

# X-NEST+: A High Bandwidth and Reconfigurable Optical Interconnects for Distributed Machine Learning and High-Performance Computing

Huaxi Gu\*, Xiaoshan Yu, Yunfeng Lu, Hong Zou, and Shuo Li

*State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an, 710071, China*

*\*hxgu@xidian.edu.cn*

**Abstract:** We propose X-NEST+, a scalable and high-bandwidth optical interconnects capable of reconfiguring both intra- and inter- cluster topology based on traffic demands. The experiment results indicate up to 8%~36% reduction in completion time for HPC and ML applications compared with Helios and RotorNet.

## 1. Introduction

The growing size of datasets and models has made machine learning the most common application in data centers and high-performance computing (HPC) systems. Distributed machine learning (ML) accelerates the training process by improving the parallelism of tasks through the cooperation of multiple computing nodes. However, the computational throughput increases much faster than the communication bandwidth, which results in the communication time occupying most of the total training time [1]. The interconnection network has become an important factor in reducing the entire training time. As different parallelisms of the ML (i.e., model parallelism, data parallelism and hybrid parallelism) exhibit different communication characteristics, it is challenging to design networks that properly balance the cost and network performance. A network with an overprovisioning of bandwidth can support various parallel execution method, but suffers from unnecessary costs. The underprovisioning networks achieve better cost and bandwidth efficiency but can only support a few specific parallel methods. To this end, optical interconnects are introduced to dynamically provide on-demand connections across the network. While significantly improving the total application execution time with limited bandwidth resources, most existing designs, including Helios [2], Flexfly [3], RotorNet [4] and X-NEST [5], deploy optical switches between the clusters or pods. All the intra-cluster flows are still delivered by a pure electrical network with a large number of static links and switches. As the empirical study reveals that the intra-cluster traffic of most HPC and ML applications is still skewed and sparse [3], it is possible to further introduce the reconfigurable optical connections within the cluster. Therefore, X-NEST+, a rack-level reconfigurable optical interconnect, is proposed in this paper. It adopts an all-optical interconnect for inter-cluster communication, while using an optical and electrical hybrid interconnect for the intra-cluster communication. Compared to Flexfly and X-NEST, X-NEST+ makes the network more flexible by changing the connection relationships among any of the racks and improves the link utilization by assigning idle links to the communication-intensive node pairs. A fast circuit scheduling strategy is proposed to dynamically match topologies to the traffic demand. Furthermore, a distributed routing algorithm is designed to adaptively route traffic over the dynamic network. We built a testbed consisting of 8 hosts and it can reconfigure the network topology based on communication demands. We implement several applications on different network configurations, and the experimental results show that compared to Helios and RotorNet, X-NEST+ can reduce the completion time of the HPC applications by 10% ~ 36%. For two sequentially executed ML applications, it can reduce the total completion time by up to 8% ~ 15%.

## 2. System Architecture and Control Plane

Figures 1 (a) and (b) give an overview of the X-NEST+ architecture, which is organized into intra- and inter-pod interconnection structures. As shown in Fig. 1 (b) and (c), there are  $k$  ( $k$  is an even number) Top-of-Rack (ToR) switches in one pod, each connecting  $k$  servers. These switches are equally divided into two groups. Each ToR switch in one group uses  $k/2$  25 Gbps electrical links to connect to every ToR switch in the other group. Overall, these  $k$  ToR switches use their electrical links to form a bipartite graph, which is used to deliver the control packets and to aggregate the inter-pod flows. In addition to this electrical network portion, every pod contains  $r$  MEMS switches, each using its  $k$  100 Gbps optical links to connect to all the ToR switches in the pod. This optical network portion is mainly responsible for the intra-pod communication. The ToR switches are given addresses of the form

