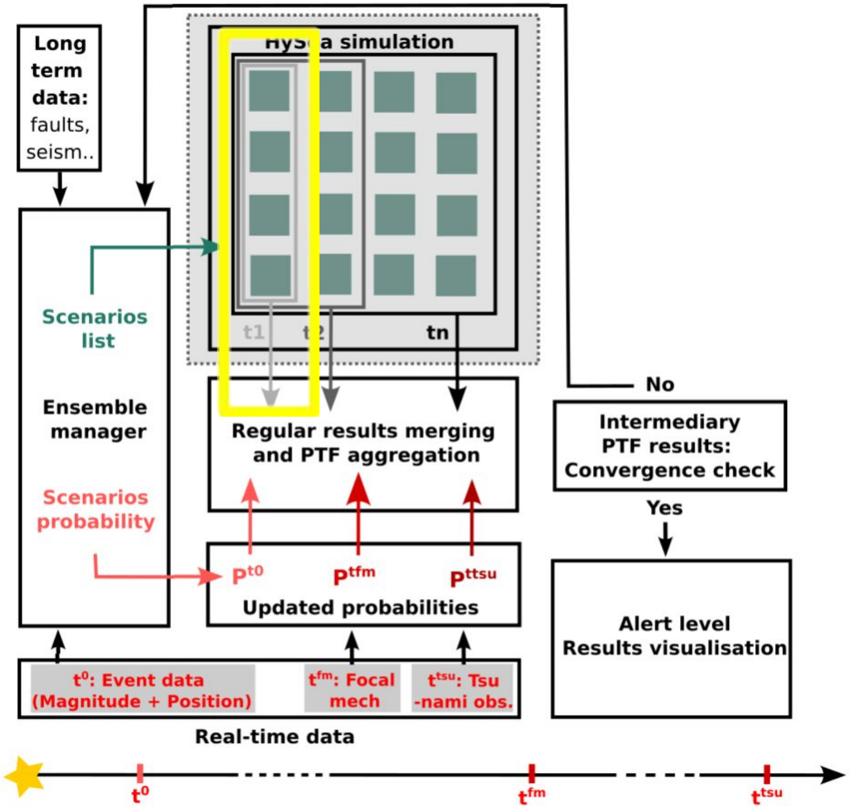
Use of new incoming information to update the scenarios' probabilities

