Networked Answer to “Life, The Universe, and Everything”

Giles Babich, Keith Bengston, Andrew Bolin, John Bunton, Yuqing Chen,
Grant Hampson, David Humphrey, and Guillaume Jourjon
CSIRO-Space&Astronomy
first.last@csiro.au

ABSTRACT

In the last few years, Input/Output (I/O) bandwidth limitation of legacy computer architectures forced us to reconsider where and how to store and compute data across a large range of applications. This shift has been made possible with the concurrent development of both smartNICs and programmable switches with a common programming language (P4), and the advent of attached High Bandwidth Memory within smartNICs/FPGAs. Recently, proposals to use this kind of technology have emerged to tackle computer science related issues such as fast consensus algorithm in the network, network accelerated key-value stores, machine learning, or data-center data aggregation. In this paper, we introduce a novel architecture that leverages these advancements to potentially accelerate and improve the processing of radio-astronomy Digital Signal Processing (DSP), such as correlators or beamformers, at unprecedented continuous rates in what we have called the “Atomic COTS” design. We give an overview of this new type of architecture to accelerate digital signal processing, leveraging programmable switches and HBM capable FPGAs. We also discuss how to handle radio astronomy data streams to pre-process this stream of data for astronomy science products such as pulsar timing and search. Finally, we illustrate, using a proof of concept, how we can process emulated data from the *Square Kilometer Array* (*SKA*) project to time pulsars.

CCS CONCEPTS

• **Networks** → **Programmable networks**; • **Hardware** → **Digital signal processing**; *High-speed input / output*.

KEYWORDS

Digital Signal Processing, In-network computing, Astronomy

ACM Reference Format:
Giles Babich, Keith Bengston, Andrew Bolin, John Bunton, Yuqing Chen,, Grant Hampson, David Humphrey, and Guillaume Jourjon. 2021. Networked Answer to “Life, The Universe, and Everything”. In *Symposium on Architectures for Networking and Communications Systems (ANCS’21), December 13–16, 2021, Lafayette, IN, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3493425.3502770

1 INTRODUCTION

In the 20^{th} century, high-end radio-telescope correlators were built with custom ASICs (e.g., VLA, Westerbork, Australia Telescope). As we entered the 21^{st} century FPGAs came to the fore (e.g., eVLA, CABB), but due to the huge data flows they needed massive dedicated data transport systems such as optical backplane connectivity between boards [14, 22]. In the context of the future *Square Kilometer Array* low frequency project (*SKA*.Low), it is envisioned that the initial implementation of the telescope will contain more than 131k antennas. These antennas would produce continuously in the excess of 6Tb/s of raw data signal across 364 channels. This amount of data is then processed by correlators and beamformers to produce up to 8Tb/s of results, with a processing load of about two Peta multiply-accumulate operations per second (PMACs).

The scale of such a telescope makes it difficult to build a custom data transport system as has been done in the past. Recently the Collaboration for Astronomy Signal Processing and Electronics Research (CASPER) group [18] took a different path and used network switches for data transport but this imposed a significant extra cost, requiring considerable time, effort and interaction with the switch vendor to fully commission the system. This issue of adapting legacy networking elements tailored to a specific need has long been identified in the networking community.

As a result of these issues there have been technology advances. The past five years has seen the appearance of a new generation of Ethernet switches (In-Network Processors) where the data plane is fully program-mable using languages such as P4 [1] and OpenFlow [12]. A second development has been the combination of FPGAs and High Bandwidth (attached) Memory (HBM). HBM memory has an I/0 bandwidth that is more than 30 times that of a 100GbE link [21].

In the context of a radio telescope, this allows the full data from antennas to be buffered to memory multiple times. Each stage of processing can then operate independently of any other, effectively as independent asynchronous subroutines (whereas all early correlator designs were synchronous). The multiple “subroutines” can implement different processing

ANCS’21, December 13–16, 2021, Layfette, IN, USA

ACM ISBN 978-1-4503-9168-9/21/12. . . $15.00
https://doi.org/10.1145/3493425.3502770

blocks in the end-to-end signal processing chain. These FPGAs are now available as COTS hardware ready to plug into a standard server and have 100GbE ports that can interface with in-network processors.

