University of Minho Informatics Department

Dissertation for Master's Degree in Informatics Engineering

Efficient processing of ATLAS events analysis in platforms with accelerator devices

André Pereira (Student) pg19828@alunos.uminho.pt

Alberto Proença (Advisor) aproenca@di.uminho.pt

António Onofre (Co-Advisor) onofre@fisica.uminho.pt

Braga, November 2012

Abstract

Most event data analysis tasks in the ATLAS project require both intensive data access and processing, where some tasks are typically I/O bound while others are compute bound.

This dissertation work mainly focus on compute bound issues at the latest stages of the ATLAS detector data analysis (the calibrations), complementing a parallel dissertation work that addresses the I/O bound issues.

The main goal of the work is to design, implement, validate and evaluate an improved and more robust data analysis task which involves tuning the performance of the kinematical reconstruction of events within the framework used for data analysis in ATLAS, to run on computing heterogeneous platforms based on multi-core CPU devices coupled to PCI-E boards with many-core devices, such as the Intel Xeon Phi and/or the NVidia Fermi/Kepler GPU devices.

An experimental framework, GAMA, will be used to automate (i) the workload distribution among the available resources and (ii) the transparent data management across the physical distributed memory environment between the shared multi-core memory and the many-core device memory.

1. Context

The Large Hardron Collider (LHC) [1] is high-energy particle accelerator, built by the European Organization for Nuclear Research (CERN) [2]. The LHC is used by physicists to validate high-energy physics (HEP) theories by experiment. There are several experiments being conducted along the LHC, where detectors, with different configurations according to their experiment purposes, gather data relative to the head-on collision of particles, which is considered an event. This data is mostly related to the characteristics of the resultant particles of the collision, such as momentum, energy and mass. Millions of particle collisions occur each second at the LHC, generating a massive amounts of data by the detectors.

One of the most important experiments running on the LHC is ATLAS [3], which results from a collaboration of many research groups around the world. One of those groups is the Laboratório de Experimentação e Física Experimental de Partículas (LIP) [4], which, as the other groups, continuously performs analysis on the data gathered by the ATLAS detector. The ATLAS research groups compete against each other for being the first to publish relevant results, meaning that they want to analyze the most data possible within very strict time schedules.

An analysis consists of an application that executes a set of algorithms on the collected data, in order to extract valuable information, from the HEP perspective. Before being analyzed, the data is incrementally refined, where from each iteration results a different data format holding the information. Different formats are optimized for different purposes, but oriented by a write-once many-reads policy. The more refined the data is, the smaller its size [5].

Every analysis uses the ROOT framework [6]. It provides data structures and I/O functionality, oriented for some of the data formats, and statistical analysis and visualization of the application output. Moreover, it can build the skeleton of the analysis based on the input data, offers parallelization packages for shared and distributed memory systems and has libraries for linear algebra, numerical algorithms and other utilities.

Since the skeleton that ROOT creates for the analysis has some limitations, the LIP research group built one according to their specific needs, so that most of their analysis can take advantage of it. The objective is to accelerate the development of analysis, which must be as fast as possible to fit in the strict time schedules already mentioned. Each time an update must be done on a given analysis, there is little time to test and refine the implementation, causing the applications to be badly structured.

During the last year work has been done on one of the analysis, TTBAR_Dilep, on the scope of Parallel and Distributed Computing curricular unit, exploring the potential of the kinematical reconstruction of events on various platforms. The objective was to run the kinematical reconstruction as many times as possible, per each event processed, within a reasonable time frame, which allows to process many events in such a way that enough results are obtained. Running the reconstruction many times allows overcoming the experimental resolution of the ATLAS detector, by being able to chose the most precise reconstructions, leading to more accurate results and better physics. Approaches using shared and distributed memory systems, as well as heterogeneous systems with manycore accelerating devices (GPUs in this case), were tested and compared in that work [7].

