Your Data Center Is a Router: The Case for Reconfigurable Optical Circuit Switched Paths

Guohui Wang*, David G. Andersen†, Michael Kaminsky‡, Michael Kozuch‡,
T. S. Eugene Ng*, Konstantina Papagiannaki‡, Madeleine Glick‡, Lily Mummert‡
**Rice University,* †*Carnegie Mellon University,* ‡*Intel Labs Pittsburgh*

1 Introduction

The rising tide of data-intensive, massive scale cluster computing is creating new challenges for traditional, hierarchical data center networks. In response to this challenge, the research community has begun exploring novel interconnect topologies to provide high bisection bandwidth—examples include Fat trees [2, 12, 8], DCell [9], and BCube [10], among a rapidly growing set of alternatives, many adapted from earlier solutions from the telecom and supercomputing areas.

We argue that these solutions may provide too much—full bisection bandwidth on packet timescales—at too high a cost—literally tons of wiring and thousands of switches. In this work, we suggest that research should take a look back not only at historical *topologies*, but also historical *technologies*. More specifically, we suggest that a hybrid packet-switched/circuit-switched network can provide the functions and ease-of-use of today's all-packet networks, while providing high bandwidth for a large class of applications at lower cost and lower network complexity. Taking advantage of this network requires, however, a philosophical change to the design of data center networks. We propose to augment the electrical switch architecture with an optical circuit-switched network. Implementing this approach requires a network re-design to provide substantial pre-optical queueing at the nodes, treating the entire data center as one large virtually output-queued router. We explain this argument briefly, and expand upon our proposed solution in the sections that follow.

Today's data center networks typically place 10–40 servers in a rack, with an aggregation ("Top of Rack", or ToR) switch in each. The ToR switches are the leaves in a tree of Ethernet switches that connects all of the racks. The bandwidth at the top of the tree is typically a fraction of the incoming capacity, creating a large bottleneck.

Recently proposed architectures for electrical networks are based on variants of full-bisection networks such as fat-trees, butterflies, or variants of hypercubes. These new network designs, however, often require a large number of node-to-node links and switches. For example, if a *k*-level fat tree is used to connect N servers, at least $N \times k$ wires are needed to construct the network. Additionally, to guarantee full bisection bandwidth, these designs require complex, structured wiring, which makes them hard to construct physically and scale to accommodate growth. This construction and management complexity could limit their adoption.

Augmentation with an optical network provides a simpler solution. Ignoring for the moment practical questions of implementation, the strengths of optical switching complement those of electrical switching. Modern optical circuit switching technologies are, modulo the introduction of a small amount of noise, rate-free: they switch whatever rate is modulated at the ends. They are therefore comparatively cheap for high-bandwidth (40+Gbps) connections. However, all-optical packet switching has been an elusive goal for decades, and, we suspect, will remain so for some time. Electronics are better suited for making per-packet forwarding decisions, but electrical switching is comparatively slow, expensive, and power-hungry. In theory, at least, the combination of circuit switched optical (affordably high-bandwidth, but coarse switching granularity) and electrical switching (affordably fine-grained switching, expensive bandwidth) might offer the best of both worlds. But why should one believe that combining them is possible or even practical?

Data center workloads The characteristics of many data center workloads may render full bisection bandwidth at the *packet granularity* excessive. There are three reasons for this: non-network bottlenecks, biased traffic distributions, and amenability to batching. 1) Applications that hit CPU, disk I/O, or synchronization bottlenecks on some computers will not saturate the network link, and many applications will not saturate the network all of the time—some data center traffic follows a strong ON-OFF pattern [4]. 2) Other applications, such as many scientific applications, have skewed communication patterns where most nodes only communicate with a small number of partners [3, 11]. These patterns do not require arbitrary full-bisection capacity. 3) A final set of applications, such as large MapReduce-style computations, may be amenable to batched data delivery: instead of sending data to destinations in a fine-grained manner (e.g., 1, 2, 3, 1, 2, 3), sufficient buffering can be provided to batch this delivery (1, 1, 2, 2, 3, 3).

Design outline. Our basic design is to use a circuit switched optical network to connect top-of-rack switches with a high-bandwidth link, and to also connect them through a relatively low-bandwidth, traditional electrical switch hierarchy. This design limits the required optical port count, but means that high bandwidth becomes available from one group of machines to another group of machines in a circuit time-slot. We consider MEMS switches that can reconfigure on a timescale of several to tens of milliseconds. Recent trends in the pricing of optical MEMS switches with these characteristics confirm the viability of such a design option for deployment in data centers.

