High-Performance Scalable Stream Computing with Multiple FPGAs

Antoniette Mondigo, Tomohiro Ueno, Daichi Tanaka, Kentaro Sano, and Satoru Yamamoto
Graduate School of Information Sciences, Tohoku University, Sendai, JAPAN
{apmondigo, ueno, tanaka, kentah, yamamoto}@caero.mech.tohoku.ac.jp

I. INTRODUCTION

High performance, flexibility, and power efficiency are among the necessary requirements in high-demand, compute-intensive solutions. Among the available computing platforms, field programmable gate arrays (FPGAs) are recently becoming attractive solutions due to their increased capacity of floating point processing, low power utilization, and application-specific customized hardware capability. In addition, FPGA's flexibility allows straightforward connection to other devices through any physical interface standard. This extends to direct FPGA interfacing, which highly suggests performance scalability. To scale the performance with multiple FPGAs, certain conditions must be considered, such as efficient parallel computing architecture, high-speed and low-latency communication, and sufficient bandwidth requirement. For high performance computations with a limited memory bandwidth, stream computing is promising because it allows processing of multiple operations through a deep pipeline with regulated memory access. Inter-FPGA communication could also be bandwidth-enhanced [1] alongside the high-speed serial I/Os for low-latency communication in modern FPGAs. Contributions of this work are (1) scalable architecture for deeply-pipelined stream computations; (2) evaluation with FPGA-based numerical computing cores; and (3) performance estimation for cascaded Intel Arria 10 FPGAs.

II. DESIGN AND RESULTS

We had created an FPGA cluster with 16 Intel Stratix V FPGAs [2] and now extending the work to a cluster composed of Intel Arria 10 FPGAs. This upgrade constitutes a number of improvements including a faster inter-communication full duplex bandwidth of 80 Gbps via QSFP ports from Stratix V's 10.31 Gbps. Each of the four host nodes have four FPGA boards and each FPGA is inter-connected in a 1D ring topology through the QSFP ports for stream computing. Inside each FPGA, a pipeline of stream processing elements (SPEs) for a custom computation is implemented. This computing pipeline takes an input stream from memory and outputs a data stream as the computation result for a certain number of time steps. For simplicity in the localization of computational data, we are implementing a configuration of one master with multiple slaves, where the master FPGA has an exclusive access to the external memories and cascades the data stream down to the slave FPGAs, as illustrated in Fig. 1. For evaluation purposes, Lattice Boltzmann Method (LBM) [2] and tsunami simulation [3] computing cores are used. To further increase throughput and maximize the use of available hardware resources, performance scaling is explored through spatial and temporal parallelism, by duplicating and deepening the pipeline of SPEs, respectively. As support for temporal parallelism between FPGAs, a hardware-based bandwidth compression scheme [1] is utilized to logically increase the communication bandwidth, thus, enabling spatially-paralleled computing cores to transfer more data streams.

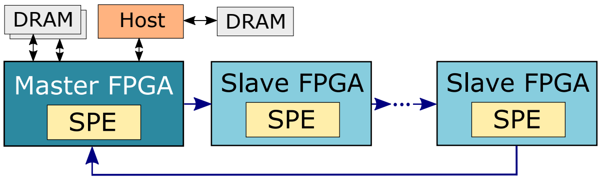

Fig. 1. 1D ring of master and slave FPGAs

A performance model is derived and used to estimate the peak performance. When a single computing pipeline (*N=1*) is used, the available bandwidth for memory and communication is sufficient, therefore, all streaming cycles are fully utilized. However, when the number of parallel cores are increased to two and four, stalls are necessary to synchronize the computations. For instance, in LBM, the stall ratio for *N=2* and *N=4* are 44% and 72%, respectively, due to insufficient communication bandwidth. This is reduced to 5% and 52% when bandwidth compression is applied with a compression ratio of 2.25. When the number of FPGAs are increased (*M>1*), the performance is scaled only within the bounds of available memory bandwidth due to the enhancement done by the bandwidth compression. Currently, we are implementing the 1D ring of 16 Intel Arria 10 FPGAs, in which evaluation of the sustained performance will follow. In our future work, we will extend the multiple FPGA architecture to form a 2D torus array of FPGAs, to further scale the performance.

REFERENCES

[1] T. Ueno, R. Ito, K. Sano, and S. Yamamoto, "Bandwidth compression of multiple numerical data streams for high performance custom computing," in *2014 IEEE 25th International Conference on Application-Specific Systems, Architectures and Processors*. IEEE, jun 2014, pp. 190–191. [Online]. Available: http://ieeexplore.ieee.org/document/6868660/

[2] K. Sano, T. Ueno, D. Tanaka, and S. Yamamoto, "High-Performance Fluid Simulation using Multiple FPGAs with Bandwidth-Compressed Links," in *H2RC 2016*, vol. 1691, 2016, pp. 21–27. [Online]. Available: https://h2rc.cse.sc.edu/papers/paper{_}16.pdf

[3] K. Nagasu, K. Sano, F. Kono, and N. Nakasato, "FPGA-based tsunami simulation: Performance comparison with GPUs, and roofline model for scalability analysis," *Journal of Parallel and Distributed Computing*, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0743731516301915

Acknowledgments: This research was partially supported by Grant-in-Aid for Scientific Research (B) No.17H01706 from MEXT, Japan. We thank the support of Intel university program.