Updated April 15, 2014

Tesseral 3D-3C seismic modeling on CPU/GPU clusters

Client Guide

Tesseral Technologies Inc. proposes a comprehensive set of tools to generate 3D-3C synthetics in affordable time on modern CPU/GPU clusters. The set includes 3D acoustic and elastic methods, 2.5D anisotropic visco-elastic method and Haskell-Thomson method for horizontally layered media.

- ✓ We can help you in estimation of minimal hardware configuration which fit to your sample model and survey
- ✓ We can help you in estimation whether an available hardware configuration fits to your sample model, survey and performance requirements

Contents

3D-3C Modeling Options

The *Tesseral parallel Computational Engine* allow generating (X-,Y- areal) synthetic 3C gathers for the given 3D seismic properties model and 3D survey geometry. It may be:

- 3D wavefield propagation in horizontally-layered medium:
  - 1D-3C visco-elastic TTI anisotropic (including more simple cases) Haskell-Thompson analytical method
- 3D wavefield propagation in a model where changes of physical properties along structural strike can be neglected:
  - 2.5D visco-elastic TTI anisotropic method for the most complex cases of anisotropy and fracturing (including more simple cases)
- 3D wavefield propagation in true 3D medium:
  - 3D-3C full-wave (finite-difference) acoustic method
  - 3D-3C full-wave elastic method
  - 3D-3C full-wave VTI anisotropic elastic method (release planned in Q4 2014)

Equipment Options

Tesseral Pro includes parallel 3D-3C computations on workstation for testing on small low-frequency samples. It also contains basic 3D model builder capabilities which allow describing a model by a set of horizons with constant velocities between them. We started development of the Tesseral 3D model builder supporting thin layers calculated from well logs. In more complex cases it may be needed to prepare (seismic) properties (SEG-Y) models in 3rd party software.

Tesseral Linux cluster parallel computation engine is intended to run practical 3D modeling tasks which are usually too big for a single workstation or even server equipped with several GPUs. GPUs are more efficient for such programs. It is estimated that using of GPU's allows achieving 30 or more better ratio per computation-equipment price. So generally we suggest using GPU clusters for such computations.

All the programs can run on either multi-core clusters or GPU clusters based on NVIDIA CUDA compatible GPUs. The typical bottleneck for the 3D modeling jobs is memory (RAM) size per computing device (which can be either a cluster node or a GPU).

For initial estimation of the cluster memory (nodes and/or GPUs) requirements can be taken following example

- ✓ 2D model of practical size (5000 m receiver far offsets and 5000 m depth) and modeled signal frequency (30-60 Hz) can have up to 3000x3000 (X-,Z-) node grid. In Y-direction let estimate 3000 grid samples (nodes) – overall 9 GBx40 (bytes per grid node for elastic modeling) = 270 GB of computing memory. If we assume that the cluster core has 12 GB of memory/node – ~24 nodes may be needed. For power GPU can be assumed 6 GB/GPU – ~34 GPU's. For 3D acoustic modeling such requirements are 1.5 times less.
- ✓ To achieve good quality results for 3D elastic modeling about a day per shot computation time may be needed. Computation time for 3D acoustic modeling is about 6 times less.
- ✓ Ideally, the best performance can be achieved by a "supercomputer-like" cluster which contains either all required (by computing memory) GPUs on one node or all required memory on one node along with some number of GPU's.
- ✓ Really, a cluster with 32 GB of memory per node for all (8-16) cores and with 32 GB of memory for all (4) GPU's allocated to a node can be considered as a (potentially) high productivity equipment.
- ✓ Generally the more nodes and GPUs per node the cluster has the better performance can be achieved
- ✓ Usually multiple nodes of a cluster participate in simulation of a single 3D-3C wavefield. That is why the interconnect performance (both bandwidth and latency) becomes a priority. So in practice it should be a scalable high-performance system.
- ✓ The number of nodes needed for execution of the particular 3D-3C modeling project also depends on their real performance (measured performance of the system during the 3D-3C modeling tests)

To summarize, our typical demands include:

- NVIDIA CUDA compatible GPUs with at least 4 GB on-board memory (e.g. general purpose high-end GeForce GXT Titan Black videocard has 6 GB and cost ~1000 USD, computing purpose high-end Testla K20 has 8 GB and price ~8000 USD).
- Infiniband interconnect (FDR or at least QDR)
- Shared network storage 100GB
- Software: Open MPI v.1.6.5

- Other possibilities can be examined too

The Tesseral 3D-3C modeling programs also allow:

- ✓ using a cluster without GPUs. In this case it is important to have big amount of memory per nodes or extra quality interconnect and modern CPUs on the cluster
- ✓ generating first arrivals and maximum energy 3D timefields, e.g for using in 3D (Vector) Wave Equation depth migration programs
- ✓ produce several 3D wavefield snapshots as SEG-Y cubes
- ✓ support sources of different types, like exploding point and surface, and of different modes: omnidirectional, directed monopole, dipole and double couple

Parallel Computational Options

The simulation of multiple wafefields by multiple CPUs can be parallelized (both on workstation and on Linux cluster and for each of 2D, 2.5D, 3D modeling cases) in two ways:

1. At first, the program can simulate each wavefield by its own CPU.
2. At second, the program can use multiple CPUs together to simulate single wavefield.

Option 1

- is preferable when the number of the wavefilelds (usually equal to the number of shotpoints) is bigger than the number of CPUs
- another reason for using option is better visual control over the computation process. Thus the "Run/Run 2D modeling" command uses this mode
- is recommended for 2.5D-3C method in all cases except for single shotpoint (or exploding horizons, or exploding surface) with cylindrical source. This is because 2.5D computations are split into a big number of quasy-2D sub-problems running in parallel
- 3D-3C method is usually memory limited and so it's always better to specify maximum number of processes per shotpoint regardless of the survey

Option 2

- Is preferable if one simulates less wavefilelds than the number of CPUs In this case some CPUs will be idle. This is the reason to apply multiple CPUs together
- Allows to decrease big synchronization overheads (relatively to option 1) at the end of each propagation time step
- The efficient mode "one CPU per wavefield" is implemented within "Run/NET: Run modeling" when 1 process per shotpoint has been selected