A EUPEX-ChEESE cooperation: the SPECFEM3D(++) example; context, preliminary results and next steps

Présenter Piero Lanucara CINECA

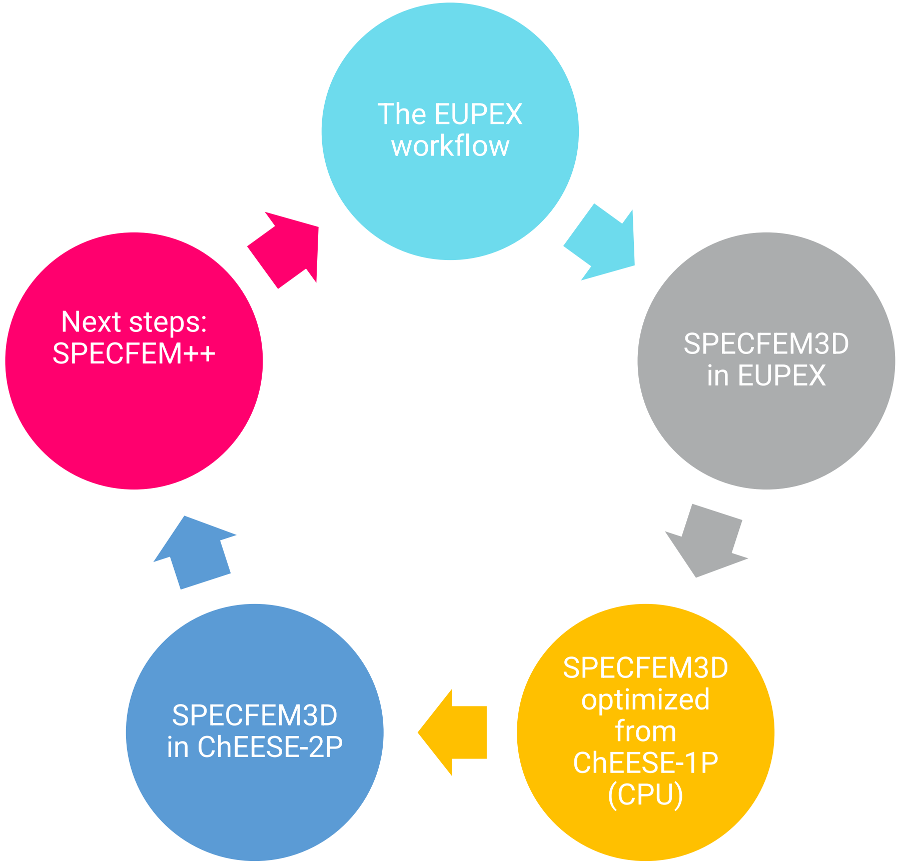
EUPEX Forum 2024

EUPEX

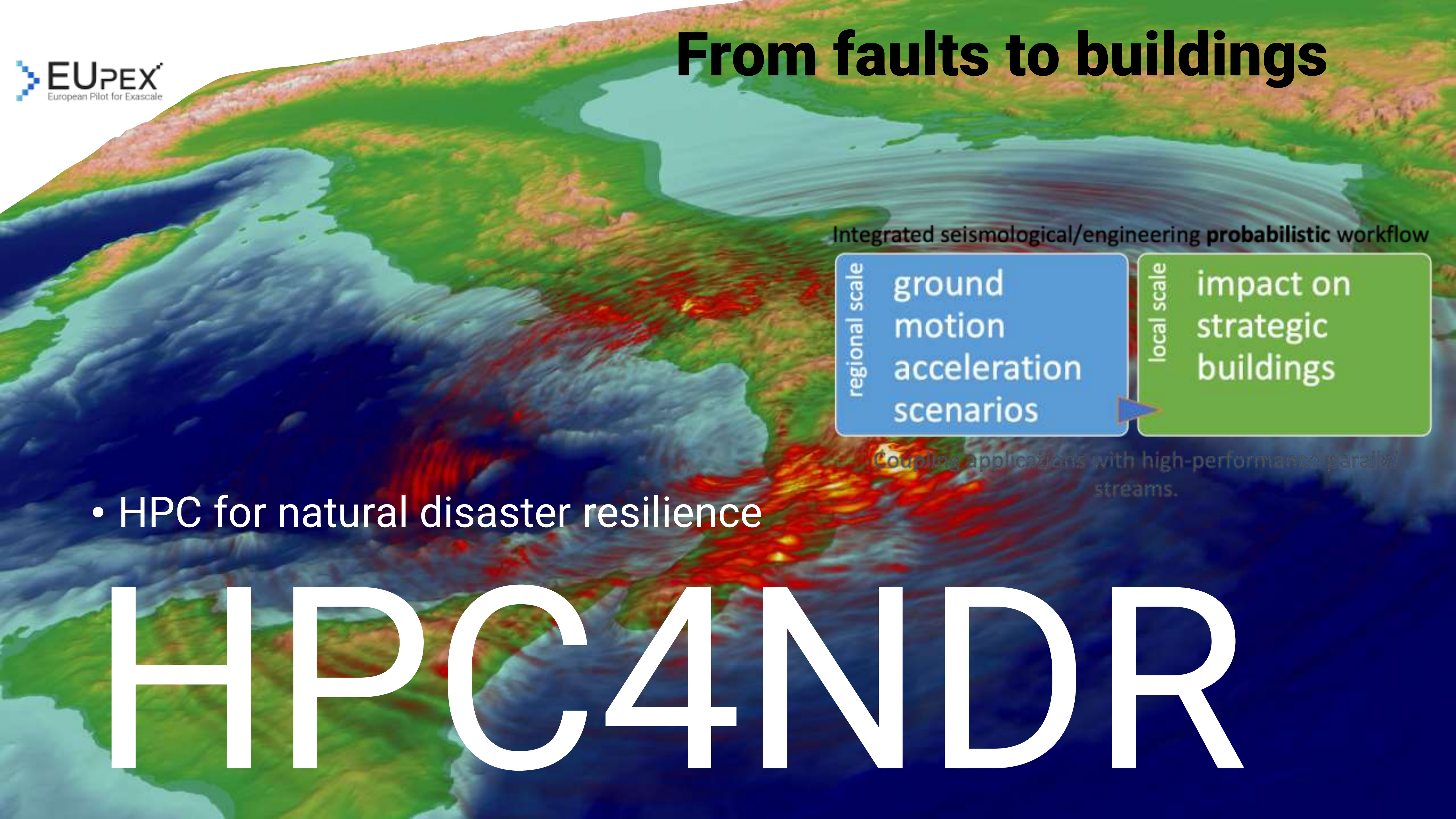
EuroHPC Joint Undertaking

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101033975. The JU receives support from the European Union's Horizon 2020 research and innovation programme and France, Germany, Italy, Greece, United Kingdom, Czech Republic, Croatia.

