A Flexible HyperX Topology using Silicon Photonic Switching for Bandwidth Steering

Yu-Han Hung, Shijia Yan, Yiwen Shen, Ziyi Zhu, Min Yee Teh, Madeleine Glick and Keren Bergman

Department of Electrical Engineering, Columbia University, New York, NY 10027, USA

***Abstract* – We propose and experimentally demonstrate bandwidth steering using silicon photonic switches within a HyperX topology-based high-performance computing environment. Results running deep learning applications on a physical testbed show 20% improvement in execution time.**

I. Introduction

With the increasing demand for executing high-performance computing (HPC) applications on massive numbers of processors, the efficiency of the interconnection network among compute nodes can become a key performance bottleneck [1, 2]. A network topology featuring both low latency and high throughput is therefore desired for performance scalability. The HyperX topology is considered an efficient solution because it is a low-diameter direct network [3, 4]. However, because a standard HyperX is normally a nonreconfigurable topology, bandwidth congestion could appear in some links when using a minimal routing strategy [3]. To bypass the potential bandwidth bottleneck, non-minimal adaptive routing strategies are required, however, the adaptive routing strategies often result in additional latency due to the longer path length. In this paper, we propose a flexible HyperX topology with silicon photonic (SiP) switches that are inserted in the network, to relieve congestion and also to maintain low latency using a minimal routing strategy. We construct an experimental test-bed and observe the proposed scheme improves total application execution time by 20%.

II. System Architecture and Operation

The HyperX topology is a generalized topology covering Hypercube and Flattened Butterfly topology [3, 4]. HyperX is also a direct network since the Electronic Packet Switches (EPSs) are fully connected in each dimension. Fig. 1 shows three examples of a two-dimensional standard HyperX topology. The minimal routing in the HyperX topology requires at most two hops. As shown in Fig. 1(a), the traffic stream from EPS2 to EPS8 needs two hops whereas the traffic stream from EPS7 to EPS8 needs only one hop. As will be shown in the following, the minimal routing can lead to bandwidth congestion when multiple traffic streams share the same link in the standard HyperX topology.

We have built a 16-node HPC testbed arranged in a flexible HyperX topology, as shown in Fig. 2. The testbed consists of a data plane and a control plane. The control plane includes a Ryu-based SDN controller serving as the top-level management of both the EPSs through the OpenFlow protocol and our designed FPGA controller. In the control plane, the ToR EPSs are controlled and monitored through the OpenFlow protocol. Our FPGA controller is used to bias the SiP switch, manufactured by AIM photonics, to perform wavelength and spatial switching for bandwidth steering [5]. The data plane includes 16 servers, one microring resonator (MRR) SiP switch, and eight EPSs. In the data plane, each EPS is attached to two servers using 10G SFP+ electrical transceivers. The EPSs are also wired in two groups of mesh topology using 10G SFP+ electrical transceivers. The two groups of mesh topology are then connected using 10G SFP+ optical transceivers in the C-band as shown in Fig. 2. Compared to the standard HyperX topology, our proposed flexible HyperX topology integrates SiP switches so that optical circuit switching can be operated within a conventional electronic packet switched environment to flexibly exploit the communication bandwidth while maintaining low communication latency.

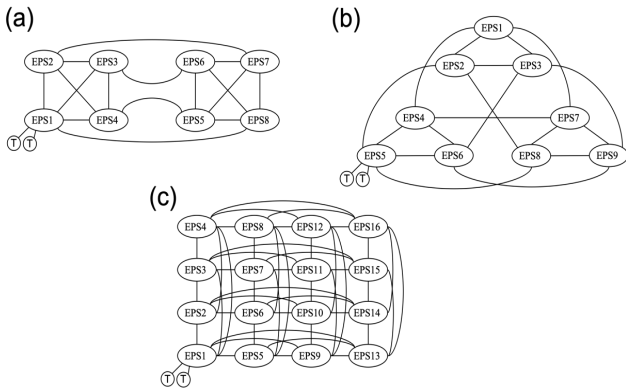

Fig. 1. Three examples of HyperX topology [3, 4]. T, terminals of compute nodes. EPS, electrical packet switch.

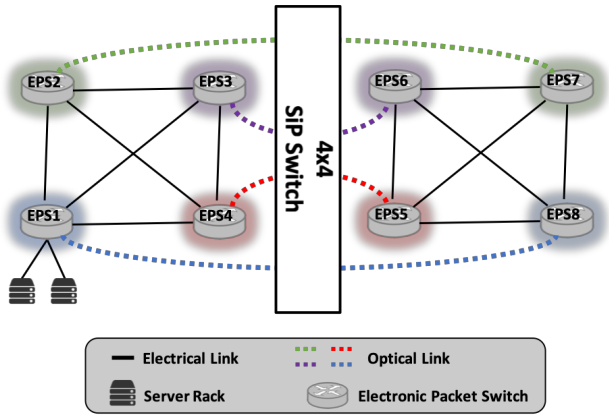

Fig. 2. Proposed testbed consisting of a HyperX topology and a SiP MRR switch. EPS, electrical packet switch.

