

Short summary

- The following slides outlines required reporting on remaining WP2 effort and possible requests for support from CINECA. It also summarizes updated and planned work, including new performance-portability campaigns, pending improvements from Task T2.3, code-porting activities, and the integration of TALP into performance-monitoring workflows. Additional updates cover ongoing and planned TALP development and its porting to different GPU architectures.



Codes updates (code owners)

- **Updated results (if any)**
 - **New performance portability campaigns** We will soon install Elmer(/ice) on our new GH200 national platform (CSC) roihu and will have direct comparison between LUMI Mi200X and GH200
 - **Improvements:** Redoing primitive version of Elmer on LUMI using AMD-clang/flang (<https://github.com/cscva/elmer-install-scripts>) after system update. New I/O library ADIOS2 integrated in Elmer (work within TerraDT) – reduces amount of single written files and the over-all size (should ease some issues with LUSTRE FS)
 - **Porting activities** No
 - **Use (integration with code/workflow) of TALP in performance monitoring activities** No (needs clang/flang to work)
 - **AOB**



Codes updates to come (code owners)

- **Mandatory section**
 - **please, specify if you still have additional effort in WP2 for your code** We ran out of WP2 time allocation quite a while ago
 - **if not, please, specify if you plan to request effort from CINECA (or someone else in the project) from now on** We are partly compensating with other project that are aligned – nothing needed from within ChEESE consortium
- **Planned updates (if any)**
 - **New performance portability campaigns** (in the presence of new hardware and/or new mini-apps or existing mini-apps with different flavors) Comparison of GH200 with Mi200X (as mentioned before)
 - **Improvements** (in the presence of new enhancements identified in Task T2.3 that are still pending) we are working (together with AMD) on compiling Elmer on LUMI with AMD-clang/flang (AFAR). We are currently building a stack (with compiler + all needed libraries)
 - **Porting activities** (full code) No
 - **Use (integration with code/workflow) of TALP in performance monitoring activities** indirectly, as the compilation with clang/flang theoretically should open the door of TALP device analysis
 - **AOB**



TALP updates (BSC)

- **Updated TALP-based performance monitoring stuff**
 - Development of TALP
 - Porting to various GPU flavors
- **Planned TALP-based performance monitoring stuff**
 - Development of TALP
 - Porting to various GPU flavors

