Accelerating of High Performance Data Centers using Silicon Photonic Switch-enabled Bandwidth Steering

Yiwen Shen, Sébastien Rumley, Ke Wen, Ziyi Zhu, Alexander Gazman, Keren Bergman

(1) Lightwave Research Laboratory, Columbia University, USA, ys2799@columbia.edu

Abstract *We demonstrate significant performance improvements for data centers and high-performance computing systems through dynamic bandwidth steering enabled by silicon photonic switches on a physical testbed. Experimental running the GTC benchmark on the testbed showed >30% reduction in total execution time.*

Introduction

The ability of data centers (DCs) and high-performance computing (HPC) systems to process and communicate information has become increasingly dependent on the bandwidth capacity and power efficiency of the interconnection network[1]. Current systems use a best-for-all approach characterized by static, over-provisioned networks with highly-connected topologies[2]. However, because many DC and HPC applications have unbalanced communication patterns that concentrate traffic within only a small percentage of the total available connections, this results in both under-utilized links that wastes power consumption as well as over-subscribed links that limit system performance[3].

The unique traffic characteristics of these applications present an opportunity to use reconfigurable optical networks that are able to flexibly steer bandwidth to match an application's specific traffic demands. Low-cost bandwidth steering can be realized through the means of silicon photonic (SiP) switches, which boast advantages such as low power consumption as well as low fabrication costs due to their compatibility with CMOS manufacturing techniques.

In this work we extend our network architecture[4] into a miniature data center/supercomputer testbed with SiP switch-enabled bandwidth steering capability. The testbed consists of 32 high-performance servers arranged in a Dragonfly topology, and is able to execute parallel HPC applications using the Message Passing Interface (MPI) programming standard. We report the execution of a skeletonized version of the Gyrokinetic Toroidal Code (GTC) benchmark application on our testbed in two configurations: 1) the baseline case (original Dragonfly topology) and 2) an optimized case (bandwidth steered using SiP switching) to better fit the traffic patterns of the GTC application. We observed differences of 30% to 42% in performance improvement depending on the GTC problem sizes.

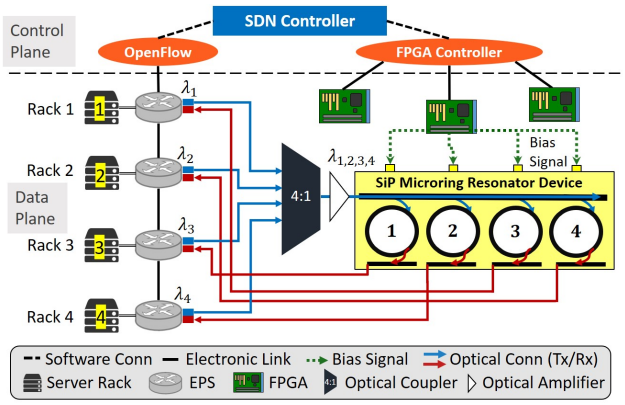

Fig. 1: Network architecture showing the integration of SiP device to conventional electronic packet switched network

Network Architecture

The network architecture used to run our applications was designed in a previous work[4] to integrate SiP devices for optical circuit switching within a conventional electronic packet switched environment (Fig. 1). The control plane features an SDN controller which manages both the SiP devices used for optical circuit switching, as well as the top-of-rack (ToR) electronic packet switches (EPSs) which connects groups of servers. During switching operations, the EPSs are managed by modifying their flow tables through the OpenFlow protocol, while control of the SiP device is achieved through an FPGA interface, which takes flow update packets from the SDN controller and applies pre-defined voltages to digital-to-analog converters (DACs) to configure the SiP switch elements. In this work a microring resonator (MRR) SiP device is used to facilitate wavelength and spatial switching for bandwidth steering. Each rack transmits signals at a unique wavelength through a ToR EPS and are then multiplexed together into the MRR device.

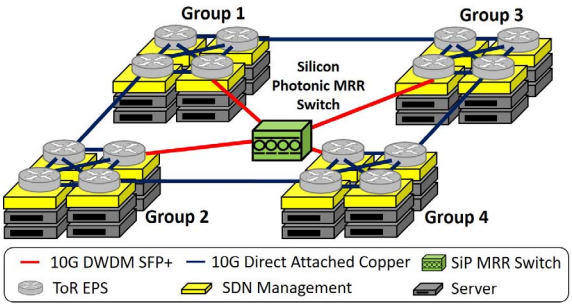

Fig. 2: Physical system testbed

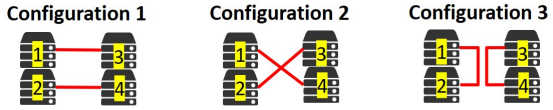

Fig. 3: Possible inter-group configurations enabled by the SiP device with 4 MRRs

Different racks are chosen to be bi-directionally connected to each other by tuning the resonances of the rings to input wavelengths. This redistributes the input signals to different output ports, resulting in one of the possible configurations shown in Fig. 3.

