Contents

Tesseral LC – Installation, Licensing, Running

- From company's website http://www.tesseral-geo.com **you can download**:
- ✓ *… latest versions of the Tesseral Products using* ***page /Download.*** You can also download ***Samples Database*** containing a *set of prebuilt models, templates and data samples which can be useful at initial learning and application of the package*.
- ✓ ***User Documentation*** *using page* ***/Support***.
- ✓ For ***other variants of Tesseral products*** please refer to corresponding product tab at http://www.tesseral-geo.com/support.en.php.
- ✓ Due to onrush of technology sharply increasing multi-core CPU and GPU-computing power and volume of operative and disk memory, PC's have reached productivity of high-power workstations at incomparably lower cost. The UNIX-like operation systems earlier functioning only on servers and workstations, now are accessible for PC running under Linux-system.

- ✓ The seismic modeling earlier accessible only in the simplified mode (ray-tracing schemes, the scalar wave equation) is also thriving in parallel with growth of computing facilities power. It has reached incomparably greater accuracy of calculations owing to use of more complete approximations of the wave equation – acoustic, elastic, elastic with anisotropy. However volumes of the calculations demanded for realization of more complex approximations of the wave equation, sharply grow with increase of quantity of the parameters used for greater adequacy of seismic model to a real geologic medium.

GPU modeling procedures

- ✓ In connection with the progress of high-performance computing, especially on a general-purpose or cluster graphics cards (GPU) is provided for 2D-2C and 2.5D-3C modeling. In case of modeling 3D-3C wave fields it becomes possible to run modeling computations for complexly built geologic media in a reasonable time span with 25-100-more-fold acceleration relating to 1-core performance:**Tesseral LC** includes the *parallel computations engine*, which can be installed on *Linux cluster* nodes.

Tesseral LC is using the cluster's standard software (any *MPI* realization), it can be freeware *MPICH, LAM MPI, MVAPICH, SCALI MPI*, etc. Usually, *computing modules* running under *Tesseral Linux* are compiled for functioning in such environments.

Tesseral LINUX cluster computational engines are provided in *http://www.tesseral-geo.com/accept.en.php*, browse to tab *tab /Farm and Cluster* and select corresponding computational **Tesseral LC *2D, 2.5D and 3D*** engine for download.

Tesseral LC Linux Cluster Computation Engines

- ✓ ***Tesseral Workstation (Pro or 2D)*** is needed to properly prepare task jobs for Linux Cluster computations. After calculations are finished, resulting data can be transferred back for subsequent processing.

Parallel computations engines ***Tesseral LC (2D, 2.5D and 3D)*** under Linux operational system allow to increase the speed of calculations in proportion to the quantity of processors/GPUs in the cluster. LINUX multiprocessor systems based on standard processors may be used to accelerate huge calculation volumes needed for full-wave modeling and data processing.

Tesseral LCs are designed to run on a 32-bit or 64-bit Cluster architecture, running a number of Linux versions. It allows to implement clusters for high-performance parallel full-wave modeling on Linux Clusters. The data to be processed must be either put in a shared directory accessible to each node or copied to each the node so to have the same pathname.

General Requirements

Software

- RedHat/CentOS Enterprise Linux of version 3 or higher (either 32 bit or 64 bit)
- Shared disk storage (NFS, SAMBA, etc.) accessible via the same path under every node
- (optional) Compilers: GNU C++ (g++) of version 3.2.3 or higher or Intel C++ Compiler (icpc) of version 9.1 or higher (to build from library)
- (optional) An MPI implementation (to build from library). We tested the software mostly on LAM MPI and installation and setup instruction of LAM MPI is described in this manual, but we suggest the software will run on most MPI implementations (MPICH, OpenMPI, Scali, etc.)

Licensing

to Register Tesseral LINUX cluster computational engines

Description of Licensing with FlexLM License Server for Tesseral LINUX cluster computational engines is provided in http://www.tesseral-geo.com/accept.en.php, browse to tab /Download/Drivers & Utilities and select corresponding User Guide. To download Tesseral LINUX cluster computational 2D, 2.5D and 3D engines, you can use tab /Farm and Cluster.

