Intallation of LOFAR in Red Hat Enterprise 5.8

This is the description of the installation of LOFAR on the cluster of the IAA in Granada. It bundles an old version of Red Hat Enterprise Linux that does not contain many of the new libraries required by LOFAR.

Description of the system:

- Operating system: Red Hat Enterprise Linux Server release 5.8 (Tikanga)
- PBS queue system: TORQUE
- Libraries available: HDF5 and old versions of boost and fftw3.

After some tests we determined that we needed to install, at least, the following software and libraries:

- scons
- WCS
- lapack
- atlas
- fftw3
- Python
    - numpy
    - virtualenv (optional)
    - h5py (optional)
    - scipy
    - matplotlib
    - pandas (optional)
    - pyfits
    - pywcs
    - astropy (optional)
    - cython
    - IPython (optional)
- Boost
- LOFAR related:
    - casacore
    - pyrap
    - casarest
    - LOFAR (LofIm)

Installation steps

The software was installed in the home of the user `jsm` (`/home/users/dae/jsm`). You should change the scripts accordingly.

We will install Python in `/home/users/dae/jsm/python` and the rest of libraries and software in `/home/users/dae/jsm/local`. The HDF5 library is already in the system in

/usr/local/hdf5-1.8.11-linux-x86_64-shared. A standalone version of casapy is also present in the system.

Previous steps:

```
## Create the installation directory
mkdir -p /home/users/dae/jsm/local/src
cd /home/users/dae/jsm/local/src
```

Download and copy the data:

```
# scons
wget http://prdownloads.sourceforge.net/scons/scons-2.3.0.tar.gz
# WCS
wget ftp://ftp.atnf.csiro.au/pub/software/wcslib/wcslib.tar.bz2
# lapack
wget http://www.netlib.org/lapack/lapack-3.4.2.tgz
# atlas
wget http://prdownloads.sourceforge.net/math-
atlas/Developer%20%28unstable%29/3.11.11/atlas3.11.11.tar.bz2
# FFTW3
wget http://www.fftw.org/fftw-3.3.3.tar.gz
# Boost
wget
http://prdownloads.sourceforge.net/boost/boost/1.54.0/boost_1_54_0.tar.gz
# LOFAR related software
# casacore
svn co http://casacore.googlecode.com/svn/tags/casacore-1.5.0
# pyrap
svn co http://pyrap.googlecode.com/svn/tags/pyrap-1.1.0
# casarest
svn co https://svn.astron.nl/casarest/trunk/casarest/
```

We had already got a copy of the LOFAR software with SVN in /home/users/dae/jsm/LOFAR:

```
# LOFAR
cp -r /home/users/dae/jsm/LOFAR .
```

We used the following paths:

```
# PATH
export
PATH=/home/users/dae/jsm/local/bin:/home/users/dae/jsm/python/bin:/usr/local
/casapy-stable-42.0.25430-001-64b:/usr/local/iraf/bin:/usr/local/mpich-
install/bin:/usr/local/bin:/bin:/usr/bin
# LD_LIBRARY_PATH
export
LD_LIBRARY_PATH=/home/users/dae/jsm/local/lib:/home/users/dae/jsm/local/atla
s/lib:/home/users/dae/jsm/python/lib:/usr/local/hdf5-1.8.11-linux-x86_64-
shared/lib:/usr/local/mpich-install/lib:/usr/local/lib
```

