A Tale for Many: Integrated Control Mechanism of Optical Circuit Switching for Data Center and Distributed Deep Learning System

Cen Wang,[1,*] Yuta Wakayama,[1] Noboru Yoshikane,[1] and Takehiro Tsuritani[1]

[1] *Photonic Transport Network Laboratory, KDDI Research, Inc., Saitama, Japan, 356-8502*

cenwang@ieee.org, xce-wang@kddi.com

Abstract: We propose an integrated control mechanism of optical circuit switching for both general data center traffics and deep distributed learning applications. Semi-physical evaluations show a relative throughput of 1.27 and a 6.18× speedup in a 256-block network constructed by MEMS-based optical switches.

1. Introduction

Since bandwidths consumption by applications such as the distributed deep learning (DDL) rapidly increase and packet switches are touching the ceil of processing capabilities, optical switching has become a generally promising approach to the evolution of data center networks (DCNs). Controlling optical switches to divert DCN traffics [1, 4, 5] has been comprehensively studied. Subsequently, various optical switching architectures have been proposed to support DDL systems [6, 8, 9]. However, it lacks cost and practical considerations that a specific architecture is built for a specific use case. Discovering a strategy to jointly support general DCN and DDL systems through a mature, low-cost, and high-scalability architecture has become an emerging goal.

Therefore, in this work, on a clos (or clos-like) network constructed by micro-electro mechanical system-based (MEMS) optical switches, we propose an joint control mechanism that simultaneously optimizes general DCN traffics and DDL jobs. More specifically, we attempt to separate some ports and connections (i.e., degrees) to achieve the topology-adaptive fast reconfigurations [7] for accelerating the communication times of the DDL jobs, and then to maximize throughput for the general DCN traffics on the remaining topology through allowing 2-hop forwarding and hour-level reconfigurations (i.e., topology engineering). Considering that scale-limited experiments are difficult to demonstrate the scalability of the solution, we develop a semi-physical platform to perform large-scale evaluations. The results show a relative throughput of 1.27 for the general DCN traffics and a 6.18× speedup for the DDL jobs in a network with 256 blocks (we do not limit the specific network elements connected to the optical switches, which can be GPUs, servers or racks, etc., collectively referred to as the blocks). In addition to the DDL, a general DCN supports various traffics such as web search, content distribution and cloud computing, as a result, our solution achieves “a tale for many”.

2. Control Mechanism

Fig. 1 shows the design principle and the algorithms of the proposed control mechanism. A uniform mesh topology cannot accommodate a non-uniform general traffic matrix (e.g., *Gravity* distribution [2]), as shown in Fig. 1 (a). To simplify, we use a dimensionless number to represent the bandwidth demand between two blocks. In a real system, this number multiplied by the capacity of a link is the traffic in *bps*. For block w, the number of connections (referring to block degree D_w here and after, a higher number of the block degrees are equivalent to a higher bandwidth) to the other blocks can be adjusted to match the traffic demands between the block pairs. When the demand of a block pair cannot be met, throughput is optimized by dispersing the traffic of this block pair onto different paths with a hop-limited constrain, as shown in Fig. 1 (b). Moreover, when the general DCN traffics are successfully mapped, there are still available block degrees, which are used to adapt DDL jobs with different parallelisms [7] and reduction orders [1]. For example, two DDL jobs can perform *Rabenseifner's AllReduce* and *Ring AllReduce* when $D = 1$ (Fig. 1 (c)) and $D = 2$ (Fig. 1 (d)), respectively. Therefore, the optical switches are divided into the slow reconfiguration and the fast reconfiguration domains, as depicted in Fig. 1 (e).