Agenda

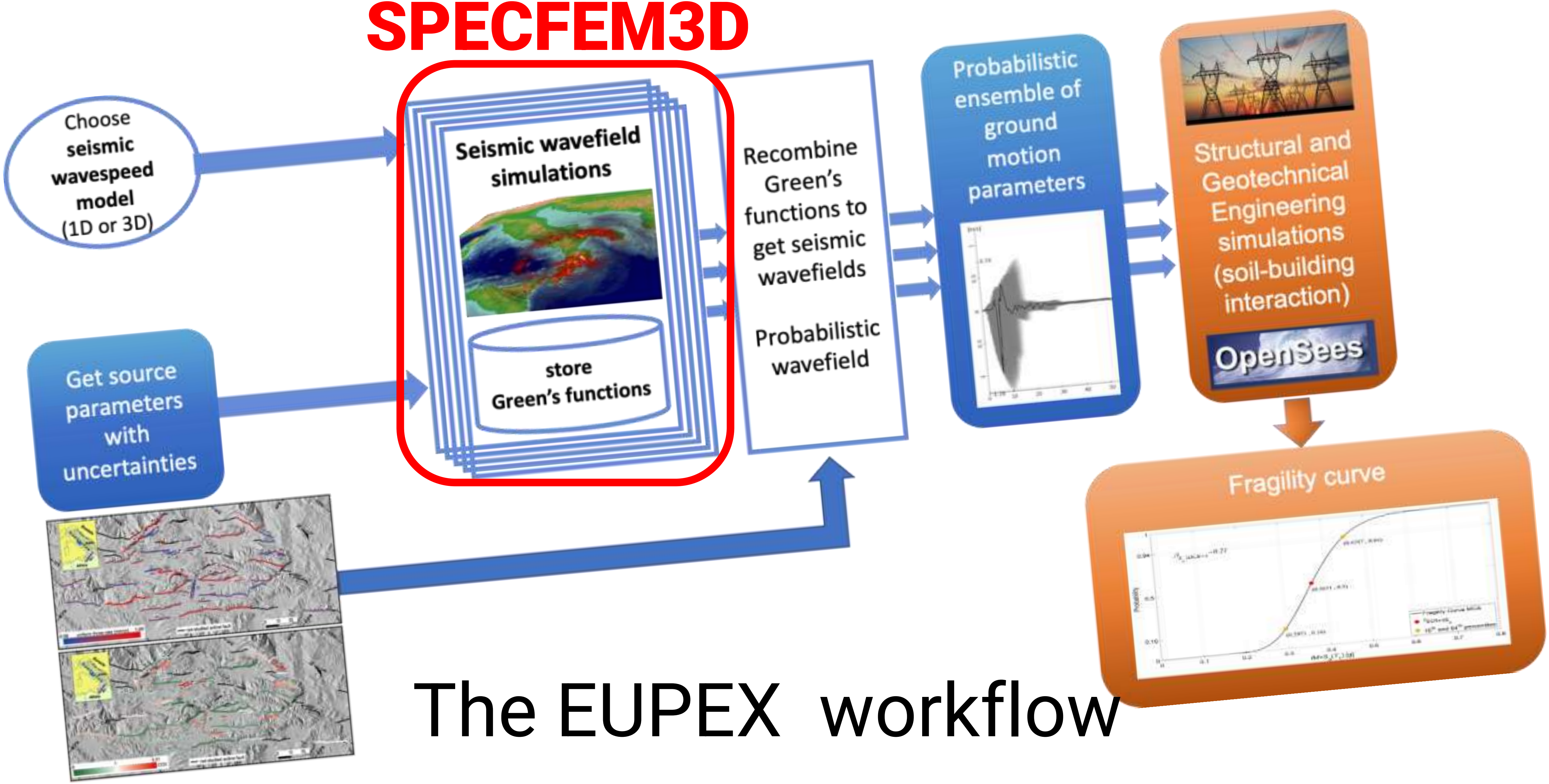

The EUPEX workflow
SPECFEM3D in EUPEX
SPECFEM3D optimized from ChEESE-1P (CPU)
SPECFEM3D in ChEESE-2P
Next steps: SPECFEM++