Install Python and some Python libraries:

```
# Compile
mkdir -p /home/users/dae/jsm/python/src
cd /home/users/dae/jsm/python/src
# Download Python **** CHANGE
# Extract Python **** CHANGE
cd Python-2.7.5
./configure --prefix=/home/users/dae/jsm/python --enable-shared
make
make install

# Install zeromq (used by IPython parallel)
cd /home/users/dae/jsm/python/src
wget http://download.zeromq.org/zeromq-3.2.4.tar.gz
tar xzf zeromq-3.2.4.tar.gz
cd zeromq-3.2.4
./configure --prefix=/home/users/dae/jsm/local
make
make install

# Install easy install and pip
cd /home/users/dae/jsm/python/src
wget https://bitbucket.org/pypa/setuptools/raw/bootstrap/ez_setup.py
python ez_setup.py
wget https://raw.github.com/pypa/pip/master/contrib/get-pip.py
# rm -r /tmp/pip-build-jsm/pip
python get-pip.py

# Install packages
pip install virtualenv
pip install numpy
pip install h5py
pip install scipy
pip install matplotlib
pip install pandas
pip install pyfits
pip install pywcs
pip install astropy
pip install cython

# Install pyzmq
cd /home/users/dae/jsm/python/src
wget https://pypi.python.org/packages/source/p/pyzmq/pyzmq-13.0.2.tar.gz --
no-check-certificate
tar xzf pyzmq-13.0.2.tar.gz
cd pyzmq-13.0.2
python setup.py configure --zmq=/home/users/dae/jsm/local/
python setup.py install

# Install IPython
```

```
cd /home/users/dae/jsm/python/src
pip install ipython[all]
```

From now on we need to enter an interactive PBS job to avoid our jobs to be killed after certain amount of time (the maximum allowed length for a job without using PBS is 10 minutes). The nodes do not have direct access to the Internet (that is why the packages were previously downloaded):

```
### Enter interactive qsub session
qsub -I -X -N LOFAR_soft

## Update the PATH and the LD_LIBRARY_PATH
# PATH
export
PATH=/home/users/dae/jsm/local/bin:/home/users/dae/jsm/python/bin:/usr/local
/hdf5-1.8.11-linux-x86_64-shared/bin:/usr/local/casapy-
stable-42.0.25430-001-64b:/usr/local/iraf/bin:/usr/local/mpich-
install/bin:/usr/local/bin:/bin:/usr/bin
# LD_LIBRARY_PATH
export
LD_LIBRARY_PATH=/home/users/dae/jsm/local/lib:/home/users/dae/jsm/local/atla
s/lib:/home/users/dae/jsm/python/lib:/usr/local/hdf5-1.8.11-linux-x86_64-
shared/lib:/usr/local/mpich-install/lib:/usr/local/lib
```

We needed to manually specify the paths to avoid cmake finding old or incorrect versions of the libraries.

Compile the libraries and tools:

```
## scons
cd /home/users/dae/jsm/local/src
tar xfz scons-2.3.0.tar.gz
cd scons-2.3.0
python setup.py install --prefix=/home/users/dae/jsm/local

## ATLAS + LAPACK + BLAS
cd /home/users/dae/jsm/local/src
tar xfj atlas3.11.11.tar.bz2
mkdir -p ATLAS/build
cd ATLAS/build
../configure -b 64 -D c -DPentiumCPS=2932 --shared \
  --prefix=/home/users/dae/jsm/local/atlas \
  --with-netlib-lapack-
tarfile=/home/users/dae/jsm/local/src/lapack-3.4.2.tgz
make build
make check
make ptcheck
make time
make install
ln -s /home/users/dae/jsm/local/atlas/lib/libcblas.a \
  /home/users/dae/jsm/local/atlas/lib/libblas.a
```

```
## WCS
cd /home/users/dae/jsm/local/src
tar xfj wcslib.tar.bz2
cd wcslib-4.19
./configure --prefix=/home/users/dae/jsm/local/
make
make install

## FFTW3
cd /home/users/dae/jsm/local/src
tar xfz fftw-3.3.3.tar.gz
cd fftw-3.3.3
./configure --enable-threads --enable-shared \
  --prefix=/home/users/dae/jsm/local/
make
make install
make clean
./configure --enable-threads --enable-shared  --enable-float \
  --prefix=/home/users/dae/jsm/local/
make
make install

## Boost
cd /home/users/dae/jsm/local/src
tar xfz boost_1_54_0.tar.gz
cd boost_1_54_0
./bootstrap.sh --prefix=/home/users/dae/jsm/local \
  --with-python=/home/users/dae/jsm/python/bin/python
./b2
./b2 install
```

Some notes:

- You should adapt the parameter “DPentiumCPS=2932” to your system.