The combination of these two technology advances allows a new approach to radio-astronomy correlators and beamformers - which we have called "Atomic COTS". This is a design that uses COTS In-Network Processors to aggregate data to COTS FPGA cards such that each card receives part of the total bandwidth for all receiving elements (antennas). The FPGA then processes the data to implement a correlator or beamformer for the bandwidth slice that it has received, independent of any other FPGA (an Atomic operation). The resulting astronomy data products can be routed back through the In-Network Processors to the next stage of processing such as imaging with visibilities and searches for Fast Radio Bursts (FRBs) using tied array beams. An Atomic COTS architecture will enable astronomy backends to upgrade and deploy incrementally. The *Atomic COTS* design principle for beamforming and correlating radio astronomy data is currently under consideration for several radio telescope proposals including the *Square Kilometer Array* (*SKA*).

The main contributions of the paper are as follows:

- General design principle for beamforming and correlating radio astronomy data streams for later use in astronomy.
- Introduction of new programming switch modus operandi tailored to radio astronomy.
- Overview of staged batch-to-completion correlation/beamforming primitives for data processing from hundreds of thousands of antennas.
- How the *Atomic COTS* approach integrates with existing radio-telescope facilities without need to reconfigure either up-stream (bespoke antenna and receiver) or downstream (science data processing).
- An example of *Atomic COTS* capability for beamforming data and recovery of pulsar signals buried 10dB beneath noise using emulated data from the *SKA* project.

Roadmap. We start by presenting the background and motivations (§2). Then we present the design principles of an *Atomic COTS* DSP (§3). Finally, we demonstrate through emulation how *Atomic COTS* could be used to time pulsars (§4), before discussing a few remaining open questions (§5).

2 BACKGROUND

Radio telescopes can be crudely divided into three blocks as illustrated in Figure 1. The first two stages of a radio telescope include custom antennas, radio receivers and digitisers. These items constitute the hardware required to receive radio signals and perform the Analog to Digital Conversion (ADC) in the frequency bands of interest. In the *SKA*.Low project, they are composed of more than 131k antennas capable of receiving radio signals between 50MHz and 350MHz. The third block in the figure, Digital Signal Processing (DSP), processes the received signals to, filter the signals into channels and perform beamforming and correlation across all the antennas for each frequency channel. As we detail in §3, these operations require processing in real-time all the digitised data from all antennnas with each frequency channel processed by a different FPGA and then recombining results at the output of the DSP. In total, the required processing power is in the two Peta multiply–accumulate operations per second (PMACs) range.

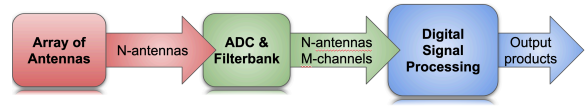


Figure 1: Simplified data flows processing in Radio-Astronomy instruments.

In this paper, we are focusing on improving the third aforementioned stage with *Atomic COTS*. In this section, we present the current trends and limitation of DSP for radio astronomy as well as the advancement in network processing, and finally we give a (very brief) overview of astronomy science products requirements.

2.1 In-network processing

Recently, the emergence of flexible networking hardware and expressive high level domain specific programming languages have enabled deeply programmable networks. With this new generation of network, algorithms can be developed and implemented directly in the switches without the need for costly and time-consuming hardware development. As a result, network elements can forward data streams at line rate (up to Tb/s) and simultaneously perform in-line computation.

These new kinds of programmable networks create the opportunity for in-network computation, i.e., offloading a set of compute operations from end hosts into network devices such as switches and smart NICs. Recently, proposals to use this kind of architecture have emerged to tackle computer science related issues such as fast consensus algorithm in the network (network assisted Paxos) [4], network accelerated key-value stores [8], streaming algorithms [7], data-center data aggregation with MapReduce [15], and even neural network acceleration [16]. In these examples, the network was finally realising the idea of Active Networks from the seminal work of Tennenhouse & Wetherall in the 90s [17].

