LOFAR synthesis data handling
Introduction

Ger van Diepen
ASTRON

Goals

Get some knowledge about handling LOFAR data

- Visibility data (MeasurementSets)
- Image data and coordinates
- Selection and manipulation of data
- Dealing with measures
- Some useful tools

Only touching the surface, no in-depth course

Contents

- Fundamentals of Tables, MeasurementSets, and Images
- casacore
- pyrap
- TaQL Table Query Language
  - Data selection and update
  - Pretty printing of date/time, positions
- Some tools on Groningen clusters

See www.astron.nl/~gvd/tutorialN.pdf N=1..5

- NOT discussed
  - C++ basics and STL
  - Python basics and libraries
  - DAL and pydal (access to beamformed data)
  - CASA
  - HDF5

links

casacore code
casacore.googlecode.com
pyrap code
pyrap.googlecode.com
casacore documentation
http://www.astron.nl/casacore/trunk/casacore/doc/html/
casacore notes
http://www.astron.nl/casacore/trunk/casacore/doc/notes/notes.html
pyrap documentation
http://www.astron.nl/casacore/trunk/pyrap/docs/index.html
TaQL
http://www.astron.nl/casacore/trunk/casacore/doc/notes/199.html
LEL (image expressions)
http://www.astron.nl/casacore/trunk/casacore/doc/notes/223.html
MeasurementSet definition
http://www.astron.nl/casacore/trunk/casacore/doc/notes/229.htm
LOFAR MeasurementSet definition
http://www.lofar.org/operations/doku.php?id=public:documents:lofar_documents#dataformats
msselect
http://www.lofar.org/wiki/lib/exe/fetch.php?media=software:msselection.pdf
casacore Measures
http://www.astron.nl/casacore/trunk/casacore/doc/html/group__Measures__module.html#_details
makems
http://www.lofar.org/wiki/lib/exe/fetch.php?media=software:makems.pdf
makebeamtables
http://www.lofar.org/wiki/doku.php?id=engineering:software:tools:makebeamtables
LOFAR tools
http://www.lofar.org/operations/doku.php?id=engineering:software:tools

Table System

- Similar to a relational data base
  - Flat collection of rows and columns (no hierarchy like HDF5)
  - Row number is the primary key
- Description (schema) defines the columns
- Headers (keyword-value pairs) for general info
  - Table keywords are mainly used to define subtables
  - Column keywords are mainly used to define units and reference frames
    - Used by TableMeasures to store Measures in tables (RA/DEC in rad, J2000)
- Strong distinction between logical and physical organization
- Value types of columns and keywords
  - Scalar
  - Array fixed or variable shape or dimensionality
  - Record nested keyword-value pairs similar to a Python dict
  - Table (only in headers)
- Data types of columns and keywords
  - bool, char, short, int, float, double, Complex, DComplex, String

Table types

- PlainTable
  - A table stored on disk (as a directory)
    - `table.dat` description and headers
    - `table.info` ASCII description of table
    - `table.lock` used for concurrent access
      one writer, multiple readers
    - `table.f<seqnr>*` column data
- RefTable
  - A referenced subset (view) of another table (tells the columns/rows to use)
    - Result of selection, sort, or iteration
    - Can be persistent
    - Modifications are written into original table
- ConcatTable
  - Virtual concatenation of tables with the same table description
    - Can be persistent
    - Modifications are written into original tables
    - Can concatenate one or more subtables
    - E.g., concatenate WSRT MSs that are split in time
      - Also see function msconcat in pyrap.tables
- MemoryTable
  - A table held in memory (not persistent)

Table data managers

- A data manager defines how the data of a column are kept
- One data manager can handle one or more columns
- Data are stored in little or big endian format (and converted as needed)

- Storage managers store the data
  - StandardStMan — buckets of N columns by M rows
  - IncrStMan — only store if next row is different (can give compression)

    Arrays can be stored directly or indirectly (in `table.f1i*`)
  - TiledStMan — store arrays in tiled (chunked) way for fast access in alll directions
  - MemoryStMan — stores in memory

