Publication Year	2015
Acceptance in OA	2020-05-11T18:01:12Z
Title	The Murchison Widefield Array Correlator
Authors	Ord, S. M., Crosse, B., Emrich, D., Pallot, D., Wayth, R. B., Clark, M. A., Tremblay, S. E., Arcus, W., Barnes, D., Bell, M., BERNARDI, GIANNI, Bhat, N. D. R., Bowman, J. D., Briggs, F., Bunton, J. D., Cappallo, R. J., Corey, B. E., Deshpande, A. A., deSouza, L., Ewell-Wice, A., Feng, L., Goeke, R., Greenhill, L. J., Hazelton, B. J., Herne, D., Hewitt, J. N., Hindson, L., Hurley-Walker, N., Jacobs, D., Johnston-Hollitt, M., Kaplan, D. L., Kasper, J. C., Kincaid, B. B., Koenig, R., Kratzenberg, E., Kudryavtseva, N., Lenc, E., Lonsdale, C. J., Lynch, M. J., McKinley, B., McWhirter, S. R., Mitchell, D. A., Morales, M. F., Morgan, E., Oberoi, D., Offringa, A., Pathikulangara, J., Pindor, B., Prabu, T., Procopio, P., Remillard, R. A., Riding, J., Rogers, A. E. E., Roshi, A., Salah, J. E., Sault, R. J., Udaya Shankar, N., Srivani, K. S., Stevens, J., Subrahmanyan, R., Tingay, S. J., Waterson, M., Webster, R. L., Whitney, A. R., Williams, A., Williams, C. L., Wyithe, J. S. B.
Publisher's version (DOI)	10.1017/pasa.2015.5
Handle	http://hdl.handle.net/20.500.12386/24709
Journal	PUBLICATIONS OF THE ASTRONOMICAL SOCIETY OF AUSTRALIA
Volume	32

Publications of the Astronomical Society of Australia (PASA), Vol. 32, e006, 14 pages (2015).

doi:10.1017/pasa.2015.5

The Murchison Widefield Array Correlator

S. M. Ord[1,2,23], B. Crosse[1], D. Emrich[1], D. Pallot[1], R. B. Wayth[1,2], M. A. Clark[3,4,5], S. E. Tremblay[1,2], W. Arcus[1], D. Barnes[6], M. Bell[7], G. Bernardi[4,8,9], N. D. R. Bhat[1], J. D. Bowman[10], F. Briggs[11], J. D. Bunton[12], R. J. Cappallo[13], B. E. Corey[13], A. A. Deshpande[14], L. deSouza[7,12], A. Ewell-Wice[15], L. Feng[15], R. Goeke[15], L. J. Greenhill[4], B. J. Hazelton[16], D. Herne[1], J. N. Hewitt[15], L. Hindson[17], N. Hurley-Walker[1], D. Jacobs[10], M. Johnston-Hollitt[17], D. L. Kaplan[18], J. C. Kasper[4,19], B. B. Kincaid[13], R. Koenig[12], E. Kratzenberg[13], N. Kudryavtseva[1], E. Lenc[2,7], C. J. Lonsdale[13], M. J. Lynch[1], B. McKinley[11], S. R. McWhirter[13], D. A. Mitchell[2,12], M. F. Morales[16], E. Morgan[15], D. Oberoi[20], A. Offringa[2,11], J. Pathikulangara[12], B. Pindor[21], T. Prabu[14], P. Procopio[21], R. A. Remillard[15], J. Riding[21], A. E. E. Rogers[13], A. Roshi[22], J. E. Salah[13], R. J. Sault[21], N. Udaya Shankar[14], K. S. Srivani[14], J. Stevens[12], R. Subrahmanyan[2,14], S. J. Tingay[1,2], M. Waterson[1,11], R. L. Webster[2,21], A. R. Whitney[13], A. Williams[1], C. L. Williams[15] and J. S. B. Wyithe[2,21]

[1]International Centre for Radio Astronomy Research (ICRAR), Curtin University, Perth, Australia
[2]ARC Centre of Excellence for All-sky Astrophysics (CAASTRO)
[3]NVIDIA, Santa Clara, California, USA
[4]Harvard-Smithsonian Center for Astrophysics, Cambridge, USA
[5]California Institute of Technology, California, USA
[6]Monash University, Melbourne, Australia
[7]University of Sydney, Sydney, Australia
[8]Square Kilometre Array South Africa (SKA SA), Cape Town, South Africa
[9]Department of Physics and Electronics, Rhodes University, Grahamstown, South Africa
[10]Arizona State University, Tempe, USA
[11]The Australian National University, Canberra, Australia
[12]CSIRO Astronomy and Space Science, Australia
[13]MIT Haystack Observatory, Westford, MA, USA
[14]Raman Research Institute, Bangalore, India
[15]MIT Kavli Institute for Astrophysics and Space Research, Cambridge, USA
[16]University of Washington, Seattle, USA
[17]Victoria University of Wellington, New Zealand
[18]University of Wisconsin–Milwaukee, Milwaukee, USA
[19]University of Michigan, Ann Arbor, USA
[20]National Center for Radio Astrophysics, Pune, India
[21]The University of Melbourne, Melbourne, Australia
[22]National Radio Astronomy Observatory, Charlottesville, USA
[23]Email: stephen.ord@curtin.edu.au

(Received October 10, 2014; Accepted January 22, 2015)

Abstract

The Murchison Widefield Array is a Square Kilometre Array Precursor. The telescope is located at the Murchison Radio–astronomy Observatory in Western Australia. The MWA consists of 4 096 dipoles arranged into 128 dual polarisation aperture arrays forming a connected element interferometer that cross-correlates signals from all 256 inputs. A hybrid approach to the correlation task is employed, with some processing stages being performed by bespoke hardware, based on Field Programmable Gate Arrays, and others by Graphics Processing Units housed in general purpose rack mounted servers. The correlation capability required is approximately 8 tera floating point operations per second. The MWA has commenced operations and the correlator is generating 8.3 TB day^{-1} of correlation products, that are subsequently transferred 700 km from the MRO to Perth (WA) in real-time for storage and offline processing. In this paper, we outline the correlator design, signal path, and processing elements and present the data format for the internal and external interfaces.

Keywords: instrumentation: interferometers, techniques: interferometric

1. INTRODUCTION

The MWA is a 128 element dual polarisation interferometer, each element is a 4 × 4 array of analog beam formed dipole antennas. The antennas of each array are arranged in a regular grid approximately 1 m apart, and these small aperture arrays are known as tiles. The science goals that have driven the MWA design and development process are discussed in the instrument description papers (Tingay et al. 2013a; Lonsdale et al. 2009), and the MWA science paper (Bowman et al. 2013). These are (1) the detection of redshifted 21 cm neutral hydrogen from the Epoch of Re-ionization; (2) Galactic and extra-Galactic surveys; (3) time-domain astrophysics; (4) solar, heliospheric and ionospheric science and space weather.

1.1. Specific MWA correlator requirements

The requirements and science goals have driven the MWA into a compact configuration of 128 dual polarisation tiles. 50 tiles are concentrated in the 100-m diameter core, with 62 tiles distributed within 750 m and the remaining 16 distributed up to 1.5 km from the core.

The combination of the low operating frequency of the MWA and its compact configuration allow the correlator to be greatly reduced in complexity, however this trade-off does drive the correlator specifications. Traditional correlators are required to compensate for the changing geometry of the array with respect to the source in order to permit coherent integration of the correlated products. In the case of the MWA, no such corrections are performed. This drives the temporal resolution specifications of the correlator and forces the products to be rapidly generated in order to maintain coherence. Tingay et al. (2013a) list the system parameters and these include a temporal resolution of 0.5 s.

The temporal decoherence introduced by integrating for 0.5 s without correcting for the changing array geometry; and the requirement to image a large fraction of the primary beam, drive the system channel resolution specification to 10 kHz. It should be noted that this does not drive correlator complexity as it is total processed bandwidth that is the performance driver, not the number of channels. Albeit the number of output channels does drive the data storage and archiving specifications. The MWA correlator is required to process the full bandwidth as presented by the current MWA digital receivers, which is 30.72 MHz per polarisation.

In summary, in order to meet the MWA science requirements, the correlator is required to correlate 128 dual polarisation inputs, each input is 3072 × 10 kHz in bandwidth. The correlator must present products, integrated for no more than 0.5 s at this native channel resolution, to the data archive for storage.

1.2. Benefits of software correlator implementations

The correlation task has previously been addressed by Application Specific Integrated Circuits (ASICs) and Field Programmable Gate Arrays (FPGAs). However the current generation of low frequency arrays including the MWA (Wayth, Greenhill, & Briggs 2009), LOFAR (van Haarlem et al. 2013), PAPER (Parsons et al. 2010) and LEDA (Kocz et al. 2014) have chosen to utilise, to varying degrees, general purpose computing assets to perform this operation. The MWA leverages two technologies to perform the correlation task, a Fourier transformation performed by a purpose built FPGA-based solution, and a XMAC utilising the xGPU library[1] described by Clark, La Plante, & Greenhill (2012), also used by the PAPER telescope and the LEDA project. The MWA system was deployed over 2012/13 as described by Tingay et al. (2013a), and is now operational.

