Silicon Photonic Flex-LIONS for Reconfigurable Multi-GPU Systems

Silicon Photonic Flex-LIONS for Reconfigurable Multi-GPU Systems

Marjan Fariborz, Xian Xiao, Pouya Fotouhi, Roberto Proietti, and S. J. Ben Yoo, *Fellow, IEEE, Fellow, OSA*

Department of Electrical and Computer Engineering, University of California, Davis, California 95616, U. S. A.

(Top Scored)

***Abstract*— The rapid increases in data-intensive applications demand for more powerful parallel computing systems capable of parallel processing a large amount of data more efficiently and effectively. While GPU-based systems are commonly used in such parallel processing, the exponentially rising data volume can easily saturate the capacity of the largest possible GPU processor. One possible solution is to exploit multi-GPU systems. In a multi-GPU system, the main bottleneck is the interconnect, which is currently based on PCIe or NVLink technologies. In this study, we propose to optically interconnect multiple GPUs using Flex-LIONS, an optical all-to-all reconfigurable interconnect. By exploiting the multiple free spectral ranges (FSRs) of Flex-LIONS, it is possible to adapt (or steer) the inter-GPU connectivity to the traffic demands by reconfiguring the optical connectivity of one FSR while maintaining fixed all-to-all connectivity of another FSR. Simulation results show the benefits of the proposed reconfigurable bandwidth-steering interconnect solution under various traffic patterns of different applications. Execution time reductions by up to 5× have been demonstrated in this study including two applications of *convolution* and *maxpooling*.**

***Index Terms*— Silicon photonics, Optical Switching, Optical Reconfiguration, Multi-GPU systems**

I. Introduction

Today's graphic processing units (GPUs) can support up to 130 TFLOPS[1] and the GPUs are widely used as accelerators in computing systems for a variety of applications from high-performance computing (HPC) applications (such as HPCG, FFT, and so on), machine learning (ML), and graph workloads. In an ideal balanced system with 1 Byte/FLOP, a 130 TFLOPS computing would require a 130 TB/s or ~1Pb/s interconnection bandwidth. As such applications drive the rapid increases in the data volume and in processing complexity, even a single GPU processor with 1000 cores cannot provide sufficient processing power. A multi-GPU approach could support higher processing throughput and more flexibility in managing the processing resources.

GPUs can be scaled either by increasing the number of computing units through a larger single node (scaled-up) [2], or they can scale through a multi-node scenario (scaled-out) [3].

In a scaled-up approach the resources increased by adding more computing units (CUs) within a monolithic single die GPU design. This design is particularly useful for applications with a high degree of parallelism. However, even with scalable applications, increasing the number of CUs beyond a certain number is impossible due to the slowdown in transistor scaling [4] and the limited maximum die size [5].

In scaled-out approach, instead of increasing the die size in a single GPU, multiple GPUs are interconnected to create a multi-GPU system. Scaled-out multi-GPU platforms have higher offerings in terms of compute power, memory capacity, and memory bandwidth.

In recent years popular GPU vendors such as NVIDIA and AMD have come up with their multi-GPU solutions. AMD Radeon R9 295X2 connects two AMD Radeon R9 Series GPUs with Hyper transport link technology, which is a direct point-to-point link with 3.2 GB/s bandwidth. NVIDIA DGX-1 upon release, had 8 Tesla P100 [6][7] GPUs and 2 Xeon processors each with 20 cores, interconnected through ×16 PCI-e 3.0. Within a year, NVIDIA provided the option of using Tesla V100 [8] within DGX-1 systems to make use of the performance improvements across two generations (7.8 TFLOPS in V100 compared to 5.3 TFLOPS in P100, double precision in a single chip). In DGX-1 eight NVIDIA Tesla p100 or Tesla v100 cores connected through NVLink and Hybrid cube Mesh topology. NVIDIA DGX-2 connects 16 NVIDIA Tesla v100 through NVLink and 12 NVSwitches. NVLink can provide 25 GB/s bandwidth in each direction. In both DGX-1 and DGX-2, there is not a direct connection between all the GPUs and some of the communications are through additional hop (through another GPU or two NVSwitches).

This work was supported in part by ARO award # W911NF1910470, DoD award # H98230-19-C-0209, NSF ECCS award # 1611560, and in part by DoE UAI consortium award # DE-SC0019582, DE-SC0019526, and DE-SC0019692

M. Fariborz, X. Xiao, P. Fotouhi, R. Proietti, and S. J. B. Yoo are with the Department of Electrical and Computer Engineering, University of California, Davis, CA 95616 USA (e-mail: mfariborz@ucdavis.edu; xxxiao@ucdavis.edu; pfotouhi@ucdavis.edu; rproietti@ucdavis.edu; sbyoo@ucdavis.edu).

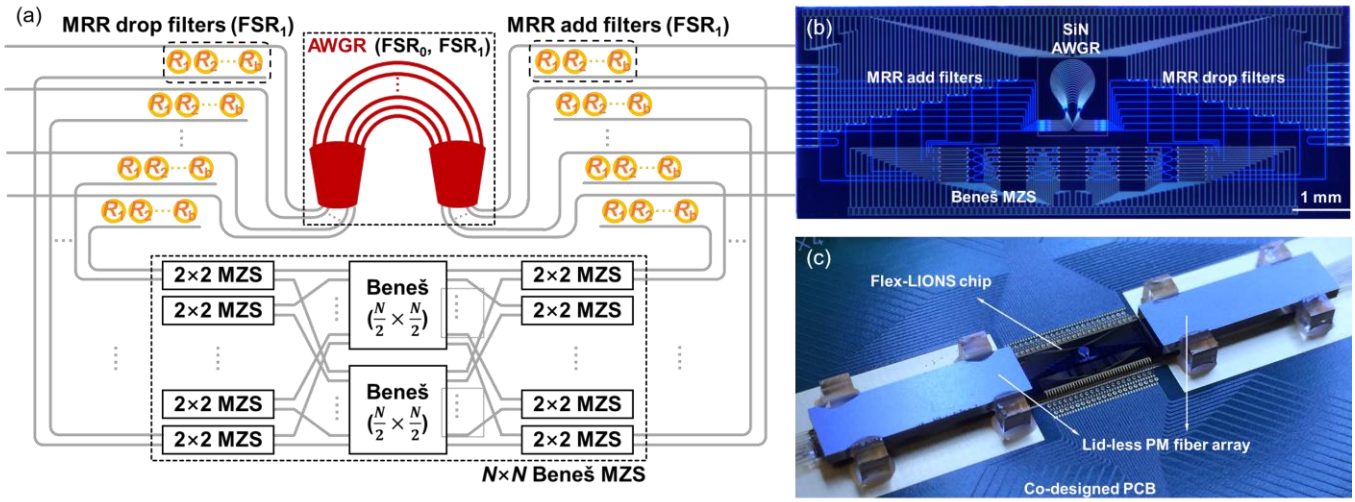
One of the major challenges in the scalability of multi-GPU platforms is to provide a scalable and high bandwidth-density interconnect. Due to the nature of the execution in the workloads (i.e. utilizing kernels for different tasks), there are spatial and temporal bursts in their communication patterns[9]. These traffic bursts create phases in the workloads where there are hotspot links in the different locations of the network. There have been several past studies that decreased the traffic on the interconnect and therefore improve the performance of the whole system [2], [10]. However, having a reconfigurable network that can adapt to these phases could significantly improve the performance in terms of latency, execution time, and energy efficiency. While today's electronic switches and optical fibers are unable to reconfigure and have a fixed topology, there has been recent work in SiPh integrated reconfigurable wavelength routing and space switching that allows for bandwidth reconfiguration [11]–[14].