(*pod*, *switch*, *k*), where *pod* denotes the pod number to which this ToR switch belongs, and *switch* denotes the position of this ToR switch in the pod. X-NEST+ contains  $k$  pods. These pods are connected by the  $2qk$  MEMS switches (core MEMS switches) in the top layer, where  $q = \lceil k/4 \rceil$ . Given the 25 Gbps electrical links and the 100 Gbps optical links, the ToR switch can achieve nonoversubscription by using  $q$  optical links for the upstream connections and  $k$  electrical links for the downstream connections. To ensure high connectivity, each ToR switch uses  $2q$  optical links to connect the core MEMS switches. As shown in Fig. 1(a),  $k$  ToR switches in the same pod are connected to  $2qk$  different core MEMS switches. Within different pods, the ToR switches with the same *switch* ID are connected to the same core MEMS switch. In addition, the ToR switches and MEMS switches are connected via an electrical network to the controller, which collects the traffic information, calculates the network topology, and informs the MEMS switches to configure accordingly.

An efficient control plane can quickly detect traffic fluctuations and make appropriate decisions. The controller in X-NEST+ periodically takes the amount of data communicated between the racks and converts it into a traffic matrix. To satisfy all possible communication relationships, the traffic matrix will be split into multiple sub-traffic matrices in a diagonally symmetric manner, as shown in Fig. 2 (a). Each sub-traffic matrix has its corresponding MEMS configurations. Keeping only a few fixed inter-port connections allows the MEMS switches to have both low switching latency and high port counts [3]. The controller will calculate the proportion of the traffic of each sub-traffic matrix in the total ( $W_i$ ), and then determine the number of MEMS switches corresponding to the different configurations. To preferentially satisfy the source-destination pairs with larger traffic, the calculation process is performed in descending order. The number of MEMS switches corresponding to the  $i$ -th configuration ( $N_i$ ) is calculated by  $N_i = \text{round}(W_i \times N_{MEMS}/2)$ , where  $N_{MEMS}$  is the total number of MEMS switches in the network. The reason for dividing  $N_{MEMS}$  by two is that only half of the MEMS switches reconfigure each time. An alternate configuration of the two-part MEMS switches can reduce the impact of the reconfiguration on the network performance. Furthermore, the controller only informs the MEMS switches to reconfigure if the topology calculated before and after is different. To allow timely awareness of the topology changes and to increase the connectivity, a distributed routing algorithm is designed. Every ToR switch periodically detects its neighbors and infers current topology. Then it can generate the forwarding table for the direct and indirect traffic.

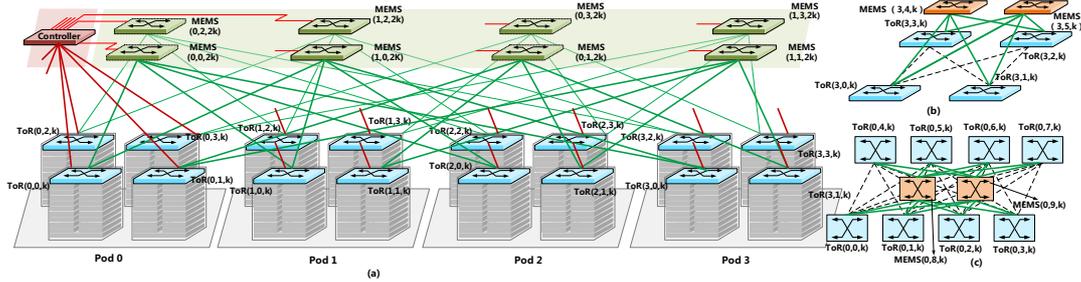


Fig. 1: An overview of the X-NEST+ architecture. (a) The inter-cluster interconnection structure with  $k=4$ . (b) The intra-cluster interconnection structure with  $k=4$ . (c) The intra-cluster interconnection structure with  $k=8$ .