System Testbed

Our system testbed has a Dragonfly topology consisting of 4 groups of 4 EPSs each, with 2 servers connected to each EPS (Fig. 2). The EPSs are virtual bridges created on two Pica8 Ethernet switches. Each intra-group is fully connected with 10G Direct-Attached copper cables. For the inter-group connections, four copper-connected links form a static background connection while the other two are optically connected with 10G DWDM SFP+ transceivers for dynamic bandwidth steering, with wavelengths 1546.92 nm, 1550.12 nm, 1554.94 nm, and 1556.55 nm. These signals are coupled into the SiP MRR device and the output ports of its four MRRs connect to the receiving side of the ToR EPSs as depicted in Fig. 1. The resonance response of each MRR is separated by 1.27 nm with an FSR of 13 nm and each have a 3 dB bandwidth of 0.7 nm.

GTC Application Configuration

GTC is a scientific application used to simulate microturbulence of plasma within nuclear fusion power plants[5]. For our experiments, we use a skeletonized version of the GTC benchmark modified to operate on a physical system. The skeletonized version of GTC is normally designed to be used within the Structural Simulation Toolkit (SST)/macro, a platform for simulating the performance of applications on large-scale emulated HPC architectures[6]. We use a skeletonized version of GTC because it can generate sufficient traffic demand to saturate the 10G bandwidth capacities of a number of links in the testbed, which the original application could not produce. The skeletonized application reproduces the same traffic behavior of the original (e.g. packet sizes, destinations, etc.), but skips the computational routines, resulting in faster execution time compared to the original application. The same quantity of traffic within a shorter time frame equates to greater overall bandwidth demand, leading to congestion of a number of links in our system testbed.

Tab. 1: Performance increase for various job sizes

Number of Ranks	Execution Time (s)		Performance Increase %
	All-to-all Topology	Bandwidth Steered Topology	
64	30	20	40%
128	46	30	42%
256	98	66	41%
512	252	186	30%

The MPI protocol is used for communication and synchronization of processes over a group of nodes. For our experiments we use MPICH[7], a portable implementation of MPI, to run the GTC application over our servers.

Experimental Results

We ran various job sizes of our skeletonized GTC application over the system testbed with a baseline all-to-all Dragonfly topology and a bandwidth-steered topology and compared total execution times as a measure of performance, shown in Tab. 1. We observed between a 30% to 42% increase in performance for job sizes, which are defined by the number of ranks assigned.

The bandwidth-steered topology is determined by identifying which inter-group links were congested and which were the least utilized. This was done by using the SDN controller to monitor the traffic of the ports of the ToR EPSs corresponding to the inter-group links while the GTC application was running over the all-to-all topology. The monitored throughput for the 6 inter-group links over the application run-time is shown in Fig. 4. The upper plot shows the traffic for a job size using 256 ranks for the all-to-all Dragonfly topology, and the lower plot shows the bandwidth-steered topology. In the all-to-all topology, the throughput between Groups 1 and 2 and Groups 3 and 4 spikes to the full 10G link capacity, while the throughput between Groups 1 and 3 and Groups 2 and 4 are nearly zero over the

