Tianlai Data Format

2016.11.17

Shifan Zuo

HDF5—Hierarchical Data Format

https://www.hdfgroup.org/HDF5/

HDF5 is a data model, library, and file format for storing and managing data. It supports an unlimited variety of datatypes, and is designed for flexible and efficient I/O and for high volume and complex data.

- Self-describing and portable to a diversity of computational environments;
- No inherent file size limitations;
- C, C++, Java, Fortran, Python interfaces;
- Can be run on single node or massively parallel/distributed systems;
- Built-in (and extensible) compression;
- Parallel reading and writing (via MPI-I/O);
- “Chunked” (tiled) data for fast data access.
- …

Tianlai Data Format — HDF5

File: 20160830124922_20160830125921.hdf5

Top level attributes

```
/.attrs["comment"]:  28hr obs
/.attrs["telescope"]:  Tianlai-Cylinder-I
/.attrs["observer"]: Fengquan Wu
/.attrs["siteelev"]:  1493.7
/.attrs["samplingbits"]: 8
/.attrs["cywid"]:  15.0
/.attrs["nfeeds"]:  96
/.attrs["timezone"]:  UTC+08h
/.attrs["corrver"]: 0.0
/.attrs["nants"]:  3
/.attrs["nfreq"]: 1008
/.attrs["lofreq"]: 935.0
/.attrs["sitename"]:  Hongliuxia Observatory
/.attrs["inttime"]:  3.99507456
/.attrs["epoch"]:  2000.0
/.attrs["keywordver"]: 0.0
/.attrs["corrmode"]: 1
/.attrs["dishdiam"]:  6.0
/.attrs["sec1970"]: 1472532562.42
/.attrs["freqstart"]: 685.0
/.attrs["obstime"]:  2016/08/30 12:49:22.415561
/.attrs["nickname"]:  28hr obs
/.attrs["freqstep"]:  0.1220703125
/.attrs["npols"]: 2
/.attrs["recvver"]: 0.0
/.attrs["sitelon"]:  91.80686667
/.attrs["cylen"]:  40.0
/.attrs["history"]: Recorded from the correlator. Transformed from raw format
/.attrs["sitelat"]:  44.15268333
```

Tianlai Data Format — HDF5

File: 20160830124922_20160830125921.hdf5

Top level dataset

```
blorder   shape =  (18528, 2)
channo   shape =  (96, 2)
   channo.attrs["dimname"]: Feed No., (Channel No. of Xpol, Channel No. of Ypol)
feedno   shape =  (96,)
feedpos   shape =  (96, 3)
   feedpos.attrs["dimname"]: Feed No., (X, Y, Z) coordinate
   feedpos.attrs["unit"]: meter
noisesource   shape =  (1, 3)
   noisesource.attrs["dimname"]: NoiseSource No., (Start, Stop, Cycle)
   noisesource.attrs["unit"]: second
nspos   shape =  (3, 3)
   nspos.attrs["dimname"]: NoiseSource No., (X, Y, Z) coordinate
   nspos.attrs["unit"]: meter
vis   shape =  (150, 1008, 18528)
   vis.attrs["dimname"]: Time, Frequency, Baseline
```