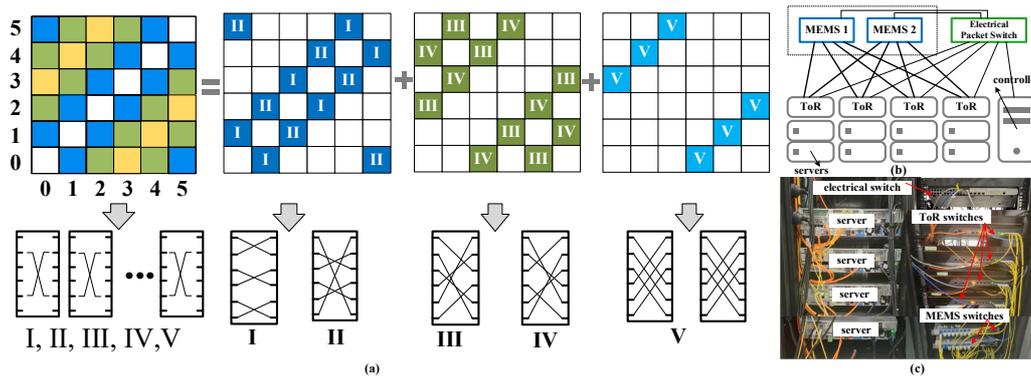


Fig. 2: (a) Diagram of traffic matrix splitting in X-NEST+ ( $k=6$ ). (b) Network topology of the testbed. (c) The testbed.

### 3. Experimental Testbed and Results

In Fig. 2 (b) and (c), we use a combination of MEMS configurations in our testbed to build three network scenarios. Scenario one (S1) uses a MEMS switch and an electrical switch to implement the general hybrid network (i.e., Helios). The optical network is reconfigured with traffic, but is only responsible for the one-hop reachable transmission between ToRs. Scenario two (S2) uses two MEMS switches that are configured according to fixed presets instead of traffic (i.e., RotorNet). The third scenario (S3) uses two MEMS switches to implement the X-NEST+ prototype and scheduling strategies. We compare the performance of several HPC and ML applications

in the three scenarios. We selected Multigrid (MG), Conjugate Gradient (CG), and 3D fast Fourier Transform (FT) from the NAS parallel benchmark used on supercomputers at NASA Ames Research Center [6] to test the acceleration of the HPC applications. In the experiment, we found that the ML application does not support reconfiguration during execution. Therefore, we use PyTorch to combine four representative convolutional Neural Network (CNN) models, AlexNet, ResNet-50, VGG-16 and VGG-19 in pairs to build six combined ML applications, which support synchronization by the parameter server (PS) or Ring-Allreduce algorithms. Fig.3 (a) and (b) show the completion time of the above HPC application under different problem sizes. The network is reconfigured every 300 seconds. Fig.3 (c) and (d) show the time to reach an accuracy rate of 0.8 (TimeToAcc) under different combinations of the ML applications and synchronization algorithms. We reconfigure the optical network during ML application switching. In Fig.3 (a), the completion time in S1 is the longest because the bandwidth of the electrical links is smaller than that of the optical links. The completion time is not much different between S2 and S3 because most of the applications whose completion time is less than 300 seconds are only configured once at the beginning. Moreover, when the amount of data is small, the network flexibility will not significantly reduce the completion time or even introduce more overhead. In Fig.3 (b), compared with S2, the completion time of CG, MG and FT in S3 decreased by 10%, 18% and 36%, respectively. This shows that the flexibility of X-NEST+ begins to play a role as the scale of the problem grows. Fig.3 (c) and (d) shows that it takes the longest TimeToAcc for all the combined applications in S1, because the bandwidth of the electrical link is small, and at this time, it cannot also benefit from optical reconfiguration during application execution. There is no obvious acceleration of the application using the PS algorithm for synchronization between S2 and S3. Because the reconfigured topologies make no difference for the PS algorithm. If the application uses the Ring-Allreduce algorithm for synchronization, The TimeToAcc of the applications in S3 reduces by 15% and 8% compared with S1 and S2, respectively.