Another challenge in these multi-GPU platforms is the non-uniform memory architecture (NUMA) nature of these systems. The NUMA effect increases the programming complexity and add to communication latency, especially with multi-hop communication between two remote nodes. This multi-hop communication among GPUs is a shortcoming in multi-GPU systems. As the size of the workloads starts to increase, mitigating the NUMA effect would be more challenging. Therefore, having an all-to-all connection can significantly improve the data movement latency and energy. Due to the limited numbers of the NVLink and PCIe ports in the GPUs, different generations of NVIDIA's multi-GPU systems are using hierarchical networks to mimic the all-to-all connection. This design raises multiple challenges, with one being the use of multiple crossbars to generate an all-to-all connection. As multi-GPU systems scale, this hierarchical deign would result in significant latency, area, and power overheads.

To solve the above issues, this paper, extends the work in [15] and propose to use a silicon photonic flexible low-latency interconnect optical network switch (Flex-LIONS) as a scalable and reconfigurable all-to-all photonic interconnect architecture for multi-GPU systems.

In particular, we propose to exploit multiple free spectral ranges in Flex-LIONS (multi-FSR Flex-LIONS) [16], [17]. In multi-FSR Flex-LIONS, the bandwidth between two nodes is increased by leveraging multiple FSRs. One FSR maintains minimum-diameter all-to-all connectivity between all the nodes, while the other FSRs can be used to increase the bandwidth between the hotspot links. Using multi-FSR Flex-LIONS enables us to have a multi-GPU computing system with reconfigurable all-to-all interconnects.

The remainder of this paper is organized as follows. Section II presents the architecture and working principle of Flex-LIONs and multi-FSR Flex-LIONS and reports the device and system results presents in [15]. Section III introduces the architecture of the proposed reconfigurable multi-GPU system. Section IV and V report on the performance evaluation methodology and results for the proposed system, demonstrating the benefits of adapting the inter-GPU interconnection according to the traffic



(d) **Before Reconfiguration $FSR_1 = FSR_0$ (All-to-All)**

	O_1	O_2	O_3	O_4	…	O_7	O_8
I_1	-	λ_2	λ_3	λ_4	…	λ_7	λ_8
I_2	λ_2	-	λ_4	λ_5	…	λ_8	λ_1
I_3	λ_3	λ_4	-	λ_6	…	λ_1	λ_2
I_4	λ_4	λ_5	λ_6	-	…	λ_2	λ_3
I_5	λ_5	λ_6	λ_7	λ_8	…	λ_3	λ_4
I_6	λ_6	λ_7	λ_8	λ_1	…	λ_4	λ_5
I_7	λ_7	λ_8	λ_1	λ_2	…	-	λ_6
I_8	λ_8	λ_1	λ_2	λ_3	…	λ_6	-

(e) **FSR_1 After Reconfiguration**

	O_1	O_2	O_3	O_4	…	O_7	O_8
I_1	-	$\lambda_2\lambda_4\lambda_5$	λ_3	λ_4	…	λ_7	λ_8
I_2	$\lambda_2\lambda_4\lambda_5$	λ_3	λ_4	λ_5	…	λ_8	λ_1
I_3	λ_3	λ_4	λ_5	$\lambda_6\lambda_4\lambda_5$	…	λ_1	λ_2
I_4	λ_4	λ_5	$\lambda_6\lambda_4\lambda_5$	λ_7	…	λ_2	λ_3
I_5	λ_5	λ_6	λ_7	λ_8	…	λ_3	λ_4
I_6	λ_6	λ_7	λ_8	λ_1	…	λ_4	λ_5
I_7	λ_7	λ_8	λ_1	λ_2	…	λ_5	$\lambda_6\lambda_4$
I_8	λ_8	λ_1	λ_2	λ_3	…	$\lambda_6\lambda_4$	-

Figure 1: (a) Two-FSR Flex-LIONS architecture with AWGR, MRR add-drop filters and Beneš MZS network. (b) Microscope image of the fabricated 8 × 8 SiPh Flex-LIONS (N = 8, b = 3) chip. (c) Photograph of the integrated Flex-LIONS module. (d) Wavelength routing table for All-to-All. (e) Wavelength routing table in FSR1 after reconfiguration.

profiles and communication phases of specific applications. Section VI concludes the paper summarizing the main results and future work.

II. Multi-FSR Flex-LIONS

A. Architecture

Flex-LIONS is a bandwidth-reconfigurable all-to-all optical switching fabric that contains an arrayed waveguide grating router (AWGR), microring resonator (MRR) add-drop filters, and Mach-Zehnder (MZ) switching networks as shown in Figure 1(a) [7][9][11][12]. For an $N\times N$ Flex-LIONS, an $N\times N$ arrayed waveguide grating router (AWGR) is used to provide all-to-all interconnection based on the wavelength-routing function. b microring resonator (MRR) drop filters at each input port of the AWGR are used to drop b wavelength division multiplexing (WDM) channels for bandwidth reconfiguration. The dropped channels are spatially switched and added to the desired output port by an $N\times N$ broadband Beneš Mach-Zehnder switch (MZS) network and b MRR add filters. In this case, the bandwidth between certain node pairs can be increased by a factor of b.

As proposed and demonstrated in [12], multiple FSRs are leveraged in Flex-LIONS architecture to maintain the all-to-all interconnectivity before and after bandwidth reconfiguration. Before reconfiguration, WDM wavelengths in two FSRs (FSR_0 and FSR_1) of the AWGR are both used for all-to-all communication due to the cyclicity of the AWGR. For bandwidth steering, only wavelengths in FSR_1 are switched by MRR add-drop filters and MZS networks while wavelengths in FSR_0 are untouched. In this case, the all-to-all interconnectivity is always maintained through FSR_0 which prevents unconnected node pairs after reconfiguration.

In the work presented in [11] and [12], a multi-FSR 8×8 integrated SiPh Flex-LIONS module (N = 8, b = 3) is experimentally demonstrated for bandwidth-reconfigurable all-to-all optical interconnects. The Flex-LIONS chip is designed and fabricated on a Si/SiN multi-layer platform as shown in Figure 1(b). The adjacent channel and non-be <−18 dB and <−28 dB which enables error-free all-to-all interconnects under the worst-case scenario. The characterized switching speed of the thermally-tuned MRR add-drop filters and the MZSs are ~10 μs. As shown in Figure 1(c), the integrated Flex-LIONS module is packaged on a co-designed printed circuit board (PCB) for electrical fan-out. Two 16-channel 127-μm-pitch polarization-maintaining (PM) fiber arrays are used for optical input and output. System testing results demonstrate bandwidth reconfiguration from 50 Gb/s to 125 Gb/s between selected pairs of nodes while error-free all-to-all optical interconnects are maintained. The insertion loss of the AWGR channels and reconfigured channels are <3.5 dB and <8.4 dB, respectively. The worst-case crosstalk penalty is measured to be 5.3dB.

