ASCOMP
MULTIPHASE FLOW SCIENCE AND TECHNOLOGY

TransAT

Manual Series

Installation Manual on Linux - 64bit

Version 5.3.1.3

© September 2017 ASCOMP

Contents

Chapter 1

System Requirements

The following allied software is required for using TransAT on Linux:

- 64 bit machine and operating system.
- Python 2.6.6 or a newer version of Python 2
- Packages from Linux distribution:
  - gcc and g++ (4.4.6 or newer)
  - gfortran
  - GNU make
  - libstdc++6
  - libstdc++6-devel or libstdc++6-dev (depending on package manager)
  - Packages for exporting images: libpng12, libtiff
  - Recommended: gnuplot, gzip
- openMPI 1.5.5
- PETSc-3.2-p7 (including HYPRE package)
- ParaView or Tecplot for post-processing

Please note: Instructions to install openMPI 1.5.5 and PETSc-3.2-p7 are given in the present document.

Please note: The libstdc++6-devel or libstdc++6-dev package names might be slightly different depending on the package manager. In some package managers, an extra version number is included in the package's name e.g. libstdc++6-4.7-dev.

Chapter 2

Hardware Requirements

The hardware requirements in terms of memory and disk space depend on the problem at hand. The memory and disk space requirements increase with the number of cells in a given simulation. Likewise, they also increase with the number of activated equations and/or models in the simulation.

To illustrate this, the hardware recommendations for a simulation with a one-million-cell grid solving only for pressure and velocities are the following:

- RAM: 2 GB
- Disk space:

  per restart file: 1 GB

  per output file: ParaView 50 MB, Tecplot 100 MB

whereas the hardware recommendations for a simulation with a one-million-cell grid solving for pressure and velocities with turbulence, multiphase and phase change models activated are the following:

- RAM: 4.5 GB
- Disk space:

  per restart file: 2 GB

  per output file: ParaView 100 MB, Tecplot 150 MB

Chapter 3

Installing openMPI and PETSc

3.1 Downloading openMPI and PETSc

To run simulations with TransAT, openMPI and PETSc need to be installed. The following versions need to be installed:

- openMPI-1.5.5 from http://www.open-mpi.org
- PETSc-3.2p7 or PETSc-lite-3.2p7 from http://www.mcs.anl.gov/petsc

Please note: Do not copy and paste the commands in the Sections below into the command line.

3.2 Extracting openMPI and PETSc archives

The downloaded files are .tar.gz files that need to be extracted.

- Open a terminal and go to the directory where you have downloaded the main archive (e.g. /home/yourname/software)
- Execute the following commands to extract the archives

```
tar xzvf openmpi-1.5.5.tar.gz
tar xzvf petsc-3.2-p7.tar.gz
```

Listing 3.1: extract archives (64-bit example)

3.3 Building openMPI-1.5.5

To build openMPI using the GNU Fortran compiler ("gfortran"),

- Open a terminal and go to the directory where you have extracted the openMPI-1.5.5 archive
- Set the FC and F77 environment variables to gfortran then run the configuration script as follows:

```
export FC=gfortran
export F77=gfortran
./configure --enable-static=yes
            --enable-shared=no
            --prefix=/home/yourname/software/openMPI
```

Listing 3.2: Build openMPI

Note that the path defined by "–prefix" is the installation directory into which openMPI should be installed.

- After the configuration step is done, build and install by executing the following commands:

```
make
make install
```

Listing 3.3: Build openMPI

3.4 Building PETSc-3.2

To build PETSC,

- Open a terminal and go to the directory where you have extracted the PETSc-3.2 archive
- Configure PETSc by executing the following commands

```
./configure --download-f-blas-lapack=yes
            --download-hypre=yes
            --with-debugging=no
            --with-shared-libraries=no
            --with-dynamic-loading=no
            --with-mpi-dir=/home/yourname/software/openMPI
            --prefix=/home/yourname/software/PETSc
```

Listing 3.4: Build PETSc

Note that the path defined by "–prefix" is the installation directory into which PETSc should be installed and the path defined by "–with-mpi-dir" is the path to the openMPI installation directory.

