FP7- Grant Agreement no. 283393 – *RadioNet3*

Project name: Advanced Radio Astronomy in Europe

Funding scheme: Combination of CP & CSA

Start date: 01 January 2012 Duration: 48 month

Deliverable D10.2

Report on the comparison of data formats, specifying the key characteristics of optimal formats for various phases in the imaging chain, indicating where and how readily available solutions can be applied.

Due date of deliverable: 2013-06-30

Actual submission date: 2013-06-28

Deliverable Leading Partner: THE CHANCELLOR, MASTERS AND SCHOLARS OF
THE UNIVERSITY OF CAMBRIDGE (UCAM)

An European project supported within the 7th framework programme

SEVENTH FRAMEWORK
PROGRAMME

1 Document information

Type	Report
WP	10
Authors	Ger van Diepen, ASTRON (for UCAM)

1.1 Dissemination Level

Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

1.2 Content

2 Introduction

2.1 Purpose of this document

As part of workstream 1 in JRA Hilado, an analysis of data formats has been made. This report gives the results of that analysis and a case study for the area where major optimisations were expected. This report thus concludes the activities on dataformats in Hilado.

2.2 Overview

This document first reports the analysis (which turned out to be rather brief) and then presents the case of the LOFAR Storage Manager. The new process of writing a MeasurementSet is described and the way to access the data is presented. Finally some conclusions and further steps are given.

3 Data formats in the imaging stream

When the workplan for JRA Hilado was initially defined, it was assumed that major performance gains were to be obtained by a careful analysis of the various data format transitions in the full imaging chain. In "classic" off-line processing chains, the balance between time ordered *versus* baseline ordered storage is well known. However, for the very large datasets that Hilado is concerned with, two clear conceptual trends can be distinguished:

- There is a gradual move of functionality into the on-line domain, where native dataformats are used that are highly optimised for specific platforms, using the specific caching mechanisms of those platforms.

- Off-line processing increasingly uses CASA, which uses Measurement Sets stored in the Casacore Table format.

Research on the optimisation of data-handling "in the stream" falls outside the scope of Hilado, as it is quite instrument specific and has to be handled within the respective projects (e.g. LOFAR and ASKAP). For the SKA, the mapping of algorithms and machines is part of the Science Data Processor (SDP) workpackage.

Off-line processing of very large datasets is increasingly relying on Casacore Tables. These use a column based tiled storage mechanism, which offers flexible ways for access optimisation (see http://www.astron.nl/casacore/trunk/casacore/doc/html/group__Tables__module.html#_details).

To illustrate the way such optimisations can be implemented, the remainder of this report describes a case study related to the LOFAR project.

4 Case description and approach

For the 50-station LOFAR instrument the data rate for a single subband is about 10 MB/s. Because multiple subbands are kept on the same storage node, it is essential that the data can be dumped to disk as fast as possible, thus with hardly any overhead. On the other hand the data should be written in the standard casacore MeasurementSet format to be able to inspect and process them with the standard toolset.

Writing a MeasurementSet in the native way incurs overhead. One reason is that under the hood buffered IO is used; another reason is that several meta data columns are written that are constant for the LOFAR case.

MeasurementSets are handled by the casacore Table System that provides a way to use a dedicated storage manager. When accessing a table, the storage manager can be loaded dynamically provided that the shared library containing the code can be found in the library path.

This feature is used by the LOFAR data writer. It creates the MeasurementSet such that the Table System knows it has to use the special LofarStMan storage manager to understand the file containing the data that are written directly to disk.

5 MeasurementSet creation

The data writer creates the MeasurementSet in the standard way, but binds the columns of the main table to the LofarStMan storage manager. It fills all subtables and creates a small file *table.f0meta* in the main table directory containing information like antenna numbers, time, endianness, etc. It also contains a version number making it possible to evolve the format. The meta file is written in the casacore AipsIO format.

After this initialisation phase the MeasurementSet is closed and the main data file *table.f0data* is created. Data are written to that file when received from the online system. If possible O_DIRECT is used to avoid the kernel file cache overhead.

In version 1 and 2 each time slot in the file contains three blocks of data:

1. A 32-bit signed integer sequence number. It is used to derive the time.
2. Per channel the number of samples (16-bit unsigned integer) used by the correlator. It is used to derive the weights and flags.
3. The data array (ncorr,nchan) as single precision complex numbers.

The meta file tells how each block is aligned. Usually this is on 512 bytes because that is required to make use of the O_DIRECT option.

There can be gaps in the sequence numbers, thus time slots might be missing.

In case of a crash, the data are always fine. I.e., no indices or so have to be updated, as would be the case if the native table format or a system like HDF5 is used.

Note that LofarStMan is only used for the raw data. The first NDPPP processing step will create a MeasurementSet using the standard storage managers.

6 Accessing an existing MeasurementSet

When an existing LOFAR MeasurementSet is opened, the Table System detects that the LofarStMan storage manager is needed and will load it dynamically from the shared library with that name. First a register function is called to make the storage manager known. Thereafter all accesses to the data are done through LofarStMan.

LofarStMan calculates the number or rows in the data file and reports it back to the Table System.

On 64-bit systems the data file is memory-mapped. In this way the system takes care of caching if random access is done. The memory space of 32-bit systems is too small for mmap, so buffered IO is used instead.

The columns handled by LofarStMan and their values are given in the following table. All columns except DATA are readonly. Note that bytes are swapped if the endianness requires so.

TIME	start time + (seqnr + 0.5)*interval
ANTENNA1	from meta file
ANTENNA2	from meta file
FEED1	0
FEED2	0
DATA_DESC_ID	0
PROCESSOR_ID	0
FIELD_ID	0
INTERVAL	interval from meta file
EXPOSURE	interval from meta file
TIME_CENTROID	same as TIME
SCAN_NUMBER	0
ARRAY_ID	0
OBSERVATION_ID	0
STATE_ID	0
UVW	calculated on the fly using FIELD and ANTENNA subtables (for phase reference direction and antenna positions)

DATA	from data file
SIGMA	1
WEIGHT	1
WEIGHT_SPECTRUM	nr of samples / nominal nr of samples
FLAG	nr of samples == 0
FLAG_CATEGORY	empty
FLAG_ROW	false

A MeasurementSet stored with LofarStMan has some special characteristics:

- All columns are readonly with the exception of the DATA column.
- It is not possible to add or remove rows.
- It is possible to remove a column.
- A column can be added to the MS, but only with a regular storage manager.

For example, it is possible to make the FLAG column writable by creating a FLAGX column, copy FLAG to it, remove FLAG, and finally rename FLAGX to FLAG.

7 Lessons learned and future plans

Besides the gain in write speed, LofarStMan proved successful for another reason:

Accidently the data writer wrote the conjugates of the data during the first period. This could be solved elegantly by adapting LofarStMan such that version 1 conjugated the data and by setting the version to 2 for the fixed version of the data writer.

MeasurementSets are often selected or sorted on TIME. This proved to be slow because the entire data file had to be traversed. Therefore a new version 3 will get an optional extra file *table.f0seqnr* containing a copy of the sequence numbers. Normally this file will be present and correspond to the data file size. If so, LofarStMan will take advantage of it. Because the file is written afterwards, it may not be present or correct in case of failures. In that case LofarStMan will get the sequence numbers from the data file.

The only real problem encountered was that an MS stored with LofarStMan could not be used in the CASA package. The reason is that CASA uses different casacore libraries than LofarStMan, so even after it has registered itself, LofarStMan was still not known to CASA. This will be solved if CASA can be built using the same libraries.