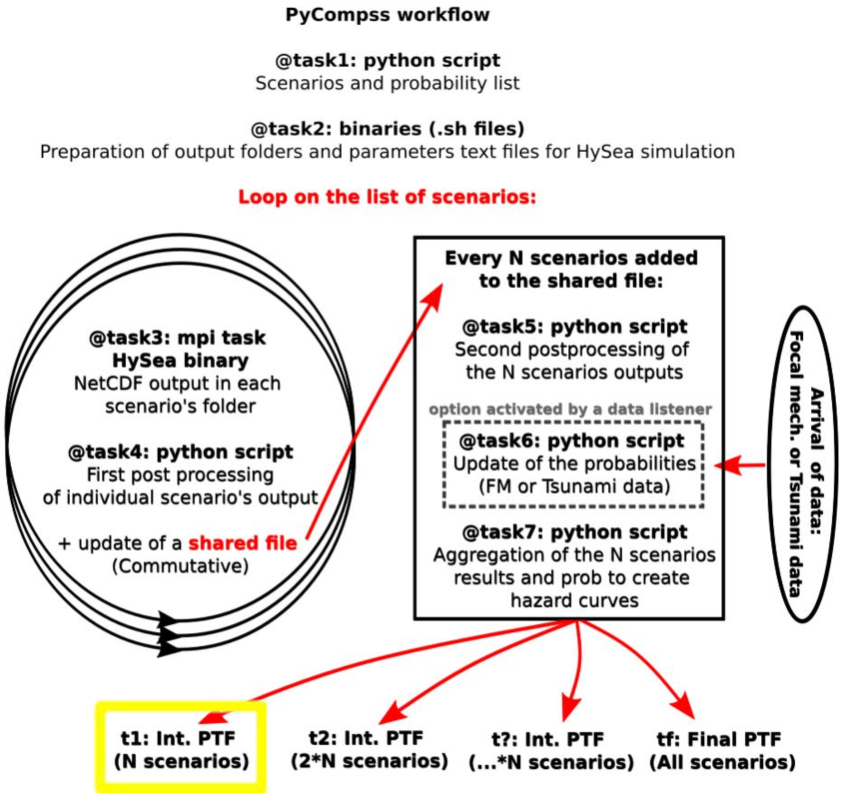


Beirut's POI Hazard Curve

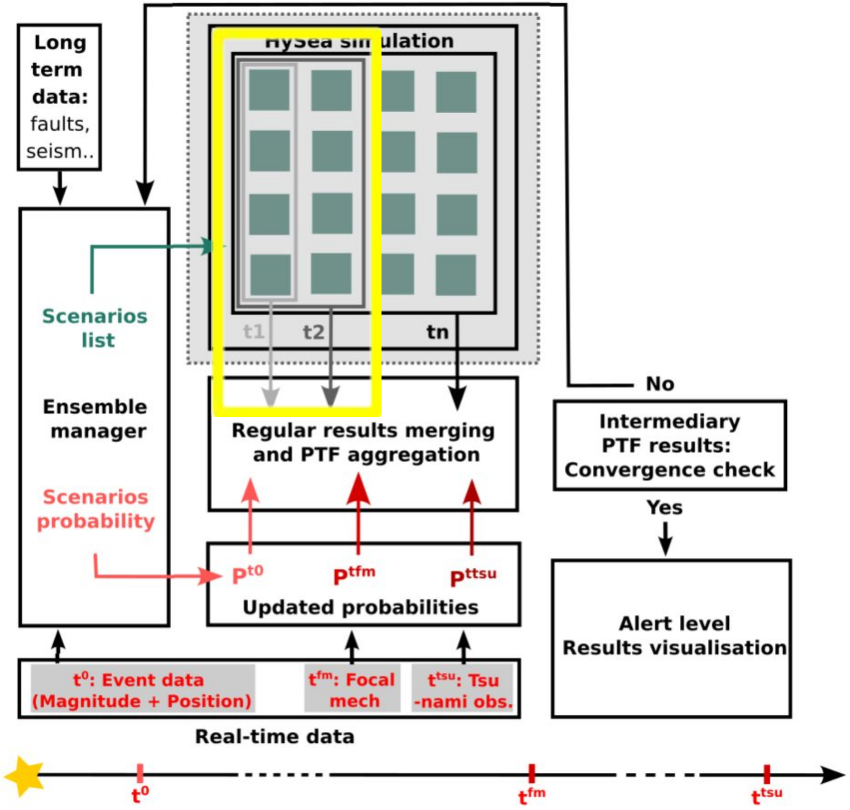
Map of all locations of the tide-Gage records, main cities and POIs used for the HC plot

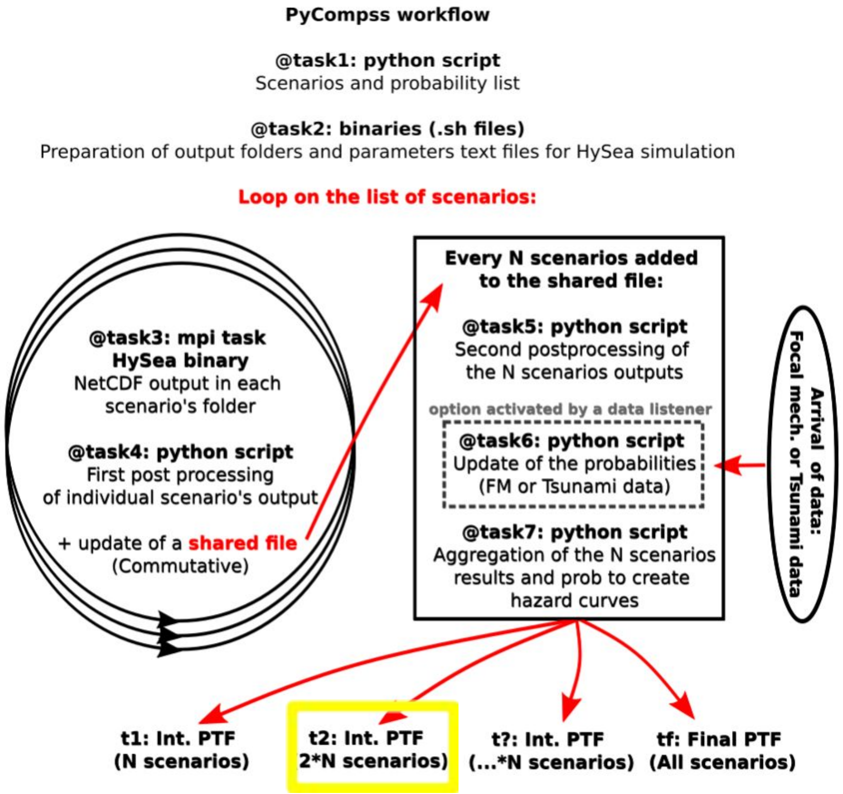
PyCOMPSs workflow: Step3, intermediate evaluation of the PTF