- After the configuration step, follow the instructions of PETSc to build and install PETSc. The commands should be similar to the following:

```
make PETSC_DIR=/home/yourname/software/petsc-3.2-p7
     PETSC_ARCH=arch-linux2-c-opt all
make PETSC_DIR=/home/yourname/software/petsc-3.2-p7
     PETSC_ARCH=arch-linux2-c-opt install
```

Listing 3.5: Build PETSc

3.5 Setting path and environment variables

Typically, path and environment variables can be set in one of the following files located in your home directory:

- **.bashrc**
  If your are using BASH. This script is executed when you open a BASH-terminal. Add the following lines to the file:

```
# openMPI
export PATH="/home/yourname/software/openMPI/bin:$PATH"

# PETSc
export PETSC_DIR="/home/yourname/software/PETSc"
```

Listing 3.6: .bashrc

  where */home/yourname/software* has to be replaced by the path to the directory where you have installed openMPI and PETSc

- **.cshrc**
  If your are using CSHELL. This script is executed when you open a CSHELL-terminal. Add the following lines to the file:

```
# openMPI
setenv PATH /home/yourname/software/openMPI/bin:$PATH

# PETSc
setenv PETSC_DIR /home/yourname/software/PETSc
```

Listing 3.7: .cshrc

where */home/yourname/software* has to be replaced by the path to the directory where you have installed openMPI and PETSc

To check if the path is set properly,

- Open a new terminal (to load the changes in .bashrc or .cshrc) and check if the executables are detected:

```
which mpicc
which mpic++
echo $PETSC_DIR
ls $PETSC_DIR
```

Listing 3.8: check if executables are found

The output should look like:

```
/home/yourname/software/openMPI/bin/mpicc
/home/yourname/software/openMPI/bin/mpic++
/home/yourname/software/PETSc
bin    conf    include    lib
```

Listing 3.9: output if executables are found

The path */home/yourname/software* refers to the directory where openMPI and PETSc have been installed.

Chapter 4

Installing TransAT 5.3.1.3

4.1 Downloading TransAT

The TransAT installation files can be downloaded at the following adress:
http://ascomp.ch/product-downloads/
Note that the software is only available on 64-bit platforms.

- If you do not know which kind of platform you have, execute the following command

```
uname –m
```

Listing 4.1: Check machine architecture (32-bit/64-bit)

If the output is **x86_64**, you have a 64-bit platform.

The Linux executables are stored in the **TransAT_5.3.1.3.tar.bz2** archive.

4.2 Extracting archives

To extract the TransAT archive

- Open a terminal and go to the directory where TransAT archive has beed downloaded (e.g. /home/yourname/software) and extract the archive:

```
tar xvf TransAT_5.3.1.3.tar.bz2
```

Listing 4.2: extract archives

This creates the "TransAT_5.3.1.3" directory containing the TransAT software.

4.3 Setting TransAT path and environment variables

Typically, path and environment variables can be set in one of the following files located in your home directory:

- **.bashrc**
  If your are using BASH, the **.bashrc** script is executed when a BASH-terminal is opened.
  - Add the following lines to the file to set the TransAT environment variables:

```
# transatMB #
export PATH="/home/yourname/software/TransAT_5.3.1.3/transatMB/bin:$PATH"

# transatUI #
export PATH="/home/yourname/software/TransAT_5.3.1.3/transatUI/bin:$PATH"

# transatPy #
export PATH="/home/yourname/software/TransAT_5.3.1.3/transatPy/bin:$PATH"
```

Listing 4.3: .bashrc

The */home/yourname/software* refers to the path to the directory where you have extracted the TransAT archive has been extracted.

- **.cshrc**
  If your are using CSHELL, the **.cshrc** script is executed when a CSHELL-terminal is opened.
  - Add the following lines to the file to set the TransAT environment variables:

```
# transatMB #
setenv PATH /home/yourname/software/TransAT_5.3.1.3/transatMB/bin:$PATH

# transatUI #
setenv PATH /home/yourname/software/TransAT_5.3.1.3/transatUI/bin:$PATH

# transatPy #
setenv PATH /home/yourname/software/TransAT_5.3.1.3/transatPy/bin:$PATH
```

Listing 4.4: .cshrc

The */home/yourname/software* refers to the path to the directory where you have extracted the TransAT archive has been extracted.

