Development of seismic tomography software for hybrid supercomputers

Alexandr Nikitin [1,2]*, Alexandr Serdyukov [1,2], Anton Duchkov [1,2]
(1) Trofimuk Institute of Petroleum Geology and Geophysics SB RAS
(2) Novosibirsk State University
*contact e-mail: NikitinAA@ipgg.sbras.ru

1. Research goals

Seismic tomography is a technique used for computing velocity model of geologic structure from first arrival travel times of seismic waves. This research deals with the following general scheme of travel time tomography:

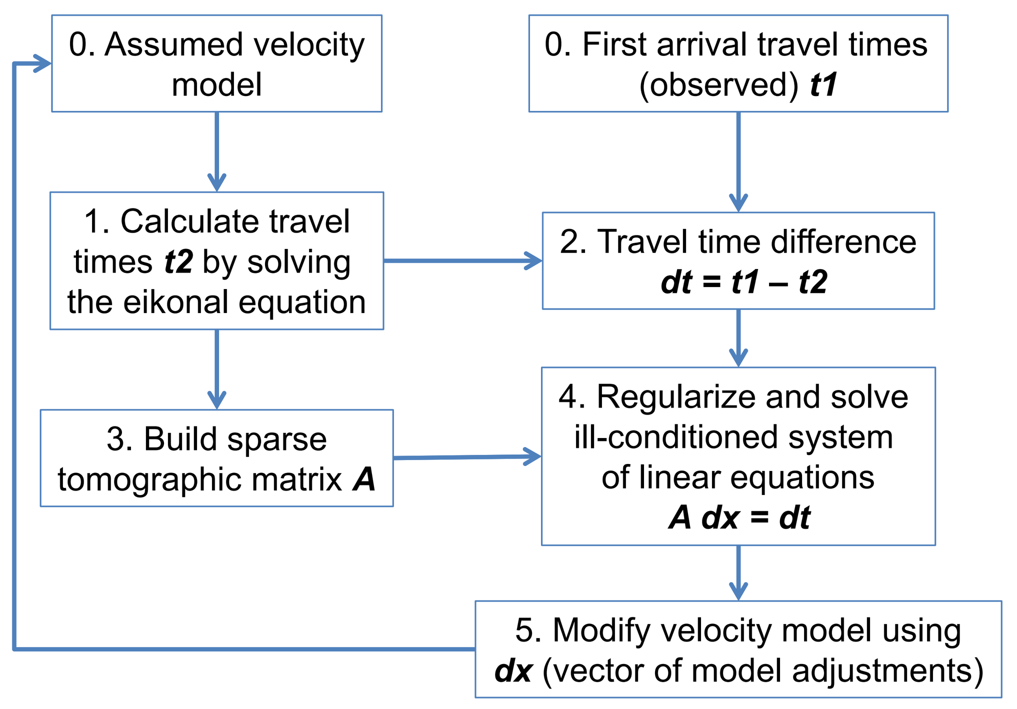

Fig. 1. General algorithm of seismic tomography

The main **purpose of this research** is the development of high performance open source 3D seismic tomography software package and related algorithms for modern supercomputers with hybrid architecture. This will allow us to significantly decrease computation time when using large numerical grids, thousands of sources/receivers and computationally intensive numerical methods with increased accuracy.

This project is in the early stages of development. The planned **project milestones** are:

By the end of 2015 – complete development, optimization and testing of the algorithms and software for execution on CPUs only

2016 – porting and optimization of the algorithms and software to support execution on NVIDIA Tesla and Intel Xeon Phi

2017 – final optimization and testing, possibly adding support for heterogeneous computation (simultaneous utilization of devices with different architectures: CPUs + Tesla/Xeon Phi)

2. The algorithm

The eikonal equation used in step 1 (Fig.1.) is given in the form:

$$|\nabla t| = f(x), x \in R^3$$
$$t(y) = 0, y \in R^3$$

where $t(x)$ is first arrival travel time from source located at point y to point x, $f(x)$ is the slowness at point x.

