Peta-Scale Embedded Photonics Architecture for Distributed Deep Learning Applications

Zhenguo Wu, *Graduate Student Member, IEEE*, Liang Yuan Dai, *Graduate Student Member, IEEE*, Asher Novick, *Graduate Student Member, IEEE, Student Member, OSA*, Madeleine Glick, *Senior Member, IEEE, Fellow, OSA*, Ziyi Zhu, Sébastien Rumley, George Michelogiannakis, *Senior Member, IEEE*, John Shalf, *Senior Member, IEEE*, and Keren Bergman, *Fellow, IEEE, Fellow, OSA*

(Invited Paper)

***Abstract*—As Deep Learning (DL) models grow larger and more complex, training jobs are increasingly distributed across multiple Computing Units (CU) such as GPUs and TPUs. Each CU processes a sub-part of the model and synchronizes results with others. Communication among these CUs has emerged as a key bottleneck in the training process. In this work, we present SiPAC, a Silicon Photonic Accelerated Compute cluster. SiPAC accelerates distributed DL training by means of two co-designed components: a photonic physical layer and a novel collective algorithm. The physical layer exploits embedded photonics to bring peta-scale I/O directly to the CUs of a DL optimized cluster and uses resonator-based optical wavelength selectivity to realize hardware multi-casting. The collective algorithm builds on the hardware multi-casting primitive. This combination expedites a variety of collective communications commonly employed in DL training and has the potential to drastically ease the communication bottlenecks. We demonstrate the feasibility of realizing the SiPAC architecture through 1) an optical testbed experiment where an array of comb laser wavelengths are shuffled by a cascaded ring switch, with each ring selecting and forwarding multiple wavelengths to increase the effective communication bandwidth and hence demonstrating the hardware multicasting primitive, and 2) a four-GPU testbed running a realistic DL workload that achieves 22% system-level performance improvement relative to a similarly-sized leaf-spine topology. Large scale simulations show that SiPAC achieves a 1.4× to 5.9× communication time reduction compared to state-of-the-art compute clusters for representative collective communications.**

Manuscript received 7 December 2022; revised 30 April 2023; accepted 11 May 2023. Date of publication 16 May 2023; date of current version 27 June 2023. This work was supported by ARPA-E ENLITENED Program (project award DE-AR00000843) and the National Security Agency (NSA) Laboratory for Physical Sciences (LPS) Research Initiative (R3/NSA) (Contract FA8075-14-D-0002-0007, TAT 15-1158). *(Corresponding author: Zhenguo Wu.)*

Zhenguo Wu, Liang Yuan Dai, Asher Novick, Madeleine Glick, and Ziyi Zhu are with the Electrical Engineering, Columbia University, New York, NY 10027 USA (e-mail: zw2542@columbia.edu; ld2719@columbia.edu; asn2137@columbia.edu; msg144@columbia.edu; zz2374@columbia.edu).

Sébastien Rumley is with the Electrical Engineering, University of Applied Sciences and Arts Western Switzerland, 2800 Delemont, Switzerland (e-mail: sr01@rumley.pro).

George Michelogiannakis is with the Computer Science, Lawrence Berkeley National Laboratory, Berkeley, CA 94720 USA (e-mail: mihelog@lbl.gov).

John Shalf is with the Computer Science, Lawrence Berkeley National Laboratory, Berkeley, CA 94720 USA (e-mail: jshalf@lbl.gov).

Keren Bergman is with the Electrical Engineering Department, Columbia University, New York, NY 10027 USA (e-mail: bergman@ee.columbia.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JLT.2023.3276588.

Digital Object Identifier 10.1109/JLT.2023.3276588

***Index Terms*—Distributed deep learning, collective communication, silicon photonics, optical interconnect.**

I. Introduction

The wide deployment of Artificial Intelligence (AI) applications has driven the demand for Deep Learning (DL) models of ever increasing accuracy. To achieve higher accuracy, models with more parameters are trained using larger dataset sizes [1]. In the past five years, model sizes have increased by over five orders of magnitude to more than 1 trillion parameters (Fig. 1). Assuming each parameter takes 20 bytes of memory [2], this translates into 20 TB of required memory which well exceeds the capacity of a single computing unit (e.g., a single Nvidia A100 GPU has 80 GB on-chip HBM memory [3]).

In response to this model and dataset growth, Distributed Deep Learning (DDL) is being adopted. In DDL, training models or datasets are partitioned and distributed onto multiple CUs.[1] For example, a training job has been distributed to over 3000 GPUs [13], interconnected through an intermediary network. Current hardware solutions, however, can only provide high-bandwidth connections for a limited *group* of CUs. The Nvidia DGX Station connects 8-16 GPUs using high speed NVSwitches and NVLinks for up to 600 GB/s aggregate bidirectional bandwidth [14]. If a training workload is scaled larger than the memory available in a group of CUs, it will result in *inter-group* communication that relies on 200 Gb/s InfiniBand links which are much slower than the *intra-group* fabric. As Fig. 2 shows, the increase in on-board memory/GPU bandwidth (red and blue curves) has not been accompanied by the same increase in inter-node NIC bandwidth (orange curve). This bandwidth discrepancy severely limits the communication efficiency during the training process [13], currently requiring DDL training to use complex partitioning strategies and communication algorithms to reduce massive data movement via the lower speed *inter-group* links.

However, implementing these complex partitioning strategies could result in a much increased management overhead at model

[1] We use the term *Computing Unit (CU)* to represent a broad range of DL accelerators (e.g., GPU, TPU, NPU, etc.) for generality.

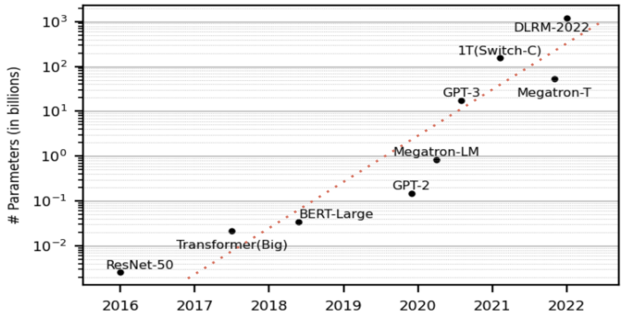

Fig. 1. Deep learning model size trend in terms of model parameters (billions) from 2016 to 2022 [1], [4], [5], [6], [7], [8], [9], [10], [11], [12].

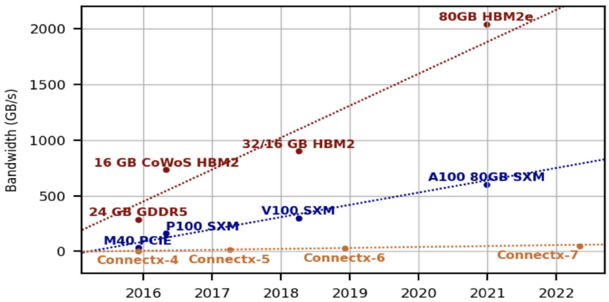

Fig. 2. GPU bandwidth trend from 2015 to 2022 [15], [16], [17], [18], [19], [20], [21].

deployment as well as complex performance trade-offs [13]. To avoid these complexities, one way is to provide uniformly high bandwidth among CUs. However, NVLinks are expensive, energy consuming and distance limited, which prevent the interconnection of CUs located in far-apart servers.

Silicon photonic technologies leveraging CMOS-compatible manufacturing platforms have been proposed as an approach to increase bandwidth density, minimize energy consumption and reduce bandwidth cost in high performance computing and datacenters [22], [23]. Commercial silicon photonic transceivers are already in the market (i.e., Luxtera [24], MACOM [25], and Intel [26]). Meanwhile, Tb/s silicon photonic transceivers have been developed [27] and show the potential of achieving more than 100 Tb/s per waveguide [28], [29]. The use of frequency "comb" sources generating in one shot a large number of wavelengths [29], [30] instead of arrays of single-wavelength lasers [31] is expected to reduce power consumption and area. These transceivers, at the scale of the data-center, are nearly distance independent.

In addition to providing distance independence and massive network bandwidth, silicon photonic technologies possess an inherent property that can also be exploited to improve network performance: the usage of dense wavelength division multiplexing (DWDM) enables wavelength selection, i.e., extracting or inserting specific wavelengths out of or into a set. This can be achieved, for example, using a micro-ring resonator (MRR), which effectively results in a compact wavelength-selective routing operation. Applying this routing operation on the massively parallel DWDM wavelengths results in an all-to-all topology with reduced component count.

In this work, we present a Silicon Photonic Accelerated Compute cluster architecture, *SiPAC*. SiPAC leverages embedded photonic transceivers, ultra-high bandwidth links, and a novel optical multi-wavelength selective switch that maps flows of data to wavelengths in order to reach desired destinations. We further present a novel collective operation algorithm that specifically leverages the capabilities of wavelength-selective optical switching to improve the communication efficiency of large-scale DL training workloads. The major contributions of our work are as follows:

- *Multi-dimensional (MD) All-to-All Connectivity:* We show how to leverage multi-wavelength selective switches and high-bandwidth DWDM links to emulate a MD all-to-all topology with reduced component count. This architecture provides high-bandwidth direct paths for DDL collective operations which also exhibit multi-dimensional communication patterns. We report small-scale system-level testbed results that show a 22% performance improvement relative to a similarly-sized leaf-spine topology on DDL workloads.
- *Multi-Wavelength Selective Micro-ring Based Switch:* We report a testbed experiment using a frequency comb source where an array of its wavelengths are shuffled by a cascaded ring switch, with each ring selecting and forwarding multiple wavelengths to increase the effective communication bandwidth. The experimental results show that the proposed switch design is able to achieve multi-wavelength switching required by the SiPAC architecture for uniformly high communication bandwidth, hence demonstrating the feasibility of our optical architecture.
- *Optimized All-Reduce Collective Algorithm:* We present a novel collective communication algorithm that leverages the MD all-to-all property of SiPAC to achieve both latency and bandwidth efficiency, demonstrating all-reduce as an example. To evaluate the performance of our proposed SiPAC architecture, we conduct detailed packet-level simulations on representative DDL workloads. Large-scale simulation results show that our architecture-collective co-design improves the communication time by a factor of 1.4 to 5.9 compared to the state-of-the-art DL accelerator clusters.

