



Contribution ID: 26

Type: **Poster**

## On MPI+(task-based OpenMP) performances

*Tuesday 21 March 2023 18:30 (1 hour)*

The architecture of supercomputers is evolving to expose massive parallelism by considerably increasing the number of compute units per node. HPC users must adapt their applications to remain efficient on current and ultimately on future hardware. Open Multi-Processing (OpenMP) and the Message Passing Interface (MPI) are two HPC programming standards widely used and both aim at performant and portable codes but work on different parallelism levels: the shared and the distributed memory levels.

OpenMP proposes a task-based programming model which composability shall enable seamless hybridization with other asynchronous programming models such as MPI. Composing the two standards thus appears as a well-suited solution for performant and portable codes.

This poster presents the level of performance to expect from an optimized MPI+(task-based OpenMP) software stack. Our methodology consists in modeling hybrid applications to a unified task graph scheduling problem on which some metrics are defined to analyze application performances. Proxy-applications are ported, analyzed, and improved through both user-code and runtime optimizations wherever it is the most suitable to preserve real-world code representativeness. Performance results show 1.9 speedup weak-scaled to 16,000 cores from our task-based over the for-loop parallel version on LULESH. We show the highest performances on fine dependant tasks of 100 us. in average with about 80% communication overlap.

### JLESC topic

**Primary author:** RPEREIRA, Romain PEREIRA

**Presenter:** RPEREIRA, Romain PEREIRA

**Session Classification:** Poster Session

**Track Classification:** Programming languages and runtimes