The MWA is unique amongst the recent low frequency imaging dipole arrays, in that it was not designed to utilise a software correlator at the outset. The flexibility provided by the application of general purpose, GPU based, software solution has allowed a correlator to be rapidly developed and fielded. This was required in order to respond to a changed funding environment, which resulted in a significant design evolution from that initially proposed by Lonsdale et al. (2009) to that described by Tingay et al. (2013a). Expeditious correlation development and deployment was possible because General Purpose GPU (GPGPU) computing provides a compute capability, comparable to that available from the largest FPGAs, in a form that is much more accessible to software developers. It is true that GPU solutions are more power intensive than FPGA or ASIC solutions, but their compute capability is already packaged in a form factor that is readily available and the development cycle is identical to other software projects. Furthermore the GPU processor lifecycle is very fast and it is also generally very simple to benefit from architecture improvements. For example, the GPU kernel used as the cross-multiply engine in the MWA correlator will run on any GPU released after 2010. We could directly swap out the GPU in the current MWA cluster, replace them with cards from the Kepler series (K20X), and realise a factor of 2.5 increase in performance (and a threefold improvement in power efficiency).

The organisation of this paper begins with a short introduction to the correlation problem, followed by an outline of the MWA correlator design and then a description of the sub-elements of the correlator following the signal path; we finally discuss the relevance of this correlator design to the Square Kilometre Array (SKA) and there are several Appendices describing the various internal and external interfaces.

2. THE CORRELATION PROBLEM

A traditional telescope has a filled aperture, where a surface or a lens is used to focus incoming radiation to a focal point. In contrast, in an interferometric array like the MWA the purpose of a correlator is to measure the level of signal correlation between all antenna pairs at different frequencies

[1] xGPU available at: https://github.com/GPU-correlators/xGPU

across the observing band. These products can then be added coherently, and phased, or focussed, to obtain a measurement of the sky brightness distribution in any direction.

The result of the correlation operation is commonly called a *visibility* and is a representation of the measured signal power (brightness) from the sky on angular scales commensurate with the distance between the constituent pair of antennas. Visibilities generated between antennas that are relatively far apart measure power on smaller angular scales and vice versa. When all visibilities are calculated from all pairs of antennas in the array many spatial scales are sampled (see Thompson, Moran, & Swenson (2001) for an extensive review of interferometry and synthesis imaging). When formed as a function of observing frequency (ν), this visibility set forms the *cross-power spectrum*. For any two voltage time-series from any two antennas V_1 and V_2 this product can be formed in two ways. First the cross correlation as a function of lag, τ, can be found, typically by using a delay line and multipliers to form the lag correlation between the time-series.

$$(V_1 \star V_2)(\tau) = \int_{-\infty}^{\infty} V_1(t)V_2(t-\tau)dt. \quad (1)$$

The cross power spectrum, $S(\nu)$, is then obtained by application of a Fourier transform

$$S_{12}(\nu) = \int_{-\infty}^{\infty} (V_1 \star V_2)(\tau)e^{-2\pi i\nu\tau}d\tau. \quad (2)$$

When the tasks required to form the cross power spectrum are performed in this order (lag cross-correlation, followed by Fourier transform) the combined operation is considered an *XF* correlator. However, the cross correlation analogue of the convolution theorem allows Equation (2) to be written as the product of the Fourier transform of the voltage time-series from each antenna

$$S_{12}(\nu) = \int_{-\infty}^{\infty} V_1(t)e^{-2\pi ivt}dt \times \int_{-\infty}^{\infty} V_2(t)e^{-2\pi ivt}dt. \quad (3)$$

Implemented as described by Equation (3) the operation is an *FX* correlator. For large-N telescopes the FX correlator has a large computational advantage. In an XF correlator for an array of N inputs the cross correlation for all baselines requires $\mathcal{O}(N^2)$ operations for every lag, and there is a one to one correspondence between lags and output channels, F, resulting in $\mathcal{O}(FN^2)$ operations to generate the full set of lags. The Fourier transform requires a further $\mathcal{O}(F \log_2 F)$ operations, but this can be performed after averaging the lag spectrum and is therefore inconsequential. For the FX correlator, we require $\mathcal{O}(NF \log_2 F)$ operations for the Fourier transform of all input data streams, but only $\mathcal{O}(N^2)$ operations per sample for the cross multiply (although we have F channels the sample rate is now lower by the same factor). Therefore, as long as N is greater than $\log_2 F$ there is a computational advantage in implementing an FX correlator. XF correlators have been historically favoured by the astronomy community, at least in real-time applications, as until very recently N has been small, and there are disadvantages to the FX implementation. The predominant disadvantage is data growth: the precision of the output from the Fourier transform is generally larger than the input, resulting in a data rate increase. There is also the complexity of implementing the Fourier transform in real-time.

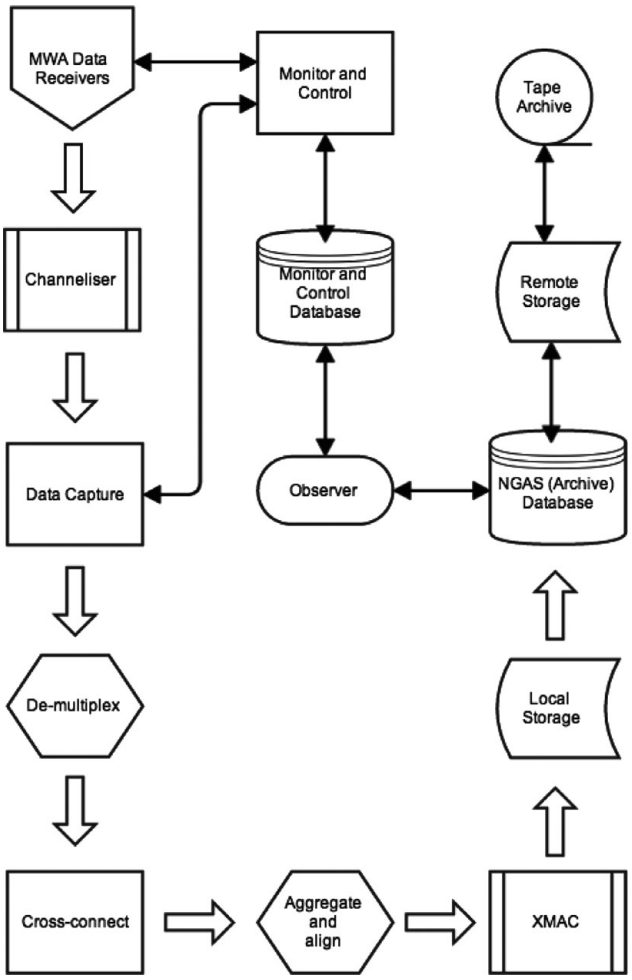

Figure 1. The MWA signal processing path, following the MWA data receivers, in the form of a flow diagram.

3. THE MWA CORRELATOR SYSTEM

The tasks detailed in this section cover the full signal path after digitisation, including fine channelisation, data distribution, correlation and output. In the MWA correlator the F stage is performed by a dedicated channeliser, subsets of frequency channels from all antennas are then distributed to a cluster of processing nodes via an ethernet switch. The correlation products are then assembled and distributed to an archiving system. An outline of the system as a whole is shown in Figures 1 and 2. As shown in Figure 3 the correlator is conceptually composed of 4 sub-packages, the Polyphase Filter bank (PFB), that performs the fine channelisation, the Voltage Capture System (VCS), responsible for converting the data transport protocol into ethernet and distributing the

