MCS - Data Recorder Preliminary Design & Verification

Christopher Wolfe*, Steve Ellingson, Cameron Patterson

August 26, 2009

Contents

*Bradley Dept. of Electrical & Computer Engineering, 302 Whittemore Hall, Virginia Polytechnic Institute & State University, Blacksburg VA 24061 USA. E-mail: chwolfe2@vt.edu

1 Introduction

1.1 Purpose

The MCS-DR, or "monitoring and control system - data recorder" is a data capture and storage system for the LWA radio telescope project implemented using general-purpose, commercially available off-the-shelf (COTS) components. A preliminary design for the MCS-DR has been completed and validated. Specifically, we have demonstrated the ability to reliably record up to 10 hours of data at 115 MiB/s. This document will summarize progress made in design and testing of the MCS-DR and will describe the hardware and software components that make up the MCS-DR.

1.2 Background

MCS-DR is part of MCS "monitoring and control system". The MCS, in turn, is part of the LWA "Long Wavelength Array". The MCS-DR records the data generated by the LWA's Digital Processing (DP) subsystem, and each individual computer is connected by 10-Gigabit Ethernet (10 GbE) to one of five possible inputs from the DP [1]. MCS controls storage and retrieval of data to and from MCS-DR. The design of each computer in the MCS-DR subsystem is identical except for configuration files.

1.3 Document Conventions

Numbers, units, and their associated prefixes and suffixes conform to the standard of IEC 60027-2 [2]. Specifically, the prefixes Ki, Mi, Gi, and Ti refer to 2^{10}, 2^{20}, 2^{30}, and 2^{40}, respectively. Likewise, the prefixes K, M, G, and T refer to 10^3, 10^6, 10^9, and 10^{12}, respectively. If a unit specifies a binary size or rate, an uppercase B represents a byte, whereas a lowercase b indicates an individual bit (i.e. MB = Megabyte, or 1,000,000 bytes, and Kb = kibibit or 1,024 bits).

1.4 Summary of Current Status of MCS-DR Development

The MCS-DR PC hardware selection's suitability has been verified in each of three critical aspects. The three critical aspects are writing to disk, moving data through memory and kernel function calls, and moving data from the physical network into system memory – all at the target data rate

of 115 MiB/s. This exceeds the highest rate required, which is 112 MiB/s (corresponding to TBN mode at it's largest specified bandwidth). Tests have verified the ability to record data streams at 115 MiB/s for a period of at least ten hours. The absolute ceiling on recording speed has not been established, but may be in the neighborhood of 150 MiB/s on average, or 200 MiB/s with optimal circumstances such as short recordings at the very beginning of the drive.

The MCS-DR PC is able to listen for and respond to a set of message types from MCS. The message types which are currently implemented allow for initiation of recording, data verification, and other development and testing functions. It is anticipated that some of these commands will become part of an MCS-DR ICD. Ultimately, each MCS-DR PC will have "subsystem status", be fully compliant with the MCS Common ICD, and be regarded by MCS in exactly the same way as the other major LWA station subsystems (e.g.: SHL, ASP, DP, and so on).

2 Design Overview

The following sections describe in greater detail the hardware and software components of the MCS-DR PC design, as well as the tests and methods used to verify different aspects of the design. The first section presents a brief overview of the hardware and software components and subsequent sections discuss factors that played a role in hardware selection and software organization.

2.1 Hardware Brief

Figure 1 outlines the hardware organization for an individual MCS-DR PC. Each MCS-DR PC is comprised of a stock Dell PC with two add-in cards and an external RAID enclosure. The stock PC is a Dell Studio XPS™model 435MT computer. The Studio XPS™435MT is based on the Intel® Core™ i7-940 processor which has four Hyper-threaded™cores operating at 2.93 GHz. At the time of purchase, the system was customized to have 6 GiB of Tri-Channel DDR3 SDRAM memory operating at 1066 MHz. The system HDD is a Seagate 1 TB 7200 RPM SATA-II hard disk drive with 16 MiB of cache memory. The Studio XPS™435MT also includes an onboard Intel gigabit Ethernet (GBE) adapter which is used for communication with the station MCS.

