Reconfigurable Lightwave Fabrics for ML Supercomputers

Hong Liu, Ryohei Urata, Kevin Yasumura, Xiang Zhou, Roy Bannon, Jill Berger, Pedram Dashti, Norm Jouppi, Cedric Lam, Sheng Li, Erji Mao, Daniel Nelson, George Papen, Mukarram Tariq, and Amin Vahdat
Google LLC

Abstract: We present the large-scale, production deployment of reconfigurable Lightwave Fabrics (LWF) for Machine Learning (ML) supercomputers. These fabrics consist of a custom developed optical circuit switch (OCS), circulators, and WDM transceiver technologies. The use of a LWF dramatically enhances the current generation 4096 tensor processing unit (TPU) system in both availability (up to 3x) as well as performance (up to 3.3x) with modest power and cost increases (1% and 6%, respectively).
OCIS codes: (060.0060) Fiber optics and optical communications; (060.4250) Networks

1. Introduction

Over the past several years, the growth of machine learning (ML) has been insatiable, with the number of model parameters scaling to one trillion and beyond. However, developing systems to run these large synchronous models is limited by reliability, due to the inherently larger number of imperfect components along with the sensitivity of synchronous models to hardware faults. As a solution to this fundamental problem of system size versus reliability/availability, we have applied our Lightwave Fabric (LWF) technology for use in our production ML systems [1]. The LWFs are constructed using large-radix optical circuit switches (OCSes), circulators, and custom WDM optical transceivers. The benefits of LWFs for datacenter networks (DCNs) have been described previously [2]. The reconfigurability of the LWF enables high availability of large ML systems at reasonable cost and power increases relative to an equivalently sized static system. In addition, we demonstrate substantial speed up in model training times by leveraging the reconfigurability created by LWFs to match and optimize the hardware configuration to the ML model. We also gain significant enhancements in utilization, modularity, deployment, and security [1].

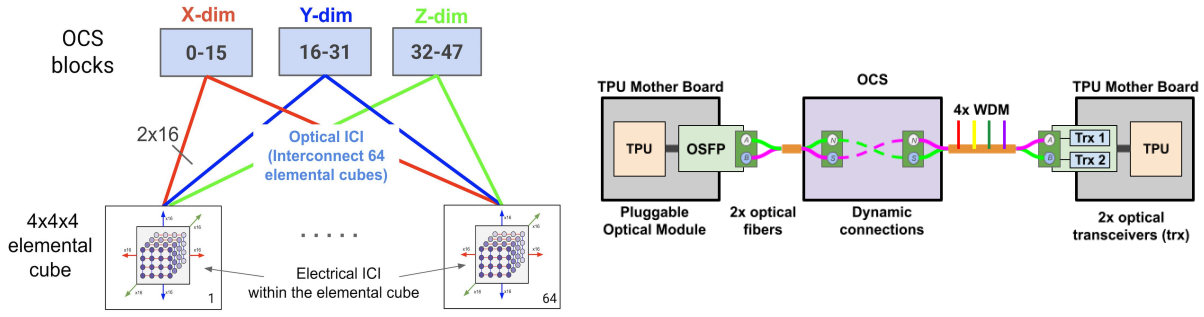


Fig. 1: Left: ML superpod containing 4096 TPUs arranged in a 3D torus topology. Each elemental cube contains 4x4x4=64 TPUs, with opposing faces of cubes interconnected via links through the OCS. Right: WDM bidirectional (bidi) links through the OCS. WDM and circulators increase multiplexing for efficient OCS port and fiber use.

2. ML TPU Superpods

The left side of Fig. 1 illustrates the ML system or superpod leveraging LWFs. Shown at the bottom of the figure, the fundamental building block consists of TPU ASICs arranged in a 4 × 4 × 4 3D torus configuration/topology. Each of these elemental cubes thus contains 64 TPUs, with intra-cube connections achieved electrically within one rack. 64 of these rack-sized cubes are then optically connected to the OCSes such that any x, y, or z face of a cube can connect to the opposite x, y, or z face of any other cube. The right side of Fig. 1 illustrates an end-to-end bidirectional (bidi) inter-cube optical link. The TPU connects to a pluggable OSFP optical transceiver, which uses WDM and a circulator to multiplex four bidirectional channels per single mode fiber strand and path through the OCS [1, 3]. The high degree of multiplexing enables efficient use of the fiber plant and reduces the total number of OCSes. This reduction improves the overall system availability, as described in the next section. Similar to the DCN

