Which can Accelerate Distributed Machine Learning Faster: Hybrid Optical/Electrical or Optical Reconfigurable DCN?

Which can Accelerate Distributed Machine Learning Better: HOE-DCN with OXC or Hyper-FleX-LION?

Hao Yang[1], Zuqing Zhu[1], Roberto Proietti[2], and S. J. Ben Yoo[2]

1. University of Science and Technology of China, Hefei, Anhui 230027, China, Email: zqzhu@ieee.org
2. University of California, Davis, Davis, CA 95616, USA, Email: sbyoo@ucdavis.edu

Abstract: We run various distributed machine learning (DML) architectures in a HOE-DCN with OXC and an optical DCN based on Flex-LIONS (Hyper-FleX-LION). Experimental results show that Hyper-FleX-LION gains better DML acceleration and improves acceleration ratio by up to 22.3%.

OCIS codes: (060.1155) Software-defined optical networks; (060.4251) Networks, assignment and routing algorithms.

1. Introduction

Recently, the wide applications of Big Data analytics have made machine learning (ML) indispensable in many fields [1, 2]. The rapid development of ML increased the complexity of ML models and caused the scale of ML training to be huge. For instance, an ML model used by Google for language processing may utilize over 10^9 parameters [2]. Meanwhile, a large Internet company can train an ML model with over 1 PB training data for click-through-rate estimation [3]. Such a massive scale of training can hardly run on a single machine. Hence, distributed ML (DML), which segments and distributes training data over multiple machines for parallel training, has attracted intensive interests recently [4]. DML is usually deployed in a data center network (DCN) to use its abundant IT resources, bringing new challenges to DCNs. Firstly, the jobs of a large-scale DML typically have to span across multiple racks. Therefore, during training, the parameter synchronization among DML nodes will generate heavy inter-rack traffic [3]. Secondly, programmers can arrange a cluster of DML nodes into various architectures, each of which leads to a traffic pattern with unique temporal and spatial characteristics [4]. The issues above can degrade the performance of the DML running in traditional DCNs based on electronic packet switching (EPS), and prolong the job completion time (JCT) [5]. For example, it is known that the EPS-based DCN built with fat-tree cannot efficiently support the DML based on the *Parameter Server* architecture due to the inter-rack bottlenecks caused by DML-induced congestions [3].

As the parameter synchronization of DML needs to exchange large amounts of data among DML nodes, the related inter-rack communications will be elephant flows [2, 3]. Hence, the inter-rack bottlenecks can be relieved by introducing optical circuit switching (OCS) and building hybrid optical/electrical DCN (HOE-DCN) [6, 7]. Wang *et al.* [5] verified that by reconfiguring the OCS part of a HOE-DCN adaptively, DML jobs could be accelerated to achieve reduced JCT. However, their HOE-DCN was built with an optical cross-connect (OXC), which can only provide one-to-one connectivity between inputs and outputs. Hence, the OCS part might have difficulty to properly adapt to the traffic matrix of an arbitrary DML architecture. On the other hand, researchers showed that the Hyper-FleX-LION architecture [8] supports reconfigurable all-to-all optical interconnects using a Flex-LIONS switch device [9].

In this work, we perform a comparative study to investigate which architecture can accelerate DML better, HOE-DCN with OXC (HOE-w/OXC) or all-optical DCN based on Flex-LIONS (Hyper-FleX-LION)? We first analyze the traffic patterns of four typical types of DML architectures [4] (*i.e.*, *Distributed Data Parallel (DDP)*, *Ring-AllReduce (Ring)*, *Parameter Server (PS)* and *Peer-to-Peer (P2P)*) and check whether the inter-rack topologies of the two types of DCNs can adapt to them well. Then, to quantitatively evaluate the performance of HOE-w/OXC and Hyper-FleX-LION on DML, we set up a small-scale DCN testbed that consists of 4 racks. Finally, we connected the racks with HOE-w/OXC or Hyper-FleX-LION and conducted experiments in various DML scenarios. Our results show that for all the experimental scenarios, Hyper-FleX-LION performs better than or at least as well as HOE-w/OXC on DML acceleration. Specifically, Hyper-FleX-LION can improve the acceleration ratio up to 22.3% (over HOE-w/OXC).