By combining these different contributions, we show that SiPAC is a viable architecture for future DL-optimized computing clusters. A conference abstract based on the testbed and simulation results in this manuscript was presented at ECOC'22 [23] and at OFC'23 [32]. While our conference abstracts provided a brief overview of our research, this article extends the work presented earlier and presents a more comprehensive analysis of the proposed SiPAC architecture. Specifically, we provide more details in the architecture's physical properties, switch analysis, co-designed collective algorithm, and the testbed and simulation setup and results.

II. Background & Related Work

In this section, we characterize the network limitations of current DDL training hardware, describe the key parallelization strategies, and provide background on silicon photonic technologies that enable our architecture.

A. Approaches and Limitations of DDL

DDL relies on parallelization strategies to place a single training task on multiple CUs to cooperatively complete the training task. In Data Parallelism (DP), each CU keeps a full copy of the entire training model and receives a partitioned batch of the input dataset. In Model Parallelism (MP), each CU keeps a full copy of the dataset and receives a partitioned training model. The model can be partitioned horizontally or vertically, resulting in *pipeline parallel* (PP) and *tensor parallel* (TP), respectively. Hybrid Parallelism (HP) combines both DP and MP to parallelize both the models and the dataset. Collective operations (e.g., all-reduce and all-to-all) dominate the communication traffic in the synchronization stage of each of these parallelism strategies. We mainly focus on all-reduce operations in this study as other collective operations such as reduce-scatter and all-gather can be derived from all-reduce (i.e., all-reduce can be decomposed into reduce-scatter and all-gather). Various all-reduce algorithms, such as ring-based [33], hierarchical ring-based [34], and mesh-based [35], have been proposed in the past with different latency vs. bandwidth trade-offs. Other topology-specific all-reduce algorithms include HiPS [35] and BML [36] which are specialized collectives designed for a specific topology.

Many specialized hardware accelerators have been proposed to accelerate DDL. Some commercially available examples include Nvidia's DGX [37] and Google's Cloud TPU [38]. Large-scale models have been reported to be trained on these systems (e.g., Megatron-LM was trained on 3072 Nvidia A100 GPUs [13]). However, past work has shown high communication cost for these collective operations when the size of the MP or DP cluster scales beyond a single server (e.g., a DGX-A100 server) since the traffic needs to go through inter-server links which are much slower than the intra-server links [13]. Benchmarks from the Sierra supercomputer [39] reported DDL communication time to be more than 10× the computation when DDL workload is trained on more than 256 GPUs [40].

Past work has taken different approaches to tackle this limitation. Some common examples include 1) methods to overlap communication with computation [9], [41], 2) compressing messages [41], [42], [43], 3) using host memory as a swap mechanism to keep the training cluster small [2], and 4) co-designing the collective operation algorithm with the network topology or switch to achieve more efficient training [44], [45], [46], [47].

B. Silicon Photonic Technologies for DDL

The problem can also be solved by providing uniform high bandwidth. In this context, an emerging trend is to incorporate embedded silicon photonic (SiP) technologies in the network as a means to achieve peta-scale high bandwidth interconnects [23]. Silicon photonic transceivers co-packaged with compute chips can provide an energy-efficient scaling of multi-Tb/s/mm^2 bandwidth densities [48], [49]. An example of a commercial SiP interface is TeraPHY [27] that supports up to 2 Tb/s bandwidth per chiplet. Recently, a promising Kerr frequency comb-driven SiP transceiver [29] has been reported; it leverages a frequency comb [31] for a DWDM light source and uses (de-)interleavers to split and combine wavelength channels in order to scale up the data transmission bandwidth within a single fiber.

TABLE I
LIST OF MATHEMATICAL SYMBOLS AND THEIR RESPECTIVE DESCRIPTIONS

Notation	Description
p	Number of CUs in the topology
r	Switch radix
l	Level in a hierarchical topology, $l \in [0, L-1]$ where $L = max(l) + 1$
b	Bandwidth per CU-pair under the same WSS
B	Total bandwidth per WDM link
w	Number of transmitter wavelengths
u	Bandwidth per wavelength
λ_k^{ij}	Wavelength $k \in [0, w-1]$ sent from CU i to CU j. $i, j \in [0, r-1]$
k	Number of CUs in a group in the hierarchical topology
α	Link latency per unit step
β	Inverse link bandwidth
n	Message size (in bytes) transferred per CU per time step

However, many past works that proposed using silicon photonic technologies for DDL training [50], [51], [52], [53], [54] have placed more emphasis on designing networks using Optical Circuit Switches (OCS) to dynamically reconfigure the network topology to cater to different DDL traffic demands. For example, SiP-OCS [53] co-designs the model partitioning and device placement with a specialized network architecture that employs a layer of reconfigurable OCS. TopoOPT [54] also leverages the reconfiguration ability of the OCS and co-designs an alternating optimization technique to find the best network topology and routing plan together with the parallelization strategy. While spatial OCSs can simultaneously switch all the wavelength channels, they lack the ability to selectively route wavelengths to realize full switching capability in the wavelength domain which is desirable to further increase switching granularity for DWDM architectures. In this work, we do not rely on the reconfigurability of the OCSes and instead leverage the wavelength selectivity of multi-wavelength optical data movement to realize a photonic architecture capable of efficiently accelerating collective communication in DDL.

III. SiPAC Architecture

In this section, we provide a detailed description of the proposed SiPAC architecture. A list of recurring mathematical symbols can be found in Table I.

A. Topology Design

The SiPAC architecture leverages the multi-wavelength selective property of the MRR-based WSS to realize a MD all-to-all topology following a BCube-like physical topology [55] that has been shown to have a low network diameter and high capacity for collective communication patterns involved in DDL training [36], [55]. BCube(r, l) is a recursively defined, server-centric network topology, where r is the switch radix and l is the level in the topology ($l \in [0, L-1]$ where $L = max(l) + 1$ is the total number of levels). A base unit BCube$_0$ is constructed

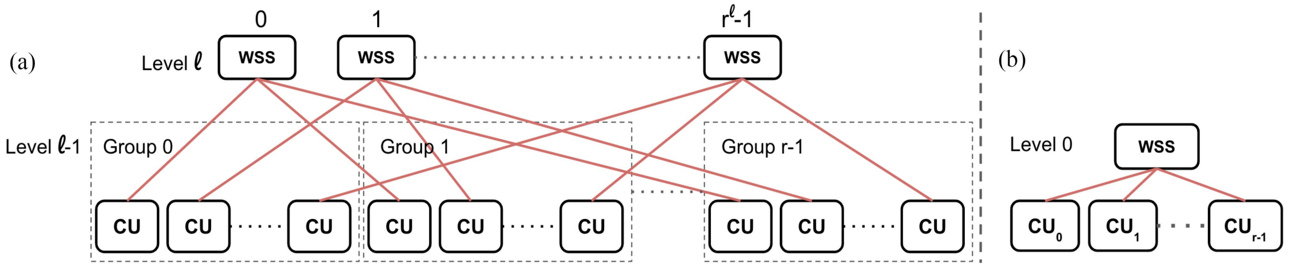

Fig. 3. (a) SiPAC architecture based on the recursive BCube topology [55]. The lth level is constructed from r^l r-port switches and r $(l-1)$th level groups. (b) The base unit of the SiPAC topology where r CUs are connected to a WSS.

from connecting r servers to an r-port switch. For SiPAC, instead of using servers as endpoints, we replace each server with a disaggregated CU, equipped with L embedded optical transceivers. In SiPAC, we also replace each electronic packet switch (EPS) with a multi-wavelength selective switch (WSS) as described in Section III-B. The rest of the physical topology is constructed similarly to a BCube, replacing each EPS with WSS at each level.

A general SiPAC$_l$ $(l \geq 1)$ of level l is therefore constructed from r^l r-port switches connecting r SiPAC$_{l-1}$ s, totaling $p = r^{l+1}$ CUs and L levels of switches, as shown in Fig. 3(a). CUs in a SiPAC$_l$ have $L = max(l) + 1$ optical ports and are connected to an optical switch in each of the L levels. L is typically small since the number of endpoints grow exponentially as a function of L. For example, using radix-16 WSSes, we could achieve a topology size of 256 for $L = 2$ and 4,096 for $L = 3$. Since the diameter of this topology is also L [55], the resulting SiPAC topology has a low diameter. To be more flexible in terms of the number of endpoints, irregular SiPACs can be built using switches of different radices similar to how partial BCubes are built in [55].

In addition to having a low diameter, the SiPAC topology provides a direct light path between any pair of CUs by enabling arbitration-free all-to-all connections for CUs connected to the same WSS. This enables each CU to send to its directly connected neighbors without contention. Details of the WSS are provided in Section III-B. This effectively achieves a generalized HyperCube topology [56] with $l + 1$ dimensions, allowing each CU to communicate directly with $(l + 1)(r - 1)$ other CUs with a reduced link count and transceiver count.

Compared to other architectures that also leverage silicon photonic technologies for DDL training [53], [54], [57], our work provides simpler network design as it does not require active topology reconfiguration via switch or wavelength tuning. Moreover, the low diameter property and the ability of each CU to directly communicate with many other CUs provide many redundant shortest paths between any CU pair. By observing the multi-dimensional nature of DDL traffic pattern when employing multiple parallelization strategies, the proposed architecture enables efficient communication that fits well with the DL application demand.

B. Silicon Photonic Technology for SiPAC

Directly integrating optical transceiver ports onto chip interposers obviates the need for expensive NICs [58]. The total bandwidth of a DWDM link B depends on the number of wavelengths w per transmitter and the per-wavelength bandwidth u, giving $B = wu$. The resulting interconnection network allows for transparent optical switching and therefore achieves direct CU-to-CU communication without any bandwidth variation that appears in commercial accelerator clusters. In conjunction with the co-designed collective algorithm, the packet-switch-less design mitigates intermediate packet buffering and reduces in-network queuing delays.

The SiPAC architecture also relies on the multi-wavelength selecting property of a MRR-based WSS to realize the MD all-to-all topology while ensuring high bandwidth per CU-pair. To this end, we design a novel MRR based multi-wavelength selective switch that exploits the periodic property of the free spectral range (FSR) of the MRRs, extending past works that used AWGR [59]. By carefully engineering the FSRs, each MRR has the ability to drop multiple wavelengths thus increasing the effective bandwidth per ring.