Figure 1(d-e) show the wavelength routing table for multi-FSR Flex-LIONS. Figure 1(d) demonstrates the allocation before reconfiguration (Both FSRs are maintaining the all-to-all connection). Figure 1(e) depicts the allocation after reconfiguration, FSR1 is used the steer the bandwidth and FSR0 is used for all-to-all connection.

B. Experimental Results

Figure 2(a) shows the experimental setup used to characterize the fabricated chip. An 8-channel 200-GHz-spacing WDM signal was generated by using eight small form pluggable (SFP) TRXs matching the AWGR channels. To align the polarization, we used eight polarization controllers (PCs) before the multiplexer and one polarizer after the multiplexer. All the WDM channels were modulated by an MZ modulator at 25 Gb/s. The driving signals were 211-1 PRBS signals generated by a high-speed digital to analog converter (DAC). The modulated signal was boosted by an erbium-doped fiber amplifier (EDFA) and split by a 1×8 splitter. We used single-mode fiber patch cables of different length to decorrelate the eight channels. The single-mode fiber patch cables were followed by PCs to align the signal polarization to the slow axis of the PM input fiber of the packaged Flex-LIONS chip. The output signal of the Flex-LIONS chip was then received by an optically pre-amplified receiver (RX). A real-time error analyzer (EA) performed BER measurements as a function of the RX input power, which is measured by the optical power monitor of the variable optical attenuator (VOA).
Figure 2 (b) and Figure 2(c) show the transmission spectrum and the BER curves when there is no reconfiguration, and the connection is all-to-all. Figure 2(b) shows the transmission spectrum from input port 4 to output port 8 with AWGR channel λ8. The power penalty from center and side input ports is measured under the worst-case crosstalk scenario (all the input signals aligned in polarization). In Figure 2(c) the BER curves demonstrate that in the selected input and output port combinations the all-to-all interconnects are error-free. The measured power penalty at BER=10-12 is in the range of 3.9 dB to 5 dB compared with back-to-back (no crosstalk signal added). Figure 2(d) and Figure 2(e) show the transmission spectrum and the BER curve in the reconfiguration scenario. In Figure 2(d) the transmission is between input port 4 to output port 8 after reconfiguration. λ8 channel is from the passband of AWGR while the other three channels (λ1, λ3, λ5) are from the path through cascaded MRR add-drop filters and Beneš MZS network. Figure 2(e) shows the error-free operation of all the four channels demonstrates 4× bandwidth steering (25 Gb/s to 100 Gb/s) between input port 4 and output port 8.

III. Optically Interconnected Multi-GPU System

Multi-GPU systems cannot scale performance linearly due to the high latency and low throughput associated with inter-GPU communication [21]. As already discussed above, in this study, we use an enabling silicon photonic technology such as Flex-LIONS to interconnect n number of GPUs (see Figure 3) to improve latency in the multi-GPU systems.
Current commercial multi-GPU systems are connected through PCIe or NVLink (cirrascale gx8 and DGX2) [18]–[20]. PCIe 3.0 can provide 8 Gb/s in one direction per lane. Therefore, PCIe 3.0 x16 has a maximum of 128 Gb/s in each direction. The

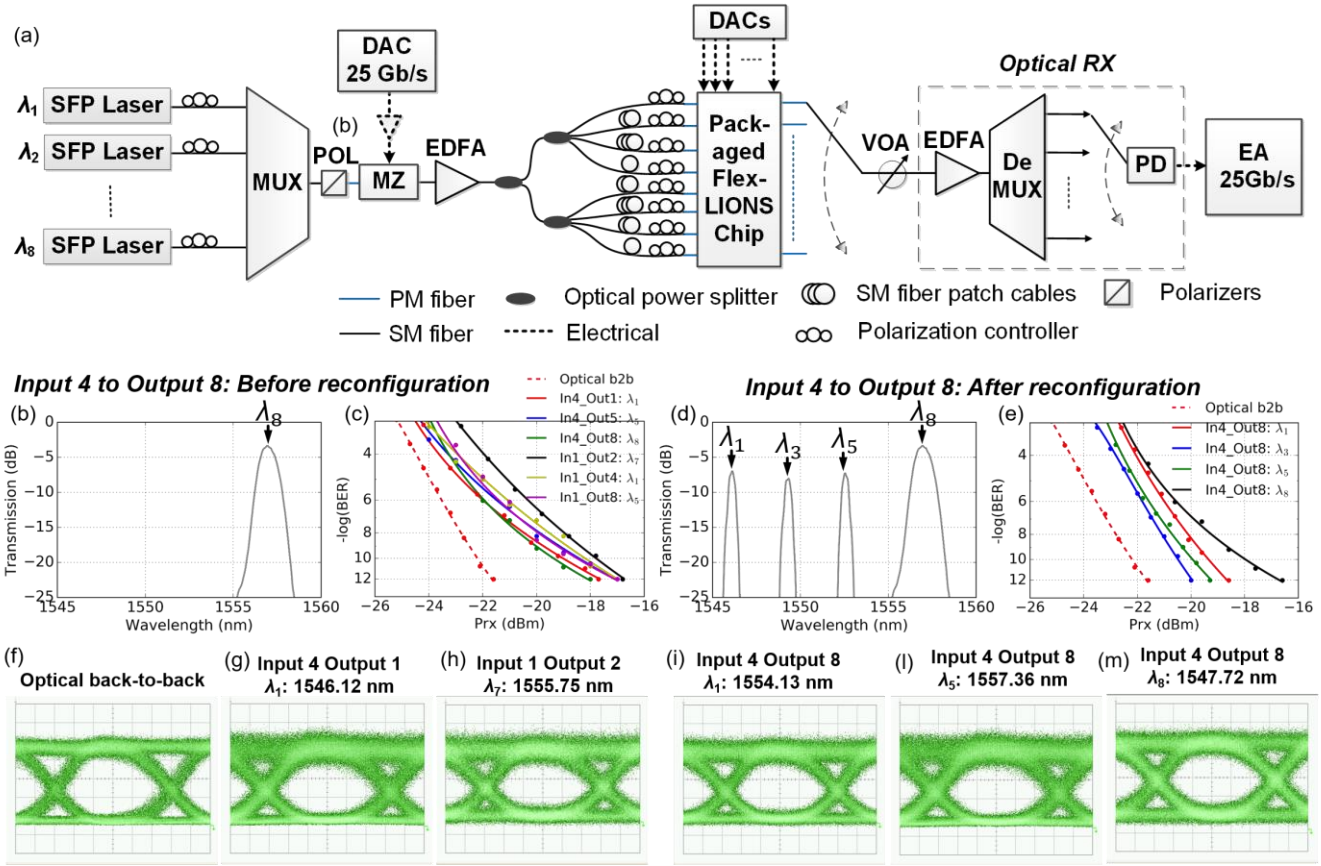


Figure 2: (a) Experimental setup. (b) Transmission spectrum from input port 4 to output port 8 before reconfiguration. (c) BER curves of all-to-all interconnects (1× bandwidth for all interconnected nodes). (d) Transmission spectrum from input port 4 to output. (e) BER curves after reconfiguration. (f)-(g): Eye diagrams of optical back-to-back and signals going through Flex-LIONS chip before and after reconfiguration.

second and third generation of NVlink provides 20GB/s and 25GB/s in each direction.

