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 com-

pute 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 Graph500 is an initiative to establish a set of large-scale benchmarks for data intensive applications, being backed by both academia and industry experts **graph500-intro** At present, the Graph500 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 Graph500 organisation are written in C and are available in sequential, OpenMP, XMT and MPI **graph500-reference-impl**

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** These benchmarks are implemented using different technologies ranging from Apache Hadoop or Spark, to MySQL and C-based programs that use MPI for inter-node communications **big-data-bench-home**

3.1.3 Intel HiBench

Intel's HiBench Hadoop benchmarking suite is a suite of 10 hadoop-based MapReduce benchmarks that are both synthetic micro-benchmarks and real-world applications **hibench-techreport** It uses the following 5 characteristings when benchmarking a system:

- Job running time.
- Number of tasks per minute or job throughput.
- HDFS bandwidth.
- Utilisation of system resources like CPU, Memory and I/O.
- Data access patterns

The suite itself only provides implementations to use Apache's Hadoop framework. The workloads cover web search, machine learning and analytical querying on large data sets. The micro-benchmarks cover basic jobs such as data sorting or extraction of information about a large data set. The suite also includes a benchmark to help determine the aggregated bandwidth that is delivered by HDFS **hibench-techreport-2**

3.1.4 Comparison and Conclusion

With all the benchmarks considered for this project outlined above, they are compared and a suite will be selected.

Whilst the Graph500 benchmark is backed by both industry leaders and academics alike, its suite contains only one benchmark that is not wholly representative of data intensive workloads. Its implementation in C and using MPI would make the use of hardware accelerators easier on more traditional compute oriented clusters. However, due to Chiron's architecture and chosen software stack, the use of MPI and/or OpenMP would not suit MapReduce and thus the suite will not be used for this project.

BigDataBench and Intel HiBench both provide extensive suites of benchmarks that can be separated into micro-benchmarks and workloads that are representative of what you would find in use in the real world. BigDataBench has the overall larger number of benchmarks available when compared to HiBench, and the benchmarking suite will also test more than just offline data analytics. It doesn't, however, provide the in-depth characterisation of the system that HiBench provides. With these considerations, Intel HiBench will be used for this project based upon its MapReduce workloads and in-depth benchmark reporting. It is also worth noting that all three benchmarks considered were not developed with the prospect of use with hardware accelerators.

3.2 Benchmark Running

With the benchmark suite having been selected for the project in section 3.1, they are now being used to benchmark Chiron without the use of hardware accelerators. With this process, there were a few issues found with the lack of software libraries and with the configuration of Chiron. These issues were reported to the Centre of Scientific Computing and were resolved, with little effect to the timetabling of the project due to the provisioning of overrun buffers in the task durations.

The benchmarking process has also been complicated with the realisation of Chiron's HDFS partitioning. HDFS is partitioned into an SSD partition and a HDD partition, resulting in differing sizes as well as read/write speeds. Due to the nature of data intensive compute being bounded by the speed of I/O, it is not beyond reason that the choice of storage medium would have an effect on results. This can be shown by difference in data access speeds for HDDs using SATA III and SSDs using 4x PCI-E 3.0 lanes: 31.56Gbps for PCI-E **understanding-pcie** and 6.0Gbps for SATA III **sata-3-standard** This results in SATA III being 5.26x slower than M.2 and its 4 PCI-E 3.0 lanes. On top of this, the node interconnects in Chiron are Mellanox InfiniBand with a maximum throughput of 56Gbps **mellanox-infiniband-manual** showing

that the network will not bottle neck I/O operations and the bottle neck in fact resides with the chosen storage media.

4 Project Management

Having discussed the progress made in the project, attention is now turned to the project's management as well as where any changes to the project timetable or risk assessment will be considered.

4.1 Timetable

With Chiron remaining in a pre-production state, it is not without configuration errors and has a lack of user documentation. These issues will, as a result, add possible delays to the project as it moves forwards and the use of untested system components starts to occur. Fortunately the project timetable as given in the specification documentation has some contingencies built into it, in the form of task padding zones marked in red, as well as extra amount of time that allows the project to overrun without issue. Figure 1 shows where the project is currently, with any complete tasks in violet. The overrun allowances for benchmark testing and accelerated benchmark testing and analysis have been extended, as well.

4.2 Risk Assessment

In an ideal world, Chiron would have had user documentation generated and have been tested thoroughly. However, due to it being in pre-production mode, it is necessary to add the following risks to the risk matrix provided in the project specification documentation:

- A confuration error with Chiron's software stack.
- Software required to run a benchmark is not installed.

The amended risk matrix, with the risks mentioned above, is shown in figure 2.