However, because the optical network is circuit switched, it can provide only one optical path for each server rack. For example, once an optical path is set up between racks *A* and *B*, the packets from *A* to *B* can be delivered at very high speed. The packets from rack *A* to other racks, however, cannot go through the optical network until the switch is reconfigured. Effective use of the optical network requires continuous monitoring of the rack-to-rack traffic demands and consequent reconfiguration of the network.

In this paper, we analyze the feasibility of using circuit-switched optical paths in data centers. We show that, even in large scale data centers, the optical paths can still be reconfigured at small time scales. Using the workloads of an existing data center, we show that a small number of optical paths has great potential to significantly relieve today's data center bottlenecks. Although this design holds great promise, it raises several substantial questions. How should the circuit-switched optical network be integrated into today's data centers with low software and hardware cost? How should a circuit-switched optical network be managed to provide maximum benefit in the presence of changing traffic patterns? What are the implications to data center applications, and for what classes of applications does this design work (or not)? We explore potential solutions to address these questions. We discuss the system elements and outline a first-cut system architecture to manage the reconfigurable circuit-switched optical paths.

2 Related Work

In addition to the aforementioned recent new data center topologies (FatTrees, DCell, BCube, etc.), the most related work is the use of optical circuit switching in supercomputer design. These efforts differ substantially from our focus. The UCLP (User Controlled Lightpath) project [1], for instance, explored the use of optical circuit switches on hour-and-longer timescales for wide-area grid computing. Researchers from IBM and others have examined the use of hybrid electrical/optical systems inside supercomputers, but focused on the lower-level hardware issues and provided node-to-node connectivity, instead of our focus on affordable switch-to-switch connectivity [3, 11].

The idea of adding reconfigurable optical paths for high bandwidth data centers has been previously proposed in [7]. The use of hybrid networks in a stream computing system has been further explored in [13]. Although no design details are provided [13], this work has mostly focused on the routing and job management of the stream computing application. The focus of our work is on integrating optical circuits in large scale data centers and understanding the impact of the hybrid network architecture on different applications. We believe that our view of treating the data center as a single virtually output queued router is novel among this prior work.

3 Optical Circuits for Data Centers

Optical Technologies. Optical links are today's standard for ultra-high speed data transmission. Telecommunications and wide-area backbone networks commonly use 40Gbps OC-768 lines, and new 100Gbps optical links have already been developed. With the increasing demand for high speed transmission in data centers and storage area networks, optical fiber is a logical choice because of its low loss, ultra-high bandwidth, and low power consumption. Traditionally, however, optical components (e.g., transceivers and switches) have been significantly more expensive than their electrical counterparts. However, recent advances in optical interconnect technology have precipitated cost reductions that might make using optical links in data centers viable.

Our design leverages MEMS-based optical switches. These devices, which offer a promising cost-performance point, provide switching by physically rotating mirror arrays that redirect carrier laser beams to create connections between input and output ports. Once such a connection is established, network link efficiency is extremely high; however, the *reconfiguration time* for such devices is long (a few milliseconds)[1].

Consequently, they act as *circuit-switching* devices.

Hybrid Packet-Switched / Circuit-Switched Network Design. Figure 1 depicts our proposed network architecture. Because the optical network is circuit-switched,

[1] e.g., Opneti's 1x8 MEMS switch requires 2ms typical, 5ms max to switch with multi-mode fiber. http://www.opneti.com/right/1x8switch.htm

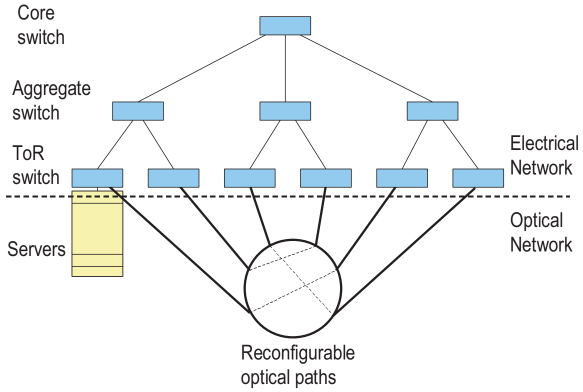

Figure 1: Network Architecture

our design includes a simple, packet-switched electrical network to provide low-latency, all-to-all connectivity. Thus, the optical network can offload the high-volume traffic that would otherwise bottleneck the electrical network, but applications are not blocked waiting for the optical network to reconfigure. The electrical paths are always available in the network, allowing the applications to communicate immediately (albeit at a lower data rate) to any server in the data center.