In this study, we exploit two FSRs of Flex-LIONS (as demonstrated in [16] where each wavelength in each FSR is modulated at 32GHz using on-off keying (OOK). This still allows to support an I/O bandwidth of 512 Gb/s (64 GB/s), which is comparable with our baseline using four NV links 2.0 as in Nvidia DGX-1 with Tesla p100 (80 GB/s). To further increase the bandwidth four-level pulse-amplitude modulation (PAM4) can be used to achieve 64 Gb/s per FSR. Previous studies have achieved beyond 100 Gb/s data rate [22], [23]. Sun et al. [23] demonstrated a SiPh microring modulator (MRM) operating at 128 Gb/s PAM4 with an integrated silicon heater can tune the ring resonator to more than one 6.6 nm FSR of the resonator.

Figure 3 (a) shows the static all-to-all interconnection called LIONS. In this architecture, the communication between each pair of GPUs is through one FSR with a bandwidth of 128Gb/s. The connection is static and cannot be reconfigured, like in current multi-GPU systems. In Figure 3(b), the inter-GPU communication is instead through Flex-LIONS. At each input/out of the AWGR there are *b* MRR add-drop filters and a multi-wavelength spatial switch. This allows us to reconfigure the lambda in the interconnect to adapt to different traffic patterns and boost the bandwidth and reduce link congestion. Using Flex-LIONS we can increase the bandwidth between each link by b×64 Gb/s.

In multi-FSR Flex-LIONS [Figure 3 (c)], tuning the specific MRR filters, up to *b* wavelengths (out of *N)* will enable the communication from a source and destination port through an *N*×*N* multi-wavelength switch, which increases the bandwidth between the corresponding two ports by *b/2*×64 Gb/s. In multi-FSR Flex-LIONS, one FSR (e.g. FSR0) can be used to maintain the minimum diameter between all the nodes, and FSR1 is used for reconfiguration to increase the bandwidth in the hotspot links.

IV. Methodology

To model our target multi-GPU system, we used MGPUSim [24], which models the Graphics Core Next 3 (GCN3) instruction set algorithm (ISA) and can support up to four GPUs. These GPUs are connected using PCIe 3.0. We used the MGPUSim model to generate the RDMA (Remote Direct Memory Access) traces between the GPU units. To model the interconnect and perform reconfiguration we used Garnet2.0 [25] the network model in the gem5 [26] simulator.

Figure 3(d) shows the architecture of our proposed system. The communication between the GPUs and the interconnect, are through optical wavelengths. Each GPU communicates to the interconnect through 8 wavelengths with two FSRs. One FSR per lambda operating at 32Gb/s. The aggregated bandwidth of each GPU is 512Gb/s. The host CPU uses PCIe link to communicate to the control plane of the Flex-LIONS and to transfer data to each GPU device.

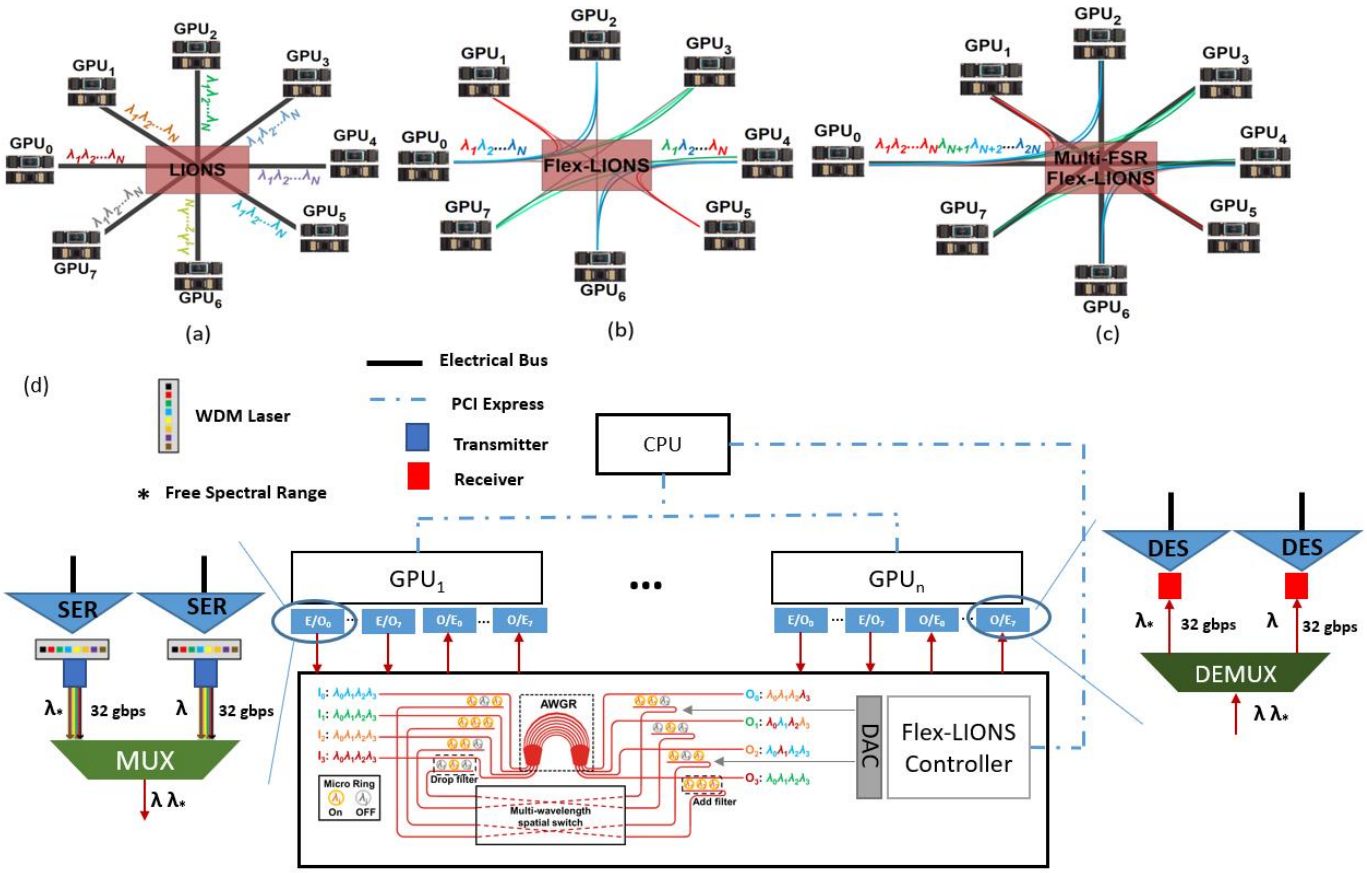


Figure 3: Proposed Architecture: a) LIONS (all-to-all connection), b) Flex-LIONS (Bandwidth reconfigurable interconnect), c) Multi-FSR Flex-LIONS (All-to-all bandwidth reconfigurable network), d) control-plane and data-plane interface in the proposed multi-GPUs.

