High Speed Load Balancing at the Datacenter Edge*

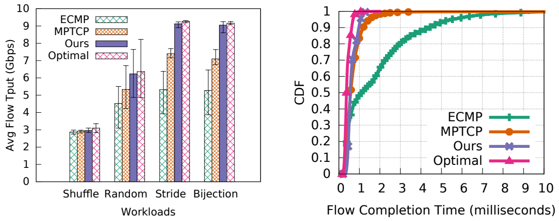
Keqiang He†, Eric Rozner‡, Kanak Agarwal‡, Wes Felter‡, John Carter‡, Aditya Akella†
†UW-Madison ‡IBM Research
{keqhe,akella}@cs.wisc.edu {erozner,kba,wmf,retrac}@us.ibm.com

1. INTRODUCTION

Datacenter networks must support an increasingly diverse set of workloads. Small latency-sensitive flows to support real-time applications such as RPCs share the network with large throughput-sensitive flows for big data analytics or VM migration. Load balancing the network is crucial to ensure operational efficiency and suitable application performance. Unfortunately, popular flow-hashing-based load balancing schemes, *e.g.*, ECMP, cause congestion when hash collisions occur and perform poorly in asymmetric topologies. A variety of load balancing schemes aim to address the problems of ECMP. Centralized schemes are reactive and very coarse-grained due to the large time constraints of their control loops or require extra network infrastructure [2]. Transport layer solutions such as MPTCP [3] can react faster but require widespread adoption and are difficult to enforce in multi-tenant datacenters. In-network reactive distributed load balancing schemes, *e.g.*, CONGA [1], can be effective but require specialized networking hardware.

2. OUR APPROACH

We piggyback on recent trends where several network functions are moving into hypervisors and software virtual switches on end-hosts and advocate to move network load balancing functionality out of the datacenter network hardware and into the software-based edge, *i.e.*, utilize vSwitches to break flows into discrete units of packets, called *droplets*, and distribute them evenly to near-optimally load balance the network. It uses the maximum TCP Segment Offload (TSO) size (64 KB) as droplet granularity, allowing for fine-grained load balancing at network speeds of 10+ Gbps. Reordering can cause TCP throughput degradation and impose significant computational burden by causing the Generic Receive Offload (GRO) handler to export many small segments. Our approach mitigates these problems by modifying GRO to ensure that large, in-order segments are pushed up to higher layers of the networking stack. Finally, our system load balances the network in the face of asymmetry and failures using a combination of fast failover and weighted multipathing at the network edge.



(a) (b)

Figure 1: System performance (a) elephant flow throughput and (b) mice flow (50KB) completion time.

3. EXPERIMENT RESULTS

We conducted experiments on a physical testbed consisting of 16 servers and a 2-tier Clos network with 8 10G switches. Then we compared our approach with ECMP, MPTCP and Optimal (*i.e.*, all the servers are connected to a non-blocking switch) using several workloads. Figure 1 shows that our load balancing scheme outperforms existing ones and closely track that of Optimal.

4. CONCLUSION

We present a near uniform sub-flow distributed load balancing scheme that can near optimally load balance the network at high networking speeds. Our scheme makes a few simple changes to the hypervisor soft-edge (vSwitch and GRO) and does not require any modifications to the transport layer or network hardware.

References

[1] M. Alizadeh, T. Edsall, S. Dharmapurikar, R. Vaidyanathan, K. Chu, A. Fingerhut, F. Matus, R. Pan, N. Yadav, G. Varghese, et al. CONGA: Distributed Congestion-aware Load Balancing for Datacenters. In *SIGCOMM*, 2014.

[2] J. Rasley, B. Stephens, C. Dixon, E. Rozner, W. Felter, K. Agarwal, J. Carter, and R. Fonseca. Planck: Millisecond-scale Monitoring and Control for Commodity Networks. In *SIGCOMM*, 2014.

[3] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley. Design, Implementation and Evaluation of Congestion Control for Multipath TCP. In *NSDI*, 2011.

*Keqiang He is a student author; without a demo