Compile casacore

```
## casacore
cd /home/users/dae/jsm/local/src
cd casacore-1.5.0
mkdir build; cd build
cmake .. -DWCSLIB_LIBRARY=/home/users/dae/jsm/local/lib/libwcs.so \
         -DWCSLIB_INCLUDE_DIR=/home/users/dae/jsm/local/include/ \
         -DCMAKE_INSTALL_PREFIX=/home/users/dae/jsm/local/casacore \
         -DUSE_HDF5=ON -DUSE_FFTW3=ON \
         -DHDF5_ROOT_DIR=/usr/local/hdf5-1.8.11-linux-x86_64-shared \
         -DLAPACK_LIBRARIES=/home/users/dae/jsm/local/atlas/lib/liblapack.a
\
         -
DBLAS_LIBRARIES=/home/users/dae/jsm/local/atlas/lib/libf77blas.a\;/home/user
```

```
s/dae/jsm/local/atlas/lib/libcblas.a\;/home/users/dae/jsm/local/atlas/lib/li
batlas.a \
         -DBLA_VENDOR=ATLAS \
         -DFFTW3_LIBRARY=/home/users/dae/jsm/local/lib/libfftw3.so \
         -
DFFTW3_THREADS_LIBRARY=/home/users/dae/jsm/local/lib/libfftw3_threads.so \
         -DFFTW3F_LIBRARY=/home/users/dae/jsm/local/lib/libfftw3f.so \
         -
DFFTW3F_THREADS_LIBRARY=/home/users/dae/jsm/local/lib/libfftw3f_threads.so \
         -DFFTW3_INCLUDE=/home/users/dae/jsm/local/include/ \
         -DCMAKE_Fortran_COMPILER:FILEPATH=/usr/bin/gfortran
make
make install
```

Install pyrap (scons needed):

```
## pyrap

# pyrap looks for libblas
cd /home/users/dae/jsm/local/atlas/lib
ln -s libcblas.a libblas.a

cd /home/users/dae/jsm/local/src
cd pyrap-1.1.0
## PATCH pyrap ******* (see next section)
python batchbuild.py --boost-root=/home/users/dae/jsm/local \
  --casacore-root=/home/users/dae/jsm/local/casacore \
  --enable-hdf5 --hdf5-root=/usr/local/hdf5-1.8.11-linux-x86_64-shared \
  --lapack-root=/home/users/dae/jsm/local/atlas \
  --wcs-root=/home/users/dae/jsm/local \
  --prefix=/home/users/dae/jsm/local
```

Install casarest

```
## casarest
cd /home/users/dae/jsm/local/src
cd casarest
mkdir build; cd build
cmake .. -DBOOST_ROOT=/home/users/dae/jsm/local \
         -DLAPACK_LIBRARIES=/home/users/dae/jsm/local/atlas/lib/liblapack.a
\
         -DLAPACK_INCLUDE=/home/users/dae/jsm/local/atlas/include/ \
         -
DBLAS_LIBRARIES=/home/users/dae/jsm/local/atlas/lib/libf77blas.a\;/home/user
s/dae/jsm/local/atlas/lib/libcblas.a\;/home/users/dae/jsm/local/atlas/lib/li
batlas.a \
         -DBLAS_INCLUDE=/home/users/dae/jsm/local/atlas/include/ \
         -DWCSLIB_LIBRARY=/home/users/dae/jsm/local/lib/libwcs.so \
         -DWCSLIB_INCLUDE_DIR=/home/users/dae/jsm/local/include/ \
         -DCASACORE_ROOT_DIR=/home/users/dae/jsm/local/casacore \
```

```
        -DCMAKE_INSTALL_PREFIX=/home/users/dae/jsm/local/casarest \
        -DHDF5_ROOT_DIR=/usr/local/hdf5-1.8.11-linux-x86_64-shared \
        -
DLIB_EXTRA_SYNTHESIS=/home/users/dae/jsm/local/atlas/lib/libcblas.a\;/home/u
sers/dae/jsm/local/atlas/lib/libatlas.a \
        -DBUILD_ALL=1
make
make install
```

