Efficient processing of physics quantities for the Processing Unit for the upgrade of the Tile Calorimeter of ATLAS

D Ohene-Kwofie[1], E Otoo[1] and B Mellado [2]

[1] School Of Elect. & Info Engineering ,University Of the Witwatersrand, Johannesburg, SA
[2] School of Physics, University of the Witwatersrand, Johannesburg 2050, South Africa

E-mail: daniel.ohene-kwofie@cern.ch

Abstract. The ATLAS detector, operated at the Large Hadron Collider (LHC) records proton-proton collisions at CERN every 25ns resulting in a sustained data flow up to Pb/s. The upgraded Tile Calorimeter of the ATLAS experiment will sustain about 5PB/s of digital throughput. These massive data rates require extremely fast data capture and processing. Although there has been a steady increase in the processing speed of CPU/GPGPU assembled for high performance computing, the rate of data input and output, even under parallel I/O, has not kept up with the general increase in computing speeds. The problem then is whether one can implement an I/O subsystem infrastructure capable of meeting the computational speeds of the advanced computing systems at the petascale and exascale level. We propose a system architecture that leverages the Partitioned Global Address Space (PGAS) model of computing to maintain an in-memory data-store for the Processing Unit (PU) of the upgraded electronics of the Tile Calorimeter which is proposed to be used as a high throughput general purpose co-processor to the sROD of the upgraded Tile Calorimeter. The physical memory of the PUs are aggregated into a large global logical address space using RDMA-capable interconnects such as PCI-Express to enhance data processing throughput. Additionally, lossless (i.e. original data can be perfectly reconstructed from the compressed data) compression schemes are explored to enhance data bandwidth utilisation and provide increased throughput.

1. Introduction

The Large Hadron Collider, is the most powerful proton-proton collider ever built [1]. The discovery of the Higgs Boson [2, 3] was independently observed by the ATLAS and Compact Muon Solenoid (CMS) detectors. A Toroidal Large Hadron Collider Apparatus (ATLAS) is the largest of all the LHC detectors. It consists of a series of concentric rings: the inner detector, Tile Colorimeters and the Muon Spectrometers. TileCal is the central hadronic calorimeter of the ATLAS experiment at the LHC at CERN. It is primarily used to measure the energy and direction of hadrons and jets as they are produced. The Trigger and Data Acquisition System (TDAQ) is designed for event selection, processing and storage of the read-out data of the detector [4]. This selection mechanism is based on three trigger levels in the data flow that defines the different domains for the read-out electronics in terms of methods and rates for this selection.

Detector electronics are being upgraded (as part of scheduled upgrade phases on the accelerator and experiments components) to allow an increase in the Level-1 acceptance rate of

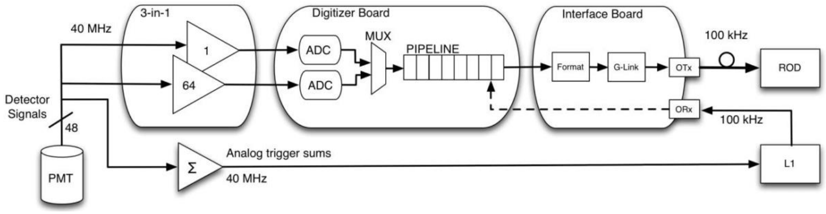

Figure 1: Current Architecture of the Tile Calorimeter.

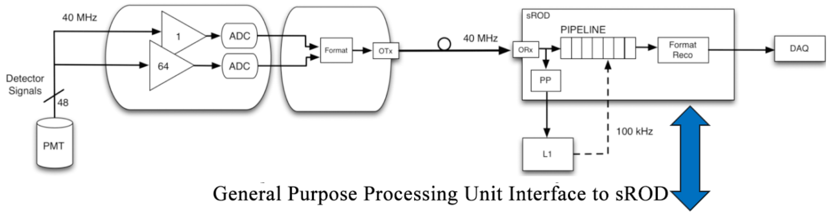

Figure 2: Upgraded Architecture of the Tile Calorimeter.

events from 70kHz to 100kHz [5]. Subsequently increasing the raw data sent for processing.

We present an architecture for on-line data processing using an ARM cluster configuration, where the data is maintained in an in-memory storage for the upgraded TileCal. The architecture consists of a cluster of ARM processing units, part of whose memories are aggregated to form a combined large logical global address space to facilitate in-memory data processing.