Heterogeneous architectures offer a different approach than the traditional multicore architectures. An heterogeneous system combines the multicore processors with specialized manycore accelerator devices. They are constituted of smaller processing units, focused only on achieving the best performance possible on certain areas, as opposed to common CPUs. Their architecture strategy is oriented for massive data parallelism processing [8], offloading the CPU from such data intensive operations. Several manycore accelerating devices are available, ranging from the general purpose GPUs, the Intel® Many Integrated Core line, currently known as Intel® Xeon Phi, and Digital Signal Processors [9, 10, 11]. A heterogeneous system may have one or more accelerating devices of the same or different types.

Different accelerating devices have different architectures, which affect the programming and tuning of the code to run on a specific device. Moreover, when combining the execution of an application on accelerating devices and regular CPUs, the load balancing becomes an important issue to manage. To optimize the resource usage, it is important to efficiently divide the workload, in such a way that neither the said devices nor the CPU becomes stalled for long periods of time. Tuning the code for different devices, which usually implies a whole new programming paradigm, and ensuring a efficient load balancing between multicore and manycore processors can become a complex task for the programmers to manage.

To ease this complexity to the programmer several frameworks were developed, addressing different domains of this subject. They differ from some SDKs as the latter is usually oriented to develop code for specific accelerating devices (such as NVIDIA® CUDA™ [12]), than for the whole heterogeneous system. StarPU is a high-level system which schedules a graph of tasks onto a heterogeneous platform in runtime [13]. It hides the complexities of programming to an heterogeneous platform by managing the load distribution, which commonly is the most complex aspect to deal with on these platforms. The GPU And Multi-core Aware (GAMA) project is an alternative to StarPU being designed specifically for GPUs with CUDA™, and also addresses irregular applications [14, 15, 16]. OpenACC is also a high-level framework, similar to StarPU, which manages initialization, shutdown and work balance of CPU and various accelerating devices [17].

Neither ROOT nor the LIP framework offers the tools to run any portion of an application in heterogeneous systems with accelerating devices. As proved in last year work, the kinematical reconstruction was fastest when running on a GPU. Not much tuning of the code was performed, and many other approaches were not explored (as concurrently execute reconstructions on the CPU and GPU, and using other accelerating devices). Also, there are newer architectures of GPUs, namely Kepler [18], and other accelerating, as mentioned before, that could improve the overall performance of the application.

2. Objectives

This dissertation will be focused on tuning the performance of the kinematical reconstruction of events, within the framework used for data analysis in ATLAS, on heterogeneous platforms.

The work will be, at first, aimed towards obtaining a highly tuned implementation using GPUs as accelerating devices. Later, the Intel Xeon Phi will also be used as an accelerating device. The execution of the kinematic reconstructions will always occur concurrently on CPU and the accelerating devices, so the load balancing will also be addressed.

Different approaches to the problem will be approached, depending on the accelerating device architecture, where the performance of the application will be fine-tuned. Also, an hybrid implementation will also be tested, using the multicore and available many-core processors on the system. Later, an approach using the GAMA framework will be tested to assess if it provides reasonable performance while reducing the implementation time.

3. Methodology

In an early phase, the main objective is to research similar problems implemented on heterogeneous systems. This can be accomplished through literature search on key areas of the subject. Also, it is important to identify the current limitations of the implementation developed last year.

The second phase will start by tuning the current GPU implementation. The kinematical reconstruction will be optimized for the NVIDIA® Fermi architecture, and later for the Kepler architecture as well. If possible, an implementation for the Intel® Xeon Phi will also be performed. Finally, a simple implementation using the GAMA framework, using only the GPUs as accelerating devices, will also be developed. This will be the most time consuming stage because of the inherent complexity of profiling and tuning the code for each accelerating device, as well as the workload balancing between CPU and the said devices.

The third stage will be dedicated to testing the efficiency of each the implementation using two metrics: performance of the code and time required for the implementation and tuning of the code for each accelerating device. The development time is an important factor because of the strict deadlines that the LIP research group has to face, as explained in the Context section.

