A 12-Rack, 180-Server Datacenter Network (DCN) Using Multiwavelength Optical Switching and Full Stack Optimization

Da Wei, Lei Xu*, Xin Jin, Yiran Li, Wei Xu**
Institute of Interdisciplinary Information Science (IIIS), Tsinghua University, Beijing 100084, China
**Torray Networks, 600 Alexander Road, Princeton, NJ 08540.*
***Department of Computer Science, Princeton University, Princeton, NJ 08540.*
Email addresses: weid14@mails.tsinghua.edu.cn; lei.xu@torraynetworks.com; xinjin@cs.princeton.edu; liyr14@mails.tsinghua.edu.cn; weixu@mail.tsinghua.edu.cn

Abstract: We present DFabric, a 12-rack, 180-server DCN using multiwavelength switching and interconnection. DFabric implements real-time network traffic and per-link utilization monitoring, and full-stack optimization by jointly optimizing optical switching and network flow routing.

OCIS codes: (060.4250) Networks; (060.4265) Networks, wavelength routing

1. Introduction

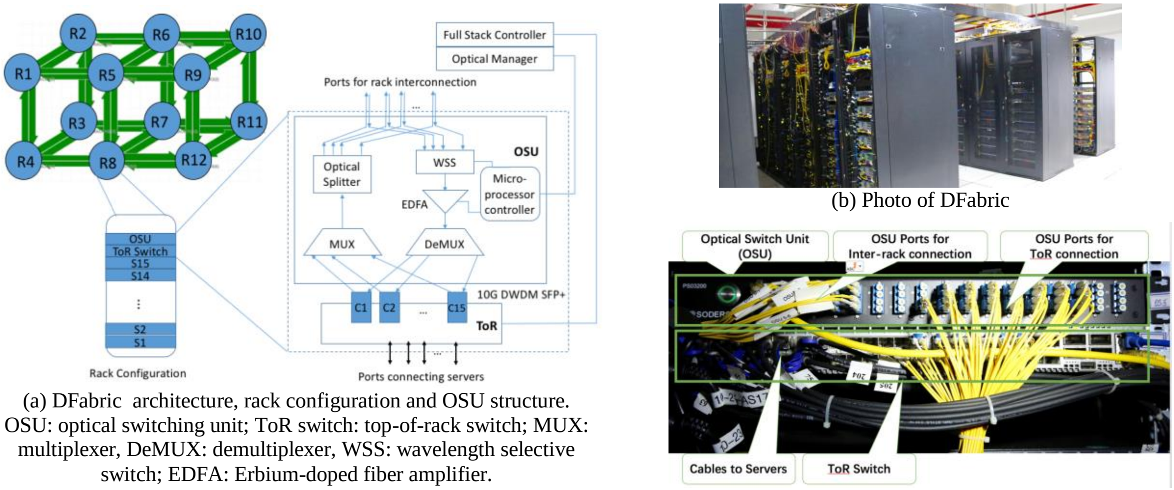
With the fast growth of big data applications and sophisticated software stack, modern data centers host a variety of interactive and batch applications spanning on a large number of servers, and many of these applications are data-intensive. Good scheduling decisions and flexible network operations, which can move the processing closer to its data, can lead to significant performance improvement. However, traditional static oversubscribed DCN topologies can cause performance bottlenecks in terms of having unpredictable latency and network congestion. Researchers have demonstrated flexible optical switching units in data centers that dynamically modify the network topology and bandwidth allocations [1-3]. In order to fully take advantages of the flexibility of an optically switched DCN, it is important to combine efficient bandwidth scheduling algorithms with traffic engineering (TE) capability.

We present DFabric, a 12-rack, 180-server DCN testbed by combining distributed wavelength selective switching, interconnection and state-of-the-art software defined networking (SDN) technology. We designed an online full-stack network optimization system that continuously monitors the computation job submissions, estimates the upcoming workload patterns, and optimizes the bandwidth allocation in the optical layer and the flow-level routing at the network layer. The optimization is designed to support the fast-changing and unpredictable DCN traffic pattern, and targets at one of the most important metrics in DCN: the tail latency [4], which is the direct result of transient congestion and packet loss. We minimize the disturbance of reconfiguration by continuously and incrementally changing a small number of optical links each time, and adopt an advanced network update mechanism to prevent packet loss during the update. We handle the unpredictability of DCN traffic pattern by minimizing the maximum link utilization ratio, leaving as much “headroom” as possible on each link.

