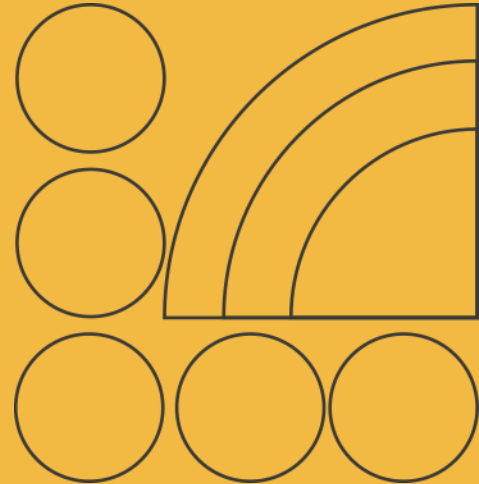




ChEESE

WP2 towards D2.4

Piero Lanucara
CINECA



WP2 Tasks



WP2 tasks					
Task	Name	Lead partner	Start month	End month	Percentage at M38 (%)
2.1	Code audits	HLRS	3(27)	15(36)	100%
2.2	Increasing the performance portability of flagship codes for different accelerators	CIN	1	48	79%
2.3	Optimisation for Exascale	CIN	6	48	76%
2.4	Continuous Performance Efficiency Monitoring	BSC	18	48	66%



WP2 Deliverables



D	Deliverable name	Short description	Lead	Type	Level	M
2.1	First code audit report	First code audit round results and the recommendations arising from the analysis	HLRS	R	PU	12
2.2	First report on improving code performance at exascale	Report on porting, optimization and tuning for the selected applications on the available hardware	CIN	R	PU	24
2.3	Second code audit report	Update of D2.1 and further recommendations arising from the analysis	HLRS	R	PU	30
2.4	Final report on improving code performance at exascale and Best Practice Guide	Technical report describing the achieved scalability level and related performances obtained for the selected applications on the available hardware including a final Best Practice guide	CIN	R	PU	48



Original D2.4 Pillars: different tasks (before the 2nd review meeting)

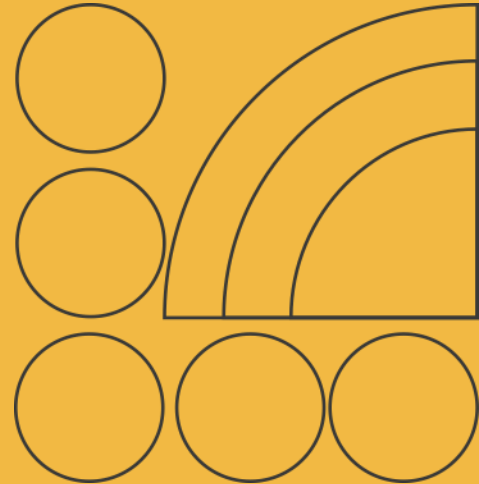
- **Finalising** the improvements already planned with respect to task T2.3 and not yet implemented.
- **Contributing**, in the presence of new mini-apps or different versions of a given mini-app, to new performance portability campaigns (T2.2).
- **Continuing** the work on TALP for performance monitoring, with particular emphasis on monitoring accelerated codes and implementations on new hardware (T2.4).
- **Developing** lessons learned on the general methodology as well as specific to each code, in anticipation of future developments and projects.





ChEESE

R2 and R3 from 2nd review meeting



R2 – Code Prioritisation and Focus

- Recommendation: **Code Prioritisation and Focus:** *It is recommended that the amount of effort versus the level of progress, as well as the prospects for each code, be reassessed. To maximize impact, we recommend to consider **limiting works on codes that show marginal potential for significant progress or speedup**. Instead, the focus should be shifted to a select few ones that are most likely to achieve a breakthrough within the community.*
 - Reassess effort vs progress and future prospects per code.
 - Focus resources on a limited number of high-potential codes.
 - Reduce effort on codes with marginal expected impact.



