Installing `Chombo/GRChombo` on Ubuntu 18.04

Leonardo Werneck

May 2018

Contents

We have just started a fresh install of Ubuntu 18.04 and we will install `Chombo/GRChombo` from scratch on it. The only applications we have installed on Ubuntu prior to this are `Google Chrome` and `Texlive/Texmaker`.

1 GNU installation

1.1 Installation summary

The following are all the needed utilities to run Chombo/GRChombo:

- make (comes with Ubuntu).
- gcc/g++ (comes with Ubuntu).
- gfortran (sudo apt install gfortran).
- mpicc/mpi++ (sudo apt install mpich).
- zlib (sudo apt install zlib1g-dev).
- csh (sudo apt install csh).
- OpenMP (sudo apt install libomp-dev).
- PAPI (sudo apt install papi-tools).
- Perl (comes with Ubuntu).
- LBLAS and LAPACK (install with apt - see below).
- HDF5 (Downloaded v1.10.2 from https://portal.hdfgroup.org/display/support/Download+HDF5).

1.2 What comes with Ubuntu

The following applications are already installed with Ubuntu:

- make.
- gcc/g++.
- Perl.

1.3 Basic utilities

To install gfortran:

```
$ sudo apt install gfortran
```

To install `mpich`:

```
$ sudo apt install mpich
```

To install `zlib`:

```
$ sudo apt install zlib1g-dev
```

To install `csh`:

```
$ sudo apt install csh
```

To install `OpenMP`:

```
$ sudo apt install libomp-dev
```

To install `PAPI`:

```
$ sudo apt install papi-tools
```

1.4 Linear algebra packages

The next two packages we will install are `LBLAS` and `LAPACK`. To do so, we use the following commands:

```
$ sudo apt install libblas3
$ sudo apt install libblas-dev
$ sudo apt install liblapack3
$ sudo apt install liblapack-dev
$ sudo apt install liblapack-doc
```

1.5 Installing `HDF5`

The tarball was downloaded from the website given on page 2. We then do the following[1]:

```
$ tar xf hdf5-1.10.2.tar.gz
$ cd hdf5-1.10.2/
```

Now we need to setup the compilers so that `HDF5` may be configured properly. For this we break the instructions on two parts.

[1] You may need to adjust this to the version of `HDF5` downloaded.

1.5.1 Serial HDF5

We start by installing the serial version of the HDF5 libraries. To this end, we setup the compilers

```
$ export CC=gcc
$ export CXX=g++
$ export FC=gfortran
```

Then we run

```
$ ./configure --prefix=/usr/local/hdf5-1.10.2/serial --enable-fortran --enable-cxx
```

Keeping in mind we are installing on a PC with 4 threads, the next command we run is

```
$ make -j 4
```

Then (note: this next step may take some time to finish)

```
$ make -i check -j 4
```

Finally, because we have set the restricted installation path, we do

```
$ sudo make install
```

1.5.2 Parallel HDF5

This next part is very similar to the previous one. We start by performing a `make clean` just to be safe. Then we do

```
$ export CC=mpicc
$ export FC=mpifort
```

As before:

```
$ ./configure --prefix=/usr/local/hdf5-1.10.2/parallel --enable-fortran
                        --enable-parallel
```

The steps are then exactly the same as before

```
$ make -j 4
$ make -i check -j 4
$ sudo make install
```

1.6 Compiling Chombo

The steps below are for a parallel compilation of `Chombo`. If one wishes to do a serial compilation, then set the flag `MPI` to FALSE and comment the `MPICXX` line below.

We will begin by visiting `https://github.com/GRChombo` and downloading the Chombo directory. This should download the "`Chombo-master.zip`" file. Now just unzip this file. Keep in mind that I rename the folder to "Chombo" and I move it to `~/Documents`.