978-1-7281-5891-4/20/$31.00 ©2020 IEEE

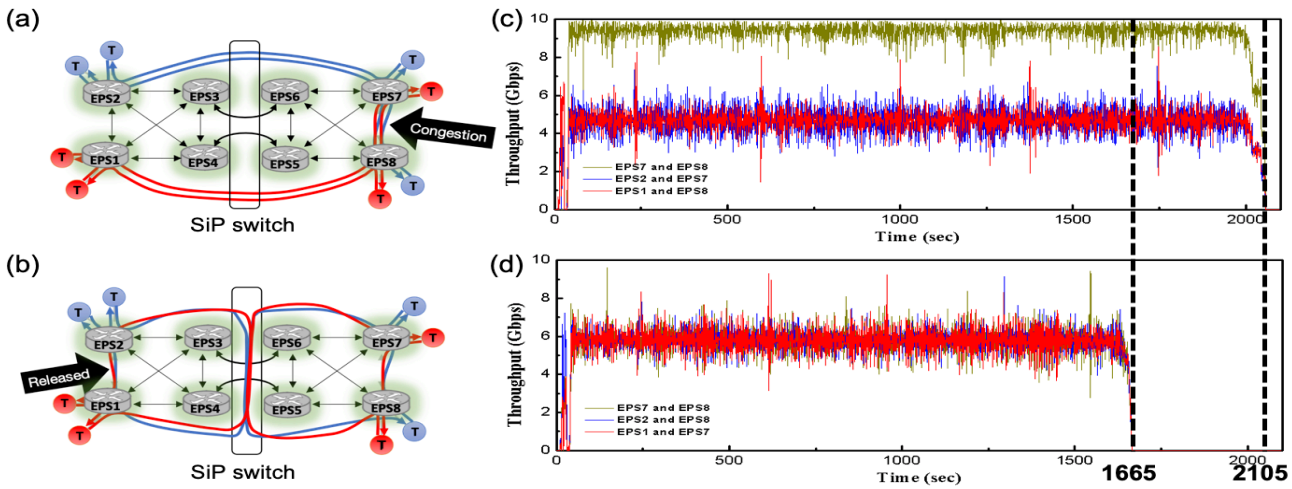

Fig. 3. Switch configuration for standard HyperX topology (a) and bandwidth-steered HyperX topology (b), respectively. T, terminals of compute nodes. EPS, electrical packet switch. (c), (d) Throughput in terms of application execution time for some links shown in (a) and (b), respectively.

III. Experimental Result and Analysis

To realize our proposed flexible HyperX topology, we first study the bandwidth congestion and run a synchronized data-parallel distributed deep learning application using ring-allreduce algorithm over a standard HyperX topology. The deep learning application we run in this study is MobileNetV2, a convolutional neural network for mobile architecture [6]. The assignment of physical machines is done as shown in Fig. 3(a). Figure 3(c) shows the throughput between EPS7 and EPS8 which includes flows traversing EPS2 to EPS8, from EPS1 to EPS7 and between EPS7 and EPS8. As shown in Fig. 3(c), the throughput between EPS7 and EPS8 occupies the majority of the full 10G link capacity, indicating the traffic is mostly congested. In addition, since a ring-allreduce communication bandwidth is determined by the link having the lowest throughput in the standard HyperX topology, we can observe the maximum communication bandwidth is approximately 5 Gbps, limited by the bandwidth congestion. An application execution time of 2105-second is observed.

The bandwidth congestion can be effectively relieved by using SiP switch-enabled bandwidth steering, as shown in Fig. 3(b). For example, with an optical circuit using the SiP switch, we can have a direct route between EPS2 and EPS8, and between EPS1 and EPS7. By doing so we can bypass the link between EPS7 and EPS8 in the standard HyperX topology, and thereby relieve the congestion. As shown in Fig. 3(d), the throughput between EPS7 and EPS8 is now reduced to approximately 6 Gbps, which is lower than the full 10G link capacity. Moreover, as shown in Fig. 3(d), we can observe the ring-allreduce communication bandwidth becomes approximately 6 Gbps, which is 20% higher than the 5-Gbps communication bandwidth shown in Fig. 3(c). Due to the enhancement of the communication bandwidth, executing the application only needs 1665 seconds in the bandwidth-steered HyperX topology, which is 20% faster than the application execution time in the standard HyperX topology.

IV. Conclusion

In this paper we propose a flexible HyperX topology and experimentally demonstrate SiP switch-enabled bandwidth steering. Since the topology integrates a SiP switch, the optical circuit switching can create a direct one-hop route for the traffic streams that normally need two-hop routes in the standard HyperX topology with a minimal routing strategy. Therefore, the bandwidth steering effectively relieves the congestion in the standard HyperX topology, and exhibits a 20% performance improvement of the application execution time.

Acknowledgment

This work is supported by Advanced Research Projects Agency—Energy (ARPA-E) (ENLITENED); U.S. Department of Energy (DOE) (DE-AR0000843). The authors would like to thank Dr. Qixiang Cheng (University of Cambridge, UK) for designing the SiP MRR switch used in this work.

References

[1] G. Michelogiannakis et al., "Bandwidth Steering in HPC Using Silicon Nanophotonics," *in Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis (SC'19)*, 2019.

[2] Yiwen Shen et al., "Silicon photonic-enabled bandwidth steering for resource-efficient high performance computing," *in Proceedings of Metro and Data Center Optical Networks and Short-Reach Links II (SPIE)*, 2019.

[3] J. Domke et al., "HyperX Topology: First At-Scale Implementation and Comparison to the Fat-Tree," *in Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis (SC'19)*, 2019.

[4] J. H. Ahn et al., "HyperX: Topology, Routing, and Packaging of Efficient Large-Scale Networks," *in Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis (SC'09)*, 2009.

[5] Y. H. Hung et al., "Silicon photonic switch-based optical equalization for mitigating pulsewidth distortion," *Opt. Express,* vol. 27, no. 14, pp. 19426-19435, 2019.

[6] Mark Sandler et al., "MobileNetV2: Inverted residuals and linear bottlenecks," *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018.