Exploiting Parallelism in the TileCal Trigger System with GPGPU

Marc Sacks

School of Physics, University of the Witwatersrand, Johannesburg 2050, South Africa

E-mail: marc.sacks@cern.ch

Abstract. After the 2022 upgrades, the Tile Calorimeter (TileCal) detector at ATLAS will be generating raw data at a rate of approximately 41 TB/s. The TileCal triggering system contains a degree of parallelism in its processing algorithms and thus presents an opportunity to explore the use of general-purpose computing on graphics processing units (GPGPU). Currently, research into the viability of an sROD ARM-based co-processing unit (PU) is being conducted at Wits University with especial regard to increasing the I/O throughput of the detector. Integration of GPGPU into this PU could enhance its performance by relieving the ARMs of particularly parallel computations. In addition to the PU, use of GPGPU in the front-end trigger is being investigated on the basis of the used algorithms having a similarity to image processing algorithms - where GPU can be used optimally. The use of GPUs in assistance to or in place of FPGAs can be justified by GPUs' relative ease of programming; C/C++ like languages as opposed to assembly-like Hardware Description Languages (HDLs). This project will consider how GPUs can best be utilised as a subsystem of TileCal in terms of power and computing efficiency; and therefore cost.

1. Introduction

A Toiroidal LHC ApparatuS (ATLAS) is a detector in the Large Hadron Collider (LHC) at CERN [1]. It is a general purpose detector used in standard model and beyond the standard model physics [1]. It comprises several sub-detectors; namely the inner detector (ID), the electromagnetic and hadronic calorimeter, and the muon spectrometer [1]. The hadronic calorimeter generates a large amount of data very rapidly: currently 205 Gbps and in 2022 about 41 Tbps [2]. This amount of data requires specialised computational systems in order to be processed. This calorimeter functions in conjunction with a read-out driver (ROD) which is responsible for reconstructing the energy profile of detected events [2]. The ROD is being updated to a super ROD (sROD), this document describes a proposed ARM-processor/GPU based sROD co-processor. Before data reaches the ROD it is processed by systems belonging to the calorimeters, the algorithms involved in these electromagnetic and hadronic calorimeters will be considered in terms of parallel processing [1].

2. The ATLAS Detector

The purpose of the various detectors is to measure the energy and trajectories of the particles formed within it due to high-energy proton-proton (p-p) and heavy ion collisions [1]. The ID reconstructs vertex and momentum measurements for charged particles, the muon spectrometer measures the momentum and position of muons. The final detector, positioned in between


Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

the ID and muon spectrometer, is the calorimeter [1]. The calorimeter has two subsystems, namely the electromagnetic (EM) and hadronic calorimeters [1]. The EM detector is a liquid argon based detector, while the hardonic calorimeter is composed of sheets or tiles of plastic scintillators sandwiched between inactive plates of steel. The tiles of plastic scintillator have led to the hadronic calorimeter being referred to as the tile calorimeter or TileCal.

3. Tile Calorimeter

TileCal is approximately 11 m in length and its inner and outer radii are 2.28 and 4.25 m respectively [1]. It is divided into three barrel-like regions; one central barrel and two lateral barrels. The barrels are made up of 64 wedges composed of plates of iron and the plastic scintillator. At each of the tiles' edges are wavelength shifting fibres connected to two photomultiplier tubes (PMTs). It is the signal from the PMTs that form the basis of the TileCal trigger system.

