Investigation Into the Use of Hardware Accelerators in Data Intensive Compute

CS310 Progress Report
David Richardson, 1314918

November 2015

This document details the progress made in the investigation into the use of hardware accelerators in data intensive compute. Section 1 reintroduces the project and its aims as well as giving a summary of the project's objectives. Section 2 identifies existing research in the project's problem domain. Section 3 details the selection of the benchmarks to be used in the project, as well as any progress to the successful benchmarking of the cluster without hardware accelerators. Section 4 reiterates the approach to project management in the project specification, outlining any necessary changes that have been brought to light through the work so far. Section 5 outlines further work and extensions to the project. Finally, Section 6 concludes by outlining the overall state of the project as well as reiterating the key points of the project's progression. More information about the specification of this project is available in the specification documentation in Appendix A.

1 Introduction

Hardware accelerators provide the ability to offload a set of compute instructions from the CPU onto specialised hardware, designed to perform the computation faster and more efficiently than the CPU itself. These accelerators generally take the forms of General Purpose GPUs (GPGPUs) or Many Integrated Core (MIC) co-processors. With these hardware accelerators being included in an ever-increasing number of compute nodes within data centres, the chance to use them is increasing. However, the amount of research into their use within the paradigm of data intensive compute is underwhelming, despite their possible gains in power efficiency **energy-efficient-gpu** and speed **accelerating-matrix-product**, **quantitative-finance-gpu** over their general CPU counterparts. The integration of these hardware accelerators into the compute phases of data intensive workloads, such as MapReduce jobs, could provide benefits such as a reduction in the total compute time and power consumed by a workload. Other possible benefits involve a reduced need to scale outwards to cope with the Tera- or Petabyte scale data sets that have come about from the data avalanche in areas such as bioinformatics **big-data-biocuration** These benefits are of interest to both academic and commercial applications, where it can reduce operational costs and reduce turn around time for compute workloads. Organisations such as that provide data-centric services such as Google or Facebook would also be able to enrich user experience with features that were not feasible due to slow compute times, also providing an increased value of service and profit.

1.1 Project Aims

The underlying aim for this project is to test the use of GPGPUs and MIC co-processors in data intensive workloads to determine if their integration has significant improvements in compute and power consumption versus a CPU-only implementation. Their use has value for scientific and commercial areas, where a reduction in compute time will generally lead to a reduction in operational costs. It will also benefit infrastructure management companies such as Amazon, by increasing the performance per Watt of their compute nodes.

1.2 Summary of Objectives

The project has two main objectives that were outlined in the project specficiation documentation:

1. To understand if current benchmarking suites are suitable for hardware accelerated data analytics clusters.
2. To determine if accelerators can be used within data analytics with little modification to current software stacks or algorithm implementations.

Where the notion of a 'suitable' benchmark is a benchmark that tests a variety of work loads, makes use of any present hardware accelerators, and can be scaled in input data set size.

2 Research Direction

With the project introduced, the main aims for its research and the project's objectives all discussed, the area of related research is now considered.

Research into the use of GPGPU and MIC co-processors within data intensive compute is limited at best, with very few technical reports or articles available.

2.1 Accelerating Breadth-First Search with Intel MIC Co-processors

Tao, Yutong, and Guang provide research into the application of the Intel MIC co-processor architecture to the Breadth-First Search (BFS) of a graph, a common data intensive compute workload. Their research considers both native and offload optimisations, outlining their optimisation procedures for both **mic-accelerate-bfs** The native solution involves performing the BFS entirely on the co-processor and the optimisation techniques involved the exploitation of thread- and data-level parallelism. The offload solution will partition the tasks within the workload as well optimise communications between CPU and co-processor. They found that a native solution could run up to 3.4x faster on two Intel Xeon Phi Knight's Corner than when run on two Intel Xeon E5-2670. The offload algorithm results in a speed up of up to 1.67x. The offload algorithm also gains performance on larger graph sizes.

3 Benchmarking Progress

With the nature of this project being mostly based around investigation and research, it is quite hard to measure its progress. However, it is possible to measure progress with regards to the timetable outlined in the project specification, where it lists the key phases to the project. The progress towards benchmark selection and execution is now to be discussed.

