JOINT INSTITUTE FOR VLBI IN EUROPE

BlackHoleCam WP1.1 task d
eVLBI interface for testing of mmVLBI observations

B. Eldering, H. Verkouter

Revision 0.4

March 4, 2015

1 Introduction

Finding Very Long Baseline Interferometry (VLBI) fringes at the (sub)mm wavelengths the Black Hole Camera/Event Horizon Telescope (BHC) will observe is challenging at best, even on the brightest sources.

The short atmospheric coherence time and strong dependence on current weather conditions necessitates frequent recalibration of the receivers. Small pointing errors cause immediate decorrelation and is solved by very frequent (compared to cm wavelength VLBI) pointing calibration observations.

In the BHC consortium, where many oversubcribed mm-observatories participate, it is important to provide quick feedback on the current performance of the array, in order to waste as little of the scarce observing time granted.

In this document a proposed plan for implementing an ”as quick as possible” fringe verification system for the BHC is described. Please note that it is recognized that not all BHC stations may have sufficient network connectivity to usefully support this. This does not, however, affect the design of the solution for stations that do have it.

Section 5 below summarizes the work effort and time estimates.

2 Background

The Joint Institute for VLBI in Europe (JIVE) has built an automated fringe checking system for the European VLBI Network (EVN). This system automatically extracts samples of raw VLBI data of a running (calibrator) scan from selected stations and uploads them via the network to a central repository. After receipt of the snippets of data from all selected stations a correlator job is scheduled and produces so-called fringe plots online.

The plan is to, grosso modo, replicate this system for the BHC. The main difference between EVN and BHC operations lies in the use of different recording systems; to wit Mark5[1] (EVN) versus Mark6[2] (BHC) recorders.

In the next section a more in-depth explanation of the existing system is given, such as to allow for a discussion about possible schemes of how the system can be altered to work for the BHC.

A major design goal is to implement a system which will run independent of the Mark6 data recording software. More specifically, the system will only run when there is no data being recorded. As such the system will rely heavily on the observing strategy: there has to be 'dead' time between observed scans. This is at least guaranteed by the frequent re-pointing and/or re-calibration observations and possibly because of scheduled gaps between scans.

By not tapping into the actual recording software, real-time performance may be lost but significant flexibility, independence of the recording software and avoiding complicating this software is gained. A decisive factor in this is that real-time performance will not be necessary, or even be possible in the first place.

[1] http://www.haystack.mit.edu/tech/vlbi/mark5/

[2] http://www.haystack.mit.edu/tech/vlbi/mark6/

3 The current system

In the EVN Mark5 recorders are used to record VLBI data to hard disks. The hardware of the Mark5 recorders can be exploited to allow for extraction of data out of a running disk recording. Using the Mark5 hardware the recorded data can be tee'ed into the computer's main memory, e.g. into a circular buffer. Once in ordinary memory the data can subsequently be transferred elsewhere.

In the current automated fringe test system, software at JIVE inspects the running experiment for calibrator scans by cross referencing the VLBI Experiment schedule file (VEX file) with a table of known calibration sources. For those matching scans, the software instructs, at the appropriate time – when the scan is being observed – the recording software at the stations to make a snapshot of the data stream. A short time sequence of data, typically two seconds worth of data (256 MB) is captured to a regular file on the Mark5's system disk. This file is then transferred via the network to a file server at JIVE.

Another piece of software monitors the incoming data area. It knows what data to expect because the initiating step leaves a record of at which stations it has triggered a snapshot to be created and uploaded. As soon as all expected data have arrived, a correlator job is prepared and started. A special post-processing script is run on the data after correlation has finished.

This script generates plots of amplitude and phase versus frequency, for all baselines and all subbands defined in the VEX file. Another output is a web page[3], showing in matrix form for all subbands by all baselines (auto- as well as cross correlations), the signal-to-noise ratio of a detected fringe, if any. Individual plots can be quickly visualized by mousing over a matrix cell or enlarged by clicking on the entry.

[3] See e.g. http://www.evlbi.org/tog/ftp_fringes/FR020/scan02/index.html for an example output web page of a fringe test experiment

4 Proposed changes for BHC use

Based on the explanation in the previous section it can readily be seen that the actual scheduling, correlation and post-processing are data capture independent; they are based solely on the VEX file. The proposal is therefore to fully re-use these components.