In multi-GPU systems, GPU development kits give access to compute engines and memory copy engines. The compute engines are responsible for running kernels, whereas the memory copy engines allow simultaneous two-way memory transfers. Achieving an efficient memory access pattern with reasonable utilization of the bandwidth to memory is a major programming challenge in the development of multi-GPU applications. In such applications, programmers are compelled to manually manage memory transfers. Kernels are ideally split evenly to multiple GPUs based on their memory access patterns. To mitigate the NUMA affect the kernels need to be portioned based on the topology of the network and also their memory access pattern. This would result in multiple phases being generated in the memory trace of these applications. Within a task, data access and storage are abstracted from the programmer. Internally, these abstractions can be implemented using different schemes, specifically tuned to each architecture, in order to provide each kernel with its required data [2], [9], [27], [28]. Due to the all-to-all topology of our network one of the advantages of our design is that the programmer no longer needs to know the topology of the network to be able to perform the most efficient data placement.

The launch of a new kernel in each GPU can change the traffic pattern in the network and therefore creates a new communication phase. With every kernel launch, the host CPU can understand the size and address range of the memory which that kernel is going to access. Using this information, the host can use a low latency link, PCIe, to communicate to Flex-LIONS controller (which is a FPGA microcontroller). The controller then uses digital to analog convertor to be able to create the current to tune the MRR add-drop filters and perform the reconfiguration. The overhead of launching a kernel from the task queue to the GPU depends on the size of the kernel. Zhang et.al. [29] showed that the kernel launch latency can be around 3µs. The Flex-LIONS can be reconfigured based on each kernel launch. The reconfiguration includes sending the new reconfiguration command from host CPU to the control plane (Flex-LIONS controller) of the Flex-LIONS fabric, generating current from the Flex-LIONS controller to reconfigure the electro-optically tunable microrings. Both kernel launch latency and reconfiguration time latency are negligible compared to the kernel execution which is different compared to the kernel size but is in the order of 100× microseconds.

Given the small size of our network, we used a simple heuristic approach to reconfigure. In more complicated networks, such as a fat tree, more sophisticated algorithms can be used for reconfiguration [30]. Existing neural network and deep learning algorithms could also be deployed on the Flex-LIONS controller to predict the traffic matrix and hotspot and coldspot links [31], [32]. However, this is outside of the scope of this paper and will be an objective for our future studies.

V. Evaluation

For evaluation, we used two workloads: *convolution* and *maxpooling*, from AMD's Accelerated Parallel Processing (APP) Software Development Kit (SDK). Both workloads ran on the four-GPU platform on MGPUSim and showed distinct communication phases. Figure 4, depicts the traces for each workload. *Convolution* can be divided into three phases and *maxpooling* can be divided into two phases. Each of these phases is 100 µs, and 1000 µs apart, respectively. Figure 4(a) shows that in phase 1, around 60% of the total traffic is represented by the requests and responses that are initiated from GPU2 to GPU1, (GPU21, shows the traffic from GPU2 to GPU1), while in phase 2, around 35% of traffic is from GPU4 toward GPU1, (GPU41).

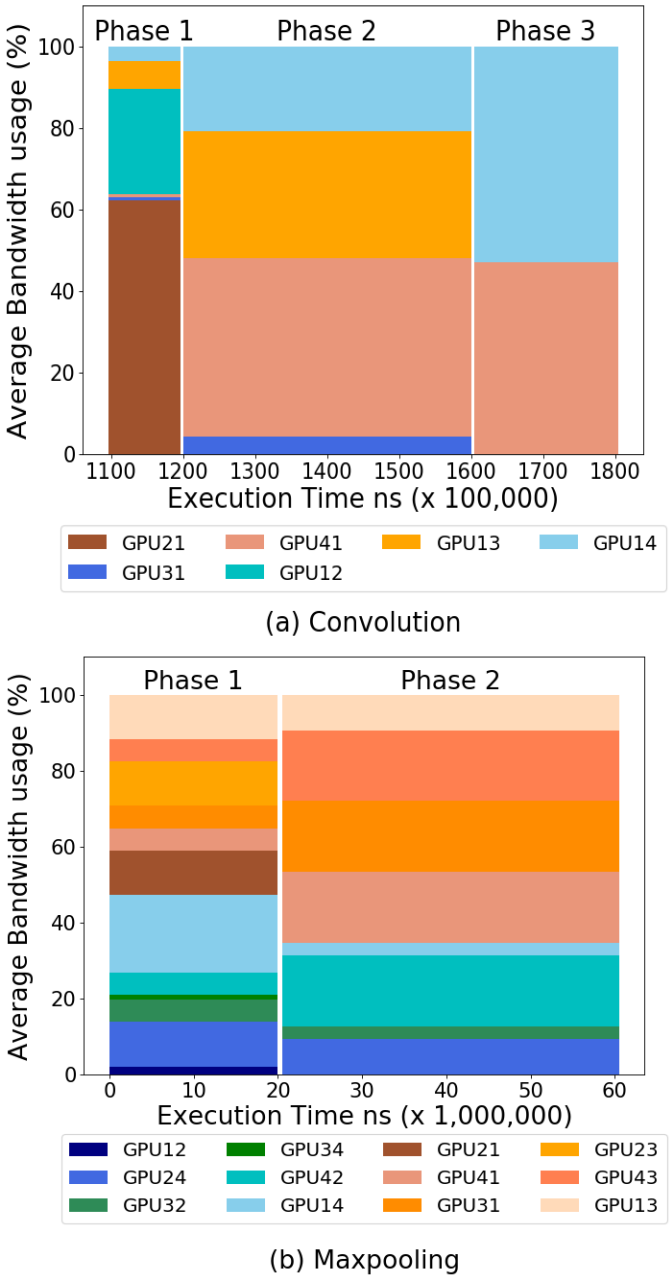


Figure 4: Phases based on the RDMA traces. y-axis shows the percentage of bandwidth usage for each link. GPU12 indicated the one-directional link that transfer requests from GPU2 to GPU1. a) Convolution, b) Maxpooling.

One interesting observation is comparing the *convolution* and *maxpooling* traces with each other. In Figure 4(a) most of the traffic is dedicated to the connection between two GPUs (around 80% in phase 1) and the communication rate between all the other GPUs is either very small or non-existent. In this type of application, using Flex-LIONS with full reconfiguration capability is more beneficial since we have a distinct hotspot between two computing nodes and we either do not have an all-to-all background traffic or traffic rate is very small. Therefore, multi-hop communication for these background traffic will not hurt the performance of the system. By doing the full reconfiguration, we can assign a large bandwidth to the hotspot link and maximize performance improvement.