3.1 Benchmark Selection

There are a number of benchmarking suites available for use with compute clusters designed for the likes of data analytics or other data intensive compute workloads. For this project I will be selecting one benchmarking suite for use to compare the effect of the integration of hardware accelerators into them.

Through my investigation into these benchmarks, a few observations have been made:

1. Most benchmarking suites come with their own scalable data generators.
2. All benchmarking suites that have been considered are developed for MapReduce or similar compute workloads.
3. All benchmarking suites investigated have not been built with the consideration for the use hardware accelerators.

These observations can be used to conclude about objective 1 that was outlined in the project specification. This is that the current suite of benchmarks are not suitable for hardware accelerated data intensive compute clusters. This is due to the lack of consideration within the benchmarking suites, when developed, for the use of hardware accelerators like GPGPUs and MIC co-processors.

With this in mind, the benchmarking suites that were considered are now discussed and compared.

3.1.1 Graph500

The Graph 500 is an initiative to establishe a set of large-scale benchmarks for data intensive applications, being backed by both academia and industry experts **graph500-intro** At present, the Graph 500 benchmark has only one workload that can be split into two kernels: generating a graph from an edge list, and a breadth-first search of the generated graph **graph500-spec** The second kernel is measured in Traversed Edges per Second (TEPS), which provides a unit for comparison akin to LINPACK with Floating Point Operations per Second (FLOPS). The reference implementations provided by the Graph 500 organisation are written in C and are available in sequential, OpenMP, XMT and MPI.

3.1.2 BigDataBench

BigDataBench is a data analytics benchmarking suite created at the ICT, Chinese Academy of Sciences, with backing from industry partners such as Huawei. The benchmarks in this suite abandon typical sequential and multithreaded workloads, that would typically use OpenMP or similar libraries, for scale-out **big-data-bench-home** workloads that are designed to better represent the distributed nature of data analytics. The benchmarks themselves in this suite are derived from a common subset of ‘dwarf’ workloads, such as social network graph analysis or word multimedia analytics **dwarf-workloads-big-data**

3.1.3 Intel HiBench

3.1.4 Comparison and Conclusion

3.2 Benchmark Running

4 Project Management

4.1 Timetable

4.2 Risk Assessment

5 Further Work and Project Extensions

5.1 Further Work

5.2 Project Extensions

6 Conclusion

A Investigation Into the Use of Hardware Accelerators in Data Intensive Compute Specification

The following 12 pages consist of the original specification document as submitted to Tabula in Week 2 of Term 1, 2015

Investigation Into the Use of Hardware Accelerators in Data Intensive Compute

CS310 Project Specification

David Richardson, 1314918

October 2015

This document presents a project on the research into the use of hardware accelerators in the data intensive compute paradigm. Whilst hardware accelerators are pervasive within the compute intensive paradigm of parallel computing, the use of them in data intensive compute remains an area of little research. Section 1 of this document provides the motivation for this project. Section 2 gives background information on the two key areas; data intensive compute and hardware accelerators. Section 3 outlines the objectives for this project. Section 4 explains the methodologies to be used throughout the project. Section 5 will discuss how the project will be managed. Section 6 will outline any legal, ethical, social or professional issues that this project will cover. Finally, Section 7 will conclude this document.

1 Introduction

Data analytics and the paradigm of data intensive compute are areas of parallel computing that are growing at enormous rates. The data avalanche in recent years has enabled ever-deepening study and research into areas such as biology, as well as commercial use such as Google's Knowledge Graph. The increase in compute power required to cope with this data avalanche is usually met with a scale-outwards approach, resulting in data centres with thousands of compute nodes. The research into the use of hardware accelerators here is underwhelming, and most software stacks do not have suitable provisions or APIs for their use. The applications of hardware accelerators could reduce program execution times, as well as increase power efficiency, being of interest to both academic and commercial applications. End users of services such as

Google or Facebook could benefit from this due to new metadata obtained from previously slow computations.