This leaves the following areas where changes are needed to meet BHC requirements:

- the decision of how much and which data to extract
- the act of initiating and execution of the scheduled data sample extraction
- the transfer to the central location

4.1 Deciding how much and which data to extract

At cm wavelengths, typically one or two seconds worth of one 16- or even an 8 MHz wide subband of data on a strong calibrator is enough for a significant fringe detection. Because of the EVN stations being well-connected and the data rates being limited to ≈1 Gbps (128 MB/s) JIVE does not even attempt to make a selection of the data but just uploads about two seconds worth of data.

4.1.1 How much data to extract?

At mm wavelenghts the situation is different. In order to get a reasonable detection the full bandwidth must be used - selection of only a sub set of the recorded bandwidth is detrimental to the signal-to-noise ratio.

Furthermore it has been indicated that at least ten seconds worth of data is required to accumulate enough signal-to-noise for a fringe detection.

Assuming a recorded bandwidth of 4 GHz the data volume of a data sample capture will be:

4 GHz × 2 polarizations × Nyquist sampling × 2 bits per sample = 32 Gbps data rate, or 4 GB per second.

Thus, a ten second long capture would require 40 GB of data to be transferred. Even for the best connected station this amount might be a challenge. It is expected that a bare minimum of ≈10 GB (8 Gbps for 10 seconds) is needed for fringe detection. Thus data volumes anywhere between 10 and 40 GB per telescope can be expected.

4.1.2 Selection of data

The raw VLBI data is produced at a station by a digital back end (DBE). This is either the U.S. developed Roach Digital Back End, RDBE, or the European developed Digital Base Band Converter, DBBC.

Depending on the exact firmware running in these DBEs, the digitized sky bandwidth may come out in any number of subbands. Current RDBE firmware versions will output 512 MHz wide channels, whereas the DBBC may output 64 MHz wide channels. Both DBEs output VLBI Data Interchange Format (VDIF[4]) formatted data streams.

Despite what was said under 4.1.1 it may be necessary to extract only a subset of data from a recording - e.g. the full data capture cannot be transferred on account of limited network connectivity to the remote file server.

Also, for obvious reasons, the data capturing software must at the very least be able to extract a time range from a recording. In order to keep data extraction as simple as possible it is proposed to support integer second time range extraction only; no sub-second time selection will be supported.

Taking into account the fact that it is VDIF formatted data which will be recorded, it is proposed to only support further data selection by VDIF thread. This specifically means that if more than one channel of data is carried in one VDIF thread, all channels in the selected VDIF thread(s) will be extracted. It is probably true that all relevant DBE firmwares will output every channel in their own VDIF thread, rendering this a non-issue.

Summarizing, the data extraction software must support:

- Integer second time range selection.
- VDIF thread selection.

4.2 Initiating and execution of the data sample extraction

In the EVN operational mode, real-time control of the recording software at the stations is necessary. The initiation and execution of the data extraction are one and the same action. As this is an unfavourable mode of operation for the BHC it is proposed to separate the two concerns into two distinct phases: the scheduling of the data sample extraction and the mechanics of doing it.

To this effect it is proposed that JIVE produces software which will run autonomously on the Mark6 recorder(s). This software will monitor a central location for ”sample extraction schedules”, downloads them and executes them at the scheduled time.

The main reasons for favouring this approach are:

- Data sample extraction schedules can be versioned. Should any last minute changes in e.g. selection criteria change, this can easily be accomodated provided the time-to-execution is longer than the schedule monitoring interval.
- Much better security because there is no need for real-time recorder control. Stations implementing strict network access rules can easily implement a scheme where the Mark6 recorder is only allowed to connect to one or two network locations outside the station: the schedule location and the file server where data is to be uploaded.

[4] http://www.vlbi.org/vdif/

- Data sample extraction can be scheduled in advance, or, if needed, instantaneously (with a delay on the order of the schedule monitoring interval of course).

- The data sample extraction itself will work, regardless of network connectivity, provided a version of the schedule was downloaded earlier. Although this could mean the upload will not work either, this can be retried at a later time.

- It is possible to build upon/extend software developed for the EVN to support 'triggered' observations. This software runs on the Mark5 recorders and polls a central location for trigger schedules. If such a trigger schedule exists, the operator at the telescope is presented with a choice of accepting or declining it. When accepted it is downloaded and the telescope switches to observing this trigger schedule in favour of the previous running observation. After the trigger schedule finishes, the previous running observation is continued. The software would seem to provide a good starting point for extending/changing it to suit BHC's needs. (E.g. the built-in operator interaction needs to be removed.)

