Installing Software Pre-Requisites for ARA Simulation and Data Analysis

Brian Clark [*1], Hannah Hasan [†1], Jude Rajasekera [‡1], and Carl Pfendner [§1]

[1]Department of Physics and the Center for Cosmology and Astroparticle Physics, The Ohio State University

March 31, 2017

Abstract

Advice on installing simulation software prerequisites (ROOT, Boost, etc) on Linux computers for ARA Simulation and Data Analysis. This is verified to play nicely with AraSim 1.0 and AraRoot 3.13. The simulation dependencies should also safely see you through some generations of the IceMC code.

1 Introduction

1.1 Requirements for Simulation Only

All simulation software needed for UHE-radio neutrino simulations of the IceMC heritage– that is, IceMC, AraSim, ShelfMC– require a set of pre-requisite software packages. Principal amongst these is ROOT [1]. Given that ROOT is not installed on most systems and its compile is operating-system specific, users must often install ROOT manually. In addition, AraSim's RayTracer code requires Boost [2].

If you are just interested in running simulation software, you should stop here, and just follow the instructions in section 2. ShelfMC and IceMC just require ROOT, and AraSim requires ROOT and Boost. The versions of these codes that we recommend are ROOT 5.34.34 and Boost 1.49.0. These are verified to play well together on RHEL 6.4, RHEL 6.7, RHEL 6.8, and Mint 17.3.

A word of caution: you must have already installed the prerequisites for these pieces of software. For example, ROOT will require `git`, `gcc` etc. You can find the full list here: `https://root.cern.ch/build-prerequisites`.

[*] clark.2668@osu.edu

[†] hasan.112@osu.edu

[‡] rajasekera.3@osu.edu

[§] pfendner.1@osu.edu

1.2 Requirements for Data Analysis

Running full-blown analysis code, like AraRoot, requires at least four other packages/dependencies: CMake [3], GSL[4], Sqlite [5], FFTW [6]. Always install ROOT last.

Versions of all these code pacakges that are verified to play together nicely are the following: ROOT 5.34.34, Boost 1.49.0, FFTW 3.3.4, Sqlite 3.9.2, CMake 3.4.0, and GSL 1.16. These are verified to play well together on RHEL 6.4, RHEL 6.7, RHEL 6.8, and Mint 17.3.

1.3 Directory Structure and General Procedure

We recommend the coding practice of having a build directory separate from the code source directory. You should have a ∼`/source` and ∼`/tools` directory. The `source` directory will contain the source code tarballs, and the `tools` directory will contain the compiled binaries, library files, etc. As such, the `tools` directory should have the following subdirectories: `bin` , `doc` , `include` , `lib` , `macros` , `scripts` , `share` , and `etc`. You can of course stick this anywhere in your file system, for example:`home/username/Desktop/source`, but for this guide we do the installation with reference to `home`.

The general procedure for installing all software dependencies is the following:

1. Move the source tarball (.tar.gz) file into the `source` directory.
2. Untar the source file: `tar xf filename.tar.gz`
3. Go into the directory, and configure the code: `./configure --prefix=~/tools`
4. compile it/make it: `make`
5. After the code compiles, install the code: `make install`

2 Installing Simulation Dependencies

The two simulation dependencies are BOOST and ROOT.

2.1 Installing ROOT

Root is a scientific software framework that provides tools that aid with big data processing, statistical analysis, visualisation and storage [1].

To download and unpack the source code, do the following. We recommend version 5.34.34.

```
wget https://root.cern.ch/download/root_v5.34.34.source.tar.gz
tar -zxf root_v5.34.34.source.tar.gz
cd root
./config --etcdir=~/tools/etc --prefix=~/tools
make
make install
```

2.2 Installing Boost

Boost is an extended C++ library necessary for AraSim's raytracing algorithm. The tar file for version 1.49.0 can by obtained via the following command:

```
wget http://sourceforge.net/projects/boost/files/boost/1.49.0/boost_1_49_0.tar.gz
```

After downloading the `tar.gz` file, it can be unzipped with

```
tar -xvzf boost_1_49_0.tar.gz
```

The `x` option extracts files; `v` specifies verbose action, i.e. the files will be listed as they are extracted; `z` specifies that the files should be decompressed using `gzip`; and `f` is necessary in order to specify what file needs to be decompressed. Additionally, if you would like the extracted files to be placed in a particular directory, then `-C [path to directory]` may be appended to the above command (without brackets), so the whole line reads:

```
tar -xvzf boost_1_49_0.tar.gz -C [path to directory]
```

Upon extraction, there will be a new directory containing the extracted files. Among these will be an executable shell script called `bootstrap.sh`. Execute it by typing

```
./bootstrap --prefix=~/tools
```

and after bootstrapping is finished,

```
./b2 install
```

to complete installation.