The electrical network (top) uses a traditional hierarchy of Ethernet switches arranged in a tree. The optical network (bottom), however, connects only the top-of-rack switches. This significantly reduces the number of additional optical interfaces, links, and switches required, but still provides the desired bandwidth improvement because a single optical switched path can handle tens of servers sending at full capacity over conventional Gigabit Ethernet links.

Potential Advantages. This design offers potential improvements in manageability, cost, and flexibility.

Manageability. Compared with a full bisection bandwidth network structure, a small optical network is much easier to construct and manage. There are no rigid topological constraints, and no need for major re-wiring to increase the data center network size. It is easier to scale up and can be incrementally deployed.

Cost. Even at today's prices, the cost of optical networking components is comparable with existing solutions. For example, it has been estimated that constructing a BCube with 2048 servers costs around $92k for switches and NICs and requires 8192 wires [10]. Today, MEMS switches are mostly aimed at the low-volume, high-margin test and measurement market, but even so, using a MEMS switch to connect 52 48-port switches would cost only approximately $110k. (On an 80-port MEMS optical switch, each port costs $200-$700, single 10Gbit optical transceiver modules cost under $350, and 48-port switches cost under $700.) We expect the cost of these switches would drop substantially were they used in commodity settings, and the cost of the transceivers drops continuously. Moreover, the optically augmented design requires only 52 fibers to interconnect over two thousand machines, compared to the thousands of wires required by full-bisection approaches.

Flexibility. The structure of the optical network is highly flexible. Depending on the data center's workload requirements, different levels of circuit capacities can be provisioned. The optical network can be constructed by a single optical switch, or through the connection of multiple switches. If fewer circuit switched paths are simultaneously required, we can use fewer ports on switches to construct the optical network. Expanding an existing network to add additional switching capacity is also relatively easy.

4 Feasibility Analysis

The feasibility of our proposal depends on the agility of optical path reconfiguration and the existence (or potential to induce) traffic skew.

4.1 Optical Reconfiguration Algorithm

Suppose the cross-rack traffic matrix is given (we discuss later how to estimate it), we need to figure out how to connect the server racks by optical paths in order to maximize the amount of traffic offloaded to the optical network. This can be formulated as a maximum weight perfect matching problem. The cross rack traffic matrix is a graph $G = (E, V)$. V is the vertex set in which each vertex represents one rack and E is the edge set. The weight of an edge e, $w(e)$, is the traffic volume between the end vertices. A *matching* M in G is a set of pairwise non-adjacent edges. That is, no two edges share a common vertex. A *perfect matching* is a matching that matches all vertices of the graph. From this formulation, the optical configuration is a perfect matching with the maximum aggregated weight. The solution can be computed in polynomial time by Edmonds' algorithm [6].

4.2 Optical Path Reconfiguration Time

The ability of the optical network to relieve bottlenecks in the electrical network will depend on its agility to reconfigure and accommodate varying traffic demands. There are several factors that influence how often one could reconfigure the optical network in our approach. First, there is the circuit setup signaling delay, which, for data center networks, should be small ($< 1ms$). Second, the setup of optical paths implies the physical manipulation of mirrors that have specific hardware switching times. During this fixed period, the optical paths are down. For MEMS-based optical switches, the hardware reconfiguration time is a few milliseconds. Third, the reconfiguration interval is also lower bounded by the time