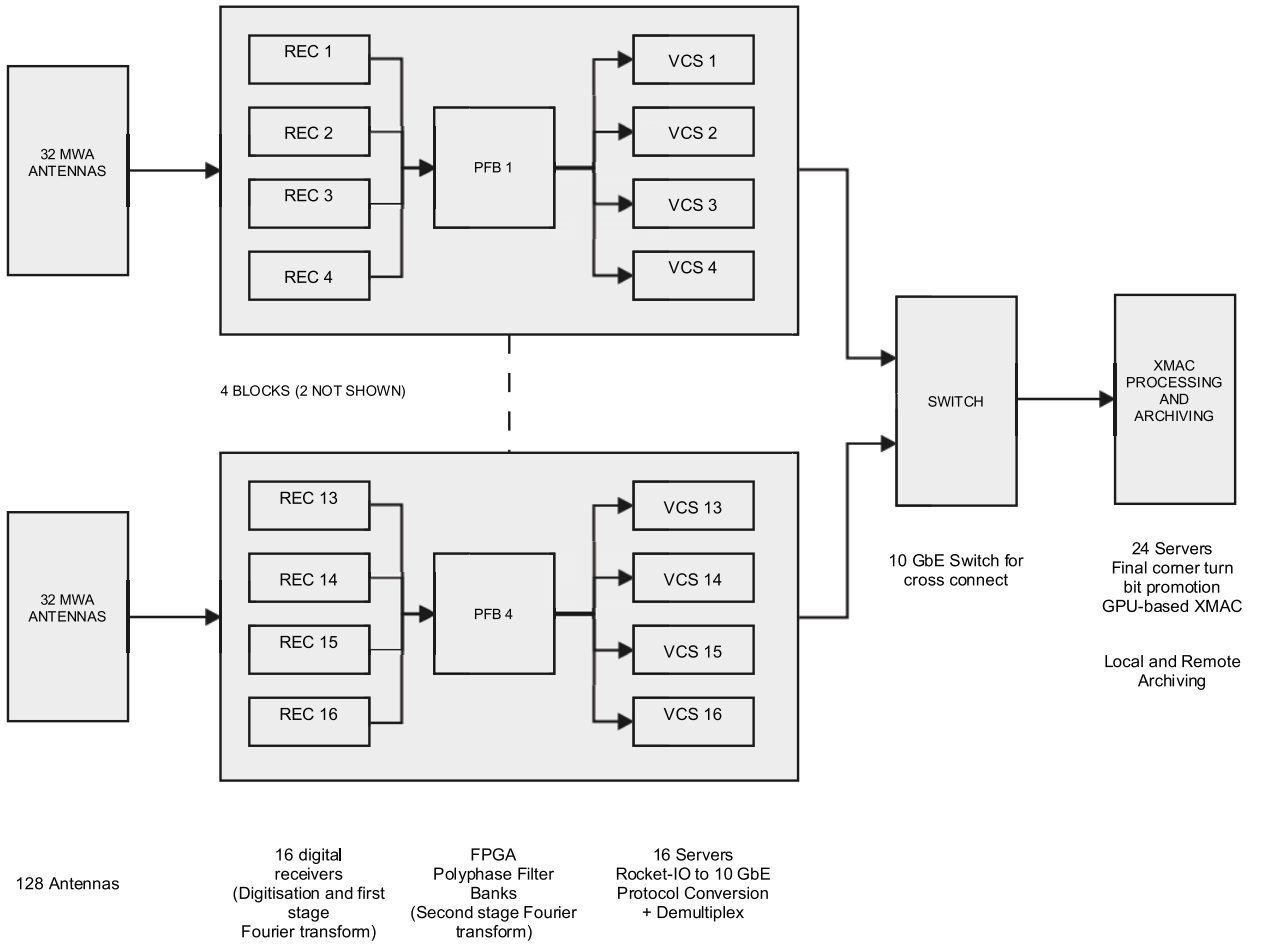

Figure 2. The full physical layout of the MWA digital signal processing path. The initial digitalisation and coarse channelisation is performed in the field. The fine channelisation, data distribution and correlation is performed in the computer facility at the Murchison Radio Observatory, the data products are finally archived at the Pawsey Centre in Perth.

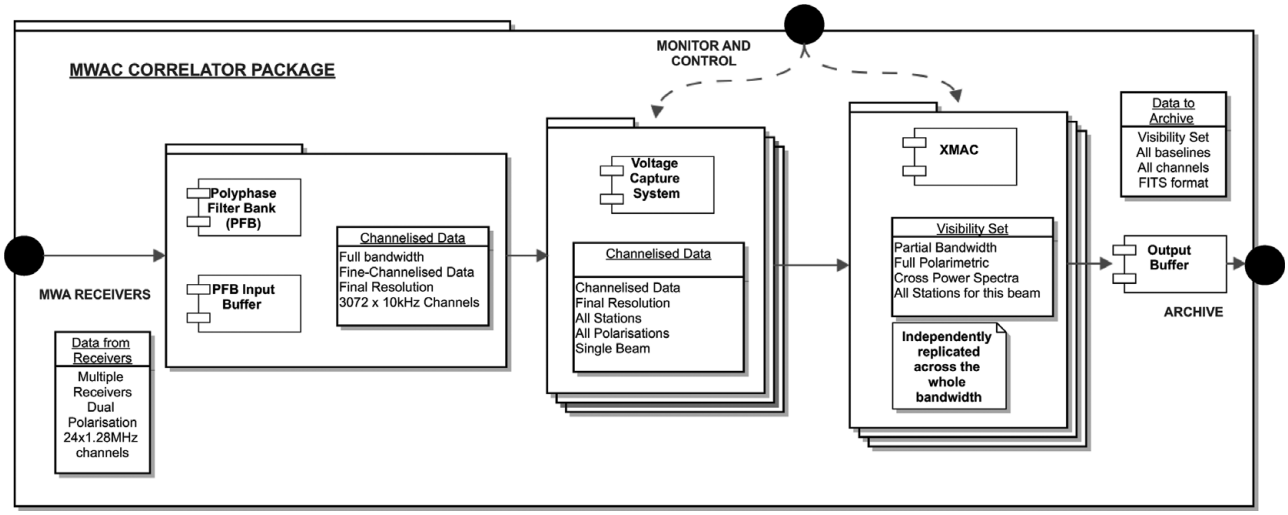

Figure 3. The decomposition of the MWA Correlator system demonstrating the relationship between the major sub-elements.

data, the correlator itself (XMAC), and the output buffering that provides the final interface to the archive. Figure 2 presents the physical layout of the system.

3.1. The system level view

The computer system consists of 40 servers, grouped into different tasks and connected by a 10 GbE data network and several 1 GbE monitoring, command and control networks. The correlator computer system has been designed for ease of maintenance and reliability. All of the servers are configured as *thin-clients* and have no physical hard-disk storage that is utilised for any critical functions–any hard-disk is used for non-critical storage and could be removed (or fail) without limiting correlator operation. All of the servers receive their operating system through a process known as the PXE (Pre-eXecution Environment) boot process, and import all of their software over NFS (Network File System) and mount it locally in memory. The servers are grouped by task into the VCS machines that house the FPGA capture card and the servers that actually perform the XMAC and output the visibility sets for archiving. Any machine can be replaced without any more effort than is required to physically connect the server box and update an entry in the IP address mapping tables.

3.2. The MWA data receivers: Input to the correlator

As described in Prabu, Srivani, & Anish (2014) there are 16 digital receivers deployed across the MWA site. Power and fibre is reticulated to them as described by Tingay et al. (2013a). The individual tiles are connected via coaxial cable to the receivers, the receivers are connected via fibre runs of varying lengths to the central processing building of the MRO, where the rest of the signal processing is located.

3.2.1. The polyphase filterbank

PFBs are commonly used in digital signal communications as a realisation of a Discrete Fourier Transform (DFT; see Crochiere & Rabiner (1983) for a detailed explanation). It specifically refers to the formation of a DFT via a method of *decimating* input samples into a polyphase filter structure and forming the DFT via the application of a Fast Fourier Transform to the output of the polyphase filter (Bunton 2003).

As described by Prabu et al. (2014) the first or *coarse* PFB is as an 8 tap, 512 point, critically sampled, PFB. This is implemented as; an input filtering stage realised by 8×512 point Kaiser windowing function, the output of which is subsampled into a 512 point FFT. This process transforms the 327.68 MHz input bandwidth into 256, 1.28 MHz wide, subbands.

The second stage of the F is performed by four dedicated FPGA based PFB boards housed in two ATCA (Advanced Telecommunications Computing Architecture) racks, which implement a 12 tap, 128 point filter response weighted DFT. Because the first stage is critically sampled the fine channels that are formed at the boundary of the coarse channels are corrupted by aliasing. These channels are processed by the pipeline, but removed during post-processing, the fraction of the band excised due to aliasing is approximately 12%. The second stage is also critically sampled so there is a similar degree of aliasing present in each 10 kHz wide channel. The aliased signal correlates, and the phase of the correlation does not change significantly across the narrow channel, any further loss in sensitivity is negligible.

Development of the PFB boards was initially funded through the University of Sydney and the firmware initially developed by CSIRO. The boards were designed to form part of the original MWA correlator which would share technology with the Square Kilometre Array Molongolo Prototype (SKAMP) (de Souza et al. 2007). Subsequent modifications to the firmware were undertaken at MIT-Haystack and the final boards, with firmware specific to the MWA, were first deployed as part of the 32-tile MWA prototype.

3.2.2. Input format, skew and packet alignment

The PFB board input data format is the Xilinx serial protocol RocketIO, although there have been some customisations at a low level, made within the RocketIO CUSTOM scheme. The data can be read by any multi-gigabit-transceiver (MGT) that can use this protocol, which in practice is restricted to the Xilinx FPGA (Vertex 5 and newer). The packet format as presented to the PFB board for the second stage channeliser is detailed in Prabu et al. (2014).

A single PFB board processes 12 fibres that have come from 4 different receivers. The receivers are distributed over the 3 km diameter site and there exists the possibility of there being considerable difference in packet arrival time between inputs that are connected to different receivers. The PFB boards have an input buffer that is used to align all receiver inputs on packet number 0 (which is the 1 s tick marker see Appendix A). The buffer is of limited length (± 8 time-samples, corresponding to several hundred metres of fibre) and if the relative delay between input lines from nearby and distant receivers is greater than this buffer length, the time delay between receivers will be undetermined. We have consolidated the receivers into 4 groups with comparable distances to the central processing building and constrained these groups to have fibre runs of 1 380, 925, 515, and 270 metres. Each of the 4 groups is allocated to a single PFB (see Figure 2). This has resulted in more fibre being deployed than was strictly necessary, but has guaranteed that all inputs to a PFB board will arrive within a packet-time. The limited buffer space available can therefore be utilised to deal with variable delays induced by environmental factors and will guarantee that all PFB inputs will be aligned when presented to the correlator.