Install LOFAR:

```
cd /home/users/dae/jsm/local/src
#cp -r ~/LOFAR .
cd LOFAR
### PATCH LOFAR ******** (see next section)
mkdir -p build/gnu_opt; cd build/gnu_opt
mkdir /home/users/dae/jsm/local/LofIm
cmake ../.. -DBUILD_SHARED_LIBS=ON \
  -DCMAKE_INSTALL_PREFIX=/home/users/dae/jsm/local/LofIm \
  -DUSE_LOG4CPLUS=OFF \
  -DBOOST_ROOT=/home/users/dae/jsm/local \
  -DHDF5_ROOT_DIR=/usr/local/hdf5-1.8.11-linux-x86_64-shared \
  -DWCSLIB_LIBRARY=/home/users/dae/jsm/local/lib/libwcs.so \
  -DWCSLIB_INCLUDE_DIR=/home/users/dae/jsm/local/include/ \
  -
DLAPACK_LIBRARIES=/home/users/dae/jsm/local/atlas/lib/libf77blas.a\;/home/us
ers/dae/jsm/local/atlas/lib/libcblas.a\;/home/users/dae/jsm/local/atlas/lib/
libatlas.a\;/home/users/dae/jsm/local/atlas/lib/liblapack.a \
  -
DBLAS_LIBRARIES=/home/users/dae/jsm/local/atlas/lib/libf77blas.a\;/home/user
s/dae/jsm/local/atlas/lib/libcblas.a\;/home/users/dae/jsm/local/atlas/lib/li
batlas.a\;/home/users/dae/jsm/local/atlas/lib/liblapack.a  \
  -DFFTW3_LIBRARY=/home/users/dae/jsm/local/lib/libfftw3.so \
  -DFFTW3_THREADS_LIBRARY=/home/users/dae/jsm/local/lib/libfftw3_threads.so
\
  -DFFTW3F_LIBRARY=/home/users/dae/jsm/local/lib/libfftw3f.so \
  -
DFFTW3F_THREADS_LIBRARY=/home/users/dae/jsm/local/lib/libfftw3f_threads.so \
  -DFFTW3_INCLUDE=/home/users/dae/jsm/local/include/ \
  -DCASACORE_ROOT_DIR=/home/users/dae/jsm/local/casacore \
  -
DCASA_CASA_LIBRARY=/home/users/dae/jsm/local/casacore/lib/libcasa_casa.so\;/
home/users/dae/jsm/local/atlas/lib/libf77blas.a\;/home/users/dae/jsm/local/a
tlas/lib/libcblas.a\;/home/users/dae/jsm/local/atlas/lib/libatlas.a\;/home/u
sers/dae/jsm/local/atlas/lib/liblapack.a \
  -DUSE_OPENMP=ON \
  -DBUILD_PACKAGES=Offline\;LofarFT\;StaticMetaData\;SPW_Combine \
  -DCASAREST_ROOT_DIR=/home/users/dae/jsm/local/casarest/ \
  -DPYTHON_LIBRARY=/home/users/dae/jsm/python/lib/libpython2.7.so \
  -DPYTHON_EXECUTABLE:FILEPATH=/home/users/dae/jsm/python/bin/python \
  -DPYTHON_INCLUDE_DIR:PATH=/home/users/dae/jsm/python/include \
```

```
  -DPYRAP_INCLUDE_DIR=/home/users/dae/jsm/local/include/pyrap \
  -DPYRAP_LIBRARY=/home/users/dae/jsm/local/lib/libpyrap.so \
  -DCMAKE_Fortran_COMPILER:FILEPATH=/usr/bin/gfortran \
  -DF2PY_FCOMPILER=gnu95 2>&1 | tee log.txt
make
make install
```

Finish the installation (

):

- Configure .bashrc
- Configure .casarc

Patches and hacks needed

Pyrap and LOFAR need to be patched to allow them to compile in this old system.