2. DFabric Datacenter Network Setup and Management Software

Fig. 1 shows the design and network topology of DFabric, and Fig.1(a) shows our current hyper-cube network topology with 12 racks. Each rack has 15 servers connected to a top-of-rack (ToR) electrical switch (Arista 7050S-64). We use Floodlight controller to control the ToR electrical switches. Each ToR switch has 48 10GbE SFP+ ports. In our network, 15 of the 10GbE ports are connected to the servers within the rack using low-cost short-reach 10G direct attach cables. 15 DWDM type 10 GbE SFP+ transceivers are plugged into the ToR switch. Their wavelength range is from 1553.33 nm to 1542.14 nm at ITU 100GHz channel spacing. Their transmitting and receiving signals are interconnected with other racks through optical switching units (OSU, Model Sodero PS03200). The OSU uses a broadcast-select approach: 1) combing the DWDM signals and broadcasting them via optical power splitter to the connected neighboring OSUs; 2) selecting the intended signals by setting the operating status of the wavelength selective switch (WSS). A small form factor EDFA is used to compensate the optical power loss and provide additional optical power budget. The WSS operating status and wavelength channel selections are controlled by a local microprocessor (Freescale i.MX53) controller board, which has its own IP address and is further remotely controlled by a central optical manager. The OSU has 8 optical ports available for inter-rack connections for different topologies. Fig. 1(b) shows a photo of the datacenter. Fig. 1 (c) shows a photo of a OSU connected to a ToR switch.

The central optical manager provides a set of powerful application programming interfaces (APIs) and tools, such as edge coloring for wavelength assignment and traffic monitoring at the end host, which are convenient for the higher-level controllers to directly interface with the optical layer. The manager can operate in two modes: active or passive. In the active mode, the optical manager monitors the data rate on each optical link, and adjust bandwidth allocation for each link based on the metric. In the passive mode, the optical manager receives wavelength assignment on each link from an external controller and simply implement the configuration.



(b) Photo of DFabric

(a) DFabric architecture, rack configuration and OSU structure. OSU: optical switching unit; ToR switch: top-of-rack switch; MUX: multiplexer, DeMUX: demultiplexer, WSS: wavelength selective switch; EDFA: Erbium-doped fiber amplifier. S1-S15: servers; C1-C15: 10G DWDM SFP+ transceivers

(c) Photo of OSU connected with ToR switch.

Fig.1 DFarbic:12-Rack 180 Server Datacenter network using multiwavelength optical switching.

3. Full Stack and Consistent Network Optimization for Low Latency DCNs

Tail latency is a key performance indicator for DCNs, and many traffic engineering (TE) work has been focusing on reducing the tail latency by avoiding transient congestion [4]. Traditional TE is performed on the network layer only, under fixed topology assumption. In DFabric, we are given the physical topology of the fibers and current traffic demand in the network, and perform joint optimization of the optical and network layers. We decide three important configuration parameters: (1) which pairs of ports of the switches are connected; (2) the capacity (number of wavelengths) on each link; and (3) flow routing to its destination. (1) and (2) are handled in the optical layer while (3) is handled in the network layer.

In DFabric, our optimization goal is to minimize the maximum single link utilization (defined as the percentage of the total capacity actually used for that link). In a DCN with fast changing traffic patterns, unpredictable incoming traffic burst can happen before our controller can take any action. Thus, it is necessary to leave enough "headroom" at even the busiest link.

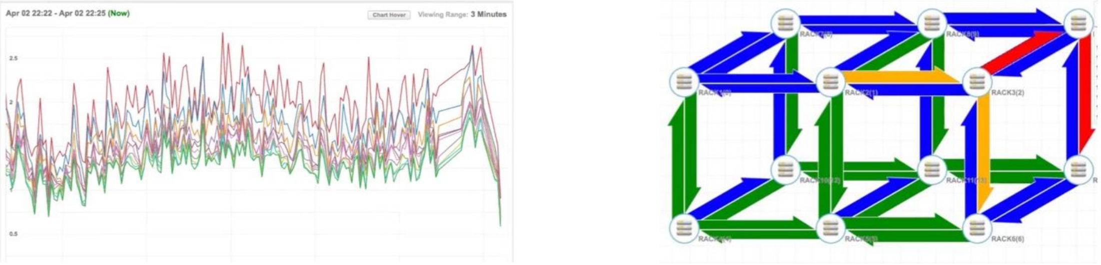
We designed a randomized approximation algorithm based on simulated annealing for the optimization problem. Specifically, when the controller runs, we randomly alter a single optical link to form a new topology, and compute the minimal max utilization in this new topology. If the result is better, the new state is accepted. Otherwise, it is accepted with probability according to an exponential function of the difference, to avoid being trapped in local minima. During each control period, we run a number of iterations of the algorithm, and choose the best one. There are two benefits in using this algorithm: (1) starting from the current topology, we are likely to achieve an incremental change affecting only a few links, reducing the overall update cost; (2) searching by topology instead of both optical and routing configurations significantly reduces the search space and makes the search viable.