In our proposal, we explore a novel avenue for in-network processing with the first ever use of this technology in support of DSP in the context of high data volume and high data rate. Indeed, radio astronomy data contained in the Parkes Pulsar Archive [6] is now in the PB realm, data for ASKAP [9] is the tens of PB realm, and astronomy is heading towards 100s of PB with the SKA. *Radio astronomy data is not only large in volume but also in data generation rate, e.g. FAST telescope in China can produce 100 TB a day, ASKAP raw telescope visibilities are approximately 200 TB per day.*

2.2 Going Beyond Custom Made Beamformers and Correlators

In legacy systems, astronomy correlators and beamformers for large telescope arrays were based on bespoke processing boards. Large numbers of ASIC or FPGA boards are typically needed, ASKAP [9] has 360 Redback boards while MeerKAT [10] contains 64 SKARABs boards. The large hardware volumes made it economic to build custom boards rather than using third party products. However, the availability of a custom board typically lagged well behind the release of a new processing chip due to the effort needed to design and debug the boards, followed by kitting parts, qualifying a board manufacturer, and building the board. Also, a research institute that built a custom board had to bear the cost of a board that did not pass Quality Assurance tests. Even after reworking a board that had failed QA, a board loss rate of 10% might still occur. Despite these disadvantages, a bespoke board did allow precise design of the many data interconnections needed between FPGAs, as lack of memory resulted in each FPGA implementing only a single main function, such as wave front correction or correlation.

This has now changed with the introduction of COTS FPGA boards into data centres. These boards are highly cost-competitive, cutting edge designs using the latest processors. They undercut the cost of custom boards due to their much high volume. Their limitation compared to bespoke boards is that even with dedicated 100G QSFPs added their I/O capabilities are an order of magnitude less. But the advent of In-network processing and high bandwidth memory coupled with the Atomic COTS paradigm circumvent this limitation.

2.3 When size matters: quick overview of *SKA*

Astronomers continually strive for more sensitive telescopes to answer increasingly difficult questions about the universe. Questions such as: are humans the only intelligent life, what is 'dark energy', or do black holes have 'hair'? Answering these requires an extremely sensitive radio telescope [2].

Historically, the way to increase sensitivity was to build a bigger dish, but the size of a single dish is limited by gravity to 100m and this limit was reached in the 1960s. Improvements in receivers have since provided increased sensitivity, but this area of improvement is now exhausted and sensitivity can only be improved by adding more antennas. For example, the Very Large Array (VLA) telescope has 27 dishes each 25m in diameter, an equivalent diameter of 130m but with the added bonus of high spatial resolution.

This problem is compounded at low frequencies (wavelengths ~1m or more) because dishes are not cost effective and instead arrays of simple fixed antennas are used such as in LOFAR [19] or Mills Cross (mark 1) [3]. Tens of thousands of antennas are needed to match a 100m dish and to surpass it hundreds of thousands of antennas are needed.

In the context of SKA.*Low, where the* Atomic COTS *solution is planned to be deployed, it is envisioned that the telescope will be composed of 131,072 antennas grouped in 512 stations in the initial phase. Each station would send data from its digitisers at a rate in excess of 12 Gb/s for a total of more than 6Tb/s. This amount of data is then processed by correlators and beamformers to produce up to 8Tb/s of results, which need to be processed in real time in a super computer. In a later phase,* SKA.*Low is envisioned as having an order of magnitude more antennas.*

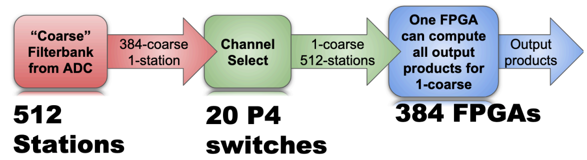


Figure 2: Operation of various components of *Atomic COTS* and their relation to the data stream.

