Photonic switched networking for data centers and advanced computing systems

Paraskevas Bakopoulos[1], Giannis Patronas[1], Nikos Terzenidis[1], Zsolt-Alon Wertheimer[3], Prethvi Kashinkunti[2], Dimitris Syrivelis[1], Eitan Zahavi[3], Louis Capps[2], Nikos Argyris[1] , Luke Yeager[2], Julie Bernauer[2], Elad Mentovich[3]

[1] NVIDIA, Ermou 56, Athens 10563, Greece
[2] NVIDIA, 2788 San Tomas Expy, Santa Clara, CA 95051, USA
[3] NVIDIA, Hakidma 26, Ofer Industrial Park, Yokneam 2069203, Israel
paraskevasb@nvidia.com

Abstract: We explore optical switching to extend network programmability to the physical layer. We present applications of our Layer-1 SDN for improving fabric resilience against hardware failures and saving network power and cost in Deep-Learning training. © 2024 The Author(s)

1. Introduction

Enterprise and hyperscale datacenters are progressively built around workloads utilizing Artificial Intelligence and Machine Learning (AI/ML). The advent of Deep Learning has accelerated scaling of training compute, driving the required FLOPs to double approximately every 6 months, outpacing single GPU scaling rate by a factor of five [1]. To keep up with the need for FLOPs, scale-out systems are needed to create larger native GPU domains interconnected with high-speed links to keep them fully utilized. For example, the latest NVIDIA DGX SuperPOD reference architecture brings together 1,024 H100 GPUs and can scale to tens of thousands of GPUs. In the reference SuperPOD, the compute fabric comprises more than 2,000 cables in a rail optimized, two-level fat tree topology that extends to three-levels for larger GPU counts [2]. At this scale, the network is an integral part of the system and its characteristics reflect on overall system power, cost and availability.

Optical Circuit Switches (OCSs) are introduced to respond to these challenges. An optically switched fabric offers physical layer reconfigurability and programmability, re-wiring the network and allocating physical connections on demand. The OCS fabric demarcates from the rigid physical infrastructure cabling used today that is considered fixed after deployment, and enables physical topology adaptation at runtime. This capability substantially extends today's software-defined network infrastructures that are programmable down to Layer 2 (L2), adding physical layer connections as a programmable resource. We introduced a workflow that extends the network's software-defined capabilities to Layer 1 (L1) and demonstrated its operation with the InfiniBand (IB) Subnet Manager (SM) [3].

Introducing programmability to L1 facilitates a multitude of new network operations but also comes with new implications regarding integration of the new functionality to the software-defined network infrastructure. This paper describes two applications of the L1 programmable dataplane: 1. fabric resilience against hardware failures and 2. training of Deep Learning (DL) models. Each application is presented separately for the sake of clarity, but the functionalities involved in each concept could be combined in a single architecture.

2. Resilience against hardware failures

Multi-node applications rely heavily on large networks and are highly susceptible to crashing when networks fail. In the case of large systems, switches or transceivers can cause a failure from every 3 to every 120 hours [4], resulting in system impact being measured in hours. Failures can result in downtime causing lost revenues due to the unavailability of the system until the next maintenance event. The common policy of rolling back to the latest checkpoint exacerbates the loss as it discards all computations performed after the checkpoint.

Recovery of failures in networks of large systems currently focuses on adapting the routing configuration to exclude the failed paths where possible, since there is little or no ability to change the actual physical connectivity in real time during a failure. These approaches come with two major limitations: First, they can be used only when alternative paths exist: some types of failures cannot be mitigated with this approach, e.g. failures on the leaf switches that result in disconnecting the servers from the network. Second, this approach cannot recover the full performance of the cluster in real time since the use of alternative paths often results in oversubscription and hence reduced performance. Multihoming can resolve this by replicating (part of) the network, e.g. connecting the server to two switches. On the downside, it requires more hardware and increases deployment and operating costs.

We leverage the L1 programmable dataplane to provide real-time recovery in an efficient, scalable, automated and software-defined manner with minimal addition of redundant hardware. We focus on the network as a major point of failure by using optical technology to dynamically heal the network connections and make failure recovery at full capacity an automated process. The hardware resilience solution relies on the addition of OCSs between the switching layers (hosts-to-leaves, leaves-to-spines, spines-to-cores), as depicted in Figure 1(left), that dynamically rewire the network on demand. To offer resilience against switch failures, redundant network switches are added to the network. The redundant switches are connected to available ports of the OCSs, similarly to the regular switches. When a device failure is detected, the corresponding optical switches are properly configured to disconnect it from the network and replace it with a redundant unit. The number of redundant switches defines a customizable level of resilience in the network, i.e., the number of failures that can be recovered simultaneously, which can vary depending on the system requirements.