  Each storage manager has its own seqnr in the file names e.g., `table.f2`

  seqnrs do not have to be consecutive
- Virtual column engines calculate data on the fly
  - DerivedMSCal — AzEl, HA, PA
  - VirtualTaQLColumn — Column as a TaQL expression of other columns
  - ForwardColumn — Forwards to an equally named column in another table (kind of reference)
- A non-standard data manager can be used
  - LofarStMan (mixes stored and virtual)

- At table creation time columns are bound to data managers
  - Data managers can only be changed by copying the table

- Standard data managers are preloaded
- Non-standard data managers are usually dynamically loaded from a shared library

  Must be in (DY)LD_LIBRARY_PATH (set by `use LofIm`)
  - `liblofarstman.so`
  - `libcasa_derivedmscal.so`

Subtable

- Usually defines values of indices in main table
- E.g. ANTENNA table of MeasurementSet
  - Defines the properties (name, location) of each antenna of a telescope (station for LOFAR)
- Is normally part of the main table (i.e., a subdirectory)
- A header keyword (e.g. ANTENNA) defines its location
  - Relative location, thus moving the main table can be done
- RefTable and ConcatTable reference original subtable(s)
  - Open a subtable using the subtable's keyword, not by its path name
  - Or: use `table::subtable` syntax

MeasurementSet

A MeasurementSet is a Table with a set of subtables
See casacore note 229 http://www.astron.nl/casacore/trunk/casacore/doc/notes/229.html
and LOFAR MS ICD http://www.lofar.org/operations/doku.php?id=public:documents:lofar_documents#dataformats

- Main table contains visibility data and indices (row numbers) in subtables

DATA	Complex[npol,nchan]
FLAG	Bool[npol, nchan] True = corresponding data point is flagged
ROW_FLAG	True = all data in this row is flagged
TIME	time stamp (MJD seconds)
UVW	UVW in meters
ANTENNA1, ANTENNA2	Baseline
DATA_DESC_ID	Band (spectral window)
FIELD_ID	Field observed
WEIGHT_SPECTRUM	float[npol, nchan] weight per data point

- The main subtables are:
  - ANTENNA
  - DATA_DESCRIPTION and SPECTRAL_WINDOW
  - FEED
  - FIELD
  - HISTORY
  - OBSERVATION
  - POLARIZATION

  Other subtables are:
  - DOPPLER, FLAG_CMD, FREQ_OFFSET, POINTING, PROCESSOR, SOURCE, STATE, SYSCAL, WEATHER

  LOFAR specific subtables:
  - LOFAR_ANTENNA_FIELD
  - LOFAR_ELEMENT_FAILURE
  - LOFAR_STATION