3 ARCHITECTURE

As illustrated in Figure 1, radio astronomy data streams go through various stages before being able to be processed by astronomers to produce science products. With *Atomic COTS*, we aim at improving the final stage of this process by performing beamforming and correlation of multiple coarse frequency channels and across a large number of antennas grouped as 512 stations. We present in Figure 2 a simplified operational view of *Atomic COTS* downstream operations. In this figure, we can see that the digitised traffic from the bespoke receiver block is received in the *Atomic COTS* through some P4 programmable switches. These switches in turn aggregate data to be able to send the correct coarse channel to one or more FPGA cards. These cards then process the data following a staged batch-to-completion model to perform either beamforming or correlation. Once this operation is accomplished, every FPGA card sends its results to the P4 switches that in turn aggregate the results to send it to the correct science product processor.

In this section, we introduce how the combination of programmable switches and FPGAs with 100Gb NICs, such as Xilinx Alveo cards, allows us to efficiently perform digital signal processing at 6Tb/s.

3.1 Network-assisted processing

In our architecture, as mentioned above, programmable switches are the first and last element to be traversed by the telescope data. In ingress, *Atomic COTS* receives UDP packets from the antenna digitisers that each contain one digitised coarse channel from one particular station. This data is encapsulated in dedicated application-level protocols on top of UDP, using formats such as SPEAD or CODIF [11]. These protocols usually contain information related to where and when antennas received and digitised the data for a given coarse frequency channel. In the context of *SKA*.Low, antenna signals are digitised and combined into 512 stations in what is referred to as the Low-Frequency Aperture Array data stream (LFAA) and then sent to *Atomic COTS* using the SPEAD format.

In order to perform DSP on this data stream, the *Atomic COTS* solution uses 20 P4 switches organised in two layers of 9 and 11 switches, where the first 9 switches, referred as I/O switches, interface with the rest of the infrastructure before fanning out to the other 11 switches, referred as intermediate switches, which connect to the FPGA cards. The two-layer structure serves two main purposes. First, it allows for an initial fan-out of the traffic to limit the potential effect of burst of traffic and second, it guarantees a better separation between the *Atomic COTS* and the rest of the telescope infrastructure.

In the ingress pathway, the main objective of the P4 switches is to isolate and deliver the correct coarse channel from all (or a subset of) the antennas to a given FPGA card (or pair of cards). This operation is made possible by routing based on the channel number and antenna ID in the SPEAD data header for the delivery to a single FPGA. In addition to this data isolation and aggregation per coarse channel the P4 switches are also responsible to act as a PTP boundary clock [13] to synchronise all FPGA cards.

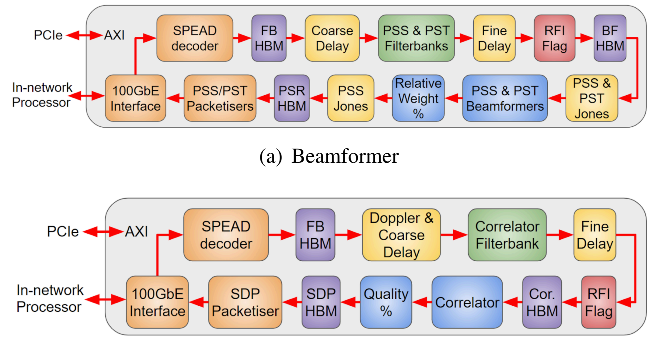
In the egress, the P4 switches send the beamformed and correlated data to dedicated servers. To do so they leverage once again P4 language and are able to perform data aggregation for the science product clusters. In the context of beamforming, this is implemented by sending a given beam number, out of the 16 beams formed channel-by-channel in multiple FPGA cards, to a single dedicated science server to, for example, search and time pulsars within a given beam.

Overall, in *Atomic COTS*, programmable switches are the cornerstone of the architecture capable of facilitating DSP through coarse channel aggregation, time synchronisation, and antenna station selection while easing the integration with the science products thanks to its output aggregation and offering precise telemetry.