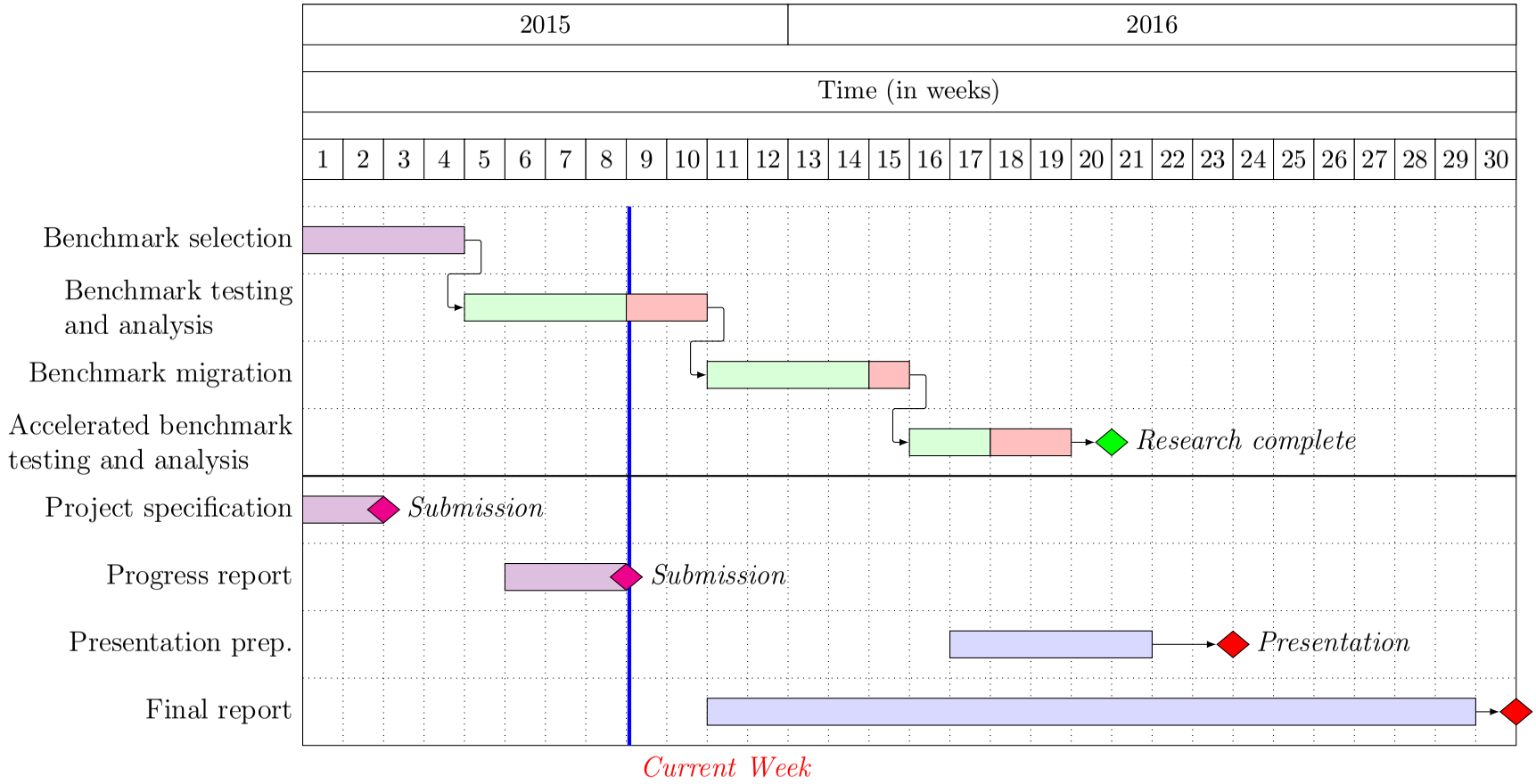

Figure 1: Revised project timetable from week 1 term 1 to week 1 term 3

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.
Configuration error with Chiron	Moderate	15%	Report the bug using the Centre of Scientific Computing's BugZilla bug tracker. Attempt other work that doesn't depend on that particular software stack.
Required software for benchmark not installed	Moderate-Severe	5%	Report the missing software to the Centre of Scientific Computing. In event of no resolution, another benchmark will be selected.

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

5 Further Work and Project Extensions

5.1 Further Work

Benchmark suite selection for this project has been finalised, and the execution of those benchmarks is underway. With this in mind, there are two remaining tasks to be completed: Benchmark migration and accelerated benchmark testing and analysis. This section will detail the remaining tasks that are to be fulfilled as part of the project's completion.

5.1.1 Benchmark Migration

The next step after the completion of benchmarking without accelerated codes is to then port these codes to the hardware accelerator platforms outlined within the project's specification documentation. This will involve identifying areas that are most suitable for application to the accelerator's architecture. After this identification process has finished, the accelerator's API calls will be injected into the codes and this code tested. Once the code has been migrated to the accelerator, re-integration into the MapReduce model will take place.

5.1.2 Accelerated Benchmark Testing and Analysis

After benchmark migration has been completed, a similar approach to the benchmarking procedure for unmodified codes will be undertaken. This will involve the execution of the accelerated benchmarks using both solid state and hard disk storage options. The input size for these benchmarks will also be varied as to provide an idea of how the solution(s) scale with data set size. This scaling may also show any points where data communications may oversaturate the accelerator nodes and overall reduce performance.

5.2 Project Extensions

Whilst this project is in an area of research, it is possible that is can be put into commercial use. An example of this would be to highlight areas for migration to hardware accelerators for existing data intensive programs. The extensions of this project reflect some of the countless possibilities in which it could be used.

Additional Accelerator Types

The research in this project is aimed only at GPGPU and MIC coprocessor hardware accelerators, but other types such as Fully Programmable Gate Array (FPGA) or Application-specific Integrated Circuit (ASIC) accelerators could be used as well. These have the benefit of being highly optimised for a specific task, although this also acts as a limitation as ASICs cannot be reprogrammed to perform another type of workload, and FPGAs have a high development overhead associated with long program compilation times and difficulty with debugging.

Infrastructure Design

This project's outcome could inform infrastructure providers such as Amazon or Google about the benefits to installing hardware accelerators into their compute nodes. This could reduce operational costs through the increased power efficiency of hardware accelerators. Compute time could also be drastically reduced, thus the number of users of the service could increase. Finally, the total number of compute nodes required for the same performance could drop, resulting in a reduced investment when purchasing infrastructure.

Additional Workload Testing

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