

---

# A THROUGHPUT-OPTIMIZED OPTICAL NETWORK FOR DATA-INTENSIVE COMPUTING

---

A NEW CIRCUIT-SWITCHED NETWORK ARCHITECTURE WITH OPTICAL-SWITCH AND BURST-MODE TRANSCEIVER TECHNOLOGY IS PROPOSED TO SUPPORT DEMANDING GRAPH ALGORITHMS IN A DISTRIBUTED-MEMORY SYSTEM. NETWORK SIMULATIONS PREDICT THAT THE SYSTEM COULD ACHIEVE GRAPH PERFORMANCE ON PAR WITH TODAY'S LEADING SUPERCOMPUTERS, AND ITS LIMITED POWER CONSUMPTION WOULD RESULT IN SEVERAL ORDERS OF MAGNITUDE OF EFFICIENCY IMPROVEMENTS THAT COULD ALLOW THE SYSTEM TO FIT WITHIN A FEW RACKS.

Laurent Schares

Benjamin G. Lee

Fabio Checconi

Russell Budd

Alexander Rylyakov

Nicolas Dupuis

Fabrizio Petrini

Clint L. Schow

IBM T.J. Watson Research

Center

Pablo Fuentes

Oliver Mattes

Cyriel Minkenberg

IBM Research-Zurich

.....The amount of data in our world has been exploding. Companies capture trillions of bytes of information about their customers, suppliers, and operations, and millions of networked sensors are being embedded in the physical world in devices such as mobile phones and automobiles, sensing, creating, and communicating data. Multimedia and individuals with smartphones and on social network sites will continue to fuel exponential growth.

The nature of unstructured data, which often comprises connections, relations, and interactions among entities, frequently lends itself well to graph-based models. As a consequence, the huge demands on the processing systems translate directly into a renewed need for efficient and scalable computer architectures to process graph algorithms.<sup>1</sup> A graph's individual data points, called *vertices*, often contain little information, but the many connections between vertices, called *edges*, are

challenging for a computing system to traverse in a timely manner. A machine's performance on graph algorithms is therefore dominated by its communication capabilities.

The challenges posed by processing huge graphs include, but are not limited to, how to keep up with updates that could have very high frequencies, how to monitor those updates, how to classify entities, and in general how to answer queries that have very little structure to exploit for optimization purposes.

To make things more challenging, graph analytics on unstructured data typically require a global look, posing extreme demands on the interconnection network that must efficiently support irregular communication patterns under high communication loads.

Although electrical networks continue to scale with advances in topologies, routing, and flow-control mechanisms,<sup>2</sup> optical switches promise to enable lower-power<sup>3</sup> and lower-latency networks than electrically switched

