Welcome information in the LOFAR EoR processing

Oscar Martinez
University of Groningen
Kapteyn Astronomical Institute
Groningen
The Netherlands
September 16, 2016

1 Introduction

First of all, welcome to the LOFAR EoR group, a very international bunch of very smart and particular people. You will have fun. Enjoy!. Now let's talk about business so you can start as soon as possible to help us to find "the signal". This documents gives a brief introduction in processing the LOFAR EoR data in the LOFAR EoR cluster.

2 LOFAR EoR cluster

The LOFAR EoR cluster is the CPU/GPU computer cluster used for the processing of the LOFAR EoR data.

Account

The first thing you need is an account in the LOFAR EoR cluster. Contact the system administrator, Eite Tiesinga (tiesinga@astro.rug.nl).

Log in

Use *ssh* to log in in the portal node (called *lofareor01*) of the LOFAR EoR cluster:

```
ssh PRIVATE
```

From this node you can access any of the nodes of the cluster.

The nodes

There are 80 nodes, named *node001* to *node080*, and they are accessed through *ssh*. Each node has 3 disks mounted in */data1*, */data2* and */data3*. Some nodes do not have all the disks and sometimes they break so do not freak out if that happens, it is natural selection. Darwin also applies to data disks. NFS is also available to access the data in different nodes. For example, if you are in node001 and you want to access the *data1* disk in *node002* you can do it with */net/node002/data1*. Few comments on the several nodes of the LOFAR EoR cluster:

- node001 to node010 can be freely used by the users. New LOFAR EoR users should be using one of them. However, ask which one you can use. Your life can be in danger if you use the wrong one.
- node011 to node074 are reserved for the processing of the project data. Do not use them without consultation. Again, your life can be in the edge.
- node075 is used as spare for the main processing nodes and also contains the backup of the LEDDB.
- node076 - node080 are nodes recently upgraded that are not functional yet. They should be working soon enough and they are supposed to be super-machines.

Using home directory and data disks

Each user has a home directory in */home/users/[user name]* that is shared by all the nodes. Do not use it for data storage or processing and never for intensive IO tasks (your life ...). If you want to store large data or do heavy IO tasks, do it in the /data directories of the nodes in a location like */dataX/users/lofareor/[username]*. You may need to ask the system administration (tiesinga@astro.rug.nl) to create this directory for you in the required nodes.

3 LOFAR EoR data

In order to find the EoR signal we require hundreds of observations. A LOFAR EoR observation, which is ten's of terabytes, consists on hundreds of measurement sets. These observations (their measurement sets) need to be processed (flagged, averaged, calibrated, etc.). There are applications for the processing of single measurement sets. There are also tools to ease the processing of a higher number of measurement sets.

Project data

The project data is stored in node011 to node074, in folders which names are in the format *L[YYYY_]XXXXX[_VVV]* where *XXXXX* is the observation identifier, *YYYY* is optional and it is the year of observing and *VVV* indicates the version of the data (if not specified it is 0). These folders are contained in the following directories:

- */dataX/users/lofareor/* for the raw data (version 0)
- */dataX/users/lofareor/pipeline* for the averaged data (version > 0)

The data out of these specified locations is not considered project data, we call it user data. Only experienced users should handle/process the project data. Please, ask before trying to use the project data.

The raw project data belongs to the user *lofardata* and only this user and the users in the linux group *lofareor* can modify it.

4 Data locations: The LEDDB

We use a database called LEDDB to store the locations of the project data (among other things).

LEDDB web UI

When accessing LEDDB web you will be asked for a user name and password, use the same ones that you use to access the LOFAR EoR cluster. Few remarks regarding what the tabs of the LEDDB web UI show:

- LDS: the different observations.
- LDSB: for each observation, the different beams.
- LDSBP: groups, for each beam of each observation, the several measurement sets that are in the same location (LOFAR EoR cluster or any other cluster or archive) and have the same averaging and processing properties.

- MS: for each LDSBP entry, the different measurement sets, but only their frequency information.
- MSP: for each MS entry, its data location.
- GAIN, QTS, QFS and QBS: diagnostic information extracted from the processing of the measurement sets.
- GAINMOVIE: locations of Gain solution animations.

Selection of rows in a tab will filter the results in other tabs in positions right of the current one. For example, if you select a row in the LDS tab and go to the LDSB tab, you will only see the LDSB rows related to the selected LDS row. In addition, you can filter and sort the rows.

The *RefFile*

As explained in the previous section, using the LEDDB web UI you can list the observations and the locations of their related data (measurement sets). You can also create a *RefFile*, which is a file that contains the locations of some measurement sets, by selecting some rows in the MSP tab and clicking on *Save RefFile* (you can also create it from other tabs). Be aware that when you create a *RefFile*, it is stored in a temporal location in the LOFAR EoR cluster and you must move it to your home directory (the temporal files are deleted on daily basis so do not complain if the *RefFile* is gone).