In SiPAC, each CU is able to directly communicate with every other CU that is connected to the same WSS with uniform high bandwidth. Each CU is equipped with the same transmitters, so the wavelengths being transmitted from the input ports are the same. We tune the MRRs in the switching cell so that distinct wavelengths are dropped to each output bus. The w wavelengths, $\lambda_k^{i,j}, k \in [0, w-1], i, j \in [0, r-1]$ being transmitted using an $r \times r$ MRR switch are divided into r groups. Each group $g \in [0, r-1]$ contains w/r wavelengths with the wavelength number $k \in [0, w-1]$ separated by an integer multiple of r. Each group of wavelengths is also labeled with their input port (i) and output port (j). The wavelengths from each input port are interleaved at the switch in a way so that the drop bus for each output port contains all different wavelengths. This can be done by tuning the rings so that the gth group of wavelengths from input port i are dropped at output port j where $j = (g + i)mod(r)$. Fig. 4 shows an example of a 3 × 3 cascaded ring structure that separates the incoming transmitter wavelengths per CU $(w = 9)$ into subgroups and recombines the interleaved wavelengths into common output buses. It shuffles the input wavelengths to different outputs and effectively achieves the optical multicasting functionality. We show this switch design as an example of MRR-based switch that can achieve both all-to-all and multi-wavelength selective switching. Other scalable MRR-based switch design include [60], [61].

The amount of allocated bandwidth b between each CU pair connected to a common WSS depends on the number of transmitter wavelengths w and the number of connections to the WSS

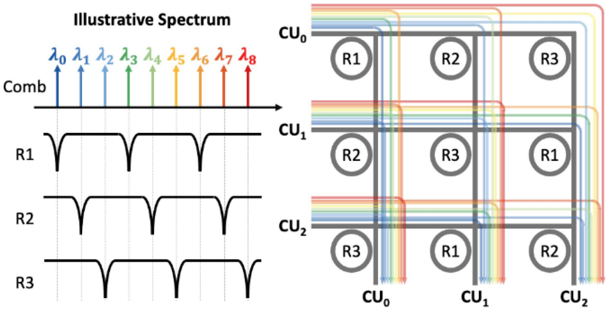
Illustrative Spectrum
λ_0 λ_1 λ_2 λ_3 λ_4 λ_5 λ_6 λ_7 λ_8
Comb
R1
R2
R3
CU$_0$
CU$_1$
CU$_2$
R1 R2 R3
R2 R3 R1
R3 R1 R2
CU$_0$ CU$_1$ CU$_2$

Fig. 4. Example schematic of wavelength multiplexing for a 3 × 3 WSS and 9 wavelengths per transmitter ($w = 9$). Each color represents a different group g of wavelengths. The colors are interleaved (by tuning the rings) vertically so that distinct wavelengths are dropped to each output port.

r in the same level: $b = \frac{wu}{r} = \frac{B}{r}$. To scale up the bandwidth b, we can therefore increase the number of comb lines w and use a higher data-rate per wavelength. We note that current comb laser sources have already reached 160 wavelengths [30] while silicon photonic modulators have achieved 128 Gb/s data rate per wavelength [62]. However, achieving these numbers requires addressing several challenges, such as managing insertion loss, temperature variations, engineering the spacing of the MRR's FSRs to align with the comb and scaling the switch port count. One approach to mitigate the effect of temperature fluctuations is to use thermal stabilization algorithms, which can actively monitor and maintain resonance wavelengths and switching states [63]. To mitigate the losses, racetrack micro-ring designs can reduce insertion loss due to off-resonance rings to 0.02 dB per switching cell [64]. The physics and fabrication of MRRs have been extensively studied [65], and constructing MRRs with effectively no dispersion can mitigate the misalignment between comb lines and resonances due to varying FSRs [66]. Scalable MRR switch designs have been shown to be feasible for port counts up to 128 [61]. Prior work on a 64-channel system fitting into a compact 0.4 mm^2 area has demonstrated peta-scale capabilities [67], providing a bandwidth density of 5 Tb/s/mm^2 when operating at 16 Gb/s/λ. These benchmarks can serve as references for future scalability while acknowledging the challenges involved.

IV. Testbed Experiments

We conduct two small-scale testbed experiments to demonstrate 1) the feasibility of the proposed multi-wavelength selection using comb wavelengths and MRRs and 2) the topological advantages of SiPAC. Due to resource limitations, we first show four comb wavelengths being dropped by two rings in a 1 × 8 MRR-based switch. Then, we use another 4 × 4 MRR-based WSS to route two wavelengths per GPU and form a SiPAC(r = 2, L =2) architecture.

A. Optical Testbed Experiment

Our experimental demonstration highlights a hardware implementation for achieving multi-wavelength optical switching via a single WSS cell, together with a high bandwidth density Kerr frequency comb source. In particular, we focus on wavelength-selective switching of multicast signals, as SiPAC relies on the multi-wavelength selecting property of WSS to realize the MD all-to-all topology while ensuring enough bandwidth per CU-pair. This multicast capability also supports the SiPCO algorithm to enable efficient collective operations.

Our testbed setup is illustrated in Fig. 5(a)). A continuous-wavelength tunable-laser-source (CW-TLS) centered on 1561.42 nm is amplified, via Erbium Doped Fiber Amplifier (EDFA), to about 200 mW optical power [31] and is used to pump a silicon-nitride Kerr comb chip (Fig. 5(b)), which generates evenly-spaced lines at 201.5 GHz (≈ 1.6 nm) intervals with a spectral flatness suitable for data communication [31], [68]. The comb chip converts the pump laser input into multiple wavelengths that are filtered by an optical bandpass filter (OBF) to include 22 channels (Fig. 5(c)). The output of the OBF is modulated with a 10 Gb/s PRBS31 via a linear reference modulator, and coupled into the cascaded 1 × 8 MRR switch (Fig. 5(d)). Each MRR has a FSR of 14.41 nm and can drop multiple channels. The dropped signals are coupled out of the switch and the signals are amplified through an EDFA to compensate for coupling and test equipment insertion losses. Fig. 5(e) and (f)) show the modulated carrier and the surrounding sidebands. Polarization controllers (PC) are used to maximize the optical power coupled into the chips, and Variable Optical Attenuators (VOAs) are used to reduce the optical power at the photo-detector (PD). We measure the optical spectra with an optical spectrum analyzer (OSA) with 10 MHz resolution. Open eyes were observed for proper operation (Fig. 5(h)).

To establish the efficacy of dropping multi-wavelength signals using a MRR, the first ring (R1) in our cascade microring switch is thermo-optically tuned to select the comb line at 1534.07 nm. Due to its 14.41 nm FSR, this also selects the channel at 1548.48 nm. In Fig. 5(e)), the optical spectrum captured at the drop port of R1, we can see that our channels of interest dominate over all other lines, with a crosstalk suppression of 13.3 dB between our selected channel and an adjacent unselected one. Very similar performance can be observed for the next ring in the switch (R2), which is tuned to select wavelengths 1532.48 nm and 1546.87 nm; lines directly adjacent to the ones selected by R1. We observe open eyes in all cases (Fig. 5(h)), although a small variance in the signal-to-noise ratios (SNRs) is observed: 6.02 dB to 7.23 dB, an effect that can be attributed to the uneven power of the comb lines. The results demonstrate the feasibility of the multi-wavelength selective MRR switch to implement the SiPAC design.

B. System Testbed Experiment

We then demonstrate the system-level performance of a small-scale SiPAC($r = 2, L = 2$) architecture using 4 Nvidia Tesla M40 GPUs with RoCEv2 enabled Mellanox ConnectX-4 NICs (Fig. 6(b)). The testbed setup is shown in Fig. 6(a)). To emulate parallel wavelength transmission, each GPU is configured to have a virtual bridge equipped with two 10 Gb/s SFP+ transceivers sending at two different wavelengths (1550.12 nm & 1556.55 nm). We use a separate 4 × 4 MRR-based WSS to

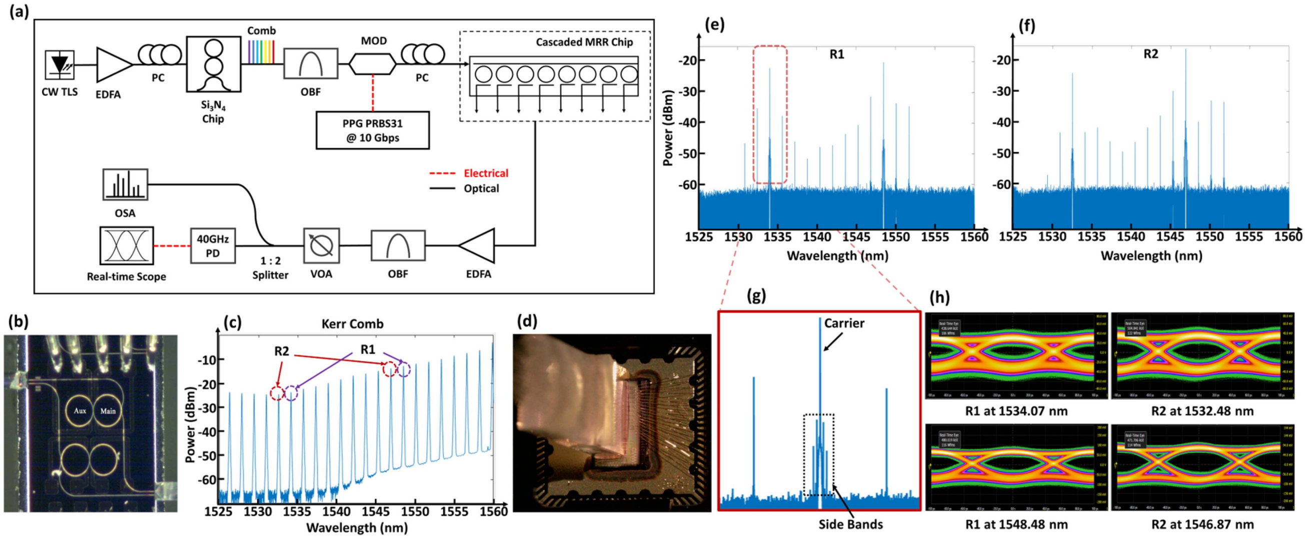

Fig. 5. (a) Schematic of the experimental setup. A Kerr frequency comb (b) is used to generate evenly-spaced lines at 201.5 GHz intervals (c). The signal is modulated with a 10 Gb/s PRBS31 via a linear reference modulator, and coupled into a cascaded 1 × 8 MRR switch (d). The optical spectra in (e) and (f) show the focused signal in our wavelength range of interest. Open eyes, required to ensure proper operation, are observed for both wavelengths (h).