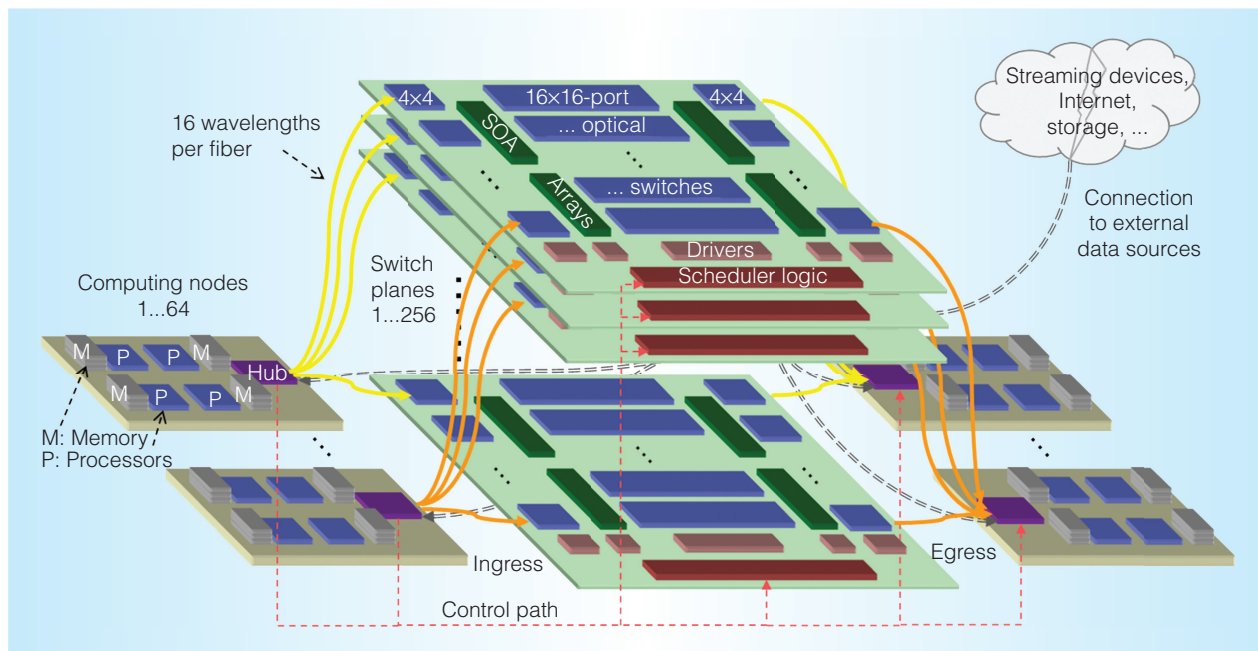


Figure 1. A 64-node system interconnected by 256 reconfigurable photonic-switch planes. Each plane contains silicon photonic switches, drivers, and scheduler logics, and optical amplifiers to overcome the insertion loss. Such a network could provide the enormous all-to-all bandwidth that may be required to solve large-scale graph applications in real time.

networks of equivalent bandwidth. However, rapidly reconfigurable, high-port-count optical switches with low insertion loss do not yet exist. Switching times of few nanoseconds have been demonstrated with chip-scale optical switches, but their radices have been limited to  $16 \times 16$  ports at best.<sup>4-7</sup> Researchers have used optical switches based on MEMS mirrors with hundreds of ports in circuit-switched network topologies within datacenters and supercomputers,<sup>8</sup> but the millisecond-scale reconfiguration times greatly exceed the target message latencies for big-data applications. Furthermore, building a low-latency network around fast photonic switches requires optical links that support rapid reconfiguration, often called *burst-mode capability*.<sup>9</sup>

As part of the multiyear DARPA Photonic Optimized Embedded Microprocessors (POEM) program ([www.darpa.mil/Our\\_Work/MTO/Programs/Photonically\\_Optimized\\_Embedded\\_Microprocessors\\_%28POEM%29.aspx](http://www.darpa.mil/Our_Work/MTO/Programs/Photonically_Optimized_Embedded_Microprocessors_%28POEM%29.aspx)), IBM Research is developing an optimized network architecture and requisite optical transceiver and switch technology to enable a petascale computer in a single or few racks. The IBM Throughput

Optimized POEM System (TOPS) program aims to develop technologies and architectures required to implement a distributed-memory system with high bandwidth and low latency enabled by a transparent photonic core circuit switch capable of rapid reconfiguration. The computing nodes may incorporate technologies developed by other groups that are part of the larger POEM program.<sup>10,11</sup> The proposed photonic switch fabric offers unique capabilities that overcome pin-count and power-dissipation limitations of electrical networks to deliver bandwidth of multiple terabytes per second. This could allow a departure point from the current computer-architecture trajectory of lightly connected systems tuned for peak performance, to a new path toward highly connected distributed-memory systems that excel at sustained productivity on data-intensive workloads.

### A circuit-switched optical network for data-intensive computing

The TOPS network is based on a fast optical core switch. The all-to-all network fabric shown in Figure 1, configured as

multiple planes of high-radix wavelength-division-multiplexed (WDM) switches, offers path diversity for flexible connectivity ranging from one-to-all broadcast with partitioned bandwidth to full-bandwidth connections between pairs of nodes. This flexibility enables degrees of freedom in optimizing the switch scheduling and allocation algorithms to minimize latency and maximize performance for fine-grained communication.

### Network architecture

We focus on a midsize computing system that can be contained in a single or few racks, based on relatively few high-performance nodes with large amounts of memory, interconnected by the TOPS network shown in Figure 1. Each node in the 64-node machine in this example has an assumed off-node bandwidth of more than 10 Tbytes per second (TBps), realized through 256 fiber pairs per node that each contain 16 wavelength channels operating at 20 Gbits per second (Gbps). The large number of optical-switch planes, 256 in our initial architecture, supports the total network bandwidth with the added benefit of significant path diversity to manage congestion. A lightweight electrical control network configures and controls the core optical network and the ingress and egress points. A network scheduler configures the switch planes, and a custom hub chip in each node is the network access point.

