Hardware-accelerated similarity search indices on the Adapteva Parallella-16 reconfigurable computing machine

Erik Regla (Student)
Rodrigo Paredes (Advisor)
Civil Computer Engineering
University of Talca

September 11, 2017

1 Proposal description

1.1 Project's context

With the current advances in transistor technology, the design of 3D layered silicon chips and 5nm lithography, Moore's Law is now a problem as now is harder to push the hardware limits -for single chips- at the same rate as ten years ago[2, 19]. As stated in an 2007 technical report made by Altera Corporation: *"For most of the microprocessor's history, application demands have risen in response to processor improvements, allowing processors to stay ahead of demand. In the last few years, however, the situation has changed. High-performance computing (HPC) applications are now demanding more than processors alone can deliver, creating a technology gap between demand and performance."*[4]

This *Technology gap* is nowadays tackled by the using of heterogeneous computer architectures, being General Purpose GPU Computing (GPGPU) one of the most recurrent solutions due to the ease of programming and solution design provided by hardware-specific programming languages like Nvidia CUDA [5]. But even promising solutions like GPGPU have their shortcomings, as energy consumption and branch divergence heavily degrades performance on control-flow intensive algorithms -which are the most common- [26].

There is a growing interest on *high-performance and low-power custom computing machines* which aim to design *Application-specific integrated circuits* (ASICs) in order to solve certain computational problems. But as ASIC design, prototyping,

and implementation is very costly, *Field Programable Gate Arrays* (FPGAs) has been proven as a cost-effective solution to implement those designs and interconnect them with our current platforms to form reconfigurable computing architectures [6, 20, 3] being only limited by their power-budget while offering a scalable compute model for certain problems even after Moore's law end.

1.2 Problem definition

One of the many approaches to perform similarity searches is to perform k-nn or range queries on permutant-based indices, which abstract the dataset dimensionality and the cost of the object-object distance calculation. To perform a search on this index, a permutation is generated for the query object and then compared to the whole dataset under the premise that computing distance between two permutations is faster than computing the full distance between the two elements. After their distances are computed, a subset is selected under a certain criteria given by the query nature and the results are filtered later in order to answer the query. [8, 7]

As permutations are abstractions of the intrinsic dataset dimension, as we reduce the permutation/dimensionality ratio, the results become inaccurate, on the other hand, if we raise the ratio to increase accuracy we end processing the whole dataset rendering the approach useless as it's more time consuming than the original problem.

This behaviour makes similarity search indices a perfect test bench for its implementation on a reconfigurable computing device[27] such as the Adapteva Parallella-16, a *System on chip* (SoC) based on the Xilinx All Programmable Zynq7000 Series SoC which packs a Dual-core 32-bit ARM Cortex-A9 host controller and a Artix-7 FPGA [22, 9]. The board also has an Epiphany III multicore accelerator coprocessor, interconnected with the host-controller though an FPGA designed ASIC in order to be used as a low-power high-performance heterogeneous computing platform.

The main problem with reconfigurable computing is the complexity of circuit design [25]. Independently of the algorithm being ported to an FPGA implementation, there is not automated way to use the same source code used on the Processing System (PS) version of the problem in the Programmable Logic (PL). As there are some tools developed for both FPGA manufacturers and 3rd parties to design circuits using the C language, they lack of precision and many considerations and "compile" optimizations must be performed in order to successfully port certain algorithms[25], such as instruction pipelines, read/write syncronization, clock gating[11], etc [13, 11, 10, 18].

1.3 Current works

Alvarez [25] implemented a positioning system based on video streams using a Dilligent Zedboard [1]. In this implementation, an FPGA accelerator were used to increase the image processing throughput by interfacing the camera input to a series of filters by using an AXI4 interface. The FPGA implementation increased eight times the original processing rate for 1080p video streams despite not being performed any code optimizations, correct pragmas, or hardware replication.[14].

Rodriges [24] developed an operating infrastructure on a Parallella-16 board, but in this work the FPGA is used in the same way as the original developers intended, to interconnect the components of the board in a custom way [22], but no acceleration hardware were designed at this point.