For storage, an American Media Systems® Venus-T5™ eSATA RAID external enclosure is connected to the system via the eSATA cable supplied with the enclosure. The Studio XPS™435MT system has an built-in eSATA port, but this was unsuitable for the needs of the MCS-DR PC (see discussion in Section 2.3 later this document), and an eSATA adapter was used instead. A Silicon

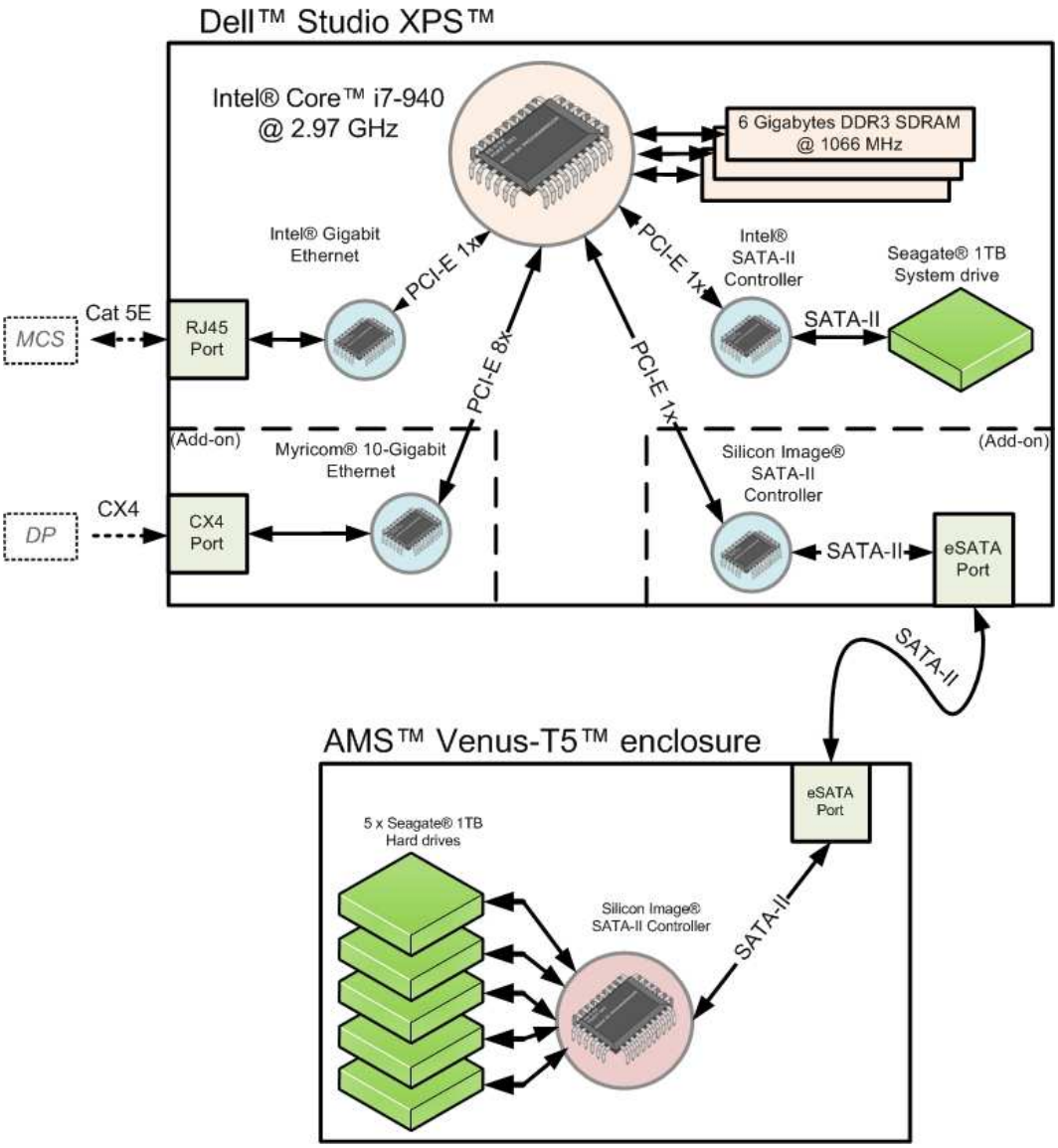

Figure 1: MCS-DR PC System Overview

Image® PCI-E ("PCI express") 1x external SATA-II adapter based on the SteelVine™series of storage controllers is added to the stock system. The eSATA adapter came packaged with the Venus-T5™ RAID enclosure. The enclosure contains five hot-swappable Seagate 1 TB 7200 RPM SATA-II hard disk drives, and provides a total storage capacity of 5 TB less file system and formatting overhead. This is the configuration that has been implemented and verified.

