Handling the LOFAR EoR data

Oscar Martinez
University of Groningen
Kapteyn Astronomical Institute
Groningen
The Netherlands
September 9, 2013

Contents

1 Introduction

This documents describes the tasks related to the management and processing of the data related to the LOFAR EoR project. For more information on the software package required for these tasks see the document *LEDAMA Software Package*. For more information on the database used for storing the data locations and the diagnostic information see the document *LEDDB description and usage*.

Finding the EoR signal requires thousands of hours of observations which will translate into few petabytes of data to be processed. Each observation (which can last from 6 to 16 hours) consists on few hundreds measurement sets. For more information regarding the measurement sets see the document *MeasurementSet description for LOFAR*.

During the main observing periods (from October to June) there can be up to 4 observations per week. An every year we have few hundreds hours of observations.

The size (including all the measurement sets) of a single observation, as it comes from the observatory (raw data), can be up to 50 TB. First of all, each raw observation needs to be averaged and flagged as soon as possible in order to avoid accumulating too many raw observations. Afterwards they also need to be processed (calibrated, imaged, etc.). In section **??** we describe the several standard processing steps.

2 Data flow

The data from the stations is sent to the Central Processing Facility (CEP) located in Groningen, the Netherlands, where it is correlated among other processing steps. After processing the data is stored in the Long Term Archive (LTA) in Groningen. From the LTA we copy the data to the LOFAR EoR CPU/GPU cluster, also in Groningen, where we process it with the LOFAR EoR pipeline (described in section **??**). The LEDDB takes care of storing the locations of the data both in LTA and the LOFAR EoR cluster, it also stores all the diagnostic data produced by the pipeline. Since we can not keep all the data in the LOFAR EoR cluster, we must archive it in the LTA but thanks to the LEDDB we still have access to all its diagnostic information.

We mainly use the Target archiving facilities as LTA.

3 Involved systems

3.1 LOFAR EoR cluster

The LOFAR EoR cluster is a CPU/GPU computer cluster located in the Duisenberg building which is part of the Zernike Complex in the University of Groningen. It has (as on date 30/08/2013, it will soon we upgraded) the following components:

- 2 log-in nodes:
    - *lofareor01*: for all users
    - *lofareor05*: only for root
- 3 gateways:
    - *gateway1*: 1 Gbps to RuG network

– *gateway2*: 1 Gbps to RuG network

– gateway: 10 (20) Gbps to Target network (removed and being replace by new gateway - DB server)

- 80 CPU/GPU nodes (*node001 - node080*):

    – 8 cores (16 threads) Intel Xeon 2.27GHz

    – 12 GB RAM

    – 2 GPU NVIDIA Tesla M1060

    – 3 disks:

        * */data1* 2TB (1.7 TB)

        * */data2* 2TB (1.7 TB)

        * */data3* 3TB (2.8 TB)

Only 64 nodes (and their disks) are used for project data (*node011 - node074*). This means that the current total storage capacity of the LOFAR EoR cluster for project data is around (1.7 + 1.7 + 2.8) TB x 64 = 396.8 TB. It is worth saying that the past experience has shown that the *data2* disks are failing quite often so its usage is not recommended. We could say that, in total, only 288 TB are trustworthy.

3.2 Target

Target currently contains more than 1 PB of LOFAR EoR data so it is the main used archiving facility. The contact people for archiving LOFAR EoR data in Target are Hanno Holties (*holties@astron.nl*) and Adriaan Renting (*renting@astron.nl*).

Target archiving is separated in TIERs. The archiving of the LOFAR EoR data is mainly done in the following ones:

- TIER F (400 TB in disks): It is the location where the raw data from the observatory is stored. From here it is transferred to the LOFAR EoR cluster. In principle the raw data should be deleted when it has been processed (and the averaged versions have been created and archived)

- TIER E (1.5 PB in disks): This is the archive of the averaged/processed observations.

Target also contains a tape system. The data should be taped and this process should be transparent to us (a measurement set is on tape if it is a tar file).