For the *maxpooling* workload shown in Figure 4(b) the traffic is instead more equally distributed among all the links. In these types of workloads, the difference between the peak link bandwidth to average link bandwidth in each phase is small, which means that the use of Flex-LIONS to do full reconfiguration can cause a significant increase of the average number of hops, degrading the performance. In these workloads, maintaining the all-to-all connectivity and reconfiguring trough one FSR (to improve the bandwidth on hotspot links), has more impact on the performance compare to performing a full reconfiguration using Flex-LIONS.

We evaluated our system based on average packet latency improvement that would translate into an improvement in terms of execution time. In our simulation we replicated the traces collected in MGPUSim (which is limited to four GPUs) to extend our simulations to an 8-GPU system. We achieved this by creating two groups of four GPUs and replicating part of the traces retrieved from MGPUSim for intra-group traffic and used the rest for inter-group traffic. Based on the discussion given in section IV, this traffic is still representative of the traffic patterns in multi-GPU systems.

We used the topology of NVIDIA p100 DGX-1 as our baseline topology, which is a hybrid cube mesh topology. The GPUs in our baseline are connected through four NVLink 2.0, (20GB/s for each direction and each link). We compared the results of using Multi-FSR Flex-LIONS with regular Flex-LIONS and static all-to-all interconnection with no reconfiguration capability. For a fair comparison, the bitrate per lambda in Flex-LIONS is twice the bitrate per lambda in multi-FSR Flex-LIONS (since in the latter there are two lambdas between each node pair). Figure 5 shows the topology of targeted systems. Figure 5(c) and Figure 5(d) show an example for reconfiguration for both Flex-LIONS and multi-FSR. Figure 6 depicts our simulation results. In the *convolution* workload, since the ratio between the peak bandwidth and the average bandwidth is high and the traffic distribution is mainly between two nodes, using Flex-LIONS would be more beneficial.

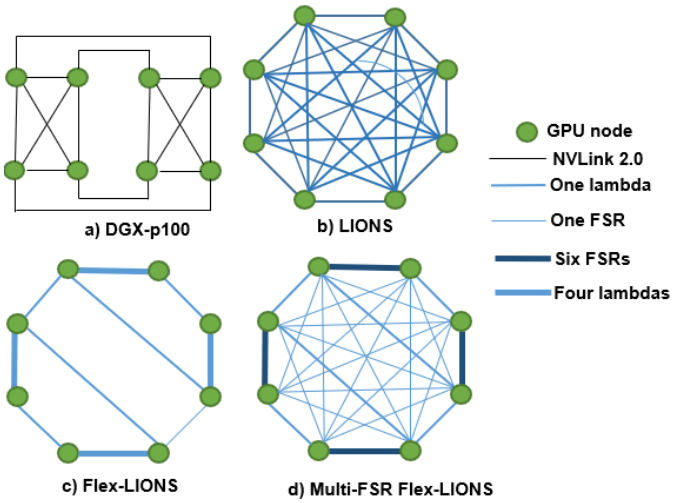


Figure 5: Topology of simulated systems.

However, when using Multi-FSR Flex-LIONS, for these phases we cannot assign the maximum bandwidth since we need to maintain the all-to-all connection.

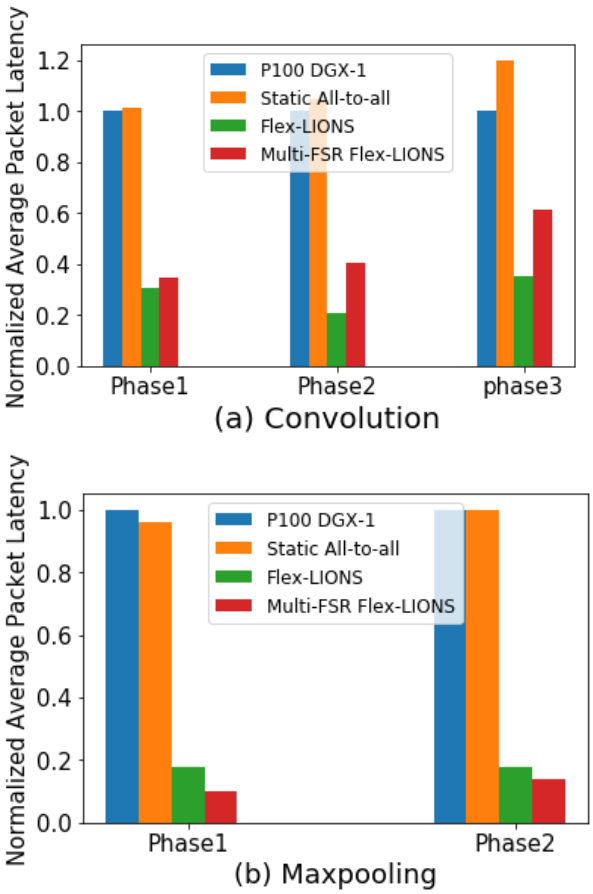


Figure 6: Simulated average packet latency for a) Convolution b) Maxpooling.

In the *maxpooling* application, the traffic is more distributed between all the nodes. In this case, when we use Flex-LIONS, maintaining the all-to-all connectivity is pivotal in improving the performance (FSR0 would guarantee the shortest path between the nodes while we steer FSR1 to increase the bandwidth). That is why Multi-FSR Flex-LIONS shows advantages.

As mentioned above, Improving the average packet latency can directly improve the execution time. For instance, in convolution, by reconfiguring the network using Flex-LIONS the average packet latency can be improved by about 70% in average across all of the phases compare to the DGX-1, which translates directly into a reduction of the execution time at each phase and the total execution time can also be improved in average by 75%. Figure 7 shows the total execution time improvement of Flex-LIONS and multi-FSR Flex-LIONS compared to the static all-to-all interconnect.

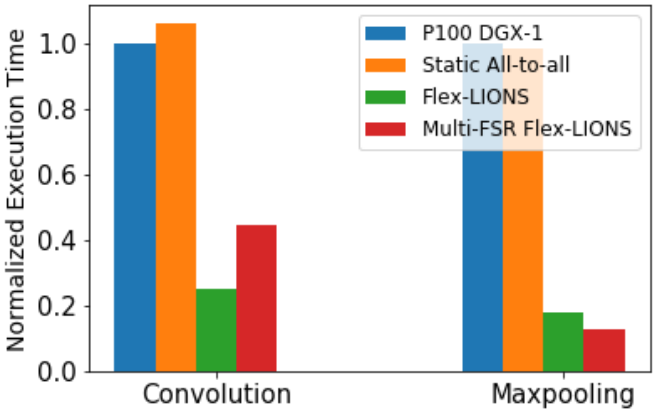


Figure 7: Normalized execution time for both applications. a) Convolution b) Maxpooling.