1.1. The Off-line Data processing Challenge

Scheduled upgrades to the ATLAS detector anticipated in 2022 will result in a much higher rate of collisions [6] at the LHC, resulting in an increase by 200 times the current rate to over 41Tb/s data output from the TileCal [4]. Storing such massive dataset for off-line processing presents a great challenge and is not desirable. The MAC project at the University Of Witwatersrand is aiming for a cost-effective, and high data throughput Processing Unit (PU), using several consumer ARM processors in a cluster configuration, as general purpose co-processor to augment the read-out system (sROD) of the upgraded TileCal.

A major challenge with processing such large volumes of data is the input/output (I/O) sub-system. The continuously growing gap between CPU and I/O speed has resulted in the conspicuous performance gap between the processor speeds and what the storage I/O subsystem can deliver. Figure 3 shows the increasing gap between CPU speed and disk storage. The solution is to reduce disk accesses and enhance in-memory data processing.

We present a brief overview of a complete in-memory storage system for on-line data processing in the ARM cluster configuration for the upgraded electronics of the TileCal. In-memory data processing provides extremely fast response time and very high throughput, with an average of about $100 - 1000$ times lower latency for a complete Random Access Memory(RAM) storage than disk-based storage systems and consequently a $100 - 1000$ times greater throughput [9].

2. PGAS Architecture for the processing of physics quantities

The architecture consists of a cluster of ARM processing units, part of whose memories are aggregated to form a combined logical global address space. The general operational architecture includes low cost ARM processing units interconnected via PCI Express interconnects (PCIe). The PCIe interconnect facilitates Remote Data Memory Access (RDMA) and offers very low-latency host-to-host transfers by copying the information directly between the host application

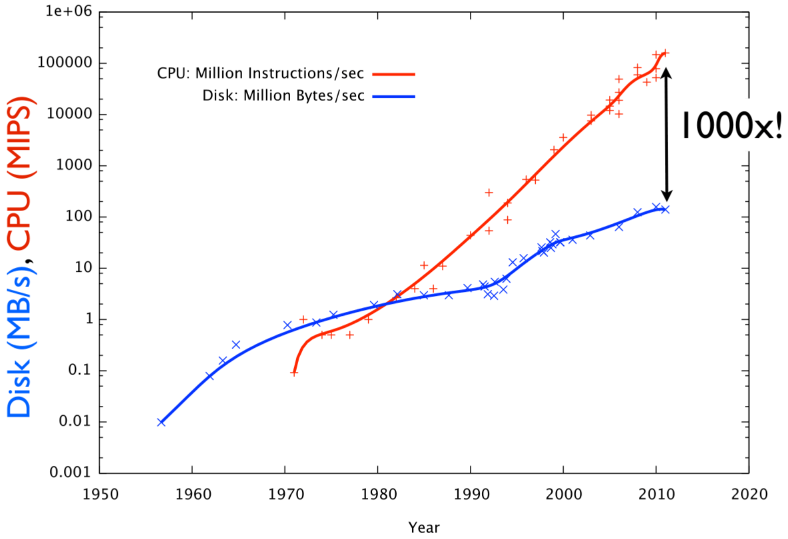

Figure 3: Increasing gap between disks and CPU speed [7].

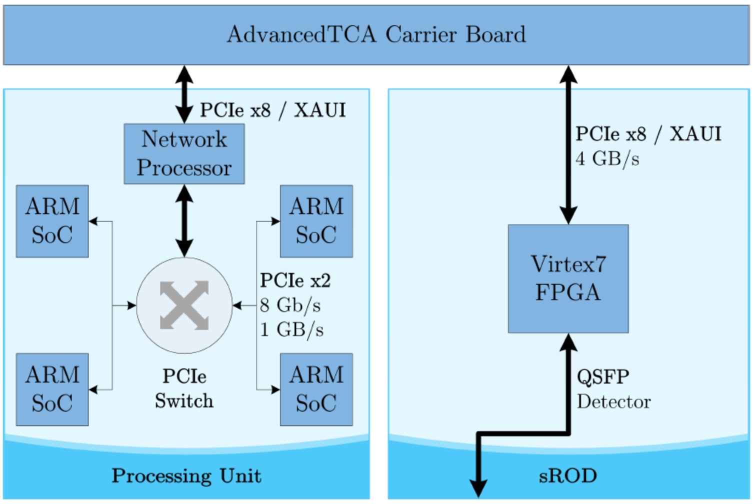

Figure 4: Schematic of the PU and SROD prototype connection [8].

memories. This enhances a seamless global logical address space for in-memory data processing as a low-overhead protocol. They are relatively affordable, low power and also provide straightforward, standards-compliant extensions that address multi-host communication and I/O sharing capabilities. Figure 5 illustrates the schematic diagram of the architecture.

