A New VDIF (and Mark5B) Decoder

7th DiFX Users/Developers Meeting, Tamna Observatory, South Korea

Walter Brisken
NRAO - Socorro

DiFX at NRAO

- VLBA Sensitivity Upgrade Project has been completed
  - This required some changes to DiFX
- Many new Mark5C specific updates
  - Support for new Directory format
  - Diagnostic programs for VDIF data
    - vdifspec, vdifd, vdifbstate, vdiffold
  - Special option in mk5cp for Mark5B data
    - Perform “fix” operation on copy (more on this later)
- Support for VDIF / Phased VLA
  - Starting with considerable effort by Adam
- Some “first fringes”:
  - LMT (w/ Haystack)
  - International fringes to SHAO 65m

VLBA DiFX Correlator

- Upgraded from 20 to 24 nodes (200 to 232 cores)
- Occasionally encounter insufficient memory conditions
  - For 200+ phase centers
- 14 Mark5 playback devices
  - Mark5A systems at VLBA antennas are being retired
  - Will increase correlator to >=20 units this year
- Thanks to Steven Tingay we will be getting DiFX hooked up to a Lustre filesystem
  - Should speed up some operations
  - Will provide more storage

Introduction and Motivation

- Mark5B support in DiFX is fairly mature already
- VDIF support has been functional for a while
- Both cope with minor interruptions to perfect data flow
- Some important cases have been difficult to deal with:
  - VDIF from VLA: sometimes threads start at vastly different times
  - VDIF from VLA: sometimes there are large gaps in a recording
  - Mark5B and VDIF at VLBA: new X-cube soft-switch drops ~10-5 packets
  - VLBA: Either Mark5C or RDBE occasionally inserts extra data into stream
  - *Basically, hardware is not perfect, so fix in software*
- VDIF performance in Mark5 playback suffers due to over-taxing the Mark5 CPU

Want robust, high performance VDIF and Mark5B processing

Design Principles

- Keep complicated algorithms in external libraries
  - Simplify testing and reuse
- Don’t compromise on speed
- Full reporting of statistics
- Simple interface for dealing with abnormal data
  - Avoid having to deal with many different exit conditions

The above principles can apply to most developments

- Separate record scan and schedule scan concepts
  - It is possible one record scan contains more than one schedule scan
  - It is possible one scheduled scan is split into multiple record scans

General approach

- First: jump to starting point in media
- Next: read a chunk of data (from file, Mark5 or network)
- Then: “condition” the data with code to reorganize/filter
- Finally: put data into datastream buffer section and assign to that section the timestamp of the first valid packet

VDIF pre-processing:

- vdifio : vdifmux() does the following
  - Reorders incoming packets
  - Pads missing data with masked data
  - Excises unwanted “interloper” data
  - Corner-turns multi-thread VDIF into single-thread VDIF
  - Output is uniform time grid of data; incomplete records are masked
- Statistics generated
  - # valid and invalid input frames
  - # duplicate frames (should never happen, but has been seen to)
  - # wrong thread frames (indicates correlator configuration problem)
  - # skipped bytes
  - # bytes processed in total
  - # of total good single-thread VDIF frames produced

Mark5B pre-processing

- mark5access : mark5bfix
  - Excises unwanted “interloper” data
  - Ensure first output packet has timestamp that has integer ns start time
  - Pads data with fill packet to replace missing packets
  - Mark5C records 1 Mark5B frame as 2 packets:
    - Look for both halves of a frame (test is not perfect though)
  - Outputs uniform time series of Mark5B packets
- Statistics generated
  - # valid and invalid input frames
  - # of interloper bytes
  - # of missing packets
  - # of bytes processed

Performance enhancing tricks (VDIF)

- Minimal data copying is done
  - First pass through data identifies data movements to be performed
  - Second pass through data corner-turns from initial location to final location
- Optimized corner turning algorithms
  - Separate algorithm for each #channels / #bits case
    - In some cases separate algorithms for 32- and 64-bit CPUs
  - Makes use of bit mask-and-shift operations
    - Usually less than 1 integer operation per sample
  - I will investigate Harro's SSE2 corner turners
- Use OpenMP for instruction-level parallelization