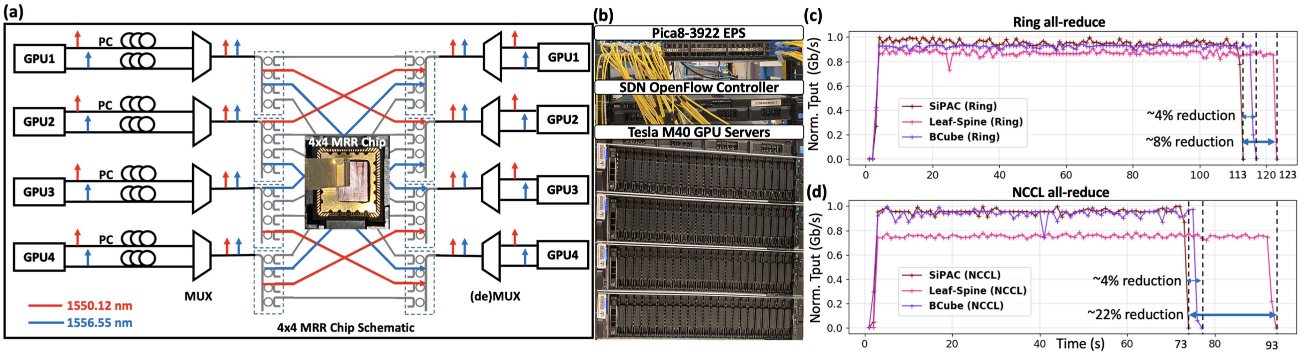

Fig. 6. (a) Schematic of the SiPAC(r = 2, L = 2) testbed setup. (b) GPU servers connected to a EPS and a SDN controller. Throughput of the injection port under (c) ring all-reduce and under (d) NCCL all-reduce.

realize the wavelength shuffling and recombining in the optical layer. Optical (de)multiplexers are used to combine and separate wavelengths coming out and going into GPU bridges. We use TensorFlow to run a distributed MobileNetV2 workload using both the ring and NCCL collective algorithm and compare SiPAC's performance with similarly sized, EPS-based leaf-spine and electronic BCube(r = 2, L = 2) topology. In the leaf-spine topology, one spine switch connects to two aggregation switches, each of which is connected to two GPUs. We run each training workload for two epochs with a batch size of 128. The network throughput is captured using the Ryu SDN OpenFlow monitoring program (Fig. 6(c) and (d)). Under ring-based all-reduce, SiPAC is able to achieve a 4% and 8% job completion time (JCT) reduction relative to BCube and leaf-spine, respectively. When using NCCL all-reduce, the JCT reduction is further increased to 22% for the leaf-spine topology as the NCCL tree-based algorithm can better leverage the multi-port property of the SiPAC architecture. The JCT improvement of SiPAC over BCube remains constant due to the similar physical connections, with BCube having more in-network buffering delay as the difference. The expected improvement over BCube increases with larger system sizes and link bandwidths due to larger in-network queuing delays at each switch with more connected endpoints.

V. Co-Designed Collective Algorithm

To fully leverage the MD all-to-all property of the SiPAC network architecture, we present a novel collective communication algorithm. We note that every collective operation involves a trade-off between latency and bandwidth: the most aggressive algorithm consists of simply sending the data to every destination at the same time (potentially exploiting a physical all-to-all topology) which guarantees minimum latency. Other approaches will trade reduced network load for higher latency. Servers in state-of-the-art HPC topologies often connect to the network using a single NIC, which would result in congestion when sending to many different destinations. In this trade-off

Algorithm 1: SiPCO All-Reduce Algorithm.

```
Input: $r, L$
for each $CU_i, i \in [0, r^L - 1]$ do
    Partition the local message into $C = rL$ chunks
    Label each $r$ chunks with group number
    $g \in [0, L - 1]$
    for each link $l \in [0, L - 1]$ connected to $CU_i$ do
        Send chunks in group $g = (l)mod(L)$ using link $l$
    end for
end for
for step $s \in [1, L]$ do
    for each $CU_i, i \in [0, r^L - 1]$ do
        for each link $l \in [0, L - 1]$ connected to $CU_i$ do
            Bcast the chunk in group $g = (s + l)mod(L)$
            using link $l$
        end for
    end for
end for
```

TABLE II
Summary of the Latency Cost for Different All-Reduce Algorithms

	Latency	Bandwidth
Ring	$2(p-1)\alpha$	$2\frac{p-1}{p}n\beta$
Mesh	2α	$2\frac{p-1}{p}n\beta$
H-Ring	$[4(k-1)+2(p/k-1)]\alpha$	$\frac{[4(k-1)+2(p/k-1)]}{k}n\beta$
SiPCO	$(L+1)\alpha$	$\frac{(L+1)(r-1)}{Lr}n\beta$

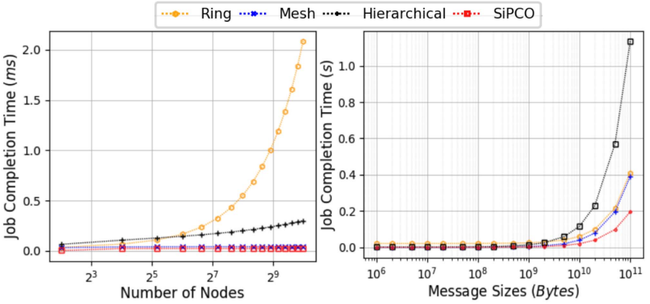

Fig. 7. Visualization of the all-reduce latency cost for representative parameters across different network sizes (at 100 MB message size) and different message sizes (at 1024 CUs). Note that these analyses are based on logical collective topologies.

context, we evaluate the algorithm presented below with the well-known latency cost model [69]: $\alpha + n\beta$ where α is the link latency per unit step, β is the transfer time per byte (inverse of the bandwidth), and n is the size of the message being transmitted on a link per unit step. Note here that the latency cost analyses are based on logical topologies with uniform link latency and bandwidth, and the performance of the algorithms on different physical topologies can vary based on their network properties.

A. SiPAC Collective Algorithm (SiPCO)

SiPCO is a collective algorithm that is co-designed with the SiPAC topology. Since all-reduce is the dominant operation in DDL communication, we will describe how SiPCO all-reduce works as an example. The SiPCO all-reduce algorithm optimizes for both latency and bandwidth by building on the hierarchical and mesh all-reduce. Unlike prior multi-stage hierarchical collectives that send messages along a single dimension during each stage [34], [70], our co-designed algorithm fully uses all the available wavelengths in all dimensions at each time step. It also eliminates the need for additional message relaying by ensuring that each transmitted message chunk requires an associated operation at every step.

The algorithm contains $L + 1$ steps, one more than the number of levels in the physical topology. Since L is typically small, the latency cost is effectively constant. To fully utilize the L links (and L transceivers) connected to each CU, the local message on each CU is partitioned into $C = rL$ chunks, each with size $\frac{n}{rL}$ bytes. We then organize them into L groups of r chunks and label the chunks in each group $g \in [0, L - 1]$. In the first step, each CU sends chunks in group $g = (l)mod(L)$ using link $l, \forall l \in [0, L - 1]$ to r different destination CUs connected to the same WSS (similar to the scatter stage in the mesh-based all-reduce). Each chunk of data is carried using w/r interleaved wavelengths. Each CU then performs a reduction on all the received chunks from other connected CUs to complete the first step. In the next $s \in [1, L]$ steps, we repeat step 1 but rotate the L groups of r chunks through the L connected links so that chunks in group $g = (s + l)mod(L)$ are sent through link l. Instead of sending different chunks, we leverage the broadcasting capability of the WSSes to broadcast the already reduced chunk in group g which means that each source CU transmits the same chunk to r destination CUs each with w/r wavelengths. After each subsequent step, every CU acquires L chunks that contain contributions from r additional CUs. Therefore, after one full rotation of $L - 1$ rounds, each CU now has L chunks that contain contributions from r^L CUs. In the last step, the L fully reduced chunks in group $g = (L + l)mod(L)$ are broadcast using links in level l so that each CU now has rL chunks from r^L CUs, thus completing the all-reduce process. The full algorithm is shown in Algorithm 1. Note that when $L = 1$, meaning all the CUs are connected to the same WSS, this algorithm reduces to the mesh-based all-reduce.

The overall latency of this algorithm can be characterized as $(L + 1)(\alpha + (r - 1)\frac{n}{rL}\beta)$ since each link transmits $(r - 1)\frac{n}{rL}$ bytes in each of the $L + 1$ steps. The latency term α is constant and the bandwidth term is close to optimal as we scale to larger r's. A summary comparing the latency cost of the SiPCO all-reduce with other collective algorithms such as ring [33], [71], mesh [35], [72], and hierarchical ring [34] can be found in Table II. A visualization of Table II for representative parameters can be found in Fig. 7. We evaluate these algorithms using job completion time (JCT): the amount of time it takes for a communication job to finish. A lower job completion time indicates better algorithmic performance. We assume a logical topology best suited for each algorithm with uniform link latency of $\alpha = 1\mu s$ and a uniform link bandwidth of 512 Gb/s ($\beta = 1/(512$ Gb/s)). For comparison across network sizes, the message size is set to be 100 MB. For comparison across message sizes, the network size is set to be 1024 CUs. We see that the SiPCO

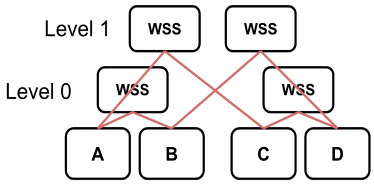

Fig. 8. Example SiPAC topology with $r = 2, L = 2$.

TABLE III
EXAMPLE OF SIPAC-ALLREDUCE SYNCHRONIZATION FOR $r = 2, L = 2$. THE CHANGES IN EACH STEP ARE MARKED IN BOLD

A	**B**	**C**	**D**
A0	B0	C0	D0
A1	B1	C1	D1
A2	A2	C2	C2
A3	B3	C3	D3

(a) In the start, the message in each CU is divided into $rL = 4$ parts.