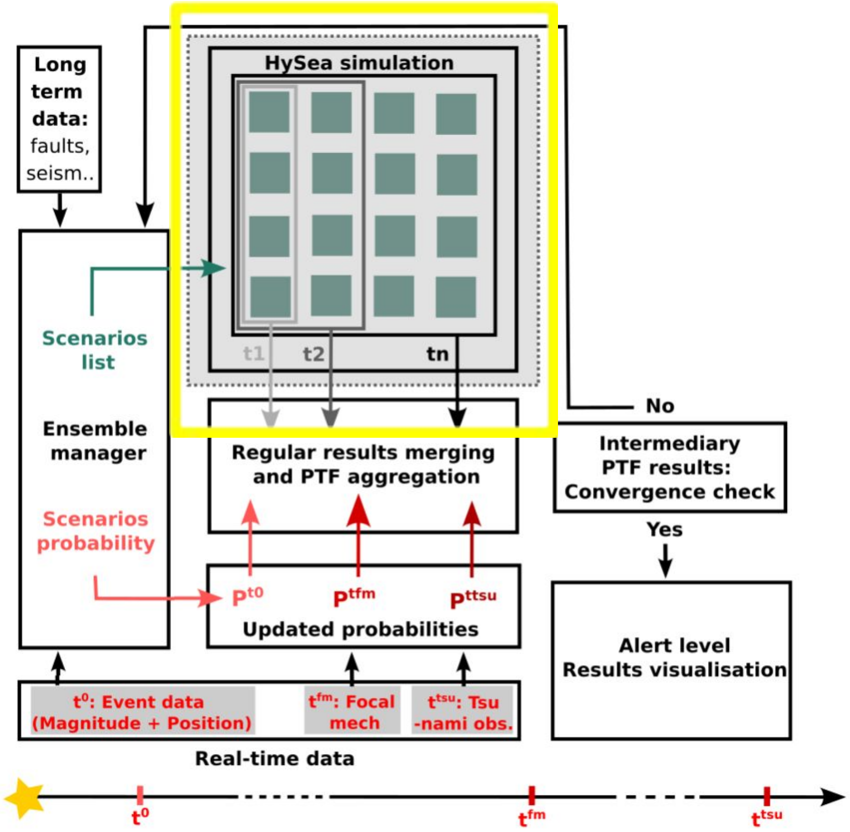


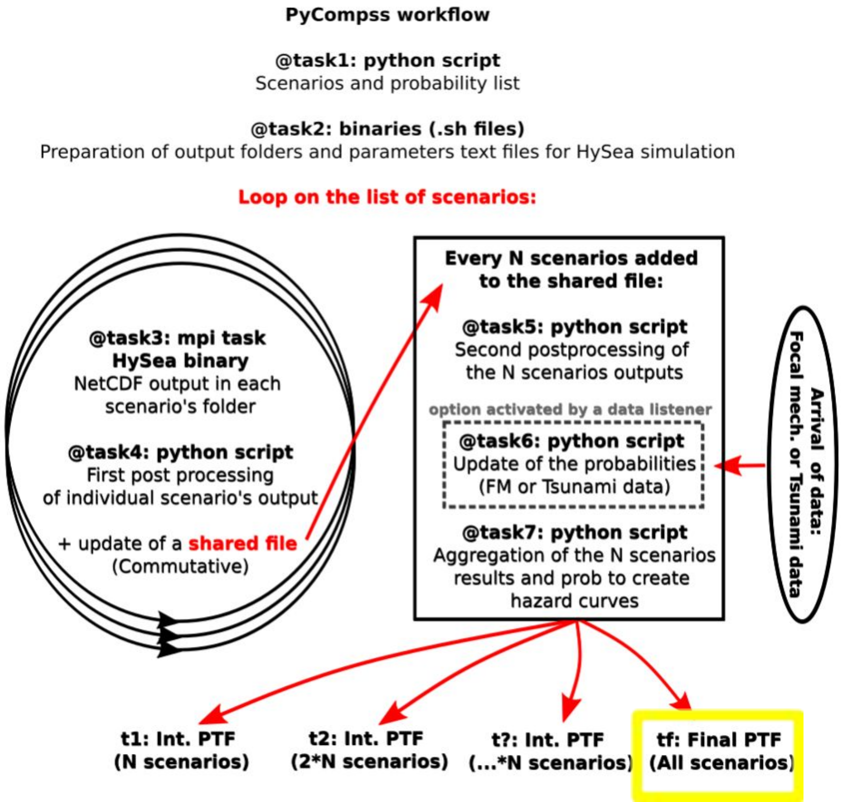
PyCOMPSs workflow: Step3, intermediate evaluation of the PTF