### Computing-node architecture

To produce a system design that achieves the highest possible performance, we must consider the complete datapath architecture. We must optimize the computing nodes for memory-intensive problems, balancing memory with computation and on- and off-node communication. Critical factors are the total memory capacity that can be packaged on a node, as well as low-latency direct-memory access across the whole machine. The two node designs we consider here are based on technology anticipated to be available within 5 to 8 years; our straw men are based on a preliminary node-packaging design and our extrapolations of commercial processor and memory roadmaps and the *International Technology Roadmap for Semiconductors*. The *traditional node* has 8 tera-ops, 1 Tbyte of

memory, and 400 Gbytes per second (GBps) of off-node bandwidth. The *TOPS node* has 8 tera-ops, 8 Tbytes of memory, and a network gateway with 10 TBps of off-node bandwidth. A significant amount of new hardware acceleration, pipelining of memory requests, and message coalescing in the hub chip will be required to support such a large off-node bandwidth.

### Graph scale and performance estimations

The Graph500 list ([www.graph500.org](http://www.graph500.org)) is an alternative to the Top500 list ([www.top500.org](http://www.top500.org)) to rank computer performance on data-intensive computing applications. The Top500's critical processing metric, floating-point operations per second (flops), is replaced by traversed edges per second (TEPS) in the Graph500. Unfortunately, despite the importance of graph algorithms in underpinning many big-data applications, today's high-performance computing machines are designed for peak floating-point computation and often perform poorly on graph-type workloads. The Scale parameter specifies the graph size: the graph has  $2^{\text{Scale}}$  vertices and  $16 \times 2^{\text{Scale}}$  undirected edges.

To estimate performance, we consider a hypothetical Graph500 implementation based on a 1D decomposition of the graph.<sup>12</sup> If the network is the limiting factor, the time required to visit a graph with a given Scale and edge factor  $EF$  can be estimated as follows:

$$t_{\text{Graph500}} = B_e \times EF \times 2^{\text{Scale}+1} \times 1/T \\ \times (N - 1)/N \\ \times (1 + B_h/B_p)$$

where  $B_e$  denotes the bytes needed to represent an edge in a message (we use a conservative estimate of 8 bytes),  $B_h$  is the packet header length (8 bytes),  $B_p$  is the packet size (256 bytes), and  $T$  is the total network throughput. The  $(N - 1)/N$  factor indicates that, on average,  $1/N$  of the edges would be destined for the local node, and need not be transferred on the network. Finally,  $2^{\text{Scale}}$  is the number of vertices; an additional factor of 2 must be included to account for the fact that in a 1D-decomposition each edge  $(u, v)$  must

also be represented as  $(v, u)$ . The Graph500 performance metric, in TEPS, is given by  $EF \times 2^{\text{Scale}} / t_{\text{Graph500}}$ :

$$\text{TEPS} = \left[ 2B_c \times 1/T \times (N-1)/N \times (1 + B_h/B_p) \right] - 1$$

For a 64-node machine, we estimate 30,984 GTEPS (giga-TEPS) when equipped with TOPS nodes, compared to only 1,550 GTEPS with traditional nodes.

We can estimate the relation between memory capacity and graph scale as follows:

$$\begin{aligned} \text{Memory capacity} \\ = (\alpha \times 2^{\text{Scale}} \times 2^5 \times B_c) + \beta \end{aligned}$$

where  $2^5$  is the number of edges per vertex,  $2^3$  is the data representation (8 bytes/edge),  $\beta$  is a machine-dependent overhead, and  $\alpha$  is an overprovisioning factor that accounts for memory failures and the need for backup copies. Typically,  $\alpha$  is between 2 and 4, so we assume  $\alpha = 2$ . A 64-node machine with TOPS nodes could analyze a scale-40 graph, whereas with traditional nodes the scale would be limited to 37 at best.

This performance estimate considers searches on static graphs, such as breadth-first searches, weighted searches, or shortest-path analysis. Figure 2 qualitatively illustrates the dependence between graph performance and scale. The performance in TEPS can be limited by the amount of memory in the system or by the network throughput. We assume that both of our nodes are bound by the network bandwidth; that is, that they have a sufficiently large memory capacity and memory bandwidth. As a rule of thumb, a graph of Scale = 32 requires an approximate memory size of 1 Tbyte; this could represent a corporate intranet, for example. A graph of Scale = 40 requires a memory size on the order of several hundred Tbytes, which might be required for Internet-scale security monitoring.