Step 0		**A**	**B**	**C**	**D**
$l = 0$		A0	**A0**+B0	C0	**C0**+D0
		A1**+B1**	B1	C1+**D1**	D1
$l = 1$		A2	B2	**A2**+C2	**B2**+D2
		A3+**C3**	B3+**D3**	C3	D3

(b) In step 0, CUs exchange the first $r = 2$ chunks using links in level $l = 0$ and exchange the next $r = 2$ chunks using links in level $l = 1$.

Step 1		**A**	**B**	**C**	**D**
$l = 1$		A0	A0+B0+**C0+D0**	C0	**A0+B0**+C0+D0
		A1+B1+**C1+D1**	B1	**A1+B1**+C1+D1	D1
$l = 0$		A2	B2	A2+**B2**+C2+**D2**	**A2**+B2+**C2**+D2
		A3+**B3**+C3+**D3**	**A3**+B3+**C3**+D3	C3	D3

(c) In step 1, chunks that have been received and reduced in step 0 are broadcast using links in levels different from the previous step.

Step 2		**A**	**B**	**C**	**D**
$l = 0$		**A0+B0+C0+D0**	A0+B0+C0+D0	**A0+B0+C0+D0**	A0+B0+C0+D0
		A1+B1+C1+D1	**A1+B1+C1+D1**	A1+B1+C1+D1	**A1+B1+C1+D1**
$l = 1$		**A2+B2+C2+D2**	**A2+B2+C2+D2**	A2+B2+C2+D2	A2+B2+C2+D2
		A3+B3+C3+D3	A3+B3+C3+D3	**A3+B3+C3+D3**	**A3+B3+C3+D3**

(d) In the final step, chunks that have been fully reduced are broadcast using the same links as in step 0.

curve remains relatively constant across topology sizes since its latency cost does not scale with increasing node number. SiPCO also exhibits better scaling performance with increasing message sizes due to its optimized bandwidth latency cost.

B. SiPCO Example

We illustrate through a simple example how this algorithm works on a SiPAC topology with $r = 2, L = 2$ as shown in Fig. 8. In the beginning, the message in each of the $r^L = 4$ computing nodes is split into $C = rL = 4$ chunks as shown in Table III(a). During step 0 (Table III(b)), CUs exchange the first $r = 2$ chunks using $l = 0$ links and exchange the second $r = 2$ chunks using $l = 1$ links. After step 0, each node has $L = 2$ partially reduced chunks that have contributions from $r = 2$ CUs (e.g., a has the second and last chunk partially reduced). In the next $L - 1 = 1$ step (Table III(c)), each CU broadcasts the partially reduced chunk to achieve L fully reduced chunks. In the last step (Table III(d)), each CU broadcasts the fully reduced chunks and finishes the all-reduce.

The algorithmic principle of SiPCO is to maximize the utilization of link resources at each timestep, making it applicable for other collective operations as well (e.g., all-to-all). The set of SiPCO algorithms could be implemented as function algorithms in libraries similar to NCCL [73] or MPI [74].

VI. SYSTEM SCALE EVALUATION

A. Methodology

To demonstrate the scalability of our proposed architecture, we conduct detailed packet-level simulations. We use Netbench, an event-driven, packet-level simulator [75] to evaluate the performance of the SiPAC architecture. We extended Netbench to support 1) topologies with varying link latencies and bandwidths and 2) traffic with blocking flow starting times that are found in collective communications.

1) Topologies: We compare the performance of the SiPAC topology against a few other state-of-the-art DL cluster topologies. For a fair comparison, we normalize the topologies using the per-CU bandwidth as described next. Unless specified, we assume the per-hop link latency to be 1 μs.

SuperPod [37]: The basic units of DGX-SuperPod are DGX-A100 servers in which eight A100 CUs are connected to an array of 6 NVSwitches using NVLinks [76]. Multiple DGX-A100 servers are then interconnected through a two-layer leaf-spine fat-tree network using eight 200 Gb/s InfiniBand host channel adapters (HCA) per node [37]. We therefore fix the inter-node bandwidth at 8×200 Gb/s $= 1.6$ Tb/s. We assume a $9\mu s$ NVLink latency [77] and 120˜ns InfiniBand switch latency [78]. We characterize the per-CU bandwidth here to be the sum of all intra-server (i.e., NVLink) bandwidths coming out of a single CU, similar to [53].

2D-Torus [38]: Google's Cloud TPU v3 Pod system directly interconnects TPUs in a 2D toroidal mesh network [38] with uniform link bandwidth and latency. For systems with sizes that are not integer squares, we pick integer sizes for each dimension with minimum differences to achieve the targeted topology size. The per-CU bandwidth here is characterized as the total bandwidth a single CU has with its four neighbors.

BCube [55]: Since the SiPAC architecture is inspired by the BCube physical topology, we evaluate a BCube built with EPSes. While we choose r and L to best fit the required system size for both BCube and SiPAC, we limit $r \leq 32$ and $L \leq 3$ to achieve a realistic WSS radix [79] and similar number of per-CU optical interfaces as the other topologies. The per-CU bandwidth for both architectures is characterized as the total bandwidth a single CU has with all connected switches (EPS for BCube and WSS for SiPAC) in each of the L layers.

2) Component Count and Energy Cost: Fig. 9 shows the link, transceiver, and switch count as a function of the network size for the described topologies. We also consider a few state-of-the-art HPC topologies such as canonical Dragonflies [80] (with one inter-group link per EPS) and three-level fat trees. The switch radix is set to be approximately the same across different topologies at similar topology sizes. We notice that the traditional topologies usually require fewer links and switches but more transceivers since they use EPS-based ToRs to aggregate many

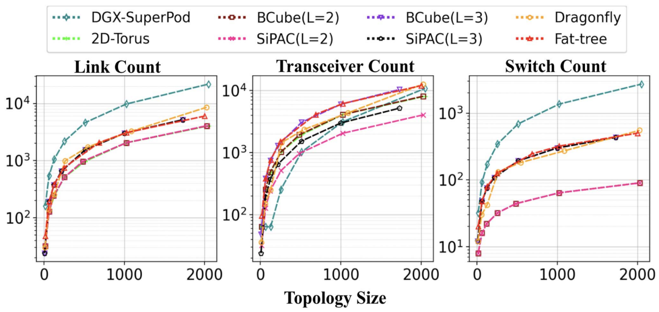

Fig. 9. Link, transceiver and switch count as a function of network size for various topologies.

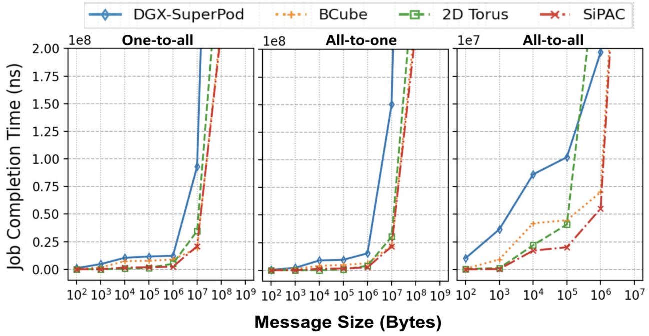

Fig. 10. JCT of primitive collective communications for 512 CUs across different message sizes.

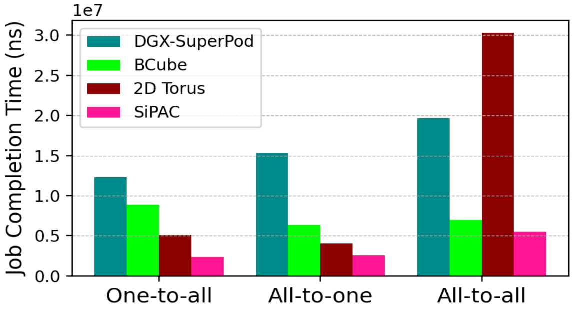

Fig. 11. JCT of primitive collective communications for 512 CUs at 1 MB message size.

server endpoints. The DL accelerator clusters, on the other hand, involve less aggregation and therefore require more links to directly connect CUs together. This reallocation of bandwidth from aggregation layers to direct connections increases the direct bandwidth among CUs, facilitating more efficient communication in collective operations. We note that SiPAC($L = 3$) requires a similar number of switches as a 3-level fat tree, but SiPAC($L = 2$) would require much fewer switches at the cost of higher switch radix. We observe that SiPAC($L = n$) has the same number of physical links as BCube($L = n$) and nD-Torus for a given number of dimension (n) and has fewer links than a similar sized SuperPod. SiPAC also requires fewer transceivers at larger topology sizes due to its usage of transparent optical switches. Therefore, any demonstrated performance that is similar to or better than the state-of-the-art architectures is achieved with a reduced component count.

In terms of the energy cost, it has been demonstrated by TeraPHY [81] that the all-inclusive energy efficiency for a 400 Gb/s optical link can be less than 5 pJ/bit. This calculation includes all the optics involved as well as all associated electronics (ADC/DACs, along with SerDes, drivers, and clocking/distribution). This pJ/bit value can be lowered further as ADCs and DACs become increasingly more energy efficient with improved fabrication technologies [82], and by moving towards low-resolution components [83]. Electrical interconnects of similar bandwidth density have shown higher efficiency, at 1.17 pJ/bit [84], but due to the rapid degradation in signal quality for electrical I/Os at high data rates, the reach of such connections is less than 10 cm. Photonics are relatively distance independent and thus are well suited for interconnecting large numbers of discrete computing resources within a large HPC system. Furthermore, using comb lasers allows generating the full optical spectrum with a single component requiring thermal tuning, reducing energy consumption compared to using an array of discrete lasers.

3) Workloads: We evaluate the performance of all architectures using three main types of workloads as described below.

Primitive Collectives: Many HPC/DDL applications exhibit all-to-one (incast), one-to-all (broadcast), or all-to-all traffic patterns under various parallelisms [54]. Therefore, we first test how different topologies perform under these general traffic patterns without assuming any specific collective algorithm. For one-to-all and all-to-one traffic, we randomly select a CU in the topology to be the root CU.

Hybrid Collectives: Many large-scale DDL training workloads employ both MP and DP to achieve better efficiency. Therefore, we also model the type of traffic pattern involved in hybrid parallelism (HP). We study a similar HP strategy as described in [53] where p computing nodes are divided into d DP groups of m MP nodes. At each iteration, each group of m MP nodes synchronize among themselves using the all-to-all collective and then synchronize across the d DP groups using all-reduce [85]. In the experiments, we employ mesh-based collectives for the intra-MP group all-to-all communication and ring-based collectives for the DP all-reduce on SuperPod and BCube. We employ ring-based algorithms for both intra-group MP and inter-group DP for 2D Torus. For the SiPAC architecture, we apply the SiPCO collective algorithm for both MP all-to-all and DP all-reduce communication.