PyCOMPSs workflow: Step3, intermediate evaluation of the PTF



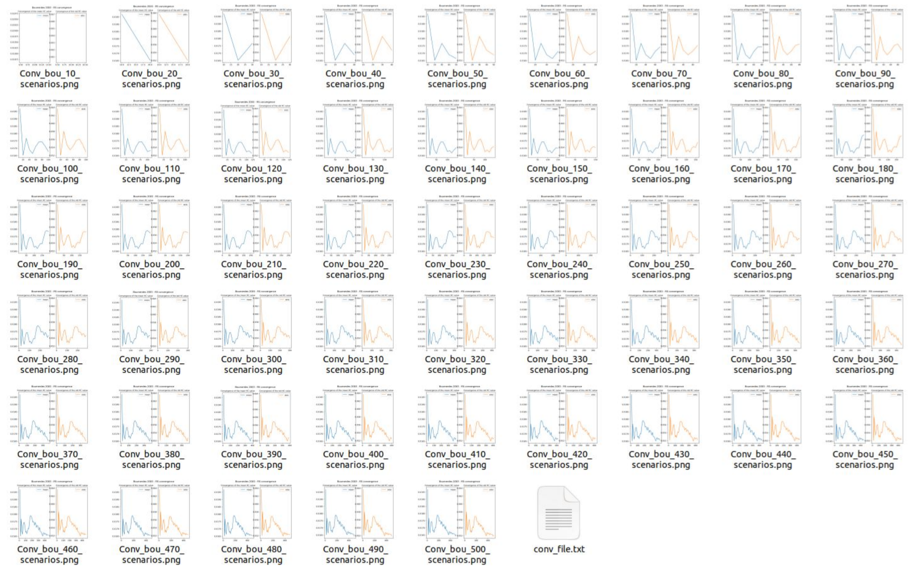


STEP 4 – Aggregation, calculation of the hazard curves

eFlows4HPC

Intermediate PTF results delivery

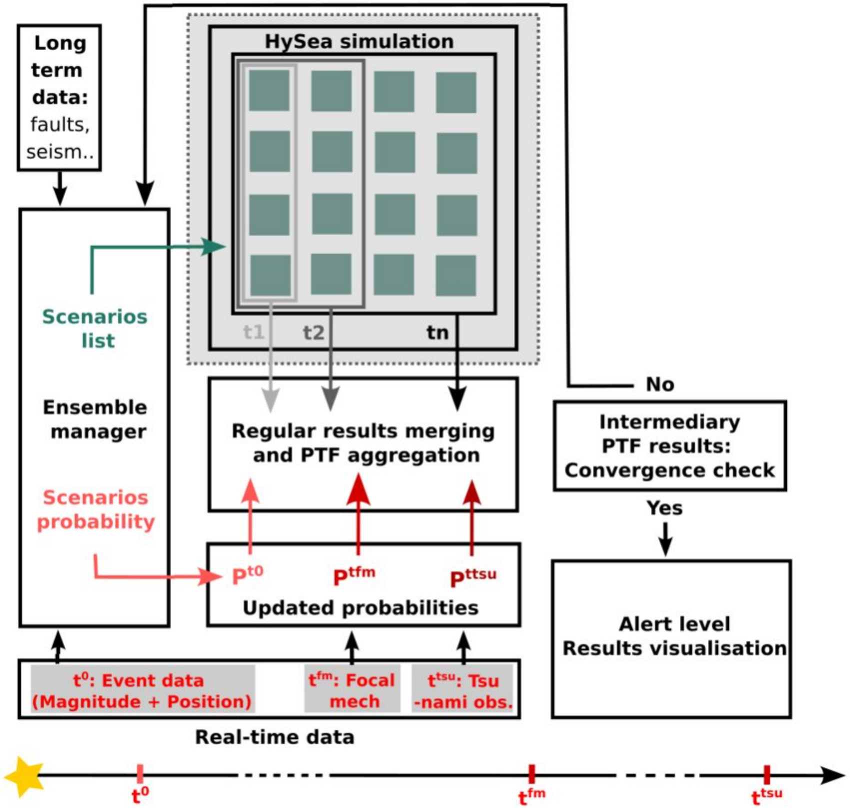
- A PyCompss commutative shared file allows the calculation of the intermediate/partial PTF hazard curves based on the available completed simulations and based on a predefined number N (every 100 scenario for example)
- Monitoring of the results: the failed simulations and the convergence of the results are monitored through the creation of different files.