Olofsson [21] explained how to create FPGA accelerators but the examples provided in their repository are not functional. Olofsson also states that building accelerators on FPGAs is a time-consuming work with a high *mistake cost* [21].

1.4 Proposed solution

In order to accelerate a similarity search engine, we will develop a naive implementation of a permutant-based metric space index without any major optimizations to study the feasibility of porting each subroutines involved based on their potential parallelism.

Some of the target routines to evaluate include incremental sorting routines[23], permutant generation and comparison[8] and massive distance calculations [7, 8], as those operations are expected to be executed in a pipelined fashion.

After the study, we will port such algorithms as Intellectual Properties (IPs) though the use of Xilinx Vivado HLS to be validated and then implemented on the FGPA as accelerators for further interconnect with the host-controller [25, 17, 15, 12, 16, 14]. Then a performance comparison will be executed in order to determine the success of the solution and to prove if Olofsson statements regarding FPGA computing [21] on the Adapteva Parallella platform were correct or not.

2 Objectives

Main objective

- Study the feasibility of implementing hardware-based accelerators for the Adapteva Parallella SoC.

Specific objectives

- Specify requirements and considerations to be accounted when porting general purpose algorithms to FPGAs.
- Study and implement a PL-PS data sharing sharing solution.
- Develop a functional FPGA-based hardware-accelerator prototype for a subset of routines involved on approximate similarity search.
- Deliver a solid guide to serve as a starting point to future computer scientists with little or no knowledge about hardware design.

3 Scope

- During this work we will not create a framework to develop new algorithms on FPGAs.
- Also, we will not work on optimizations for the original versions of the tested routines. [1]
- This work is limited only to research about FPGA hardware design, and to compare and contrast both software and hardware implementations on a Adapteva Parallella-16 SoC.

4 Methodology

Milestone 1: "Approximate search algorithms and indices"

- Analyse previously developed hardware-accelerators.
- Research about resource sharing methods and techniques for the proposed architecture.
- Implement a simple hardware-accelerator IP Core on the FPGA.
- Study possible problems which could arise when porting common algorithms on custom hardware.
- Research about approximate search indices for metric spaces.
- Research about workarounds for high-dimensionality metric space datasets.

[1] This is because we expect to obtain a performance boost by implementing the same algorithms as ASICs.

- Implement a permutant-based index and query algorithm.
- Analyse the behaviour of the implemented solution and identify potential targets for hardware-acceleration

Milestone 2: "Hardware-accelerated index implementation"

- Research about hardware prototyping.
- Research about hardware-software interconnection techniques and technologies applicable to the target platform.
- Implement as IP Cores the selected code fragments.
- Implement an interconnection protocol for resource sharing between the two platforms.
- Detect possible bottlenecks or other problems derived from the interconnection between the Atrix-7 FPGA and the ARMv7 A9 processor.
- Implement a loadable bitstream for the Parallella board and design according kernel modules.
- Replace software solution with custom hardware solution.
- Benchmark hardware implemented solution and contrast it with software implementation.

5 Work plan

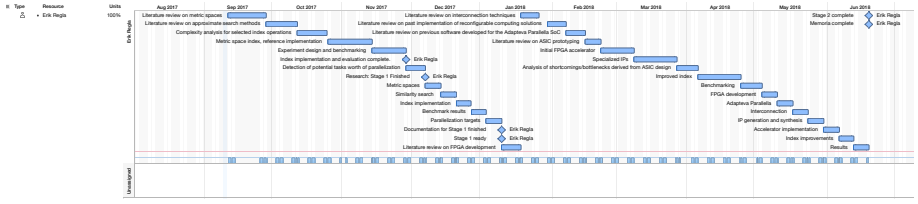

Figure 1: Work plan schedule

References

[1] Inc. Avnet. *ZedBoard (ZynqTM Evaluation and Development Hardware) User's Guide*, 1.1 edition, Aug 2012.

[2] R. Colwell. The chip design game at the end of moore's law. In *2013 IEEE Hot Chips 25 Symposium (HCS)*, pages 1–16, Aug 2013.