From faults to buildings



- HPC for natural disaster resilience

HPC4NDR

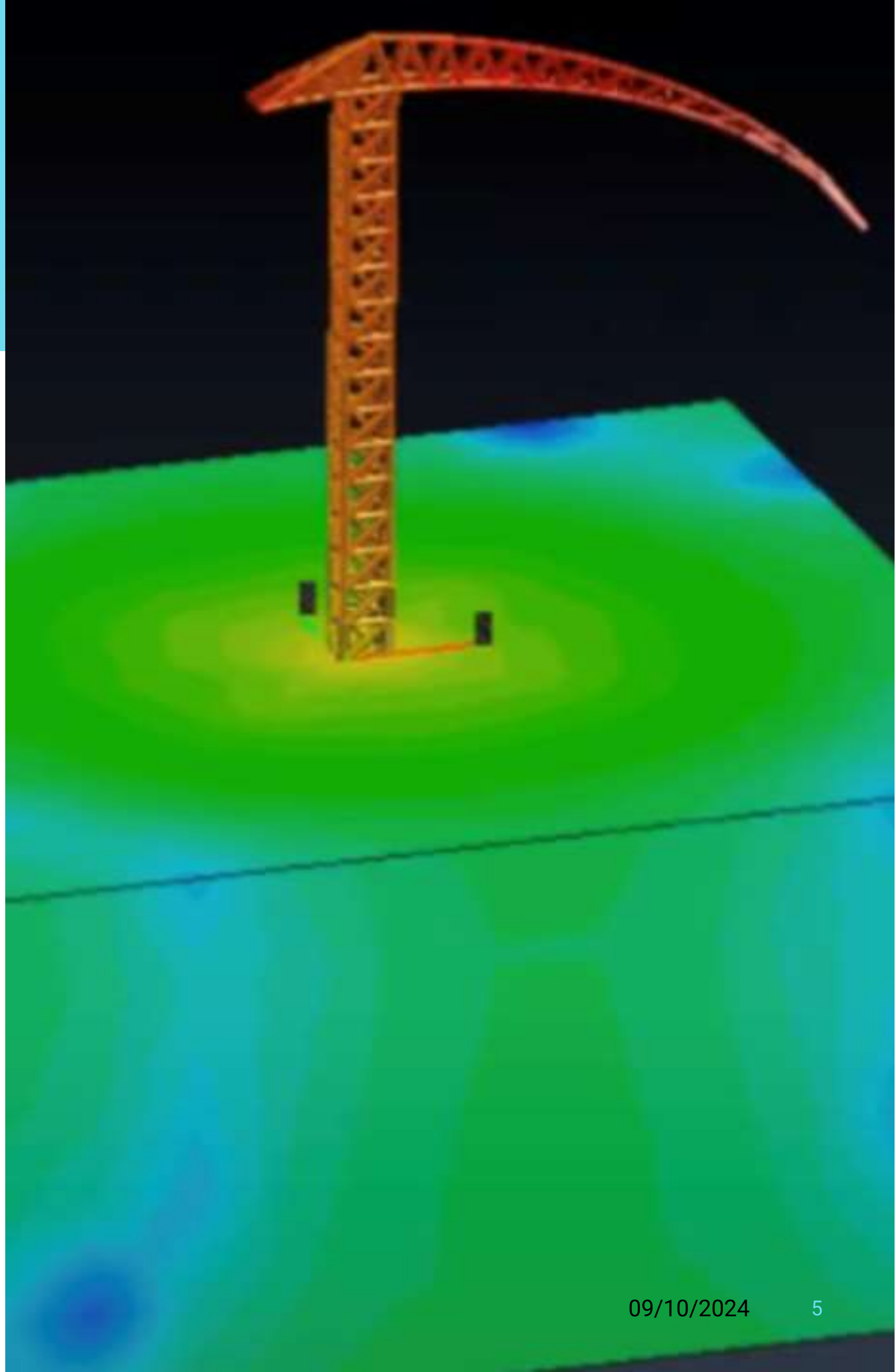


The EUPEX workflow

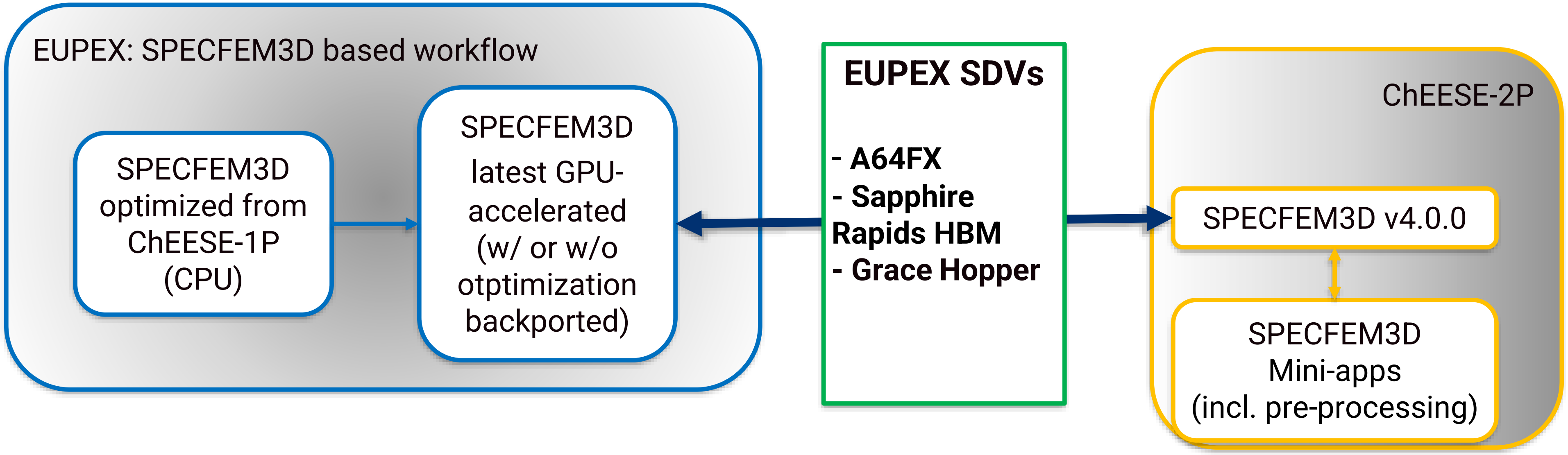
- From geology to probabilistic seismic engineering analysis

Structural and Geotechnical Engineering simulations

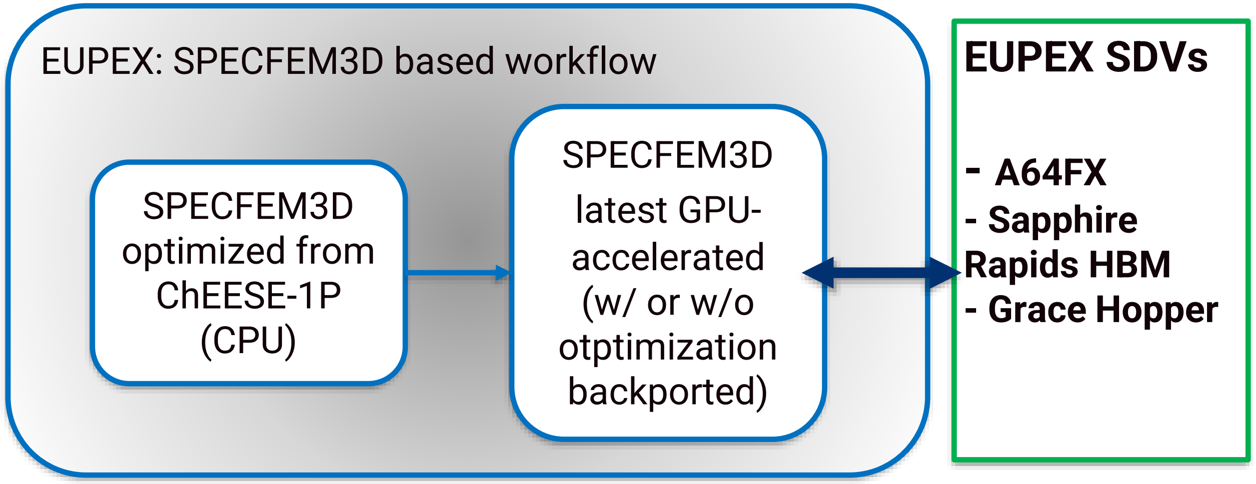
- The Engineering applications will be performed using the OPEN-SOURCE framework **OpenSeeS**
    - Soil-Structure Interaction (SSI), of large domains
    - HPC capabilities in developing detailed large scale seismic probabilistic frameworks