After we compute the new topology and routing, we need to update the optical and ToR configurations to implement the topology and routing. The key problem during the transition period is how to find an update schedule that minimizes the disruptions to applications. For example, we have to eliminate loops, black holes, or congestion that may happen during the update period. We extend the state-of-the-art network update solution Dionysus [5] to incorporate topology updates into the operation dependency computation.

4. Testing Results

4.1 Traffic monitoring and visualization from the optical manager

Fig. 2 shows real-time monitoring and visualizations by the optical network manager, during a real Hadoop-based Terasort job with a 2TB dataset spanning all 12 racks. Fig. 2(a) shows the aggregated real-time network traffic monitoring over the entire network and (b) shows the per-link real-time utilization, where different colors are defined by the users to indicate the level of link utilization. When setting the manager to active mode, the optical network manager can automatically re-assign neighboring wavelength channels to the congested channels, which works well when a small percentage of the links are occasionally congested.



(a) Aggregated real-time network traffic data

(b) Real-time per-link utilization

Fig. 2 Real-time network monitoring and visualization interface from the optical manager.

4.2 Full stack optimization and consistent update

Under more sophisticated or congested network utilization, the optical network manager is set to be passive, and a full-stack optimizer proactively manages the network. We use three traffic patterns in DCN to evaluate the effectiveness of our algorithm. Pattern 1 is a cross-network bulk data transfer, i.e. half of the machines send data to the other side. This can happen during a storage backup or an offline indexer loading the new index onto the online searchers in a search engine. Pattern 2 has two separate traffic-intensive cliques, but the traffic is rare between the two cliques. This pattern usually represents two separate jobs, such as two independent large-scale Hadoop jobs. Pattern 3 contains all-to-all uniformly distributed traffic throughout the entire network -- we include it as a reference, since in real large scale data centers single application rarely expands to the entire DC.

For each pattern, we measure latency of ping packets on each link of an eight-rack subset. Fig. 3 shows the 99th percentile latency. We show that our optimization can reduce the tail latency to almost 1/5 of the original value. This is because there is less congestion after we avoid any link to be at high utilization. Of course, we observe degraded performance from optimization in Pattern 3, where all links carries are equally busy anyways.

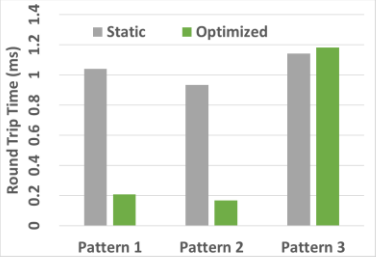


Fig. 3 99th percentile of round trip time

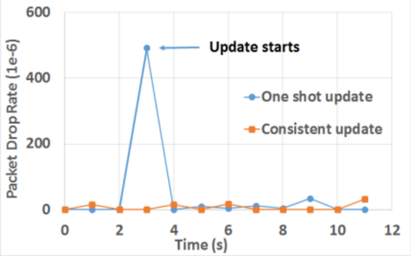


Fig.4 Consistent update vs. one shot update

We further evaluate the importance of the consistent update scheme. Fig. 4 shows packet drop rate changes on the affected links during an update. We compare the consistent update with a one-shot update scheme, where we move all affected flows onto a default link. We can see a quick yet high spike on packet drop rate with one shot update. The large drop triggers TCP back-off and it will take a long time for the network throughput to recover, and thus the consistent update scheme is necessary.

5. Conclusions

We built a 12-rack, 180-server DCN testbed and presented our full stack optimization algorithm including consistent network update, which jointly optimizes both the optical layer and the network layer to accommodate different workloads of different network applications.

References:

[1] K. Chen, A. Singla, A. Singh, L. Xu, Y. Zhang, “OSA: An Optical Switching Architecture for Data Center Networks with Unprecedented Flexibility”, Proc. of USENIX NSDI conference, April 2012.

[2] G. Wang, D. G. Andersen, M. Kaminsky, M. Kozuch, T. S. E. Ng, K. Papagiannaki, and M. Ryan, “c-Through: Part-time Optics in Data Centers", Proc. ACM SIGCOMM, Aug. 2010.

[3] N. Farrington, G. Porter, S. Radhakrishnan, H. Bazzaz, V. Subramanya, Y. Fainman, G. Papen, and A. Vahdat, “Helios: A Hybrid Electrical/Optical Switch Architecture for Modular Data Centers”, Proc. of ACM SIGCOMM, August 2010

[4] D. Zats, T. Das, P. Mohan, D. Borthakur, and R. Katz, “DeTail: reducing the flow completion time tail in datacenter networks”, Proc. of ACM SIGCOMM, August 2012.

[5] X. Jin, H. Liu, R. Gandhi, S. Kandula, R. Mahajan, M. Zhang, J. Rexford, R. Wattenhofer, “Dynamic scheduling of network updates." Proc. of ACM SIGCOMM, Aug 2014