1.1 Project Aims

The underlying aim for this project is to investigate the use of hardware accelerators within data intensive compute. Their use within data intensive compute has value for both scientific and commercial areas. For example, warehouse scale compute facilities or data centres could drastically reduce the power requirements for the same computational capabilities. Results of compute jobs could be determined faster due to a reduction in compute time, to the benefit of scientific researchers.

2 Background

Having discussed the aims of my project, I will now go on to address the background knowledge required for this project. This will include explanations of data intensive computing, hardware accelerators, the Chiron data analytics cluster and BigDataBench.

2.1 Data Intensive Compute

Data intensive computing is an emerging computing paradigm [1] designed to deal with the processing of petabyte-scale datasets. It combines high performance computation, massive data storage, high bandwidth access, and high-speed local and wide area networking [2] to address problems that were previously thought to be infeasible or impossible [3] because the compute time would have previously been too long. The paradigm is dominated by the use of distributed computing clusters, and cloud computing has facilitated its use on a large scale [3]. Data intensive compute typically uses Apache's Hadoop, an implementation of Google's MapReduce distributed batch computing white paper [4]. Other batch computing implementations are also available, such as LexisNexis' High Performance Computing Cluster (HPCC). Other compute approaches to the paradigm are also available such as Apache Storm, which allows for realtime data stream computation [5], and Apache Spark allows for in-memory data compute [6]. These different approaches are usually combined into a software stack to allow for a comprehensive suite of tools for data analysis.

The use of data intensive compute is diverse, ranging from the sequencing of genomes through genomics [7] to gain understanding of how genetic traits can alter suceptibility to diseases, to the analysis of social networks to produce further information about people and their connections to others to create a more personalised experience.

2.2 Hardware Accelerators

Hardware accelerators are specialised computer system components designed to complement the general purpose CPU within the system. These accelerators provide a wide range of benefits over a general purpose CPU architecture, such as x86. These benefits include increased power efficiency [8], measured as FLOPS/Watt, as well as significantly increased performance with specific workload types. An example of this is matrix product calculations [9]. These characteristics has resulted in their uptake within the compute-intensive workloads, such as those in computational physics [10] and quantitative finance [11]. The most common types of hardware accelerators in use today are general-purpose GPU (GPGPU) and many integrated core (MIC) co-processors. Some examples of these are the Nvidia Tesla GPGPUs and the Intel MIC Xeon Phi co-processors. Other accelerator types are also available, such as Fully Programmable Gate Array (FPGA) cards from Altera.

Hardware accelerators have many benefits, but they can be underutilised if present in a system. This is usually a result of the accelerators being an afterthought in development or they are being used in workloads that they are not suited for. This can waste power, reducing their efficiency and overal effectiveness as part of a compute system.

2.3 Chiron

Chiron is a compute cluster at the University of Warwick designed for data intensive compute [12]. It provides the capability to perform traditional MapReduce computation, as well as data streaming and in-memory analytics. It also has 2x Nvidia Tesla K40 GPGPU nodes, 2x Intel Xeon Phis nodes, and 2x Nallatech 395 FPGA nodes. The cluster is configured using

Apache YARN as a base, with the Hadoop File System (HDFS) for a distributed filesystem. MapR Hadoop is then used along with Apache Storm for bulk data analytics and data streaming, respectively.

2.4 BigDataBench

BigDataBench is a data analytics benchmarking suite created at the ICT, Chinese Academy of Sciences, along with industry partners such as Huawei. The benchmarks abandon typical sequential and multithreaded workloads for scale-out [13]. workloads that are designed to better represent the distributed nature of data analytics.

The benchmarks themselves are designed around a set of commonly used workloads that represent common use cases of data analytics. Some examples of these are social network and search engine graph analysis, multimedia analytics and bioinformatics. The benchmarks are also implemented with different systems to test as broad a range of a system as possible. These implementations range from Apache Hadoop or Spark, to MySQL, to C-based programs that use MPI [13]. The suite also comes with its own data generation tool, designed to create huge data sets of different types under controllable generation rates [14].

3 Objectives

The objectives for my project are two-fold:

1. To understand if current benchmarking suites are suitable for hardware accelerated data analytics clusters.
2. To determine if accelerators can be used within data analytics with little modification to current software stacks or algorithm implementations.