Installation steps

- First, untar the downloaded archive package:

```
tar -xzvf ./tesseral-lc-x.x.x.zip
```

- Tesseral LCs are distributed as a binaries in 'precompiled' folder:
  precompiled_ompi_1_10/ - binaries for running using Open MPI v. 1.10.3
  precompiled_ompi_2_1_1/ - binaries for running using Open MPI v. 2.1.1
  precompiled_legacy/ - binaries for running using LAM MPI or MPICH
  Tesseral-3D (or 25D or 2D)-CUDA.lexe - supports CPU and GPU cards with CUDA technology.
  Tesseral-3D (or 25D or 2D)-CPU.lexe - CPU-only binary
  libcudart.so.4 - CUDA library that must be in the same directory
  as 'Tesseral-nD-*.lexe' binary in order to run
  on computers with GPU cards.

  To use the binary Open MPI version 1.10.3 or 2.1.1 is required.
  The support of Open MPI version 3.1.2 is experimental.
  You can download it on Open MPI project site:
  http://http://www.open-mpi.org/

Precompiled binaries

1. To install the precompiled binaries select an RPM package that best suits your machine's configuration and run:

```
rpm -ivh <RPM-file>
```

For example,

```
rpm -ivh  tesseral-lc-rh3-em64t.rpm
```

The program will be installed into ***/opt/tesseral/rhXxXX/*** directory (XxXX is your architecture's abbreviation, e.g. rh3x64).

2. Create a directory for installing the ***tesseral2d-32*** or ***tesseral2d-64 (or 2.5d or 3d)*** executable. You can do this in two different ways:

   a. Create a directory, e.g. ***tesseral***, on one of the cluster nodes and make it directly accessible to every node of the cluster (e.g. with NFS).

   b. Create directories with the same name, e.g. ***tesseral***, on every node of the cluster. You can use the **rsync** Linux utility to do this. You can also use ***rsync*** to make copies of the data to be processed (see the next section).

3. Copy the ***tesseral2d-32*** or ***tesseral2d-64*** executable file created at the step 2 to the directory (-ies) created at the step 3.

Preparing data

1. Prepare the data by invoking the menu command 'Run/CLUSTER: Create task' in either Tesseral 2D or Tesseral Pro desktop Windows program. In the appeared dialog adjust all necessary options (see your desktop program *Help* for details) and choose directory for storing the data that you'll have to copy to the cluster.
2. The command prepares a runtask.ini file and several more files. **The runtask.ini file has a simple structure described in Appendix B and can be edited manually if needed**. To edit it use any text editor and save the result as a plain ASCII text file.
3. Put the files created in step 1 onto every cluster node in the directory where the executable resides (e.g. the **tesseral** directory). Apart to the **runtask.ini**, the data files are as follows:
   - for modeling: a *Tesseral's TAM format <modelname>**.tam*** file with the polygon model and optionally a grid model file in Tesseral's TGR or SEG-Y; optionally up to three *SEG-Y* files of the TTI anisotropy parameters '*epsilon', 'delta' and axis angle 'phi' (ε, δ, φ).*
   - for migration: a grid model file in *TGR or SEG-Y format*, a seismogram file in *TGR* or SEG-Y format and a Tesseral's *BAS file* with the migration aperture description; optionally a time file created at the modeling step in *TGR* format.

Running

1. Change Linux current directory to the one containing the file **runtask.ini**.
2. Load Tesseral LC environment variables and inititlize LAM MPI:

```
source /opt/tesseral/rhXxXX/tessvars.sh
```

This command should be run under common (not root) user account.

3. Run the ***Tesseral.exe*** (for 2D) or ***Tesseral25D.exe*** (for 2.5D) (-xx bit extension may be present in the name) executable in the MPI environment. Usually, you must be signed in as an ordinary user, not as a root. The sample command is

mpirun -np *<a number of processors to involve>* ***Tesseral<var>.exe***

(see your MPI documentation for details).

To run modelling on all processors of the cluster run:

```
mpirun C <path>/tesseral2d-64
```

(this works only under LAM MPI environment that is shipped with Tesseral LC)

4. If you run the program for the first time it generates a license request and stops. You have to obtain and install a license file (please see chapter *Licensing*).
5. After the program has finished its parallel calculations the resulting files are stored in the current directory.

