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

Da Wei, Yiran Li, Wei Xu
Institute of Interdisciplinary Information Science (IIIS), Tsinghua University

Lei Xu
Torray Networks Inc. / Sodero Networks Inc

Xin Jin
Department of Computer Science, Princeton University
Hyper Converged Cloud => More Sophisticated DCNs

- Hyper converged infrastructure
- Different applications running over thousands of servers
- Workloads change fast
- Mix of short and long flows
- Diverse requirements of different applications
  - Search - Latency
  - Hadoop – Throughput
  - ...
- We need a FLEXIBLE network to cope with the challenges
Previous Work on Optical DCN

Early demonstrations of optically switched DCN testbed


Ever since, optical switching for intra- and inter- DCN applications has attracted strong interests in both academia and industry.
Long Tail Latency Issues in DCN

- Tail latency directly impacts the quality of service
- Long tail latency caused by congestions from
  - Traffic bursts
  - Uneven load balancing

Two orders of magnitude variations in RTT

DFabric DCN

- 12 racks, 180 servers
- WSS-based multiwavelength switching and interconnection (without central optical switching matrix)
- Hyper-cube topology
- OpenFlow enabled top-of-rack switches (ToR)
- Full stack controller and optimization
Optical Switching Unit (OSU) Design

Built from off-the-shelf components
Traffic Monitoring and Visualization

Controlled by the optical manager:

- Aggregated real-time network traffic
- Real-time per-link utilization

Components:
- Full Stack Controller
- Optical Manager
- OSUs
- ToRs
This demo is running Terasort program on a 165 nodes Hadoop cluster.
Full-stack optimization

- Balance load on links to avoid congestion
  - Optimization goal: minimize the maximum single link utilization
- Joint optimization of the optical and network layers
  - The problem is NP-hard
  - Randomized approximation algorithm based on simulated annealing
Key Algorithm Ideas

• Reduce search space using network-layer topology as the state
• Starting with topology that is similar to the current one
Consistent Update

- Problem: ensure no packet loss during update process
- Extend the state-of-the-art network update solution Dionysus\(^3\)
- Dionysus uses dependency graph to schedule update operations
- The dependency graph includes two types of nodes:
  - \textit{fNode} – Update operation that moves a flow from an old path to a new path
  - \textit{\(\lambda\)Node} – Update operation that moves a wavelength from an old edge to a new edge

Example of dependency graph

---

\(\lambda_1\) 
\(\lambda_2\) 
\(\lambda_3\)

\(f_1\) 
\(f_2\)

Results: Long Tail Latency Reduction

- Optimized topology vs. static topology
- Subset of 8 racks with three traffic patterns
  - Pattern 1: Cross-network bulk data transfer
  - Pattern 2: Two separate traffic intensive cliques, with limited traffic in between.
  - Pattern 3: All-to-all uniformly distributed traffic

99th percentile of round trip time
Results: Effective Consistent Update

- One shot update: move all affected flows onto a default link
- Congestion causes significant packet drop
- No significant change in consistent update

Consistent update vs. one shot update
Conclusion

• We present DFabric: a 12-rack, 180-server DCN using multiwavelength switching and interconnection.

• We implemented real-time network traffic and per-link utilization monitoring, full-stack optimization by jointly optimizing optical switching and network flow routing, and network status consistent update.

• We show benefits in long tail latency reduction and packet loss drop.