Accelerating communication-intensive parallel workloads using commodity optical switches and a software-configurable control stack

Diego Lugones[1], Konstantinos Christodoulopoulos[2], Kostas Katrinis[3], Marco Ruffini[2], Donal O'Mahony[2], and Martin Collier[1]

[1] The Rince Institute, Dublin City University,
diego@rince.ie
[2] School of Computer Science and Statistics, Trinity College Dublin, Ireland
christok@tcd.ie
[3] IBM Research - Ireland
katrinisk@ie.ibm.com

Abstract. In response to the need for faster and fatter networks for large-scale HPC cluster systems, hybrid optical/electrical networks have been proposed as an affordable and high-capacity solution. Still, there is no prior work evaluating the performance of HPC workloads over such types of networks. To fill this gap, this work presents a hybrid network architecture comprising commodity-only equipment, shows its price competitiveness against fat-tree alternatives and presents a prototype implementation. We evaluated several HPC workloads over our prototype, showing that our hybrid optical/electrical network manages to significantly accelerate tested workloads, without incurring any extra cost compared to an all-electronic fat-tree network.

Keywords: high-performance computing, high-speed networks, interconnects, distributed systems

1 Introduction

The increasing compute density in modern High-Performance Computing (HPC) system as a result of higher-core integration and the use of specialized accelerators is among others pushing the need for high-speed networks that are faster and with higher capacity across all levels of their hierarchies. The latter requirement, especially when seen at large-scale, leads to massive capital and management costs, magnifying the contribution of the network to total system cost. In response to this, prior research has proposed using commercial off-the-shelf optical switches for aggregating traffic between racks, partly or entirely replacing the multiple hierarchies of electronic networks and leveraging on the interesting features exhibited by such devices, such as lower cost/port and immense rate/port capability. However, very little is still known about the impact that hybrid optical/electrical networks have to HPC applications' performance and at what cost.

To address this challenge, we lay out in this paper the architecture of such a system at scale using commodity-only single-vendor equipment, calculate its total price using current list prices and compare it against the total investment for a conventional fat-tree at various capacity levels, showing that our hybrid solution is more affordable, up to 31%. We then present a fully-functional research prototype of our system architecture, featuring among others a) a network controller that is capable of accepting workload communication pattern input and re-configuring the network in a manner that optimizes application execution and b) an end-system shim-layer to allow compute servers to route over our network without modifications to running applications or the operating system. The controller implements optimization heuristics presented in our previous work [1] and exposes a high-level programming interface for trying out further topology optimization algorithms.

We deployed our software stack in a 40-servers/4-rack testbed in our lab and used our experimental setup to compare the performance of communication-intensive HPC kernels and pseudo-applications running over our architecture against the performance obtained over equal-cost fat-tree setups. Our results manifest that - at equal cost to fat-trees - our hybrid network system implementation manages to accelerate the workloads tested, yielding up to 8x speedup in 20-rack experiments and up to 50% in the largest configuration deployment.

This paper is structured as follows. Section 2 puts past related research in the context of our work. Section 3 presents our system architecture and its price competitiveness against fat-trees. We outline in Section 4 the main components of our system prototype that we used throughout our experimentation to obtain the results reported in Section 5. Section 6 summarizes the findings and contributions of this work and outlines future work in this field.

2 Related Work

Various hybrid interconnects have been proposed for high-performance clusters [2] and datacenter architectures [3] [4] [5]. The basic differences among these proposals can be found at the network level in which the optical network is connected, also in the number and rate of the optical ports per connection point, and with regard to the use of single vs. multi-hop connections over the optical network. In Helios [3], the optical network interconnects *pods* (i.e. set of several racks connecting up to 1000 servers), while in c-Through [4] and OSA [5] the basic block is the rack. Both [3] and [4] report only single-hop transmissions over the optical network, while [5] considers multi-hop connectivity, however, without including this feature as part of its topology re-configuration heuristic. It must be noted that including multi-hop connections in the optimization makes the topology computation quickly unaffordable for large systems and the implementation of the control software more challenging.