Collecting Results

1. (2D and 2.5D) After the program has finished its parallel calculations the resulting files are stored in the current directory. Three partial velocity components (Vx, Vy (2.5D) and Vz) as well as normal stress

Uc are stored as seismograms in SEG-Y format. Snapshots for each shot point are stored as Tesseral TGR files to be viewed and played in Tesseral<*var*> Windows Workplace.

2. (2.5D) After calculation of a model a big volume of temporary files may be stored. They are K2 sections (K2 is a space frequency along Y) of both shotgathers and snapshots stored in a new directory called like the model file but without the extension '**.tgr**'. The program re-uses these files on small changes of the observation system like other Y spacing of the receiver lines or another snapshot offset. If the initial model observation calculated snapshots, you can change the snapshot parameter 'Every' not re-computing the K2 sections. Else remove them.

Appendix A. Job Specification Format

The Tesseral.exe program execution is controlled by the job passport. The job passport is a text file **runask.ini** located in the current work directory.

The job passport consists of two sections. First section is started from the string '[TASK]', second one is started after '[modeling]'. Section name is put here in square brackets. Each section contains a set of parameters in format 'name=value'. A passport sample is shown below:

[TASK]

taskType=modeling

[modeling]

Model Name=Model.tam

First Point=31

Last Point=35

Run Computation=Elastic Anisotropic

Raster Model=model_P-0.tgr

Anisotropy Epsilon File=

The passport parameters values are supposed to be changed but the parameters and section names are fixed and case sensitive words. The parameter sequence in the bounds of each section does not meter.

taskType is a parameter determining the job type. Its possible values are 'modeling' and 'migration'. For the modeling tasks first variant is required.

Model Name is a pathname to the model file in the Tesseral's TAM format. It is an obligatory parameter. You can specify either relative or absolute path to the file. Automatically generated passports always contain local paths of a file name only as all necessary files are copied in the same directory.

First Point and **Last Point** specify diapason of shots to process. So you can narrow the diapason defined in the model. But you can't expand it as the program calculates shotgathers and snapshots for intersection of these two diapasons. These parameters are optional and can be omitted.

Run Computation is an obligatory parameter with a fixed set of possible values: 'Elastic Anisotropic', 'Elastic', 'Acoustic', 'Scalar', and 'Vertical Incidence'. The values are case sensitive.

Raster Model is an optional path to the main grid (raster) model file in Tesseral's TGR or SEG-Y format. SEG-Y file has to contain compression wave velocities in m/sec. TGR can contain up to 3 useful components: compression wave velocity, shear wave velocity and density. In any case the program interpolate missing components according to built-in dependencies.

Anisotropy Epsilon File, **Anisotropy Delta File** and **Anisotropy PhiMedia File** are optional file paths to

the anisotropy components in Tesseral's TGR or SEG-Y format. In the case of TGR each correspondent component should persists in the file. You can use the same TGR file for both primary and anisotropy model parameters, but all the 4 or 6 parameters should be included in it as components. Take into account that anisotropy parameters are ignored in any sort of simulation except for the 'Elastic Anisotropic' one.

MultiTamFiles is an optional parameter with the default value of 0. If its value is non-zero different models are used for different sources. This parameter is included to support complex jobs of Tesseral Pro and is not recommended for manual editing.

Appendix B. How to Install the Execution Environment

Tesseral for Linux cluster may be run on most HPC architectures with MPI enabled. Environment installation is required only if the cluster don't have one of the required software components. The components are:

– C++ Compiler (GNU C++, Intel C++, etc.)

– Shared storage (NFS, SAMBA, GFS, Lustra, etc.)

– MPI (LAM MPI, OpenMPI, MPICH, etc.)

– (optional) Resource Manager (OpenPBS, Torque, Slurm, etc.)

Cluster can have various configurations. To be more specific we'll use node imaginary names: `Node1 – Node8` and a system user, responsible to run the computations: "`tesuser`". In the guide we provide raw list of the commands and some of the output. These sections have grey background. The commands that you need to enter by hand are marked by bold font.