To check if the path has been properly set,

- Open a new terminal (to load the changes in .bashrc or .cshrc) and check if the executables are detected by executing the following commands:

```
which transatui.py
which tmb_init_compile.py
which tmb_runinit.py
which tmb_run.py
which tmb_link.py
which transatpython
```

Listing 4.5: check if executables are found

The output should look like:

```
/home/yourname/software/TransAT_5.3.1.3/transatUI/bin/transatui.py
/home/yourname/software/TransAT_5.3.1.3/transatMB/bin/tmb_init_compile.py
/home/yourname/software/TransAT_5.3.1.3/transatMB/bin/tmb_runinit.py
/home/yourname/software/TransAT_5.3.1.3/transatMB/bin/tmb_run.py
/home/yourname/software/TransAT_5.3.1.3/transatMB/bin/tmb_link.py
/home/yourname/software/TransAT_5.3.1.3/transatPy/bin/transatpython
```

Listing 4.6: output if executables are found

Again, note that */home/yourname/software* refers to the directory where the TransAT archive has been extracted.

4.4 Setup TransAT 5.3.1.3

TransAT solver has to be linked against local libraries.

- Execute the following commands to make sure the PETSC_DIR environment variable is set and the *bin* folder of TransAT is in the PATH variable.

```
which tmb_link.py
echo $PETSC_DIR
```

Listing 4.7: check if environment variables are set

The output should look like:

```
/home/yourname/software/TransAT_5.3.1.3/transatMB/bin/tmb_link.py
/home/yourname/software/petsc-3.x.x
```

Listing 4.8: check if environment variables are set

The script to create the executable is called **tmb_link.py** and it is in the *bin* folder in the transatMB directory. It links the TransAT libraries to the local MPI- and PETSc-library to produce the executable.

- Execute the following commands to create the *bin/transatmbDP* executable

```
tmb_link.py
```

Listing 4.9: link transatmb

If the executable has been successfully linked and created in the *bin* folder, the command

```
which transatmbDP
```

Listing 4.10: link transatmb

should output a path similar to the following:

```
/home/yourname/software/TransAT_5.3.1.3/transatMB/bin/transatmbDP
```

Listing 4.11: check if environment variables are set

Chapter 5

TransAT 5.3.1.3 License

TransAT has a built-in demo version which lets you run simulations for up to 1 hour wall time, using a maximum of 4 cores. No checkpoint/restart options are provided in this version.
A license can also be installed. This provides the full usage of TransAT, without restrictions on simulation runtime and checkpoint/restart capability. A restriction on the number of cores might remain and it depends on the contract.

5.1 Install license

The license file for TransAT will be sent separately by email. Save the license file (e.g. /home/-jerry/transat_jerry.lic).

The environment variable **TRANSATLIC** should point to the license file. The license file should have both READ and WRITE permissions for the user(s). Further modifications depend on the SHELL used. Which SHELL is used can be checked in the terminal using "echo":

```
echo $SHELL
/usr/bin/csh

echo $SHELL
/usr/bin/bash
```

Listing 5.1: Check SHELL being used

If the SHELL is csh, then set the environment variable TRANSATLIC as follows (you can add this line to .bashrc in your home directory to automatically set the variable):

```
setenv TRANSATLIC /home/jerry/transat_jerry.lic
```

Listing 5.2: Set license file path (csh)

If the SHELL is bash, then set the environment variable TRANSATLIC as follows (you can add this line to .bashrc in your home directory to automatically set the variable):

```
export TRANSATLIC="/home/jerry/transat_jerry.lic"
```

Listing 5.3: Set license file path (bash)

Please note: After TransAT and the license file are installed, TransAT has to be run at least once in the same month the license file was created in order to activate it.

