Scaling Data Center Switches Using Commodity Silicon and Optics

Nathan Farrington, Erik Rubow, and Amin Vahdat
UC San Diego
{farrington,erubow,vahdat}@cs.ucsd.edu

1. INTRODUCTION

With the help of parallel computing frameworks such as MapReduce, organizations routinely process petabytes of data on compute clusters containing thousands of nodes. For these massively parallel workloads, the principal bottleneck can often be the rate at which nodes can exchange data over the network. Unfortunately, modern DCN architectures typically do not scale beyond a certain amount of bisection bandwidth and become prohibitively expensive well in advance of reaching their maximum capacity. There is interest in replacing these expensive packet switches with many smaller, commodity switches, organized into a fat-tree topology [1], or direct network topologies [2, 3]. But as the number of packet switches grows, so does the cabling complexity, the management overhead, and the difficulty of actually constructing the network.

Our goal is to design a multi-stage switch architecture leveraging merchant silicon to reduce the cost, power consumption, and cabling complexity of DCNs, all while economically delivering high bisection bandwidth. In essence, we repackage a fat tree of discrete packet switches into a single distributed multi-stage switch, while also eliminating redundant components to save cost and power. Our switch design scales to 3,456 ports of 10 Gigabit Ethernet (10GbE) with 34.56 Tb/s of bisection bandwidth. The combination of custom packaging and our own Ethernet Extension Protocol (EEP) reduces the number of inter-switch cables from 6,912 to just 96. When 64-port 10GbE switch silicon becomes available, our techniques should generalize to 64,000 ports.

2. MERCHANT SILICON

The last decade has seen the introduction of merchant silicon: networking ASICs produced for the network equipment mass market. Several companies have used these ASICs to create commodity 24-port switches. By lowering their costs, they can offer switches at a lower cost to consumers, which was the original motivation for constructing an entire data center network out of commodity switches [1].

3. DESIGN OF A 3,456-PORT SWITCH

Our 3,456-port 10GbE switch is comprised almost entirely of merchant silicon, connected in a fat-tree topology. In contrast, large Ethernet switches from traditional network equipment makers use multiple proprietary ASICs and a crossbar topology. Rather than build one monolithic switch, we separate the design into 24 satellite switches and a core switch array. A satellite switch is basically the bottom two tiers of the fat tree; the core switch array forms the top tier.

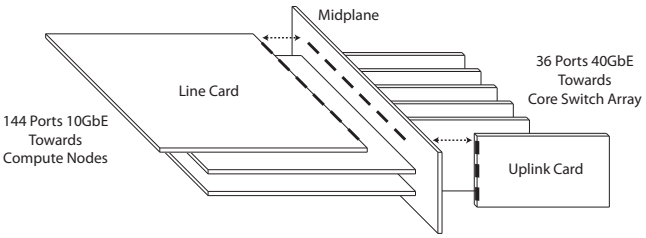

Figure 1: Satellite switch packaging.

Each satellite switch can function as a standalone 144-port 10GbE switch. But when connected to the core switch array, the satellite switches act as a single non-blocking switch. The satellite switches can also be incrementally deployed as the network is built out. When fully deployed, the switch has 3,456 ports and 34.56 Tb/s of bisection bandwidth. Each satellite switch connects to the core switch array with four parallel cables, each cable carrying 72 multimode fibers. These cables can be routed in an overhead cable tray.

The core switch array is not a switch; it is a collection of 144 individual 24-port switches. We separate the array into two modules to provide fault tolerance and to allow a limited form of incremental deployment.

4. SATELLITE SWITCH PACKAGING

The satellite switch is constructed from multiple circuit boards. At the center of the switch chassis is a single midplane circuit board. The midplane provides three important functions. First, it connects the line cards to the uplink cards through high-density high-speed electrical connectors. Second, it provides power to all of the line cards and uplink cards. Third, it contains a CPU which manages the state of the switch.

Each line card essentially replaces four discrete 24-port switches from the edge layer of the network and eliminates redundant components. The four switch ASICs separate the board into two halves. The bottom half of the board contains 48 SFP+ optical tranceiver cages and 48 PHY chips. The top half of the board contains an additional 48 PHYs and 6 electrical connectors. These PHYs convert between XAUI and 10GBASE-KR, the IEEE standard for 10GbE over backplanes.