Configure the execution environment

You can check wherther LAM MPI is available on the node by running the next command:

```
[root@node1 /]# laminfo
            LAM/MPI: 7.1.4
             Prefix: /usr/local
       Architecture: i686-pc-linux-gnu
...
            SSI rpi: sysv (API v1.0, Module v7.1)
            SSI rpi: tcp (API v1.0, Module v7.1)
            SSI rpi: usysv (API v1.0, Module v7.1)
             SSI cr: self (API v1.0, Module v1.0)
[root@node1 /]#
```

To configure LAM you need to modify the file **/opt/tesseral/rh3x32/etc/lam-bhost.def**[1]. It is your default hostfile used when you run the program. You need to configure the file only on the node from which you want to run computations. An example of **lam-bhost.def** for a 8 node cluster of 2 CPU per node is shown below.

[1] Note. Depending on your Linux version and hardware your lam directory name may vary (e.g. rh3x64, rh4x32, etc.)

node1 cpu=2

node2 cpu=2

node3 cpu=2

node4 cpu=2

node5 cpu=2

node6 cpu=2

node7 cpu=2

node8 cpu=2

To check the configuration run **`lamboot`** and make sure the output is ok. You must run this command by non-root user account:

[tesuser@node1 data]# **lamboot**

LAM 7.1.4/MPI 2 C++/ROMIO - Indiana University

[tesuser@node1 data]#

If there are error messages make sure you have rsh-service available on all cluster nodes and lamboot executable is located in one of the PATH directory:

[tesuser@node1 data]# rsh node2 which lamboot

connect to address 192.168.1.2: Connection refused

Trying krb4 rsh...

connect to address 192.168.1.2: Connection refused

trying normal rsh (/usr/bin/rsh)

/usr/bin/lamboot

[tesuser@node1 data]#

How to install GNU C++ proper version if needed

Installation of GNU C++ is required only of you have version older then 3.2.3. If you have GNU C++ version 3.2.3 and higher, you can use it and skip this chapter.

Installation of GNU C++ is required only on one computer.

Releases of most versions of GNU C++ can be obtained on ftp://ftp.gwdg.de/pub/misc/gcc/. We recommend version of 3.4.6 if you have gcc older then 3.2.3.

We recommend you to install gcc not to default directory to save your current version of gcc. To install the gcc use the following procedure:

[root@node1 gcc]# tar -xvzf gcc-3.4.6.tar.gz

...

[root@node1 gcc]# mkdir gcc-bin

[root@node1 gcc]# **cd gcc-bin**

```
[root@node1 gcc-bin]# ../gcc-3.4.6/configure \
  --prefix=/usr/local/gcc-3.4.6 --program-suffix=346 \
  --enable-shared --enable-languages=c,c++,f77,objc \
  --enable-version-specific-runtime-libs
...
[root@node1 gcc-bin]# make
...
[root@node1 gcc-bin]# make install
...
[root@node1 gcc-bin]#
```

To compile the Tesseral you may need to add the next line to **`/etc/profile`** :

PATH=$PATH:/usr/local/gcc346/bin

and change the compilation script a bit. If you installed gcc with 346 suffix you may need to change **`make*.sh`** script from Tesseral installation line from

LAMMPICXX=g++

to

LAMMPICXX=g++346

How to setup rsh service

Edit **`.rhosts`** file in tesuser's directory (**/home/tesuser/**). Specify the nodes which have rsh access to current node. .rhosts file should look like this:

node1 tesuser

node2 tesuser

node3 tesuser

node4 tesuser

node5 tesuser

node6 tesuser

node7 tesuser

node8 tesuser

Make sure you have rsh is listed in **`/etc/securetty`** file.

How to setup NFS service

To run Tesseral modeling you need a shared storage available. You can use Lustra, GFS, SAMBA, NFS, etc. In this guide we describe how to setup NFS with `Node1` as server and other nodes as clients.

If you don't have any shared storage installed select a node to be NFS server (e.g. `node1`), create a directory to be shared and specify in **`/etc/exports`** the text parameter value:

/data *(rw,sync)

Run the next command to apply the settings.

[root@node1 /]# chkconfig –-levels=35 nfs on

[root@node1 /]# service nfs restart

...

[root@node1 /]#

To setup a NFS client enter the next line in **/etc/fstab** file on each client node:

node1:/data /data nfs rw,rsize=8192,wsize=8192

To establish the shared folder connection use the command:

[root@node1 /]# **mount /data**