An alternative to the Venus-T5 enclosure being considered for the delivered version is a rack-mountable 1U SATA II enclosure. This option would reduce the server rack space required for the MCS-DR, and would facilitate easy removal and replacement of data storage while keeping the drives together. Testing of this option is will begin shortly.

The MCS-DR PC's high speed Ethernet interface is a Myricom® model 10G-PCIE-8A-C+E 10 gigabit Ethernet adapter. The adapter is a PCI-E 8x adapter which connects to the network via a 10GBase-CX4 physical interface. The cables used to connect the MCS-DR PC to the DP subsystem are Myricom® 10G-CX4-1M 10GBase-CX4 copper cables.

2.2 Software Brief

The MCS-DR PC software is a BSD-Sockets based Linux application operating in a polling paradigm. The software is written in ANSI C, and is a single process, though interacting with the host computer requires short-lived child processes in a few instances. The software uses the Posix.1b real-time extensions library (librt[1]) for asynchronous transfers to and from disk, and for queuing messages from MCS. Figure 2 illustrates the organization of the software and outlines the scope of the application within the MCS-DR PC. The operating system is Ubuntu Desktop 9.04 AMD64. The main processing loop of the application polls a socket for command messages from MCS. Upon receiving commands to start a specific operation, the main loop enables components of the data path necessary for receiving data from the network, writing data to disk, reading data from disk, and transmitting data to the network. The main processing loop then checks each portion of the data path to see if action is required to move data along, taking action where necessary. Once an operation is complete, the data path is disabled and the system is returned to the idle state, making it available for future operations. To interact with the host computer and operating system, the software contains functions which gather environment and machine status information such as CPU and hard drive temperatures, free disk space, and so forth, as well as functions to format the drive array, mount partitions, and perform general maintenance functions.

[1]Documentation available online at http://compute.cnr.berkeley.edu/cgi-bin/man-cgi?librt+3