Beyond the architectural features and algorithmic approaches that we innovate on, this paper deviates from related work in the perspective we take on the problem. That is, in addition to assessing price competitiveness of hybrid optical/electrical interconnects, addressing the required system adaptations and showing viability, our end goal is to deliver on the untouched hypothesis as to whether such systems lead to better performance for parallel/distributed applications and to what extent. Our workload-centric approach is reflected in our work and specifically in this paper by developing a fully-functional prototype used to evaluate the cost-constrained performance of target parallel workloads.

3 System Architecture and Competitiveness

Incrementally to the architectures mentioned in section 2, we are interested in exploring the scalability limits posed by the hybrid architecture under the constraint of using commercially available equipment (cf. section 4), as well as comparing the cost competitiveness of the approach against currently employed network solutions.

3.1 Data- and Control-Plane Architecture

We depict a full-scale embodiment of our system architecture in Figure 1, comprising 320 server racks and a dual network option: a) a high-speed single-level circuit-switched network driven by high-speed (10Gbps in this embodiment) Ethernet Top of Rack (TOR) switches (depicted as *TOR-X* in Figure 1) and b) a lower-rate, packet-switched Ethernet network driven by lower-rate (1Gbps in this embodiment) Ethernet TOR switches (depicted as *TOR-B-X* in Figure 1). The server integration factor (32 servers/rack) stems from the currently "standard" 64-port density of high-end 10Gbps Ethernet switches used as TOR switches, allowing construction of full bisection bandwidth trees for racks of such integration. The high-speed network is implemented with commodity Micro Electro Mechanical Systems (MEMS) optical switches that exhibit interesting features (cost/port, rate-free, protocol-agnostic, low power consumption) to be used as cluster/datacenter interconnects. The ability to arbitrarily cross-connect any pair of ports of any MEMS switch enables direct low-latency connectivity between racks, as opposed to the cumbersome switching and high-latency that multi-level electronic interconnects suffer from (e.g. fat-trees). Still, MEMS optical switches suffer inherently from high - relative to the transmission time of a typical packet or message size at 10Gbps - switching latency (in the order of tenths of milliseconds) and therefore can only be perceived as circuit switching elements, carrying high-volume, long-lived flows between pairs of racks in our system. Lower-rate communication (e.g. short messages, barriers, application signaling), occurs in our system via the lower-end electronic network built out of inexpensive Ethernet switches that are arranged in a highly over-subscribed tree topology (bisection bandwidth is 12.5% of the full in the embodiment shown in Figure 1).

In the control-plane, the low-rate electronic part of our network architecture can be realized with well-researched solutions (e.g. [6]) for implementing large-scale networks over redundant topologies of Ethernet switches without applying

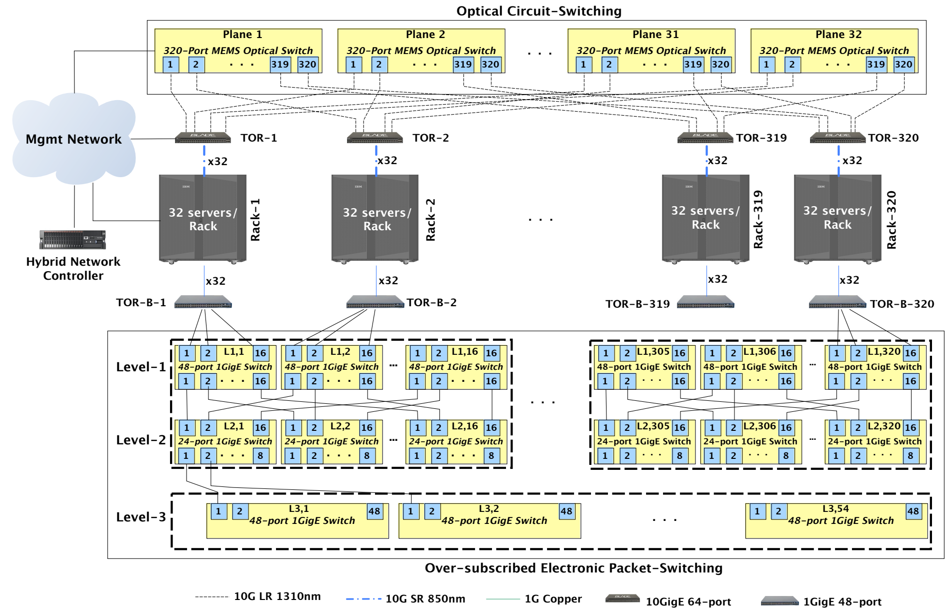


Fig. 1: Cluster architecture comprising a hybrid optical/electrical network spanning 10240 servers at an integration factor of 32 servers/rack.