### Figures of merit

Figures 3a and 3b plot the graph performance versus the number of nodes, illustrating a large performance improvement of  $20\times$  for a machine with 64 TOPS nodes over a similarly sized machine with traditional nodes.

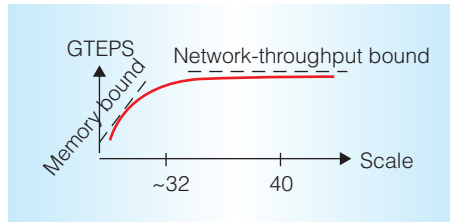


Figure 2. Characteristic dependence between graph performance (in traversed edges per second, or TEPS) and graph scale. The performance in TEPS can be limited by the network throughput, and the scale can be limited by the amount of memory in the system. Within power and size constraints of a rack-scale system, the performance of an electrically-interconnected system appears to be limited.

The  $8\times$  difference in achievable graph scale is a consequence of the TOPS nodes' greater memory capacity. Although the machine's total memory capacity is a function of the packaging density and the total number of nodes, our main focus in this work is the significant increase in TEPS, a consequence of the massively larger off-node bandwidth.

Figure 3c compares the power consumption for the TOPS optical network to an equal-bandwidth, electrically switched fabric with optical I/O ports. The advantage of the WDM optical switches are illustrated in both scalability and power: for the electrical switch, the number of ports is multiplied by the number of wavelengths, because the electrical switch cannot switch a full WDM stream but must operate on each channel. Thus, far more electrical switches than optical switches are needed for equivalent bandwidth. In addition, the number of optical links in the electrical switch network is doubled compared to the optical network, because optical-to-electrical (O/E) and electrical-to-optical (E/O) conversion is required at the electrical-switch ports. We assume the optical-link efficiency to be 1 pJ/bit.<sup>10</sup> The optical-switch efficiency of 2.5 pJ/bit is largely caused by two optical amplifiers in each path. The basis for comparison is a 128-port electrical-circuit switch with a per-port switching efficiency of 10 pJ/bit, substantially below current commercial switches. The plot

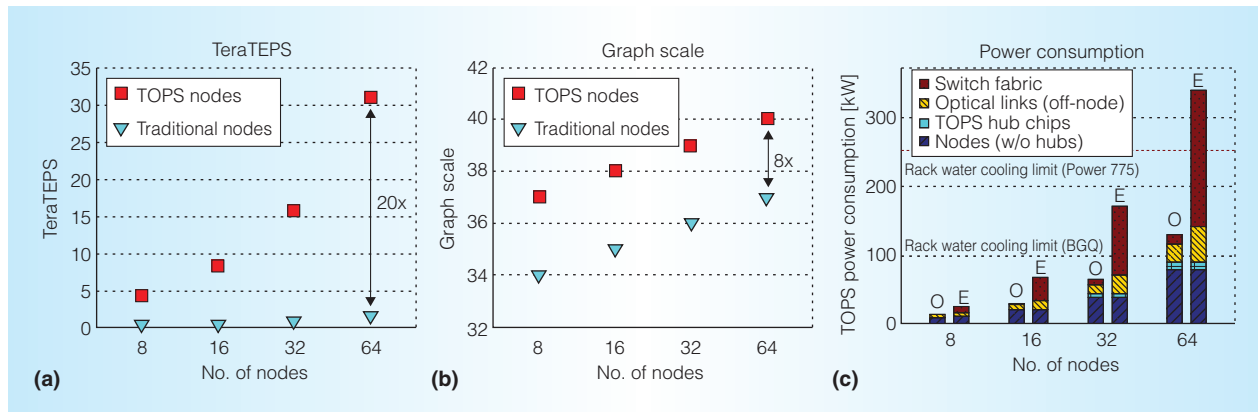


Figure 3. Figures of merit. TeraTEPS (a) and graph scales (b) versus number of TeraTEPS and traditional nodes. Power consumption versus number of TeraTEPS nodes, connected by either an electrical (E) or an optical (O) network (c). (TeraTEPS: Throughput Optimized POEM System.)