ChEESE-EUPEX Collaboration Diagram for SPECFEM3D



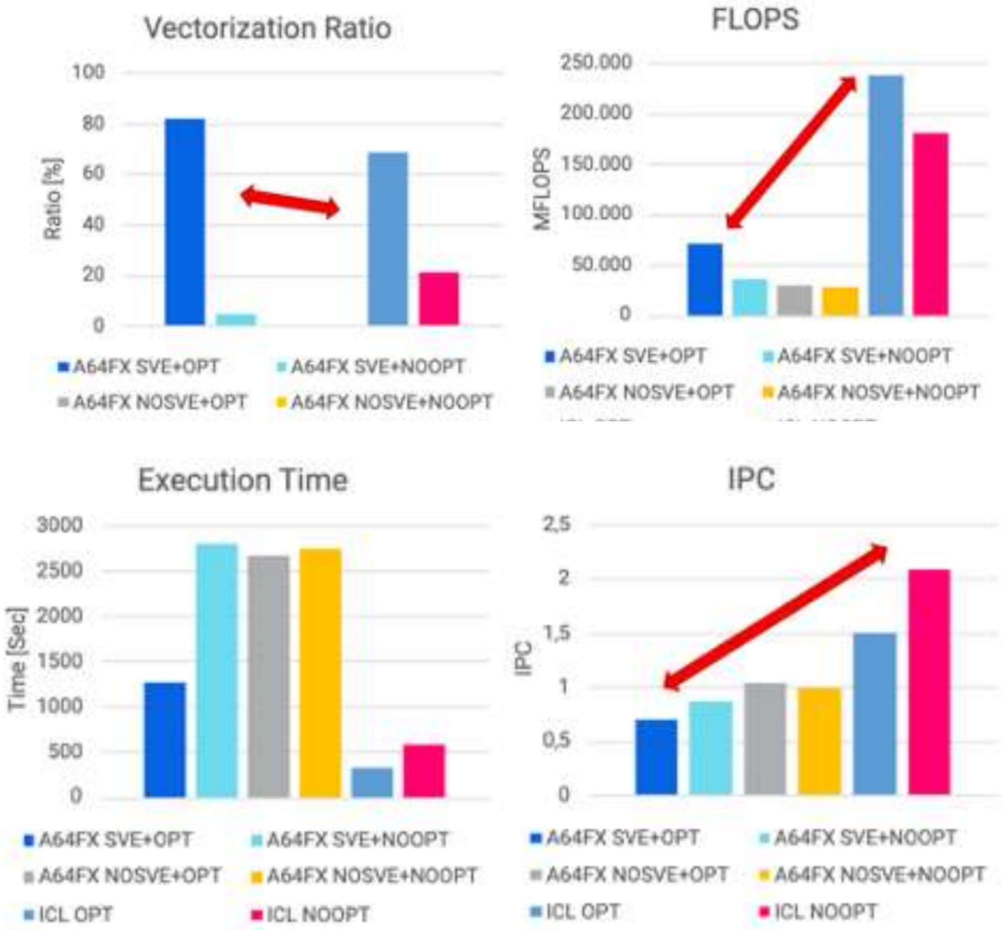
SPECFEM3D IN EUPEX

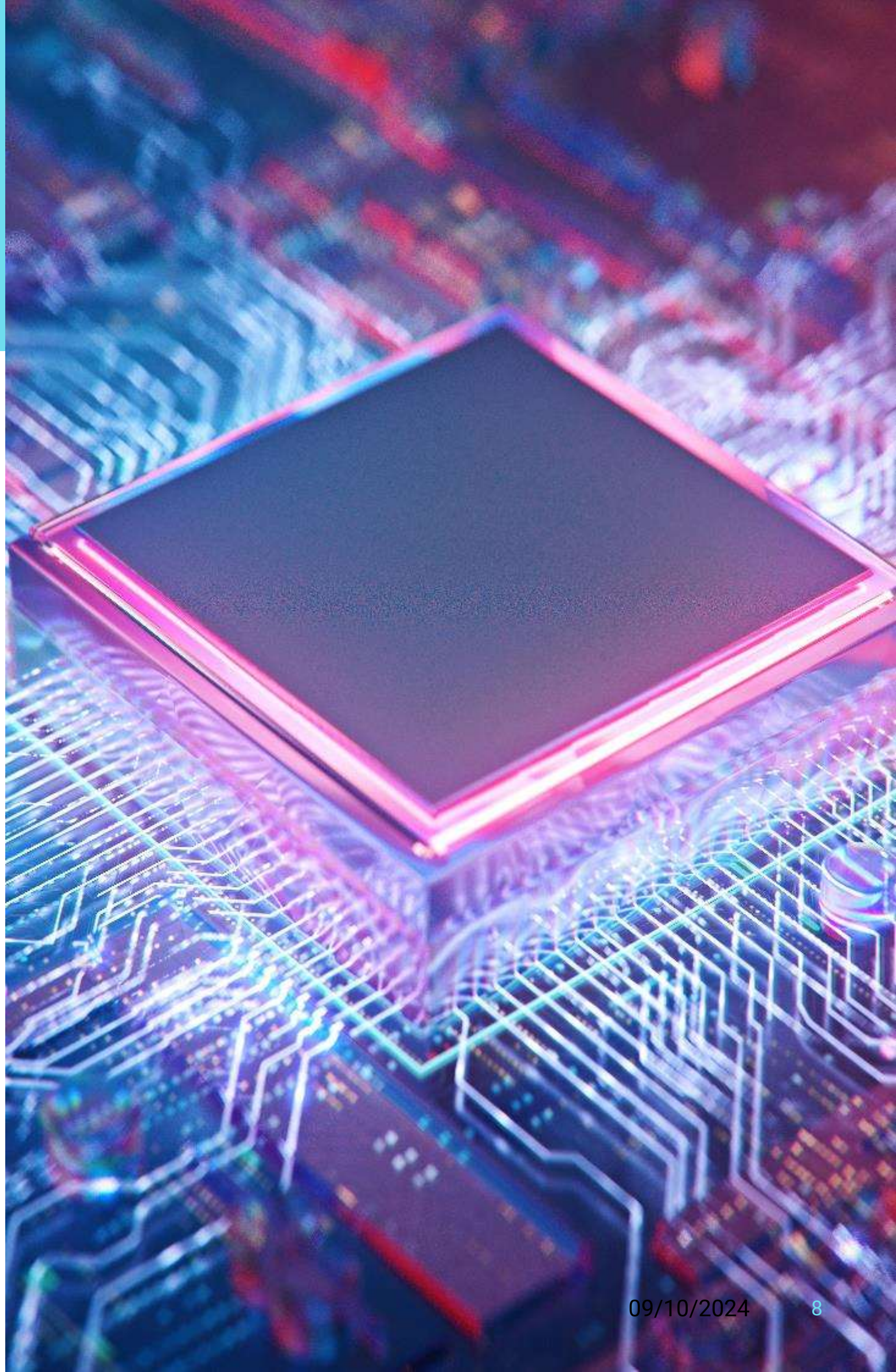


- Target a complete workflow including SPECFEM3D
- Focus on optimized version from ChEESE-1P (CNRS/Atos/Eviden version)
- Two main branches:
  - Elastic simulation (vectorization improved for CPU architectures)
  - Full SPECFEM3D (including seismic tomography features): still missing some optimization/tuning (anisotropic materials). Available in zenodo
- But target latest accelerated version for production level test case and associated workflow
- A64FX performance tests done, will continue on Grace Hopper (soon)

SPECFEM3D optimized from ChEESE-1P (CPU)

- Performance assessment of the ChEESE-1P optimized version on Irene (A64FX)
- Comparison with G100 (x86 machine in CINECA)
- Preliminary work done evaluating HBM improvement (on-going)





ChEESE-2P CoE

ChEESE

17 **Beneficiary Organisations**
From 8 different countries

1 **Affiliated Entity**
IPGP (affiliated to CNRS)

---

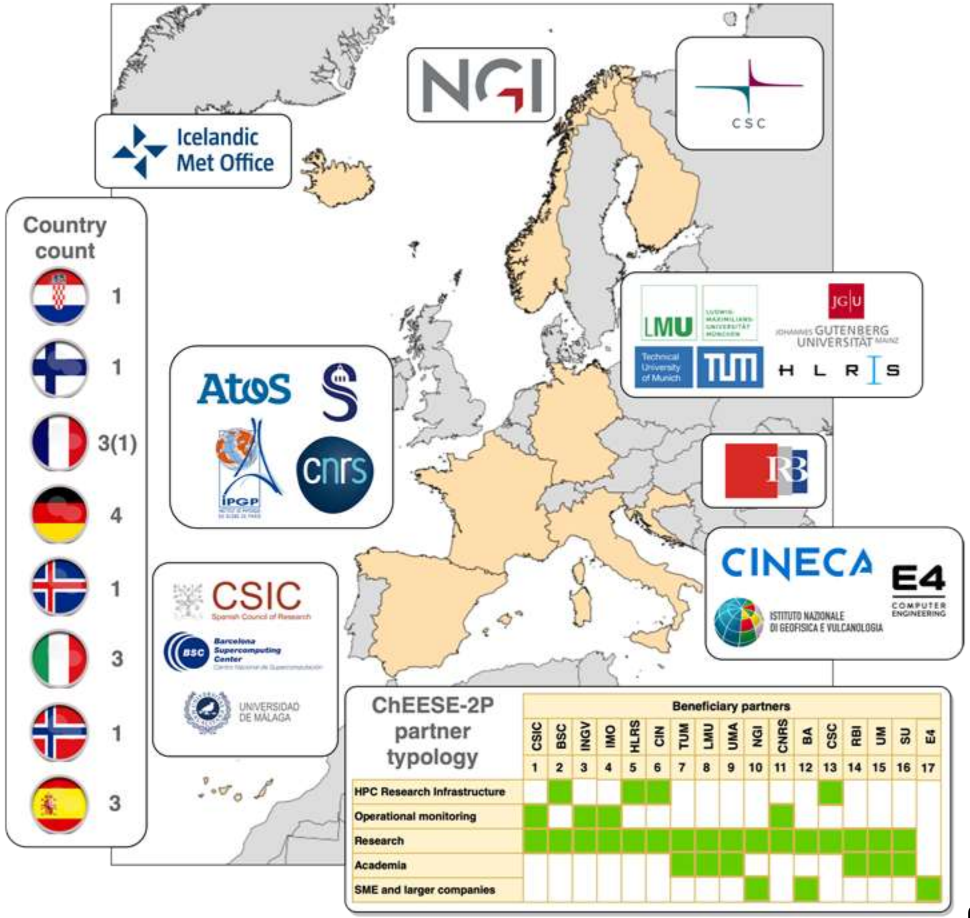
4 **HPC tier-0 Centers**
BSC, CIN, HLRS, CSC