Deep Learning Workloads: Our evaluations of DL workloads are based on open-sourced application communication taskgraphs from [54]. The applications simulated are VGG [86], Candle [87], and Transformer (BERT) [6]. For each of these workloads, we simulate iterations of collective communication within a 2 s window, with message sizes extracted from the taskgraphs. We experiment with ring-based, mesh-based and hierarchical ring-based collective algorithms on all the architectures. For hierarchical-based all-reduce, we set the group size, k, to be equal to the number of CUs in a physical group or dimension in the topology. For SiPAC and BCube architectures, we employ the SiPCO all-reduce algorithm.

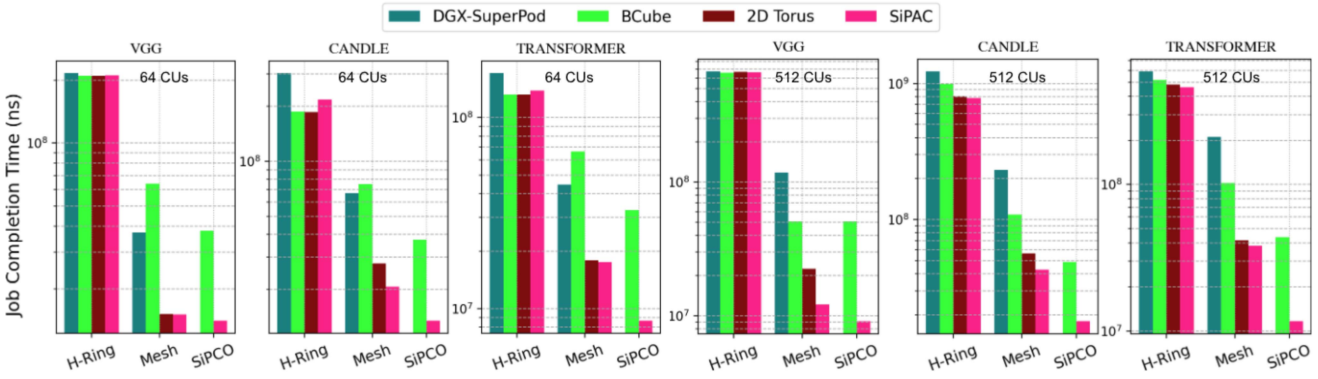

Fig. 12. JCT of different topology-collective combinations at $p = 64, 512$ for three types of DDL workloads.

We model only communication, not computation, in our simulations as communication accounts for an increasingly larger proportion of total training time, exceeding 50% for larger topologies. [40]. We assume that any improvement in communication time could help enhance the overall training efficiency when computation and communication cannot be efficiently overlapped which has been shown to be the case for large scale training (i.e., $p \geq 64$ for strong scaling and $p \geq 256$ for weak scaling) [40].

B. Results

1) Primitive Collective Communication: For this experiment, the topology size is set to be 512 and the per-CU bandwidth is set to be 2048 Gb/s, which corresponds to $r = 8$ and $L = 3$ for SiPAC. The configuration is based on the feasible assumption that the MRR-based switching architecture can scale, as demonstrated in [61], [67]. We vary the message size from 100 B to 1 GB, following the same order of magnitude as some common DDL workloads [88], [89], [90]. As shown in Fig. 10, the SiPAC topology outperforms the other topologies at small message sizes and can support larger message sizes before saturation. Since SiPAC requires fewer transceivers and switches at the same topology size, it suggests that SiPAC can achieve similar network performance at a lower component count.

To show more details on the relative performance at medium message sizes, we plot results at the 1 MB data point in Fig. 11. We observe that SiPAC consistently performs well, with 3.6× to 5.3× JCT improvement over similarly sized SuperPod topology, 1.4× to 5.9× over 2D Torus, and 1.4× to 3.4× over electronic BCube. This is due to SiPAC's low network diameter and its ability to enable simultaneous direct transmissions to and from $(l + 1)(r - 1)$ different endpoints without intermediate switch buffering. We note that the 2D Torus performs well for the one-to-all and all-to-one collectives since it also enables multiple direct connections with other CUs. However, for the all-to-all traffic pattern, messages need to be queued at intermediate CUs before getting forwarded to their destination, causing significant delay at the endpoints.

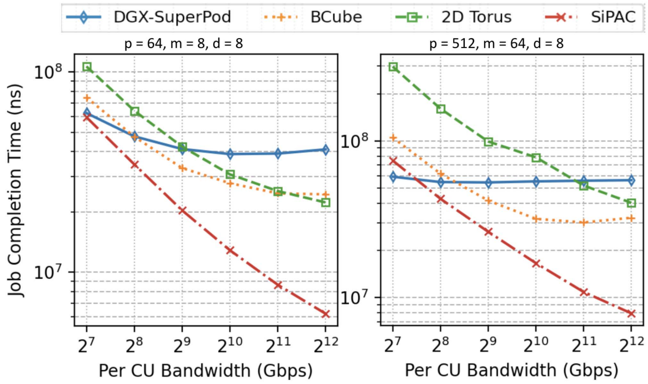

Fig. 13. JCT for hybrid parallel communication at two network sizes for a message size of 100 MB. Left: $p = 64, m = 8, d = 8$; right: $p = 512$, $m = 64, d = 8$.

2) Hybrid Collective Communication: For hybrid parallel traffic workloads, we vary the per-CU bandwidth for each architecture from 128 Gb/s to 4096 Gb/s and set the message size to be 100 MB. Fig. 13 shows the performance of each architecture at two topology scales ($p = 64$ and $p = 512$). These sizes correspond to $r = 8$ and $L = 2, 3$ for both SiPAC and BCube. We set $d = 8, 64$ and $m = 8, 8$ for $p = 64, 512$, respectively. At $p = 64$, BCube, SuperPod, and SiPAC start out with similar JCT at 128 Gb/s due to the uniformly low bandwidth across the network. While the JCT of SiPAC continues to decrease as the per-CU bandwidth increases, the JCT for the other topologies does not improve much further. Taking SuperPod as an example, the communication becomes severely bottlenecked at the slower inter-server links. The JCT for the 2D Torus and electronic BCube scales better than SuperPod but does not gain as much benefit with increasing bandwidth as SiPAC does. 2D Torus is limited by its large diameter in each dimension whereas BCube is limited by the queuing delay incurred in the intermediary switches.

For $p = 512$, SiPAC initially performs worse than SuperPod. This is due to the lower per-CU bandwidth as compared to SuperPod when $L = 3$ which incurs a higher bandwidth cost. When the bandwidth cost is lowered with increasing bandwidth, SiPAC soon outperforms the other topologies. At this scale, the SuperPod topology is always bottlenecked at the inter-server links and therefore exhibits a flat line as intra-server link bandwidth increases. The proposed architecture enables efficient communication that fits well with the DL application demand of the multi-dimensional nature of HP traffic pattern. The SiPAC architecture with optimized collective communication is able to achieve much better scaling which shows its promise for future generations of peta-scale high-bandwidth silicon photonic technologies.

3) Deep Learning Workloads: Next we examine the performance of various architectures using realistic DL workloads. Fig. 12 demonstrates the performance of different topology-collective combinations at two topology sizes with normalized per-CU bandwidth of 2048 Gb/s. The ring-based workloads do not finish within the 2 s window and are therefore left out from this analysis. Across all workloads and network sizes, the hierarchical-ring all-reduce performs the worst on each topology since it incurs the highest latency cost. This is because this algorithm only allows send and receive from one other CU at each time step, which leaves many links under-utilized for these HPC/DL specialized topologies with multiple connections per CU. This is not the case for mesh all-reduce and SiPCO all-reduce since these two collectives can take advantage of the multi-port property of the CUs in these topologies. We further note that the SiPCO all-reduce performs better than the mesh-based all-reduce on BCube and SiPAC topology. By replacing EPSes in the BCube topology with WSSes, we set up direct light paths among CUs in the SiPAC topology. These direct light paths do not have in-network packet buffering, thus reducing the queuing latency. In addition, the multi-wavelength parallel transmission property of SiPAC allows packets to be sent in a non-blocking fashion from each CU. Both of these factors contribute to its improved performance over BCube. While the JCT for hierarchical ring all-reduce increases as the topology size increases, the JCT for the SiPCO all-reduce remains relatively constant. This is due to the linear dependency of the latency term on the topology size for the ring algorithm which dominates over the bandwidth term at large topology sizes. The hierarchical and mesh-allreduce both trade higher bandwidth cost with lower latency cost, which allows them to do well at lower message sizes. However, with larger message size, their higher bandwidth cost could still render them sub-optimal.

VII. Conclusion

In this work, we propose the SiPAC architecture to accelerate DDL. SiPAC achieves efficient multi-dimensional all-to-all network topology using a novel multi-wavelength selective switch, accompanied by a collective algorithm that reduces the required latency cost in collective communications. Using realistic packet-level simulations, we assess the effect of topology size, message size and per-CU bandwidth on SiPAC's performance. We report system-level simulations whose results indicate that SiPAC clusters achieve an 1.4× to 5.9× communication time reduction compared to current state-of-the-art compute clusters for representative collective communications. Our experimental testbed results show the photonic MRR switch's capability to achieve compact and high bandwidth multi-wavelength switching, demonstrating the feasibility of the SiPAC architecture. For future work, we aim to extend our analysis on job placement in SiPAC as real training jobs in multi-tenant training clusters may place jobs on non-adjacent CUs.

References

[1] T. Brown et al., "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020 vol. 33, pp. 1877–1901.

[2] S. Rajbhandari, O. Ruwase, J. Rasley, S. Smith, and Y. He, "Zero-infinity: Breaking the GPU memory wall for extreme scale deep learning," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal.*, 2021, pp. 1–14.

[3] "Nvidia a100 tensor core GPU," 2021. [Online]. Available: https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/a100/pdf/nvidia-a100-datasheet-us-nvidia-1758950-r4-web.pdf

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[5] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[7] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-xl: Attentive language models beyond a fixed-length context," 2019, *arXiv:1901.02860*.

[8] A. Radford et al., "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, 2019, Art. no. 9.