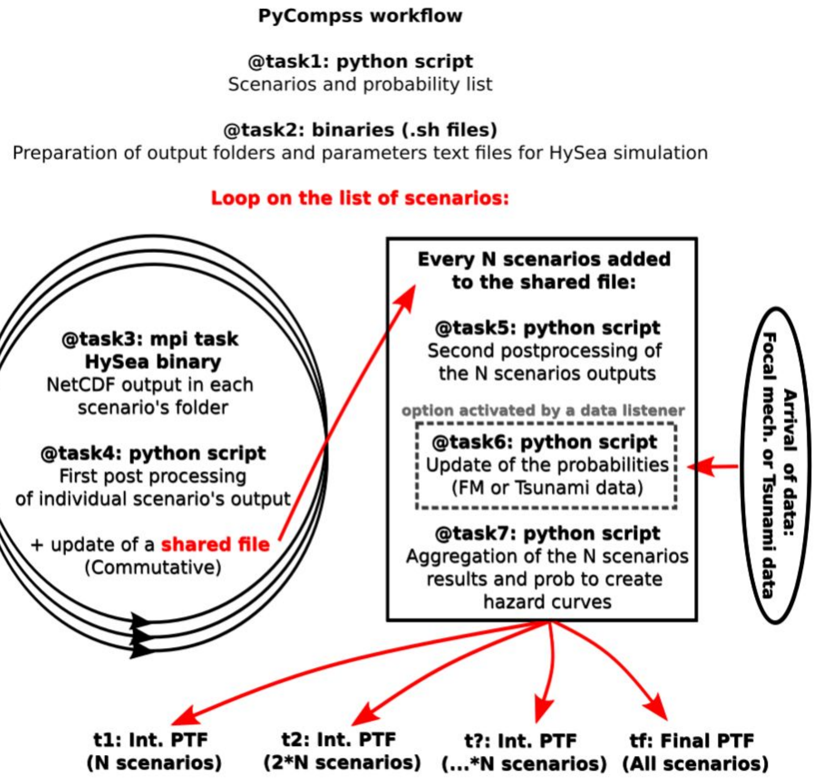


conv_file.txt

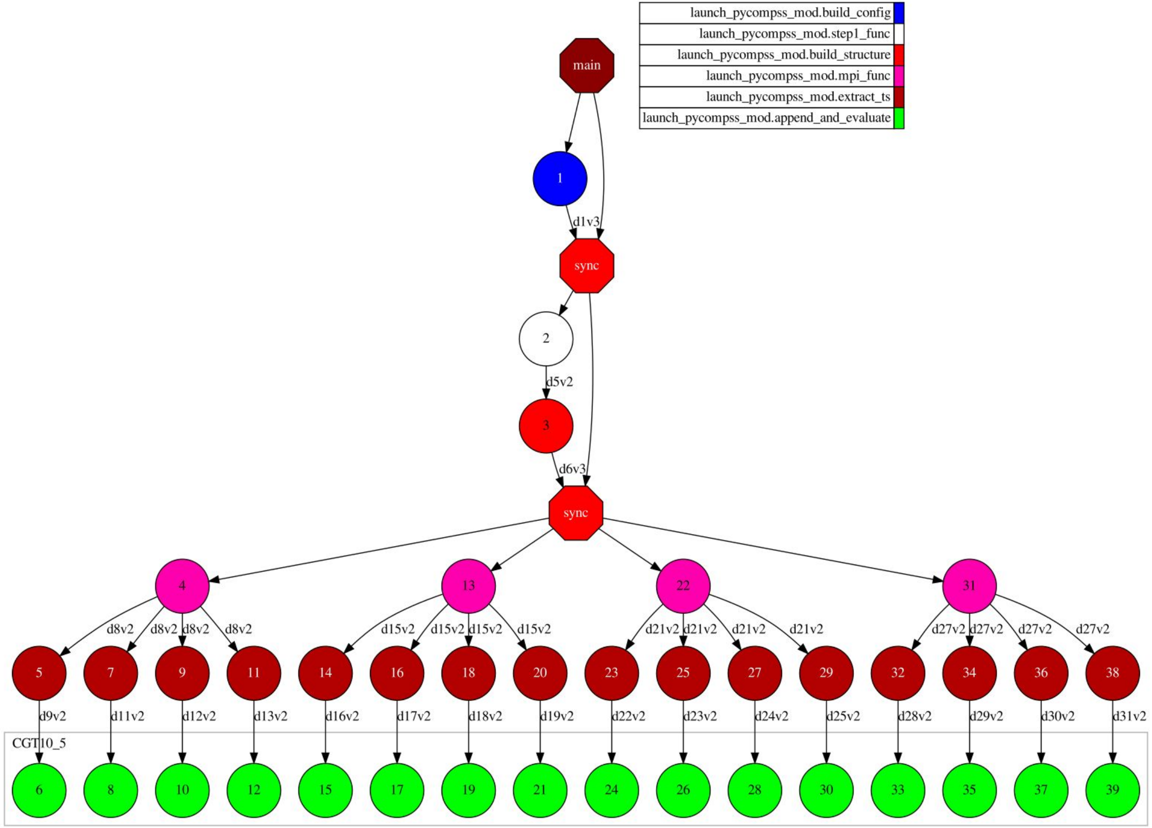
Nb	Mean	Var
10	0.019685	0.063672
20	0.016520	0.052025
30	0.017670	0.058008
40	0.016902	0.054563
50	0.016640	0.053382
60	0.017121	0.054338
70	0.017426	0.055758
80	0.017436	0.056078
90	0.017133	0.054667
100	0.016775	0.052910
110	0.016838	0.053351
120	0.016680	0.052805
130	0.016882	0.053854
140	0.017027	0.054219
150	0.017011	0.053979
160	0.017426	0.055319
170	0.017834	0.056664
180	0.017904	0.056869
190	0.017836	0.056807
200	0.017737	0.056587
210	0.017462	0.055572
220	0.017512	0.055533
230	0.017458	0.055412
240	0.017248	0.054687
250	0.017443	0.055113
260	0.017338	0.055023
270	0.017098	0.054295
280	0.017223	0.054679
290	0.017228	0.054623
300	0.017057	0.054005
310	0.016919	0.053462
320	0.017104	0.054020
330	0.016827	0.053031
340	0.016817	0.053000
350	0.016743	0.052754
360	0.016610	0.052235
370	0.016516	0.051909
380	0.016708	0.052685
390	0.016737	0.052766
400	0.016622	0.052415
410	0.016685	0.052604
420	0.016666	0.052568
430	0.016566	0.052198
440	0.016624	0.052511
450	0.016587	0.052321
460	0.016585	0.052307
470	0.016568	0.052282
480	0.016542	0.052271
490	0.016431	0.051900
500	0.016514	0.052024