Response to R2 – Vertical vision (Per Code basis)

- Prioritisation led by Code owners
- Each code defines its priorities within WP2 tasks
 - Priorities analysed and validated by the WP2 management
 - Consolidation into a WP2 priority roadmap
 - Alignment of effort, milestones, and scientific impact



Response to R2 – Overall vision (for each code)

- Prioritisation led by WP2 Management
 - **Focusing** on **Scientific Grand Challenges** (aka Simulation Cases)
 - **Privileging Production-ready** accelerated Code versions already running on pre-exascale systems (whenever possible)
 - **Targeting** the project **Exascale ambition** (see the [presentation](#) done during the 2nd review meeting in Luxembourg for details)

Priority List "=" Vertical + Overall vision



R3 – Transversal Analysis of Results

- Recommendation: **Transversal Analysis of Results.** *Numerous code improvements have addressed issues affecting the efficiency of high-performance codes on accelerators. It is recommended that the consortium **present results in a more cross-cutting manner rather than per an individual code basis.** We recommend adopting a transversal representation of the results and codes. This involves presenting progress on common issues that cut across a collection of codes and libraries, rather than reporting on each one individually.*
 - Present results in a cross-cutting manner
 - Highlight common bottlenecks and shared solutions
 - Emphasize reusable methodologies



Response to R3

- Results collected **per code** and **per task (as usual)**
 - D2.4 will be organised in **transversal format**
 - **Harmonised** contributions from different codes
 - Structured around **common challenges** w.r.t. an hypothetical (pre)exascale hardware



Early Start of D2.4

- As the new transversal structure was not originally foreseen, WP2 management proposes to initiate D2.4 preparation well in advance.
- **Concrete steps:**
 - Share a preliminary D2.4 skeleton early with the full WP2 team.
 - Enable continuous feedback and iterative refinement.
 - Reuse preliminary ideas and parts of the ToC from the former ChEESE D2.2 (see [here](#)), adapting them to the new transversal structure.
 - This approach aims to ensure alignment, coherence, and timely delivery.



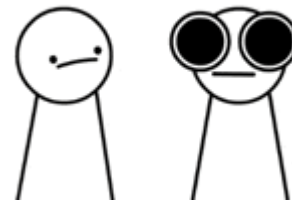
Early Start of D2.4: 4-bullet criteria

- **Priority setting** based on **Scientific Grand Challenges (Simulation Cases)**, combining code-level (vertical) and overall visions.
- **Performance optimisation and scalability improvement** towards (pre-)exascale architectures.
- **Cross-code coordination and harmonisation** of approaches, solutions, and methodologies.
- **Monitoring, validation, and impact assessment**, ensuring alignment with the project's **Exascale ambition**.



The Exascale ambition “Big Picture”

I HAVE ALREADY SEEN IT



Code	Performance analysis	Performance Portability	Code Restructuring	Alignment to EPI
SeisSol		CUDA, ROCm, SYCL support; high bandwidth	TensorForge batching; fused kernels	EuPEX/EuPILOT ;
SPECFEM3D		CUDA optimal, HIP ~87-100%	Mini-app kernel optimization, GPU-Aware MPI	EuPEX;
ExaHyPE		NVIDIA A100/H100, Intel Max 1550	CUDA prototypes; optimized OpenMP offload	
Tandem	No code audit data	LUMI-G (AMD), Leonardo (NVIDIA), LUMI-C (CPU); HIP efficient	Runtime selection CPU-GPU	EuPEX;
xSHELLS	PTA bottleneck	CUDA/HIP optimized; shared-memory tuned	Parallel Thomas Algorithm & pipelining	EuPEX/EuPILOT;
HySEA		NVIDIA A100/H100/GH200, Intel Max 1100/1550; efficiency >=0.98	SYCL porting maintains CUDA functionality	EuPEX/EuPILOT;
FALL3D	POP GPU efficiency drops at high GPU counts	Unified codebase: NVIDIA/AMD GPUs + CPUs	MPI+OpenACC hybrid; device-aware MPI	EuPILOT;
OpenPDAC			Algorithmic optimization (~2x faster); variable-iteration solver	
LaMEM	Multigrid hotspot	PETSc backends (CUDA, HIP, Kokkos); NVIDIA & AMD GPUs	Multi-node GPU scaling; solver tuning	
pTatin3D	MPI load imbalance audit; POP efficiency 6 ~36%	AMD GPU support (HIP)	PETSc pc_telescope fixes; mini-app kernel tests	
ELMER/CE	Computational efficiency drop at 38% at 512 GPUs	NVIDIA AMGX (CUDA), AMD rocALUTION (HIP)	Matrix assembly offloaded to GPU	