3.1. TileCal Trigger System

The trigger system was developed out of the need to separate rare physical processes from background noise. Proton bunches collide in ATLAS every 25 ns [1]. This 40 MHz bunch crossing rate produces a number of detectable particles which, when converted into a digital signals, generate data at 205 Gbps. After the 2022 upgrade of TileCal, this will increase to 41 Tbps. Storing this amount of data is neither possible nor desirable. Instead the approach is to detect events of interest in real-time and to store those events for analysis - all other data is discarded. The trigger system is divided into a Level-1, Level-2 and Higher Level Trigger (HLT). Level-1 is a rapid hardware-based algorithm which selects events from the 40 MHz bunch crossing in such a way as to reduce the recorded events to approximately 75 kHz. The software based Level-2 trigger and HLT further reduce the recorded event rate to 3.5 kHz and 200 Hz respectively. A system overview is shown in Figure 1. Signals from the calorimeters are digitised and undergo processing which organises data into region of origin and time of origin (and therefore bunch crossing). The data is then sent to the Cluster Processor Modules (CPMs). These modules are responsible for implementing the logic which dictates whether an event will be sent to the read-out driver (ROD) and eventually to the Level-2 trigger and HLT. The logic implemented by the CPMs will be discussed in relation to GPU after a discussion of parallel programming.

4. Parallel Programming Paradigms and GPGPU

The progress of CPU technology has seen a trend of ever faster clock speeds [3]. This of course implies the ability to perform more calculations per second in a serial fashion, meaning calculation *n+1* cannot occur before calculation n has terminated. Due to physical and monetary limitations, clock speeds can no longer be made faster and so to achieve more efficient computing (in terms of power consumption, time and budget) an alternative paradigm must be sought. Parallel programming and processing attempts to increase computational efficiency by processing calculation n+1 while simultaneously processing calculation n (provided the calculations are independent of one another.) Parallel processing is essentially the distribution of a computational load over several processors. These processors can be entire computers, as is the case for a network, or it could be a single integrated chip (IC) possessing multiple cores [3]. The latter is the true for Graphic Processing Units (GPUs), which can possess hundreds of CPUs on a single IC [3]. Each of these cores need not operate as rapidly as a conventional processor to achieve efficient computational results, because the cores operate in tandem to solve multiple calculations simultaneously. GPUs, as the name suggests, were developed with the specific intention of processing and rendering graphics. Initially GPUs were largely inaccessible and used only for their original purpose, but as their usefulness became recognised, they begun to

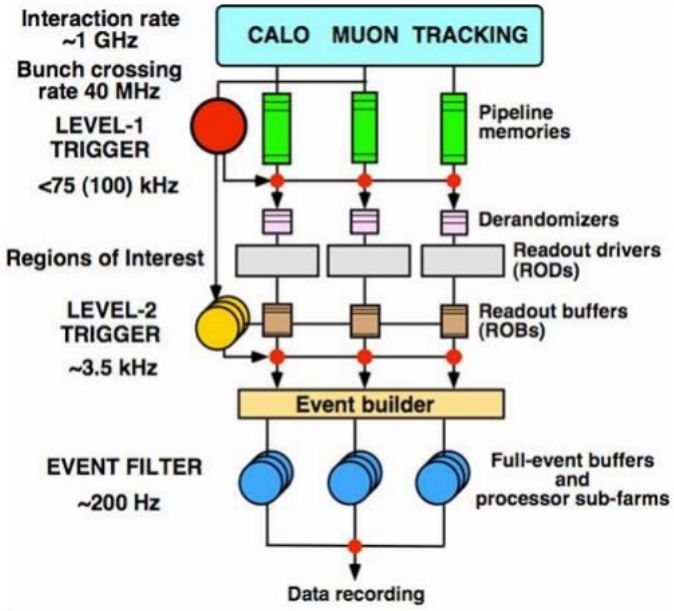

Figure 1. An overview of the ATLAS calorimeter trigger system.

be utilised for processing computational loads other than graphics. To facilitate this, GPU programming languages have been developed which are similar in style to C/C++ and are thus accessible to a large number of software developers. Certain computations are more suited to parallel processing than others. The computations must be large enough to benefit from a distribution over multiple processors, and the calculations must be as independent of each other as possible. If the calculations depend on one another then it could be the case that they cannot run in parallel, hence defeating the purpose of parallel processing.

5. GPGPU Applicability to Calorimeters

The problem then becomes one of assessing how appropriate parallel computing is in the case of the TileCal Level-1 trigger.