Each block depicts an ARM node with 4 cores. The architecture seeks to provide low latency guarantees by ensuring effective fast access to data in-memory for application processing. Thus, minimising I/O which is the bottleneck for high throughput computing required by the ATLAS. Part of the on-going research work is to address a number of challenges related to fault-tolerance and high speed access of the memory resident data.

3. The PGAS High Data Throughput Architecture

The Figure 5 and the experiments conducted so far only demonstrates the processing capability of the PU under the PGAS computing model. This is easily transformed into a configuration

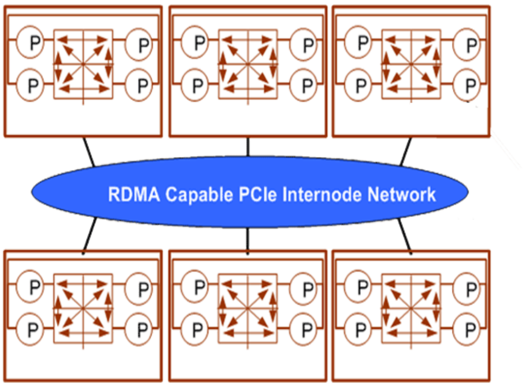

Figure 5: Schematic Diagram of the Architecture.

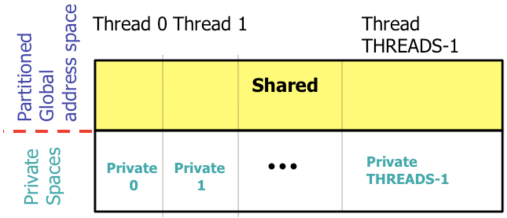

Figure 6: PGAS Memory layout.

where the aggregated memories of the ARM processors serve as a large in-memory buffer for data staging. By augmenting the ARM cluster with data input ports and data output ports to external hierarchical storage devices, data can be streamed into the ARM cluster memory using parallel I/O which can then be compressed and streamed out using parallel I/O.

Lossless data compression methods such as zlib and bzlib2, and general purpose compression schemes, present methods to encode the data with fewer bits by removing the redundancies in the data before either storage or transmission over a communication channel. Such techniques further enhance the efficiency and ensure effective use of available bandwidth of the communication channel. This results in the reduction of the amount of data injected in the channel, thereby enhancing the data throughput of the physics quantities of the upgraded ATLAS Tile Calorimeter.

4. Preliminary Evaluations

Preliminary investigations conducted show promising results. We benchmarked the system with the NASA Advanced Supercomputing (NAS) Parallel Benchmarking tool [10]. This benchmark was designed not just for parallel-aware algorithmic and software methods but also to provide an easy verifiability of correctness of results and performance figures. The evaluations are run using the Fast Fourier Transform(FFT) algorithm which solves a 3D partial differential equation using an FFT-based spectral method [10], also requiring long range communication. FFT performs three one-dimensional(1-D) FFT's, one for each dimension. The FFT benchmark is adopted since they are applicable in several signal processing algorithms, and also to the optimal filtering task of the Tilecal.

The evaluation was done on 4 nodes of the Wits High Throughput Electronic Lab (HTEL) Tegra K1 (2.3GHz Quad-Core ARM Cortex-A15) cluster with 2GB of memory each and 1Gbp Ethernet interconnect between nodes

We ran the benchmark with varying workloads as well as varying number of threads. Each workload is ran 6 times and the resulting throughput in floating point operations per second(MFLOPS) reported. A maximum of 4 threads per node are spawned and the dimensions of the FFT is varied from small ($64 \times 64 \times 64$ 3D grid) to large ($256 \times 256 \times 128$ 3D grid). The User Datagram Protocol (UDP) is used as the inter-process communication protocol between cluster nodes. UPC supports UDP, Message Passing Interface (MPI) as well as RDMA capable interconnects(e.g infiniband, PCIe, 10GbE, etc.) for inter-node communication. The current experimental setup did not have RDMA and therefore UDP was used since it is usually faster than MPI as far as inter-node communication over Ethernet is concerned.