3 **Private Companies**
BA, NGI, E4

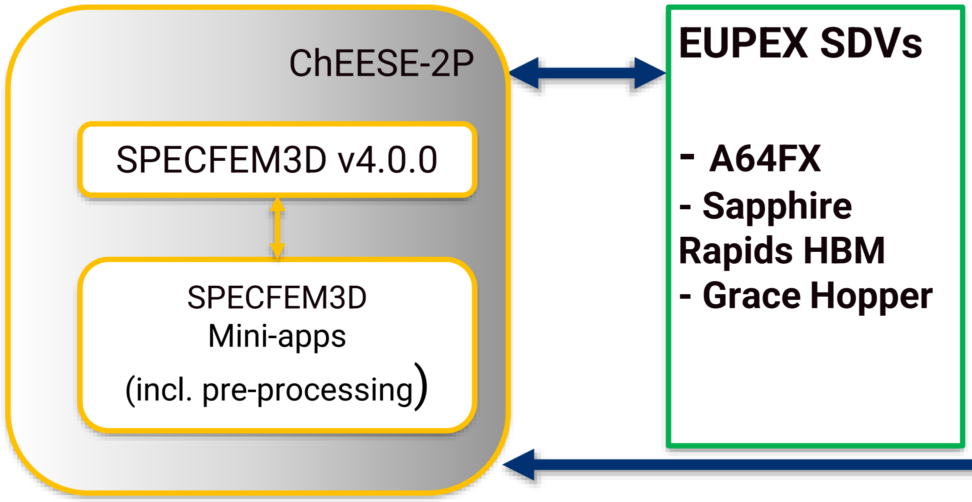
4 **Operational Monitoring**
CSIC, INGV, IMO, CNRS

6 **Academia**
TUM, LMU, UMA, RBI, UM, SU

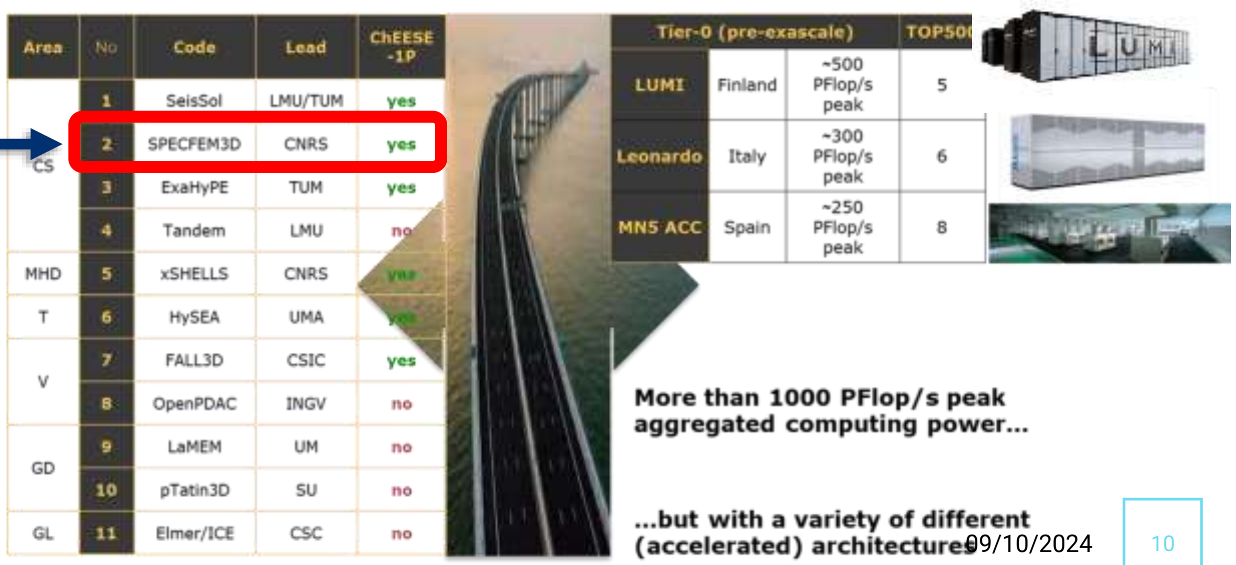
ChEESE-2P Consortium Composition



SPECFEM3D IN ChEESE-2P



- WP2 "Exascale technical challenges in flagship codes"
  - Leonardo booster code audit (scaling etc.) and suggestions using the full app (published in D2.1)
  - Ongoing work for D2.2 (Dec. 2024) on performance portability campaign on GPU (MN5, LUMI, Leonardo) on the mini-apps.
  - Optimization in WP2: Reduce memory access, try different compression strategies (i.e., compressed arrays), need to check precision and do some tests, now on CPU then on GPU.
- WP3 "Co-design with European HPC vendors and technologies"
  - Focus on mini-apps, target ARM SVE and HBM before EUPEX availability.



Area	No	Code	Lead	ChEESE -1P
CS	1	SeisSol	LMU/TUM	yes
	2	SPECFEM3D	CNRS	yes
	3	ExaHyPE	TUM	yes
	4	Tandem	LMU	no
MHD	5	xSHELLS	CNRS	yes
T	6	HySEA	UMA	yes
V	7	FALL3D	CSIC	yes
	8	OpenPDAC	INGV	no
GD	9	LaMEM	UM	no
	10	pTatin3D	SU	no
GL	11	Elmer/ICE	CSC	no

Tier-0 (pre-exascale)			TOP50
LUMI	Finland	~500 PFlop/s peak	5
Leonardo	Italy	~300 PFlop/s peak	6
MN5 ACC	Spain	~250 PFlop/s peak	8