The uplink card performs two functions. First, it acts as a switch fabric for the satellite switch, connecting together the 12 switch ASICs on the 3 line cards. Second, it forwards

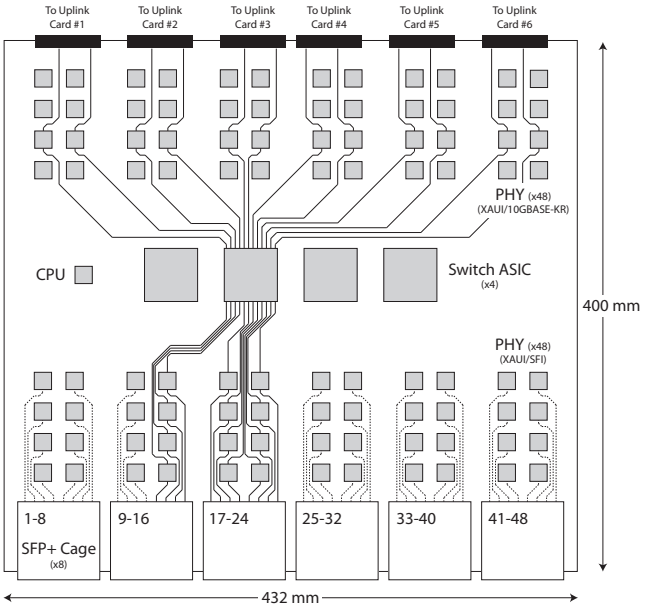

Figure 2: Line card layout.

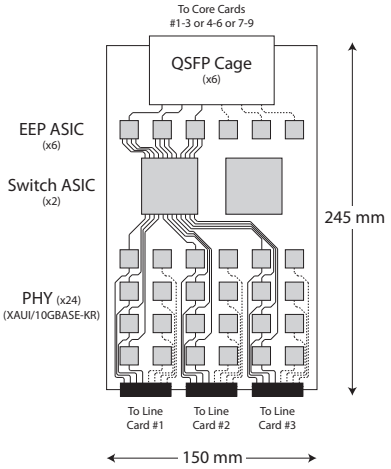

Figure 3: Uplink card layout.

traffic to and from the core switch array.

Each uplink card replaces two discrete 24-port switches from the aggregation layer of the network. The bottom half connects to the midplane. The 8 traces from each line card are routed to 8 PHYs, and then split between the two switch ASICs. The top half of the board contains six EEP ASICs and six QSFP cages. Each EEP ASIC connects to four ports of the switch ASIC using the XAUI protocol and connects to one of the QSFP modules using the QSFP electrical interface.

The top half of the board contains six EEP ASICs and six QSFP cages. Each EEP ASIC connects to four ports of the switch ASIC using the XAUI protocol and connects to one of the QSFP modules using the QSFP electrical interface. Esentially, the top half of the board is aggregating the 24 ports of 10GbE into 6 ports running the custom EEP protocol at 40 Gb/s.

The core switch array (not shown) contains 18 independent core switch array cards, partitioned into two modules with 9 cards each. The choice of core switch array packaging is largely arbitrary since the 144 separate 24-port switches do not communicate with each other directly. The cards do not connect to a backplane and their co-location is merely a matter of simplifying cable management and packaging.

5. ETHERNET EXTENSION PROTOCOL

The EEP ASIC is an Etheret traffic groomer, a device that aggregates frames from multiple low-rate links and tunnels them over a single higher-rate link. Our rationale for EEP comes from the properties of optical transceivers. From Table 1, it is actually less expensive in terms of cost, power consumption, and physical area to aggregate to higher-rate links, and then to deaggregate back to the original rate. Our custom protocol is lightweight and requires no configuration. We implemented EEP using a Xilinx FPGA.

EEP provides the abstraction of a set of 16 virtual Ethernet links, multiplexed over a single physical link. EEP breaks up an Ethernet frame into a sequence of 64B segments, and encapsulates each segment into an EEP frame. Each EEP frame originating from the same port is assigned the same virtual link ID.

	SFP	SFP+	QSFP
Rate	1 Gb/s	10 Gb/s	40 Gb/s
Cost/Gb/s	$35	$25	$15
Power/Gb/s	500mW	150mW	60mW

Table 1: Comparison of optical transciever modules.

The very first EEP frame in the sequence has the SOF (Start of Frame) bit set. All other EEP frames in the sequence have the SOF bit cleared. The last EEP frame has the EOF (End of Frame) bit set. The LEN bit indicates that the second byte in the EEP frame is part of the header rather than the payload. This second header byte records the number of valid bytes in the payload in the case where the value is not 64. Most EEP frames will use one header byte.

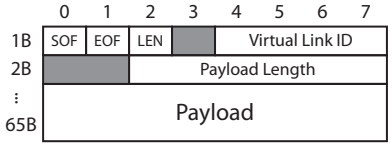

Figure 4: EEP frame format.

6. REFERENCES

[1] M. Al-Fares, A. Loukissas, and A. Vahdat. A Scalable, Commodity, Data Center Network Architecture. In *SIGCOMM*, 2008.

[2] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu. BCube: A High Performance, Server-centric Network Architecture for Modular Data Centers. In *SIGCOMM*, 2009.

[3] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu. DCell: A Scalable and Fault-Tolerant Network Structure for Data Centers. In *SIGCOMM*, 2008.