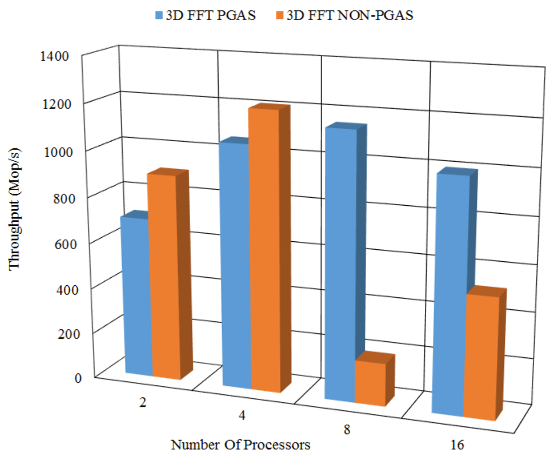

Figure 7: Throughput for small workloads.

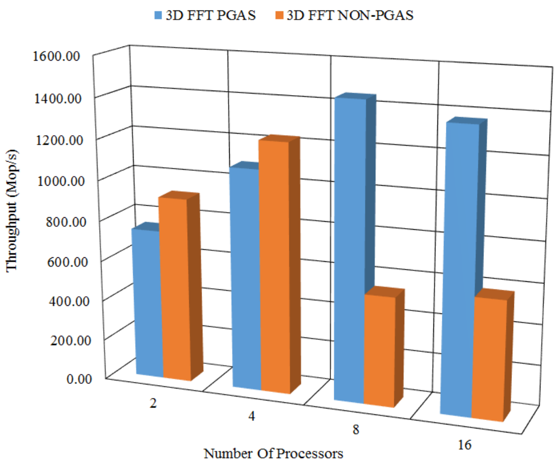

Figure 8: Throughput for large workloads.

Figures 7 and 8 show the results for the data processing throughput in MFLOPS with varying number of threads and workloads. Generally, as the number of threads increase, there is a corresponding increase in the data processing throughput since less processing is done per thread. We observe a significant and better performance increase with PGAS as depicted in Figure 8 (about 3× more throughput than the NON-PGAS FFT with 8 threads). The NON-PGAS experienced a significant drop in performance when the number of threads increased from 4 to 8. This could be due to the communication overhead as data is transferred across nodes for processing and aggregation.

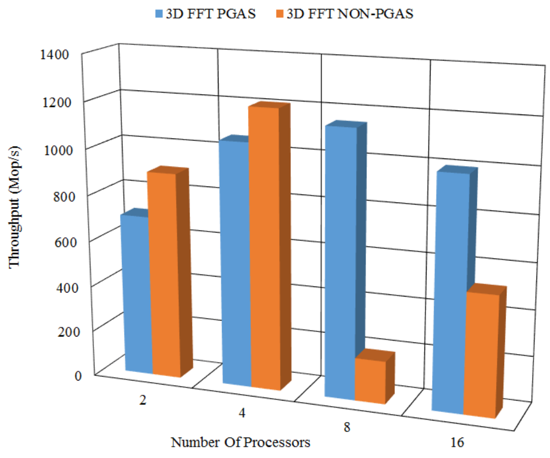

Figure 9: Average latencies for varying workloads.

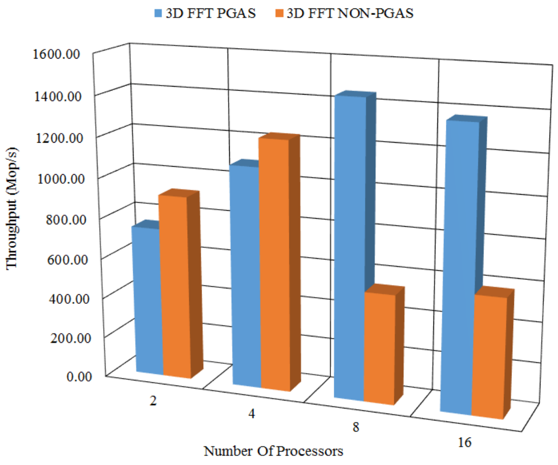

Figure 10: Latencies for large workloads.

Figure 9 shows the average latencies for each run of the experiment. There is a general increase in latency as workload size increases as expected. We also observe that as the number of threads increase latency drops a little and begins to increase after 8 threads. This is due to increase in inter-node communication. PGAS performs much better with lower latencies as compared to its NON-PGAS variant.

