Accelerating Hamming Distance Comparisons for Locality Sensitive Hashing (LSH) using FPGAs

Martin Kaiser, Sarah Pilz, Florian Porrmann, Jens Hagemeyer and Mario Porrmann
CITEC, Bielefeld University
Bielefeld, Germany
{mkaiser, spilz, fporrmann, jhagemey, mporrmann}@cit-ec.uni-bielefeld.de

Jim Hogan
Queensland University of Technology
Australia
j.hogan@qut.edu.au

Extended Abstract

Today's bio-computing applications require a vast amount of computational power due to large data sets as well as the complexity of the used algorithms. Therefore, such applications greatly benefit from emerging technologies, which provide high-performance as well as improved energy efficiency. A promising approach for comparing DNA sequences is Locality Sensitive Hashing (LSH), where hash values are generated from subsequences and compared with each other, e. g., by calculating their hamming distances [1, 2]. In order to compare two bacterial genomes of $3 * 10^6$ BP, about 30 000 to 100 000 hash values are constructed, resulting in $9 * 10^8$ to $1 * 10^{10}$ hamming distance calculations. Due to the absence of data dependencies between the calculations, massive parallel architectures like GPGPUs and FPGAs are highly suitable for accelerating these computations.

This work uses an FPGA-based system architecture to accelerate the hamming distance comparisons as it is the most computationally intensive part of LSH. The algorithm has been implemented in a hardware description language (VHDL) and is optimized for highly parallel, stream-based processing on Xilinx FPGAs (Figure 1). Hash values of sequences A are stored locally on the FPGA while hash values of sequences B are streamed from the host to the FPGA. Supported lengths for the hash values are configurable between 32 bit and 8196 bit to cover a wide range of use cases. For hash values with a length of 512 bit up to 230 Hamming Processing Elements (HPE) can be implemented on a Xilinx Virtex-7 FPGA, resulting in a peak performance of 46 GH/s (Billion Hamming Distance Calculations per second). Multiple FPGA can be arranged in a daisy chain for passing streamed sequences to neighboring FPGA without the need to re-transferring them. This mitigates the data transfer bottleneck from the host to the FPGA, which would otherwise limit the practical performance to 10.3 GH/s.

For evaluation, the proposed design was compared to optimized implementations on the server platform RECS, which provides a modular microserver architecture for heterogeneous hardware, e. g. CPUs, GPUs and FPGAs and is capable of on-line measuring performance and power dissipation [4, 3]. The CPU implementation, written in C++ using SSE 4.2 and AVX2 instructions, achieved a performance of 0.8 GH/s and an energy efficiency of 19 million hashes per second per Watt (MH/s/Watt). An OpenCL-based implementation optimized for a NVIDIA P100 GPGPU is 13

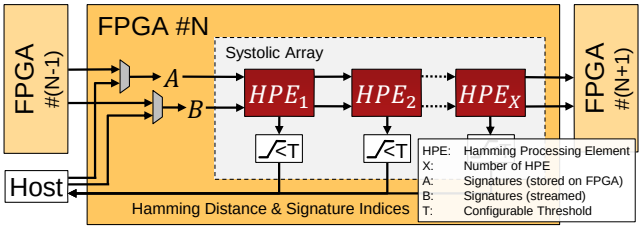

Figure 1: Scalable, streaming based system architecture for parallel hamming distance comparisons

times faster than the CPU implementation and achieves a 3.6 times better energy efficiency. Although the used Xilinx Virtex-7 FPGA on the RAPTOR platform [5] has been released five years ago, it performs as fast as the implementation on a state of the art GPGPU, but reaches a 1.7 times higher energy efficiency. The FPGA-performance scales with the number of used FPGAs, which can be explained through the increased number of processing elements, which process the data in parallel without the need to re-transfer data by the host.

Table 1: Evaluation results for Hamming Distance comparisons of hash values with a length of 512 bit

Platform	Architecture	Power Watt	Perf. GH/s	Energy Eff. MH/s/Watt
Intel E3-1226 v3	CPU	40	0.75	18.8
NVIDIA P100	GPGPU	150	10.1	67.6
Bielefeld Univ. RAPTOR	1x FPGA (Virtex-7)	90	10.3	114.4
Bielefeld Univ. RAPTOR	4x FPGA (Virtex-7)	195	41.2	211.3

The results show that FPGAs are highly suitable for accelerating hamming distance calculations, especially for large scale compute clusters, where energy efficiency is rather important. When using state of the art Xilinx UltraScale+ FPGAs, a speed-up by a factor of 6.5 and a 4.5 times higher energy efficiency is expected, compared to the currently employed FPGA hardware. Due to the generic design of the proposed system architecture, the applied distance metrics can be exchanged with little effort, to evaluate a wide range of different applications using LSH for bio-computing.

Acknowledgment

This work was funded as part of the Cluster of Excellence - Cognitive Interaction Technology *CITEC* (EXC 277), Bielefeld University and supported by the EU Horizon 2020 funded projects M2DC (Grant Agreement no. 688201) and LEGaTO (Grant Agreement no. 780681).

Keywords

Locality Sensitive Hashing, LSH, Hamming Distance Calculation, FPGA, Accelerator, Hardware-Acceleration, Biocomputing, Energy Efficiency, High-Performance Computing, HPC

1. REFERENCES

[1] L. Buckingham, J. M. Hogan, S. Geva, and W. Kelly. Locality-sensitive hashing for protein classification. In *Conferences in Research and Practice in Information Technology*, volume 158. Australian Computer Society, Inc, 2014.

[2] T. Chappell, S. Geva, and G. Zuccon. Approximate nearest-neighbour search with inverted signature slice lists. In *European Conference on Information Retrieval*, pages 147–158. Springer, 2015.

[3] M. Kaiser, R. Griessl, J. Hagemeyer, D. Jungewelter, F. Porrmann, S. Pilz, M. Porrmann, M. vor dem Berge, and S. Krupop. A reconfigurable heterogeneous microserver architecture for energy-efficient computing. In *Third International Workshop on Heterogeneous High-performance Reconfigurable Computing (H2RC'17)*, 2017.

[4] A. Oleksiak, M. Kierzynka, W. Piatek, G. Agosta, A. Barenghi, M. Porrmann, J. Hagemeyer, R. Griessl, J. Lachmair, M. Peykanu, L. Tigges, M. vor dem Berge, et al. M2DC – Modular Microserver DataCentre with heterogeneous hardware. *Microprocessors and Microsystems*, 52:117–130, 2017.

[5] M. Porrmann, J. Hagemeyer, C. Pohl, J. Romoth, and M. Strugholtz. Raptor–a scalable platform for rapid prototyping and fpga-based cluster computing. *Parallel Computing: From Multicores and GPU's to Petascale, Advances in Parallel Computing*, 19, 2010.