Chapter 6

Running TransAT 5.3.1.3

Executables that run on distributed memory machines can not be run like normal executables on Linux. The parallel environment has to be set up and the number of processors involved has to be defined.

On small machines (desktops computer with multicore processors), the user can directly submit parallel jobs if openMPI is installed. For instance, to run TransAT executable "transatmbinitialDP" on 4 processors from the current folder the following command can be executed:

```
mpiexec -n 4 ./transatmbinitialDP
```

The **tmb_runinit.py** and **tmb_run.py** scripts from the transatMB/bin folder allow users to run small parallel jobs locally (see the next section 6.1).

On clusters, many users run simulations simultaneously and consequently there is a job system to manage parallel jobs. The job management system and the job submission process on specific clusters is information that is usually documented by the provider of the serivce. Section 6.2 shows which executables from TransAT should be submitted to the job system.

6.1 Running without job system

All the following commands should be executed from the project directory:

- Compiling initialconditions.cxx
  Use the script
  tmb_init_compile.py -n nprocessors
  to compile your initialconditions (initialconditions.cxx) in the project folder and link it to the code library. This will create the executable **transatmbinitialDP** inside the project directory and run it locally. The argument **-n nprocessors** defines the number of processors

used in the initialisation. If not given, the executable will not be run; it can be run later using **tmb_runinit.py**, described below.

- Initalizing the simulation
  Use the script
  tmb_runinit.py (-n nprocessors) to run the initialisation executable locally. The optional argument (-n nprocessors) defines the number of processors used in the simulation. If not given, the simulation will run on one processor. This is not needed if **tmb_runinit.py** has been used with the argument **-n nprocessors** (See previous point).

- Running the simulation
  Use the script
  tmb_run.py (-n nprocessors) to run the executable locally. The mandatory argument (nprocessors) defines the number of processors used in the simulation.

6.2 Running on a cluster with job system

All the following commands should be executed from the project directory, e.g. from your own project folder.

- Compiling initialconditions.cxx
  Use the script
  tmb_init_compile.py
  to compile your initialconditions (initialconditions.cxx) in the project folder and link it to the code library. This will create the executable **transatmbinitialDP** inside the project directory.

- Running initialconditions
  Use the executable **transatmbinitialDP** to submit your parallel job to the job system.

- Running the simulation
  Use the executable from TransAT directory. The full path to the executable is:
  /home/yourname/software/TransAT_5.3.1.3/transatMB/bin/transatmbDP
  (depending on where you installed TransAT, the path will differ)

6.3 Running a TransAT - OLGA coupled simulation

The commercial simulation software OLGA is not available for Linux, thus to run a TransAT-OLGA coupled simulation from Linux, a separate Windows machine is needed which must meet the following requirements:

- OLGA 7.2 (supported version) installed
- Python 2.7.3 (or newer version of Python 2) installed
- OpenOPC 1.2.0 package installed

6.4 Troubleshooting: modifying transatui.target

transatui.target is used by the Graphical User interface to locate the executables of TransAT. For a classical installation, it is not needed to modify this file.

If the TransAT executable is not found when running a simulation, the issue might come from incorrect setting of paths in the transatui.target file. The transatui.target file can be found at the following location:

/home/yourname/software/TransAT_5.3.1.3/transatUI/bin

where */home/yourname/software* refers to the path to the directory where TransAT has been installed.

The transatui.target file contains the paths to the the installation directories of the TransAT solver and to the executable of a default text editor used to modify initial conditions. The path to transatUI and transatMB installation directory can be either absolute or relative. The following is an example of the content of transatui.target.

```
[TransATUI]
Installation_Directory = ../
Bin = ./
[TransATMB]
Installation_Directory = ../../transatMB
[TextEditor]
Bin = /usr/bin/emacs
[PythonCmd]
Bin = ../../transatPy/bin/transatpython
```

The “Bin” variable in section ”TextEditor” defines the path to text editor used by TransATUI. Note that the path to the text editor executable has to be absolute. The text editor is the only

variable that should be changed by the user, all the other directories should be correct after the installation.

The “Bin” variable in the ”PythonCmd” section defines the python executable which TransAT uses to execute the python scripts.