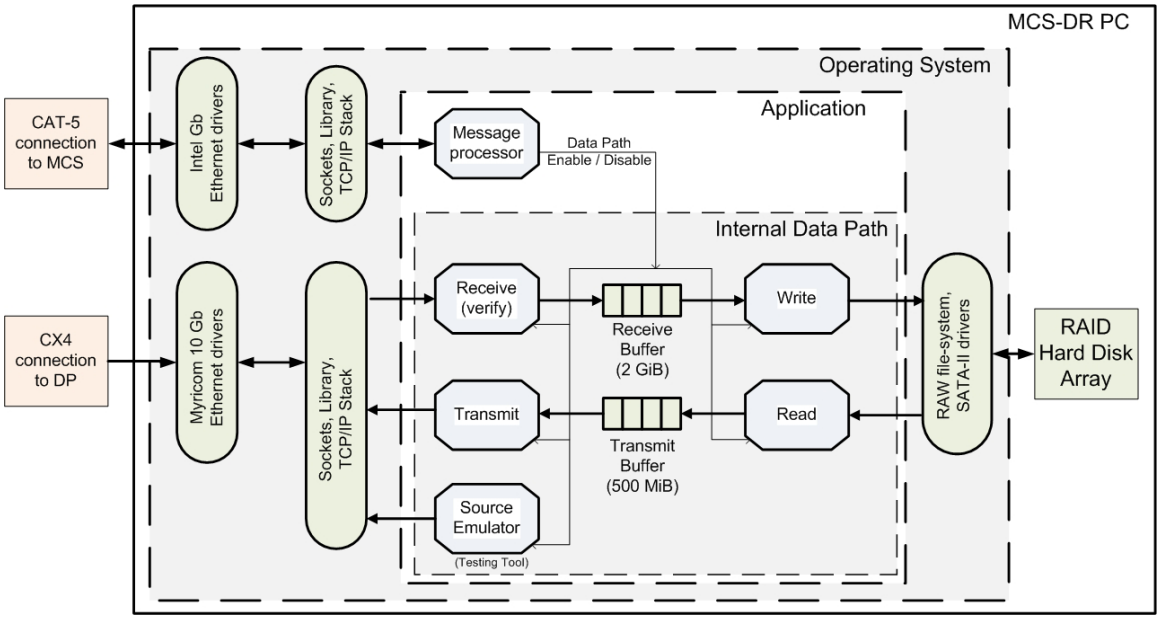

Figure 2: MCS-DR PC System Software Overview

Initial tests with the ext2 file system yielded inconsistent results. Consequently, we implemented a custom file system based on raw access to the drive array. The file system used by the software is a flat file system with a simple bitmap structure written to the beginning of the drive, and the remaining space available for file storage. The current (validated) file system supports as many as 1023 recordings, all of which combined may be up to 4.9 TiB in size. The file system can be easily modified to support an arbitrarily large number of files.

The benefits of using a raw file system are two-fold. Bypassing the ext2 file system allows writing contiguously without having to periodically update inodes and inode tables. Also, it allows rapid file deletion, and formatting – operations that could take hours with ext2. Because speed is critical, kernel caching is disabled for file reads and writes. The Linux kernel's caching algorithms are optimized for random access, but data streams are recorded sequentially. Tests with caching enabled were unable to meet the rate requirement. Tests with caching disabled, however, were able to meet and exceed the requirements.

2.3 Hardware Considerations

The MCS-DR PC must be capable of recording data streams at a sustained rate of 112 MiB/s with data payload sizes depending on the data source and operational mode. The hardware for the MCS-DR PC was selected such that the speed of all components of the internal data path exceeded this requirement. The only deviation from this is with the hard drives, where the speed requirement is met by having 5 drives in RAID 0 instead of one drive capable of the desired rate.

The Core™ i7 series of processors from Intel® fit the need for a variety of reasons. In addition to providing four distinct cores, each of which is Hyper-threaded to provide two virtual cores, for a total of eight virtual cores, the i7 does away with traditional bus-based architecture, which allows for high-speed serial communication between the CPU and peripherals. With three independent memory channels operating at 1066 MHz each, the Core™ i7 processor fits well with the intended usage profile of the MCS-DR.

The computer chosen for the MCS-DR is the Studio XPS™ desktop pc from Dell™. Based on the Core™ i7-940 processor, with 6 GiB of high speed DDR3 memory, it also offers a built-in eSATA connection, three PCI-E 1x slots, one PCI-E 16x slot, and a 1TB hard drive suitable for containing the operating system and system software. In addition to meeting the hardware requirements, the Studio XPS™ is cost-effective option starting at around $750.00 USD, which is on track with the target unit price of around $2000.00 USD per MCS-DR PC when combined with a low-cost (approx. $675.00 USD w/ drives) RAID configuration and the Myricom 10 GbE Ethernet adapter (approx. $495.00 USD).

For storage, the American Media Systems® Venus-T5™ external SATA-II enclosure was selected. With high-speed stream recording, hard drives, rather than the busses they are attached to, tend to be the bottleneck. Since the Venus-T5™ is capable of supporting five drives, the effective maximum stream rate is multiplied by five for a RAID level 0 configuration. The enclosure was selected because it complies with the SATA-II standard yielding a theoretical maximum transfer rate of 300 MiB/s – well in excess of the requirements of the MCS-DR PC. Initially, the onboard Intel SATA-II eSATA port was to be used to connect the RAID enclosure to the system. However, the controller does not support SATA port-multiplication and thus was unable to make use of all 5 drives in the Venus-T5 enclosure. Fortunately, though, the Venus-T5 ships with a 2-port PCI-E 1x eSATA adapter which supports port-multiplication. Testing confirms that it meets the needs of the MCS-DR.

The final hardware component of the MCS-DR is a 10 GbE adapter from Myricom®. Each MCS-DR PC records data from one of the outputs of the station DP. The 10G-PCIE-8A-C+E from

Myricom offers 10 Gb/s transfer rates, large receive off-loading, automatic checksum generation, and an open-source API for software interfacing (as well as open-source, Linux-friendly drivers).

2.4 Software Considerations

As a low-cost alternative to available commercial data capture options, it was desired to avoid proprietary technologies and their consequent licensing royalties. As a result, the software of the MCS-DR (including its operating system) is based exclusively on public domain and open source software.