This packet alignment aligns all inputs onto the same 1 s mark but does not compensate for the *outward* clock signal delay, which traverses the same cable length, and produces a large cable delay between the receiver groups, commensurate with the differing fibre lengths. The largest cable length

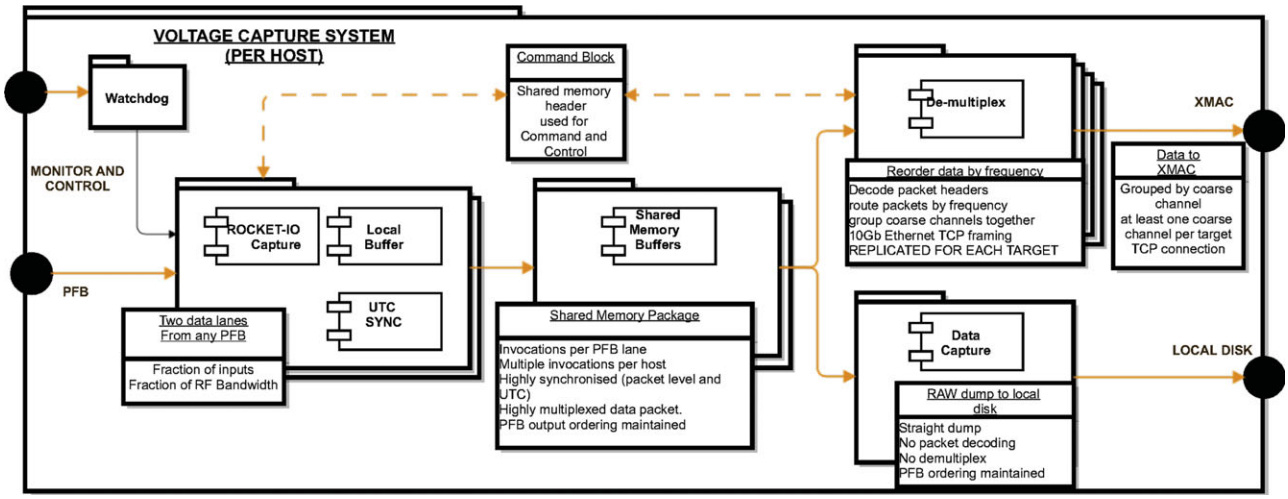

Figure 4. The decomposition of the Voltage Capture System. The diagram shows the individual elements that comprise the VCS on a single host. There are 16 VCS hosts operating independently (but synchronously) within the correlator system as a whole.

difference is 1 110 metres, being the difference in length between the shortest and longest fibres and is removed during calibration.

3.2.3. PFB output format

As the MWA PFB boards were originally purposed as the hardware F-stage of an FPGA based FX correlator, the output of the PFB was never intended to be processed independently. The output format is also the Xilinx serial protocol RocketIO, and the PFB presents eight output data lanes on two CX4 connectors. These connectors generally house four-lane XAUI as used by Inifiniband and 10 GbEt, but these are simply used as a physical layer. In actuality all of the pins on the CX4 connectors are driven as transmitters, as opposed to receive and transmit pairs, therefore only half of the available pins in the CX4 connectors were being used. We were subsequently able to change the PFB firmware to force all output data onto connector pins that are traditionally transmit pins, and leave what would normally be considered the receive pins unused. We then custom designed breakout cables that split the CX4 into 4 single lane SFP connectors (see Appendix A).

3.3. The voltage capture system

In order to capture data generated by the PFBs with off-the-shelf hardware, the custom built SFP connectors were then plugged into an off-the-shelf Xilinx based RocketIO capture cards that are housed within Linux servers. Each card can capture 2 lanes from the PFB cards, therefore 4 cards were required per PFB, and 16 cards required in total. A set of 16 machines are dedicated to this VCS, these are 16 CISCO UCS C240 M3 servers. They each house two Intel Xeon E5-2650 processors, 128 GB of RAM, and 2x1TB RAID5 disk arrays. These machines also mount the Xilinx-based FPGA board, supplied by Engineering Design Team Incorporated (EDT) that is used to capture the output from the PFB. All of the initial buffering and packet synchronisation is performed within these machines.

3.3.1. Raw packet capture

The data capture and distribution is enabled by a succession of software packages outlined in Figure 4. First, the EDT supplied capture card transfers a raw packet stream from device memory to CPU host memory. The transfer is mediated by the device driver and immediately copied to a larger shared memory buffer. At this stage the data are checked for integrity and aligned on packet boundaries to ensure efficient routing without extensive re-examination of the packet contents. Each of the 16 MWA receivers in the field obtain time synchronisation via a distributed 1 pulse per second (PPS) that is placed into the data stream in the 'second tick' field as described in Appendix A. We maintain synchronisation with this 'tick'and ensure all subsequent processing within the correlator labels the UTC second correctly. As each machine operates independently the success or failure of this synchronisation method can only be detected when packets from all the servers arrive for correlation, at which point the system is automatically resynchronised if an error is detected. No unsynchronised data are correlated.

3.3.2. The demultiplex

Data distribution in a connected element interferometer is governed by two considerations; all the inputs for the same frequency channels for each time step must be presented to the correlator; and the correlator must be able to keep up with the data rate.

As there are four PFB boards (see Figure 2), each processing one quarter of the array, each output packet contains the antenna inputs to that PFB, for a subset of the channels for a single time step. Individual data streams from each

antenna in time order need to be *cross-connected* to satisfy the requirement that a single time step for *all antennas* is presented simultaneously for cross correlation. To achieve this we utilise the fact that each PFB packet is uniquely identified by the contents of its header (see Appendix A). We route each packet (actually each block of 2 000 packets) to an endpoint based upon the contents of this header, each VCS server sends 128 contiguous, 10 kHz channels to the same physical endpoint, from all of the antennas in its allocation. This results in the same 128 frequency channels, from all antennas in the array, arriving at the same physical endpoint. This is repeated for 24 different endpoints–each receiving a different 128 channels. These endpoints are the next group of servers, the XMAC machines.

This cross-connect is facilitated by a 10 GbE switch. In actuality this permits the channels to be aggregated on any number of XMAC servers. We use 24 as it suits the frequency topology of the MWA. Regardless of topology some level of distribution is required to reduce the load on an individual XMAC server in a flexible manner, as compute limitations govern how many frequency channels can be simultaneously processed in the XMAC stage. Clark et al. (2012) demonstrates that the XMAC engine performance is limited to processing approximately 4 096 channels per NVIDIA Fermi GPU at the 32 antenna level and 1 024 channels at the 128 antenna level. This is well within the MWA computational limitations as we are required to process 3 072 channels, and have up to 48 GPU available. However, the benefit of packet switching correlators in general is clear: if performance were an issue we could simply add more GPU onto the switch to reduce the load per GPU without adding complexity. Conversely, as GPU performance improves, we can reduce the number of servers and still perform the same task.

Each of the VCS servers is required to maintain 48 concurrent TCP/IP connections, two to each endpoint, because each VCS server captures two lanes from a PFB board . This amounts to 48 open sockets that are being written to sequentially. As there are many more connections than available CPU cores individual threads, tasked to manage each connection, the operation is subject to the Linux thread scheduler, which attempts to distribute CPU resources in a *round robin* manner. Each thread on each transmit line utilises a small ring buffer to allow continuous operation despite the inevitable device contention on the single 10 GbE line. Any such contention causes the threads without access to the interface to block. This time spent in this *wait condition* allows the scheduler to redistribute resources. We also employ some context-based thread waiting on the receiving side of the sockets to help the scheduler in its decision making and to even the load across the data capture threads. We have found that this scheme results in a thread scheduling pattern that is remarkably fair and equitable in the allocation of CPU resources and provided that a reasonably-sized ring buffer is maintained for each data line, all the data are transmitted without loss.

3.3.3. VCS data output

To summarise, we are switching packets as they are received and they are routed based upon the contents of their header. Therefore any VCS server can process any PFB lane output. Each server holds a single PCI-based capture card, and captures two PFB data lanes. The data from each lane is grouped in time contiguous blocks of 2 000 packets (50 ms worth of samples for 16 adjacent 10 kHz frequency channels), from all 32 of the antennas connected to that PFB. The packet header provides sufficient information to uniquely identify the PFB, channel group, and time block it is, and the packet block is routed on that basis. Further information is given in Appendix B. A static routing table is used to ensure that each XMAC server receives a contiguous block of frequency channels, the precise number of which is flexible and determined by the routing table, but it is not alterable at runtime.[2]

3.3.4. Monitoring, command, control and the watchdog

Monitor and control functionality within the correlator is mediated by a watchdog process that runs independently on each server. The watchdog launches each process, monitors activity, and checks error conditions. It can restart the system, if synchronisation is lost and mediates all start/stop/idle functionality as dictated by the observing schedule. All interprocess command and control, for example propagation of HALT instructions to child processes and time synchronisation information, is maintained in a small block of shared memory that holds a set of key-value pairs in plain text that can be interrogated or set by third-party tools to control the data capture and check status.