VI. Conclusion

This paper studies the application of Flex-LIONS [15], a SiPh reconfigurable interconnect solution, for multi-GPU systems. Based on our simulation studies, both Flex-LIONS and multi-FSR Flex-LIONS exhibit lower latency compared to the current multi-GPU system developed by NVIDIA, DGX-1 working with Tesla P100 [7]. Depending on the type of traffic pattern, it is useful to reconfigure one or all the FSRs. In applications with similar links' utilization maintaining the all-to-all connection while reconfiguring improves the performance of the system. Whereas, in applications with high congested links and low all-to-all traffic, using full reconfiguration can further improve the latency and overall execution time of an application. In this work, the total I/O bandwidth of each GPU is 64GB/s in each direction (similar to our physical layer experiment with on-off keying modulation format (OOK) and eight-port Flex-LIONS with two free-spectral ranges). There are multiple ways to increase this bandwidth density in AWGR-based devices like Flex-LIONS. For instance, we could increase the bandwidth with PAM4 modulation [33] or spatial division multiplexing to achieve higher bisection bandwidth.

Introducing the reconfigurability of the interconnect in the multi-GPU system will not be beneficial if the reconfiguration is slower than the changes in the traffic pattern (e.g., workloads with micro traffic patterns with small phases). However, due to the single instruction, multiple thread (SIMT) nature of traffic patterns in GPUs and due to the rapid rise of the dataset sizes,

we expect all phases of the traffic to get longer not shorter. This is due to the programing models in the GPUs where workloads are executed in a highly multi-threaded fashion.

In this study, we have used NVIDIA p100 DGX-1 as the baseline topology, which consists of eight GPUs interconnected in a Hyper-X topology. State of the art NVIDIA multi-GPU system consist of 16 GPUs in which, the topology has changed from Hyper-X to NVSwitch-based interconnect topology. Due to NUMA architecture of current multi-GPU systems, the current applications are topology dependent. A meaningful performance investigation at larger scale will require implementing a full system simulator capable to run applications rather than trace-based simulations. This is beyond the scope of this paper and will be left as part of our future studies. In addition, we will investigate work partitioning algorithms and study the characteristics of RDMA traffic patterns at each node to develop a reconfiguration algorithm to dynamically reconfigure Flex-LIONS.

VII. References

[1] "TITAN RTX Ultimate PC Graphics Card with Turing | NVIDIA." [Online]. Available: https://www.nvidia.com/en-us/deep-learning-ai/products/titan-rtx/. [Accessed: 30-Jul-2020].

[2] A. Arunkumar, E. Bolotin, B. Cho, U. Milic, E. Ebrahimi, O. Villa, A. Jaleel, C.-J. Wu, and D. Nellans, "MCM-GPU: Multi-Chip-Module GPUs for Continued Perfor-mance Scalability," vol. 17.

[3] "DGX-2 : AI Servers for Solving Complex AI Challenges | NVIDIA." [Online]. Available: https://www.nvidia.com/en-us/data-center/dgx-2/. [Accessed: 01-Aug-2020].

[4] "ITRS Reports - International Technology Roadmap for Semiconductors." [Online]. Available: http://www.itrs2.net/itrs-reports.html. [Accessed: 30-Jul-2020].

[5] "ASML products & services | Supplying the semiconductor industry." [Online]. Available: https://www.asml.com/en/products. [Accessed: 30-Jul-2020].

[6] "Tesla P100 Data Center Accelerator | NVIDIA." [Online]. Available: https://www.nvidia.com/en-us/data-center/tesla-p100/. [Accessed: 17-Sep-2020].

[7] "(No Title)." [Online]. Available: https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf. [Accessed: 08-Nov-2020].

[8] "NVIDIA V100 | NVIDIA." [Online]. Available: https://www.nvidia.com/en-us/data-center/v100/. [Accessed: 17-Sep-2020].

[9] T. Ben-Nun, M. Sutton, S. Pai, and K. Pingali, "Groute: An Asynchronous Multi-GPU Programming Model for Irregular Computations," *ACM SIGPLAN Not.*, vol. 52, no. 8, pp. 235–48, Jan. 2017.

[10] U. Milic, O. Villa, E. Bolotin, A. Arunkumar, E. Ebrahimi, A. Jaleel, A. Ramirez, and D. Nellans, "Beyond the socket: NUMA-aware GPUs," in *Proceedings of the Annual International Symposium on Microarchitecture, MICRO*, 2017, vol. Part F131207, pp. 123–35.

[11] X. Xiao, R. Proietti, S. Werner, P. Fotouhi, and S. J. B. Yoo, "Flex-LIONS: A Scalable Silicon Photonic Bandwidth-Reconfigurable Optical Switch Fabric," in *OECC/PSC 2019 - 24th OptoElectronics and Communications Conference/International Conference Photonics in Switching and Computing 2019*, 2019.

[12] K. Wen, P. Samadi, S. Rumley, C. P. Chen, Y. Shen, M. Bahadori, K. Bergman, and J. Wilke, "Flexfly: Enabling a Reconfigurable Dragonfly through Silicon Photonics," in *International Conference for High Performance Computing, Networking, Storage and Analysis, SC*, 2017, pp. 166–77.

[13] X. Xiao, R. Proietti, G. Liu, H. Lu, P. Fotouhi, S. Werner, Y. Zhang, and S. J. B. Yoo, "Silicon Photonic Flex-LIONS for Bandwidth-Reconfigurable Optical Interconnects," *IEEE J. Sel. Top. Quantum Electron.*, p. 1, 2019.

[14] Z. Cao, R. Proietti, M. Clements, and S. J. B. Yoo, "Experimental demonstration of flexible bandwidth optical data center core network with all-to-all interconnectivity," *J. Light. Technol.*, vol. 33, no. 8, pp. 1578–85, Apr. 2015.

[15] X. Xiao, R. Proietti, G. Liu, H. Lu, Y.-C. Ling, Y. Zhang, and S. J. Ben Yoo, "Integrated SiPh Flex-LIONS Module for All-to-All Optical Interconnects with Bandwidth Steering," in *Optical Fiber Communication Conference (OFC) 2020*, 2020, p. Th3B.3.

[16] X. Xiao, R. Proietti, G. Liu, H. Lu, Y. Zhang, and S. J. B. Yoo, "Multi-FSR Silicon Photonic Flex-LIONS Module for Bandwidth-Reconfigurable All-To-All Optical Interconnects," *J. Light. Technol.*, vol. 38, no. 12, pp. 3200–8, Jun. 2020.

[17] X. Xiao, Y. Zhang, K. Zhang, R. Proietti, and S. J. B. Yoo, "Multi-FSR On-Chip Optical Interconnects Using Silicon Nitride AWGR," in *CLEO: QELS_Fundamental Science*, 2019, pp. JTh2A--70.

[18] "Dedicated Hardware | Cirrascale Cloud Services." [Online]. Available: https://cirrascale.com/solutions_dedicatedhardware.php. [Accessed: 03-Aug-2020].

[19] "Essential Instrument for AI Research | NVIDIA DGX-1." [Online]. Available: https://www.nvidia.com/en-us/data-center/dgx-1/. [Accessed: 01-Aug-2020].