[9] M. Shoeybi, M. Patwary, R. Puri, P. LeGresley, J. Casper, and B. Catanzaro, "Megatron-lm: Training multi-billion parameter language models using model parallelism," 2019, *arXiv:1909.08053*.

[10] S. Smith et al., "Using deepspeed and megatron to train megatron-turing NLG 530b, a large-scale generative language model," 2022, *arXiv:2201.11990*.

[11] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," *J. Mach. Learn. Res.*, 2021, *arXiv:2101.03961*.

[12] D. Mudigere et al., "Software-hardware co-design for fast and scalable training of deep learning recommendation models," in *Proc. 49th Annu. Int. Symp. Comput. Architecture*, 2022, pp. 993–1011.

[13] D. Narayanan et al., "Efficient large-scale language model training on gpu clusters using megatron-lm," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal.*, 2021, pp. 1–15.

[14] "The world's highest-bandwidth on-node switch," 2018. [Online]. Available: https://images.nvidia.com/content/pdf/nvswitch-technical-overview.pdf

[15] "Nvidia v100 tensor core gpu," 2020. [Online]. Available: https://images.nvidia.com/content/technologies/volta/pdf/volta-v100-datasheet-update-us-1165301-r5.pdf

[16] "Nvidia tesla p100 GPU accelerator," 2016. [Online]. Available: https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/tesla-p100/pdf/nvidia-tesla-p100-datasheet.pdf

[17] "Nvidia tesla m40 GPU accelerator," 2016. [Online]. Available: https://images.nvidia.com/content/tesla/pdf/78071_Tesla_M40_24GB_Print_Datasheet_LR.PDF

[18] "Nvidia connectx-7 400 g," [Online]. Available: https://www.nvidia.com/content/dam/en-zz/Solutions/networking/ethernet-adapters/connectx-7-datasheet-Final.pdf

[19] "Nvidia connectx-6," 2022. [Online]. Available: https://nvdam.widen.net/s/5j7xtzqfxd/connectx-6-infiniband-datasheet-1987500-r2

[20] "Nvidia connectx-5," 2021. [Online]. Available: https://nvdam.widen.net/s/pkxbnmbgkh/networking-infiniband-datasheet-connectx-5-2069273

[21] "Nvidia connectx-4," 2020. [Online]. Available: https://network.nvidia.com/files/doc-2020/pb-connectx-4-lx-en-card.pdf

[22] Q. Cheng, M. Bahadori, M. Glick, S. Rumley, and K. Bergman, "Recent advances in optical technologies for data centers: A review," *Optica*, vol. 5, no. 11, pp. 1354–1370, 2018.

[23] K. Bergman, "Peta-scale embedded photonics for high performance computing," in *Proc. Eur. Conf. Opt. Commun.*, 2022.

[24] P. De Dobbelaere, "Silicon photonics transceivers for hyper-scale datacenters: Deployment and roadmap," in *Proc. Eur. Conf. Opt. Commun.*, 2016.

[25] "Optoelectronics amp; photonics portfolio," [Online]. Available: https://www.macom.com/files/live/sites/macom/files/Brochures/MACOM_Opto%20brochure.pdf

[26] "Intel silicon photonics 200 g fr4 qsfp56 optical transceiver brief," 2021. [Online]. Available: https://www.intel.com/content/www/us/en/architecture-and-technology/silicon-photonics/optical-transceiver-200g-fr4-qsfp56-brief.html

[27] "Teraphy," Dec. 2021. [Online]. Available: https://ayarlabs.com/teraphy/

[28] [Online]. Available: https://www.darpa.mil/news-events/2020-03-25

[29] A. Rizzo et al., "Integrated kerr frequency comb-driven silicon photonic transmitter," 2021, *arXiv:2109.10297*.

[30] B. Corcoran et al., "Ultra-dense optical data transmission over standard fibre with a single chip source," *Nature Commun.*, vol. 11, no. 1, pp. 1–7, 2020.

[31] B. Y. Kim et al., "Turn-key, high-efficiency kerr comb source," *Opt. Lett.*, vol. 44, no. 18, pp. 4475–4478, 2019.

[32] Z. Wu, L. Y. Dai, Z. Zhu, A. Novick, M. Glick, and K. Bergman, "Sip architecture for accelerating collective communication in distributed deep learning," in *Proc. Opt. Fiber Commun. Conf.*, 2023, Paper W1G.1.

[33] G. Andrew, "Bringing HPC techniques to deep learning."

[34] X. Jia et al., "Highly scalable deep learning training system with mixed-precision: Training imagenet in four minutes," 2018, *arXiv:1807.11205*.

[35] S. Wang, J. Geng, and D. Li, "Impact of synchronization topology on DML performance: Both logical topology and physical topology," *IEEE/ACM Trans. Netw.*, vol. 30, no. 2, pp. 572–585, Apr. 2022.

[36] S. Wang et al., "Bml: A high-performance, low-cost gradient synchronization algorithm for DML training," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, vol. 31.

[37] "Nvidia dgx superpod," 2021. [Online]. Available: https://images.nvidia.com/aem-dam/Solutions/Data-Center/gated-resources/nvidia-dgx-superpod-a100.pdf

[38] "Google cloud tpu," [Online]. Available: https://cloud.google.com/tpu

[39] Lawrence livermore national laboratory, sierra. [Online]. Available: https://hpc.llnl.gov/hardware/compute-platforms/sierra

[40] N. Dryden et al., "Aluminum: An asynchronous, GPU-aware communication library optimized for large-scale training of deep neural networks on hpc systems," Lawrence Livermore Nat. Lab., Livermore, CA, USA, Tech. Rep. LLNL-CONF-757866, 2018.

[41] S. Rashidi et al., "Enabling compute-communication overlap in distributed deep learning training platforms," in *Proc. IEEE/ACM 48th Annu. Int. Symp. Comput. Architecture*, 2021, pp. 540–553.

[42] A. F. Aji and K. Heafield, "Sparse communication for distributed gradient descent," 2017, *arXiv:1704.05021*.

[43] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," 2017, *arXiv:1712.01887*.

[44] W. Won, S. Rashidi, S. Srinivasan, and T. Krishna, "Exploring multi-dimensional hierarchical network topologies for efficient distributed training of trillion parameter dl models," 2021, *arXiv:2109.11762*.

[45] X. Hou, R. Xu, S. Ma, Q. Wang, W. Jiang, and H. Lu, "Co-Designing the topology/algorithm to Accelerate Distributed Training," in *Proc. IEEE Int. Conf. Parallel Distrib. Process. Appl., Big Data Cloud Comput., Sustain. Comput. Commun., Social Comput. Netw.*, 2021, pp. 1010–1018.

[46] J. Huang, P. Majumder, S. Kim, A. Muzahid, K. H. Yum, and E. J. Kim, "Communication algorithm-architecture co-design for distributed deep learning," in *Proc. ACM/IEEE 48th Annu. Int. Symp. Comput. Architecture*, 2021, pp. 181–194.

[47] D. De Sensi, S. Di Girolamo, S. Ashkboos, S. Li, and T. Hoefler, "Flare: Flexible in-network allreduce," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal.*, 2021, pp. 1–16.

[48] S. Daudlin et al., "3D-integrated multichip module transceiver for terabit-scale DWDM interconnects," in *Proc. Opt. Fiber Commun. Conf.*, 2021, Paper Th4A–4.

[49] K. Hosseini et al., "8 Tbps co-packaged FPGA and silicon photonics optical IO," in *Proc. IEEE Opt. Fiber Commun. Conf. Exhib.*, 2021, pp. 1–3.

[50] Y. Lu, H. Gu, X. Yu, and P. Li, "X-NEST: A scalable, flexible, and high-performance network architecture for distributed machine learning," *J. Lightw. Technol.*, vol. 39, no. 13, pp. 4247–4254, Jul. 2021.

[51] L. Liu, Q. Jin, D. Wang, H. Yu, G. Sun, and S. Luo, "PSNet: Reconfigurable network topology design for accelerating parameter server architecture based distributed machine learning," *Future Gener. Comput. Syst.*, vol. 106, pp. 320–332, 2020.

[52] T. T. Nguyen and R. Takano, "On the feasibility of hybrid electrical/optical switch architecture for large-scale training of distributed deep learning," in *Proc. IEEE/ACM Workshop Photon.- Opt. Technol. Oriented Netw., Inf. Comput. Syst.*, 2019, pp. 7–14.

[53] M. Khani et al., "SiP-ML: High-bandwidth optical network interconnects for machine learning training," in *Proc. ACM SIGCOMM 2021 Conf.*, 2021, pp. 657–675.

[54] W. Wang et al., "Topoopt: Co-optimizing network topology and parallelization strategy for distributed training jobs," in *Proc. 20th USENIX Symp. Networked Syst. Des. Implementation*, 2022, vol. 2023, pp. 739–767.

[55] C. Guo et al., "BCube: A high performance, server-centric network architecture for modular data centers," in *Proc. ACM SIGCOMM Conf. Data Commun.*, 2009, pp. 63–74.

[56] L. N. Bhuyan and D. P. Agrawal, "Generalized hypercube and hyperbus structures for a computer network," *IEEE Trans. Comput.*, vol. C-33, no. 4, pp. 323–333, Apr. 1984.

[57] T.-N. Truong and R. Takano, "Hybrid electrical/optical switch architectures for training distributed deep learning in large-scale," *IEICE Trans. Inf. Syst.*, vol. 104, no. 8, pp. 1332–1339, 2021.

[58] N. C. Abrams et al., "Silicon photonic 2.5 D multi-chip module transceiver for high-performance data centers," *J. Lightw. Technol.*, vol. 38, no. 13, pp. 3346–3357, Jul. 2020.

[59] M. Fariborz, X. Xiao, P. Fotouhi, R. Proietti, and S. B. Yoo, "Silicon photonic Flex-LIONSs for reconfigurable multi-GPU systems," *J. Lightw. Technol.*, vol. 39, no. 4, pp. 1212–1220, Feb. 2021.

[60] L. Y. Dai, Y.-H. Hung, Q. Cheng, and K. Bergman, "Experimental demonstration of pam-4 transmission through microring silicon photonic clos switch fabric," in *Proc. Opt. Fiber Commun. Conf.*, 2020, Paper M1H–3.

[61] Q. Cheng, M. Bahadori, Y.-H. Hung, Y. Huang, N. Abrams, and K. Bergman, "Scalable microring-based silicon clos switch fabric with switch-and-select stages," *IEEE J. Sel. Topics Quantum Electron.*, vol. 25, no. 5, pp. 1–11, Sep./Oct. 2019.