We implement the topology engineering and the topology adaptation through the two-algorithm processes shown in Fig. 1 (f), which respectively correspond to throughput maximization for general DCN traffics and dynamic topologies matching for DDL parallel patterns. During the process, we constantly try to increase the number of reserved D to support the DDL jobs, and constantly evaluate whether the remaining D can accommodate the general DCN traffics. The topology engineering is inspired by *Jupiter Evolving* [2] and is modeled as a linear programming (LP) problem. The improvement lies in simplifying the constraint formula and reducing the number of constraints by designing the $\mathbf{X}$ matrix. $\mathbf{X}_{(N^2-N)\times N} = \{x_{iw}\}$, where x_{iw} refers to a fraction of a traffic $< s_i, d_i, r_i >$ (s_i, d_i and r_i are source block, destination block and bandwidth request, respectively) between block

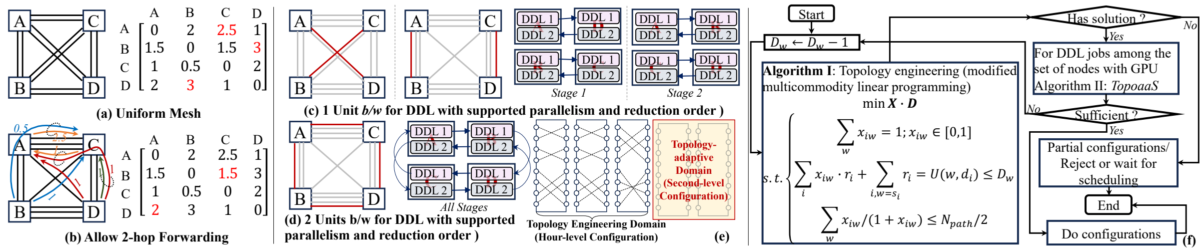

Fig. 1. Control mechanism design.

pair i forwarding along the path p_w (the route is s_i, w, d_i; if $w = s_i$, $x_{iw} = 0$; if $w = d_i$, it indicates the direct path between s_i and d_i; and if no p_w exists, $x_{iw} = 0$). Additionally, *Jupiter Evolving* [2] does not limit the number of traffic divisions, which greatly increases implementation complexity or easily causes out-of-order effects in a real system. $\mathbf{D}$ is the hop matrix from s_i to w, and minimizing $\mathbf{X} \times \mathbf{D}$ through proper searching of an $\mathbf{X}$ is the objective of the LP model. We leave it a future work to tune $\mathbf{X}$ by machine learning methods. We restrict the number of traffic divisions N_{path} by the third constraint. Since LP does not support Boolean variables, traffic divisions are difficult to count, so we relaxed this constrain, and the solution may contain traffic divisions larger than N_{path}. The other constrains are equivalent to that of a common multicommodity problem.

Topology adaption is based on our previous work Topology-as-a-Service (*TopoaaS*) [7]. *TopoaaS* is to do maximum matching on bi-graph to generate a series of isomorphic subgraphs adapting to stages of a parallelism. The corresponding domains in optical switches are dynamically reconfigured according to the subgraphs. When applying for multiple concurrent jobs, *TopoaaS* can additionally perform coincidence and re-accommodation to support a maximum number of jobs. Note that these two algorithms are not yet suitable for on-demand mode, and they perform relying on long-term traffic prediction (left for future work, it is supposed to have prior knowledge of traffic demands in this paper) between block pairs and DDL job requests submitted in advance.

3. Semi-physical Evaluation

Our experimental setup includes $N_{block} = 4$ blocks and there are $D_0 = 2$ block degrees from each block to every other block. Each block has a GPU and extra 2 block degrees for DDL (total $D_w = D_0 \times (N_{block} - 1) + 2 = 8$). The blocks are high performance servers and the link capacity is 10 Gbps. We use 96×96 Polatis (MEMS) optical switch and virtualize it according to D_0 and extra D for a GPU block. As shown in Fig. 2 (a), we use an Ethernet interface card to bridge physical experiment platform and a soft platform deployed in a server. The core simulator in the soft platform is developed by discrete event engine. The timing of the core simulator is the same as the real-world time for synchronization between physical and soft platforms. To guarantee that software execution does not mismatch with real-world timing, processing logic in a soft-block should not be too complex. Thus, a traffic emulated in the soft platform is only with $< s_i, d_i, r_i >$ information and without actual payload. The traffics from physical platform needs to be transformed accordingly. The transmission time of each traffic is counted given a setting bandwidth. In this way, scalable evaluations can be achieved.

