ASIC prototyping for hardware-accelerated index operations for similarity search.

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

August 29, 2017

1 Proposal description

1.1 Project's context

Moore's law is now on decline as we're approaching to atomical scale of transistors and then to minimize the lithography of processing units will be impossible. This is a source of worry for many engineers because Moore's law statements are becoming hard to maintain with each passing day. This encourages the need of new computational models and architectures to solve some problems, for instance General Purpose GPU Computing (GPGPU) in which graphic cards are used as massively parallel processors to work with large amounts of data at the cost of a reduced, and somewhat limited instruction set.
There is a growing interest on High-performance and low-power custom computing machines implemented on FPGAs as they pose a flexible platform to implement custom algorithms in the form of combinatorial circuits. These solutions are only limited by their power-budget, 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.
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 acurracy we end computating the whole dataset rendering the approach useless as it's more time consuming than the original problem.

1.3 Current works

TODO

1.4 Proposed solution

In order to tacke this problem, we propose to port fragments of the routines involved on both indexing and searching procedures to an FPGA-based hardware-accelerator, in the hopes of reducing both compute time and energy consumption by taking advantage of the nature of the combinatorial circuits which could be implemented directly on hardware.
To test our solution we will implement them on a Adapteva Parallella board, an heterogeneous parallel SoC capable of running linux which embeds together a 16-core Epiphany III processor, an ARM A9-based host controller and a Zynq7000 Series FPGA, all of this in a single credit-card form factor.

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 scientist 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.
- This work is limited only to research about FPGA hardware design, and to compare and contrast both implementations.

4 Methodology

Milestone 1: "Approximate search algorithms and indices"

- Analise 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.
- Implement a permutant-based index and query algorithm.
- Analise the behiavour 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 techlologies applicable to the target platform.

- Implement as IP Cores the selected code fragments.
- Implement an interconection protocol for resource sharing between the two platforms.
- Detect possible bottlenecks or other problems derived from the interconection 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 constrast it with software implementation.

5 Work plan

Milestone 1: "Approximate search algorithms and indices"

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

Milestone 2: "Hardware-accelerated index implementation"

References