ToDo: (for the scheduling)

- Analyze trigger software for accurate estimation of work effort required to adapt to BHC needs.
- Design a simple 'scripting language' which will describe a data sample capture and upload sequence
- Edit trigger software to update data sample extraction scripts from the central location.
- Implement software which parses the data sample capture and upload scripts and schedules the execution of the steps, allowing for redefinition or removal of the steps. No software to build on or start development from exists.

Then there is the mechanics of the extraction itself. As argued under 4.1.2 the program should only support two selection mechanisms. The program must intimately know about how the data is recorded. The recording software stripes the data across the disks in blocks of some size, holding an integral number of VDIF frames. When a block is written to disk, first a small header containing the block sequence number and the block size is written. Thus, a recording consists of a number of individual files, one per disk that was selected for the recording. Each file in turn contains a stream of blocks.

For the purpose of the semi-automatic fringe test data extraction, it is sufficient to read blocks from a recording containing data from the requested time range and/or selected VDIF thread(s).

Given the simple file format, easy to parse VDIF headers and minimal requirements on the extraction software it is proposed to develop a stand-alone utility which delivers this functionality. In this context 'stand-alone' means that there will be no dependency on the Mark6 recording software or tools delivered with that package at all.

ToDo: (for the extraction)

- Write data extraction program, allowing time range selection based on integer seconds and/or VDIF thread id(s).

4.3 The transfer to the central location

After the data sample capture has finished, a file of 10-40 GB size should exist containing only the VDIF frames of the time range and thread(s) of interest.

The final step is to upload this file to the central file server. In the EVN, JIVE has gained a lot of practical experience of moving sizeable amounts of data (certainly larger than these snapshots) across intercontinental high-speed networks. The TCP protocol is very ill-suited for these kinds of transfers and typically grinds to a halt, ending with a time-out on the connection and leaving the user with an incompletely transferred file.

JIVE has developed software - `jive5ab` - which includes a UDP based reliable protocol[5] which does perform adequately on fat, long round-trip time network links. The protocol originates from the Grid Computing effort and is called ”UDT”. It requires no configuration other than potentially one hole in a firewall - one UDP port must be open in both directions.

In order for a transfer, two `jive5ab` server programs must be running: one on the sender and one on the receiver. A Python based driver script - `m5copy` - was developed to ease copying of files from one location to another.

There is a choice of using a push or a pull model; either the central facility pulls the snapshot from the Mark6 to itself or the Mark6 pushes the snapshot to the central facility. This can be arranged to the preference of the individual stations. Either way, one TCP port must be opened, beit from the central facility into the Mark6 (pull model) or from the Mark6 to the central facility (push model).

Considering that the proposed scheduling software will run on the Mark6, a push model seems most suitable; the data transfer to the central location can be programmed at an appropriate time using the scheduling mechanism.

Another benefit of the push model could be that, as was hinted at before, the schedule execution software could retry the transfer in case it fails due to e.g. a broken or unavailable network connection. Experience with building production ready e-VLBI has shown this not to be a hypothetical situation: it happens in practice.

Currently the software does not support a ”resume” functionality. A partially transferred file must therefore be retransferred in whole. As the file sizes will be large and the network connections to the stations (comparatively to the transfer size) slow, this is unfavourable.

ToDo:

- Support for resume functionality built into the `jive5ab` and `m5copy` programs.

[5] UDP itself is unreliable by design

5 Description of work

The following table lists the deliverables and in which month number, counting from 1 January 2015, they will be delivered.

#	Description	Sect.	Month
1	Analyze trigger software for accurate estimation of work effort required to adapt to BHC needs.	4.2	4
2	Design a simple 'scripting language' which will describe a data sample capture and upload sequence.	4.2	7
3	Write data extraction program, allowing time range selection based on integer seconds and/or VDIF thread id(s).	4.2	9
4	Edit trigger software to update data sample extraction scripts from the central location.	4.2	12
5	Implement software which parses the data sample capture and upload script and schedules the execution of the steps, allowing for redefinition or removal of the steps. No software to build on or start development from exists.	4.2	15
6	Support for resume functionality built into the `jive5ab` and `m5copy` programs.	4.3	17
7	Deliver functional system		23