The traffics the network can accommodate increases as N_{block} and D_w growing. Considering fairness, when the network scale expands from 4 to 256, D_0 remains unchanged (D_w increases accordingly), and the overall traffics remain at 85% of the corresponding total uniform mesh traffics. As for DDL jobs, the number of GPU blocks in a DDL job is the same as N_{block} of the network. Among Fig. 2 (b) to (d), we mark the data obtained from purely physical experiment. The other results are from the semi-physical platform.

We first evaluate the performance of the topology engineering part in the joint control mechanism through the metric of relative throughput. Given a bidirectional N_{block} all-to-all general traffic matrix, relative throughput is the maximized throughput from the solution of the LP model relative to the throughput on the corresponding uniform mesh topology. In the evaluation, we already exclude the block degrees for DDL jobs and do not count DDL demands in a general traffic matrix. It can be observed in Fig. 2 (b), when D_0 is set larger, the relative throughput is higher, i.e., the optimization performance of topology engineering is better. The reason may be that a large block degree leads to a greater space for topology adjustments, i.e., the flexibility of the traffic division strategy increases. In contrast, as the network scaling, it can accommodate larger traffics. Since the proportion of traffics blocked by the uniform mesh topology decreases in the total traffics, the relative throughput reduces. The topology engineering also tries to minimize hops and the number of traffic divisions. We evaluate such the performance as well in Fig. 2 (c). The number of hops is less than 1.4, and become smaller when D_0 increases due to easier searching of direct paths. The number of hops increases slightly as the network scale expanding, owing to a higher probability of dividing a traffic onto a 2-hop path. The traffic division is the number of paths a block-pair traffic is divided onto. There is a correlation between the varying trend of traffic divisions and that of the number of hops, because that when difficult to find a direct path, it means a traffic is more likely to be divided. In addition, it can be seen that the average traffic divisions exceeds 2.0. It means that a single traffic could be divided more

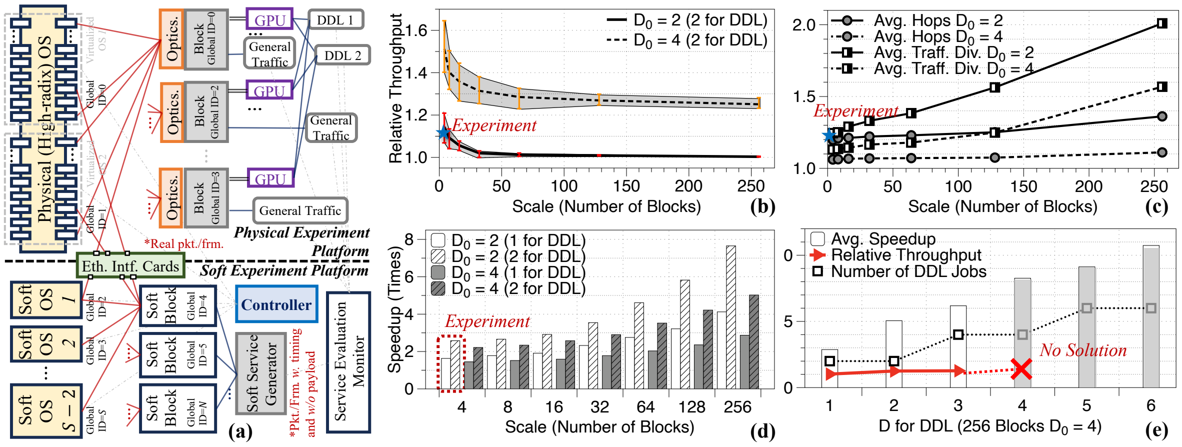

Fig. 2. Semi-physical platform and evaluation results.

than $N_{path} = 2$ caused by the aforementioned relaxed constraint.

