The 5th IVS General Meeting Proceedings, 2008, p. 395–399

Progress in Technology Development
and the Next Generation VLBI System

Mark 5C Software Development Program

Chester Ruszczyk

MIT Haystack Observatory, USA

Abstract. The product road map for Mark 5C software development program that will support 4 Gbps recording over a 10 GigE data connection is presented.

The Mark 5C software development effort is being staged as a three phase project. The first phase of the project will accept data via a standard network interface card and store data to a COTS storage device. The second phase of the project, dubbed 'Mark 5C-', will accept data from a standard network interface card and will have the capability to store the data in Mark 5B format to either a standard Mark 5 disk modules or to a standard filesystem (at a reduced data rate). This second phase will allow other software applications to communicate with the Mark 5C application, with the aim to reduce the time to deliver/debug the software, and to help facilitate the burst-mode development effort. The final phase of the project will release the fully functional Mark 5C application that will configure, control and manage the actual Mark 5C hardware.

The Mark 5C software development effort is being supported by MIT Haystack Observatory and NRAO.

1. Introduction

The Mark 5C software development programs objective is to create an application that can support a 4 Gbps recording capability on a Mark 5C hardware platform by the end of 2008. The paper in this proceeding [1] provides an overview of the ongoing Mark 5C VLBI data program at Haystack Observatory from a hardware and system point of view.

The software development effort of Mark 5C application, referred to in this article as the VLBI transfer application, will focus on record functionality required to minimize the software development time after a functional Mark 5C hardware system is available. Playback of the data is expected to be supported through the use of 3^{rd} party applications with all storage devices appearing as a standard filesystem or transfer of the recorded data in Mark 5B compatibility mode to an existing hardware correlator.

2. VLBI Transfer Application

The development of the VLBI transfer application will use a phased approach. The goal of this approach is twofold: 1) develop an application without the availability of fully functional hardware, and 2) allow third party developers the opportunity to begin integration of the application into their own software development time-line, at a reduced feature set. The result will be two preliminary software releases followed by the first official release.

Each release will support a target goal and a specific set of features. The goals will focus on a specific record and playback path, or the path the incoming data takes from the network interface to the storage device, and vice verse, and the features will focus on the command and control functions necessary and the method used to communicate this information to the application. The goals and features of each individual release, referred to by their version numbers, are introduced in Sec. 2.1–2.3.

It should be noted that once a feature is introduced to the application that it is not removed from the application but carried forward to the next release along with bug fixes.

Key characteristics of the VLBI transfer application are:

- developed in "C" to maximize the development team strengths;
- targeted to execute on the latest Linux distribution, e.g. Debian lenny, with the latest kernel, 2.6.24;
- provide an initialization option, via a configuration file loaded up at start-up;
- the periodic update of status information;
- support for Ethernet "Layer 2" data communications as specified in the Mark 5C V1.0 Specification [2];
- command and control of the application through a:
  - TCP port,
  - VSI-S [3] standard protocol syntax.

To aid in the software development cycle of the application, we will utilize the following standard software practices:

- software version control, e.g. CVS;
- linux software version numbering;
- major feature releases;
- standard Linux packaging;
- bug reporting system.

Fig. 1 shows the featured record and playback data paths that each version will support.