For objective 1 it is important to define what the notion of suitability is. For the purpose of this project, a benchmarking suite is suitable if it meets the following criteria:

- The suite tests a variety of workloads with differing characteristics.
- The workloads may make use of accelerators if they are present in the system.
- The size the input data set for the workloads can be varied or scaled.

4 Methodology

My methodology for this project can be split into two parts: research and software development.

My software development methodology will be of a test-driven nature. This will allow for thorough testing of any (re)implementation of data analytics algorithms. The methodology will also be of an agile nature, allowing for changes to be made in the project during any software development stages that may arise [15]. The specific methodology to be used here will be Extreme Programming (XP) [15], allowing for the interleaving of development and testing of any software that is required. XP does require that the customer be involved every few weeks in the development cycle, but as this project is focused on research this is not feasible. Meetings shall also be set up during development on a regular basis with the project supervisor to make the most of this agile methodology.

For my research, I will be focusing on the Apache Hadoop software stack, which makes extensive use of the Java programming language though has support for other languages such as C++ [16]. I will also focus on GPGPU and MIC co-processors for the hardware accelerators in this project. These accelerators typically use C/C++ for programming, which may necessitate the use of the Java Native Interface (JNI) to execute any algorithm implementations designed for these accelerators.

My research methodology will start by selecting industry recognised benchmark(s) such as the BigDataBench benchmark for study. These benchmarks will have their data analytics workloads executed on the Chiron data analytics cluster to form a baseline for comparison. From there, I will attempt to make use of GPGPU and MIC accelerators within these benchmarks. From there I will benchmark the accelerated algorithms with the same data input sizes. Finally, I will compare the different benchmarks and draw conclusions with regards to the original objectives stated in section 3.

5 Project Management

Now that the discussion of the project's aims, objectives, requirements and methodologies has been completed, attention is now to be given to the project management aspect of the project. More specifically, this section will discuss the project's timetable, the resources required and give a risk assessment of the project, providing insight into any risk mitigations or actions to be taken.

5.1 Timetable

Figure 1 provides an overview of the project timeline. It includes the official project deadlines, as well as outlines the core stages of the project: benchmark selection, benchmark testing and analysis, benchmark migration, and accelerated benchmark testing and analysis. The benchmark migration stage may also contain a sub-stage for software development which will run at the same time. The timetable also provides task dependencies in the form of directional arrows between stages.

- Benchmark selection: 5th October 2015 to 1st November 2015
- Benchmark testing and analysis: 2nd November 2015 to 30th November 2015
- Benchmark migration: 1st December 2015 to 31st January 2016
- Accelerated benchmark testing and analysis: 1st February 2016 to 14th February 2016
- Project specification write up: 5th October 2015 to 14th October 2015
- Project specification submission: 15th October 2015
- Progress report write up: 15th November 2015 to 28th November 2015
- Progress report submission: 30th November 2015
- Presentation preparation: 8th February 2016 to 7th March 2016
- Presentation: 7-9th March 2016
- Final report write up: 10th January 2016 - 21st April 2016
- Final report submission: 28th April 2016

5.2 Tools & Resources

Third party tools will be used wherever possible to speed up the process of research as well as its documentation. This will include:

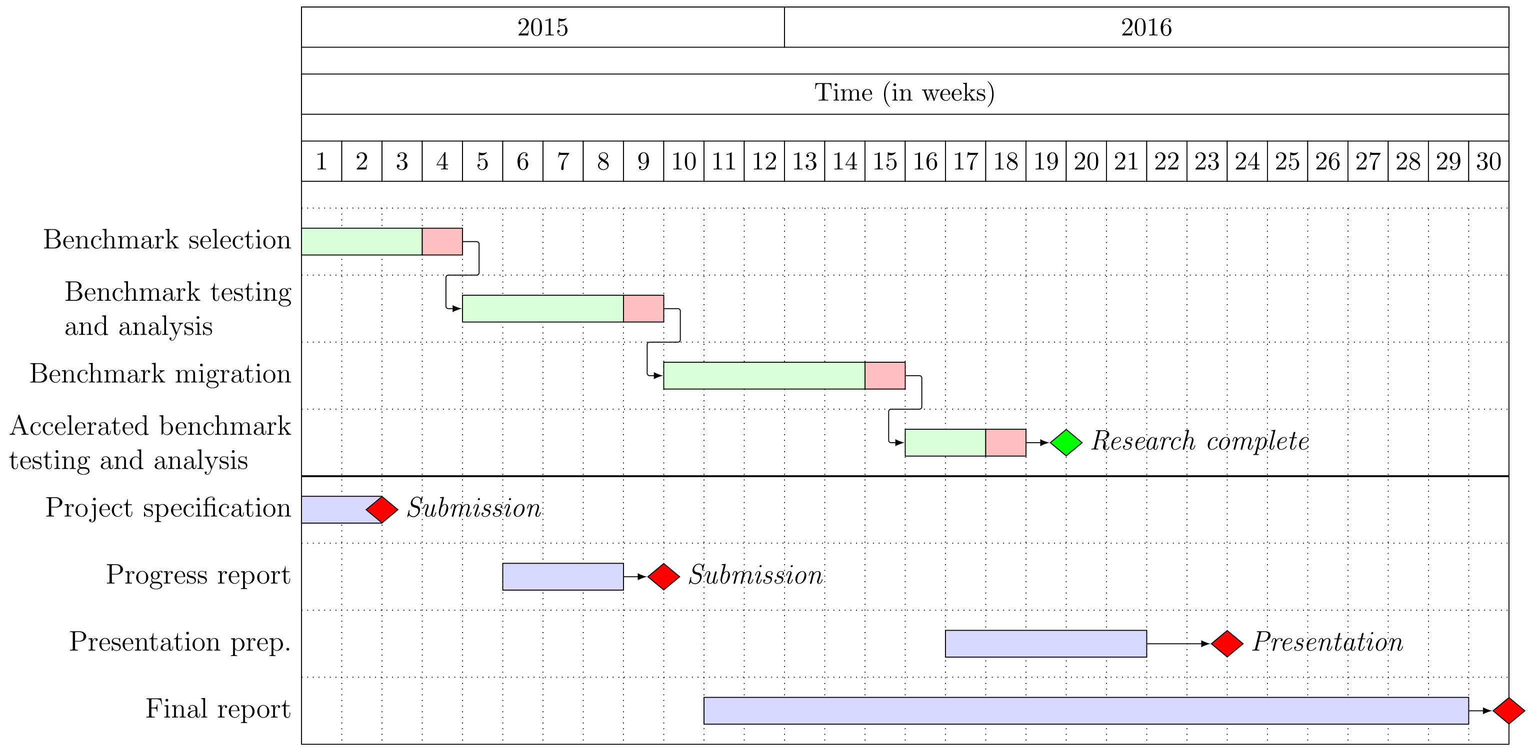

Figure 1: Project timetable from week 1 term 1 to week 1 term 3

Git VCS
: This version control system, along with GitHub, will provide a mechanism of recording previous and current versions of the project in a centralised location.

Nvidia CUDA Toolkit
: The Nvidia CUDA toolkit will be a vital tool for any development of data analytics algorithms for use with Nvidia Tesla GPGPUs. It provides a wide range of language bindings, giving the ability to use accelerators with ease in the Apache data analytics software stack.

IntelliJ IDEA
: This development IDE will prove to be very beneificial during the migration of benchmarks to accelerators. It has many language plugins as well as native test integration, aiding the test driven development methodology previously discussed.

LaTeX
: LaTeX is to be used as the means to produce the documentation for the project. It provides a way to create professional looking documents with easy, whilst also being powerful enough to display any extensive findings without the need for more 3rd party tools.

Along with these tools, the following resources will be crucial to perform the research:

Chiron
: Chiron is an Apache YARN-based data analytics cluster with accessible GPGPU and MIC accelerators. This is useful because it will allow the benchmarking of accelerated and un-accelerated programs on the same cluster.

5.3 Risk Assessment

Figure 2 provides an overview of possible risks throughout the project, as well as their severities and any possible actions that could be taken to mitigate or prevent them from occurring.