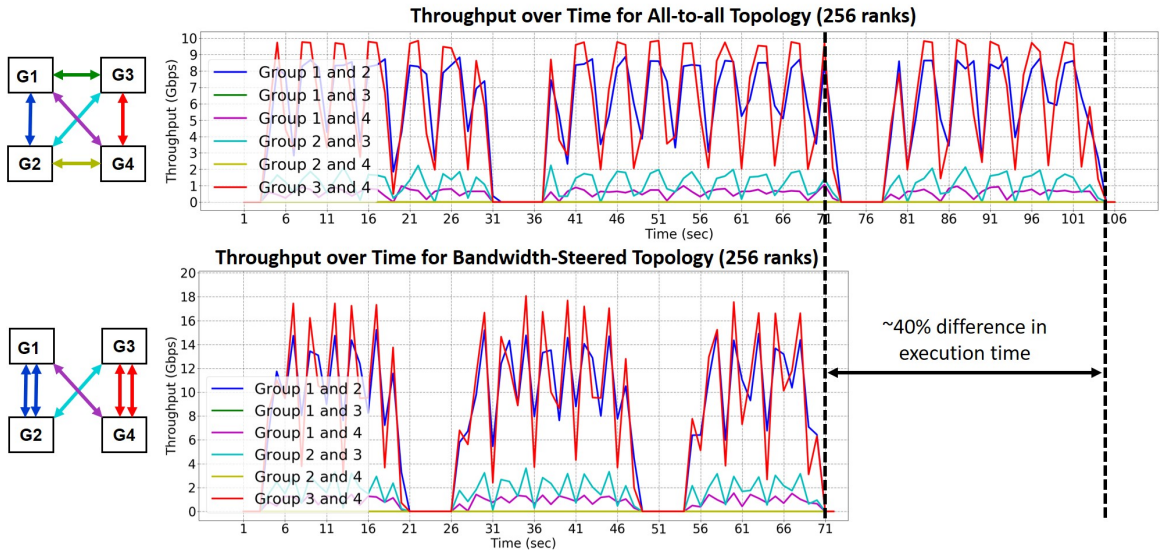

Fig. 4: Throughput over time of inter-group links over the run-time of GTC for an all-to-all Dragonfly and a bandwidth-steered topology

complete run-time of the application. Therefore, we steer the under-utilized bandwidth between Groups 1 and 3 to the highly used connection between Groups 1 and 2, and from Groups 2 and 4 to Groups 3 and 4. This is achieved by switching from Configuration 1 to Configuration 3 on the SiP MRR switch as depicted in Fig. 3, so that a total of 20 Gbps capacity is now available between Groups 1 and 2 and between Groups 3 and 4. In the second run with the bandwidth-steered topology, the throughput between these two sets of groups take advantage of the newly allocated bandwidth and traffic intensity reaches up to 18 Gbps, allowing the application to finish approximately 40% faster.

Note that in the bandwidth-steered topology, packets that previously used the direct inter-group link between Groups 1 and 3 in the all-to-all topology must first be forwarded to Group 2 or 4, and then to Group 1 or 3. Similarly, packets that traveled between Groups 2 and 4 must now first be forwarded to Group 1 or 3 before reaching the destination group. This extra hop explains why traffic between Groups 1 and 4 and between Groups 2 and 3 is observed to have risen a small amount in the bandwidth-steered topology in Fig. 4. However, since the traffic for these inter-group connections is so low, it does not have a detrimental effect on the performance as there is available bandwidth for the new path.

Conclusion

We demonstrate the applicability of SiP-enabled bandwidth steering to DC and HPC systems and show 30% to 42% performance improvement when operating the GTC application on our system testbed. In future work we will apply bandwidth steering to other traffic-intensive DC and HPC applications and utilize SiP switches with greater radices for increased network flexibility, as well as apply our network architecture to InfiniBand networks.

Acknowledgments

AIM Datacom; U.S. Department of Energy (DoE) Advanced Simulation and Computing (ASC) program through contract PO 1319001 with Sandia National Laboratories; Advanced Research Projects Agency Energy (ARPA-E) under the Enlightened Project

References

[1] S. Rumley, et al., “Optical interconnects for extreme scale computing systems,” Parallel Computing **64** (2017).

[2] J. Wang, S. Basu, C. McArdle and L. P. Barry, ”Large-scale hybrid electronic/optical switching networks for datacenters and HPC systems,” International Conference on Cloud Networking (CloudNet), Niagara Falls, ON (2015)

[3] K. Wen et al., “Flexfly: Enabling a Reconfigurable Dragonfly through Silicon Photonics,” Proc. SC16 International Conference for High Performance Computing, Networking, Storage and Analysis, Salt Lake City, UT, (2016).

[4] Y. Shen, et al., “Autonomous dynamic bandwidth steering with silicon photonic-based wavelength and spatial switching for Datacom networks,” Proc. OFC, Tu3F.2, (2018).

[5] K. Ibrahim, et al., “Analysis and optimization of gyrokinetic toroidal simulations on homogenous and heterogenous platforms,”. International Journal of High Performance Computing Applications **27** (2013).

[6] The Structural Simulation Toolkit, URL: http://sst-simulator.org/

[7] MPICH, URL: https://www.mpich.org/