FCsN: A FPGA-Centric SmartNIC Framework for Neural Networks

Anqi Guo*, Tong Geng*, Yongan Zhang†, Pouya Haghi‡, Chunshu Wu‡,
Cheng Tan†, Yingyan Lin‡, Ang Li*,and Martin Herbordt*
*ECE Department, Boston University, Boston, MA
†Pacific Northwest National Laboratory, Richland, WA
‡Rice University, Houston, TX
§Microsoft, Santa Clara, CA
Email: *{anqiguo,haghi,happycwu,herbordt}@bu.edu †{tong.geng,ang.li}@pnnl.gov
‡{yz87,yingyan.lin}@rice.edu §{chengtan}@microsoft.com

Network communication is increasingly becoming the performance bottleneck for scaled-out HPC and warehouse applications, as enormous CPU processing is devoted to packet processing, contributing to long latencies. To reduce this latency, advanced network interface cards known as SmartNICs have been introduced to handle networking functions. Dozens of commercial FPGA-based SmartNICs have been released (e.g., [1]–[3] and see surveys [4], [5]). Other commercial SmartNICs have been developed also with the aim of near-network processing [6]–[9]. There is also prior art that uses SmartNICs as compute resources [10], [11]. For instance, COPA [12], INCA [13], sPIN [14] provide a portable programming model to offload simple packet processing. Other work (e.g., [15]–[21]) supports collectives in FPGA-based hardware.

FPGA-based SmartNICs [22]–[24] offer great potential to significantly improve the performance of high-performance computing and warehouse data processing by tightly coupling support for reconfigurable data-intensive computation with cross-node communication, thereby mitigating the von Neumann bottleneck. Nevertheless, existing FPGA-based SmartNICs are constrained by three limitations. (i) *Host-control*: Although the offloading of some simple compute kernels has been demonstrated, this work generally assumes a host-device programming model, leaving the majority of control, scheduling, and management tasks to the host CPUs. This not only incurs an extra burden on the host CPUs, but also leads to poor utilization of the SmartNICs for handling the control-dependencies with the host through PCIe and software stacks. (ii) *Limited scalability*. Existing SmartNIC applications rarely involve offload of non-local tasks, missing opportunities for system-level designs that can span a distributed cluster, eliminate unnecessary data-movement, and support more efficient scheduling and workload balance. (iii) *Programmability*. As the control is performed by the host, most existing SmartNICs only handle relatively simple kernels.

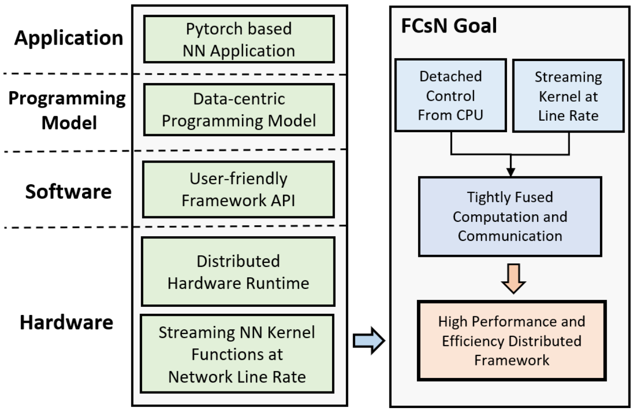
Application
Pytorch based NN Application
Programming Model
Data-centric Programming Model
Software
User-friendly Framework API
Hardware
Distributed Hardware Runtime
Streaming NN Kernel Functions at Network Line Rate
FCsN Goal
Detached Control From CPU
Streaming Kernel at Line Rate
Tightly Fused Computation and Communication
High Performance and Efficiency Distributed Framework

Fig. 1. Overview of FPGA Centric SmartNIC Design

In this work, we address these problems by presenting a user-friendly framework for neural network inference on **F**PGA-**C**entric **s**mart**NIC** (**FCsN**) that can perform computation, communication, and control altogether at the same time, allowing flexible and fine-grained task creation, distribution, execution, and finalization across multiple SmartNIC devices. FCsN uses a data-centric programming model that enables asynchronous, fine-grained task scheduling and is equipped with Python-based programming APIs; on the hardware side, FCsN is equipped with a hardware-based SmartNIC runtime to achieve CPU-detached scheduling and support high-performance execution of NN kernels at line-rate. This results in maximally hiding the computation latency with network communication for streaming applications at line-rate, and achieving high FPGA utilization and high performance at system level by avoiding CPU intervention. The current FCsN framework focuses on Neural Network applications, but has the potential of being extending into a general framework.