3.2 Staged Batch-to-completion Correlation and Beamforming

At the heart of the *Atomic COTS* proposal are the beamforming and correlation functions that enable astronomers to find answers to fundamental questions about the universe as mentioned before. These two functions are performed in FPGA cards that also contain an embedded 100Gb NIC, such as the Xilinx Alveos, following what we called a staged batch-to-completion model for both operations. In this processing model, the FPGA processes data in batches from one HBM and stores the result of a subset of sub-routines inside the next allocated HBM pages for the next stage of DSP. We illustrate this operational model in Figure 3.

To detail how DSP could be implemented and deployed in the *SKA* following the *Atomic COTS* philosophy, we focus in the remainder of this section on the beamformer as shown in Figure 3(a). Operations shown in Figure 3(b) for the correlator follow the same logic but with different signal processing sub-routines.



(a) Beamformer

(b) Correlator

Figure 3: Block diagram representation of proposed staged batch-to-completion DSP.

Using an array of antennas and digital signal processing, beamforming combines the received signals to improve signal to noise ratio by delaying antenna signals so that wavefronts from desired directions add coherently but nose adds only incoherently. In radio-astronomy, this beamforming technique is used to focus an array of antenna on a particular region of the sky. Implementing the delays in firmware allows them to be adjusted, steering the beam to follow the sky as the Earth rotates. For best performance, this signal enhancement requires data from multiple antennas over a set of coarse frequency channels, as follows.

First, as illustrated in Figure 3(a), a given FPGA is receiving data from the 131,072 antennas (aka 512 stations of antennas) for a given coarse channel via the P4 switches. This data is deserialised and buffered in the "FB HBM" (filterbank memory). Once data from all antennas is received, and the beamformer received at least 64ms of data, the FPGA reads out the HBM data with different starting offsets for each station that correct for transmission delays ("coarse delay"). Each of the 512 station data streams is then sent through a filterbank, using FIR filters and FFTs to split it into finer channels. These fine channels then have a finer sub-sampling-interval delay applied as a phase shift, and are checked for radio-frequency interference (RFI) before storing the results into the next buffer labelled "BF HBM".

The beamformer then performs the actual beamforming by summing data from all stations, in small groups of channels. During the summation, various per-station and per-beam complex weights are applied to correct for ionospheric polarisation distortions and to lower beam sidelobes. Using all data from the 512 different stations (131+k antennas), the FPGA calculates 16 beams in parallel, which are stored in the "PSR HBM" buffer before being formatted, serialised and sent for science processing via the P4 switches.

4 ATOMIC COTS FOR PULSAR TIMING

We have implemented and deployed in a lab environment a first iteration of the beamforming capability of *Atomic COTS* as described in §3. In particular, we use an APS network

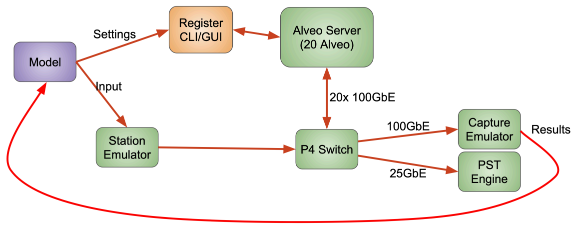


Figure 4: Testing Topology for Pulsar Timing.

```
{"sky":
   [
      [
         {"index":[1],
          "ascension":[20],
          "declination":[10],
          "rate":[1],
          "coarse":[66],
          "RMS":[20],
          "Sinusoids":[-3072],
             "pulsewidth":[512],
          "polarisation":[1,0,1,0],
          "BW":[9000],
          "skyImage":["twoDots.png"],
          "skyImageDimension":[3]
         }
      ]
   ]}
```

Figure 5: Configuration of emulator

BF6064X-T programmable switch with Barefoot Networks’ Tofino chip.The beamforming processing has been implemented in a Xilinx Alveo U50LV card with 8GB HBM. We have installed 20 of those cards in a Comtel server [5].

In order to generate data, we have implemented a simple data generator in C language. This generator takes as an input raw data produced by our Matlab model of the sky and sends it via 40GbE link to the P4 switch. The P4 switch is then connected to 20 Alveo cards in their server, as well as to a capture server and Pulsar Timing (PST) science product server from the *SKA*. The overall topology of our setup is shown in Figure 4.