SPECFEM3D IN ChEESE-2P

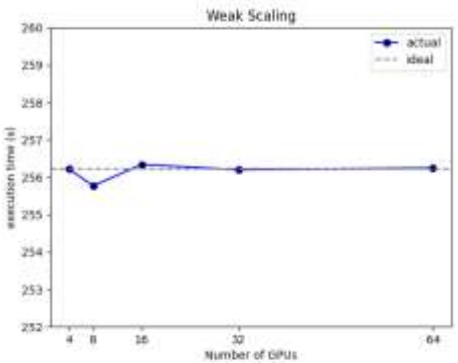
ChEESE

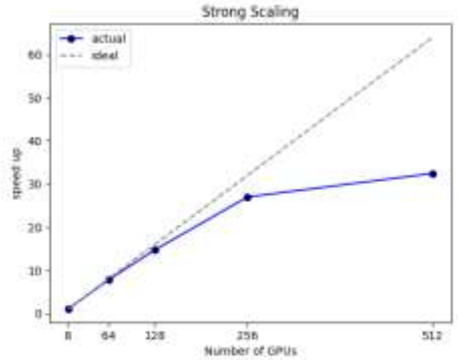
https://specfem.org/

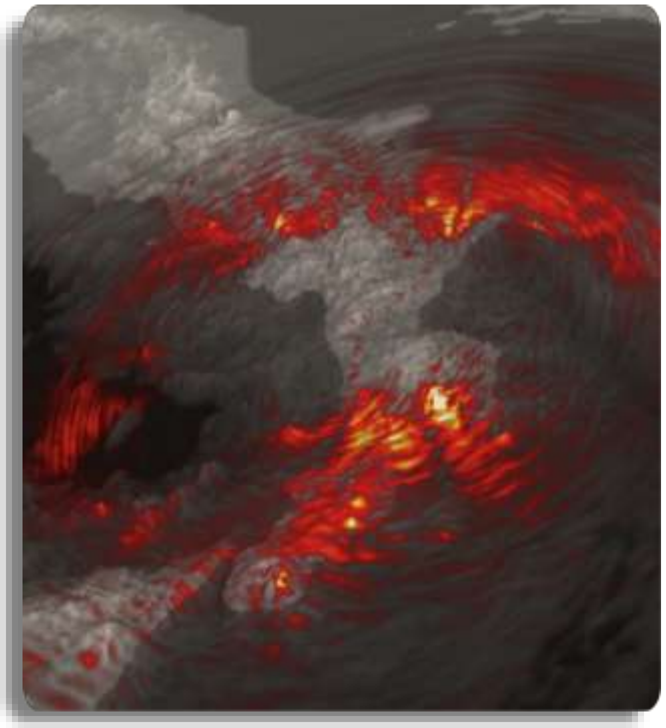
- Spectral-element method:
  - CPU, GPU, MPI (domain decomposition)

- Solves linear seismic wave propagation in 3D models :
  - Elastic, viscoelastic, piroelastic, and fluid solid interactions
  - Implements imaging and Full Waveform Inversion for complex models

- Scalability evaluated on Leonardo booster:
  - Weak scaling : from 4 to 64 GPUs, 360448 mesh elements per GPU
    excellent weak scaling up to 64 GPUs (and likely beyond)
  - Strong scaling :
    good strong scaling up to 256 GPUs
    overlapping MPI communication with GPU computation





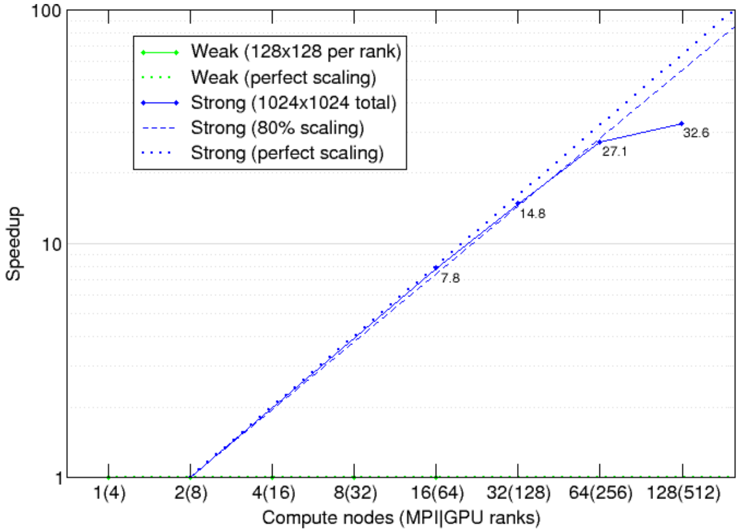


SPECFEM3D has been optimized for various GPU architectures (using primarily CUDA and HIP as GPU programming model).

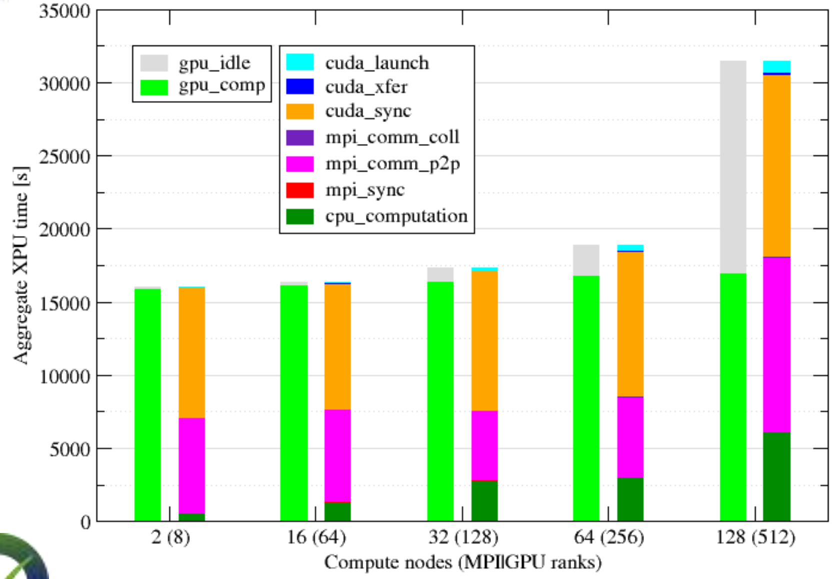
Performance Optimisation and Productivity

ChEESE-2P GPU assesment on SPECFEM3D

ChEESE



Problem size	1024x1024	1024x1024	1024x1024	1024x1024	1024x1024
MPI|GPU ranks	**8**	**64**	**128**	**256**	**512**
Wall time [s]	**2001.806**	**255.948**	**135.478**	**73.846**	**61.400**
Global scaling efficiency	0.995	0.973	0.919	0.843	0.507
- Computation time scaling	*1.000*	0.987	0.972	0.950	0.937
- Parallel efficiency	0.995	0.986	0.945	0.887	0.541
- - Load balance efficiency	1.000	0.998	0.996	0.994	0.989
- - Orchestration efficiency	0.995	0.988	0.948	0.892	0.547



POP3_AR_002

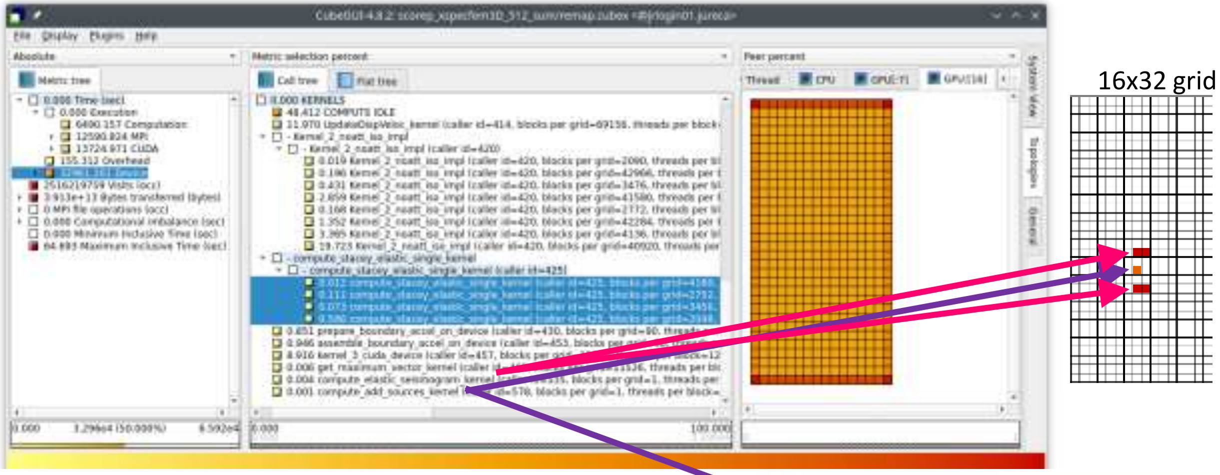
- *iterate_time* (solver) chosen as Focus of Analysis
- Good strong scaling up to 256 GPUs
- With 512 GPUs no longer able to sufficiently overlap MPI communication with CUDA kernels