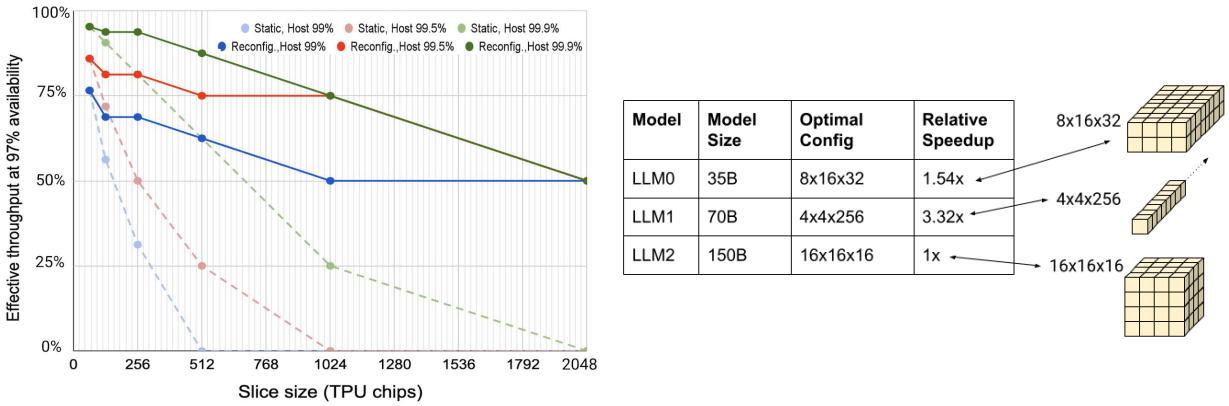
use case [2], the OCS and single mode fiber are data rate and modulation format agnostic, allowing reuse across multiple generations and the associated CapEx costs to be amortized across multiple decades versus multiple years.

The cluster-level job scheduling system uses the LWF to dynamically connect one or more elemental cubes to create compute "slices" and program the inter-chip-interconnect (ICI) network for the slice. Most slices have a 3D torus topology with wrap-around links. The number of cubes connected per dimension of the torus can be different, i.e., when the entire pod of $64^2 = 4096$ nodes is used, slice topologies ranging from $4 \times 4 \times 256$ to $16 \times 16 \times 16$ can be configured with the minimum increment of four set by the size of the elemental $4 \times 4 \times 4$ cube. Furthermore, multiple different-sized workloads with differing communication patterns can be configured within a single pod. The non-blocking nature of the OCS permits the scheduling of new slices of different sizes that are physically isolated from the other jobs that are already scheduled and running.

3. Superpod Evaluation

Availability: The ability to reliably compose slices is critically dependent on the availability of the entire LWF. We readily achieve >99% availability in our production environments [1]. Correspondingly, the overall effective throughput (i.e., goodput) for larger slices significantly improves compared to a static configuration over a wide range of host/server availability conditions.

To quantify this improvement, we use a single OCS availability of 99.9% and hold the overall system availability constant at 97%. For a single elemental cube slice (64 TPUs), to achieve this target availability requires holding back some elemental cubes within the pod so that there are enough working cubes to achieve the target system/cube availability. The number of elemental cubes that are held back increases with the failure rate of an individual server. Larger failure rates decrease the goodput because more cubes need to be held back to achieve the target system availability. This can be seen in the left side of Fig. 2 for a single elemental cube slice (64 TPUs). As the server availability increases from 99% to 99.9% (blue to green curves), the goodput increases because fewer elemental cubes need to be held back. For a slice that is a single cube, no reconfiguration between cubes is used and thus the goodput is the same for both the static and the reconfigurable topologies (dashed, solid lines).



Model	Model Size	Optimal Config	Relative Speedup
LLM0	35B	8x16x32	1.54x
LLM1	70B	4x4x256	3.32x
LLM2	150B	16x16x16	1x

Fig. 2: Left: Goodput/efficiency of the superpods as a function of slice size. Right: Relative model training speed up achieved by optimizing slice shape. The baseline for comparison is the $16 \times 16 \times 16$ cube performance.

For slice sizes larger than a single cube (64 TPUs), the total number of cubes required for a reconfigurable LWF to achieve the target system availability is much smaller than the number of cubes required for a static fabric because the reconfigurable LWF can swap out a bad elemental cube whereas a static configuration cannot. For slices of the same size that are larger than 64 TPUs, this leads to the goodput for the static configuration (shown as dashed lines) rapidly degrading compared to the goodput for the corresponding reconfigurable LWF (solid lines). As an example, for a server availability of 99.9% (green curves), the static configuration can only support a 1024 TPU slice size with 25% goodput, whereas the reconfigurable superpod can support 1024 slice size with 75% goodput. Given the simplifying constraint that the slices are the same size, as the slice size increases, the overall goodput is eventually dominated by the availability of the slice size and not by the individual server/host availability. At a slice size of 1024, this leads to the convergence of the goodput for a server availability of 99.9% (green curve) with the goodput for a server availability of 99.5% (red curve). Accounting for the hold back, three 1024 slices can be