In the remainder of this section, we first validate our solution against our model of the sky then we demonstrate how its integration with existing radio-astronomy science processing can be accomplished. Overall, we show that we were able to detect a very bright pulsar using *Atomic COTS*.

4.1 Validation with Emulating the Sky

In order to validate our idea, we needed to create a model of the sky against which we could compare our proposal. Although the details of the model are beyond the scope of the present article, Figure 5 shows how our model is configured and how it would generate emulated data for each station of antennas. In a nutshell, our model takes various parameters describing the orientation and rotation rate of the sky, which coarse frequencies to emulate, the envelope of the signal, the signal to noise ratio and more importantly the image of the sky. In this example, we are interested in the very simple sky represented in the Figure 6, where we have two pulsars.

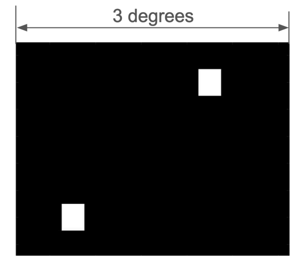


Figure 6: Emulated Sky. White is a bright source, black is no source.

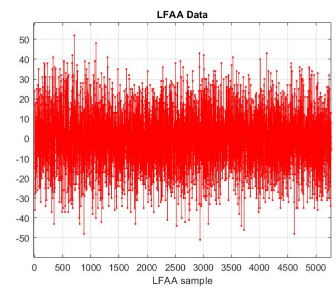


Figure 7: Sample of 3.32ms of Input Data from an emulated station (SNR = -10dB, 2 beams each looking at a different pulsar).

Using these parameters, our model generates signals for each station of antennas with some added noise. Overall, in our emulation, we have generated signals corresponding to 64 stations located in a circle of 15km diameter. We show in Figure 7 an example of such signal for a given station. As expected, the signal from a single station is very noisy and not really usable on its own, however, two different pulsars are buried in the noise, with period of 3072 LFAA samples (3.32ms), a pulse width of 512 LFAA samples (553μs), and SNR of -10dB. Finally, in addition to the raw traffic from each station, our model generates beamformer parameters that are pushed to the FPGA firmware prior to the experiment.

Our traffic generator then sends the traffic from the 64 emulated stations to the *Atomic COTS* following the topology displayed in Figure 4. The amount of traffic entering the system is about 14Gb/s for the coarse channel we are interested in. Upon reception, the P4 switch routes this traffic to a single FPGA card using the coarse channel number. Then, the FPGA forms two beams based from the 64 stations before sending this output back to the P4 switch. Finally, upon reception of the beamformed data, the P4 switch sends one beam to our control server and the second one to the production ready Pulsar Timing engine from the *SKA* project.

We validated the *Atomic COTS* by capturing all packets from a single beam in a Linux server through a ConnectX5 NIC. After processing we were able to visualise the single beam as shown in Figure 8. In this figure, we display the

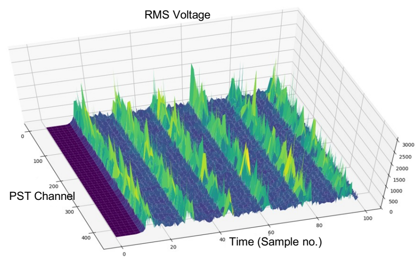


Figure 8: Visualisation of one Beam in simulated environment.

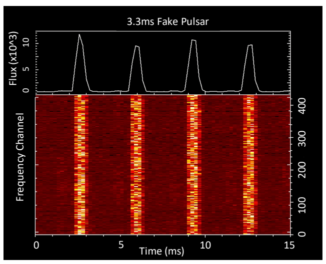


Figure 9: Visualisation of one Beam in production environment with DSPSR framework.

frequency-time output of beamforming processing for single beam. As expected, the simulated pulsar's signal is now visible as periodic pulses across numerous frequency channels.