2. Matching Degree between DML Traffic Patterns and DCN Architectures

Without loss of generality, we analyze the traffic patterns of *DDP*, *Ring*, *PS* and *P2P* by assuming that the nodes of each DML architecture are deployed in four racks of a DCN. The architecture and operation principle of Hyper-FleX-LION is shown in Fig. 1(a). Here, we use off-the-shelf components to build the Hyper-FleX-LION, but it can also be realized in a much more compact and energy-efficient way with integrated optical chips [9]. We have an arrayed waveguide grating router (AWGR) sitting in the middle, and the transmitting and receiving structures of each rack are located at its left and right sides, respectively [9]. For each rack, its top-of-rack (TOR) switch uses four transceivers (TRXs) as shown in Fig. 1(a), where we use different numbers to indicate the wavelengths used by the TRXs (the color of each number represents the source rack). In the transmitting structure, all the outputs of a ToR switch are multiplexed and then enter a wavelength selective switch (WSS). One of the WSS' outputs is connected to the AWGR, while the other

three outputs go directly to the WSS in the receiving structures of other racks. In the receiving structure of each rack, a WSS selects the received signals to distribute them to the TRXs of the ToR switch by a de-multiplexer (DEMUX). Then, by utilizing the wavelength switching capability of the AWGR and adjusting the WSS' switching states, we can obtain various topologies to interconnect the racks (*e.g.*, the configuration in Fig. 1(a) leads to a full-mesh).

Fig. 1(b) shows the traffic patterns of the DML architectures. The colorful arrows denote the traffic generated by DML, and the purple and blue arrows represent the acceleration bandwidth for DML that can be provided by HOE-w/OXC and Hyper-FleX-LION, respectively. Here, we define *the acceleration bandwidth for DML (Accel-BW)* as the bandwidth that a DCN can provide to DML in addition to that for setting up the basic inter-rack communications. For instance, both *DDP* and *Ring* have ring-like traffic matrices, whose basic inter-rack communications can be supported with the fat-tree-based EPS part of HOE-w/OXC or two TRXs on each ToR switch in Hyper-FleX-LION. But, then, as the OXC in HOE-w/OXC only provides one-to-one connectivity for Accel-BW, it can just connect the racks in pairs (*e.g.*, 1 to 2 and 3 to 4 in Fig. 1(b)), but cannot realize a ring-like inter-rack topology. On the other hand, as shown in Fig. 1(c), we can reconfigure Hyper-FleX-LION to set up another ring-like inter-rack topology for Accel-BW. This verifies that, in principle, Hyper-FleX-LION can accelerate *DDP* and *Ring* better than HOE-w/OXC.

Next, we consider *PS*, which uses a tree-like topology including a server node and n worker nodes, and denote a DML with *PS* as *PS(n)*. We first address *PS(2)*, and place the server node on *Rack* 1 and two worker nodes on *Racks* 2 and 4. Fig. 1(b) suggests that the Accel-BW from HOE-w/OXC only works for one branch of the tree-like inter-rack topology (*e.g.*, 1 to 2). Meanwhile, with sufficient TRXs and adaptivity, Hyper-FleX-LION can establish another tree-like topology for Accel-BW. However, the situation becomes different for *PS(3)*, when an additional worker is placed on *Rack* 3. As shown in Fig. 1(b), the Accel-BW from HOE-w/OXC still only works for one branch of the tree-like topology, but as Hyper-FleX-LION already uses 3 TRXs on *Rack* 1 for the basic inter-rack communications of *PS(3)*, its Accel-BW can only work for a branch of the tree too. Hence, for *PS(3)*, the Accel-BWs from HOE-w/OXC and Hyper-FleX-LION are similar. Finally, it is *P2P*, in which each DML node only talks with one peer at a time. Here, both HOE-w/OXC and Hyper-FleX-LION can provide Accel-BW, but Hyper-FleX-LION can allocate two more TRXs on each rack to communicate with its peer rack. Hence, Hyper-FleX-LION also provides larger Accel-BW for *P2P*.