5 Data processing

The several applications used for the processing of the LOFAR EoR data are:

- For flagging: AOFlagger (it can be done in application NDPPP) via the command *rfi-console*.
- For averaging: NDPPP (usually we do averaging and flagging in single execution) via the command *NDPPP*.
- For calibration: BBS via the command *calibrate-stand-alone*, and SAGECAL via the command *sagecal* (in this case you also need to "*source /software/users/lofareor/eor-init.sh*").
- For imaging: CASAimager via the command *casapy*, and AWImager via the command *awimager*.

Processing many measurement sets

All the data processing programs presented above are meant to process a single measurement set. You should test them for a while to get a feeling on how they work. Once you have experience in processing a single measurement set there is a tool to help you processing hundreds of them with a single command. Again, only use this tool when you are "the master" of single measurement set processing.

The tool, which in general requires a *RefFile*, is available by using the LEDDB web UI, through the *Data Manager* button, or by using, in any node of the LOFAR EoR cluster, the command *ExecuteLModule* (use -h to get help). In fact, the web version is just a GUI that helps you filling the parameters required to run the command. Few comments on the tool:

- In general this tool will allow you to define some distribution options, i.e. how many simultaneous different nodes are processing measurement sets and how many different measurement sets are simultaneously processed in each node.
- Normally it also allows you to define a logging folder. Each individual execution to a measurement set will log its progress in a file in this folder. If you want to have all the log files in the same folder (which is the usual desire) you should use a location that is shared among all the nodes, for example within your home directory.
- In addition there is normally a query mode (usually option -q) that will only show all the commands (that would be executed to all the involved measurement sets) without actually executing them. It is much recommended to use this mode before each execution, specially when you are starting your trip with the LOFAR EoR group.

Cluster Monitor

There is a tool, the *Cluster Monitor*, that shows real-time information of the usage of the nodes. This includes CPU, memory, network traffic and data disks usage. The tool is available in the LEDDB web UI, through the *Cluster Monitor* button. It is also available using the command *ExecuteLModule ClusterMonitor* in any node of the LOFAR EoR cluster. This is particularly handy when you are running something in multiple nodes and want to see what is going on (why there is fire in the nearby building).

6 Examples

In the next paragraphs we present a few examples on how to process many measurement sets. We will use a test *RefFile* called *testrefffile.ref*.

Note that we do not specify how many measurement sets are pointed by this *RefFile* so whatever we explain in the following examples is valid for 5 measurement sets but also for 500.

6.1 Copying

Let's assume that the *testrefffile.ref* points to project data. Since we have to be especially careful with the project data, we will first make a copy of the data pointed by the *testrefffile.ref* and do the test processing with the new copies.

When copying the data, in addition to the input *RefFile*, you need to specify the nodes where the data will be copied and the new location in them.

```
ExecuteLModule CopyData -i testrefffile.ref -o myrefffile.ref
-u node001-004 -c /data1/users/lofareor/martinez/test -p 1 -n 2
```

The execution of the previous command copies all the measurement sets pointed by *testrefffile.ref*. The new measurement sets, which locations will be described in *myrefffile.ref*, are distributed in 4 nodes (node001 to node004) and they are copied in the folder

/data1/users/lofareor/martinez/test. Only two nodes are simultaneously receiving new measurement sets and only one at a time.

6.2 Flagging

For flagging the data we normally use NDPPP (adding a flag step in the NDPPP parset). However, there is also a command to "only" run the flagger. In addition to the input *RefFile* you need to specify an execution path where some temporal files are created. Do not use the home directory for this purpose.

```
ExecuteLModule LaunchAOFlagger -i myrefffile.ref
-e /data1/users/lofareor/martinez/tempflag -l logsflag
```

The execution of the previous command runs the AOFlagger to all the measurement sets pointed by *myrefffile.ref*. The used nodes are the ones that contain the data. Each of these nodes will use the folder */data1/users/lofareor/martinez/tempflag* as execution path (it is created if it does not exist). In this case all the processes running in the used nodes will log their progress in *logsflag*.