Performance Optimisation and Productivity

https://zenodo.org/records/13643996

ChEESE-2P GPU assesment on SPECFEM3D

ChEESE

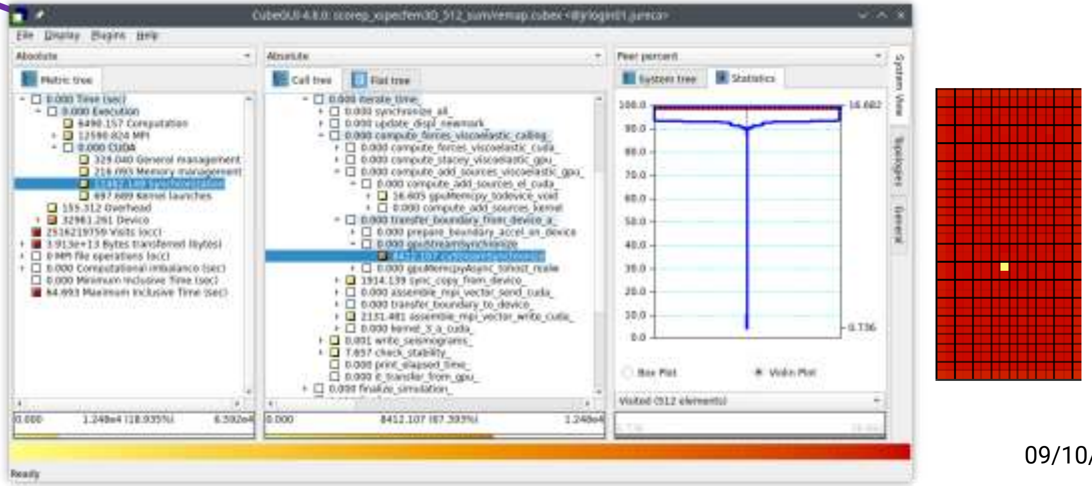


2D domain decomposition on GPUs

- many kernels are well balanced
- some kernels execute much faster for interior compared to edges
- only four GPUs handle seismogram receivers
- only one GPU (#243) executes *compute_add_sources_kernel*

compute_add_sources_kernel executed on single GPU (#243) is rather short, however, results in all other GPUs having very long synchronization times in following *transfer_boundary_from_device_a*

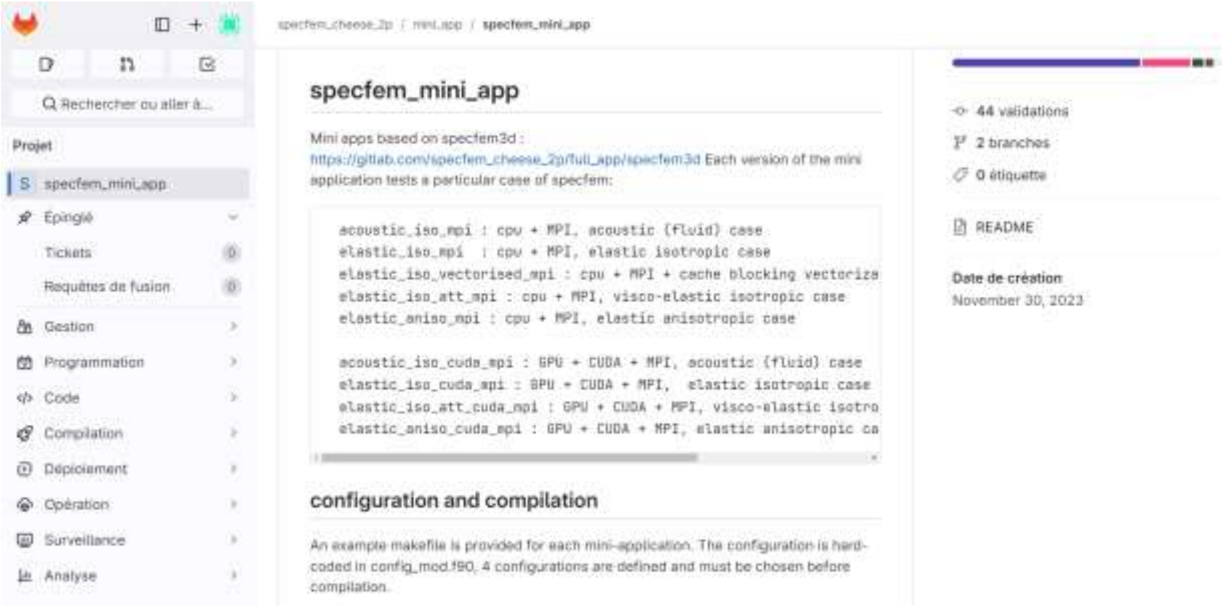
- over two-thirds of CUDA synch time and over 30% of total CPU time



ChEESE-2P co-design on SPECFEM3D

ChEESE

- Mini-apps developed by Vadim Monteiller (CNRS)
  - CPU, GPU (CUDA) and MPI
  - https://gitlab.com/specfem_cheese_2p/mini_specfem/specfem_mini_app
- Goals
  - Ported and optimized for EUPEX and EUPILOT
  - Performance portability by testing different programming models/languages
- Status
  - Performance characterization on x86 ✓
  - Ported to ARM (Ampere Altra and Grace) ✓
  - Vectorization optimization ✓
  - HBM study ⌛
  - More optimization dedicated to HBM & ARM SVE



ChEESE-2P co-design on SPECFEM3D

ChEESE

- First optimizations to improve vectorization on main kernel "compute_forces":
  - First attemps by hand: loop reordering, temporary arrays, etc.
  - Finally: addition of !$OMP SIMD directives leads to best performance
- Results on AMD Milan 7763 (1 socket with 64c)
- As mini-apps are all memory bound, the HBM should be effective in improving performance
- Next step: study the performance on HBM and optimize memory layout and access

Mini app	Execution time	Compute bound	Memory bound	Vectorized
elastic_aniso_mpi	259 s	77%	23%	No
elastic_iso_mpi	140s	49%	51%	No
elastic_iso_att_mpi	275 s	49%	51%	No

Add !$OMP SIMD directives at the right spots

Mini app	Execution time	Speedup	Compute bound	Memory bound	Vectorized
elastic_aniso_mpi	166 s	1,6	61%	39%	Yes
elastic_iso_mpi	116 s	1,2	40%	60%	Yes
elastic_iso_att_mpi	240 s	1,1	37%	63%	Yes