The OCSs are controlled by a custom control plane software called the Optical Fabric Manager (OFM) that serves as an SDN stack extension for physical layer resources. The OFM calculates the target topology of the cluster, enforces the configuration to the physical layer devices and notifies the L2 SDN controller (IB Subnet Manager) of the changes. This enablement facilitates automatic failover switching to the redundant equipment as well as additional functionalities, such as predictive maintenance and scheduled rollout of software and hardware upgrades.

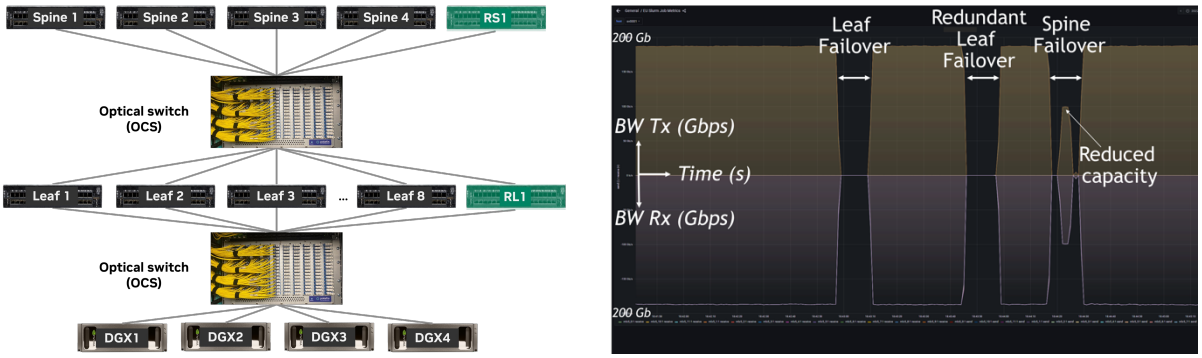

Fig. 1. Left: Resilience architecture and experimental setup. Right: Bandwidth recovery demonstration with NCCL all-reduce. Telemetry data of Tx (yellow) and Rx (purple) bandwidth over time.

We have demonstrated the application of hardware resilience with our L1 programmable dataplane in a small-scale DL testbed. The setup consists of 4 DGX A100 servers and 14 IB Quantum switches, connected as 8 Leaves, 4 Spines, 1 redundant Leaf (RL) and 1 redundant Spine (RS). The servers and switches were populated with 114 CWDM 200 Gb/s transceivers. A single commercial OCS was partitioned to serve both server-to-leaf and leaf-to-spine connections.

We emulate IB switch failures and trigger the OFM to initiate the failure mitigation process. Figure 1(right) shows the results of our tests with NCCL [5] collective communications library. The telemetry measurements show bandwidth of all IB interfaces of a DGX server over time, for all-reduce benchmark. Leaf failures would normally result in an application crash and the IB interface going offline until the failure is fixed. In case of spine failures the application would not crash given the availability of alternative paths, but the system would work at reduced capacity. When our resilience system is enabled, the full performance of the cluster is restored within a few seconds. We have tested the setup with UCX [6] and NCCL microbenchmarks and real-world applications (MLPerf BERT, ResNet-50 and NeMo Megatron) and achieved the same performance. In our experiments we used the IB network, but the same concepts can be applied to other fabrics such as Ethernet or NVLink.

3. Deep Neural Network training

The second application of our L1 programmable data plane is DL training clusters. We focus on prominent examples of DL as manifested by large language models (LLMs) and deep learning recommendation systems (DLRMs), with exceptionally promising applications and transformational impact to society.

A multi-level fat tree (FT) network topology based on electrical packet switches can be used to interconnect all the nodes in the cluster at full bisection bandwidth. This simplifies placement of jobs and routing but comes at a steep price. The energy consumption and cost of the FT network account for a considerable portion of the total system power and cost, which is increasing with every speed upgrade. It is possible to replace the multi-level FT with a

leaner network fabric considering the communication needs of the application. Firstly, connections across jobs are not needed; in fact isolating jobs is advantageous as it mitigates cross-job interference and improves security. In addition, each DL job follows well-defined traffic patterns implemented through communication collectives, and therefore does not need all the bisection at all levels. We propose a network fabric that replaces higher layers of the FT topology with an OCS layer. We show that LLM and DLRM jobs can be served with this L1 programmable data plane without impacting cluster efficiency, yet saving network latency, energy (>50%) and cost (>30%).