Currently the communication / transfers between Target - LOFAR EoR cluster is though a 10 (20) Gbps line. We use the FDT tool for the transfers (*http://monalisa.cern.ch/FDT/*).

There is also some old data in another location in Target (the one called TARGET_E_OPS).

3.3 Other locations

There is also old data in SARA (Amsterdam) and Juelich (Germany).

4 Synchronization of the LEDDB

The locations (cluster / host / path) of the data are stored in the LEDDB:

- The locations of the data in the LOFAR EoR cluster are updated every day. There is a cron job in each node that updates the LEDDB with all the measurement sets in the standard locations. See section **??** for more information on the standard locations.

- The locations of the data in both TIER E and TIER F of Target are updated every morning. There is a cron job in the LEDDB server for this purpose. Since we can not *ls* in Target we require that in the root directory of both TIERs there is a file listing the different folders. There is a folder for each processed version of an observation. See section **??** for more information on the folder structure in Target. Each of these folders will also contain a file listing the several measurement sets stored in it. We use those files (the one in the root directory and the ones in each data folder) to fill the LEDDB.

- The data in the other used location of Target (TARGET_E_OPS) is old and their locations are not synchronized. They were manually added using some *LModules* (See *LEDAMA Software Package* for more information regarding the *LModules*). Whenever they are deleted they will have to be manually removed from the LEDDB (using the *RemoveFromLEDDB LModule* specifying the LDSBPs identifiers).

- The data in the other locations, SARA and Juelich, were also manually added so they will also need to be manually removed if ever deleted.

IMPORTANT: Take into account that any transfer done during the day or any changes in the measurement sets will not be reflected in the LEDDB until next morning.

5 Processing steps

Here we present the standard steps in the processing of an observation:

- The observation is performed, its data is correlated and moved to CEP. This data is 64ch2secs.

- Transfer data (64ch2secs) from CEP to Target. This is done by people out of the LOFAR EoR group. Note that the locations of the new data in Target will not appear in the LEDDB until next morning. You can use the *CreateRefFileFromTarget LModule* if you need the locations before then.

- Transfer data (64ch2secs) from Target to LOFAR EoR cluster using the *CopyData LModule*. There is also a script to help you here (*gencopyeor2target.py*).

- NDPPP: flag and average from 64ch2secs (version 0) to 15ch2secs (version 1). Quality data is created in 15ch2secs version with high resolution (64ch2secs). You can use the *LModules CreateNDPPPParsetFiles* and *LaunchNDPPP*.

- NDPPP: average from 15ch2secs to 3ch2secs (version 2). Quality data is automatically carried on to 3ch2secs. Since we do not need this the LEDDB always ignores quality data when version is not 0.

- Transfer (archive) 15ch2secs from LOFAR EoR cluster to Target using the *CopyData LModule*. There is also a script to help you here (*gencopytarget2eor.py*).

- Delete 64ch2secs from LOFAR EoR cluster (using *DeleteData LModule*) and send message to LOFAR LTA people (Hanno and Adriaan) that may delete the 64ch2secs also from Target.

- NDPPP: average from 3ch2secs to 1ch10secs (version 3). Like in the version 2 the quality data is ignored by the LEDDB.

- Delete 15ch2secs from EoR (the user who created the data must execute this since he owns the data).

- Calibrate the 3ch2secs (version 2) and 1ch10secs (version 3) with BBS and/or SAGECAL.

- We recommend to delete the MODEL_DATA column in the calibrated versions (this decreases 33% of their size).

- Transfer (archive) 3ch2secs from LOFAR EoR cluster to Target.

- Delete 3ch2secs from LOFAR EoR cluster (the user who created the data must execute this since he owns the data).

- Transfer (archive) 1ch10secs from LOFAR EoR cluster to Target.

- For now we will keep the 1ch10secs in the LOFAR EoR cluster.

Versions and sizes

Here we present a table with information regarding the different versions that we are handling.