4.2 Atomic COTS and Existing Science Software

Using emulated signals from 64 stations, we have demonstrated through trace captures in a laboratory environment how the *Atomic COTS* proposal was capable of beamforming radio-astronomy data. In particular, we have shown in Figure 8 that the beamformer output appears to contain a pulsar. Using state-of-the-art DSP for pulsar astronomy, the Digital Signal Processing Software for Pulsar Astronomy (DSPSR) framework [20], we are further validating our proposal while also demonstrating its integration in the more general picture of a functioning radio-telescope.

As explained above, the beamformer implemented in a Xilinx Alveo FPGA card was configured to form two beams from 64 stations of antennas. One beam was captured in a Linux server while the second was sent by the P4 switch to DSPSR server. From a management point of view, this integration was done without any major disturbance. We only had issues with the server sending ARP request when receiving the first data packet, that was solved by implementing ARP within the P4 switch to act on behalf of FPGA cards.

In terms of results, the DSPSR toolkit was able to detect the emulated pulsar as illustrated in Figure 9. In this figure, we can observe 432 fine channels, and our "fake" millisecond pulsar is measured to have a period of 3.31776ms and a width of 0.55296ms as configured in the model.

5 OPEN QUESTIONS AND ROADMAP

In this article, we have introduced the *Atomic COTS* approach for radio-astronomy DSP when facing extreme data throughput from hundreds of thousands of receiver antennas. This approach advocates for a mix of in-network processor supporting DSP done in COTS FPGAs with attached High Bandwidth Memory and embedded 100Gb NIC. In this architecture, we have proposed to use P4-programmable switches as data aggregator for FPGA card (selection of the correct frequency channel and correct sets of antennas to analyse) while easing the integration with legacy systems. Additionally, we proposed to perform beamforming and correlation from potentially all physical antennas in a single FPGA for a given coarse frequency channel, hence performing an Atomic DSP in each FPGA. We demonstrated the potential of this approach in an emulated environment aiming at reproducing the pulsar timing capability of the *SKA* telescope.

Nonetheless, our approach is far from being a definitive answer for in-network assisted DSP, whether this DSP is focused on radio-astronomy or for telecommunication networks. Indeed, while we demonstrated that DSP was possible with our approach, we have also buried numerous parameters and architectural decisions that made this demonstration possible. For example, in the beamforming filterbank, we mentioned how we were using a 64ms window for performing the FFT, thus guaranteeing a given frequency sampling, what would happen if we could increase this precision? Could we do that without a strict need of proportionally increasing the amount of available memory? In regards to the role of in-network processors, we are currently proposing to use them as data aggregator and multicast agent without modifying the radio-astronomy data. Would it be possible to use them further and combine signals in the frequency domain to calculate correlation between antennas knowing that this data is already quantized and thus very suitable for non-floating operations?

Finally, while the *Atomic COTS* proposal is currently focusing on the processing of radio-astronomy data before being usable by astronomers, there might be an avenue for extending its range down the stream (i.e., after the beamforming and correlation). In the current architecture, after the DSP, data is sent to supercomputers where it can either be analysed in real-time or archived. Therefore, there could be an argument to be made for some further pre-processing tailored to very specific astronomy related algorithms. These changes would require tight collaboration with astronomers as it might require a more thorough questioning of data protocols.

ACKNOWLEDGMENTS

This work is funded by the Australian Government Department of Industry, Science, Energy and Resources (DISER) and the Australia Research Council (ARC) Linkage Infrastructure, Equipment and Facilities (LIEF) grants. The authors would like to also thank APS-Networks and Xilinx for their continuous support.

REFERENCES

[1] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese, and D. Walker. P4: Programming protocol-independent packet processors. *SIGCOMM Comput. Commun. Rev.*, 44(3):87–95, July 2014.

[2] C. Carilli and S. Rawlings. Motivation, key science projects, standards and assumptions. *New Astronomy Reviews*, 48(11-12):979–984, Dec 2004.

[3] CSIRO. The flowering of fleurs: an interesting interlude in australian radio astronomy. https://www.atnf.csiro.au/news/newsletter/jun02/Flowering_of_Fleurs.htm, 2002.

[4] H. T. Dang, M. Canini, F. Pedone, and R. Soulé. Paxos made switch-y. *ACM SIGCOMM Computer Communication Review*, 46(2):18–24, 2016.