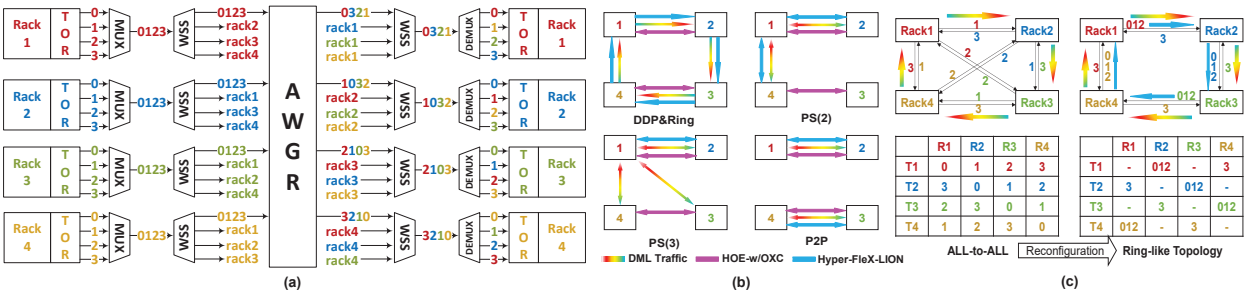


Fig. 1. (a) Hyper-FleX-LION, (b) Traffic patterns of DML architectures, and (c) Reconfiguration of Hyper-FleX-LION.

3. Experimental Evaluations

To quantify the performance difference of HOE-w/OXC and Hyper-FleX-LION on the four DML architectures, we built a small-scale DCN testbed including four racks, each of which consists of two servers. Each server contains four 6-core CPUs and 32 GB of memory. To ensure fair comparisons, we equip four 1GbE optical ports on each ToR switch. For HOE-w/OXC, we connect three ports on each ToR switch to the EPS-based inter-rack topology, which is based on fat-tree, and the fourth port goes to an OXC. For Hyper-FleX-LION, the ports on ToR switches are connected with the architecture in Fig. 1(a), consisting of one 8×8 AWGR, eight 1×9 WSS', and several other passive components. Note that the OXC, AWGR, and WSS' in the experimental setup are all commercially available products.

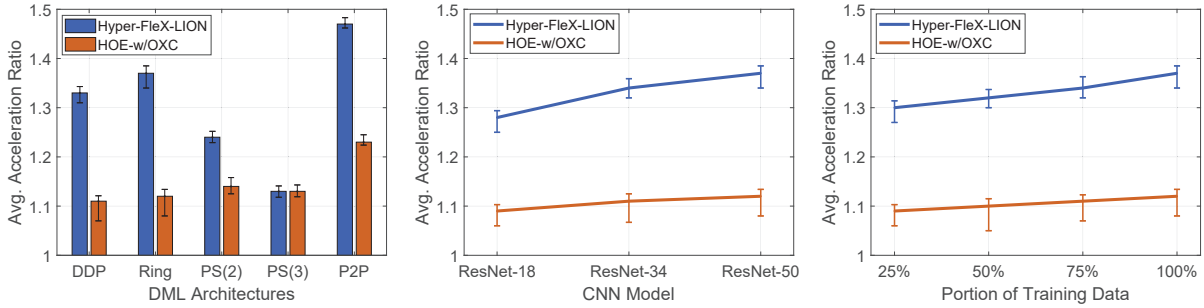
As for the DML, we use the famous CIFAR-10 data set (containing 60,000 32×32 color images in 10 different classes) and train a convolutional neural network (CNN) for image classification. In a DML job, each node runs on one server. To stress out the DCNs, we run multiple DML jobs simultaneously in each experiment and average the job completion time (JCT) from 10 independent runs to get the average JCT of the experiment. In each experiment, we first run the DML job in the EPS part of HOE-w/OXC and record the average JCT as the baseline, and then we get the actual average JCT by running the job in HOE-w/OXC or Hyper-FleX-LION. Next, the *acceleration ratio* of HOE-w/OXC or Hyper-FleX-LION can be obtained by dividing the baseline with the actual average JCT [5].

We architect the CNN with the well-known ResNet model, make its depth as 50 layers (*i.e.*, ResNet-50), and use the DML architectures in Fig. 1(b) to train it with 100% of the data in CIFAR-10. The results on acceleration ratio are

shown in Fig. 2(a). As expected, Hyper-FleX-LION provides larger acceleration ratios than HOE-w/OXC when the DML uses *DDP*, *Ring*, *PS(2)* and *P2P*, while the two DCNs perform similarly for *PS(3)*. This verifies our analysis above. Finally, in Fig. 2(a), it is interesting to observe that even though the traffic patterns of *DDP* and *Ring* are the same in Fig. 1(b), the acceleration ratios of *DDP* are smaller than those of *Ring* in both HOE-w/OXC and Hyper-FleX-LION. This is because *DDP* incorporates specific processing to reduce the data transfers among the DML nodes, *i.e.*, *DDP* incurs less inter-rack communications than *Ring* [4]. In all, the acceleration ratios in Fig. 2(a) suggest that for the five DML architectures, the improvements on acceleration ratio achieved by Hyper-FleX-LION over HOE-w/OXC have a maximum of 22.3% (*Ring*) and an average value of 14.1%.