Version	t	Nch	res	Cal	Q	QT	QNch	Size[TB]	SizeC0[PB]	Arch
0	2	64	3	N	N	-	-	36	2.3	N
1	2	15	12	N	Y	2	64	13	0.8	Y
2	2	3	61	Y	N	2	64	7	0.5	Y
3	10	1	183	Y	N	2	64	1	0.1	Y

Where:

- *Version* indicates the version index that the data should have in the LEDDB.

- t is integration time in seconds.

- Nch is the number of channels.

- res is frequency resolution in KHz (we assume SB bandwidth is 183 KHz).

- Cal indicates if data should be calibrated.

- Q indicates if the Quality data is taking into account by the LEDDB.

- Qt is the time resolution of the Quality data.

- $QNch$ is the number of channels of the Quality data.

- About the sizes:

  – 8bit mode is assumed (so 488 SBs).
  – $Size$ is the size for an observation (assuming 12 hours).
  – $SizeC0$ is the expected size for the whole Cycle 0 observations (784) hours.

- *Arch* indicates if data is archived.

So, for the whole Cycle 0 we would be storing approximately **1.4 PB**.

Note that when the data is calibrated its size gets multiplied by factor 3 (if we store both CORRECTED_DATA and MODEL_DATA columns) or 2 (if we only store CORRECTED_DATA).

6 Rules

6.1 In LOFAR EoR cluster

Here are some rules that everybody processing data in the LOFAR EoR cluster must follow. These rules apply to the project data, not to the user-generated copies.

- There are two types of project data :

  - RAW: data with same average properties as provided by the observatory, the extension of these MSs must be .MS (not .dppp). RAW data is stored in node011-node074 in */datax/users/lofareor/LYYYY_IIIII* or */datax/users/lofareor/LIIIII* where x is 1,2 or 3, YYYY is the year and IIIII is the LDS (the same as in the LEDDB).
  - PIPELINE: data that has been somewhere averaged. The extension must be .dppp. PIPELINE data is stored in */datax/users/lofareor/pipeline/LYYYY_IIIII[_VVV]* or */datax/users/lofareor/pipeline/LIIIII[_VVV]* where [_VVV] indicates the version number of the data. Please, whenever you want to create a new version, check in the LEDDB that you use a version number that is not being used. Version number must be 3 digits. This version number helps the LEDDB to understand that this data is different from other datasets.

- Normally, for each observation, there are 488 SBs (assuming 8-bit mode). These SBs are distributed over node011-node074 (DO NOT USE node001-node010 nor node075-node080). The nodes that are used in each case depend on the processing requirements, the cluster availability and the several users needs:

  - With high processing requirements and low users needs ALL the nodes (i.e. node011-node074) are used for the distribution. In this case, there 8 SBs per node.
  - With high users needs we suggest to use node011-node042 for NCP related data and node043-node074 for data related to ALL the other fields. In this case, there are 16 SBs per node.
  - node079 and node080 can be used as spare nodes if some of the latter nodes is not available. In such case pay attention to other users also using these nodes.

- There are many users that want to use the cluster for processing purposes. We created a Google Drive spreadsheet for this purpose:

  `https://docs.google.com/spreadsheet/ccc?key=0Ag6OGS8DewSRdHJmeUNHOURlSkhORmVqRjREdUNtWkE#gid=0`

  Use your *gmail* account or the *lofareor gmail* account. If you have any problems accessing this table contact Vibor Jelic (*vjelic@astro.rug.nl*).

  In this table the several users planning to use the cluster should write down which nodes they need, for how long and how intensive will be their usage. This applies to any major processing in the processing nodes, i.e. *node011-node074*. The table has several columns:

  - user: the user who will run the task.
  - task: a minor description of the task that will be running in the nodes.
  - nodes: the used nodes.
  - start_time: planned starting time. Please format as YYYY/MM/DD/hh:mm:ss.

- end_time: estimated end time. Same format as start_time.
- percentage_cpu: estimated percentage of CPU used.
- percentage_mem: estimated percentage of Memory used.
- finished: Indicates if the task is finished.

The estimated CPU and Memory usages are used to allow the users to know if some node can be used for a new task even if it is already in use. This could be the case when running tasks that are not very consuming. In all cases we recommend that you directly contact the user who is running the tasks (which you would overlap).