composed for either server availability leading to a goodput of 75% for both server/host availabilities. In contrast, only two 1024 slices with a goodput of 50% can be composed for the lower server availability of 99% (blue curve) because the hold back needed to achieve a 97% system availability for this case exceeds 1024 TPUs. At a slice size of 2048, which is half the pod size of 4096, only one slice can be composed leading to a goodput of 50% regardless of the server/host availability because some hold back is needed to achieve the target system availability for any server/host availability less than 100%. In practice, a distribution of slice sizes running different size models is used. For further availability improvements, the total number of OCS in the system can be reduced by adopting additional WDM in the transceivers. One approach is to incorporate four additional wavelengths that are offset by 10nm from the traditional CWDM4 grid used for the DCN (8 channel, 10nm grid) [1]. This optimization of interconnect for the ML use case reduces OCS count by two, improving reliability of the LWF and the overall system.

Training Speedup: The reconfigurability of the LWF enables the creation of slices and provides significant performance optimization for large-scale ML model training. In normal operation, the routing is deterministic and set by the slice configuration. For a full-scale superpod containing 4096 TPU V4 chips, the dynamic reconfigurability of the LWF enables changing the shape of the slice to any configuration between a symmetric slice of 16 × 16 × 16 to a highly asymmetric slice of 4 × 4 × 256. We have found that this unique capability is especially important for emerging transformer-based large language models (LLMs) [1]. Performance optimization for these LLMs is critical, because these LLMs are very expensive to train, using hundreds to tens-of-thousands of computing elements including TPUs or GPUs for months [4]. The right side of Fig. 2 shows that the reconfigurability of the LWF unlocks more than a 3.3× performance uplift for production-scale LLMs, using an optimally configured 4096 node TPU V4 superpod compared to the static baseline 16 × 16 × 16 configuration which has the highest bisection bandwidth among all possible static configurations of a 3D torus.

Another important observation from the right side of Fig. 2 is that there is no “one-size-fits-all” optimal slice configuration for ML models. As can be seen, the symmetric 16 × 16 × 16 configuration of the LWF is optimal for some LLMs, while asymmetric configurations are optimal for others. When training on large-scale ML systems, LLMs rely on state-of-the-art model parallelism (including model pipelining) and data parallelism [1] to achieve good training speed. The amount of inherent model and data parallelism for an LLM determines the optimal slice configuration. In general, model size determines the amount of inherent model parallelism, while global batch size determines the amount of inherent data parallelism.

The diverse dependence of ML model performance on the hardware configuration requires the size and topology of the slice to be late binding after hardware is deployed. The ability of the LWF to match the shape of the slice to the ML model is a unique and profound capability that not only achieves substantial performance gains for ML models today but also offers a powerful form of reconfigurability for future ML models at scale.

Efficiency and Utilization: The combination of an elemental cube with only 64 TPU chips and the reconfigurability enabled by the LWF leads to simplified slice scheduling, which increases overall efficiency. In the previous generation TPU V3 pod [4], scheduling a 256-node slice required finding 256 physically contiguous nodes that were idle and functional. For the TPU V4 superpod, the smaller 64-node block size in combination with the use of a non-blocking LWF means that there are many more potential solutions to configure the LWF to a connect a set of four idle, not-necessarily contiguous 4 × 4 × 4 elemental cubes to form a directly connected 256-node slice. In addition, the scheduler is able to defragment the pods more effectively. In practice, we are able to run the TPU V4 fleet at a higher (>98%) utilization than earlier-generation systems despite the need to support 4× larger slices.

4. Conclusion

In this paper, we reviewed our ML superpod incorporating LWFs. Benefits of employing the LWF are higher availability of larger systems/slices, training time reduction with slice shape optimization, as well as efficiency benefits (bin packing) and improved deployment velocity. This was all done at a modest 6% increase in cost and 1% increase in power compared to a static fabric [1].

References

[1] H. Liu, *et al.*, “ Lightwave fabrics: At-scale optical circuit switching for datacenter and machine learning systems,” ACM SIGCOMM 2023.
[2] L. Poutievski, *et al.*, “Jupiter evolving: Transforming Google’s datacenter network via optical circuit switches and software-defined networking,” ACM SIGCOMM 2022.
[3] R. Urata, *et al.*, “Mission Apollo: Landing optical circuit switching at datacenter scale,” arXiv:2208.10041.
[4] N. Jouppi, *et al.*, “Ten lessons from three generations shaped Google’s TPUv4i: Industrial product,” ACM/IEEE ISCA 2021.