3.3.5. Voltage recording

The VCS system also has the capability to record the complete output of the PFB to local disk. The properties and capabilities of the VCS system will be detailed in a companion paper (Tremblay et al. 2015).

3.4. The cross-multiply and accumulate (XMAC)

The GPU application for performing the XMAC is running on 24 NVIDIA M2070 Fermi GPU housed in 24 IBM iDataplex servers. These machines are housed in racks adjacent to the VCS servers and connected via a 10 GbE network. The package diagram detailing this aspect of the correlator is presented in Figure 5 and the packet interface and data format is described in Appendix C. The cross-multiply servers receive data from the 32 open sockets and internally align and unpack the data into a form suitable for the GPU. The GPU kernel is launched to process the data allocation and internally integrates over a user defined length of time. There are other operations possible, such as incoherent beam forming and the raw dump of data products (or input voltages).

[2] In order to achieve optimal loop unrolling and compile-time evaluation of conditional statements, xGPU requires that the number of channels, number of stations, and minimum integration time are compile-time parameters.

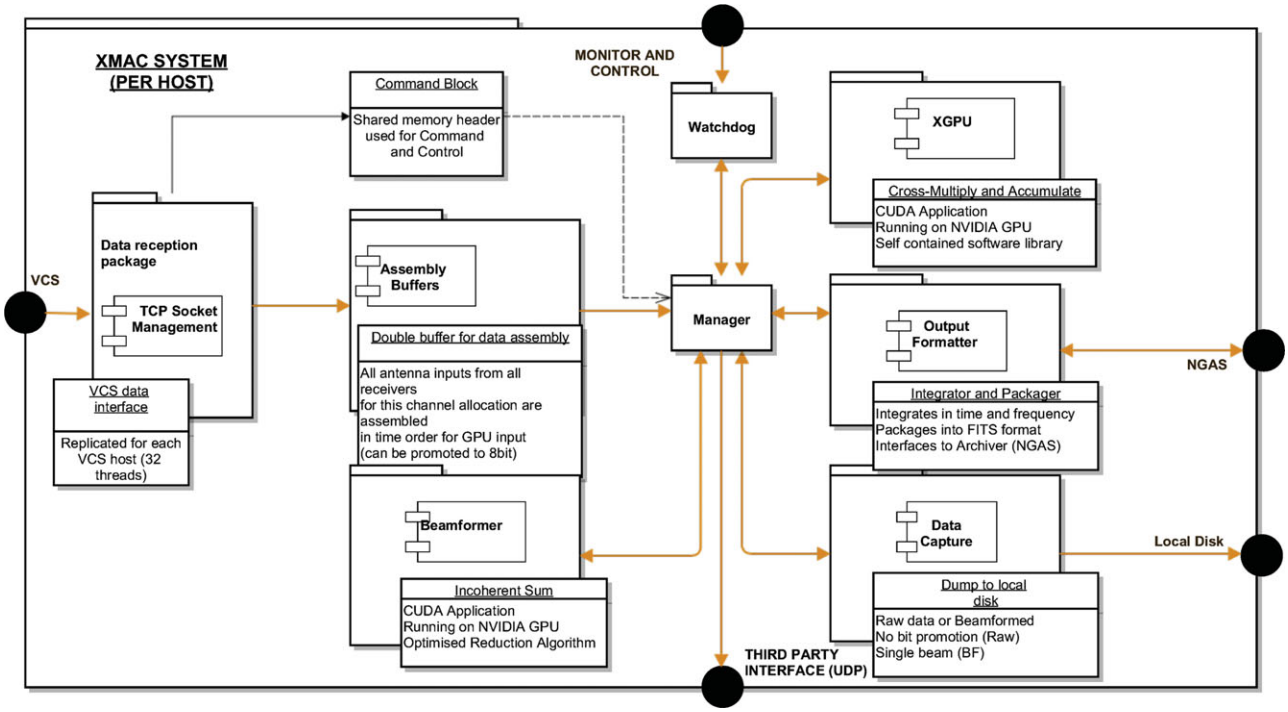

Figure 5. The decomposition of the cross-multiply and accumulate operation running on each of 24, GPU enabled IBM iDataplex servers.

These modes will be expanded upon during the development of other processing pipelines and will be detailed in companion papers when the modes are available to the community.

3.4.1. Input data reception, and management

A hierarchical ring buffer arrangement is employed to manage the data flow. Each TCP connection is managed by a thread that receives a 2 000 packet block and places it in a shared memory buffer. Exactly where in a shared memory block each of the 2 000 packets is placed is a function of the time/frequency/antenna block that it represents. A small *corner-turn*, or transpose, is required as the PFB mixes the order of time and frequency. The final step in this process is the promotion of the sample from 4 to 8 bits, which is required to enable a machine instruction that performs rapid promotion from 8 to 32 bits within the GPU. This promotion is not performed if the data are just being output to local disk after reordering.

Each of the 32 threads is filling a shared memory block that represents 1 s of GPU input data, and a management thread is periodically checking this block to determine whether it is full. Safeguards are in place that prevent a thread getting ahead of its colleagues or overwriting a block. The threads wait on a condition variable when they have finished their allocation, so if a data line is not getting sufficient CPU resources it will soon be the only thread *not* waiting and is guaranteed to complete. This mechanism helps the Linux thread scheduler to divide the resources equitably. The use of TCP allows the receiver to actually prioritise which threads on the send side of the connection are blocked to facilitate an even distribution of CPU resources.

Once the management thread judges the GPU input block to be full, it releases it, launches the GPU kernel and frees all the waiting threads to fill the next buffer while the GPU is running.

3.4.2. xGPU - correlation on a GPU

It is possible to address the resources of a GPU in a number of ways, utilising different application programming interfaces (API), such as OpenCL, CUDA, and openGL. The MWA correlator uses the xGPU library as is described in detail in Clark et al. (2012). The xGPU library is a CUDA application which is specific to GPUs built by the NVIDIA corporation. There are many references in the literature to the CUDA programming model and examples of its use (Sanders & Kandrot 2010).

The performance improvement seen when porting an application to a GPU is in general due to the large aggregate FLOPS and memory bandwidth rates, but to realise this performance requires effective use of the large number of concurrent threads of execution that can be supported by the architecture. This massively parallel architecture is permitted by the large number of processing cores on modern GPUs. The TESLA M2070 are examples of NVIDIA Fermi architecture and have 448 cores, grouped into 14 *streaming multiprocessors* or SM, each with 32 cores. The allocation of resources follows the following model: threads are grouped into a *thread block* and a thread block is assigned to an SM; there can be more thread blocks than SM, but only one block will be executing at a time on any SM; the threads within each block are then divided into groups of 32, (one for each core of the associated SM), and this subdivision called a *warp*;

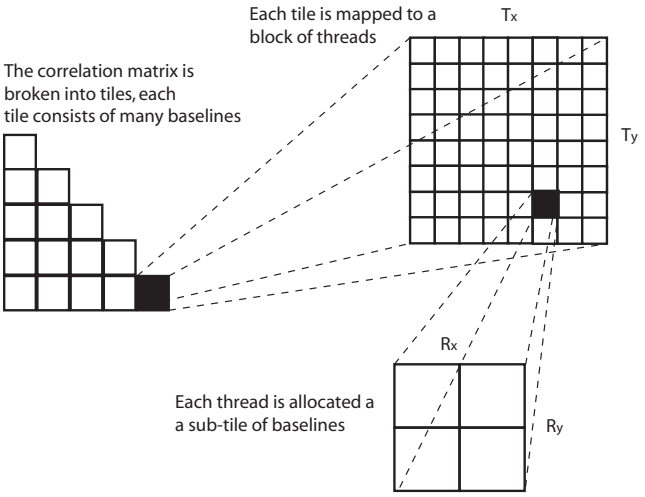

Figure 6. The arithmetic intensity of the correlation operation on the GPU is increased by tiling the correlation matrix. Threads are assigned groups of baselines instead of a single baseline.

execution is serialised within a warp with the same instruction being performed by all 32 threads, but on different data elements in the *single instruction multiple data* paradigm.