#### 4. Conclusion

We propose a rack-level reconfigurable optical interconnect X-NEST+ for HPC and ML applications. A fast control plane and the distributed routing algorithm are designed to make the efficient circuit scheduling and traffic delivering. The experiment results show that X-NEST+ can effectively accelerate HPC and ML applications.

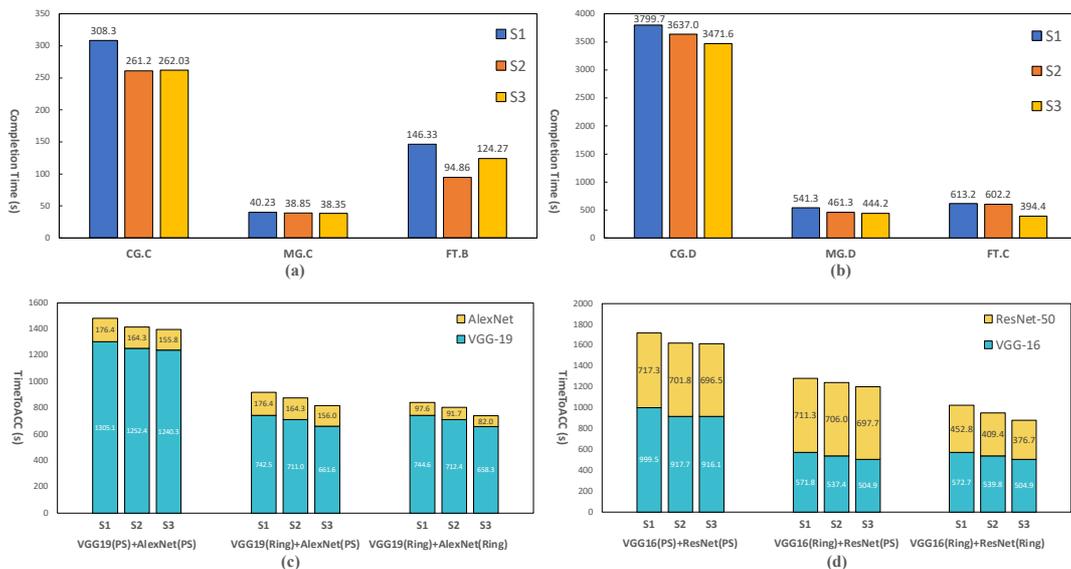


Fig. 3: (a) and (b) are the completion time of each HPC application under different problem sizes. (c) and (d) are the TimeToAcc of each ML application under different model combinations and synchronization algorithms.

#### References

- [1]. M. Khani et al., "SiP-ML: high-bandwidth optical network interconnects for machine learning training," in ACM SIGCOMM, 2021
- [2]. N. Farrington et al., "Helios: a hybrid electrical/optical switch architecture for modular data centers," in ACM SIGCOMM, 2010
- [3]. K. Wen et al., "Flexfly: enabling a reconfigurable dragonfly through silicon photonics," in Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC'16), 2016
- [4]. W. M. Mellette et al., "RotorNet: a scalable, low-complexity optical datacenter network," in ACM SIGCOMM, 2017
- [5]. Y. Lu et al., "X-NEST: a scalable, flexible, and high-performance network architecture for distributed machine learning," Journal of Lightwave Technology, vol. 39, pp. 4247-4254, 2021
- [6]. NAS Parallel Benchmarks, URL: <https://www.nas.nasa.gov/software/npb.html>

This work was supported in part by the National Key R&D Program of China under Grant No. 2018YFE0202800, the National Natural Science Foundation of China under Grant No. 61901314 and 61934002, the Natural Science Foundation of Shaanxi Province for Distinguished Young Scholars under Grant No. 2020JC-26. This work was also supported by The Youth Innovation Team of Shaanxi Universities.