PyCOMPSs workflow

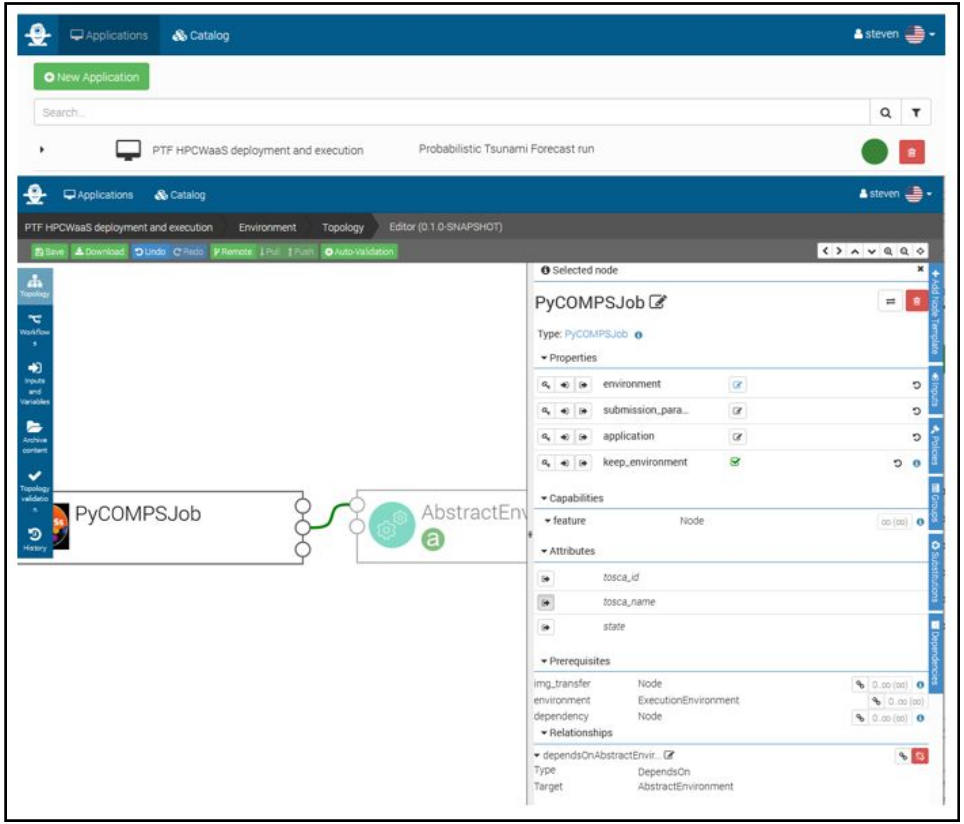


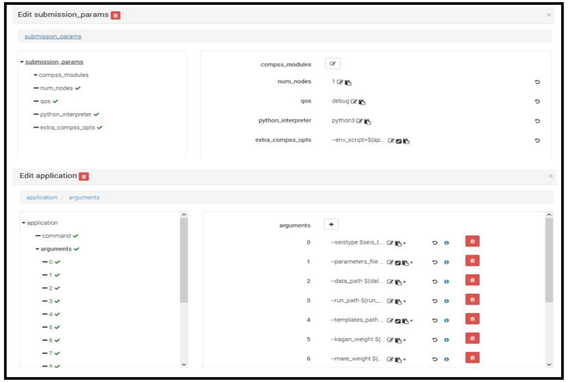


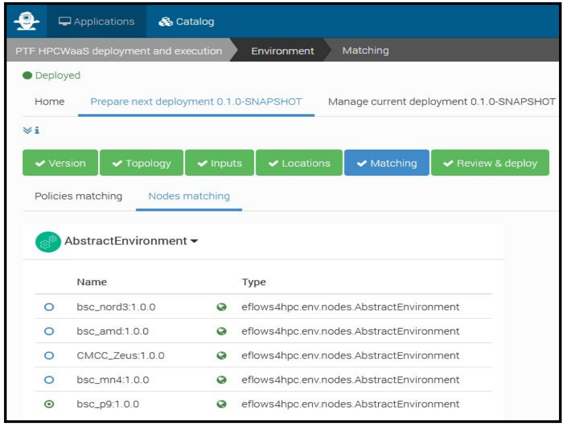
First end-to-end version of PTF orchestrated with PyCOMPSs



Deployment on HPCWaaS (TOSCA - ALIEN4CLOUD)