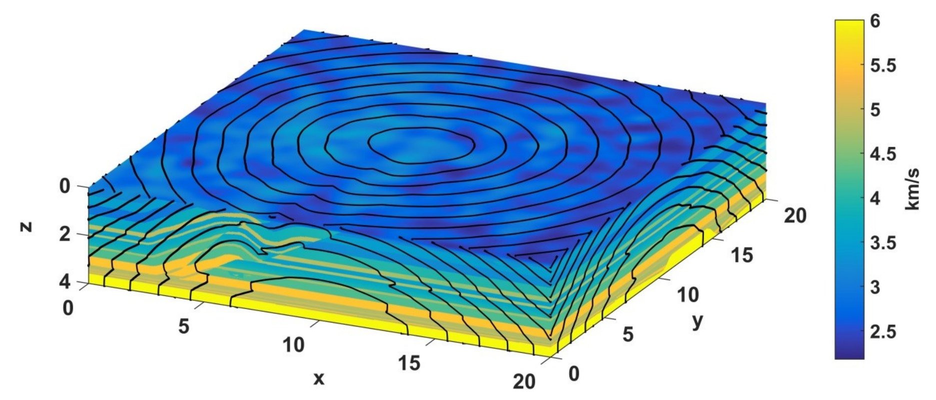

Fig. 2. The eikonal equation solution for SEG/EAGE Overthrust model

Calculating numerical solution of the eikonal equation is one of the most computationally intensive parts of the seismic tomography algorithm being used, since we have to obtain it for each source and receiver.

We use "fat rays" [Husen, Kissling, 2001] to build tomographic matrix A in step 3. Width of fat ray is given by $|t_{sx} + t_{rx} - t_{sr}| \leq T/2$, where t_{sx}, t_{rx} are travel times to point x from source s and receiver r, t_{sr} is the total travel time between s and r, T is the dominant period of wave. Tomographic matrix element for sr ray and k cell, where sum_k/vol_{sr} is the portion of the fat ray in k cell:

$$A_{srk} = \frac{sum_k}{vol_{sr}} t_{sr}$$

This step is computationally simple and easily parallelizable.

For regularization and solution of the system of linear equations in step 4 we utilize truncated Singular Value Decomposition:

$$dx_k = \sum_{i=1}^{k} \frac{u_i' dt}{\sigma_i} v_i$$

where k is the number of σ_i singular values, u_i and v_i are left and right singular vectors of matrix A. Currently, We use SVD solvers from SLEPc library [Hernandez et al., 2005] in this step.

3. Eikonal solver

In present work, we are using **Fast Sweeping Method (FSM)** [Zhao, 2005] as the numerical algorithm for our eikonal solver.

Fast Sweeping Method is an iterative method that uses Godunov upwind difference scheme to discretize partial derivatives and Gauss-Seidel iterations with alternating sweeping ordering.

The original algorithm is sequential: $t_{i,j,k}$ can be computed only after $\{t_{p,q,r} | p \in [0, i-1], q \in [0, j-1], r \in [0, k-1]\}$ have been updated (for sweep direction I=0:NI-1, J = 0:NJ-1, K = 0:NK-1). There are 8 total sweeping directions in 3D.

