Virtual Screening on FPGA (demo)

Tom Vander Aa[1], Thomas J. Ashby[1], and Roel Wuyts[1]

ExaScience Life Lab at imec, Leuven, Belgium tom.vanderaa@imec.be

Abstract. Virtual Molecule Screening (VMS) is a computational technique used in drug discovery that uses machine learning to predict if a chemical compound is likely to bind to a drug target. Since many compounds need to be tested, virtual screening requires much computational power. In this demo we show the use of FPGA accelerators to speed up the virtual screening process. Thanks to some key benefits of FPGAs, we can obtain a 10x speed-up by using FPGAs for virtual screening, compared to using CPUs, and a similar performance as GPUs, both at a much lower power consumption.

Chemogenomics and Matrix Factorization In chemogenomics the key problem is the identification of candidate molecules that affect proteins associated with diseases. If this is done using machine learning techniques, the process is called compound-activity prediction. Bayesian Matrix Factorization (BMF) is a technique borrowed from recommender systems that has been successfully used for compound activity prediction. Thanks to the Bayesian approach, BPMF has been proven to be more robust to data-overfitting and Gibbs sampling makes the models feasible to compute. Another benefit, compared to for example deep learning, is that predictions from BMF models include confidence estimation, that can be used as a quality metric. In the European H2020 project ExCAPE (ExaScale Compound Activity Prediction) we have built large scale chemogenomics models using different machine learning techniques, such as Support Vector Machines (SVM), Deep Learning (DL) and Matrix Factorization. Amongst those, matrix factorization models perform very well because of the aforementioned *Bayesian benefits*.

Virtual Screening Once the models have been built, drug companies want to use them to evaluate millions of molecules in so-called virtual screens. Figure 1 shows a simplified view on the prediction flow for activity prediction. A molecule represented by its chemical fingerprint is fed in on the left. From the fingerprint an internal latent representation is computed and this latent representation is used to make predictions on one or more protein targets. The amount of computation depends on the number of molecules, the number of Gibbs samples, the size of the latent representation and the number of protein targets.

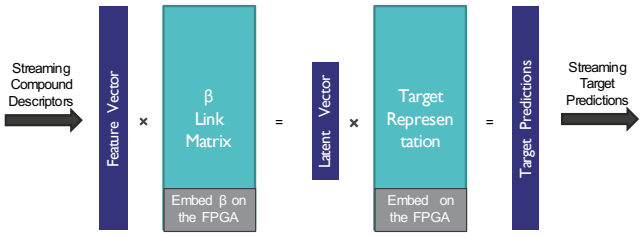

Fig. 1. Virtual Molecule Screening (VMS) Pipeline

FPGA acceleration Using Field-Programmable Gate Arrays (FPGAs) to accelerate large scale predictions has become popular in recent years in many domains, especially in combination with Neural Networks as is evident from the plethora of available software packages and publications on the topic. We have mapped the prediction flow as shown in Figure 1 on FPGA. The FPGA implementation is 10x faster than to the original CPU implementation and has a similar performance as GPUs. However, the FPGA consumes 3x less power than the GPU or CPU.

The key properties we needed to exploit to achieve this result are:

Parallelism: Compared to CPUs, FPGAs have a clock-speed that is 10x lower than CPUs. They compensate for this by providing many more parallel resources (DSP blocks, memory's, LUTs, ...). We were able to benefit from these extra resources by exploiting parallelism at many levels: inside the prediction pipeline, but also across proteins and molecules.

Code Complexity: Mapping code efficiently on FPGA is a time consuming tasks, even with help of high-level synthesis tools. Thanks to the simplicity of the code and the use of a code generator, we were able to reach good speed up with relatively little effort.

Memory Bandwidth: As with all accelerators, getting data in and out efficiently is key to good performance. In this case we achieved this by streaming the fingerprints in and predictions out in a linear fashion, and by storing the model itself in the FPGA on-chip memory beforehand.

Bit-Width Reduction: FPGAs deal much better with reduced bit-width fixed point numbers than with double or single floating point numbers. We were able to reduce the bit width of the input and output streams, and of the model itself from 64 bit floating point to 16bit fixed point, without a significant loss in prediction accuracy.

The Demo The demo will consist of a ZCU102 Evaluation Board running the VMS pipeline, a set of slides explaining the key concepts as detailed in this abstract and an accompanying video. The video is available at `https://vimeo.com/256141454`.

Acknowledgments The authors would like to acknowledge funding from the European Union's Horizon 2020 Research and Innovation programme under Grant Agreement no. 754337 (EuroEXA).