modifications to hardware or protocol standards. We believe that such solutions have matured, therefore we focus our research and system prototyping efforts on the control-plane of the optical part of our network architecture. As shown in Figure 1, all devices comprising the network (MEMS optical switches and TOR switches) and server racks connect via a low-rate management network to a dedicated server hosting specialized software that implements a *network controller*. The primary role of the network controller is to periodically, or upon application request, configure the optical part of the network in a manner that benefits the execution of parallel/distributed applications. At a high-level, the controller delivers this in a three step process: a) ingest input specifying application task mapping to server racks, b) calculate an optical network topology that maximizes throughput, constrained on available network resources (spare optical ports) and c) implement the computed optical network topology by applying corresponding cross-connections to the array of MEMS optical switches and by applying required state changes to TOR switches facing the optical MEMS switches. We elaborate further in the workings of our network controller and the specificities of the control-plane of the optical network part in section 4.

Our previous work [1] proposed efficient algorithms for solving steps a) and b) of the above process and showed the performance improvement they bring at various scales via simulation with application traces as input. The present work closes the loop of this part by addressing step c) of the aforementioned cycle in commodity systems, as well as by showing system-level feasibility and performance improvement brought to real workloads in a real system prototype. Recognizing that in advance knowledge of the application communication pattern cannot be assumed across all parallel applications, we limit the scope of this work to a class of applications that we term ”static”. The term static refers here to the fact that these workloads exhibit per application logic (e.g. mesh simulations): a logical communication pattern that is invariant over application executions and that can be profiled through a test execution. Extending the scope of our research to embrace applications exhibiting dynamic communication patterns, as well as evaluate the performance impact of their co-existence with static parallel applications is part of our ongoing work.

3.2 Competitiveness Analysis

Since the changes we are proposing are to a great extent disruptive and not just incremental to an existing architecture, we assess in the following the estimated total list price of our network architecture and compare it against a conventional electronic packet network of equal nominal bisection bandwidth performance, namely a fat-tree implemented with top-end Ethernet switches [6]. The fact that our architecture is employing solely commodity off-the-shelf equipment is helpfull in assessing the price competitiveness of our approach. We recognize that list prices can be highly volatile, subject to market demand and the maturity of technology and therefore the following analysis can only serve as a snapshot of today that may not endure over time. Still, we contend that this is the only objective approach for drawing cost-related conclusions, when it is nearly impossible to scientifically reason about any mid- or long-term price trends (e.g. we had a hard time validating the cost trends reported in [3], even two years after their appearance).

We start with pricing our hybrid optical/electrical network solution for a cluster size of 10K server, equivalent to the system depicted in Figure 1. A hybrid network of this size can be readily built today using commercially available 320-port MEMS optical switches (e.g. Calient S320), while the rest of the equipment is commonly used in building high-end clusters and datacenters. We list the equipment description, corresponding prices per item and symbols used to refer to each equipment type in Table 1. All list prices used were drawn from publicly available sources [7], with the exception of the list price of the optical MEMS switch port, for which we used an averaged representative list price after discussions we had with respective vendors. Implementing the three levels of the low-rate electronic part of the hybrid network requires $\#S1_{1G} = 694$ 48-port 1G switches, $\#S2_{1G} = 320$ 24-port 1G switches and $\#C_C = 7680$ copper cables. We then calculate the total price of the optical part of the network, as a function of the nominal bisection bandwidth of the optical network part. For this, we use an integer parameter β that denotes the divisor that needs to be applied to the full bisection bandwidth to derive the nominal bisection bandwidth of the optical part of the network. For instance, β=1 corresponds to the full-bisection bandwidth setup shown in Figure 1, while β=4 corresponds to applying 1:4 over-subscription to the network that optically connects racks (or equivalently,

