**Seismic Imaging**

Several techniques exist to supply subsurface images. Among them, those based on seismic waves impose, especially Reverse Time Migration (RTM) which focuses on the the interfaces detection, as illustrated on Fig. 1.

![Marmousi model (left) and RTM result (right)](image)

Fig. 1 - Marmousi model (left) and RTM result (right)

The accuracy level is directly related to the available computational power. Fig. 2 depicts this correlation where color blocks represent the different trends of seismic imaging techniques and the red curve is the power of the highest top500 supercomputer. RTM takes place on top right. In this strong HPC context, efficient and portable codes are mandatory.

Fig. 2 - Seismic Imaging (techniques) requirements in computational power and HPC evolution

**From MPI to task-based**

The space discretization is based on the Discontinuous Galerkin Method (DGM) coupled with a Leap-Frog time scheme, see e.g. [1]. This combination transforms the first-order elastic wave equation into an explicit scheme:

$$
\frac{v^{n+1} - v^n}{\Delta t} = \frac{\sigma^{n+1/2} \Delta \mathbf{M}_f_R}{\rho^{n+1/2} \Delta t^2} \mathbf{D}^{n+1/2} \mathbf{M}_f_R v^n
$$

where the unknowns are the velocity field $v$ and the stress tensor $\sigma$.

**Performance Portability**

The runtime system PaRSEC [3] provides a generic framework for microtask scheduling on distributed many-cores heterogeneous architectures, however, we first decided to address shared memory machines only.

Fig. 5 - Speedup by the use of PaRSEC (tasks) over MPI-based code

We extended single processor results on shared memory machines:
- a large cache-coherent Non-Uniform Memory Access (NUMA) node, Fig. 6 is a selected trace comparison;
- an Intel Many Integrated Core (MIC) architecture accelerator, Fig. 7 shows the parallel efficiency in native mode.

The PaRSEC code remains unchanged, see further results in [4].

Fig. 6 - Trace execution on 32 cores for 10 time steps, red and orange sections refer to computational kernels, light grey is the idle time, for MPI (left) and PaRSEC (right) codes, at the same time scale dark grey represents the gain

**Preliminary results**

Preliminary results highlight the code and performance portability of our task-based programming model on different shared memory architectures. Ongoing work tackles distributed memory architectures.

Fig. 7 - Parallel efficiency of PaRSEC and MPI-based codes on an Intel Xeon Phi co-processor

**Perspectives**

The adaptability of the proposed solution to novel architectures will continue to be evaluated with a short-term goal targeting Intel Knight Landing many-core processors, especially inside hybrid nodes.

**Bibliography**

[1] Boillot L., Contributions to the mathematical modeling and to the parallel algorithms for the optimization of an elastic wave propagator in anisotropic media, PhD thesis - Université de Pau, 2014