We have evaluated the FCsN framework with NN kernel functions and NN applications using an Alveo U280 board cluster (2-4 nodes) with communication links directly connected through on-chip QSFP ports. Our result that FCsN can achieve 10× speedups over the standard MPI-based system baseline in neural network applications (VGG, RestNet, Mobilenet, MLP [25]–[28]) and GNN models [29]–[31] with five datasets(Cora, CoraFull, Pubmed, CoautherPhysics, and Reddit [32], [33]).

978-1-6654-8332-2/22/$31.00 © 2022 IEEE

References

[1] "Alveo U25 SmartNIC Accelerator Card," https://www.xilinx.com/products/boards-and-kits/alveo/u25.html.

[2] "The Industry's First SmartNIC With Composable Hardware," https://www.xilinx.com/applications/data-center/network-acceleration/alveo-sn1000.html.

[3] "The Industry's First SmartNIC With Composable Hardware," https://www.intel.com/content/www/us/en/products/network-io/smartnic.html.

[4] H. Shahzad, A. Sanaullah, and M. Herbordt, "Survey and Future Trends for FPGA Cloud Architectures," in *IEEE High Performance Extreme Computing Conference*, 2021.

[5] C. Bobda, J. Mandebi, P. Chow, M. Ewais, N. Tarafdar, J. Vega, K. Eguro, D. Koch, S. Handagala, M. Leeser, M. Herbordt, H. Shahzad, P. Hofstee, B. Ringlein, J. Szefer, A. Sanaullah, and R. Tessier, "The Future of FPGA Acceleration in Datacenters and the Cloud," *ACM Transactions on Reconfigurable Technology and Systems*, vol. 15, no. 3, pp. 1–42, 2022, doi: 10.1145/3506713.

[6] Nvidia mellanox bluefield smartnic for ethernet. [Online]. Available: https://www.mellanox.com/files/doc-2020/pb-bluefield-smart-nic.pdf

[7] Liquidioii smart nics. [Online]. Available: https://www.marvell.com/products/data-processing-units.html

[8] Mellanox innova-2 flex open programmable smartnic. [Online]. Available: https://www.mellanox.com/files/doc-2020/pb-innova-2-flex.pdf

[9] Smartnic overview - netronome. [Online]. Available: https://www.netronome.com/products/smartnic/overview/

[10] A. Caulfield, P. Costa, and M. Ghobadi, "Beyond smartnics: Towards a fully programmable cloud: Invited paper," in *2018 IEEE 19th International Conference on High Performance Switching and Routing (HPSR)*, 2018, pp. 1–6.

[11] K. D. Underwood, R. Sass, and W. B. Ligon, "Cost effectiveness of an adaptable computing cluster," *ACM/IEEE SC 2001 Conference (SC'01)*, pp. 30–30, 2001.

[12] V. Krishnan, O. Serres, and M. Blocksome, "Configurable network protocol accelerator (copa) † : An integrated networking/accelerator hardware/software framework," in *2020 IEEE Symposium on High-Performance Interconnects (HOTI)*, 2020, pp. 17–24.

[13] W. Schonbein, R. E. Grant, M. G. F. Dosanjh, and D. Arnold, "Inca: In-network compute assistance," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '19. New York, NY, USA: Association for Computing Machinery, 2019. [Online]. Available: https://doi.org/10.1145/3295500.3356153

[14] T. Hoefler, S. Di Girolamo, K. Taranov, R. E. Grant, and R. Brightwell, "Spin: High-performance streaming processing in the network," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '17. New York, NY, USA: Association for Computing Machinery, 2017. [Online]. Available: https://doi.org/10.1145/3126908.3126970

[15] S. Gao, A. G. Schmidt, and R. Sass, "Hardware implementation of MPI barrier on an FPGA cluster," in *FPL 09: 19th International Conference on Field Programmable Logic and Applications*, 2009, pp. 12–17.