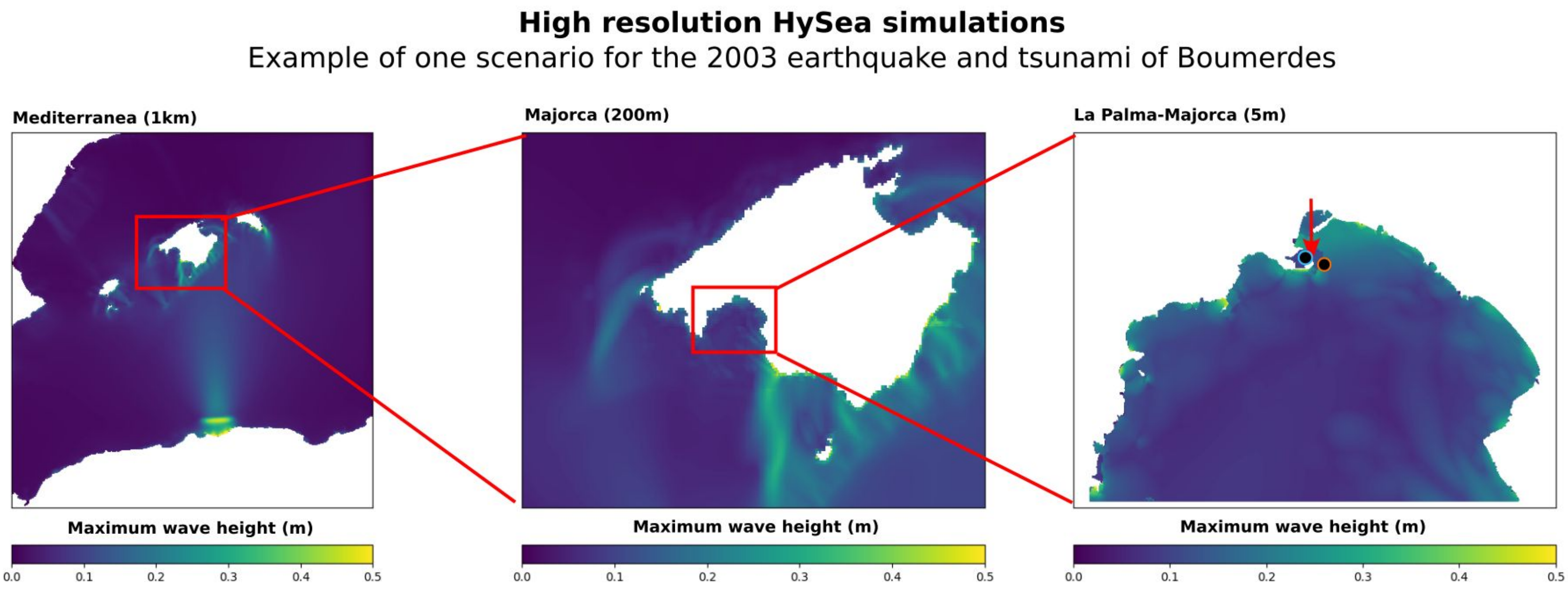






Further developments: High-Resolution PTF-PyCOMPSs workflow

eFlows4HPC

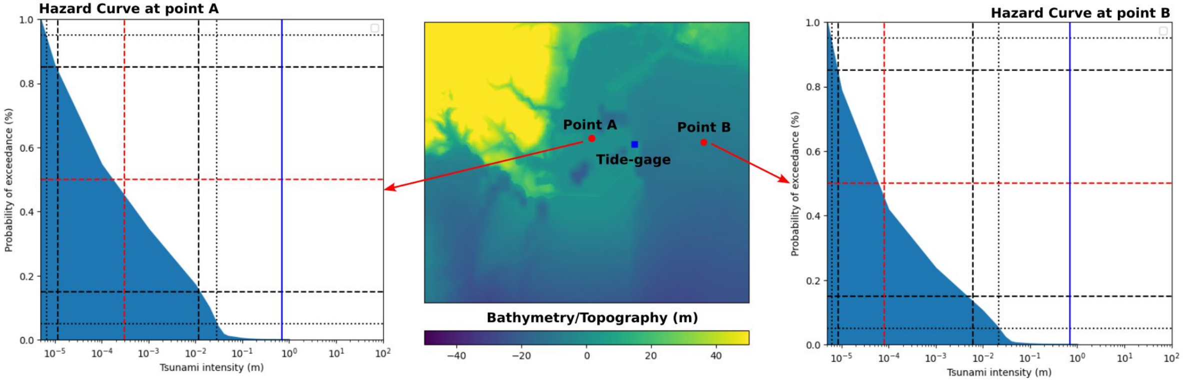


Further developments: High-Resolution PTF-PyCOMPSs workflow

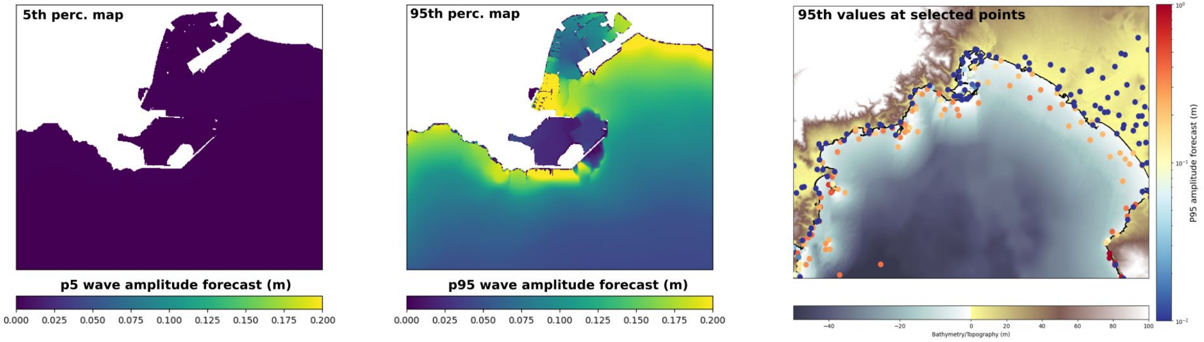
eFlows4HPC

Aggregation of the results of 500 simulations for the forecast calculation

Calculation of the hazard curves at each point of the grid



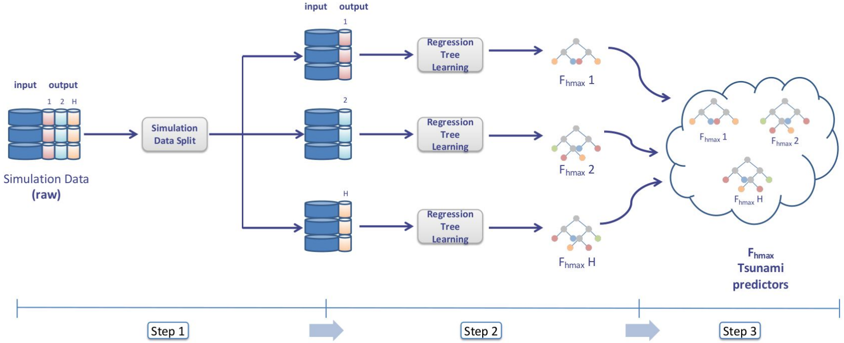
Creation of mean or percentiles maps (mean, p5, p95) - Extraction of the values at specific locations



Further developments: ML / AI

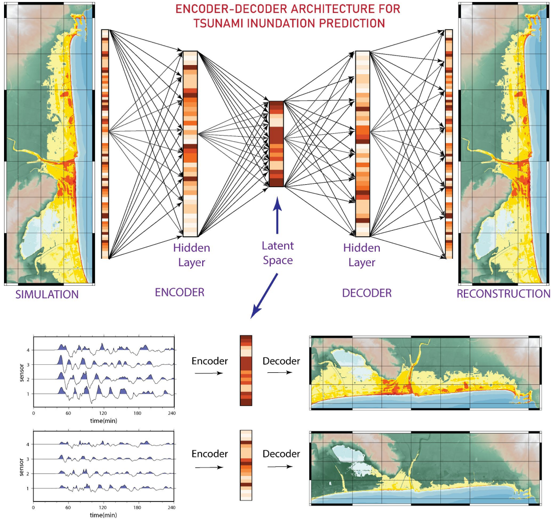
Tsunami Forecasting exploiting Regression and Classification Trees

This activity is aimed at developing machine learning approaches based on regression and classification trees, to model and forecast tsunami simulation results.