Native Mark5 implementation

- Data is gotten from a Mark5 module with XLRRead(…)
  - This function blocks
    - Any extra processing *will* reduce datastream throughput
  - It performs a DMA transfer and takes effectively zero CPU
  - It has known problems for reads > 25 MB
- A new posix thread is now spawned for the reading of data
  - Reads > 20 MB are split into a series of smaller reads
- A posix barrier is used to synchronize reading with pre-processing
  - A series of posix mutex locks are used to control a circular buffer for incoming data

File implementation

- Not that different from existing Mk5 datastream
- Uses new vdifsum() and mark5bsum() functions to generate summaries of input files to allow for a quick jump to the expected start of job data.
- Could possibly benfit from a similar threading model used by Mark5 datastream, but thought to be less important as OS-level file read-ahead effectively does this for free

Network (eVLBI) implementation

- Not yet started
  - Much thought has gone into it though
- A double buffer system similar to that used by the Mark5 implementation will be used
- End result code may be simpler than existing eVLBI code in DiFX as the packet reordering / gap filling code is already compartmentalized
- Want to support UDP, TCP and raw socket operations
- Support for simultaneous reads from multiple network cards is a natural extension
  - This may be needed by LOBO (see slides at end)…

Development status

- VDIF work began in early 2013
- Work accellerated in July
  - X-cube packet loss in Mark5B data prompted immediate effort on the Mark5B front
- Currently the code is working, is fast, and is in use by the VLBA for all wideband data
- Code is being developed in the vdifnew branch of mpifxcorr
- Currently there is a separate class derived from DataStream for each of:
  - VDIF on Mark5 (vdifmark5.cpp/h)
  - VDIF on files (vdiffile.cpp/h)
  - Mark5B on Mark5 (mark5bmark5.cpp/h)
  - Mark5B on files (mark5bfile.cpp/h)
  - Ultimately some consolidation of code should occur

Development status (2)

- Some functionality is still missing:
  - “Fake” mode for benchmarking
  - Network datastream
    - See section on VLITE below
  - Jump
    - The new datastream classes ignore the requested data time in the manager’s data request
      - Cases where a correlator job skips scans require reading all intermediate data
    - A jump at the beginning of the job is performed properly

Workshop goal: pave way for merging this into trunk and getting a new DiFX release planned before 2013 comes to an end!

VLITE

- VLA Low-band Ionosphere and Transient Experiment
- A commensal P-band (~330 MHz) system
  - Observes all the time at 330 MHz, even when used for other science
- Funded by Naval Research Laboratory
- A 10-antenna ramp up to LOBO: the LOw-Band Observatory

- Consists of
  - Samplers/channelizer/formatter for 10 VLA antennas
  - Real-time DiFX correlator
  - Real-time integration with VLA M&C system
- Being developed by NRAO Oct 2013 to Sept 2014
  - I will do the DiFX development

New calibration data path

- difx2fits attaches Tsys, Pcal, weather and flag data to FITS files
- Previously such data could only be provided as one-file-per-table (tsys, pcal, weather, flag) except when DiFX extracted Pcal or Tsys
- Now difx2fits looks for <obscode>.<station>.<type>
  - <obscode> is experiment name: BD152, ...
  - <station> is antenna code: HH, ...
  - <type> is one of: tsys, pcal, weather, flag
- Priority now is:
  - 1. DiFX extracted (pcal / tsys only)
  - 2. <obscode>.<station>.<type>
  - 3. <type>

WFB in 2014 and beyond…

- I am moving!
- Off to Minneapolis Minnesota
- Will retain a 50% position with NRAO
  - Less focus on operations/VLBA development
  - Heavy focus on VLITE project
    - No end of my DiFX involvement in sight!
  - Beyond Sep 2014 my NRAO role becomes less clear…
- Will acquire a 50% position at UMN
  - Working with the Cerenkov Telescope Array project
- Beyond Sept. 2014, some uncertainty of NRAO position