Next, we focus on *Ring* to further compare the performance of HOE-w/OXC and Hyper-FleX-LION on DML. First, we consider three CNN models, *i.e.*, ResNet-18, ResNet-34, and ResNet-50, and train them with 100% of the data in CIFAR-10, using the DML with *Ring*. The experimental results in Fig. 2(b) indicate that Hyper-FleX-LION outperforms HOE-w/OXC for all the CNN models. Meanwhile, we notice that as the complexity of the CNN model increases, the improvement in acceleration ratio achieved by Hyper-FleX-LION becomes larger. This is because when the CNN model is more complex with a larger number of layers, the parameters to be optimized and synchronized in a DML job increase, which leads to more inter-rack data transfers. The results in Fig. 2(b) indicate that the improvements in acceleration ratio achieved by Hyper-FleX-LION over HOE-w/OXC have an average value of 20.1%.

Finally, as certain DML might not use the full training data set in training, we consider the cases that DML with *Ring* uses $\{25\%, 50\%, 75\%, 100\%\}$ of the data in CIFAR-10 to train the CNN in ResNet-50. The results are plotted in Fig. 2(c). Once again, the advantage of Hyper-FleX-LION on DML acceleration can be seen clearly. Moreover, the results also suggest that when the amount of training data increases, the advantage of Hyper-FleX-LION becomes more significant. This is still because when the amount of inter-rack data transfers increases, the higher match degree between inter-rack topology and DML traffic matrix in Hyper-FleX-LION exhibits a larger effect on reducing JCT. In Fig. 2(c), the average improvement in acceleration ratio achieved by Hyper-FleX-LION is 20.5%. The results in Figs. 2(b) and 2(c) confirm that the advantage of Hyper-FleX-LION on DML acceleration increases with the CNN model's complexity and the amount of training data. This makes Hyper-FleX-LION more promising for large-scale DML.



(a) Different DML architectures (b) Various CNN models (c) Different sizes of training data

Fig. 2. Results on acceleration ratio (those in (b) and (c) are obtained with the DML architecture of *Ring*).

4. Summary

We compared HOE-w/OXC and Hyper-FleX-LION on DML acceleration theoretically and experimentally. Our results verified that for all the DML scenarios, Hyper-FleX-LION performs better than or at least as well as HOE-w/OXC.

References

[1] P. Lu *et al.*, "Highly-efficient data migration and backup for Big Data applications in elastic optical inter-data-center networks," *IEEE Netw.*, vol. 29, pp. 36-42, Sept./Oct. 2015.

[2] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," *in Proc. of OSDI 2016*, pp. 265-283, Nov. 2016.

[3] S. Wang *et al.*, "A scalable, high-performance, and fault-tolerant network architecture for distributed machine learning," *IEEE/ACM Trans. Netw.*, vol. 28, pp. 1752-1764, Aug. 2020.

[4] J. Verbraeken *et al.*, "A survey on distributed machine learning," *arXiv:1912.09789*, Dec. 2019. [Online]. Available: https://arxiv.org/abs/1912.09789.

[5] C. Wang *et al.*, "Acceleration and efficiency warranty for distributed machine learning jobs over data center network with optical circuit switching," *in Proc. of OFC 2021*, paper W1E.3, Jun. 2021.

[6] N. Farrington *et al.*, "Helios: a hybrid electrical/optical switch architecture for modular data centers," *in Proc. of SIGCOMM 2010*, pp. 339-350, Aug. 2010.

[7] H. Fang *et al.*, "Predictive analytics based knowledge-defined orchestration in a hybrid optical/electrical datacenter network testbed," *J. Lightw. Technol.*, vol. 37, pp. 4921-4934, Oct. 2019.

[8] G. Liu *et al.*, "Architecture and performance studies of 3D-Hyper-FleX-LION for reconfigurable all-to-all HPC networks," *in Proc. of SC 2020*, pp. 1-16, Nov. 2020.

[9] X. Xiao *et al.*, "Multi-FSR silicon photonic Flex-LIONS module for bandwidth-reconfigurable all-to-all optical interconnects," *J. Lightw. Technol.*, vol. 38, pp. 3200-3208, Jun. 2020.

This work was supported in part by NSF ECCS award #1611560.