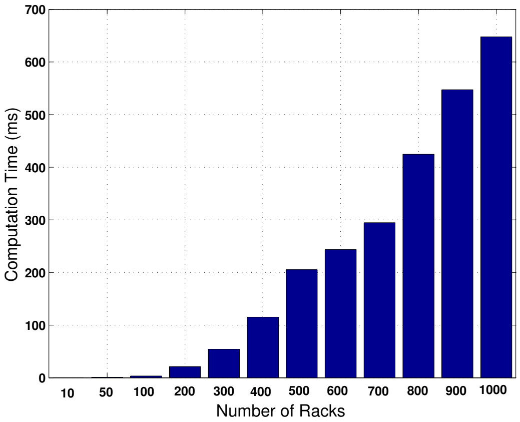

Figure 2: Optical Path Computation Time

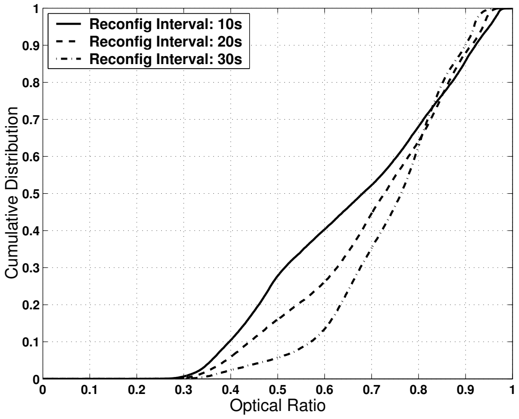

Figure 3: The Cumulative Distribution of Optical Ratios for One-week Data Center Traffic

required to compute the rack-to-rack optical path configuration, which depends on the network size.

We use the Blossom VI implementation [5] of Edmonds' algorithm to compute the optical path configuration. Figure 2 shows the configuration computation time for random rack matrices with different numbers of racks. For each rack matrix size, we generate 5 random matrices and present the average computation time versus rack size in Figure 2. The computation runs on one core of an Intel Xeon 3.2GHz processor with 2GB of memory. The results show that the optimal configuration can be computed rapidly—around 640 ms for 1000 racks. Consequently, even in very large data centers, one could envision the reconfiguration of the optical network happening at relatively small time scales, thus able to accommodate varying traffic demands.

4.3 Evidence for Traffic Skew

Augmenting a traditional data center architecture with a reconfigurable circuit-switched optical network introduces additional capacity that may relieve some of its inherent bottlenecks. However, this holds true only if the rack-to-rack traffic is, or can be made, skewed, and if this skew can be identified and exploited appropriately. We study the workload of one small operational data center to assess the properties of today's data center workloads (even though we admit that our results could not generalize across all possible data centers).

Our analysis is based on a small seven-rack research datacenter with a total of 155 servers and 1060 cores. We instrument all servers with the IPTables NetFlow module and export data every 10 seconds. We then aggregate this server-to-server traffic matrix into a rack-to-rack traffic matrix based on the datacenter topology. The traffic captured includes a variety of workloads, such as MPI, Hadoop, and scientific computing applications.

We input the rack-to-rack traffic matrix to Edmonds' algorithm for the perfect weighted matching that will identify the 3 rack-to-rack flows that can be routed on top of the optical network (the 7 racks allow for 3 non-overlapping rack-to-rack connections). The sum of the volume of those 3 flows represents the maximum amount of traffic that could be offloaded from the electrical network onto the optical paths. We analyze a one-week traffic trace using three different optical path reconfiguration intervals. We define the optical ratio to capture the total traffic volume of those 3 rack-to-rack optical flows over the overall cross-rack traffic in the data center. We then plot its empirical cumulative distribution function in Figure 3. Note that if rack-to-rack traffic is uniform, the fraction of offloaded traffic would be $3/21 = 14\%$ (out of the 21 rack-to-rack flows, only 3 can be routed optically).

Instead, Figure 3 shows that setting up 3 optical paths between the 7 racks in the data center can offload more than 50% of the total cross rack traffic in most cases. Reconfiguring the optical network every 30 seconds results in higher fractions of the overall traffic routed optically, taking advantage of increased aggregation. This result demonstrates that using even a few optical paths has the potential to offload significant amounts of traffic from the electrical network. Further study is needed to derive the best optical reconfiguration time based on the dynamic traffic demands.

5 Managing Optical Paths

The key challenge imposed by our proposed architecture is that the circuit switched component requires explicit set-up and tear-down of links, and is only available periodically. In this section, we outline a first-cut system architecture to *monitor* traffic patterns, *induce* traffic skew on a rack-to-rack basis, and *reconfigure* the optical network to optimize the transmission of this skewed traffic.

5.1 Design Elements

Traffic Measurement: The system must estimate the rack-to-rack traffic to input to the optical path reconfiguration algorithm in the previous section. This information could be exported from the switches themselves, but in keeping with our philosophy that the datacenter is a router, it is most easily and scalably monitored at the end-hosts using the same NetFlow infrastructure we used for the feasibility analysis. An OS-independent approach would be to collect this data from the end-host NICs, but we leave such an evaluation for future work—certainly, at least, exporting flow data is a less intrusive modification than actually routing data through end-hosts, as has been proposed for other architectures [9, 10].