Do not remove the lines of tasks that have finished. You should just specify that is finished (with the corresponding column). In this way we will have a history of the different tasks.

In all cases, during the weekly data processing meetings we will discuss which user has the maximum priority and over which nodes.

- All the data is owned by *lofardata* (with some exceptions). By default only *lofardata* can modify the data but the write permissions can be extended to one additional user. When you want to work with some data these permissions should be extended to your user (contact Oscar, Sarod or Vibor for that). Then you will also have permissions to write to the data (but only you).

- Extension of permissions to certain users can be done with *ExecuteLModule ExtendPermissionsData -i reffile -u [username]*. This must be executed by the owner of the data, normally *lofardata* user (only Oscar, Sarod and Vibor will remain as *lofardata*).

- Hence, do not use *lofardata* to process stuff. *lofardata* should only be used as an owner of the raw data and to run all the code for the LEDDB synchronization.

- If you decide you want to create a new averaged version of the data (with NDPPP) do it with your normal user. If you belong to lofareor group (meaning the linux permissions group) you should be able to write in the location. Remember you should write the new MS into a path like */datax/users/lofareor/pipeline/LYYYY_IIIII_VVV* (REMEMBER THE VERSION!).

6.2 In Target

The locations that are stored in the LEDDB from the data in Target are relative to the location where the FDT servers are running.

- The raw data of an observation will be in the TIER F of Target and the (relative) location is *LIIIII* where IIIII is the LDS (the same as in the LEDDB)

- The several averaged versions must be archived in TIER E of Target and the (relative) location is *LIIIII_VVV* where IIIII is the LDS and [_VVV] indicates the version number of the data (as in the LEDDB and the LOFAR EoR cluster).

Since we do not have direct access to Target any action on the data there can only be done by Hanno Holties or Adriaan Renting.

7 Tips

7.1 Related to LEDDB and the web UI

- If you create a new average copy of the data (with NDPPP) the new data will not appear in the LEDDB until next day (we synchronize the LEDDB every night). But you can use *ExecuteLModule CreateRefFileFromPath* to create a reffile from the path where you wrote the data. For example:

```
ExecuteLModule CreateRefFileFromPath -i /data1/users/lofareor/pipeline/L68888_005
 -s node011-074 -o mynewreffile.ref
```

- About the LEDDB tabs:
  - LDS is only the name of the observation (and some basic information about it).
  - LDSB shows the beams of an observation.
  - LDSBP: THIS IS THE SET OF DATA!. For the same observation we will have different LDSBP depending on the storage place, the averaging properties and the version (as specified in the path, i.e. _VVV).
  - MS is just frequency information of the MSs that form a LDSBP.
  - MSP: THIS IS THE DATA, i.e. these are the MS files/folders. Whenever you create a reffile, this will contain as many lines as selected rows in the MSP table!.

- About general usage:
  - You can select all rows by clicking in the first cell of the header row.
  - If you click the header of the tables the data is sorted with that column.
  - When you make a selection in a tab and you change to another tab in a position right of the last one, the shown rows will be related to whatever you selected.
  - About Filters: There are two types of filters.
    * Secondary-table filters: You can access them (PROJECT, FIELD, ANTENNA_TYPE, STATION, BASELINE, STORE, STORE_HOST and QUALITY_KIND) through their related button. Important: They are persistent in all tabs.
    * Column filters: They only affect the current tab.
  - Remember that the LEDDB also contains all the data in the various archives, i.e. Target, SARA (Amsterdam) and Juelich (Germany). So, when creating your reffiles, go to LDSBP and use the filter to see data only where STORE=EoR (you can only access this data!). Once you have selected your LDSBP go to MSP tab and see the number of rows. Select all of them and store the reffile.
  - Whenever you create a reffile and want to start doing something with it, your first step should be to have a look at the reffile itself. Are its contents what you expected?

7.2   Related to LModules

- The data manager in the LEDDB web does the same than the *ExecuteLModule* command.

- You can always type *ExecuteLModule -h* to see all the available *LModules*.