[3] K. Compton, S. Hauck, and Katherine Compton. An introduction to reconfigurable computing. *IEEE Computer*, 2000.

[4] Altera Corporation. White paper: Accelerating High-Performance Computing With FPGAs. Technical Report WP-01029-1.1, 2007.

[5] NVIDIA Corporation. NVIDIA Tesla V100 GPU Aarchitecture, The world's most advanced data center GPU. Technical Report WP-08608-001_v01, Jun 2017.

[6] G. Estrin. Reconfigurable computer origins: the ucla fixed-plus-variable (f+v) structure computer. *IEEE Annals of the History of Computing*, 24(4):3–9, Oct 2002.

[7] K. Figueroa and K. Frediksson. Speeding up permutation based indexing with indexing. In *2009 Second International Workshop on Similarity Search and Applications*, pages 107–114, Aug 2009.

[8] K. Figueroa and R. Paredes. Approximate direct and reverse nearest neighbor queries, and the k-nearest neighbor graph. In *2009 Second International Workshop on Similarity Search and Applications*, pages 91–98, Aug 2009.

[9] Linley Gwennap. Adapteva: More flops, less watts, 2011.

[10] Xilinx Inc. Achieving High Performance DDR3 Data Rates. Technical Report WP383 ver.1.2, Aug 2013.

[11] Xilinx Inc. Reducing Switching Power with Intelligent Clock Gating. Technical Report WP370 ver.1.4, Aug 2013.

[12] Xilinx Inc. *Zynq-7000 All Programmable SoC Software Developers Guide*, 12.0 edition, Sep 2015.

[13] Xilinx Inc. *AXI Memory Mapped to PCI Express (PCIe) Gen2 Product Guide*, 2.8 edition, Apr 2017.

[14] Xilinx Inc. *UltraFast High-Level Productivity Design Methodology Guide*, 2017.2 edition, June 2017.

[15] Xilinx Inc. *Vivado Design Suite Reference Guide*, 2017.1 edition, Apr 2017.

[16] Xilinx Inc. *Vivado Design Suite Tutorial*, 2017.1 edition, May 2017.

[17] Xilinx Inc. *Vivado Design Suite User Guide*, 2017.1 edition, Apr 2017.

[18] Xilinx Inc. *Zynq-7000 All Programmable SoC and 7 Series Devices Memory Interface Solutions*, 4.2 edition, Jul 2017.

[19] R. Colin Johnson. The limits of moore's law limits.

[20] B. Liu, D. Zydek, H. Selvaraj, and L. Gewali. Accelerating high performance computing applications: Using cpus, gpus, hybrid cpu/gpu, and fpgas. In *2012 13th International Conference on Parallel and Distributed Computing, Applications and Technologies*, pages 337–342, Dec 2012.

[21] Andreas Olofsson. Creating an fpga accelerator in 15 minutes.

[22] Andreas Olofsson, Tomas Nordström, and Zain-ul-Abdin. Kickstarting high-performance energy-efficient manycore architectures with epiphany. *CoRR*, abs/1412.5538, 2014.

[23] E. Regla and R. Paredes. Worst-case optimal incremental sorting. In *2015 34th International Conference of the Chilean Computer Science Society (SCCC)*, pages 1–6, Nov 2015.

[24] Sandro Rodrigues. Operating infrastructure for an fpga and arm system. Master's thesis, Universidade de Aveiro, 2015.

[25] Angel Alvarez Ruiz. Co-diseño hardware-software de un sistema de posicionamiento basado en procesado de video. Master's thesis, Universidad de Cantabria, 2016.

[26] J. Sartori and R. Kumar. Branch and data herding: Reducing control and memory divergence for error-tolerant gpu applications. *IEEE Transactions on Multimedia*, 15(2):279–290, Feb 2013.

[27] R. Uribe-Paredes, P. Valero-Lara, E. Arias, J. L. Sánchez, and D. Cazorla. Similarity search implementations for multi-core and many-core processors. In *2011 International Conference on High Performance Computing Simulation*, pages 656–663, July 2011.