[5] C. Electronic. 20 slot-pcie 4.0 server. https://comtel-online.com/systems/20-slot-pcie-4/, 2021.

[6] G. Hobbs, D. Miller, R. Manchester, J. Dempsey, J. M. Chapman, J. Khoo, J. Applegate, M. Bailes, N. Bhat, R. Bridle, et al. The parkes observatory pulsar data archive. *Publications of the Astronomical Society of Australia*, 28(3):202–214, 2011.

[7] T. Jepsen, M. Moshref, A. Carzaniga, N. Foster, and R. Soulé. Life in the fast lane: A line-rate linear road. In *Proceedings of the Symposium on SDN Research*, page 10. ACM, 2018.

[8] X. Jin, X. Li, H. Zhang, R. Soulé, J. Lee, N. Foster, C. Kim, and I. Stoica. Netcache: Balancing key-value stores with fast in-network caching. In *Proceedings of the 26th Symposium on Operating Systems Principles*, pages 121–136. ACM, 2017.

[9] S. Johnston, R. Taylor, M. Bailes, N. Bartel, C. Baugh, M. Bietenholz, C. Blake, R. Braun, J. Brown, S. Chatterjee, et al. Science with askap. *Experimental astronomy*, 22(3):151–273, 2008.

[10] J. Jonas and M. Team. The meerkat radio telescope. *Proceedings of MeerKAT Science: On the Pathway to the SKA*, pages 25–27, 2018.

[11] J. Manley, M. Welz, A. Parsons, S. Ratcliffe, and R. Van Rooyen. Spead: streaming protocol for exchanging astronomical data. *SKA document*, 2010.

[12] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74, 2008.

[13] L. Montini, T. Frost, G. Dowd, and V. Shankarkumar. Precision time protocol version 2 (ptpv2) management information base. 2017.

[14] R. Perley, P. Napier, J. Jackson, B. Butler, B. Carlson, D. Fort, P. Dewdney, B. Clark, R. Hayward, S. Durand, et al. The expanded very large array. *Proceedings of the IEEE*, 97(8):1448–1462, 2009.

[15] A. Sapio, I. Abdelaziz, A. Aldilaijan, M. Canini, and P. Kalnis. In-network computation is a dumb idea whose time has come. In *Proceedings of the 16th ACM Workshop on Hot Topics in Networks*, HotNets-XVI, pages 150–156, New York, NY, USA, 2017. ACM.

[16] A. Sapio, M. Canini, C.-Y. Ho, J. Nelson, P. Kalnis, C. Kim, A. Krishnamurthy, M. Moshref, D. R. K. Ports, and P. Richtárik. Scaling Distributed Machine Learning with In-Network Aggregation. In *Proceedings of NSDI'21*, Apr 2021.

[17] D. L. Tennenhouse and D. J. Wetherall. Towards an active network architecture. *Computer communication review.*, 26(2):5–17, 1996.

[18] B. University. Collabo-ration for astronomy signal processing and electronics re-search. https://casper.berkeley.edu/.

[19] M. P. van Haarlem, M. W. Wise, A. Gunst, G. Heald, J. P. McKean, J. W. Hessels, A. G. de Bruyn, R. Nijboer, J. Swinbank, R. Fallows, et al. Lofar: The low-frequency array. *Astronomy & astrophysics*, 556:A2, 2013.

[20] W. van Straten and M. Bailes. Dspsr: Digital signal processing software for pulsar astronomy. *Publications of the Astronomical Society of Australia*, 28(1):1–14, 2011.

[21] Z. Wang, H. Huang, J. Zhang, and G. Alonso. Benchmarking high bandwidth memory on fpgas. *arXiv preprint arXiv:2005.04324*, 2020.

[22] W. E. Wilson, R. Ferris, P. Axtens, A. Brown, E. Davis, G. Hampson, M. Leach, P. Roberts, S. Saunders, B. Koribalski, et al. The australia telescope compact array broadband backend (cabb). *arXiv preprint arXiv:1105.3532*, 2011.