- *ExecuteLModule CheckData*: it checks if the data from a *RefFile* is where the *RefFile* points it is. Do this before you start anything.

- *ExecuteLModule MSInfo*: It shows information (number of channels, integration time, number of stations, etc) about the data in a *RefFile*.

- Before you execute some intensive task (like NDPPP, BBS...) check the *ClusterMonitor* to see if the nodes that you want to use are not already being heavily used.

- In general all the *LModules* have the option -q. Use it to get an idea of what commands are going to be executed. The option -q only shows the command, so it does not execute anything.

- In *CopyData LModule*, it is recommended that you first do a run with options -q and -o. This will create a *RefFile* with the future locations of the copies that you are going to perform, without actually executing the copy commands. Are these locations what you expected?

- In general a *LModule* is just distributing tasks over some nodes, so it does multiple executions of the same command (NDPPP, calibrate-stand-alone, etc...) in different nodes. So, when you use a *LModule* some internal subprocesses are created (normally one process per SB). If you cancel or kill the main *LModule* process (with ctrl-c for example) it is possible that there are still some subprocesses running in whatever nodes you were using, and, in some cases, you have to manually kill these processes. For this purpose you can use the *LModule ShowProcesses*.

8   Examples

In this section we will go through a realistic example of the data management and processing of an observation. We also detail what things may go wrong and try to offer solutions for them.

8.1   Observation is performed

There is a new observation done last night. The first thing we should get is a message from Michiel Brentjens (or someone from the Science Support group). The subject of the message will be something like *LC0_019: L80897 successful* (from now one we assume this observation for the rest of the example).

Basically this means there is a new observation which data is currently flowing to Target (from CEP). However, there can be several important details in the message (missing data, malfunction of stations, etc) so you better have a quick look to it.

8.2 Checking data in Target

If the transfer from CEP to Target was done more than a day ago, the data should be in the LEDDB. If this is the most recent observation it should appear in the last row of the LDS tab in the LEDDB web. Select the row *L80897* and go to LDSBP. You should only see one line for each beam (of the raw data in TARGET_EOR_F). Select all the beams and create a *RefFile* or create different files for different beams. Normally what we do is to create a *RefFile* for the beam 0 and another *RefFile* for the rest of beams.

Since the LEDDB is only synchronized during the night, it could happen that the data is already on Target but still not visible in the LEDDB. If you think this is the case you can use the *LModule CreateRefFileFromTarget*. The following command creates a *RefFile* without using LEDDB, i.e. directly requesting the data from Target (from TIER F more concretely):

```
ExecuteLModule CreateRefFileFromTarget -s TARGET_F_EOR -i L80897 -o L80897.ref
```

Whether you used the LEDDB or the *LModule CreateRefFileFromTarget*, when the *RefFiles* are created you should check that they have the number of measurement sets that you expect and that their sizes are coherent.

In the whole process you can have the following problems:

1. It is more than a day ago that the data is in Target but it does not appear in the LEDDB.

2. You try to get the *RefFile* with the *LModule CreateRefFileFromTarget* but you get errors.

3. You get the *RefFile* but it does not have the content you expected.

Possible reasons (and suggestions for their solving) for these problems are:

- The reason for problem 1 can be that the LEDDB is not being updated. Check the daily logs of the LEDDB Target update in

  /home/users/lofardata/martinez/logs/leddbupdate/TARGET_FDT

- The reasons for problems 1 and 2 can be that the file with the measurement set locations (*L80897 /filelist.txt*) could not get copied from Target which can mean:

  - There is something wrong with the line Target - LOFAR EoR cluster, either some machine in Target is down or the LOFAR EoR gateway is not working fine. Try the following:

    * ping lotar1.staging.lofar. If the ping fails ask help (Eite Tiesinga, Hanno Holties and Adriaan Renting).
    * Manually get the *filelist.txt* with the following command:
      ```
      java -jar /home/users/lofardata/martinez/software/fdt.jar
      -p 20002 -noupdates -silent -c lotar1.staging.lofar -pull
      -r -d . L80897/filelist.txt
      ```
      Try to see what is failing and ask for help.