[20] N. R. Tallent, N. A. Gawande, C. Siegel, A. Vishnu, and A. Hoisie, "Evaluating On-Node GPU interconnects for deep learning workloads," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018, vol. 10724 LNCS, pp. 3–21.

[21] C. Li, J. Yang, Y. Sun, L. Jin, L. Xu, Z. Cao, *et al.*, "Priority-Based PCIe Scheduling for Multi-Tenant Multi-GPU Systems," *IEEE Comput. Archit. Lett.*, vol. 18, no. 2, pp. 157–60, Jul. 2019.

[22] B. Casper, G. Balamurugan, H. Rong, H. Li, H. Jayatilleka, J. Jaussi, *et al.*, "A 112 Gb/s PAM4 Silicon Photonics Transmitter With Microring Modulator and

CMOS Driver," *J. Light. Technol. Vol. 38, Issue 1, pp. 131-138*, vol. 38, no. 1, pp. 131–8, Jan. 2020.

[23] J. Sun, R. Kumar, M. Sakib, J. B. Driscoll, H. Jayatilleka, and H. Rong, "A 128 Gb/s PAM4 Silicon Microring Modulator With Integrated Thermo-Optic Resonance Tuning," *J. Light. Technol.*, vol. 37, no. 1, 2019.

[24] "MGPUSim: Enabling Multi-GPU Performance Modeling and Optimization - IEEE Conference Publication." [Online]. Available: https://ieeexplore.ieee.org/document/8980359. [Accessed: 29-Jun-2020].

[25] N. Agarwal, T. Krishna, L.-S. Peh, and N. K. Jha, "GARNET: A Detailed On-Chip Network Model inside a Full-System Simulator."

[26] N. Binkert, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, *et al.*, "The gem5 simulator," *ACM SIGARCH Comput. Archit. News*, vol. 39, no. 2, p. 1, Aug. 2011.

[27] T. Ben-Nun, E. Levy, A. Barak, and E. Rubin, "Memory access patterns: The missing piece of the multi-GPU puzzle," in *International Conference for High Performance Computing, Networking, Storage and Analysis, SC*, 2015, vol. 15-20-November-2015.

[28] V. Young, A. Jaleel, E. Bolotin, E. Ebrahimi, D. Nellans, and O. Villa, "Combining hw/sw mechanisms to improve numa performance of multi-GPU systems," in *Proceedings of the Annual International Symposium on Microarchitecture, MICRO*, 2018, vol. 2018-October, pp. 339–51.

[29] L. Zhang, M. Wahib, and S. Matsuoka, "Understanding the Overheads of Launching CUDA Kernels."

[30] G. Michelogiannakis, Y. Shen, M. Y. Teh, X. Meng, B. Aivazi, T. Groves, *et al.*, "Bandwidth steering in HPC using silicon nanophotonics," in *International Conference for High Performance Computing, Networking, Storage and Analysis, SC*, 2019, vol. 15, pp. 1–25.

[31] Y. Li, H. Liu, W. Yang, D. Hu, and W. Xu, "Inter-data-center network traffic prediction with elephant flows," in *Proceedings of the NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, 2016, pp. 206–13.

[32] A. Azzouni and G. Pujolle, "NeuTM: A neural network-based framework for traffic matrix prediction in SDN," in *IEEE/IFIP Network Operations and Management Symposium: Cognitive Management in a Cyber World, NOMS 2018*, 2018, pp. 1–5.

[33] S. Moazeni, S. Lin, M. Wade, L. Alloatti, R. J. Ram, M. Popovic, and V. Stojanovic, "A 40-Gb/s PAM-4 Transmitter Based on a Ring-Resonator Optical DAC in 45-nm SOI CMOS," *IEEE J. Solid-State Circuits*, vol. 52, no. 12, pp. 3503–16, Dec. 2017.

Marjan Fariborz received the B.S. degree from Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran and M.S. degree from Lehigh University, PA, USA in 2014 and 2018 respectively. She has been working on her Ph.D. degree in Electrical and Computer Engineering at the University of California Davis since 2018. Her current research interest is in the area of computer architecture and networks, high-performance computing, and utilizing optical interconnects in state-of-the-art computing systems.

Xian Xiao received the B.S. and M.S. degrees from Tsinghua University, Beijing, China, in 2012 and 2015. He has been working toward the Ph.D. degree in electrical and computer engineering at the University of California, Davis, CA, USA, since 2015. He was a research intern with Nokia Bell Labs in the summer of 2016 and 2017, with Lawrence Berkeley National Laboratory from 2017 to 2018, and with Hewlett-Packard Labs in the summer of 2018.
His current research interest includes silicon photonics, optical interconnects, 2.5D/3D photonic integration, neuromorphic computing.

Pouya Fotouhi received the B.Sc. degree in electrical engineering from the University of Isfahan, Iran, and M.Sc. degree in computer engineering from the University of Delaware, Newark, DE, USA, in 2017. He is currently working toward the Ph.D. degree in computer engineering with the Department of Electrical and Computer Engineering at the University of California, Davis, CA, USA. His research interests include optical interconnects, flat memory systems, and heterogeneous computing

Roberto Proietti received the M.S. degree in telecommunications engineering from the University of Pisa, Pisa, Italy, in 2004, and the Ph.D. degree in electrical engineering from Scuola Superiore Sant' Anna, Pisa, in 2009. He is a Project Scientist with the Next Generation Networking Systems Laboratory, University of California, Davis. His research interests include optical switching technologies and architectures for supercomputing and data center applications, high-spectral-efficiency coherent transmission systems, and elastic optical networking.

S. J. Ben Yoo [S'82, M'84, SM'97, F'07] currently serves as a distinguished professor of electrical engineering at UC Davis. His research at UC Davis includes 2D/3D photonic integration for future computing, communication, imaging, and navigation systems, micro/nano systems integration, and the future Internet. Prior to joining UC Davis in 1999, he was a Senior Research Scientist at Bellcore, leading technical efforts in integrated photonics, optical networking, and systems integration. His research activities at Bellcore included the next-generation Internet, reconfigurable multiwavelength optical networks (MONET), wavelength interchanging cross connects, wavelength converters, vertical-cavity lasers, and high-speed modulators. He led the MONET testbed experimentation efforts and participated in ATD/MONET systems integration and a number of standardization activities. Prior to joining Bellcore in 1991, he conducted research on nonlinear optical processes in quantum wells, a four-wave-mixing study of relaxation mechanisms in dye molecules, and ultrafast diffusion-driven photodetectors at Stanford University. Prof. Yoo received his B.S. degree in electrical engineering with distinction, his M.S. degree in electrical engineering, and his Ph.D. degree in electrical engineering with a minor in physics, all from Stanford University, California, in

1984, 1986, and 1991, respectively. He is Fellow of IEEE, OSA, NIAC and a recipient of the DARPA Award for Sustained Excellence (1997), the Bellcore CEO Award (1998), the Mid-Career Research Faculty Award (2004 UC Davis), and the Senior Research Faculty Award (2011 UC Davis).