The underlying principle leverages the programmable data plane to adjust the network topology according to the expected communication pattern for each job [7-8]. Our approach assumes knowledge of basic information on the jobs arriving to the cluster, such as the in-network collectives and parallelization approach followed. We have developed a resource management tool that places the job across GPU, allocates network resources and assigns them to circuits through the OCS implementing the target network topology. For each job and allocation of ranks, the programmable data plane implements a suitable topology (e.g. tori and full graphs) that provides full bisection bandwidth for the expected traffic pattern. We have developed specific rules and heuristics for resource management targeting LLM and DLRM jobs. For the current work, we rely on DGX servers with multiple GPUs interconnected through NVLink. The DGX servers are connected to the programmable data plane through InfiniBand DPUs.

Flattening the network with the L1 programmable data plane does not affect per-job bandwidth compared to a FT topology, as we leverage the known communication patterns. We further investigate the effect of the simplified data plane on the utilization of the system, to assess whether our strategies may result in stranded resources. We developed a network solver that takes as input a queue of DL training jobs and allocates resources for their placement while creating the required network topologies. We compare utilization over time for a system employing the programable dataplane against a system with an ideal FT. We show that the system utilization remains well within 1% of the FT based system for DLRMs, LLMs as well as for their mix.

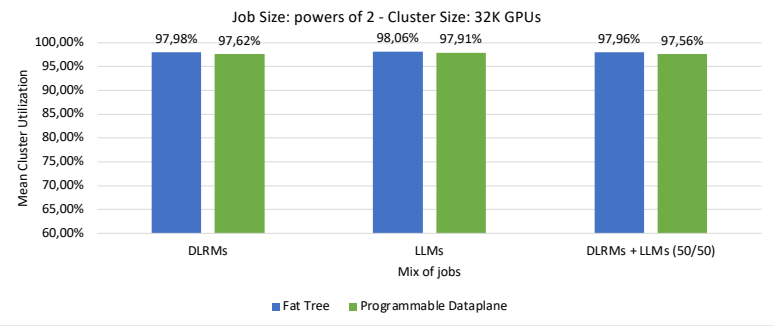

Fig. 2. System utilization for different mixes of jobs

3. Conclusion

Photonic switching extends SDN programmability to the physical layer. The programmable data plane enables new functionalities to sustain growth in scale-out systems. We demonstrated real-time failover to redundant IB switches, successfully restoring the full performance of the cluster to improve availability. We also presented a flat architecture for LLM and DLRM training that saves network power and cost without compromising bandwidth or cluster utilization. As scale is capped by the power available on-site, improving energy efficiency is paramount.

References

[1] Epoch (2023), "Key trends and figures in Machine Learning". Published online at epochai.org. Retrieved from: 'https://epochai.org/trends'
[2] NVIDIA DGX SuperPOD: Next Generation Scalable Infrastructure for AI Leadership. Retrieved from: https:/docs.nvidia.com/dgx-superpod-reference-architecture-dgx-h100.pdf
[3] G. Patronas et al., "Software-defined, programmable L1 dataplane: demonstration of fabric hardware resilience using optical switches," in Optical Fiber Communication Conference (OFC) 2023, Technical Digest Series (Optica Publishing Group, 2023), paper Th2A.15.
[4] Rachee Singh, Muqeet Mukhtar, Ashay Krishna, Aniruddha Parkhi, Jitendra Padhye, and David Maltz. 2021. Surviving switch failures in cloud datacenters. SIGCOMM Comput. Commun. Rev. 51, 2 (April 2021), 2–9. https://doi.org/10.1145/3464994.3464996
[5] https://developer.nvidia.com/nccl
[6] https://docs.nvidia.com/networking/display/hpcxv24/Unified+Communication+-+X+Framework+Library
[7] N. Jouppi et. al., "TPU v4: An Optically Reconfigurable Supercomputer for Machine Learning with Hardware Support for Embeddings," in ISCA '23: Proceedings of the 50th Annual International Symposium on Computer Architecture, No. 82, June 2023.
[8] W. Wang et. al., "TopoOpt: Co-optimizing Network Topology and Parallelization Strategy for Distributed Training Jobs," in Proceedings of the 20th USENIX Symposium on Networked Systems Design and Implementation. April 17–19, 2023, Boston, MA, USA.