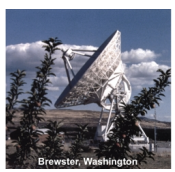
Native Mark6 Playback in DiFX and Related Developments

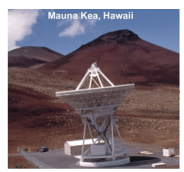
W. Brisken

National Radio Astronomy Observatory (Socorro, NM)

Max Planck Institüt für Radioastronomy (Bonn, Germany)

2015 Nov 17





Mark6 General Characteristics

Physical

* Same chassis, different backplane, COTS disk controller
* Slave chasses possible for > 2 simultaneous modules
* Simultaneously record from up to 4 10 Gb Ethernet interfaces
* Similar modules: back-panel power, front-panel I/O

Recording Software

* C-plane in Python: op. int., mounting, grouping of m modules
* D-plane in C: high performance I/O code; uses pfring

Operational model

* $\sim$ 10 MB chunks from each interface captured into write buffer
* Non-deterministic assignment of each chunk to one of the $8m$ drives

See `http://www.haystack.edu/tech/vlbi/mark6/mark6_memo/05-Mark6_Design_and_Status.pdf` for more details

On-disk Format

Each drive has two partitions:

Meta-data partition

* Stores drive/module identity, module groupings, and directory
* Directory is stored as a Python list/dictionary in text format
* Info on all drives in a group is identical

Data partition

* One file (per drive) is created per scan
* File starts with small binary header containing: sync word, header version, block size, packet format, and packet size
* Each chunk is preceded by a chunk header containing: block number and size of chunk+chunk header
* Each chunk is an integer number of frames.
* Chunk data is always non-legacy VDIF
* Mark5B data is converted to VDIF on-the-fly

VDIF EDV 4

* VDIF supports user-defined *Extended Data* to be stored within each VDIF header
* Multi-thread VDIF data can be *multiplexed* into single-thread VDIF where each thread contains all of the channels
* Multiplexing data that contain missing packets, especially those where a particular thread is flaky, poses problems in assigning validity to the output packet
* Mark6 case with multiple Ethernet streams is an important driver for this
* EDV4 allocates up to 64 bits to assign per-channel data validity
* EDV4 being assessed by the VDIF committee amid some minor controversy (discuss?)
* *Provisional support is in vdifio now*

Multiple Datastreams per Antenna

* Many Mark6 (and some Mark5C) deployments use multiple independent recorders for wider recording bandwidths
* Core of mpifxcorr has always distinguished between a *datastream* and an *antenna*
* Not all of the external software (e.g., vex2difx, difx2fits, ...) did
* Considerable effort over summer 2015 put into restructuring code (mostly vex2difx) to support this
* Mostly working now: one outstanding issue in difx2fits w.r.t. pulse cal

Multiple Datastreams per Antenna: Simple Example

The following `.v2d` file lines will create two datastreams for antenna SC:

```
DATASTREAM d1 { file=tomato format = VDIF/0:1:2:3/5032/2 }
DATASTREAM d2 { file=banana format = VDIF/7:6:5:4/5032/2 }
ANTENNA SC { datastreams=d1,d2 }
```

Note that many parameters from the ANTENNA block can be put in the DATASTREAM block instead. See `http://www.atnf.csiro.au/vlbi/dokuwiki/doku.php/difx/vex2difx`. Note that documentation for vex2difx is being developed in the `/doc` directory of the vex2difx distribution.

DiFX Native Mark6 Implementation

Code changes were needed in many parts of the DiFX software suite:

* difxio: basic data structures
* vex2difx: accepting Mark6 as a data source
* vdifio:
  - Mark6 per-packet gather infrastructure
  - EDV4 support in multiplexing
* mark5access: new functions (`blank_vdif_EDV4()` & `blank_vdif_EDV4_complex()`) to blank data *after* decoding
* mpifxcorr:
  - `VDIFMark6DataStream` class
  - Flip some switches and call the EDV4 functions as needed

Alternate Approachs

Clean separation of Streams and Formats

* Introduce Mark6Reader (or Mark6Source) concept
* Each is fanned out to form one VDIFDatastream per thread
* One or more VDIFDatastreams available per antenna
* Would solve issue associated with multiple antennas' data in one stream

mark6sg library?

* Wanted to use it but ...
* Architecture depended on per-frame gather, not per-block gather
* Above Mark6Stream approach would be compatible with this

Practical Details: How to Correlate Directly Off Mark6

Assumptions

* All Mark6 drives needed for scan are properly mounted
* Env. Var. `MARK6_ROOT` set if data partition mount points are other than `/mnt/disks/?/?/data/`

`.v2d` file

* Now: `ANTENNA PT { mark6files=scan1.vdif ...}`
* Later: `ANTENNA PT { module=NRAO%012,NRAO%013 }`

Status

- Testing has begun on a couple data sets; success see with:
  - Complex samples works
  - Four input streams (one thread each)
  - Multi-channel multi-thread input
- Helge to comment on logistical issues

Remaining Work

* Treat Mark6 more like Mark5
  - Specify Module(s) rather than file names: make use of directory structure
  - No plans to retire file mode
* DBBC3 *fan-out mode* support
  - Certain DBBC3 modes split alternate samples from one thread across different Ethernet streams
  - Fix should be easy! Logically equivalent to multiplexing but with changed number of output channels
* Mark6 issue:
  - File writes become clumpy causing multiple 10 MB chunks from one Ethernet stream to be written in a row, complicating reordering code in DiFX.
  - Symptom: lower than expected playback weights

Special Thanks To ...

* Helge Rottmann
* Jan Wagner
* Roger Cappallo
* Walter Alef