  - The data has not started to flow yet. If it should be there, ask for help. If it is maybe to early, be a bit patient.

  - The data started to flow but the *filelist.txt* is still not generated (it is generated every hour). Also be patient or ask for help if it is long time.

- The reasons for problem 3 can be:

  - The transfer is not yet finished but it is on its way.
  - The transfer is finished but with errors.

  In any case, try again after one hour and if you get the same result contact Hanno Holties or Adriaan Renting and tell them the CEP-Target transfer may have errors.

8.3 From Target to LOFAR EoR cluster

Let's assume that the data is in Target and that you have created two *RefFiles*, one for the first beam called L80897_B0_TARGET.ref and another for the rest of beams called L80897_B1-6_TARGET.ref. The format of the name of these files is important if you want to use the provided scripts in LEDAMA package. The format must be LXXXXX[_VVV][_BYY]_TARGET.ref where [VVV] is optional and is for the version (if not specified we assume version 0, which is the case when dealing with raw data like in this example) and BYY is also optional and is for the beam (but if you specify it, it must contain the B).

Now, assuming that you want to distribute the data in node011 - node042 (before executing the transfer process you should check that the nodes receiving the data have space enough), you can use the following script to generate the copy commands:

```
/home/users/lofardata/martinez/software/ledama/datamanagement/scripts/gencopytarget2eor.py
 -r L80897_B0_TARGET.ref,L80897_B1-6_TARGET.ref -n 11-42 &> txL80897.sh
```

Check the generated script and run it to start the transfer of the data.

Once the transfer is running you can use the *LModule CheckData* to see its status. Each *CopyData* command will show you the log folder location.

You can also use the *Cluster Monitor UI* in the LEDDB web to see the network traffic in the nodes.

IMPORTANT: We always run the transfer from Target to LOFAR EoR cluster with the *lofardata* user and you can run in any node of the LOFAR EoR cluster (except the lofareor01 that does not have the proper libraries).

Several things may go wrong during this process:

1. There is some strange error message that some code is not found.

2. You start a transfer but both *CheckData LModule* and *Cluster Monitor UI* show than no data is being received.

3. The transfer have finished but when I do a last checking with *CheckData* it shows that the data is not completed (i.e. it does not show a 100 %)

Possible reasons (and suggestions for their solving) for these problems are:

- The reason for problem 1 is maybe that you are running the script in *lofareor01*. You must use any node of the node001 - node080.

- The reason for problem 2 maybe that the LOFAR EoR - Target line is not working. Check the log files in the log folder (you can find the log folder in the generated copy command) and try to see errors. You will probably have to end up contacting Eite Tiesinga (if you think the error in due to LOFAR EoR cluster) or Hanno Holties and Adriaan Renting (if the problem seems to be in Target site).

- The reasons for problem 3 may be:

  - Some node/disk has not space left. Delete the incomplete SBs. Create a new *RefFile* with the missing SBs (you can use the LEDDB or the *CreateRefFileFromRefFile LModule*) and use *gencopytarget2eor.py* or directly the *CopyData LModule* to get the missing SBs. Important: remember this time to select nodes with space to allocate the new data.

  - Some node has a broken disk. If you suspect that a node may have a broken disk (usually data2 disks) go to the node and test to write a file in it (for example *touch /data2/users/lofareor/testfile*). If you get an error message that disk is Read-Only it means the disk is broken. Go to section **??** to learn how to deal with a broken disk.

  - There were few errors in few processes because some processes got accidentally killed or some punctual error in Target. You can use the same *CopyData* command with the option –resume. This will copy only the SBs that are not already where they should be.

8.4   Observation with calibrators

In certain fields (usually the NCP) for each observation (6-16 hours) there are also two calibrators (short observations in surrounding fields), one before and one after the main observation.

Even though the calibrators are not processed we also copy them in the LOFAR EoR cluster (so the LEDDB can register their meta-data) using the same procedure that for the main observation. However in this case we do not separate data by beams.