Risk	Severity	Likelihood	Mitigating Action(s)
Chiron unavailable	Severe	0.01%	Locate suitable replacement for use in testing — replacement should have similar feature set to Chiron.
Benchmark code unavailable	Severe	5%	Check internet archives for possible location of older version. Find other suitable benchmarks.
Networking failure	Moderate-Severe	5%	Temporarily locate to different area to use a different network.
Project leader falling ill	Moderate	10%	Do work that can be done without further risk to health.

Figure 2: Risk matrix that associates possible risks with severity and mitigating actions

6 Legal, Ethical, Social and Professional Issues

For the project there are a few professional and legal issues that must be addressed. There are, however, no social or ethical issues to be discussed.

6.1 Professional Issues

The project must be completed to a professional standard to ensure it can be extended in further research as well as used commercially. This will involve thorough testing of any new code, along with complete documentation through code commenting. Project documentation should also be of a high standard.

6.2 Legal Issues

The project will be utilising free, open-source software (FOSS) under varying licenses such as the Apache 2 license. Any software development libraries used within the project, such as the Nvidia CUDA toolkit, will also be under their own license agreements. These agreements will need to be adhered to.

7 Conclusion

The underpinning aim of this project is to investigate the use of hardware accelerators within data intensive compute. Acting towards this goal, a set of methodologies, management practices and test plans have been established and adopted to best attain the achievement of the stated goal. Furthermore, the refinement of the defined requirements is to be considered throughout the early stages of the project.

The progress towards the goals of this project will be documented in the next deliverable; the progress review. This document will be completed for and available on the 30th November 2015.

References

[1] Richard T. Kouzes, Gordon A. Anderson, Stephen T. Elbert, et al. "The changing paradigm of data intensive computing". In: *IEEE Computer* (Sept. 2009).

[2] Mario Cannataro, Domenico Talia, and Pradip K. Srimani. "Parallel data intensive computing in scientific and commercial applications". In: *Parallel Computing* (May 2002).

[3] Borko Furht and Armando Escalante. *Handbook of Cloud Computing*. Springer, 2010.

[4] EMC Education Services. *Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data*. John Wiley & Sons, Mar. 2015.

[5] The Apache Software Foundation. *Apache Storm Homepage*. June 2015. URL: `https://storm.apache.org/`.

[6] The Apache Software Foundation. *Apache Spark Homepage*. Oct. 2015. URL: `https://spark.apache.org/`.

[7] Aisling O'Driscoll, Jurate Daugelaite, and Roy D. Sleator. "'Big data', Hadoop and cloud computing in genomics". In: *Journal of Biomedical Informatics* 42 (Oct. 2013).

[8] S. Huang, S. Xiao, and W. Feng. "On the energy efficiency of graphics processing units for scientific computing". In: *IEEE International Symposium on Parallel & Distribued Processing* (May 2009).

[9] F. Bensaali, A. Amira, and A. Bouridane. "Accelerating matrix product on reconfigurable hardware for image processing applications". In: *Circuits, Devices and Systems, IEE Proceedings* 152 (June 2005).

[10] Ari Harju, Topi Siro, Filippo Federici Canova abnd Samuli Hakala, et al. "Computational Physics on Graphics Processing Units". In: (2013).

[11] Jr. Gerald A. Hanwerck. *Accelerating Quantitative Financial Computing with CUDA and GPUs*. Mar. 2013. URL: `http://on-demand.gputechconf.com/gtc/2013/presentations/S3373-Accelerating-Quantitative-Financial-Computing.pdf`.

[12] Warwick Institute for the Science of Cities. "Chiron: Data Intensive Computing for Warwick". Original slides detailing Chiron. 2015.

[13] Chinese Academy of Sciences ICT. *BigDataBench Homepage*. Oct. 2015. URL: `http://prof.ict.ac.cn/BigDataBench/`.

[14] Zijian Ming, Chunjie Luo, Wanling Gao, et al. "BDGS: A Scalable Big Data Generator Suite in Big Data Benchmarking". In: (2013).

[15] Ian Sommerville. *Software Engineering*. Pearson, Aug. 2015.

[16] The Apache Software Foundation. *Apache Hadoop MapReduce Tutorial.* June 2015. URL: http://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html#Overview.