Table 1: List of equipment and corresponding list price/item used in the analysis

Symbol	**Equipment Name**	**Equipment Description**	**Price/Item [$]**
S_{10G}	IBM RackSwitch G8264R	10Gbps Ethernet TOR switch	30,000
T_{LR}	IBM SFP+ SR Transceiver	10Gbps 850nm Transceiver	665
T_{SR}	IBM SFP+ LR Transceiver	10Gbps 1310nm Transceiver	1600
OPT	MEMS Switch (96-320 ports)	Price per optical port	340
$S1_{1G}$	Juniper 48 Port 1Gb EX2200	1Gbps Ethernet Switch	3595
$S2_{1G}$	Juniper 24 Port 1Gb EX2200	1Gbps Ethernet Switch	1995
C_{MM}	LC-LC 50μm Fiber Cable	Multi-mode fiber cable	28
C_{SM}	9μm Fiber Cable	Single-mode fiber cable	25
C_C	Cat5 Copper Cable	Copper cable for Gb Ethernet	10

that each server can source/sink at a maximum off-rack traffic rate of 2.5Gbps at full network load). We note that over-subscription leads to fewer fiber links between the TOR switches and the optical array and thus to a reduction in the number of optical MEMS switches (”optical planes”) required. Following from the above, implementing the optical part of the network for 10K servers requires $\#OPT = \frac{32}{\beta}$ 320-port MEMS optical switches, $\#T_{LR} = \frac{320 \cdot 32}{\beta}$ LR-transceivers, $\#C_{SM} = \frac{320 \cdot 32}{\beta}$ single-mode fiber cables, $\#S_{10G} = 320$ 10G Ethernet TOR switches, $\#T_{SR} = 320 \cdot 32$ SR-transceivers and $\#C_{MM} = 320 \cdot 32$ multi-mode fiber cables. Next, we breakdown the equipment quantity required to realize a fully electronic fat-tree (again parametrically to its bisection bandwidth) built out of 64-port 10G low-latency Ethernet switches (e.g. IBM RackSwitch G8264R) as in [6]. Due to space limitations, we defer here a concise presentation of the fat-tree structure and dimensioning and refer the reader to [8]. Parametrically to β carrying the semantic defined above, implementing an all-electronic fat-tree in this manner requires three levels and particularly: $\#S_{10G} = 320 + \frac{320}{\sqrt{\beta}} + \frac{160}{\beta}$ 10G 64-port Ethernet switches, $\#T_{SR} = 10240 + \frac{20480}{\sqrt{\beta}} + \frac{20480}{\beta}$ SR-transceivers and $\#T_{SR} = 5120 + \frac{10240}{\sqrt{\beta}} + \frac{10240}{\beta}$ multi-mode fiber cables. We note that the $\sqrt{\beta}$ factor comes from the fact that over-subscription is applied in uniform multiplicative steps as we move from the first to the third level of the fat-tree.

Using the price values listed in Table 1 and the item quantities calculated for each network separately above, we calculated the total list price of a hybrid (resp. a fat-tree) network interconnecting a 10K server cluster and plot the results at various capacity levels in Figure 2. We observe that the hybrid network is by 31% cheaper at maximum capacity and remained cheaper compared to the all-electronic fat-tree throughout for all capacity levels up to the minimum capacity that is conceivable for the hybrid network (corresponding to β=32 or 10Gbps exiting the rack using one optical MEMS switch).