5.1. GPGPU and the sROD

The ROD receives data from TileCal and is responsible for reconstructing the energy profiles of events before sending them to the Level-2 and HLT [2]. The energy reconstruction is done using an optimal filtering algorithm based on a weighted sum of the data samples [2]. The main components of the ROD are two FPGAs and two DSPs [2]. For the 2022 upgrade, an updated ROD referred to as the super ROD or sROD is proposed [2]. The current sROD design is FPGA based, however research into an ARM processor based co-processor unit for the sROD is currently being conducted at the University of the Witwatersrand [2]. Apart from performance enhancement, the design is a move from specialised, costly equipment to readily accessible inexpensive equipment. The ARM processors are capable of performing the optimal filtering algorithm, however it is proposed that GPUs can be used in conjunction with ARM processors to form a more robust co-processing unit, especially in light of the fact that the algorithm is not serial in nature.

5.2. Cluster Processor Module Algorithms

The Cluster Processor Module (CPM) algorithm decides whether an event is sent to the ROD [1]. It operates as follows. As stated, the three barrel regions are made up of 64 wedges. Each of these wedges is radially segmented into 3 layers. These 3 layered wedges are referred to as trigger towers. The CPM algorithm considers the trigger towers in sets of 4x4 overlapping windows, one such window it shown in Figure 2. The EM algorithm sums the E_T for each of the four 1x2 and 2x1 combinations as illustrated in Figure 2, and each combination is checked against a programmable threshold. This sum is then compared to the sums for the 12 adjacent windows as indicated in Figure 3 in order to locate a local maximum, this is referred to as the isolation threshold. The same is done with the hadronic algorithm, but each sum is then added to the sum of the 2x2 hadronic calorimeter region. The same isolation threshold is performed. Of importance to this discussion is the sliding-window nature of the CPM module. If one considers each position of the window to be a pixel making up an image, then the algorithm can be thought of as an image processing one. It is thus suggested that GPUs could be used as a replacement to the FPGAs currently in use.

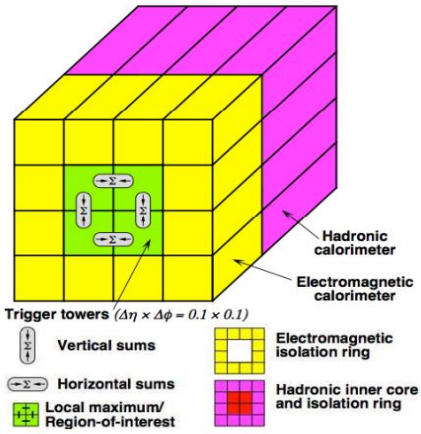

Figure 2. An illustration of the CPM summing process [1].

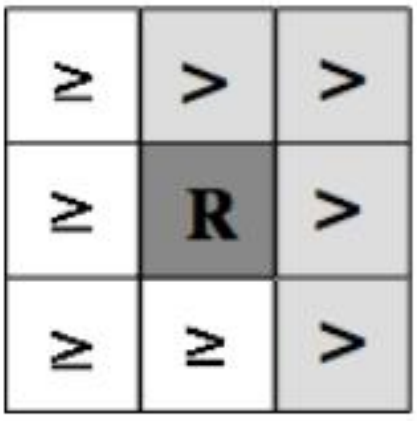

Figure 3. An illustration of the isolation threshold [1].

6. Conclusion

In order to deal with the massive amount of data generated by the hadronic calorimeter after the 2022 upgrade, GPGPU could be used in combination with ARM processors to provide an energy-efficient, user-friendly processing unit.

References

[1] The ATLAS Collaboration *et al* 2008 *JINST* **3** S08003
[2] Cox M, Reed R, Mellado B 2015 *JINST* **10** C01007
[3] Pacheco P S 2011 *An Introduction to Parallel Programming* (USA: Elsevier Inc)