8.5   Data processing

Once the data is completed in the LOFAR EoR cluster (and we can check it with the *CheckData LModule*) it is the time for processing. As indicated in the section **??** the standard processing steps are:

- NDPPP (flag and average) from 64ch2secs (raw data) to 15ch2secs.

- NDPPP (average) from 15ch2secs (raw data) to 3ch2secs.

- NDPPP (average) from 3ch15secs (raw data) to 1ch10secs.

- Calibrate the 3ch2secs and 1ch10secs data.

For all these processing steps there are *LModules*, in particular *CreateNDPPPParsetFiles*, *LaunchNDPPP*, *LaunchCalibrateSA* and *LaunchSagecal*.

8.6   Data archiving and deleting

The standard procedure is that the 15ch2secs and the 3ch2secs data is archived. In some cases we also archive the 1ch10secs but we keep a copy in the LOFAR EoR cluster. When the 15ch2secs data is archived we can delete the raw data from both the LOFAR EoR cluster and Target .

In order to archive the data we need to create the *RefFiles*. For this purpose we can use the LEDDB or the *CreateRefFileFromPath LModule*. To create from the LEDDB select

the appropriate row in the LDS tab and go to the LDSBP tab. Now you will see many rows. They are the different groups of measurement sets depending on their location and their processing stage, which should be linked to the version, remember: version 0 for the raw data, i.e. 64ch2secs, version 1 for the 15ch2secs, version 2 for the 3ch2secs and version 3 for the 1ch10secs. They are ordered by storage and version. In the case of archiving the 15ch2secs, select all the rows of version 1 and location EoR. See the column #MSPs to see it is the number you expect. Create the *RefFile* from these selection. Do the same for the rest of versions you want to archive.

At this point what may happen is that in any version there are less measurement sets than expected or that their sizes are strange. In this case contact the person in charge of the processing and let him know.

Once you have the *RefFiles* of the data you want to archive you can use the following script to generate the copy commands:

```
/home/users/lofardata/martinez/software/ledama/datamanagement/scripts/gencopyeor2target.py
 -o L80897 -a L80897_001.ref -b L80897_002.ref -c L80897_003.ref  &> archL80897.sh
```

Before executing this script you should check that there is space enough in Target (TIER E where the processed data is archived by default). For this you can use the *Cluster Monitor UI* in the LEDDB web to see the storage in Target.

Like in any other copy process, once it is running you can check the status with *CheckData LModule*.

The problems that may arise during these process are the same that in the case of the Target to LOFAR EoR cluster transfer. Basically problems in the connection. Also if Target runs out of space you should contact Hanno Holties or Adriaan Renting.

8.7 Generating a Gain movie

For all the measurement sets in the LEDDB (in the standard locations), their diagnostic data is automatically stored in the LEDDB. But the generation of Gain movies is not automatically done so it is required to generate them from time to time. There is a script to generate Gain movies from LDSBP identifiers.

First thing you need to go to the LEDDB web UI, and look for LDSs that have Gain solutions but do not have Gain movies. Select the ones you want to create movies from and go to LDSBP. In this tab you need to filter rows in order to get only the rows that are related to measurement set that have Gain solutions. This usually means you need to add a Filter to see only data in EoR cluster, you also should add a Filter to see only rows with *hasGain = True*. And normally we only generate movies of the data related to the first beam (0) so you can also add a Filter for this purpose. After adding all the filters select the rows from which you want to generate the movies. In the upper left page there is a button (arrow-like) that, if you press it, will show all the LDSBP identifiers that you have selected. Copy them and use them with the following script:

```
/home/users/lofardata/martinez/software/ledama/dataanalysis/scripts/gengainmovies.py
 -i [LDSBP ids] -o /data3/users/lofareor/movies
 -e /home/users/[user name]/temp_animations -n 11-42  &> archL80897.sh
```

You should always run this script in node007 (and use the output folder *data3/users/lofareor/movies*). In each execution use the nodes (in the example node011 - node042) that are free.