With a much higher bandwidth device such as 20Gb/s PCIe connectivity or a 10GbE, we expect minimal latencies and much higher throughput (GFLOPS).

5. Conclusions

Management, efficient access and analysis of the Petabytes of data, that is likely to be generated and/or used in the upgraded ATLAS TileCal present extremely challenging tasks. I/O bottlenecks in processing such huge amounts of data require techniques that utilise higher levels of the memory hierarchy to enhance data throughput.

The PGAS model presents an efficient way to process data in a distributed environment by providing a global partitioned shared address space for in-memory data. This enhances the data processing throughput. Additionally, the use of low cost CPUs such as ARM with PCIe interconnects, ensure low power consumption (high performance per watt) and thus cost effective alternative for data processing in ATLAS TileCal. Data compression frameworks are also a great strategy to both effectively utilise I/O bandwidth and thus increase data throughput. Since off-line processing of the huge data volume from the TileCal is a great challenge, on-line processing using cost effective PUs, enhanced with PGAS processing techniques, has been proposed. The strategy both ensures high throughput and efficient data bandwidth utilisation when coupled with various compression techniques.

Future work anticipated includes further rigorous experimentation using the ARM PUs with RDMA capable PCIe intra/interconnects for kernel bypass applications. Additionally, various lossless compression frameworks will be explored to further determine which scheme is effective for on-line data processing without compromising on high throughput.

Acknowledgments

We acknowledge the financial assistance of the South African National Research Foundation (NRF) and SA-CERN towards this research. We would also like to extend our gratitude to the School of Physics for making resources available for this project.

References

[1] CERN 2014 The Large Hadron Collider `http://home.web.cern.ch/topics/large-hadron-collider`

[2] Baines J T and et al 2004 *IEEE Transactions on Nuclear Science* **51** 361–366 ISSN 0018-9499

[3] Reed R and et al 2013 A Revised High Voltage Board for the Consolidation of Front End Electronics on the Tile Calorimeter of the ATLAS Detector at the LHC *SAIP 2013* (Johannesburg, South Africa)

[4] Carrioa F and et al 2014 *Journal of Instrumentation* **C02019**

[5] The ATLAS Collaboration 2012 Letter of Intent for the Phase I Upgrade of the ATLAS `https://cdsweb.cern.ch/record/1402470`

[6] The ATLAS Collaboration 2012 Letter of Intent for the Phase II Upgrade of the ATLAS `https://cdsweb.cern.ch/record/1502664`

[7] Jonathan Dursi, SciNet 2012 Parallel I/O doesnt have to be so hard: The ADIOS Library `http://wiki.scinethpc.ca/wiki/images/8/8c/Adios-techtalk-may2012.pdf`

[8] Cox M A, Reed R and Mellado B 2015 *Journal of Instrumentation* **C01007**

[9] Ousterhout J, Agrawal P, Erickson D, Kozyrakis C, Leverich J, Mazières D, Mitra S, Narayanan A, Ongaro D, Parulkar G, Rosenblum M, Rumble S M, Stratmann E and Stutsman R 2011 *Commun. ACM* **54** 121–130 ISSN 0001-0782

[10] Serres O, Andreev N, Francois C, Abhishek A, Smita A, Veysel B, Yiyi Y, Chauvin S, Vroman F and El-Ghazawi T 2011 UPC NAS Parallel Benchmarks `http://threads.hpcl.gwu.edu/sites/npb-upc`

[11] Phoboo A E 2014 Dealing With Data `http://atlas.ch/news/2014/dealing-with-data.html`

[12] ATLAS Collaboration 2008 *Journal of Instrumentation* **S08003**

[13] George M 2014 *Journal of Instrumentation* **C05004**

[14] Baines J T, Bee C P, Bogaerts A, Bosman M, Botterill D, Caron B, Dos Anjos A, Etienne F, Gonzalez S, Karr K, Li W, Meessen C, Merino G, Negri A, Pinfold J L, Pinto P, Qian Z, Touchard F, Werner P, Wheeler S, Wickens F, Wiedenmann W and Zobernig G 2004 *IEEE Transactions on Nuclear Science* **51** 361–366 ISSN 0018-9499

[15] El-Ghazawi T, Carlson W, Sterling T and Yelick K 2005 *UPC:Distributed Shared Memory Programming* (Hoboken, New Jersey: John Wiley & Sons Inc.)