Inducing Traffic Skew: The core of skew induction is to provide extensive buffering at the edge nodes, and to manage these queues as part of a virtually output-queued router. Each node, therefore, will maintain several hundred megabytes of queued data that is aggregated on a destination rack basis. We discuss this aspect further in Section 5.2.

Optical Configuration Manager: The optical configuration manager collects traffic measurements, runs the optical path algorithm and issues configuration directives to the switches, and informs hosts which paths have become high-bandwidth. We envision an initial implementation of this component as a small, central management component attached to the optical switch (equivalent to a router control plane) running the fast Blossom VI implementation, but a distributed implementation is an intriguing future approach.

Optional Host Traffic Controller: This optional component runs on the hosts to update applications and the kernel about the inter-node connectivity to permit applications to batch and transmit data appropriately. For many applications, this functionality may be accomplished sufficiently by TCP's AIMD algorithm and may be omitted, providing transparent acceleration for a wide range of applications. Yet other applications, such as a client requesting data from many file servers, may benefit from being able to tailor their requests and computation to the state of the network. Understanding this design space is one of the key challenges and opportunities exposed by the proposed architecture.

5.2 Optical Management System Sketch

Our design goal is to implement traffic measurement and control in-kernel to make the system transparent to applications and easy to deploy. Figure 4 shows the architecture of the optical path management system.

The server kernel scheduling module manages the queues to ensure that data can be sent at high speed to a remote rack when the optical link is available. It operates by greatly enlarging the TCP socket buffers, allowing applications to push hundreds or more megabytes of data into the kernel. We buffer at the end hosts, not the switches, so that the system scales well with increasing node count; to take advantage of the relatively cheap server DRAM; and to ensure fate sharing with the data source. This design introduces the possibility of head-of-line blocking, however, where large bulk flows waiting for the optical paths may greatly increase the latency observed by small control flows sharing the same destination-rack buffer. To address this problem, the system will use a mechanism similar to router virtual output queues. The scheduler maintains one virtual queue for the packets destined to each rack. When an optical path is set up for a rack A, the scheduler can decide when to release a batch of packets from rack A's virtual queue to the optical path. Different policies can be used to control the traffic transmitted to electrical paths and optical paths. There are several interesting questions (Section 6) about the ways traffic maps to these paths.

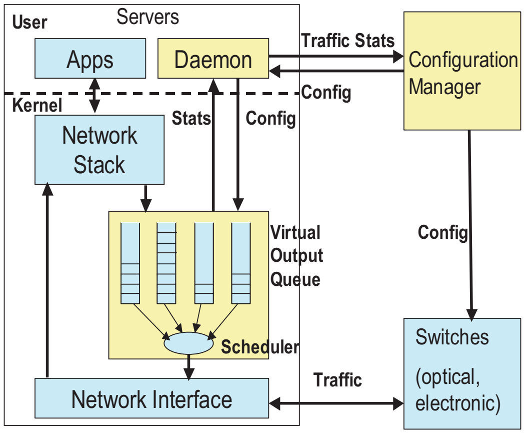

Figure 4: Optical Management System Architecture

The user-space server management daemon reads traffic statistics from the server's scheduling module and reports them to the central manager. The traffic statistics report how much traffic is buffered to each destination rack, permitting the optical configuration manager to perform an optimal configuration of the lightpaths.

6 Challenges

Our preliminary study suggests that using reconfigurable optical paths has great potential to provide high bandwidth for data-intensive applications in data centers. However, this approach brings with it a number of challenging issues for the architecture itself, the network, and for the applications, that must be addressed before it truly becomes practical.

By allowing applications to push large amounts of data into the in-kernel per-rack queues, this architecture creates the possibility for severe latency increases and head-of-line blocking. These large queues will eventually drain through the electrical network, so applications will not be *stopped*, but data packets may be delayed for hundreds of milliseconds or longer—a delay unacceptable for applications such as search or other user-interactive applications. At a high level, we believe the way to address this is through more-or-less conventional QoS approaches, but *which* of these approaches is an open question. Application/library-assisted tagging of low-latency (or bulk) data provides a simple mechanism, but requires source code changes; port number based inference is also simple, but fragile; automatic prioritization of low-volume flows seems more robust, but may not always provide the right answer (and requires more heavy-weight queueing mechanisms). While more study is clearly needed, we suspect that good solutions can be found for a wide variety of datacenter applications.