GPGPU application performance is often limited by memory access bandwidth. A good predictor of algorithm performance is therefore *arithmetic intensity*, or the number of floating point operations per byte transferred. A complex multiply and add for two dual polarisation antennas at 32 bit precision requires 32 bytes of input, 32 FLOPS (16 multiply-adds), and 64 bytes of output. The arithmetic intensity of this is $32/96 = 0.33$. Compare this to the Fermi C2070 architecture which has a peak single-precision performance of 1030 GFLOPS and memory bandwidth of 144 GB/s; the ratio of which (7.2) tells us that the performance of the algorithm will be completely dominated by memory traffic. The high performance kernel developed by Clark et al. (2011) increases the arithmetic intensity in two ways. Firstly, the output memory traffic can be reduced by integrating the products for a time, I, at the register level. Secondly, instead of considering a single baseline, groups of $(m \times n)$ baselines denoted tiles (see Figure 6), are constructed cooperatively by a block of threads. In order to fill an $m \times n$ region of the final correlation matrix, only two vectors of antennas need to be transferred from host memory. A thread block loads two vectors of antenna samples (of length n and m) and forms nm baselines. These two steps then alter the arithmetic intensity calculation to

$$\text{Arithmetic Intensity} = \frac{32mnI}{16(m+n)I + 64mn}, \qquad (4)$$

which implies that the arithmetic intensity can be made arbitrarily large by increasing the tile size until the number of available registers is exhausted. In practice, a balance must be struck between the available resources and the arithmetic intensity and this balance is achieved in the xGPU implementation by *multi-level tiling* and we direct the reader to Clark et al. (2012) for a complete discussion.

The MWA correlator is not a delay tracking correlator. Therefore, we do not have to adjust the correlator inputs for whole, or partial sample delays. The correlator always provides correlation output products phased to zenith. The system parameters (integration time, baseline length, channel width) have all been chosen with this in mind and the long operating wavelengths result in the system showing only minimal (1%) decorrelation even far from zenith for typical baseline lengths, integration times and channel widths (see Figure 7). Typical observing resolutions have been between 0.5 and 2 s in time and 40 kHz and 10 kHz in frequency.

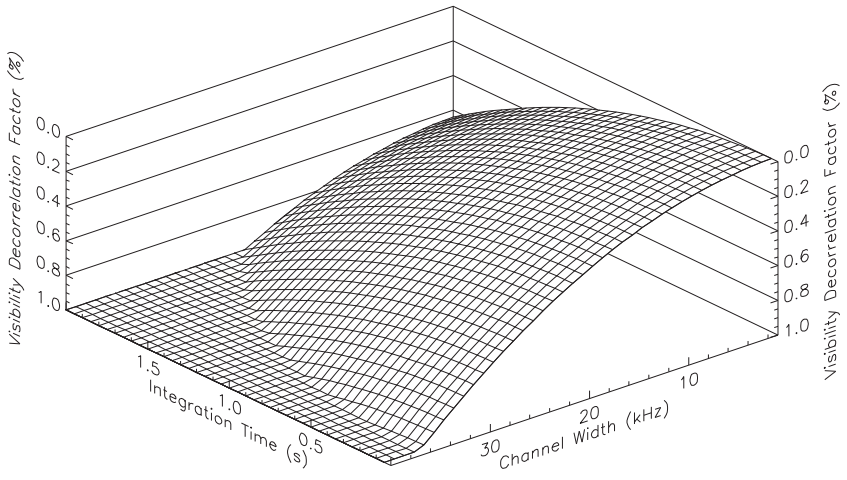

Figure 7. The combined reduction in coherence due to time and bandwidth smearing 45° from zenith on a 1km baseline, at an observing frequency of 200 MHz as a function of channel width and integration time, demonstrating that even this far from zenith the decorrelation is generally less the 1%. The longest MWA baselines are 3 km, and these baselines show closer to 5% decorrelation for the same observing parameters. Decorrelation factors above 1% are not plotted for clarity.

3.4.3. Output data format

The correlation products are accumulated on the GPU device and then copied asynchronously back to host memory. The length of accumulation is chosen by the operator. Each accumulation block is topped with a FITS header (Wells, Greisen, & Harten 1981) and transferred to an archiving system where the products are concatenated into larger FITS files with multiple extensions, wherein each integration is a FITS extension (see Appendix D). The archiving system is an instantiation of the Next Generation Archiving System (NGAS) (Wicenec & Knudstrup 2007; Wu et al. 2013) and was originally developed to store data from the European Southern Observatory (ESO) telescopes. The heart of the system is a database that links the output products with the metadata describing the observation and that manages the safe transportation and archiving of data. In the MWA operating paradigm all visibilities are stored, which at the native resolution is 33 024 visibilities and 3 072 channels or approximately 800 MBytes per time resolution unit (more than 6 Gb s^{-1} for 1 s integrations). Due to the short baselines and long wavelengths used by the array it is reasonable to integrate by a factor of two in each dimension (of time and frequency, see Figure 7 for the parameter space), but even this reduced rate amounts to 17 TBytes day^{-1} at 100% duty cycle. The large data volume precludes handling by humans and the archiving scheme is entirely automated. NGAS provides tools to access the data remotely and provides the link between the data products to the observatory SQL database. As discussed in Tingay et al. (2013a), the data are taken at the Murchison Radio Observatory in remote Western Australia (WA) and transferred to the Pawsey HPC Centre for SKA Science in Perth, where 15 PB of storage is allocated to the MWA over its five-year lifetime. Subsequently the archive database is mirrored by the NGAS system to other locations around the world: MIT in the USA; The Victoria University of Wellington, NZ; and the Raman Research Institute in India. These remote users are then able to access their data locally.

4. INSTRUMENT VERIFICATION AND COMMISSIONING

After initial instrument commissioning with simple test vectors and noise inputs the telescope entered a commissioning phase in September 2012. The commissioning team initially consisted of 19 scientists from 10 institutions across 3 countries and was led by Curtin University. This commissioning phase was successfully concluded in July 2013 giving way to "Early Operations" and the MWA is now fully operational.

4.1. Verification

The development of the MWA correlator proceeded separately to the rest of the MWA systems and the correlator GPU based elements were verified against other software correlator tools. The MWA correlator is comparatively simple, it does not track delays, performs the correlation at 32-bit floating point precision, and does not have to perform any Fourier transformations. The only verification that was required was to ensure that the XMAC was performed to sufficient precision and matched results generated by a simple CPU implementation.

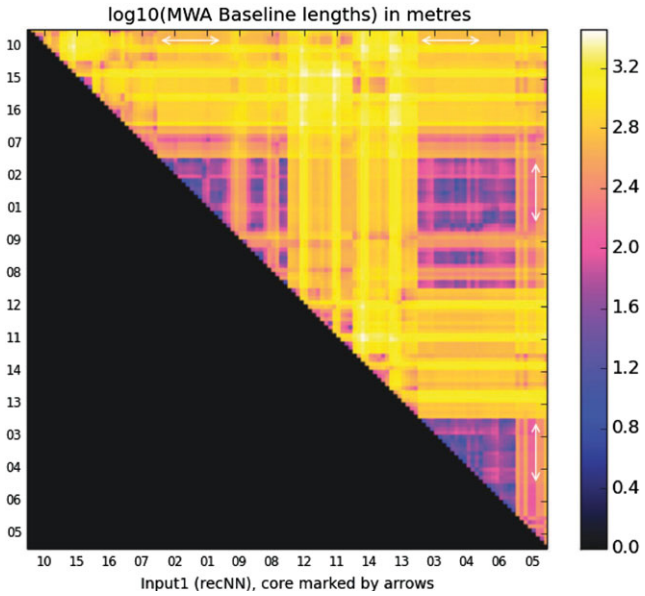

Figure 8. A correlation matrix showing the baseline length distribution. The antennas are grouped into receivers, each servicing 16 antennas and the antenna layout displays a pronounced centrally dense core. The layout is detailed in Tingay et al. (2013a). The colours indicate baseline length and the core regions are also indicated by arrows within the figure.

Subsequent to ensuring that the actual XMAC operation was being performed correctly, the subsequent testing and integration was concerned with ensuring that the signal and path was maintained. This was a relatively complex operation due to the requirement that the correlator interface with legacy hardware systems. The details of these interfaces are presented in the Appendices. In Figure 8, a correlation matrix is shown, the colours representing the length of the baseline associated with that antenna pair. The following Figure 9 shows the response of the baselines when the telescope is pointed at the radio galaxy Centaurus A, demonstrating the response of the different baseline lengths of the interferometer to structure in the source. Images of this object obtained with the MWA can be found in McKinley et al. 2013. The arrows on Figure 8 indicate those baselines associated with the densely-packed core (Tingay et al. 2013a).