The reception of UDP datagrams and actual recording of UDP datagrams occur in different portions of the MCS-DR PC's data path. The first of these, the "Receive" portion, is responsible for moving data from the network adapter into the application's memory space. At the hardware level, the Ethernet adapter uses DMA[2] transfers to place packet data into system memory. The act of receiving from the socket copies the data into an intermediate buffer where it is then removed by the "Write" portion of the data path as it writes the data to the hard drive array. Because of the magnitude of the transfer rates involved, efficiency and economic use of the CPU and memory is critical. Consequently, the data is only copied once from the time it arrives in system memory until the time it is written to the hard drives. Tests (See Section 3) have shown that the system supports transferring data this way at data rates exceeding 450 MiB/s, and that the performance bottleneck of the system, as expected, is with the hard drives themselves.

3 Testing and Verification

Several key aspects of the prototype system have been characterized, and have been tested to ensure they meet the requirements set forth in the "MCS Subsystem Definition" [1].

3.1 Hard Drive Performance

The draft design of the MCS-DR PC included Seagate's 7200.11 series of 1 TB drives, but problems with meeting the required rates prompted us to use Seagate's SV35.5 series of 1 TB drives instead. With the 7200.11 series drives having a maximum sustained transfer rate of 120 MiB/s, and with five

[2]Direct Memory Access

drives in a RAID 0 configuration, the limiting factor should have been the SATA-II bus. However this was not the case, and sustained transfer rates for the RAID as a whole were limited to about 120 MiB/s. Most of this loss was due to the way Linux caches writes to the drive, but part was due to the non-optimal factory tuning of the drives' firmware. When testing the SV35.5 series with the same options, the performance was approximately the same. However, more recent tests circumvented kernel cache usage and were able to achieve rates up to 150 MiB/s for several hours. Tests of the 7200.11 series have not yet been performed with this option. However, given the negligible increased cost of the SV35.5 option, we do not plan to consider the 7200.11 series further.

The test itself consisted of opening several files on the RAID array and writing known data that was easily verified afterwards. The blocks of data written to the files consisted of either 3900 bytes or 1008 bytes, the first 8 of which were used as a serial identifier, while the remaining bytes were filled with an 8-bit counter value, increasing by one for each successive byte and rolling over to 0 after 255. Each time the block was written to the file, the serial identifier was incremented by 1 before writing the block again. This test was performed several times for varying durations, and the maximum sustained transfer rate was approximately 120 MiB/s when using the kernel cache for all tests shorter than ten hours. For tests in which the kernel cache was bypassed, a maximum sustained transfer rate has not been established, though tests indicate this number to be at least 150 MiB/s for all tests shorter than 8 hours.

3.2 File System Performance

The MCS-DR uses a software RAID level 0 array with a custom, raw-mode file system. Other file systems were considered before selecting a raw file system. Knowing that the journaling operations of the ext3 file system would require too much overhead, the first choice of file systems was ext2 for simplicity. Initial testing had established that the ext2 file system was capable of meeting the 112 MiB/s requirement, but the need to bypass kernel caching made working with ext2 files difficult. Tests were run with the xfs, fat32[3], and raw file systems. The xfs test yielded a transfer rate of 75 MiB/s, and the fat32 test resulted in an inconsistent 70 MiB/s, and tests of the raw file system achieved rates of 150 MiB/s.

[3]It should be noted that the fat32 file system would not have supported the full 5 TiB capacity of the RAID array

3.3 Memory and CPU Testing

The first tests which incorporated socket-based communication aimed at verifying that the CPU and memory could maintain sufficient transfer rates. By creating a socket connection to "localhost", the test was able to send UDP datagrams from one part of the data path and have them received in another. Because the Linux socket library copies the data from application memory into system memory when sending, and from system memory to application memory when receiving, this test effectively measured the maximum transfer rate of the CPU and memory under the same usage pattern as the MCS-DR requires. Like the hard drive benchmarking tests, packets consisting of a serial identifier, and a series of 8-bit counter values were used. In this case, however, the exhaustive checking of the entire packet would have perturbed the results of the test, and only the serial identifier was verified. The noted maximum transfer rate was 465 MiB/s with 500 byte packets. The performance limitation in this scheme comes not from the overall data rate, but rather it exists as a relationship between the size of the packets and the number of packets transfers required per second. With arbitrarily large packets, the limit approaches the maximum memory bandwidth of the system, and as the packets get smaller, the overhead of kernel IO_MMU[4] calls required to copy the data dominates. Since the minimum packet size of TBN, TBW, and DRX packets is the TBN packet size of 1008 bytes, 500 bytes was arbitrarily chosen as being sufficiently smaller than the packets of interest as to ensure that success of the test would imply that any larger sized packet transfer would also meet the data rate requirements.