Next steps: SPECFEM++

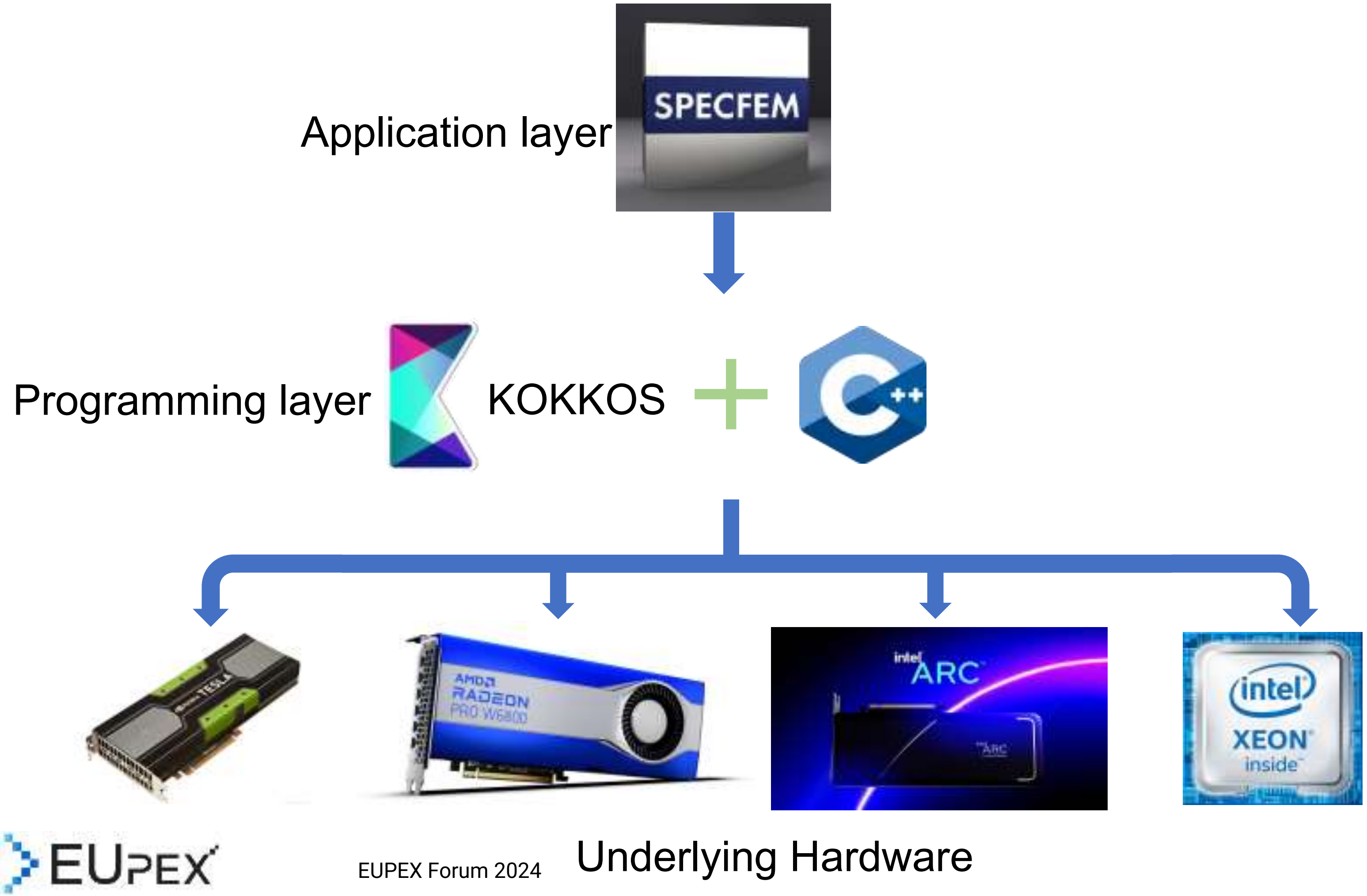
- On production: use of the accelerated version of SPECFEM3D (on one node: 30x speedup CPU vs GPU on Jean Zay machine).
- CUDA, HIP and CPU version in the same code with 2 different memory structures (one for CPU et one for GPU), difficult to maintain.
- Maintain the CPU version anyway.

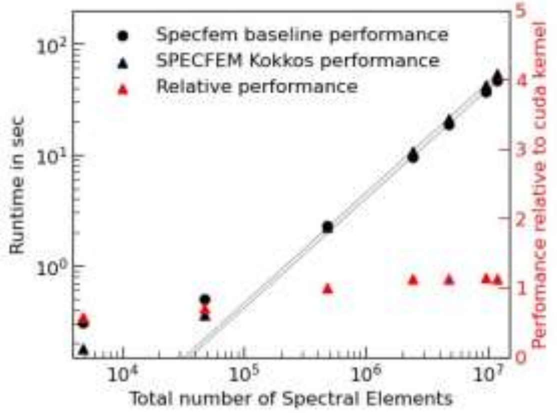
Why use Kokkos?

	CUDA/HIP/DPC++	OpenMP 5/ OpenACC	Kokkos
Portability across architectures	No. Need to write separate kernels for every architecture	Yes. Single source code with pragma-based approach	Yes. Single source code implemented using Kokkos functions
Performance	Optimized performance	Tough to optimize for performance	Very good performance
Cost of portability	Very high	[illegible] Why use Kokkos?	High
Cost of maintenance	Very high. Newer architectures might require tuning of kernels	Low. Assuming compilers do a good job of implementing the standard	Low. Assuming Kokkos backend is always optimized
Compiler dependence	N/A	These are standards, vendors have a flexibility on implementation	N/A
Fortran Support	No. Could use bindings	Yes	No. Could use bindings

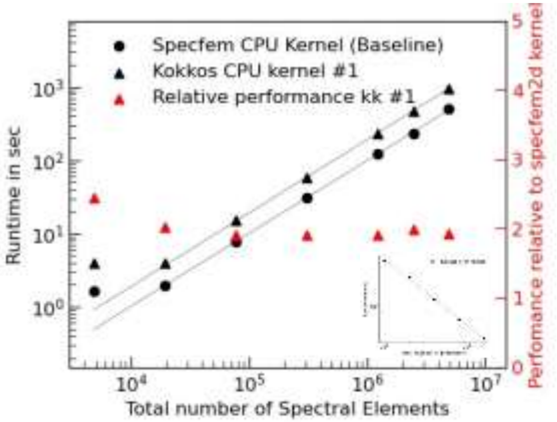
PRINCETON UNIVERSITY 24

Next steps: SPECFEM++





Comparison on GPU



Comparison on CPU

Acknowledgement

Funded by the European Union. ChEESE has received funding from the European High Performance Computing Joint Undertaking (JU) and Spain, Italy, Iceland, Germany, Norway, France, Finland and Croatia under grant agreement No 101093038.

Thanks to (among the others):

- Vadim Montellier
- Emanuele Casarotti
- Brian Wylie
- Rohit Kakodkar
- Pelagie Alves
- Stéphan Jauré

CINECA, CNRS, INGV/HPC4NDR, BULL/ATOS/EVIDEN, Princeton Univ.

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101033975. The JU receives support from the European Union's Horizon 2020 research and innovation programme and France, Germany, Italy, Greece, United Kingdom, Czech Republic, Croatia.