8.8 How to deal with large diagnostic data

When the time passes and the database grow it may happen that the queries to the diagnostic tables are to slow. Since the diagnostic tables are partitioned, the recommendation is that the old partitions that are not used are emptied from the database. Obviously if we ever want to use them again you could refill them from the backup of that specific partition. We also recommend that in the case of emptying one partition you create a copy of its backup (in a different node/disk that the normal backup).

8.9 How to deal with a broken disk

In this section there is a small manual of how to deal with a broken disk.

The first thing you need to do is to copy the data from the broken disk to another empty disk. For this you will probably need to physically put a new disk in a node. Follow these steps as root (we assume a new disk has been put in a node but it does not contain a partition nor a filesystem):

- *fdisk -l* in the node with the new disk to find which *sdx* is the new disk

- *fdisk /dev/sdx* (change the *x* with whatever you found in first step). This is to create a new primary partition. You need to go through the helper. Press *n* for new partition, *p* for primary, *1* as the number and *w* to write it.

- *mkfs.ext3 -L /dataY /dev/sdxx -m 0* (change *xx* and *Y* with the proper values). This creates the file system.

- *mount /dev/sdxx /dataY -o acl*. This mounts the new disk

- *mkdir /dataY*. Creates the */dataY* directory (maybe not necessary)

- *mkdir -p /dataY/users/lofareor/pipeline*. Create the directory for the lofareor raw data and the pipeline data.

- *mkdir -p /dataY/users/lofareor/images*. Create directory for images

- We still need to set the proper permissions:

  cd /dataY/users

  chown -R lofardata:lofareor lofareor

  chmod -R g+rws lofareor

Once the disk is in its place you need to copy the data in the broken disk (only data in folders starting with *L*, the pipeline and the images directory) to the new spare disk (also with root user and using cp –preserve to preserve the ownership of the files)

When the data is copied you need to go to the LOFAR EoR building (physically) and put the spare disk in the position of the broken disk and take the broken disk to repair (give it to Eite Tiesinga)

IMPORTANT: */etc/fstab* must be coherent in the moment of booting, so for each node the missing disks should be commented in */ets/fstab* or the new disks should be added if they are ready to be used.

8.10 How to generate SIPs

In some moment in the close future we will be asked to provide for each measurement set that we want to archive in the LOFAR LTA, a XML file that contains information on the data to be archived. The code to handle all this is already implemented and it is working.

The idea is that if we want to archive want measurement set we also need to provide this XML file, which is called SIP.

When a raw measurement set is created in the observatory a new SIP is also created. So, the what we need to do is get the SIP related to the raw measurement set and update it with the required information. This information basically explain the several processing steps that have been applied to the measurement set that is archived.

Right now you could get a single SIP from a raw observation by using the MOM (*https://lofar.astron.nl/mo* but there are some problems with permissions. Also it is a pain when you want to get all the SIPs for all the measurement sets. The LOFAR LTA people is working on this. Right now we need to do a copy-paste the table that shows the data products of an observation.

Then, you can use the following tool to retrieve the SIP related to the raw data of the observation from which you want to archive some data:

```
ExecuteLModule GetSIPs -i table -p LC0_019 -o sips
```

This script requires the copy-pasted table (from which we will read the data product IDs). It can also work by providing the observation ID used in the LTA catalog (*http://lofar.target.rug.nl/*) but since the catalog is almost empty it is useless.

Assuming that you have managed to get those SIPs, now you need to "update" them in order to reflect the several processing steps that have been applied to the data.

For this you need a *LDSBPUpFile*. This file is an ASCII file that you can edit to detail the processing steps. You can generate a proposal of *LDSBPUpFile* with the command:

```
ExecuteLModule CreateLDSBPUpFile -i [LDSBP ID] -o ldsbpupfile
```

You need to provide the ID of the LDSBP that your data belong to. After executing the command it will generate a *LDSBPUpFile* that you can edit to modify information. Once this file reflect the changes that you have done you are ready to create the "updated" SIPs:

```
ExecuteLModule CreateSIPs -i reffile -s sips -o osips -f ldsbpupfile
```

Acknowledgements