View and annotate documents including comments and text highlights

Efficient Live Expansion for Clos Data Center Networks

2 of 20 — Automatic Zoom





also like to complete each stage as quickly as possible, since rewiring does reduce our spare capacity, and thus exposes us to an increased risk of simultaneous failures.

Over the course of a multi-stage expansion, we may need to rewire many links. If we were to directly connect links between switches, the resulting manual labor for moving long wires would be slow, expensive, and error-prone. Instead, we introduce a patch-panel layer in our Clos DCNs (see Fig. 1). These DCNs are three-tier Clos topologies, with tier-1 top-of-rack (ToR) switches connected to tier-2 server blocks, each of which connects to a set of tier-3 spine blocks. By connecting all the server blocks and all the spine blocks through a group of patch-panels, a DCN topology can thus be created and modified by simply moving fiber jumpers on the back side of the patch panels. Each series of rewiring steps can hence be done in proximity to a single patch panel, although an entire stage may require touching several panels.

Our scale has grown to the point where a simple version of this patch-panel-based expansion technique is too slow to support the rate at which we must execute expansions. Therefore, we needed to minimize the number of rewirings per expansion, while maintaining bandwidth guarantees.

The primary contribution of this paper is a minimal-rewiring solver for Clos DCN topology design. In the literature, most Clos DCN topologies are designed purely to optimize cost and/or performance at a single chosen size [1, 18, 26, 29, 35]. In contrast, our solver explicitly considers the pre-existing topology when designing a larger one. Our solver uses Integer Linear Programming (ILP) to directly minimize the total number of rewirings. By enforcing a number of balance-related constraints, the resulting topology is also guaranteed to have high capacity and high failure resiliency. With minimal rewiring, a DCN expansion can be done in fewer stages, while still maintaining high residual bandwidth during expansions.

Because we build each DCN incrementally over a period of years, we need to incorporate new technologies incremen-

ables whenever possible. We have a proof that the aggregated decision variables can be decomposed in a later step (see Appendices) . Our block-aggregation technique can use different aggregation strategies. With the fastest strategy, all 4500 synthesized DCN configurations can be solved within 10 seconds.

We measure the quality of our solutions in terms of a *rewiring ratio*, the fraction of wires between server blocks and spine blocks in the pre-existing topology that must be disconnected during an expansion. When we use block aggregation, we face a tradeoff: aggregation improves runtime scalability, but sacrifices rewiring optimality. However, we cannot predict the aggregation strategy that will produce the best (lowest) rewiring ratio subject to a chosen deadline. Therefore, our *parallel solver* runs multiple minimal-rewiring solvers with different aggregation strategies at the same time, and picks the solution with the lowest rewiring ratio. This allows us to solve about 99% of the synthesized DCN configurations with a rewiring ratio under 0.25; the median ratio is under 0.05. In turn, these low rewiring ratios allow us to significantly accelerate the entire expansion pipeline. For example, under a constraint that preserves 70% of the pre-expansion bandwidth during expansion, our minimal-rewiring solver reduces the average number of expansion stages required from 4 to 1.29.

2 Prior Work on Expansions

Prior work has described DCN designs that support incremental expansion, and techniques for conducting expansions. Our work focuses on Clos topologies, the de-facto standard for large-scale DCNs; most prior work on expansions has used non-Clos designs.

DCell [19] and BCube [8] are built using iterative structures. As a result, they can only supports expansions at a very coarse granularity, which could lead to substantial stranded DCN capacity after expansion. Similar iteratively-designed DCN structures are also proposed in [20, 27, 28].

Annotations — Export

We have a proof that the aggre- gated decision variables can be decomposed in a later step (see Appendices) . Our block-aggregation technique can use different aggregation strategies. With the fastest strategy, all 4500 synthesized DCN configurations can be solved within 10 seconds

a minute ago

This block aggregation strategy is interesting and I'm going to have to read more about it.

a few seconds ago

Therefore, our parallel solver runs multiple minimal- rewiring solvers with different aggregation strategies at the same time

a few seconds ago

I assume decoupling enabled parallelism.

Cancel Comment