6.3 Averaging (and flagging) with NDPPP

NDPPP requires a parset file that describes where is the measurement set and what processing must be done. Hence, in order to run NDPPP on many measurement sets, first you need to create parset files for all the measurement sets to be processed. There is a command that given a template parset will create a new parset file for each measurement set pointed by a *RefFile* .

```
ExecuteLModule CreateNDPPPParsetFiles -i myrefffile.ref
-t /home/users/martinez/NDPPP/template.parset
-o /home/users/martinez/NDPPP/parsets -c /data3/users/lofareor/martinez/averagetest
```

The execution of the previous command will create a parset file for each measurement set pointed by *myrefffile.ref*. The new parset files are stored in */home/users/martinez/NDPPP/parsets* and they are copies of the template */home/users/martinez/NDPPP/template.parset*, only that, in each case, it changes the part describing the location of the data. The input measurement set location is given by the input *RefFile* and the output measurement set will be stored in the specified path, i.e. */data3/users/lofareor/martinez/averagetest*. Once the new parsets are created you can use:

```
ExecuteLModule LaunchNDPPP -i /home/users/martinez/NDPPP/parsets -l logsndppp
```

The execution of the previous command will execute NPPPP for all the parsets that we just created. The output data will be written in the same nodes than the input data (in the path indicated when creating the parsets). If you wish to write the output data in different nodes you can specify it when running the *ExecuteLModule LaunchNDPPP*. In this example all the NDPPP processes will log their progress in *logsndppp*.

After the completion of the previous execution, you will have created new data. You may need a new *RefFile* to do more processing to the data you just created:

```
ExecuteLModule CreateRefFileFromPath -i /data3/users/lofareor/martinez/averagetest
 -s node001-004 -o myreffileavg.ref
```

The execution of the previous command will create a new *RefFile* with all the measurement sets in node001 to node004 contained in the path */data3/users/lofareor/martinez/averagetest*. The new *RefFile* is called *myreffileavg.ref*.

6.4 Calibrating with BBS

Running BBS is slightly easier than NDPPP because it does not require to create a parset for each measurement set. It does need a parset file but the same one is used for all the measurement sets. In addition to the input *RefFile*, the parset file and the sky model file, you need to specify an execution path where some temporal files are created. Do not use the home directory for this purpose.

```
ExecuteLModule LaunchCalibrateSA -i myreffileavg.ref
-e /data1/users/lofareor/martinez/tempbbs
-a /home/users/martinez/BBS/parset -m /home/users/martinez/BBS/sky -l logsbbs
```

The execution of the previous command calibrates all the measurement sets pointed by *myreffileavg.ref.* The exact calibration is defined in the parset */home/users/martinez/BBS/parset.* The used sky model is read from the file */home/users/martinez/BBS/sky.* In each node with data it will use the folder */data1/users/lofareor/martinez/tempflag* as execution path, so this path will be created in all the used nodes. In this case all the processes running in the used nodes will log their progress in *logsbbs.*

6.5 Calibrating with SAGECAL

SAGECAL needs, in addition to the input *RefFile*, a sky model file (like in BBS) and a cluster file.

```
ExecuteLModule LaunchSagecal -i myreffileavg.ref
-s /home/users/martinez/SAGECAL/sky -c /home/users/martinez/SAGECAL/cluster
```

The execution of the previous command calibrates all the measurement sets pointed by *myreffileavg.ref.* The used sky model is read from the file */home/users/martinez/SAGECAL/sky* and the cluster information is extracted from the file */home/users/martinez/SAGECAL/cluster.*

6.6 Imaging with the CASAImager

In this example we do imaging with the CASAImager even though it is not the only imager available. Like NDPPP, the CASAImager requires a parset file that describes where is the measurement set and how the imaging must be done. Hence, in order to run CASAImager on many measurement sets, first you need to create parset files for all the measurement sets. There is a command that given a template parset will create a new parset file for each measurement set pointed by a *RefFile.*

```
ExecuteLModule CreateCASAImagerParsetFiles -i myreffileavg.ref
-o /home/users/martinez/CASA/parsets -t /home/users/martinez/CASA/template.parset
```

The execution of the previous command will create a parset file for each measurement set pointed by *myreffileavg.ref.* The new parset files are stored in */home/users/martinez/CASA/parsets* and they are copies of the template */home/users/martinez/CASA/template.parset*, only that, in each case, it changes the part describing the location of the input data. The input measurement set location is given by the input *RefFile.* The location of the generated images is specified when running the CASAImager. Once the new parsets are created you can use:

```
ExecuteLModule LaunchCASAImager -i myreffileavg.ref
-p /home/users/martinez/CASA/parsets -o /data3/users/lofareor/martinez/images
-l logscasa
```

The execution of the previous command will execute CASAImager for all the parsets that we just created. The output images will be written in the same nodes than the input data in the path */data3/users/lofareor/martinez/images*. If you wish to write the output images in different nodes you can specify it when running the *ExecuteLModule LaunchCASAImager*. In this example all the CASAImager processes will log their progress in *logscasa*.