**Table 1. Comparison of performance targets to state-of-the-art IBM supercomputers.**

Performance metrics	BlueGene/Q <sup>13</sup>	Power 775 <sup>14</sup>	TeraTEPS target
Graph500 performance (GTEPS)	16,599	1,172	31,000
Graph efficiency (GTEPS/kW)	3.55	N/A	260
Peak performance per rack (Tflops)	200	96	1,000
GTEPS per (peak) Tflops	1.1	0.8	31
Efficiency (Tflops/kW)	2	0.4	10
Node escape bandwidth (TBps)	0.04	0.42	10
Total node-to-node latency (ns)	2,000	N/A	500
No. of I/O fibers per node	12	672	512
Data rate per fiber (Gbps)	10	10	320

includes water-cooled racks’ power limits. Our estimates show that a 64-node system with an electrical-switch fabric would not fit in a single rack with state-of-the-art water cooling. Our projections also show improved network performance efficiency expressed in GTEPS per network power. We estimate more than 5x improvement for a 64-node machine with an optical-switch fabric compared to a similarly sized machine with an electrical-switch fabric.

**Comparison to state-of-the-art supercomputers**

Table 1 compares key performance and efficiency metrics of IBM supercomputers to the TeraTEPS targets. We expect the TeraTEPS system to achieve over 70 times better efficiency on graph algorithms than IBM’s BlueGene/

Q-Sequoia, which is one of the highest-performing data analytic machines and most power-efficient high-performance computing machines to date. On the Graph500 June 2014 list, Sequoia achieved 16.6 TeraTEPS on a graph of Scale = 40, but this result required a machine with 65,536 nodes. The ambitious TeraTEPS target is to achieve on-par performance with a system with one or a few racks.

**Network technology**

Our machine’s potential benefits hinge on whether the optical-network technologies can deliver aggressive bandwidth and latency targets. Even though many of these technologies do not exist today to the degree



prescribed here, we outline a realistic vision for the hardware development that will be required to realize such a machine.

### Switch fabric

As Figure 1 shows, each switch plane interconnects one fiber from each node's transmission fiber array with one fiber from each node's reception fiber array through an  $N \times N$  optical-switch fabric, where  $N$  is the number of nodes in the system. If each node has  $P$  transmitting plus  $P$  receiving fibers, then  $P$  switch planes are required. In order for the entire switch fabric to occupy only a small portion of a rack, very high photonic integration density coupled with advanced 3D-IC packaging technology will be required. Within a switch plane, each optical path must allow parallel transmission on multiple wavelengths from source to destination.

### Switching technology

With traditional planar lightwave circuit technology, each switch plane would occupy a prohibitively large footprint. Recently, however, researchers have achieved significant advancements leveraging high-index-contrast silicon photonics. They have applied silicon photonics technology to demonstrate  $4 \times 4$  and  $8 \times 8$  port switches with monolithically integrated driver circuits operating over a large spectral bandwidth with nanosecond-scale reconfiguration times and milliwatt-scale power dissipations.<sup>4</sup> A looming challenge is to scale the number of switch ports beyond the single digits that have been shown to date, while being limited by the insertion losses incurred in the switch elements, waveguide crossings, I/O coupling, and routing.

### The need for optical amplification

Realistic assumptions for laser output power and receiver sensitivity result in optical-link-loss budgets that cannot be achieved without compensating for any switch losses by means of optical amplification. Semiconductor optical amplifiers (SOAs) are a natural solution, offering low power dissipation and a small footprint enabling integration into the switch planes. SOAs can provide ample gain over a broad spectrum and support high-bit-rate and multiwavelength operation.

We estimate that two stages of SOAs deliver sufficient amplification to compensate for expected losses of a  $64 \times 64$  port photonic switch. The two stages result in 128 SOAs per switch plane, which can be partitioned into multiple chips that contain arrays of individual amplifiers. With realistic power and area projections of 0.4 W and  $0.2 \text{ mm}^2$  per device, the integration of SOAs within the switch fabric appears to be feasible. The SOA chips will be integrated in a photonics-enabled carrier providing the active photonic-switching function. A switch driver and control IC will be packaged with the carrier to actuate the photonic switches and provide electrical power and control for the SOAs. The goal is to demonstrate a modular platform that can scale to larger port counts by incorporating additional switch and amplifier stages.

### