In this tutorial let us use `topIIvol_ParMesher` to create a volume mesh from a point cloud cluster `./xyz/point-cloud-fine.xyz` which contains $(x \times y) = 500 \times 451 = 225500$ points. `topIIvol_ParMesher` is a parallel computing tool, it will takes in a point-cloud as an input (`.xyz`) and generates volumetric meshes that can be extracted in medit's `*.mesh` format. Let us say we would like to create the volumetric mesh `out-mesh.mesh`, with volume stretching upto a depth (z) of -2000 and z direction should be meshed with 25 layers. Let us accomplish this task using 2 parallel processes (2 MPI ranks). To generate the mesh

```
mpirun -n 2 topIIvol_ParMesher --xpoints 500 --ypoints 451 --zpoints 25 \
--depth -2000 --in ./xyz/point-cloud-fine.xyz --out Parallel-out-mesh.mesh
```

The output mesh `Parallel-out-mesh.mesh` should contain $32,335,200$ tetrahedra, $989,304$ surface triangles, and $5,637,500$ verticies.

Note that as we used `-n 2`, i.e, we used two parallel processes to generate the mesh. To gauge the importance of parallel processing rerun the mesher with only one process (as if it were a sequential mesher)

```
mpirun -n 1 topIIvol_ParMesher --xpoints 500 --ypoints 451 --zpoints 25 \
--depth -2000 --in ./xyz/point-cloud-fine.xyz --out Parallel-out-mesh.mesh
```

Comparing the CPU time statistics that appear on the terminal screen you should be able to see a considerable speed up when comparing the CPU timings of the two cases. For instance, benchmark on an Intel(R) Core(TM) i7-5600U CPU @ 2.60GHz produces the following results

```
#Single process  -n 1
   - point cloud partitioning took : 0.007211  sec
   - program finished in           : 30.806807 sec
#Two processes   -n 2
   - point cloud partitioning took : 0.004244  sec
   - program finished in           : 16.011780 sec
```

Notice that the CPU time has approximately halved as we used two processes.

To report bugs, issues, feature-requests contact:*

- **mohd-afeef.badri@cea.fr**
- mohd-afeef.badri@hotmail.com