Inundation Prediction from Offshore Time-Series

To use Machine Learning to Predict High-Resolution inundation (expensive computations) from Offshore Time-Series (far cheaper computations) for Accurate Hazard Prediction in Urgent Tsunami Computations



Conclusions

eFlows4HPC

- The development of UC workflows for earthquakes and tsunamis has been incorporating the deployment of advanced tools and the development of complex tasks to reach an operational level.

- The sustainability and improvements in the workflows will be done under ChEESE-2p and DT-GEO projects

Thank you

eFlows4HPC

Enabling dynamic and Intelligent workflows in the future EuroHPC ecosystem

www.eFlows4HPC.eu

@eFlows4HPC   eFlows4HPC Project

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 955558. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Spain, Germany, France, Italy, Poland, Switzerland, Norway.

Requirements & Metrics for Pillar III workflows

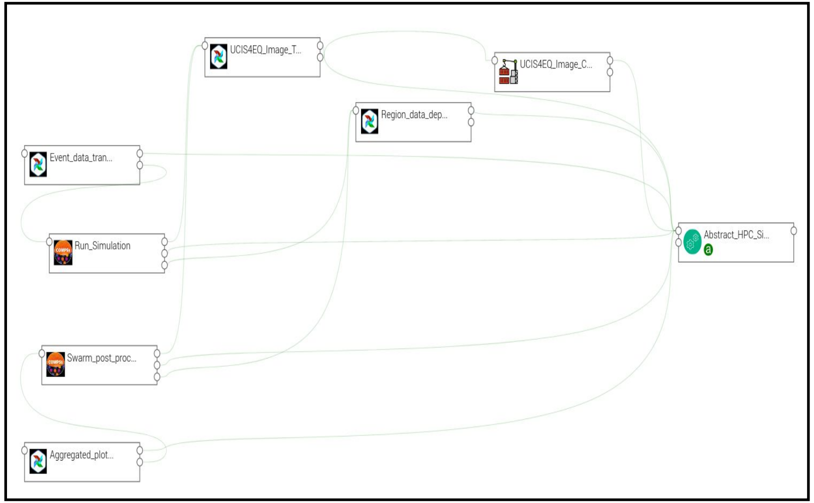
eFlows4HPC

ID	Name	Priority
1	Urgent computing access	must
2	Data accessibility	should
3	Data replication	must
4	Execution robustness	must
5	Infrastructure interoperability	must
6	Portability and Reusability	may
7	Streaming Data Source	must
8	Integrated workflow manager	must
9	Integration with permanent storage	must
10	Inference with online/offline ML models	must
11	DA integration	may
12	Workflow malleability	should

Acronym	Name	Description
RT	Response time	End-to-solution time constraints in an urgent computing context. This metric is defined as the clock time measured from the event reception until a first valid solution for the event is delivered to the stakeholders such as civil protection agencies.
UAR	Urgent Assignment of Resources	Time to obtain necessary resources for an urgent computing execution. The inclusion of this metric quantifies the QoS in HPC facilities that provide UC services. Moreover, it is a measure to evaluate if the adopted policies are adequate for an UC execution
RUQ	Results Uncertainty Quantification	High-fidelity and high-accuracy results. This metric is proposed to fulfill the specific UCIS4EQ workflow. RUQ metrics is related to the uncertainty of the service outputs, as it is crucial to constrain and reduce the uncertainty of provided impact estimates.
Conv	Convergence	This metric is proposed to evaluate the specificity of the PTR/FTRT requirements in particular the convergence of the results based on a reference solution

Deployment with HPCWaaS platform

To facilitate the reusability of these complex workflows in federated HPC infrastructure.



Integration of the UCIS4EQ in the HPCWaaS platform describing the TOSCA components using the Alien4Cloud software.

TOSCA components involved in the deployment and execution of the UCIS4EQ workflow:

Setup phase:
The ***Abstract_HPC_Site*** component defines the properties (login node address, CPU architecture, supported container engine, …) of the HPC system where we mean to deploy and run the workflow.

Deployment phase:
The ***UCIS4EQ_Image_Creation*** component implements the interaction with the eFlows4HPC Container Image Creation (CIC) service to build a container Image including all the software components required for the workflow. The ***UCIS4EQ_Image_Transfer*** component implements the interaction with the Data Logistics Service (DLS). It depends on the *UCIS4EQ_Image_Creation* component because it has to know the URL of the generated container image in order to perform the image deployment. The ***Region_Data_deployment*** component interacts with the DLS, but in this case it is configured to download the data of a region (maps, etc.) from the data-set stored in the UCIS4EQ B2DROP repository.

Execution phase:,

two TOSCA components (Run_Simulation and Swarm_post_processing) and **two data pipelines** (the stage-in of the event data to simulate (*Event_data_transfers)* and for the stage-out to upload the generated plots at the end of the swarm post-processing workflow (*Aggregated_plots_upload*).