[16] O. Arap, G. Brown, B. Himebaugh, and M. Swany, "Implementing MPI_Barrier with the NetFPGA," in *International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA)*, 2013.

[17] J. Stern, Q. Xiong, J. Sheng, A. Skjellum, and M. Herbordt, "Accelerating MPI_Reduce with FPGAs in the Network," in *Workshop on Exascale MPI*, 2017, https://www.bu.edu/caadlab/exampi17.pdf.

[18] P. Haghi, A. Guo, T. Geng, J. Broaddus, D. Schafer, A. Skjellum, and M. Herbordt, "A reconfigurable compute-in-the-network fpga assistant for high-level collective support with distributed matrix multiply case study," in *2020 International Conference on Field-Programmable Technology (ICFPT)*, 2020, pp. 159–164.

[19] P. Haghi, A. Guo, T. Geng, A. Skjellum, and M. C. Herbordt, "Workload imbalance in hpc applications: Effect on performance of in-network processing," in *2021 IEEE High Performance Extreme Computing Conference (HPEC)*, 2021, pp. 1–8.

[20] P. Haghi, A. Guo, Q. Xiong, C. Yang, T. Geng, J. Broaddus, R. Marshall, D. Schafer, A. Skjellum, and M. Herbordt, "Reconfigurable switches for high performance and flexible mpi collectives," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 2, 2022, doi: 10.1002/cpe.6769.

[21] Z. He, D. Parravicini, L. Petrica, K. O'Brien, G. Alonso, and M. Blott, "Accl: Fpga-accelerated collectives over 100 gbps tcp-ip," in *2021 IEEE/ACM International Workshop on Heterogeneous High-performance Reconfigurable Computing (H2RC)*, 2021, pp. 33–43.

[22] A. Forencich, A. C. Snoeren, G. Porter, and G. C. Papen, "Corundum: An open-source 100-gbps nic," *2020 IEEE 28th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, pp. 38–46, 2020.

[23] N. Zilberman, Y. Audzevich, G. Covington, and A. W. Moore, "Netfpga sume: Toward 100 gbps as research commodity," *IEEE Micro*, vol. 34, no. 05, pp. 32–41, sep 2014.

[24] R. Jaganathan, K. Underwood, and R. Sass, "A configurable network protocol for cluster based communications using modular hardware primitives on an intelligent nic," in *11th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, 2003. FCCM 2003.*, 2003, pp. 286–287.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2015.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[27] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, vol. abs/1704.04861, 2017.

[28] M.-C. Popescu, V. E. Balas, L. Perescu-Popescu, and N. Mastorakis, "Multilayer perceptron and neural networks," *WSEAS Trans. Cir. and Sys.*, vol. 8, no. 7, p. 579–588, jul 2009.

[29] T. Geng, A. Li, R. Shi, C. Wu, T. Wang, Y. Li, P. Haghi, A. Tumeo, S. Che, S. Reinhardt, and M. C. Herbordt, "Awb-gcn: A graph convolutional network accelerator with runtime workload rebalancing," in *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2020, pp. 922–936.

[30] T. Geng, C. Wu, Y. Zhang, C. Tan, C. Xie, H. You, M. Herbordt, Y. Lin, and A. Li, *I-GCN: A Graph Convolutional Network Accelerator with Runtime Locality Enhancement through Islandization*. New York, NY, USA: Association for Computing Machinery, 2021, p. 1051–1063. [Online]. Available: https://doi.org/10.1145/3466752.3480113

[31] T. Geng, C. Wu, C. Tan, C. Xie, A. Guo, P. Haghi, S. Y. He, J. Li, M. Herbordt, and A. Li, "A survey: Handling irregularities in neural network acceleration with fpgas," in *2021 IEEE High Performance Extreme Computing Conference (HPEC)*, 2021, pp. 1–8.

[32] P. Sen, G. M. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, "Collective classification in network data," *AI Magazine*, vol. 29, no. 3, pp. 93–106, 2008.

[33] V. P. Dwivedi, C. K. Joshi, T. Laurent, Y. Bengio, and X. Bresson, "Benchmarking graph neural networks," *arXiv preprint arXiv:2003.00982*, 2020.