Installing Firefly with OpneMPI (32-bit) libraries

by Pavlo Solntsev (pavlo.solntsev@gmail.com)

An idea of this tutorial is to provide easy to use and comprehensive instructions to setup Firefly on GNU/Linux OS (cluster, workstation, desktop computer laptop etc.). Since 64-bit OS's become more and more popular and Firefly is a 32-bit program yet, 32-bit libraries should be available to run Firefly not only in parallel but on single processor as well. This tutorial was tested on 64-bit Ubuntu 12.04 LTS (3.2.0-29-generic #46-Ubuntu SMP) but of course can be used for any GNU/Linux OS. The installation consist of the following steps:

1. Compilation and installation 32-bit libraries of OpenMPI

2. Creating appropriate settings withing script file to run Firefly

1. Installing dependencies for 32-bit libraries OpenMPI

To compile 32-bit libraries additional packages should be available. In Ubuntu 12.04 LTS they are:

g++-4.6-multilib
g++-multilib
gcc-4.6-multilib
gcc-multilib
gfortran-4.6-multilib
gfortran-multilib

Basically, these libraries provide 32-bit standard libraries to link 32-bit applications. You can check, if they are present or not (in an Debian based distribution) by command:

$ dpkg -l | grep multilib

An equivalent command should be used in case of RPM package manager:

rpm -qa | grep <package_name>

Unfortunately, name of the packages with 32-bit libraries may vary from distribution to distribution. Of course, all compiler (C, C++, Fortran90/95, Fortran 77) also should be available. First of all we need to download OpenMPI source code from www.open-mpi.org. For Firefly version 8 OpenMPI 1.4 should be used, while 1.2 versions of OpenMPI should be used for earlier versions of Firefly. In this tutorial openmpi-1.4.5 (http://www.open-mpi.org/software/ompi/v1.4/downloads/openmpi-1.4.5.tar.bz2) was used together with Firefly 8.. In a terminal we need to do the following actions:

```
# Unpack archive with source code
$ tar xvfj openmpi-1.4.5.tar.bz2
$ cd openmpi-1.4.5
# configure our Makefile. A value for --prefix variable can be different. It is a place where your
# program will be installed. Default /usr/local
# We need also to specify 32-bit compilation via additional flag "-m32" that should be passed
# to ALL compilers. f
$ ./configure --prefix=/opt/soft/openmpi-1.4.5 CFLAGS=-m32 CXXFLAGS=-m32 FFLAGS=-
m32 FCFLAGS=-m32
# Standard compilation
$ make
# Installation. Root rights or equivalents are needed.
$ sudo make install
```

In the presented example. OpenMPI is installed in **/opt/soft/openmpi-1.4.5**, but location can be easy changed via --prefix variable at ./configure stage.

2. Creating appropriate settings withing script file to run Firefly

To run Firefly, the best way (at least for me) is to make script. Script below is suitable to run Firefly on desktop computer or workstation (without task manager, such as PBS). Suppose Firefly is installed in folder **/home/user/progs/ff8** then script file to start Firefly looks (all contents of the script file highlighted as blue):

```
#!/bin/bash
# $1 - number of processors for parallel run
# $2 - input file without extension
# $3 - basis set file
export PROJNAME=$2
export FFINPUT=""
export FFOUTPUT=$PROJNAME.out
export WORKDIR=$PWD
if [ -f $PROJNAME.inp ]; then
  export FFINPUT=$PROJNAME.inp
elif [ -f $PROJNAME.INP ]; then
  export FFINPUT=$PROJNAME.INP
else
  echo "Can't find input file $PROJNAME.inp or $PROJNAME.INP in $PWD"
  exit
fi
```

Firelfy 8 with OpenMPI 1.4 (32-bit)

```
# Check for basis set file
export BASISF=$3
echo $3
if [ $# -eq 3 ];then
  if [ -f $3 ]; then
        BASISF=$3
  elif [ -f $PWD/$3 ]; then
BASISF=$PWD/$3
  else
        echo "Can't find basis file $3 in $PWD. Exit..."
        exit
  fi
fi
 # Check for number of processors on workstation
MAXCPUS=`cat /proc/cpuinfo | grep processor | wc -l`
echo "Maxcpu's = $MAXCPUS"
export NCPUS=$1
if [ $NCPUS -gt $MAXCPUS ]; then
echo "You requested CPU's more then you have installed"
echo "Number of CPU's will be $MAXCPUS"
NCPUS=$MAXCPUS
fi
# Firefly settings . You can change them according to your  ystem
export FF8_HOME=$HOME/progs/ff8x
export FF8_TEMP=$HOME/tmp/
if [ ! -d $FF8_TEMP ]; then
  mkdir -p $FF8_TEMP; if [ $? -ne 0 ]; then echo "Can't create $FF8_TEMP dirrectory" ; fi; exit
fi
# OpenMPI settings . See OpenMPI installation section..
export OMPI_HOME=/opt/soft/openmpi-1.4.5
export PATH=$OMPI_HOME/bin:$PATH
# If you have 64-bit libraries installed in your system it better to overwrite LD_LIBRARY_PATH
# as in line below
export LD_LIBRARY_PATH=$OMPI_HOME/lib
# We are ready to go. Make sure name of executable file is correct.
mpirun -np $NCPUS $FF8_HOME/firefly8.openmpi14.ex -f -r -p -stdext -ex $FF8_HOME -i
$WORKDIR/$FFINPUT -o $WORKDIR/$FFOUTPUT -b $BASISF -t $FF8_TEMP
# Cleaning scratch folder
rm -rf $FF8_TEMP
rm -rf $FF8_TEMP.*
exit
```

Firefly can also be placed into any directory being already in PATH variable, such as $HOME/bin, for instance or /usr/local/bin for system wide access. However, it strongly recommended to put the script in to a PATH variable mentioned above. After that Firefly can be started via command

ourscript <number_processors> <inputfile_w/o_extension> <basis_set_file>

Input file also should be in the same folder. Of course we need OpenMPI compiled version of Firefly. Depending on your need you can adjust script. It is just example for beginners. To run Firefly on clusters via PBS you need add appropriate PBS parameters to your file:

```
#!/bin/bash
#PBS -N <job_name>
#PBS -l nodes=1:ppn=4  # Number of nodes and cpu's
#PBS -l pmem=450mb  # memmory
#PBS -l walltime=48:00:00  # Time for job
#PBS -o $PBS_JOBID.o  # Standard output for PBS
#PBS -e $PBS_JOBID.e  # Standard error output for PBS
export OMPI_HOME=/opt/soft/openmpi-1.4.5   # change according to your system
export PATH=$OMPI_HOME/bin:$PATH
export LD_LIBRARY_PATH=$OMPI_HOME/lib
export FF8_HOME=$HOME/progs/ff8x  # change according to your system
export NCPU=`wc -l < $PBS_NODEFILE`
export WORKDIR=$PBS_O_WORKDIR
export TEMP_DIR=/scratch/$USER/$PBS_JOBID # change according to your settings
$OMPI_HOME/ mpirun -np $NCPU $FF8_HOME/firefly -f -r -p -stdext -ex $FF8_HOME -i
$WORKDIR/$FFINPUT -o $WORKDIR/$FFOUTPUT -b $BASISF -t $FF8_TEMP
rm -rf $TEMP_DIR
rm -rf $TEMP_DIR.*
exit
```

Based on this tutorial, you should be able to setup Firefly using another version of MPI.

Good luck with Firefly.

If you have any suggestions, questions or corrections fill free to write me e-mail (see above)