We note here that the cluster size picked for our analysis is the maximum, for we used the maximum size of MEMS switches that are commercially avail-

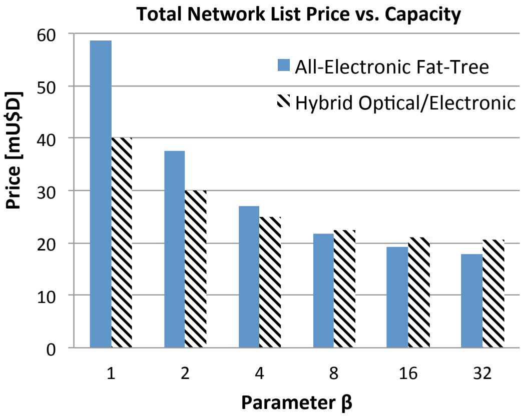


Fig. 2: List price of a hybrid network (resp. a full-electronic fat-tree) interconnecting a 10K server cluster at various capacity levels. The parameter β denotes the bisection bandwidth of the network as the fraction of full bisection bandwidth

able today. Scaling such setups beyond this limit would require either building denser switches (1024-port MEMS switches have been shown in-vitro [2]) or experimenting with multi-stage alignments of existing commercial optical switches (constrained on optical performance requirements). The latter path forms part of our future agenda for creating wider scale-out designs.

4 Network Control and Host Adaptations

4.1 Network Controller

The role of the network controller is to ingest input expressing the communication requirements of a mapped workload, compute a ”good” configuration of the (re-configurable) optical network for the given input and take all necessary control-plane actions to enforce the computed configuration on all involved devices. We depict a toy but illustrative example showing the steps taken by the controller upon receiving a request to match the optical infrastructure to an input workload in Figure 3. The input comes in the form of a traffic matrix, whereby each matrix element corresponds to the (normalized) volume of communication between two processing elements (cores). Following a clustering step to derive the rack-level traffic matrix and given the physical connectivity (wiring between TORs and optical switches) that the controller discovers during its initialization phase, the controller computes in the next step a connectivity graph between the racks involved, aiming at minimizing average traffic load throughout workload execution and thus speeding up workload completion. We haved presented the theoretical and optimization underpinnings of these steps and evaluated the performance of our topology configuration heuristic via simulation in [1]. We have implemented these in our network controller prototype for the purpose of

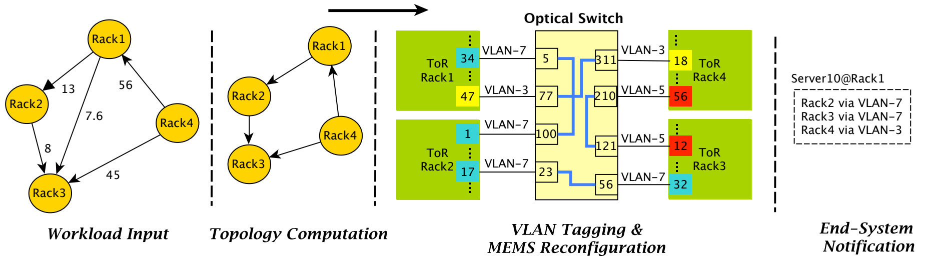


Fig. 3: Steps undertaken by the hybrid network controller to match the optical part of the hybrid network to the input workload

showing viability and to obtain the performance evaluation results reported in the next section.

In the next phase, the controller enforces the computed workload-specific topology to the physical network. For this, it sends the right set of commands (via the specialized TCP-based API) to all optical switches involved to cross-connect the pairs of ports corresponding to the computed connectivity between TOR switches. In parallel, the controller tags the TOR switch ports with VLAN-IDs, whereby each circuit is assigned a distinct (in the broadcast domain it touches) VLAN-ID. The reason we choose to operate our forwarding substrate using VLANs is to allow parallel links, rings and generally setup of paths that would otherwise be impossible, had we used Ethernet's spanning tree routing. A similar approach has been employed in [9] using static VLAN allocation, which is though shown not to scale. Instead, we measured that our controller is capable of installing VLANs in up to 32 TOR switch ports in less than 1.5 seconds and therefore we employ dynamic VLAN allocation for increased scalability. Generally, the multi-threaded implementation of our controller achieves state installation across all network devices in less than 2 seconds, which is a negligible overhead compared to the runtime of scaled-out workloads. Last, it is important to note that unlike alternative solutions, our system is able to utilize "multi-hop" communication, i.e. have a flow traverse the optical array multiple time until it reaches its destination TOR. This increases the search space for good topologies, while we have also been able to obtain better sharing and thus utilization of the optical resources.