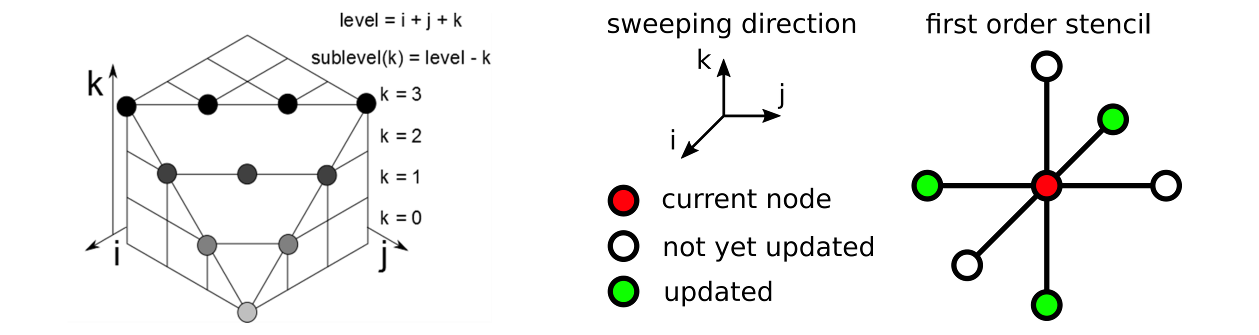

Fig. 3. Data dependencies in FSM (3D) and stencil. Only elements from the same level can be computed simultaneously.

Our current **MPI+OpenMP implementation** uses 2D data decomposition between processes, 1D decomposition into subtasks inside each process. Shadow elements are exchanged between neighbors. Values in subtasks are updated using OpenMP implementation of [Detrixhe et al., 2013].

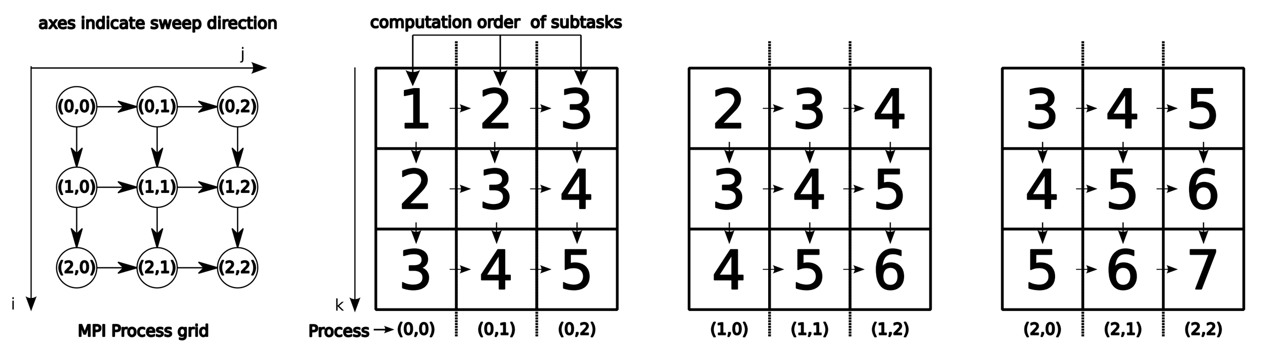

Fig. 4. MPI implementation of FSM. Communication topology and order of computation of subtasks. Communication is shown for forward exchange of shadow elements between processes.

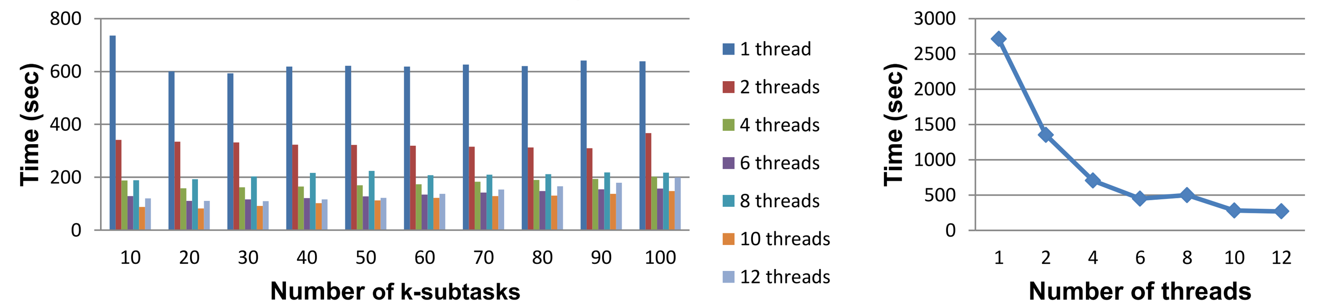

Fig. 5. Performance results for 1000^3 homogenous model (9 iterations). left) MPI+OpenMP 2x2 process grid; right) OpenMP on CPUs. Testing was done on NUSC Cluster (2 x Xeon X5670 per node).

4. Xeon Phi Performance

We investigated what kind of cross platform performance we can get out of our FSM OpenMP code on Intel Xeon Phi in native mode without making any changes to it.

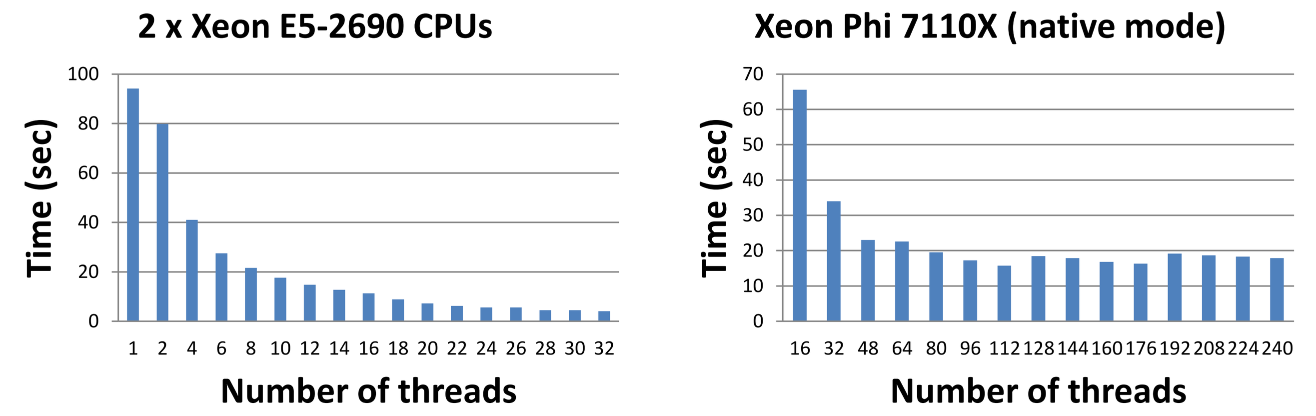

Fig. 6. Xeon Phi 7110X vs 2 x Xeon E5-2690 OpenMP performance comparison. Testing was done on JSCC RAS MVS-10P cluster, using 500^3 homogenous model.

5. Conclusions

As seen from the results, FSM OpenMP implementation with good CPU efficiency fails to achieve sufficient speedup on Xeon Phi without optimization specifically for this architecture. This is most likely due to the poor vectorization of the code, but further testing is required to confirm this. We are planning to investigate this problem further in the future, and optimize our implementation of FSM for Xeon Phi if possible.

6. References

1. Husen, Stephan, and Edi Kissling. "Local earthquake tomography between rays and waves: fat ray tomography." Physics of the earth and Planetary Interiors 123.2 (2001): 127-147.
2. Hernandez, Vicente, Jose E. Roman, and Vicente Vidal. "SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems." *ACM Transactions on Mathematical Software (TOMS)* 31.3 (2005): 351-362.
3. Zhao, Hongkai. "A fast sweeping method for eikonal equations." *Mathematics of computation* 74.250 (2005): 603-627.
4. Detrixhe, Miles, Frédéric Gibou, and Chohong Min. "A parallel fast sweeping method for the Eikonal equation." Journal of Computational Physics 237 (2013): 46-55.

Financial support information
This work is partially supported by RFBR grant 15-05-06752 and BP research grants for young scientists.