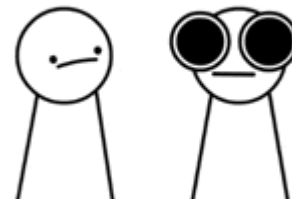
Comments:

- Good progress. 3 codes (**SeisSol**, **SPECFEM3D** and **HySEA**) fully aligned to their ChEESE-2P Exascale ambitions.
- 3 other codes are more or less on track (**ExaHyPE**, **xSHELLS** and **FALL3D**) but **xSHELLS** bottleneck deserve particular attention (to be reported in RP3)
- **OpenPDAC** activity in ChEESE-2P is strongly devoted to CPU investigation (with more than an eye to external initiatives like the ExaFOAM project)
- performance analysis shows some interesting issue in the performances. Part of those issues could be analyzed and solved in RP3.



The Exascale ambition “Big Picture”

I HAVE ALREADY SEEN IT



Code	Performance analysis	Performance Portability	Code Restructuring	Alignment to EPI
SeisSol		CUDA, ROCm, SYCL support; high bandwidth	TensorForge batching; fused kernels	EuPEX/EuPILOT ;
SPECfem3D		CUDA optimal, HIP ~87-100%	Mini-app kernel optimization, GPU-Aware MPI	EuPEX;
ExaHyPE		NVIDIA A100/H100, Intel Max 1550	CUDA prototypes; optimized OpenMP offload	
Tandem	No code audit data	LUMI-G (AMD), Leonardo (NVIDIA), LUMI-C (CPU); HIP efficient	Runtime selection CPU-GPU	EuPEX;
xSHELLS	PTA bottleneck	CUDA/HIP optimized; shared-memory tuned	Parallel Thomas Algorithm & pipelining	EuPEX/EuPILOT;
HySEA		NVIDIA A100/H100/GH200, Intel Max 1100/1550; efficiency >=0.98	SYCL porting maintains CUDA functionality	EuPEX/EuPILOT;
FALL3D	POP GPU efficiency drops at high GPU counts	Unified codebase: NVIDIA/AMD GPUs + CPUs	MPI+OpenACC hybrid; device-aware MPI	EuPILOT;
OpenPDAC			Algorithmic optimization (~2x faster); variable-iteration solver	
LaMEM	Multigrid hotspot	PETSc backends (CUDA, HIP, Kokkos); NVIDIA & AMD GPUs	Multi-node GPU scaling; solver tuning	
pTatin3D	MPI load imbalance audit; POP efficiency 6-36%	AMD GPU support (HIP)	PETSc pc_telescope fixes; mini-app kernel tests	
ELMER/CE	Computational efficiency drop at 38% at 512 GPUs	NVIDIA AMGX (CUDA), AMD rocALUTION (HIP)	Matrix assembly offloaded to GPU	

Immediate actions for code owners:

- “**Reassess the effort** in relation to the **progress and future prospects**” of the Codes in order to help us in defining a list of priorities.
- what we mean by ‘significant’ or ‘marginal’ progress? roughly, **yellow and red boxes** of this table should be exploited (revising/updating the big picture as well)
- **Everything to be clarified and finalized in the next few weeks (hopefully...)**