4. Timing

- **November 2012 to December 2012 :**

  Literature search for the state of the art and analysis of previous implementations on heterogeneous systems;

- **January 2013 to February 2013 :**

  Implementation and profiling of the application:

  1. in a **CPU+GPU**, using a Fermi GPU, environment with CUDA;
  2. in a **CPU+GPU**, using a Kepler GPU, environment with CUDA;
  3. in a **CPU+Xeon Phi** environment.

- **March 2013 to April 2013**

  Hand tune/optimize the kinematical reconstruction code for the heterogeneous platform;

- **May 2013**

  Perform a comparative evaluation of the:

  1. **CPU+GPU**, both using Fermi or Kepler GPUs, approach and the **CPU+Xeon Phi** alternative;

2. both **CPU+GPU** alternatives and the **CPU+Xeon Phi** approach against an alternative one using a framework to abstract the execution management in a heterogeneous platform (e.g. GAMA);
3. **June 2013**

   Writing up the dissertation.

References

[1] European Organization for Nuclear Research, "The Large Hadron Collider", November 2012.

[2] European Organization for Nuclear Research, "CERN European Organization for Nuclear Research", November 2012.

[3] European Organization for Nuclear Research, "ATLAS experiment", November 2012.

[4] Laboratório de Experimentação e Física Experimental de Partículas, "Laboratório de Experimentação e Física Experimental de Partículas", November 2012.

[5] C. Biscarat G. Brandt G. Duckeck P. van Gemmeren A. Peters RD. Schaffer, W. Bhimji and I. Vukotic, "IO performance of ATLAS data formats", October 2010.

[6] B. Bellenot O. Couet A. Naumann G. Ganis L. Moneta V. Vasilev A. Gheata P. Russo F. Rademakers, P. Canal and R. Brun, "ROOT", November 2012.

[7] A. Pereira and R. Silva, "Particle Collision Detection Application", July 2012.

[8] John L. Hennessy and David A. Patterson, *Computer Architecture, Fifth Edition: A Quantitative Approach*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 5th edition, 2011.

[9] NVIDIA Corporation, "What is GPU Computing?", http://www.nvidia.com/object/what-is-gpu-computing.html, [Online; accessed November 22, 2012].

[10] Intel Corporation, "Many Integrated Core (MIC) Architecture", http://www.intel.com/content/www/us/en/architecture-and-technology/many-integrated-core/intel-many-integrated-core-architecture.html, [Online; accessed November 20, 2012].

[11] Texas Instruments, "Digital Signal Processors", http://www.ti.com/lsds/ti/dsp/overview.page, [Online; accessed November 23, 2012].

[12] NVIDIA Corporation, "CUDA", http://www.nvidia.com/object/cuda_home_new.html, [Online; accessed November 22, 2012].

[13] Cédric Augonnet, Samuel Thibault, and Raymond Namyst, "StarPU: a Runtime System for Scheduling Tasks over Accelerator-Based Multicore Machines", Rapport de recherche RR-7240, INRIA, March 2010.

[14] João Barbosa, "Gama framework: Hardware aware scheduling in heterogeneous environments", Tech. Rep., Informatics Department, University of Minho, September 2012.

[15] Artur Mariano, Ricardo Alves, João Barbosa, Luís Paulo Santos, and Alberto Proença, "A (ir)regularity-aware task scheduler for heterogeneous platforms", in *Proceedings of the 2nd International Conference on High Performance Computing*, Kiev, October 2012, pp. 45–56.

[16] Ricardo D. Q. Alves, "Distributed shared memory on heterogeneous cpus+gpus platforms", Master's thesis, Universidade do Minho, October 2012.

[17] OpenACC Corporation, "OpenACC", November 2012, [Online; accessed November 20, 2012].

[18] NVIDIA Corporation, "Kepler Architecture", http://www.nvidia.com/object/nvidia-kepler.html, [Online; accessed November 23, 2012].

Appendix A. Signatures

(Student) André Martins Pereira

(Advisor) Alberto Proença

(Co-Advisor) António Onofre