Pyrap tries to find the libraries in lib64 in a Red Hat linux. This happens even if the libraries are manually compiled and are located in a `lib` directory. If the user cannot create a symbolic link from `lib64` to `lib`, pyrap needs to be patched. The `lib64` entries linked to `redhat` found in the setupext.py files and utils.py need to be edited to `lib`. Edit:

- pyrap_fitting/current/setupext.py
- pyrap_functionals/current/setupext.py
- pyrap_images/current/setupext.py
- pyrap_measures/current/setupext.py
- pyrap_quanta/current/setupext.py
- pyrap_tables/current/setupext.py
- pyrap_util/current/setupext.py
- libpyrap/current/scons-tools/utils.py

Pyrap cannot determine the location of some shared libraries in this old system. *Ugly hack* - The file `libpyrap/current/SConstruct` can be edited to ad some linker flags: After

```
    conf = Configure(env)
```

insert:

```
    conf.env.Append(LINKFLAGS = ["-Wl,-
rpath,/home/users/dae/jsm/local/casacore/lib:/usr/local/hdf5-1.8.11-linux-
x86_64-shared/lib:/home/users/dae/jsm/local/lib"])
```

Pyrap looks for libblas instead of libcblas. A symbolic link can be made:

```
cd /home/users/dae/jsm/local/atlas/lib
ln -s libcblas.a libblas.a
```

LOFAR uses the flag `-Wno-ignored-qualifiers` that is not supported by this old system. It has to be removed from the following files:

- CEP/Imager/LofarFT/src/CMakeLists.txt
- CEP/Calibration/BBSTools/src/CMakeLists.txt
- LCS/AMC/AMCImpl/src/CMakeLists.txt

The version of cmake bundled in the system contains buggy versions of the recipes to find BLAS and LAPACK (FindBLAS.cmake and FIindLAPACK.cmake). Ii is possible to see in the CMakeCache.txt files that these libraries are not detected by CMAKE in this system (for example in the case of **casacore**). The manual setting of the cache variables *LAPACK_LIBRARIES* and *BLAS_LIBRARIES* made the compilation and linking work. The variables *LAPACK_INCLUDE* and *BLAS_INCLUDE* were ignored. But... take into account that these cache variables may not be present in modern compilations of cmake.

The ugliest hack is this line in the cmake options of **LOFAR**:

```
-
DCASA_CASA_LIBRARY=/home/users/dae/jsm/local/casacore/lib/libcasa_casa.so\;/
home/users/dae/jsm/local/atlas/lib/libf77blas.a\;/home/users/dae/jsm/local/a
tlas/lib/libcblas.a\;/home/users/dae/jsm/local/atlas/lib/libatlas.a\;/home/u
sers/dae/jsm/local/atlas/lib/liblapack.a
```

The library librfiengine.so is linked with dependencies to LAPACK (and this can present dependencies with BLAS). These dependencies are not linked in some of the remaining packages because they are not a direct dependency but the linker will fail. We overrided the cache variable pointing to one of the casacore libraries. Wherever librfiengine.so is linked, the libcasa_casa.so is linked as well. This ugly thing worked.

Warnings and advice

- Compile the libraries with the `shared` option (-fPIC in the compiler).
- The tools `ldd` and `nm` could help you to debug problems with the linking of libraries.
- Have a look to CMakeCache.txt before running cmake to spot possible problems with the libraries.
- If you use the ifort FORTRAN compiler you might find problems with the linking of ifort shared libraries.
- Cmake could find some of the casapy libraries and link them as dependencies. If you want to avoid that you should specify the cache variables by hand (as it is done in this example).
- To link to precompiled external libraries (like the ones provided by casapy or Anaconda) might be a bad idea.
- The option “–blas” of pyrap did not work and made the build script fail.
- Final advice: Try to work on modern systems (e. g. Ubuntu 12.04 LTS) if you can :).

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

Permanent link:
https://www.astron.nl/lofarwiki/doku.php?id=public:install_lofar_redhat5.8&rev=1381411673

Last update: **2013-10-10 13:27**