[62] 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. Lightw. Technol.*, vol. 37, no. 1, pp. 110–115, Jan. 2019.

[63] M. Hattink, L. Y. Dai, Z. Zhu, and K. Bergman, "Streamlined architecture for thermal control and stabilization of cascaded DWDM micro-ring filters bus," in *Proc. IEEE Opt. Fiber Commun. Conf.*, 2022, pp. 1–3.

[64] A. Novick, K. Jang, A. Rizzo, R. Parsons, and K. Bergman, "Low-loss wide-FSR miniaturized racetrack style microring filters for 1 Tbps DWDM," in *Proc. Opt. Fiber Commun. Conf.*, 2023, Paper Th3A.3.

[65] M. Bahadori et al., "Design space exploration of microring resonators in silicon photonic interconnects: Impact of the ring curvature," *J. Lightw. Technol.*, vol. 36, no. 13, pp. 2767–2782, Jul. 2018.

[66] Y. Wang et al., "Dispersion-engineered and fabrication-robust SOI waveguides for ultra-broadband DWDM," in *Proc. Opt. Fiber Commun. Conf. Exhib.*, 2023, pp. 1–3.

[67] A. Rizzo et al., "Petabit-scale silicon photonic interconnects with integrated kerr frequency combs," *IEEE J. Sel. Topics Quantum Electron.*, vol. 29, no. 1: Nonlinear Integrated Photonics, pp. 1–20, Jan./Feb. 2022.

[68] A. Novick et al., "Error-free Kerr comb-driven sip microdisk transmitter," in *Proc. IEEE Conf. Lasers Electro- Opt.*, 2021, pp. 1–2.

[69] R. Thakur, R. Rabenseifner, and W. Gropp, "Optimization of collective communication operations in mpich," *Int. J. High Perform. Comput. Appl.*, vol. 19, no. 1, pp. 49–66, 2005.

[70] S. Rashidi, S. Sridharan, S. Srinivasan, and T. Krishna, "ASTRA-SIM: Enabling SW/HW co-design exploration for distributed DL training platforms," in *Proc. IEEE Int. Symp. Perform. Anal. Syst. Softw.*, 2020, pp. 81–92.

[71] A. Sergeev and M. Del Balso, "Horovod: Fast and easy distributed deep learning in tensorflow," 2018, *arXiv:1802.05799*.

[72] P. Xie et al., "Distributed machine learning via sufficient factor broadcasting," 2015, *arXiv:1511.08486*.

[73] "Nccl collective operations," 2020. [Online]. Available: https://docs.nvidia.com/deeplearning/nccl/user-guide/docs/usage/collectives.html#allreduce

[74] "Open MPI: Open source high performance computing," 2023. [Online]. Available: https://www.open-mpi.org/

[75] "Netbench," 2018. [Online]. Available: https://github.com/ndal-eth/netbench

[76] "Dgx-a100 system user guide." 2023. [Online]. Available: https://docs.nvidia.com/dgx/pdf/dgxa100-user-guide.pdf

[77] A. Li et al., "Evaluating modern GPU interconnect: PCIe, NVLink, NV-SLI, NVSwitch and GPUDirect," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 1, pp. 94–110, Jan. 2019.

[78] "Introducing 200 G HDR infiniband solutions," 2019. [Online]. Available: https://network.nvidia.com/sites/default/files/doc-2020/wp-introducing-200g-hdr-infiniband-solutions.pdf

[79] A. S. P. Khope, R. Helkey, S. Liu, A. A.M. Saleh, R. C. Alferness, and J. E. Bowers, "A scalable multicast hybrid broadband crossbar wavelength selective switch for datacenters," in *Proc. IEEE 11th Annu. Comput. Commun. Workshop Conf.*, 2021, pp. 1585–1587.

[80] J. Kim, W. J. Dally, S. Scott, and D. Abts, "Technology-driven, highly-scalable dragonfly topology," in *Proc. Int. Symp. Comput. Architecture*, 2008, pp. 77–88.

[81] M. Wade et al., "TeraPHY: A chiplet technology for low-power, high-bandwidth in-package optical I/O," *IEEE Micro*, vol. 40, no. 2, pp. 63–71, Mar./Apr. 2020.

[82] T. Drenski and J. C. Rasmussen, "ADC & DAC-technology trends and steps to overcome current limitations," in *Proc. Opt. Fiber Commun. Conf.*, 2018, Paper M2C–1.

[83] S. Almonacil, F. Boitier, and P. Layec, "Performance model and design rules for optical systems employing low-resolution DAC/ADC," *J. Lightw. Technol.*, vol. 38, no. 11, pp. 3007–3014, Jun. 2020.

[84] W. J. Turner et al., "Ground-referenced signaling for intra-chip and short-reach chip-to-chip interconnects," in *Proc. IEEE Custom Integr. Circuits Conf.*, 2018, pp. 1–8.

[85] J. A. Yang, J. Park, S. Sridharan, and P. T. P. Tang, "Training deep learning recommendation model with quantized collective communications," in *Proc. Conf. Knowl. Discov. Data Mining*, 2020.

[86] W. J. Turner et al., "High-bypass learning: Automated detection of tumor cells that significantly impact drug response," in *Proc. IEEE/ACM Workshop Mach. Learn. High Perform. Comput. Environments Workshop Artif. Intell. Mach. Learn. Sci. Appl.*, 2020, pp. 1–10.

[87] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[88] K. Tanaka et al., "Large-message size allreduce at wire speed for distributed deep learning," in *Proc. Poster Session Presented at SC18*, 2018.

[89] M. Naumov et al., "Deep learning training in facebook data centers: Design of scale-up and scale-out systems," *arXiv:2003.09518*, 2020.

[90] J. Fei, C.-Y. Ho, A. N. Sahu, M. Canini, and A. Sapio, "Efficient sparse collective communication and its application to accelerate distributed deep learning," in *Proc. 2021 ACM SIGCOMM 2021 Conf.*, 2021, pp. 676–691.

Zhenguo Wu (Graduate Student Member, IEEE) received the B.S. and M.S. degrees in electrical engineering in 2020 and 2021, respectively, from Columbia University, New York, NY, USA, where he is currently working toward the Ph.D. degree with a focus in photonic network design. His research interests include reconfigurable network topologies and network simulations for data center and high performance computing systems.

Liang Yuan Dai (Graduate Student Member, IEEE) received the Bachelor of Engineering degree from the New Jersey's Institute of Technology, Newark, NJ, USA. He is currently working toward the Ph.D. degree working with the Lightwave Research Laboratory, Columbia University, New York, NY, USA, directed by Dr. Keren Bergman. His research interests include photonic switch architectures, in addition to silicon photonics device testing and packaging.

Asher Novick (Graduate Student Member, IEEE) received the B.S. and M.Eng. degrees in electrical and computer engineering from Cornell University, Ithaca, NY, USA, in 2015 and 2016, respectively. He is currently working toward the Ph.D. degree in electrical engineering with the Lightwave Research Laboratory, Columbia University, New York, NY, USA. Between 2016 and 2019, he was with Panduit's Fiber Research Lab, where he researched and developed new patentable technologies for optical fiber-based communication in data center and enterprise applications. His research interests include modeling, design, and testing of silicon photonic systems and devices for scalable, and efficient link architectures.

Madeleine Glick (Senior Member, IEEE) received the Ph.D. degree in physics from the Columbia University, New York, NY, USA, for research on electro-optic effects of GaAs/AlGaAs quantum wells. She is currently Senior Research Scientist with Columbia University. From 1992 to 1996, she was a Research Associate with CERN, Geneva, Switzerland, as part of the Lightwave Links for Analogue Signal Transfer Project for the Large Hadron Collider. From 2002 to 2011, she was Principal Engineer at Intel (Intel Research Cambridge U.K., Intel Research Pittsburgh) leading research on optical interconnects for computer systems. Her research interests include applying photonic devices and interconnects to computing systems. Dr. Glick is a Fellow of OPTICA.

Ziyi Zhu received the B.Eng. degree in electronic science and technology from Sichuan University, Chengdu, China, in 2015 and the M.S. and Ph.D. degrees in electrical engineering from Columbia University, New York, NY, USA, in 2017 and 2022, respectively. He is currently with Programmable Solutions Group - Chief Technology Office at Intel Corporation. He works on multi-chip package platforms, co-packaged optics, and IP integration to enable technological advancements in compute for FPGAs.

Sébastien Rumley received the M.S. and Ph.D. degrees in communication systems from the Swiss Federal Institute of Technology Lausanne (EPFL), Lausanne, Switzerland, in 2005 and 2011. He is currently Associate Professor of software engineering with the University of Applied Sciences and Arts Western Switzerland (HES-SO), Delémont, Switzerland. His research interests include complex systems modeling, simulation platforms development, optical interconnects and green IT.

George Michelogiannakis (Senior Member, IEEE) is currently a Research Scientist for the computer architecture Group (CAG) with the Computational Research Division (CRD), Lawrence Berkeley National Lab, Berkeley, CA, USA. He has extensive work on networking (both off- and on-chip) and computer architecture. His research interests include the post-Moore's law era looking into specialization, emerging devices (transistors), memories, photonics, and 3D integration. He is also currently working on optics and architecture for HPC and data center networks.

John Shalf (Senior Member, IEEE) is currently the Department Head of computer science with Lawrence Berkeley National Laboratory, Berkeley, CA, USA, and formerly the deputy Director of hardware technology for the DOE Exascale Computing Project and the Leader of the Green Flash Project. Contact him at jshalf@lbl.gov.

Keren Bergman (Fellow, IEEE) received the B.S. degree in electrical engineering from Bucknell University, Lewisburg, PA, USA, in 1988, and the M.S. and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 1991 and 1994, respectively. She is currently the Charles Batchelor Professor of Electrical Engineering with Columbia University, Berkeley, CA, USA, where she is also the Faculty Director of the Columbia Nano Initiative. At Columbia, he leads the Lightwave Research Laboratory encompassing multiple cross-disciplinary programs at the intersection of computing and photonics. He serves on the Leadership Council of the American Institute of Manufacturing (AIM) Photonics leading Projects that support the institute's silicon photonics manufacturing capabilities and Datacom applications. She was the recipient of the 2016 IEEE Photonics Engineering Award. She is a Fellow of Optica.