

Installing LOFAR Software on a CentOS 6.5 (Final) system (cluster Newton of the AIP Potsdam)

— [Frank Breitling](#) 2015/06/26 18:47

I was mainly following the previous installation notes provided by others:

http://www.lofar.org/operations/doku.php?id=public:user_software:lofar

http://www.lofar.org/operations/doku.php?id=public:user_software:ubuntu_12_4

However a few problems occurred and had to be solved as described below.

Ideas and support by **Arno Schoenmakers**, **Ger van Diepen** and **Marcel Loose** were very helpful and highly appreciated.

1. Setup of the environment

The following packages are already provided on the cluster:

- BLAS
- LAPACK
- Boost 1.41.0
- cfitsio 3.240
- HDF5
- OpenMP

The cluster also provides a *module system* to load newer versions of the installed software. We will activate it and replace the default version of gcc 4.4.7 with gcc 4.8.3 for C++11 support.

1.1 Pathes

This instruction will install all packages in `$HOME/local/`.

And for the installation of Pyrap (python-casacore) we need a Python path with write access.

So I added the following lines to `$HOME/.bashrc`:

```
export PYTHONPATH=$HOME/local/lib64/python2.6/site-packages:$PYTHONPATH
export LD_LIBRARY_PATH=$HOME/local/lib:$LD_LIBRARY_PATH
export PATH=$HOME/local:$PATH
source /usr/share/Modules/init/sh # activate the module system
module load gcc # use gcc 4.8.3 (instead of 4.4.7) for C++11 support
```

1.2 CMake

The 4 Newton head nodes provide CMake 2.6 by default, but the LOFAR Software requires at least 2.6.6. A sufficiently recent version of CMake 3.0.0 is provided via [Environment Modules](#) (see also <http://modules.sourceforge.net/>).

The environment modules also provide more recent HDF5 libraries. Both could be loaded by adding `module load cmake hdf5` to `$HOME/.bashrc`.

But unfortunately the CMake 3.0.0 loaded on the head node newl1 is broken. Furthermore loading the HDF5 libs has no effect on the libs found by CMake, which are still the system's default. So we don't load any module and don't add this line to the `.bashrc`. Instead we build our own CMake as follows:

```
mkdir -p $HOME/local/src
cd $HOME/local/src
wget http://www.cmake.org/files/v3.2/cmake-3.2.3.tar.gz
tar xf cmake-3.2.3.tar.gz
cd cmake-3.2.3
bootstrap --prefix $HOME/local
make -j12
make install
```

2. Installation of required packages

Next we need to build and install these other packages as follows:

```
cd $HOME/local/src/
wget ftp://ftp.atnf.csiro.au/pub/software/wcslib/wcslib.tar.bz2
tar xf packages/wcslib.tar.bz2
cd wcslib-5.6/
./configure --prefix=$HOME/local/
make install

cd $HOME/local/src/
wget http://www.fftw.org/fftw-3.3.4.tar.gz
cd fftw-3.3.4
configure --prefix=$HOME/local --enable-threads --enable-shared --enable-float
make -j12 install
```

3. Installation the LOFAR software

Now we can continue building the LOFAR software.

3.1 Essential CMake switches

We will need these essential CMake switches:

1. -DUSE_OPENMP=YES
2. -DCMAKE_INSTALL_PREFIX:PATH=\$HOME/local
3. -DBoost_NO_BOOST_CMAKE=YES
4. -DENABLE_TABLELOCKING=NO (only for Casacore)
5. -DUSE_LOG4CPLUS=NO (only for the LOFAR packages)
6. -DUSE_BACKTRACE=NO

1.) OpenMP makes use of multi-core CPUs. We want to use this since it speeds up our processing.

2.) Sets the installation path to \$HOME/local/.

3.) Is necessary when using some older versions of boost with cmake-2.8.6-rc2 or later, where the boost finding has changed.

(See

<https://stackoverflow.com/questions/9948375/cmake-find-package-succeeds-but-returns-wrong-path>.)

Otherwise make will show this error later:

```
make[2]: *** No rule to make target
`/usr/lib64/lib64/libboost_program_options-mt.so.5', needed by `segment'.
Stop.
make[1]: *** [CMakeFiles/segment.dir/all] Error 2
make: *** [all] Error 2
```

4.) On the AIP Lustre file system Casacore cannot acquire read locks leading to errors like:

```
msoverview: Version 20110407GvD
2015-06-30 19:32:19      INFO      Process 13592: waiting for read-lock
on file
/lustre/fkbreitl/data/L206894/L206894.cal/L206894_SAP000_SB200_uv.MS.dppp_00
001~00040/table.lock
Error: Error (Function not implemented) when acquiring lock on
/lustre/fkbreitl/data/L206894/L206894.cal/L206894_SAP000_SB200_uv.MS.dppp_00
001~00040/table.lock
```

With table locking switched off these errors are avoided. For practical work there is no disadvantage in disabling table locking.