4.2. Verification experiments

The MWA Array and correlator is also the verification platform for the Aperture Array Verification System (AAVS) within the SKA reconstruction activities pursued by the Low Frequency Aperture Array (LFAA) consortium(bij de Vaate

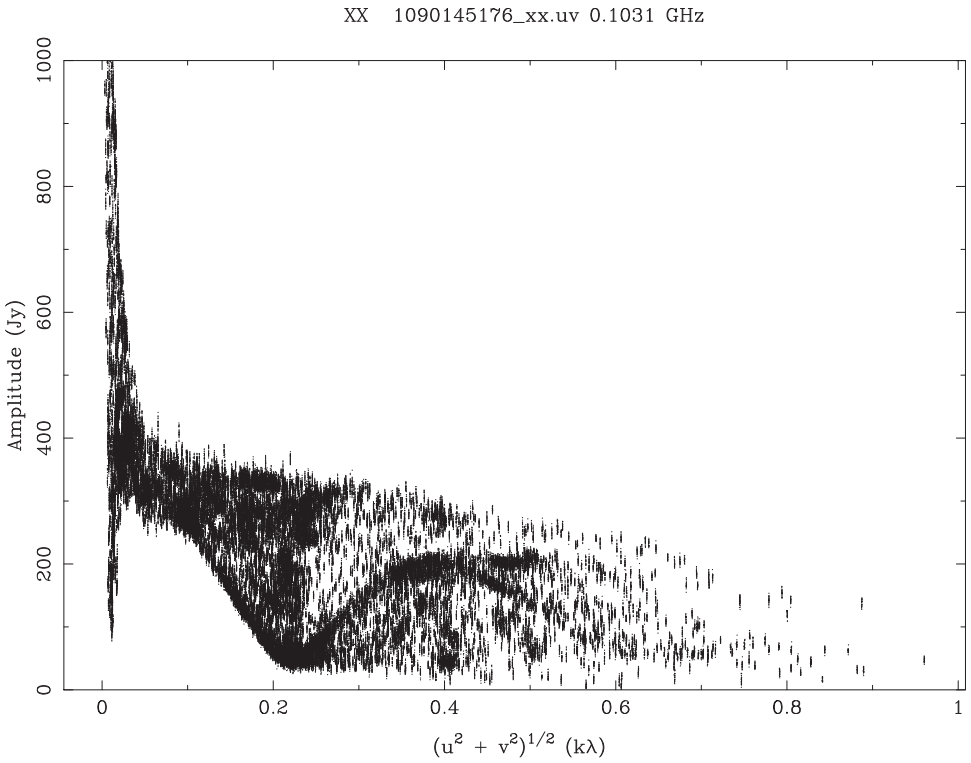

Figure 9. Visibility amplitude vs baseline length for a 2 min snapshot pointing at the giant radio galaxy Centaurus A. Centre frequency 120 MHz. Only 1 polarisation is shown for clarity. The source has structure on a wide range of scales and does not dominate the visibilities as a point source.

et al. 2011). This group have recently performed a successful verification experiment, using both the MWA and the AAVS telescopes. Using an interferometric observation of the radio galaxy Hydra A taken with the MWA system, they have obtained a measurement of antenna sensitivity (A/T) and compared this with a full-wave electromagnetic simulation (FEKO[3]). The measurements and simulation show good agreement at all frequencies within the MWA observing band and are presented in detail in Colegate et al. (2014).

4.3. Commissioning science

In order to verify the instrumental performance of the array, correlator and archive as a whole, the MWA Science Commissioning Team have performed a 4 300 deg^2 survey, and have published a catalogue of flux densities and spectral indices from 14 121 compact sources. The survey covered approximately 21 h $<$ Right Ascension $<$ 8 h, $-55^\circ <$ Declination $< -10^\circ$ over three frequency bands centred on 119, 150, and 180 MHz. This survey will be detailed in Hurley-Walker et al. (2014). Data taken during and shortly after the commissioning phase of the instrument has also been used to demonstrate the tracking of space debris with the MWA (Tingay et al. 2013b), novel imaging and deconvolution schemes (Offringa et al. 2014), and to present multi-frequency observations of the Fornax radio galaxy (McKinley et al. 2013) and the galaxy cluster A3667 (Hindson et al. 2014).

5. THE FUTURE OF THE CORRELATOR

5.1. Upgrade path

We do not fully utilise the GPU in the MWA correlator and we estimate that the X-Stage of the correlator could currently support a factor of 2 increase in array size (to 256 dual polarisation elements). As discussed in Tingay et al. (2013a) the operational life-span of the MWA is intended to be approximately five years. The GPUs (NVIDIA M2070s) the MWA correlator are from the Fermi family of NVIDIA GPU and are already almost obsolete. One huge advantage of off-the-shelf signal processing solutions is that we can easily benefit from improvements in technology. We could swap out the GPU in the current MWA cluster, replace them with cards from the Kepler series (K20X) with no code alteration, and would benefit from a factor of 2.5 increase in performance (and a threefold improvement in power efficiency) as not only are the number of FLOPS provided by the GPUs increasing, but also their efficiency (in FLOPS/Watt) is improving rapidly. This would permit the MWA to be scaled to 512 elements. However, not all of our problem is with the correlation step and increasing the array to 512 elements would probably

[3] www.feko.info

require an upgrade to the networking capability of the correlator.

5.2. SKA activities

The MRO is the proposed site of the low frequency portion of the SKA. Curtin University is a recipient of grants from the Australian government to support the design and pre-construction effort associated with the construction and verification of the LFAA and software correlation systems for the SKA project. The MWA correlator will therefore be used as a verification platform within the prototype for the LFAA, known as the AAVS. One of the benefits of a software based signal processing system is flexibility. We have been able to easily incorporate AAVS antennas into the MWA processing chain to facilitate these verification activities, which have been of benefit to both LFAA and the MWA (Colegate et al. 2014). Furthermore, we will be developing and deploying SKA software correlator technologies throughout the pre-construction phase of the SKA using the MWA correlator for both testing and verification.

6. SUMMARY

This paper outlines the structure, interfaces, operations and data formats of the MWA Hybrid FPGA/GPU correlator. This system combines off-the-shelf computer hardware with bespoke digital electronics to provide a flexible and extensible correlator solution for a next generation radio telescope. We have outlined the various stages in the correlator signal path and detailed the form of all internal and external interfaces.

ACKNOWLEDGEMENTS

We acknowledge the Wajarri Yamatji people as the traditional owners of the Observatory site. Initial correlator development was enabled by equipment supplied by an IBM Shared University Research Grant (VUW & Curtin University), and by an Internal Research and Development Grant from the Smithsonian Astrophysical Observatory.

Support for the MWA comes from the U.S. National Science Foundation (grants AST CAREER-0847753, AST-0457585, AST-0908884 and PHY-0835713), the Australian Research Council (LIEF grants LE0775621 and LE0882938), the U.S. Air Force Office of Scientific Research (grant FA9550-0510247), the Centre for All-sky Astrophysics (an Australian Research Council Centre of Excellence funded by grant CE110001020), New Zealand Ministry of Economic Development (grant MED-E1799), an IBM Shared University Research Grant (via VUW & Curtin), the Smithsonian Astrophysical Observatory, the MIT School of Science, the Raman Research Institute, the Australian National University, the Victoria University of Wellington, the Australian Federal government via the National Collaborative Research Infrastructure Strategy, Education Investment Fund and the Australia India Strategic Research Fund and Astronomy Australia Limited, under contract to Curtin University, the iVEC Petabyte Data Store, the Initiative in Innovative Computing and NVIDIA sponsored CUDA Center for Excellence at Harvard, and the International Centre for Radio Astronomy Research, a Joint Venture of Curtin University and The University of WA, funded by the Western Australian State government.

PASA, 32, e006 (2015)
doi:10.1017/pasa.2015.5

REFERENCES

bij de Vaate, J. G., et al. 2011, General Assembly and Scientific Symposium, 2011 XXXth URSI, 1
Bowman, J. D., et al. 2013, PASA, 30, 31
Bunton, J. D. 2003, ALMA Memo Series, 1
Clark, M. A., La Plante, P. C., & Greenhill, L. J. 2012, International Journal of High Performance Computing Applications, 27(2), 178
Colegate, T. M., et al. 2014, in Antenna Measurements & Applications (CAMA), 2014 IEEE Conference on (IEEE), 1–4
Crochiere, R. E., & Rabiner, L. R. 1983, Multirate Digital Signal Processing (Prentice-Hall Processing Series), 1st ed. (Englewood Cliffs, NJ: Prentice-Hall)
de Souza, L., Bunton, J. D., Campbell-Wilson, D., Cappallo, R. J., & Kincaid, B. 2007, in FPL, ed. K. Bertels, W. A. Najjar, A. J. van Genderen, & S. Vassiliadis, IEEE Conference Publications (Boston, MA: IEEE), 62–67
Hindson, L., et al. 2014, ArXiv e-prints
Hurley-Walker, N. 2014, PASA, 31, 45
Kocz, J., et al. 2014, JAI, 3, 50002
Lonsdale, C. J., et al. 2009, IEEE, 97, 1497
McKinley, B., et al. 2013, MNRAS, 436, 1286
Offringa, A. R., et al. 2014, ArXiv e-prints
Parsons, A. R., et al. 2010, AJ, 139, 1468
Prabu, T., Srivani, K., & Anish, R. 2014, ExA, http://arxiv.org/abs/1502.05015
Sanders, J., & Kandrot, E. 2010, CUDA by Example, An Introduction to General-Purpose GPU Programming (Boston, MA: Addison-Wesley Professional)
Thompson, A. R., Moran, J. M., & Swenson, Jr., G. W. 2001, Interferometry and Synthesis in Radio Astronomy (2nd edn.; John Wiley and Sons)
Tingay, S. J., et al. 2013a, PASA, 30, 7
Tingay, S. J., et al. 2013b, AJ, 146, 103
Tremblay, S., et al. 2015, PASA, accepted
van Haarlem, M. P., et al. 2013, A&A, 556, A2
Wayth, R. B., Greenhill, L. J., & Briggs, F. H. 2009, PASA, 121, 857
Wells, D. C., Greisen, E. W., & Harten, R. H. 1981, A&AS, 44, 363
Wicenec, A., & Knudstrup, J. 2007, Msngr, 129, 27
Wu, C., Wicenec, A., Pallot, D., & Checcucci, A. 2013, ExA, 36, 679

APPENDIX

A. MWA F-STAGE (PFB) TO VOLTAGE CAPTURE SYSTEM (VCS)

The contents of this Appendix are extended from the original MWA ICD for the PFB written by R. Cappallo. We have incorporated the changes made to the physical mappings, electrical connections and header contents to enable capture of the PFB output by general purpose computers.