3.4 Network Performance

Testing has been performed which verified the network adapter's ability to meet the system requirements. Two sets of tests were performed. The first, preliminary tests of hardware driver and functionality were included as part of the driver package from by Myricom. The included loopback test measures transfer speeds of the network adapter. This test was run for four hours and for ten hours, with the transfer rate approaching 4 Gib/s in both cases. The second set of tests were the duration tests described in the next section. The successes of both series of tests are sufficient to validate the network hardware selection because they utilize all of the hardware components essential to the MCS-DR PC's core functionality.

[4]Input Output Memory Management Unit

3.5 Complete System / Duration Testing

To perform the complete system and duration test, the hardware from a second MCS-DR PC was used to emulate the DP subsystem and provide a stream of data for the MCS-DR PC under test. The duration tests involved streaming data from the emulated DP and having it recorded on an MCS-DR PC. Tests confirmed that the hardware is capable of the required rates. The test was to generate packets with a specific pattern on the emulated DP, transmit them to MCS-DR PC under test, and record the packets to disk. Following the recording part of the test, the recorded file was then checked to confirm that no errors were introduced in the data transfer/storage process. This test used the TBN packet size of 1008 bytes, the first eight of which indicate a serial identifier, and the remaining 1000 bytes were patterned with an 8-bit counter value that rolled over to 0 when it reached 255. For ten hours, packets were transmitted at a data rate of 115 MiB/s, and the resulting recording was successfully verified to match the transmitted pattern.

4 Ongoing and Future Efforts

Several tasks still remain to be completed before the MCS-DR will be CDR-ready, though testing so far indicates that no identifiable and significant risk remains. Software changes will likely be a matter of reworking the interface and command set to suit the MCS and other subsystems. Outstanding tasks include:

- Documentation Tasks
  - MCS-DR ICD and MIB specification
- Software Tasks
  - MCS Common ICD compliance verification
  - Application 1st "Release" Candidate
  - Source code review and validation
  - Doxygen source code documentation
  - Regression testing
  - Evaluate possibility of USB-bootable image with RAID internal to Studio XPS system
- Hardware Tasks

  - Evaluate/test 1U rack-mount enclosure to replace Venus T5
  - Re-testing of 7200.11 series drives but with cache disabled. (time permitting)
  - Establish maximum rate sustainable for 2, 4, 6, 8, and 10 hour observations. (time permitting)

- Candidate Future (Post-CDR or Post-IOC) Development Tasks
  - Linux VFS-compliant file system extension to allow mounting of MCS-DR PC's custom, raw file system.
  - Explore possibility of servicing multiple DP sources from a single MCS-DR PC
  - Compact version of MCS-DR consisting of rack-mountable PC with only internal drives (entire MCS-DR in this form would be very easily transportable and would still provide 2-3 TiB (4-6 hours worst case) of recording)

5 Document History

- Version 0.4 (Aug 26, 2009):
  - Fourth draft of document.
  - Updated Title, Figures, Ongoing and Future Efforts, Software Brief
  - removed typos
- Version 0.3 (Aug 26, 2009):
  - Third draft of document.
  - Included discussion of file system changes
  - extraneous content removed
- Version 0.2 (Aug 23, 2009):
  - Second draft of document with changes in RE: hardware, software, and testing.
- Version 0.1 (Jul 14, 2009):
  - Initial draft of document.

References

[1] S. Ellingson, "MCS Subsystem Definition," Ver. 2, Long Wavelength Array Engineering Memo MCS0004, Feb. 23, 2009. [online] http://www.ece.vt.edu/swe/lwavt/.

[2] International Electrotechnical Commission, "Letter symbols to be used in electrical technology Part 2: Telecommunications and electronics," Third Ed., 2005. [online] http://www.iec.ch/