This switch is only needed for Casacore!

5.) Is necessary for the LOFAR packages if the Log4cplus libs are not installed.

6.) Is necessary if building shared libraries (the default with LOFAR), but libiberty.so is not installed on your system, only libiberty.a.

(See https://support.astron.nl/lofar_issuetracker/issues/8046).

So we can build the rest of the packages as follows.

3.2 Casacore

Casacore also requires the CASA data. This is for example provided with [CASA](#).

Here we assume CASA is already installed and \$HOME/local/casa is a link to the CASA folder. Then the

data is in \$HOME/local/casa.

We will specify this via -DDATA_DIR, otherwise we need a ~/.casarc file with the line:
measures.directory: \$HOME/local/casa/data .

```
cd ~/local/src/
git clone https://github.com/casacore/casacore
mkdir build/casacore
cd build/casacore
time cmake -DBUILD_PYTHON=YES -DCMAKE_INSTALL_PREFIX=~/local -DUSE_FFTW3=YES
-DENABLE_TABLELOCKING=NO \
  -DUSE_OPENMP=YES -DDATA_DIR=~/local/casa/data -DUSE_HDF5=YES -
DCMAKE_PREFIX_PATH=$HOME/local ../../casacore
time make -j16
make install
```

3.3 Pyrap

```
cd ~/local/src/
git clone https://github.com/casacore/python-casacore.git
cd python-casacore
python setup.py build_ext -I/usr/include/cfitsio:$HOME/local/include -
L$HOME/local/lib
python setup.py install --prefix=$HOME/local
```

3.3.1 Alternative Pyrap installation

Alternatively we can install Pyrap with *pip* if we install pip before:

```
wget https://bootstrap.pypa.io/get-pip.py
python get-pip.py --user
~/local/bin/pip install python-casacore --global-option=build_ext --global-
option=-I/usr/include/cfitsio:$HOME/local/include \
  --global-option=-L$HOME/local/lib --install-option=--prefix=$HOME/local
```

3.4 Casarest

Now we continue with the rest:

```
cd ~/local/src/
svn co https://svn.astron.nl/casarest/trunk/casarest
mkdir build/casarest
cd build/casarest
cmake -DCASACORE_ROOT_DIR=$HOME/local -DBUILD_ALL=1 -
DCMAKE_INSTALL_PREFIX:PATH=$HOME/local \
  -DCMAKE_PREFIX_PATH=/usr/include/cfitsio -DBoost_NO_BOOST_CMAKE=YES
```

```
../../casarest
time make -j16
make install
```

3.5 AOFlagger

Since LOFAR Offline release 2.15.0 also an installation of the AOFlagger is required.

A download link and installation instructions are given at

https://sourceforge.net/p/aoflagger/wiki/installation_instructions/

and http://www.lofar.org/wiki/doku.php?id=public:offline_release_2_15_0

```
cd ~local/src/
tar xf aoflagger-2.7.1.tar.bz2
mkdir aoflagger-2.7.1/build/
cd aoflagger-2.7.1/build/
cmake -DCMAKE_INSTALL_PREFIX:PATH=$HOME/local -DBUILD_SHARED_LIBS=YES ../
time make -j 16
make install
```

3.6 The LOFAR packages

Finally we can build the LOFAR packages. We can chose between a release or the latest version in trunk/ by commenting in and out the corresponding RELEASE lines.

An overview of available releases is given at

http://www.lofar.org/operations/doku.php?id=public:lofar_offline_releases .

```
cd ~local/src/
RELEASE=LOFAR-Release-2_16 && svn checkout --ignore-externals -N
https://svn.astron.nl/LOFAR/branches/$RELEASE
#RELEASE=LOFAR && svn checkout --ignore-externals -N
https://svn.astron.nl/LOFAR/trunk $RELEASE
svn update $RELEASE/CMake
mkdir -p build/gnu_opt
cd build/gnu_opt
time cmake -DCASACORE_ROOT_DIR=$HOME/local -DBUILD_PACKAGES="DP3
Calibration" -DCMAKE_INSTALL_PREFIX:PATH=$HOME/local \
  -DUSE_OPENMP=YES -DUSE_LOG4CPLUS=NO -DUSE_BACKTRACE=NO -
DBoost_NO_BOOST_CMAKE=YES ../../$RELEASE
time make -j16
make install
```

Done!

Now the commands NDPPP and calibrate-stand-alone should be available on this system.

Enjoy!

From:
<https://www.astron.nl/lofarwiki/> - **LOFAR Wiki**

Permanent link:
https://www.astron.nl/lofarwiki/doku.php?id=public:install_lofar_centos6.5&rev=1482419882

Last update: **2016-12-22 15:18**