showtable

```
MacDiepen2.CASA> showtable in=~/data/GER.MS dm=f sort=t
showtable: Version 2013Oct16GvD

Structure of table /Users/diepen/data/GERsel.MS
------------------ Measurement Set
10 rows, 30 columns (using 8 data managers)
out of /Users/diepen/data/GER.MS (12528 rows, 30 columns)

  ANTENNA1            Int      scalar
  ANTENNA2            Int      scalar
  ARRAY_ID            Int      scalar
  CORRECTED_DATA      Complex  ndim=2
  DATA                Complex  shape=[4,1] unit=Jy directly stored
  DATA_DESC_ID        Int      scalar
  EXPOSURE            double   scalar unit=[s]
  FEED1               Int      scalar
  FEED2               Int      scalar
  FIELD_ID            Int      scalar
  FLAG                Bool     shape=[4,1] directly stored
  FLAG_CATEGORY       Bool     array
  FLAG_ROW            Bool     scalar
  IMAGING_WEIGHT      float    ndim=1
  INTERVAL            double   scalar unit=[s]
  MODEL_DATA          Complex  ndim=2
  OBSERVATION_ID      Int      scalar
  PHASE_ID            Int      scalar
  PROCESSOR_ID        Int      scalar
  PULSAR_BIN          Int      scalar
  SCAN_NUMBER         Int      scalar
  SIGMA               float    shape=[4] directly stored
  STATE_ID            Int      scalar
  TIME                double   scalar unit=[s] measure=epoch,UTC
  TIME_CENTROID       double   scalar unit=[s] measure=epoch,UTC
  UVW                 double   shape=[3] unit=[m,m,m] measure=uvw,ITRF directly stored
  WEIGHT              float    ndim=1
  WEIGHT_SPECTRUM     float    ndim=1
```

```
SubTables:
    /Users/diepen/data/GER.MS/ANTENNA
    /Users/diepen/data/GER.MS/FEED
    /Users/diepen/data/GER.MS/FIELD
    /Users/diepen/data/GER.MS/SPECTRAL_WINDOW
    /Users/diepen/data/GER.MS/OBSERVATION
    /Users/diepen/data/GER.MS/SOURCE
    /Users/diepen/data/GER.MS/SYSCAL
    /Users/diepen/data/GER.MS/WEATHER
    /Users/diepen/data/GER.MS/NFRA_TMS_PARAMETERS
    /Users/diepen/data/GER.MS/DATA_DESCRIPTION
    /Users/diepen/data/GER.MS/FLAG_CMD
    /Users/diepen/data/GER.MS/HISTORY
    /Users/diepen/data/GER.MS/POINTING
    /Users/diepen/data/GER.MS/POLARIZATION
    /Users/diepen/data/GER.MS/PROCESSOR
    /Users/diepen/data/GER.MS/STATE
    /Users/diepen/data/GER.MS/SORTED_TABLE
```

PagedImage, HDF5Image

Table containing an image cube and auxiliary information (see below)

- A single column (`map`) contains the data cube
  - Usual axes are ra, dec, stokes, freq
- Zero or more masks; a default mask can be defined
- Normal data type is float; other data types are also supported
- Table keywords
  - coords image coordinates (with wcs info)
  - logtable name of logging subtable to log operations on image
  - imageinfo
  - miscinfo
  - units usually Jy/beam
- LOFAR image has extra subtables (or HDF5 groups)
  - contains all info from observation(s)
  - Mainly used for archiving purposes

    LOFAR_QUALITY, LOFAR_ORIGIN, LOFAR_OBSERVATION, LOFAR_SOURCE
    LOFAR_POINTING, LOFAR_FIELD, LOFAR_ANTENNA, LOFAR_STATION, LOFAR_HISTORY

Data access

Three layers

will be discussed in next tutorials

- casacore C++
- pyrap python
- TaQL SQL-like

- LEL makes it possible to form image expressions

  E.g. in python:

```
im=pim.image('image1.img - image2.img')
```

  See note 223

Casacore (C++ level)

- casa
  - Utilities (sort, median)
  - Containers (Record, ValueHolder)
  - Array, Vector, Matrix, Cube
  - Quanta (value with unit)
- tables
  - Creation, read/write, iteration, selection, sort
- measures
  - Quanta with reference frame
- ms/MeasurementSets
  - Tables holding visibility data
- images and coordinates
- scimath
  - Functionals and Fitting (linear and non-linear)
  - Mathematics (FFT, DFT, interpolation)

Pyrap (python level)

Python interface on top of some casacore functionality

- `pyrap.tables`
- `pyrap.quanta`
- `pyrap.measures`
- `pyrap.images` and `pyrap.images.coordinates`
- `pyrap.functionals` and `pyrap.fitting`

TaQL (C++, python, shell level)

SQL-like data selection and manipulation

- `SELECT`
- `UPDATE`
- `INSERT`
- `DELETE`
- `CREATE TABLE`
- `CALC`
- `COUNT`

Setup on CEP1,CEP2

```
use LofIm          # for pyrap, TaQL, LOFAR
use Casa           # for casapy
use LUS            # for pybdsm, etc.
use MonetDB        # for gsmutils.py
```

set up the environment

However, best not to mix the first three

- they use different libcasa_ libraries