A second challenge is packet re-ordering and burst losses during optical network reconfiguration, and different flows experiencing different rates over time if some form of QoS is enabled. While intuition suggests that most applications do not depend on implicit arrival synchronization across flows, the variety of assumptions made by programmers in contravention of "common sense" can be staggering—this issue deserves study beyond a glib handwaving-off. Whether or not it is worthwhile to use controller-to-host signalling to prevent this reordering and loss is an interesting empirical question for future work.

The third, and most important, question is whether applications benefit from the periodically available high capacity links, and how much application modification is needed to allow them to. Two extreme cases illustrate the challenges here—completely elastic point-to-point FTP-like transfers will clearly run faster; completely inelastic, scientific applications with frequent barrier synchronizations will be unimproved. While we speculated in the previous section about a buffering-based design that would accommodate loosely-synchronized applications with large data transfer requirements, such as MapReduce applications, this issue begs an answer—and, eventually, a useful taxonomy of application communication patterns to allow datacenter architects and application programmers to understand what types of connectivity are needed for their particular applications.

7 Conclusion

This paper argues that network researchers should cast off their packet-centric yoke in search of designs that provide cost-effective capacity for datacenter networks. Our approach augments a traditional (limited bisection bandwidth) hierarchical Ethernet with a periodically reconfigurable circuit switched optical network between top-of-rack switches. The success of this approach depends on the ability to rapidly compute a good set of ToR switches to interconnect, which we show in Section 4.1 is feasible. It depends on the existence of traffic skew, already present in many workloads, or the ability to induce traffic skew through extensive buffering, a design for which we sketch in Section 3. While substantial challenges lie on the path to a full realization of this architecture, we believe it shows remarkable promise and applicability to a variety of datacenter applications.

Acknowledgments

We would like to thank Richard Gass and Michael Ryan, of Intel Labs Pittsburgh, for their help with the setup of the experiments. This research was supported in part by NSF awards 0546551 and 0619525. Wang and Ng are sponsored by the NSF through CAREER Award CNS-0448546, CNS-0721990, and by an Alfred P. Sloan Research Fellowship.

References

[1] UCLP Project. http://www.uclp.ca/.

[2] M. Al-Fares, A. Loukissas, and A. Vahdat. A scalable, commodity, data center network architecture. In *Proc. ACM SIGCOMM*, Seattle, WA, Aug. 2008.

[3] K. Barker, A. Benner, and R. H. et al. On the feasibility of optical circuit switching for high performance computing systems. In *Proc. SC05*, 2005.

[4] T. A. Benson, A. Anand, A. Akella, and M. Zhang. Understanding data center traffic characteristics. In *Proc. Workshop: Research on Enterprise Networking*, Barcelona, Spain, Aug. 2009.

[5] W. Cook and A. Rohe. Computing minimum-weight perfect matchings. *INFORMS Journal of Computing*, 11:138–148, 1999.

[6] J. Edmonds. Paths, trees and flowers. *Canadian Journal on Mathematics*, pages 449–467, 1965.

[7] M. Glick, D. G. Andersen, M. Kaminsky, and L. Mummert. Dynamically reconfigurable optical links for high-bandwidth data center networks. In *Optical Fiber Comm. Conference (OFC)*, Mar. 2009.

[8] A. Greenberg, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. Maltz, P. Patel, and S. Sengupta. VL2: A scalable and flexible data center network. In *Proc. ACM SIGCOMM*, Barcelona, Spain, Aug. 2009.

[9] 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 *Proc. ACM SIGCOMM*, Seattle, WA, Aug. 2008.

[10] 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 *Proc. ACM SIGCOMM*, Barcelona, Spain, Aug. 2009.

[11] S. Kamil, D. Gunter, M. Lijewski, L. Oliker, and J. Shalf. Reconfigurable hybrid interconnection for static and dynamic scientific applications. In *Proc. Conference on Computing Frontiers*, 2007.

[12] R. N. Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat. Portland: A scalable fault-tolerant layer2 data center network fabric. In *Proc. ACM SIGCOMM*, Barcelona, Spain, Aug. 2009.

[13] L. Schares, X. Zhang, R. Wagle, D. Rajan, P. Selo, S. P. Chang, J. Giles, K. H. adn D. Kuchta, J. Wolf, and E. Schenfeld. A reconfigurable interconnect fabric with optical cicuit switch and software optimizer for stream computing systems. In *Optical Fiber Comm. Conference (OFC)*, Mar. 2009.