Then we will move to the directory `/Chombo/lib/mk`. In this directory we will create a new text file called `Make.defs.local`. In it we will write the following:

```
DIM = 3
PRECISION = DOUBLE
OPT = TRUE
DEBUG = FALSE
CXX = g++ -ftemplate-depth-150 -fopenmp
FC = gfortran -ftemplate-depth-150 -fopenmp
MPI = TRUE
MPICXX = mpicc -ftemplate-depth-150 -fopenmp -lmpi
USE_64 = TRUE
USE_HDF = TRUE
HDFINCFLAGS = -I/usr/local/hdf5-1.10.2/serial/include -DH5_USE_16_API
HDFLIBFLAGS = -L/usr/local/hdf5-1.10.2/serial/lib -lhdf5 -lz
HDFMPIINCFLAGS = -I/usr/local/hdf5-1.10.2/parallel/include -DH5_USE_16_API
HDFMPILIBFLAGS = -L/usr/local/hdf5-1.10.2/parallel/lib -lhdf5 -lz
OPENMPCC = TRUE
OMP = TRUE
syslibflags = -lblas -llapack
```

Save the file and go to the `/Chombo/lib` directory. Type in the following command

```
$ make lib -j 4
```

where "4" should be replaced by the number of threads of your processor (my computer has an Intel i7 processor with 4 threads, hence the number used).

Running this should produce a lot of text on your terminal sreen. It should not, however, return any errors. Then run

```
$ make all -j 4
```

This should also complete without errors. The `make all` command compiles all the tests within `Chombo`. It is not necessary for `GRChombo` to work,

but if you get errors when running `make all` on the `Chombo` directory it could indicate you will have problems compiling `GRChombo`.

1.7 Environment variables

At this point we need to set a few permanent environment variables. Before we do so, it is useful to move the `Chombo` directory to whichever directory you want. Keep in mind that if you change `Chombo`'s folder location after this step you will need to change these variables again. It is also a good idea to plan ahead where your `GRChombo` folder will be located at. In what follows I will give as an example the location I have chosen on my machine.

Open your `.bashrc` file with a text editor, e.g. "`vim ~/.bashrc`". Add the following lines to the bottom of the file:

```
export CHOMBO_HOME=~/Documents/Programs/GRChombo/Chombo/lib
export GRCHOMBO_SOURCE=~/Documents/Programs/GRChombo/GRChombo/Source
export LD_LIBRARY_PATH=/home/linuxbrew/.linuxbrew/Cellar/hdf5/lib
```

Save the file and close all terminal windows; open a new one. Type `$ CHOMBO_` and press the `TAB` key. The terminal should complete the variable name and that should lead you to the directory specified above. The same should happen for the other variables. If they fail to do so, check your `.bashrc` file to see if you have indeed made the desired changes.

1.8 Compiling GRChombo

Now we are almost done. Go to `https://github.com/GRChombo` and download the `GRChombo` directory. This should download the "`GRChombo-master.zip`" file to your computer. Unzip it and move the files to the previous specified location. Keep in mind I also rename the folder to `GRChombo`.

Go to the `/GRChombo/Examples/BinaryBH` directory (could be any other example in principle) and type

```
$ make all -j 4
```

As before, this should not return any errors. If this is the case, go to the program's main directory, `/GRChombo`, and type

```
$ make all -j 4
```

That's it, you're all set!

1.9 Installing VisIt

VisIt is the software used by the developers of GRChombo to make beautiful plots and visual simulations. It is compatible with .hdf5 files, so it is a nice idea to install it.

Let us start by going to the following webpage

https://wci.llnl.gov/simulation/computer-codes/visit/executables

and downloading both the install script (copy the page to a file, in my case I have created the file visitinstall.sh) and the Ubuntu 14.04 executable. At the time of writing, version 2.13.0 was downloaded.

Then give permission so that the file can be executed

```
chmod 755 visitinstall.sh
```

and type

```
./visitinstall.sh 2.13.0 linux-x86_64-ubuntu14 /usr/local/visit
```

When prompted, choose the "No System Configuration" option. Then open your ∼/.bashrc file again and include at the bottom of the file the line

```
export PATH="/usr/local/visit/bin:$PATH"
```

Save the file and close it. Close all terminal windows and open a new one. Type in visit. This should open VisIt on your system. To set it up, go to "Options→Host profiles" and add a New Host. You may change the number of nodes/processors accordingly. Hit apply, then dismiss.

To make VisIt look a little better on Ubuntu, go to "Options→Appearance", uncheck the box and choose "Plastique" under "GUI style". You may change the colors if you wish as well. Hit apply.

Next, go to "Options→Save Settings" to keep these modifications when you restart VisIt.