2.3 Modifying your .bashrc

The last step to a successful installation is you need to modify your `.bashrc` file. The `.bashrc` is a hidden file in your home directory. You need to add the following lines. Again, you can replace `~/tools` with `/your/path/to/tools/`.

```
export PLATFORM_DIR=~/tools/
export DYLD_LIBRARY_PATH=${PLATFORM_DIR}/lib:$DYLD_LIBRARY_PATH
export LD_LIBRARY_PATH=${PLATFORM_DIR}/lib:$LD_LIBRARY_PATH
export PATH=${PLATFORM_DIR}/bin:$PATH
export ROOTSYS=~/opt/root
. thisroot.sh
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ROOTSYS/lib
export PATH=$ROOTSYS/bin:$PATH
```

Some people have more success with configuring the ROOT lines as follows:

```
cd ~/tools/lib/root
. thisroot.sh
cd ~
export ROOTSYS=~/tools
```

Those who are bash savvy will notice the use of `PLATFORM_DIR` really isn't necessary, and you're welcome to configure that away.

3 Installing Data Analysis Dependencies

Aside from designating the install directory (through `--prefix=∼/tools`), Carl Pfendner documented necessary arguments and CFLAGS for the code to compile correctly on computing clusters. If you would like to try an automated installations script for all these dependencies, you can use `https://github.com/clark2668/AutomaticInstall`.

3.1 Installing GSL

The GNU scientific library is needed for ROOT to compile the MathMore library. For GSL, we recommend version 1.16. To download it from the web, do:

```
wget http://gnu.mirror.constant.com/gsl/gsl-1.16.tar.gz
```

The configure command with necessary flags is:

```
./configure CFLAGS=-m64 --with-pic --enable-shared --prefix=∼/tools
```

Then run `make` and `make install` to finish the installation.

3.2 Installing FFTW

FFTW is the analysis software's Fourier transform engine. For FFTW, we recommend version 3.3.4. To download it from the web, do:

```
wget ftp://ftp.fftw.org/pub/fftw/fftw-3.3.4.tar.gz
```

The configure command with necessary flags is:

```
./configure --enable-shared --prefix=∼/tools
```

Then run `make` and `make install` to finish the installation.

3.3 Installing CMake

CMake is the build system for AraRoot. For CMake, we recommend version 3.9.2. To download it from the web, do:

```
wget https://cmake.org/files/v3.4/cmake-3.4.0.tar.gz
```

The configure command with necessary flags is:

```
./configure --prefix=∼/tools
```

Then run `gmake` and `make install` to finish the installation.

3.4 Installing Sqlite

Sqlite is used by AraRoot to store data about the stations, such as antenna locations, etc. For Sqlite, we recommend version 3.9.2. To download it from the web, do:

```
wget http://www.sqlite.org/2015/sqlite-autoconf-3090200.tar.gz
```

The configure command is

```
./configure --enable-shared --prefix=∼/tools
```

Then run `make` and `make install` to finish the installation.

3.5 Installing ROOT

When installing ROOT for compatibility with data analysis, it has to be asked to compile with FFTW and with GSL so that it generates the MathMore library. Because ROOT is annoying, you must also specify an `etcdir`. The configure command to use is:

```
   ./configure --with-fftw3-incdir=~/tools/include
               --with-fftw3-libdir=~/tools/lib
               --with-gsl-incdir=~/tools/include
               --with-gsl-libdir=~/tools/lib
               --enable-gsl-shared
               --etcdir=~/tools/etc
               --prefix=~/tools
```

Then run `make` and `make install` to finish the installation.

3.6 Modifying your .bashrc

The last step to a successful installation is you need to modify your `.bashrc` file. The `.bashrc` is a hidden file in your home directory. You need to add the following lines. Again, you can replace `~/tools` with `/your/path/to/tools/`.

```
export PLATFORM_DIR=~/tools/
export DYLD_LIBRARY_PATH=${PLATFORM_DIR}/lib:$DYLD_LIBRARY_PATH
export LD_LIBRARY_PATH=${PLATFORM_DIR}/lib:$LD_LIBRARY_PATH
export PATH=${PLATFORM_DIR}/bin:$PATH
export ROOTSYS=~/opt/root
. thisroot.sh
export BOOST_ROOT = ~/source/source_boost
export LD_LIBRARY_PATH=${BOOST_ROOT}/stage/lib:$LD_LIBRARY_PATH
export DYLD_LIBRARY_PATH=$DYLD_LIBRARY_PATH:${BOOST_ROOT}/stage/lib
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ROOTSYS/lib
export PATH=$ROOTSYS/bin:$PATH
export SQLITE_ROOT=${PLATFORM_DIR}
export GSL_ROOT=${PLATFORM_DIR}
export FFTWSYS=${PLATFORM_DIR}
```

Some people have more success with configuring the ROOT lines as follows:

```
cd ~/tools/lib/root
. thisroot.sh
cd ~
export ROOTSYS=~/tools
```

4 Other Resources

There are a few other good places to find help with installing pre-requisites.

1. For IceMC: `anita.physics.ohio-state.edu/documentation/svndirections.php`
2. For AraSim: `ara.physics.ohio-state.edu/documentation/svndirections.php`
3. For ROOT FFTW Wrapper: `https://www.hep.ucl.ac.uk/uhen/libRootFftwWrapper/`
4. For AraRoot: `https://www.hep.ucl.ac.uk/uhen/ara/araroot/branches/3.13/`

References

[1] https://root.cern.ch/.

[2] http://www.boost.org/.

[3] https://cmake.org/.

[4] https://www.gnu.org/software/gsl/.

[5] https://www.sqlite.org/.

[6] http://www.fftw.org/.