A.1. Physical

- PFB presents data on two CX4 connectors. ALL pins are wired with transmit drivers, there are no receivers on any pins.

A.2. Interface details

- Two custom made CX4 to SFP breakout cables. This cable breaks a CX4 connector into 4 1X transmit and receive pairs on individual SFP terminated cables.
- Data flows only in a single direction (simplex) from PFB to CB. Each (normally Tx/Rx) pair of differential pairs is wired as two one-way data paths (Tx /Tx). We have distributed these one-way data paths to occupy every other CX4 pin. Therefore, all cables are wired correctly as Tx/Rx but the Rx channels are not utilised.
- The serial protocol to be used a custom protocol as described below. The interface to the Xilinx Rocket I/O MGTs is GT11_CUSTOM, which is a lean protocol. The word width will be 16 bits.
- The mean data rate per 1X cable will be 2.01216 Gb s^{-1}, carried on a high speed serial channel of burst rate of 2.08 Gb s^{-1} formatted, or 2.6 Gb s^{-1} including the 10B $8B^{-1}$ encoding overhead.

A.3. Data format

- Each data stream carries all of the antenna information for 384 fine frequency channels. In the entire 32T correlation system there are 8 such data streams going from the PFB to the CB, each carrying the channels making up a 3.84 MHz segment of the processed spectrum.
- Each sample in the stream is an 8-bit (4R + 4I) complex number, 2 s complement with 8 denoting invalid data, representing the voltage in a 10 kHz fine frequency channel.
- Data are packetised and put in a strictly defined sequence, in f1t[f2a] (frequency-time-frequency-antenna) order. The most rapidly varying index, a, is the antenna number (0, 16, 32, 48, 1, 17, 33, 49, 2, 18, 15, 31, 47, 63), followed by f2, the fine frequency channel (n..n + 3), then time (0..499), and most slowly the fine frequency channel group index (0, 4, 8, 12, 128, 132, 136, 140, 2 944, 2 948, 2 952, 2 956 for fibre 0; increment by 16 n for fibre n). The brackets indicate the portion of the stream that is contained within a single packet. The antenna ordering is nonintuitive as for historical reasons each PFB combines its inputs into an order originally considered conducive to its interface to a full mesh backplane and a hardware FPGA based correlator.
- A single packet consists of all antenna data for a group of four fine channels for a single time point. Thus, there are 500 sequential time packets for each of the 96 frequency channel groups. Over the interval of 50 ms there are 48 000 (= 96channel groups × 500 time samples) packets sent per data stream, so there are 960 000 packets s^{-1}.
- Using a packet size of 264 bytes allows a 16-bit packet header (0 × 0800), two 16-bit words carrying the second of time tick and packet counter, 256 bytes of antenna data, finally followed by a 16-bit checksum (see Table 1).
- The header word is a 16-bit field with value 0 × 0800, used to denote the first word in a packet. Note that it cannot be confused with the data words, since the value 8 is not a legal voltage sample code.
- The second tick in word two is in bit 0, and has the value 0 at all times, except for the first packet of a new second, when it is 1.
- The leading bits of word two, and all of word three contains the following counter values that can be decoded to determine the precise input and channel that a given packet belongs to.
  - *sec_tick* (1 bit): Is this the first packet for this data lane, for this one second of data? 0 = No, 1 = Yes. Only found on the very first packet of that second. If set, then the mgt_bank, mgt_channel, mgt_group and mgt_frame will always be 0. The preceding 3 bits are unused.
  - *pfb_id* (2 bits): Which physical PFB board generated this stream (0–3). This defines the set of receivers and tiles the data refers to. A lane's pfb_id will remain constant unless physically shifted to a different PFB via a cable swap.
  - *mgt_id* (3 bits): Which 1/8 of coarse channels this RocketIO lane contains. Should be masked with 0× 7, not 0× f. Contains [0–7] inclusive. A lane's mgt_id will remain constant unless physically shifted to a different port on the PFB via a cable swap.

  Within an individual data lane, the data packets cycle in the following order, listed from slowest to fastest.
  - *mgt_bank* (5 bits): Which one of the twenty 50 ms time banks in the current second is this one? [0–19] [0–0 × 13].
  - *mgt_channel* (5 bits): Which coarse channel [0–23] [0–0 × 17] does the packet relate to?
  - *mgt_group* (2 bits): Which 40 KHz wide packet of the contiguous 160 KHz does this packet contain the 4 × 10 KHz samples for? [0–3]
  - *mgt_frame* (5 bits): Which time stamp within a 50 ms block this packet is. [0–499] or [0–0 × 1F3]. Cycles fastest. Loops back to 0 after 0 × 1F3. There are 20 complete cycles in a second.
- The last word of the packet is a 16-bit checksum, formed by taking the bitwise XOR of all 128 (16 bit) words of antenna samples in the packet.

B. VCS TO XMAC SERVER

B.1. Physical

- Data presented on either fibre optic or direct attach copper cables terminated with SFP pluggable transceivers. Note that all optical transceivers used in the CISCO UCS servers must be supplied by CISCO, the firmware within the 10 GbE cards requires it.
- A single 10 GbE interface is sufficient to handle the data rate.

B.2. Interface details

- Ethernet IEEE 802.3 frames, with TCP/IPv4.
- The packet format is precisely as described in the PFB to VCS interface.
- The communication is from the VCS to the XMAC is unidirectional and mediated by a switch.
- Each XMAC requires a data packet from all VCS machines. This requires 32 open connections mediated by TCP.
- Data from all sources is stripped of header information and assembled into a common 1 s buffer which requires 20 blocks of 2 000 packets from each of the 32 connections to be assembled.

- As the packets for four PFB are being combined, the antenna ordering is also being concatenated. The order within each packet is maintained with the 64 inputs concatenated into a final block of 256 inputs.

C. XMAC TO GPU INTERFACE

C.1. Physical

The interface is internal to the CPU host, being a section of memory shared between the GPU device driver and the host.

C.2. Interface details

C.2.1. GPU input

The data ordering in the assembly buffer for input buffers, running from slowest to fastest index is time(t), channel(f), station(s), polarisation(p), complexity(i). The data is an 8-bit integer, promoted from 4-bit twos-complement sample via a lookup table. The correlator is a dual polarisation correlator as implemented, but the input station/polarisation ordering is not in a convenient order as we have concatenated 4 PFB outputs during the assembly of this buffer. Two options presented themselves. One that we re-order the input stations and polarisation to undo the partial corner turn. The disadvantage begin that each input packet would need to be broken open and the stations reordered–which would have been extremely compute intensive. Or to correlate *without* changing the order and remapping the output products to the desired order. This is preferable as although there are now N^2 products instead of N stations, the products are integrated in time and can be reordered in post-processing.

C.2.2. GPU output

The data are formatted as 32-bit complex floating point numbers. An individual visibility set has the order (from slowest index to fastest index): channel (f), station (s1), station (s2), polarisation (p1), polarisation (p2), complexity (i). The full cross-correlation matrix is Hermitian so only the lower, or upper, triangular elements need to be calculated, along with the on-diagonal elements which are the auto-correlation products. Therefore, only a packed triangular matrix is actually transferred from the GPU to the host. The GPU processing kernel is agnostic to the ordering of the antenna inputs and the PFB output order is maintained. Therefore, the correlation products cannot be simply processed without a remapping.

The remapping is performed by a utility that has been supplied by the builders to the data analysis teams that simply reorders the data products into one that would be expected if no PFB reordering had taken place. It is important to keep track of which products have been generated but the XMAC as conjugates of the desired products, and which need conjugating. It should further be noted that this correction is performed to the order as presented *to* the PFB and may not be the order as expected by the ordering of the receiver inputs. Care should be taken in mapping antennas on the ground as the order of the receiver processing within a PFB is not only function of the physical wiring but the firmware mapping. We have carefully determined this mapping and the antenna mapping tables are held within the same database as the observing metadata.

For historical reasons, the files are transferred into an intermediate file format that was originally used during instrument commissioning before subsequent transformation to UVFITS files (or measurement sets) by the research teams.

D. XMAC SERVER TO NGAS (NEXT GENERATION ARCHIVING SYSTEM)

D.1. Physical

The interface is internal to the CPU host, being a section of memory shared between the XMAC application and an NGAS application running on the same host.

D.2. Interface details

The data are handed over as a packed triangular matrix with the same ordering as was produced by the GPU. It has a FITS header added and is padded to the correct size as expected by a FITS extension. The FITS header includes a time tag.

Once the buffer is presented to the NGAS system, it is buffered locally and presented to a central archive server, subsequently it is added to a database and mirrored to multiple sites across the world.