4.2 End-system Support

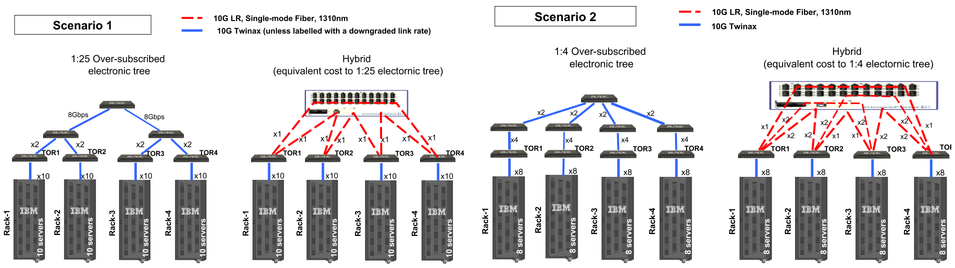
We leverage on the vast set of configuration tools and networking software available at commodity servers to send/receive packets to/from the two networks existing in the hybrid system, in fact without applying any modification to the underlying operating system or the application(s). For this, we injected a custom *translation shim layer* into the network stack of each server. The translation service uses as input the connectivity information communicated by the network controller (see last step in Figure 3) and creates the required virtual network interfaces accordingly. Given that each server has two network interfaces (leading to the optical network or the low-rate electronic network), the shim layer needs to decide which interface to pick to forward the packets of a specific workload. We accomplish this in a manner transparent to applications by having our shim layer rewrite the IP source/destination header values of each packet using the NAT feature of iptables. Specifically to the optical network case, the shim layer rewrites the IP address headers in a way that the packets are routed via the right VLAN and thus the circuit that leads to the destination rack of the packets. We defer here due to space limitations a more thorough presentation of the internal workings of our shim layer, which we plan to report in future public communication.

5 System Validation and Evaluation Results

We conducted various trials to validate our system prototype and targeted experiments to compare the performance of our solution to that of standard electronic tree-based solutions. Our testbed comprised 40 servers (12 cores each) mounted in 4 racks, eight 10G Ethernet ToR switches (IBM RackSwitch G8264) and one MEMS optical switch with 96 bi-directional ports (Crossfiber Liteswitch 96). Each server connects via a 10G SR-transceiver to each rack's ToR switch and each of the 4 ToR switches that have servers attached connects via 10 LR single-mode transceivers to the MEMS switch. We also used four additional 10G Ethernet switches to create a slice of an all-electronic fat-tree network. Our network controller ran on a dedicated server that connects via an 1G management network to the management ports of the TOR switches and the optical switch, as well as to all servers.

The rationale behind the creation of our experimental scenarios is as follows: constrained on the scale of our prototype (10 fiber links from each TOR to the optical switch), our goal was to compare instantiations of our hybrid network prototype against **equal-cost** instantiations of an all-electronic fat-tree; and in fact do so using real parallel workloads. To this end, we used our cost models to obtain two scenarios, each comprising two equal-cost network instantiations: a) scenario-1 compares a hybrid network with 6 TOR-to-optical fiber links against a 1:25 over-subscribed fat-tree and b) scenario-2 compares a hybrid network with 20 TOR-to-optical fiber links against a 1:4 over-subscribed fat-tree.

Our use-case involves a 10K multi-tenant cluster (or datacenter) with 32 servers per rack and a user requesting to execute a parallel job. The user is effectively allocated the requested number of servers in different racks. To address the general case, where this allocation may lead to racks without physical proximity - due to resource fragmentation or for better resilience against shared risk failures - we force inter-rack communication in the three-level fat-tree case to traverse the root of the tree. For scenarios 1 and 2 we assume the use of 10 and 8 servers in each of the 4 racks, respectively, and a uniform bandwidth allocation to servers in each rack. As such, in both the hybrid and the fat-tree networks the 10 servers in each rack in scenario-1 are allocated 1/3 of the available inter-rack bandwidth