Subsequently, we evaluate the performance of the joint control mechanism for the DDL jobs. The speedup is using the communication time in bad case to divide optimized communication time. The bad case in the evaluations is that given a uniform mesh topology and a traffic matrix, the demands of a DDL job is mapped onto direct paths with remaining bandwidth (such the bandwidth can be smaller than a demand) by first fit. In such the case, the communication time is determined by the fitted path with the minimum remaining bandwidth. We use this metric also to demonstrate the necessity of the topology adaption to a DDL job. We do not reject a DDL job (although it may be rejected in a real system) because this would render it unevaluable. Fig. 2 (d) shows that a larger D_0 results in a less speedup. The reason is that in a network with a larger D_0, a DDL job tends to find direct paths with higher remaining bandwidths, thereby the communication time slowdown is not as much as that in a network with a smaller D_0. The speedups become higher as the network scale expanding because that the number of blocks for a DDL job increase and a block-pair demand more easily suffers insufficient remaining bandwidth compared to a DDL job using less blocks.

In Fig. 2 (e), in the 256-block network with $D_0 = 4$, We deduced the process of continuously providing block degrees to the DDL jobs in the proposed control mechanism. Since the number of block degrees used for the general DCN traffics is reduced, and topology engineering can still guarantee the same amount of the general DCN traffic to be accommodated, the relative throughput increases slightly. It can be seen that when $D = 3$ block degrees are provided for the DDL jobs, the relative throughput can keep at 1.27 for the general DCN traffics while achieving a 6.18$\times$ speedup for DDL jobs. When $D = 4$, the topology engineering then has no solution. On the other hand, a larger D means more DDL jobs can be executed simultaneously. Given a topology, there is always an upper limit for traffic accommodation. If multiple concurrent DDL jobs are expected to be satisfied, the density of the general DCN traffics needs to be reduced or increasing D_w is required further. The number of constrains in topology engineering and computing complexity of bigraph matching in topology adaption are both $O(N_{block}^2)$. The two-algorithm control mechanism takes tens of minutes to finish, thus can meet the hour-level reconfigurations if predicting traffics in advance.

4. Conclusion

Building multiple optical switching data center/computing system networks in multiple scenarios goes against the principle of evolution due to high cost and low efficiency. Therefore, this paper proposed an low-cost and practical integrated optical switching control mechanism of MEMS-based optical circuit switching that simultaneously optimizes general DCN traffics and accelerates DDL jobs, through topology engineering and *TopoaaS*, respectively. We developed a semi-physical platform to perform demonstration while realizing large-scale evaluations. The results show a 1.27 relative throughput for the general DCN traffics while achieving a 6.18$\times$ speedup for the DDL jobs in the 256-block network.

References

1. W. Wang, M. Khazraee, Z. Zhong, and M. Ghobadi, et. al., *NSDI '23*, pp. 739–767.
2. L. Poutievski, O. Mashayekhi, and J. Ong, et. al., *SIGCOMM '22*, pp. 66–85.
3. C. Wang, and N. Yoshikane, et. al., *Computer Networks*, Volume 214, 2022, 109191, ISSN 1389-1286.
4. W. M. Mellette, R. McGuinness, and A. Roy, et. al., *SIGCOMM '17*, pp. 267–280.
5. M. Channegowda, and T. Vlachogiannis, et. al., *OFC 2016*, paper W3F.2.
6. M. Fariborz, and X. Xiao, et. al., *Journal of Lightwave Technology*, vol. 39, no. 4, pp. 1212-1220, 15 Feb.15, 2021.
7. C. Wang, N. Yoshikane, and D. Elson, et. al., *Journ. of Opt. Comm. and Netw.* Vol. 16, No. 1, 2023, *to appear*.
8. M. Khani, and M. Ghobadi, et. al., *SIGCOMM '21*, pp. 657–675.
9. H. Liu, R. Urata, and K. Yasumura, et. al., *SIGCOMM '23*, pp. 499–515.