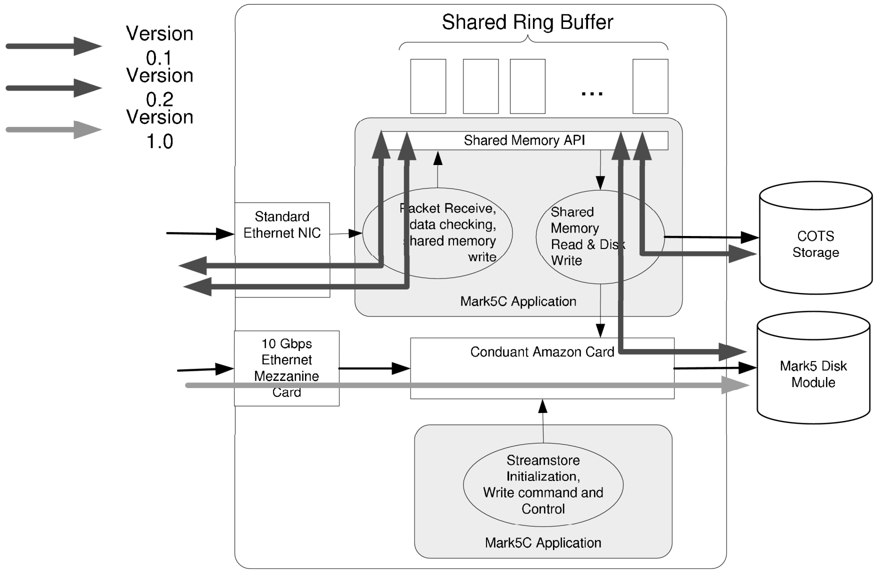

Figure 1. The VLBI Transfer Application Record and Playback Paths

2.1. Version 0.1

The goal of version 0.1 of the VLBI transfer application is to provide the initial framework for the application with support for a maximum record and playback capability of 512 Mbps. Version 0.1 will accept data from a standard 1 or 10 Gbps Ethernet Network Interface Card (NIC) at layer 2 and store it on a COTS storage device, e.g. system disk, RAID array. The recommended playback capability is to use a 3^{rd} party application to read data from the standard COTS storage device and output it over the standard NIC.

To control the VLBI transfer application a separate control application will provide a command line interface for basic configuration, record and status commands. This application will support a subset of the Mark 5C commands [4] and focus on commands associated with non-Streamstor command and control. This application will use a standard TCP socket to communicate with the VLBI transfer application using the VSI-S syntax.

2.2. Version 0.2

The goal of version 0.2 of the VLBI transfer application is to add support to record data to Conduant disk module via the Conduant's Streamstor Amazon controller card using their Software Development Kit (SDK) version 8.X at a maximum rate of 512 Mbps. Version 0.2 will receive data from the standard 1 or 10 Gbps Ethernet NIC at layer 2 and store it to Conduant's storage device in a Mark 5B compatibility format [5].

To read, or playback, the data from the Conduant disk modules it is recommended that a Linux kernel module known as File in UserSpacE (FUSE) [6] and a separate application, e.g. Fuse Mk 5A [7], be used to allow the Conduant disk module to appear as a standard filesystem. As in version 0.1, then any 3^{rd} party application can access and transfer the data as if it was a standard COTS storage device, via the standard NIC interface.

Support for additional commands of the Mark 5C Software Interface Specification [4] will be available via the control application that targets traditional transfer of data from the motherboard to the disk modules. In addition an initial version of a Graphical User Interface (GUI) application will be made available supporting a limited command set.

2.3. Version 1.0

The goal of version 1.0, the official release, of the VLBI transfer application is to add support to record data at a rate of 4 Gbps. This capability requires a fully functional Mark 5C hardware system that includes Conduant's 10 Gbps Ethernet daughter card attached to the Amazon card and the corresponding version of their SDK. Therefore, the applications primary function is the initialization, monitoring, command and control of the hardware supporting all of the features specified in [2].

To playback data, the functionality is identical to that of Version 0.2 and will require that the appropriate application, e.g. fuseMk5A, can support the latest version of the SDK and it associated features.

A fully functional command and control set for the VLBI transfer application will be available via command line interface or a GUI.

Corresponding to the release of the software a users manual and design documentation will be made available.

3. Summary

The Mark 5C software development road map, being supported by both MIT Haystack Observatory and NRAO, with the goal to support 4 Gbps recording capability was presented. Preliminary versions of the software are expected in August 2008 and the official release date of the application of December 2008.

References

[1] Whitney, A. The Mark 5C VLBI Data System. IVS 2008 General Meeting Proceedings. This volume.

[2] MIT Haystack Observatory, NRAO, Mark 5C Specification. NRAO VLBA Sensitivity Upgrade Memo 12 or Mark 5 Memo 57, Version 1.0 Feb. 19, 2008; http://www.haystack.edu/tech/vlbi/mark5/mark5_memos/057.pdf.

[3] Whitney, A. VLBI Standard Interface — Software. 2003;

http://www.haystack.edu/tech/vlbi/vsi/docs/2003_02_13_vsi-s_final_rev_1.pdf.

[4] MIT Haystack Observatory, NRAO, Mark 5C Software Interface Specification. NRAO VLBA Sensitivity Upgrade Memo 18 or Mark 5 Memo 61, Jan. 7, 2008; http://www.haystack.edu/tech/vlbi/mark5/mark5_memos/061.pdf.

[5] MIT Haystack Observatory, NRAO, Mark 5C Data Frame Specification. NRAO VLBA Sensitivity Upgrade Memo 13 or Mark 5 Memo 58, Version 1.0 Feb. 19, 2008;
http://www.haystack.edu/tech/vlbi/mark5/mark5_memos/058.pdf.

[6] Filesystem in UserSpace, FUSE, Open Source Software;
http://fuse. sourceforge.net/fuse.

[7] Wagner, J. fuseMk5A Software Application;
http://sourceforge.net/projects /fusemk5a/.