Scenario 1
10G LR, Single-mode Fiber, 1310nm
10G Twinax (unless labelled with a downgraded link rate)
1:25 Over-subscribed electronic tree
Hybrid (equivalent cost to 1:25 electornic tree)
8Gbps
8Gbps
x2
x1
TOR1
TOR2
TOR3
TOR4
x10
Rack-1
Rack-2
Rack-3
Rack-4
IBM
10 servers
Scenario 2
10G LR, Single-mode Fiber, 1310nm
10G Twinax
1:4 Over-subscribed electronic tree
Hybrid (equivalent cost to 1:4 electornic tree)
x2
x4
x1
x8
8 servers

(a) Scenario 1

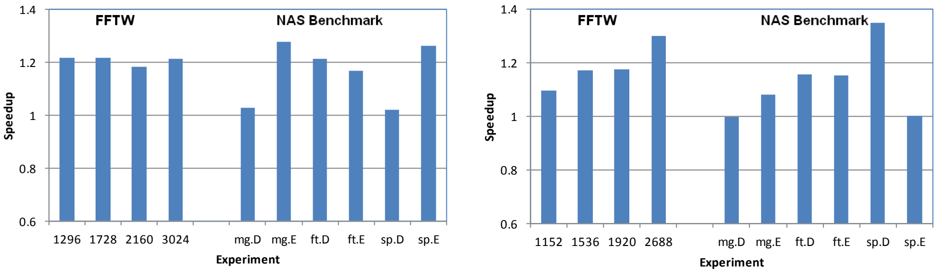
(b) Scenario 2

Fig. 4: Network configurations implementing the two experiment scenarios for the two network types under test in our testbed.

(8 Gbps and 2x10 Gbps for the tree and the hybrid respectively), while the 8 servers in scenario 2 are allocated 1/4 of the available inter-rack bandwidth (2x10 Gbps and 5x10 Gbps for the tree and the hybrid respectively). Figures 4a and b illustrate the instantiations of the fat-tree and hybrid networks according to the two scenarios outlined above.

We executed the following MPI parallel applications over all four network configurations: FFTW [10] which is a discrete Fast-Fourier Transform kernel, the FT (discrete 3D fast Fourier Transform) kernel, the MG (Multi-Grid on a sequence of meshes) kernel, and the SP (Scalar Penta-diagonal solver) pseudo-application; the last three are part of the NAS Parallel Benchmarks (NPB) suite [11]. For scenario-1 (40 servers in total) we used 4 input sizes for the FFTW ranging from 1296x1296x1296 to 3024x3024x3024, while for scenario-2 (32 servers in total) we used again 4 input sizes ranging from 1152x1152x1152 to 2688x2688x2688 . For FT, MG, SP NAS benchmarks we executed class D and E problem sizes. In the case of the hybrid network, the parallel execution involved using our network controller stack and utilizing our VLAN and translation shim-layer solution. Note that in the hybrid network configurations of both scenarios the constructed topologies include loops, which would be broken by disabling one or more link, if standard Ethernet switching was used. Instead, our VLAN-based routing enabled the loops, thus yielding higher throughput over the same network configuration.

Figure 5a shows the *speedup* results obtained in the set of experiments for scenario-1 that is, for the case of 1:25 tree and the equivalent cost hybrid network and 40 servers in total. In this context, speedup is defined as the ratio of the completion time of a single application execution in the tree network over the completion time in the hybrid network. Results show a measurable acceleration across all tested workloads, reaching up to 30%. Figure 5b depicts the speedup results for scenario-2, that is, for the case of 1:4 tree and the equivalent cost hybrid network and 32 servers in total. The improvements in this scenario are quite similat to those observed in the 1:25 case. For small size problems (small fft



(a) Scenario 1

(b) Scenario 2

Fig. 5: Evaluation results of speedup achieved by the hybrid optical/electrical network over the all-electronic fat-tree to the various workloads tested in scenario-1(left) and scenario-2(right).

problem sizes and NAS class D experiments) accelaration is low, for the capacity provided to the network in both cases is enough to satisfy the communication needs of the executed workloads. For large problems though, acceleration was higher, up to 35%. These workloads are bandwidth demanding and there is a clear advantage of capacity in favor of the hybrid network, which accelerates the tested workloads.

6 Conclusions

Despite all the research effort put on system specification, prototyping and evaluation of system features for hybrid optical/electrical interconnects in support of large-scale server co-locations (HPC clusters and datacenters), none of these efforts has to the best of our knowledge reached production deployment to date. Although we recognize that the shift from an entirely packet-switched to a hybrid circuit-/packet-switched world is per se not easy, we contend that the above is to a great extent due to the lack of evidence with regard to the benefit that such a shift can bring to applications. To address this gap, this work presented essentially a cost/benefit analysis for such systems. In particular, we delivered a concise price analysis of a hybrid interconnect comprising commodity parts and showed that it is cheaper compared to its most prominent competitor, namely a full electronic fat-tree. To deliver on the benefit part, we prototyped a network controller that computes efficient workload-input specific topologies and is capable of orchestrating the control-planes of the various network devices and the network stack of end-systems involved to create an optical substrate transparent to applications using it. We deployed our system prototype in a 4-rack testbed and showed through real experimentation that in most cases tested parallel workloads are accelerated at an equal network investment with a state-of-the-art solution.

Based on these promising findings, we are conducting work on expanding the range of applications evaluated, as well as scaling-out our testbed to enable larger-scale experiments. Our future agenda contains also dealing with applications with dynamically changing communication patterns and evaluating them in a multi-application scenario.

References

1. Christodoulopoulos, K., Ruffini, M., O'Mahony, D., Katrinis, K.: Topology configuration in hybrid eps/ocs interconnects. In: Proceedings of Euro-Par 2012. Lecture Notes in Computer Science, vol. 7484, Springer (2012) 701–715
2. Barker, K.J., Benner, A., Hoare, R., Hoisie, A., Jones, A.K., Kerbyson, D.K., Li, D., Melhem, R., Rajamony, R., Schenfeld, E., Shao, S., Stunkel, C., Walker, P.: On the feasibility of optical circuit switching for high performance computing systems. In: Proc. ACM/IEEE SC 2005 Conf. Supercomp. (2005)
3. Farrington, N., Porter, G., Radhakrishnan, S., Bazzaz, H.H., Subramanya, V., Fainman, Y., Papen, G., Vahdat, A.: Helios: a hybrid electrical/optical switch architecture for modular data centers. SIGCOMM Comput. Commun. Rev. **40** (August 2010) 339–350
4. Wang, G., Andersen, D.G., Kaminsky, M., Papagiannaki, K., Ng, T.E., Kozuch, M., Ryan, M.: c-through: part-time optics in data centers. SIGCOMM Comput. Commun. Rev. **40** (August 2010) 327–338
5. Chen, K., Singlay, A., Singhz, A., Ramachandranz, K., Xuz, L., Zhangz, Y., Wen, X., Chen, Y.: Osa: an optical switching architecture for data center networks with unprecedented flexibility. In: Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation. NSDI'12, Berkeley, CA, USA, USENIX Association (2012) 18–18
6. Greenberg, A., Hamilton, J.R., Jain, N., Kandula, S., Kim, C., Lahiri, P., Maltz, D.A., Patel, P., Sengupta, S.: Vl2: a scalable and flexible data center network. In: Proceedings of the ACM SIGCOMM 2009 conference on Data communication. SIGCOMM '09, New York, NY, USA, ACM (2009) 51–62
7. IBM: ibm.com. `http://www.ibm.com/` (2013)
8. Dally, W., Towles, B.: Principles and Practices of Interconnection Networks. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2003)
9. Zhang, X.J., Wagle, R., Giles, J.: Vlan-based routing infrastructure for an all-optical circuit switched lan. In: Proceedings of the 28th IEEE conference on Global telecommunications. GLOBECOM'09, Piscataway, NJ, USA, IEEE Press (2009) 365–370
10. Frigo, M., Johnson, S.: The design and implementation of fftw3. Proceedings of the IEEE **93**(2) (feb. 2005) 216 –231
11. NASA: Nas parallel benchmarks. `http://www.nas.nasa.gov/publications/npb.html/` (2013)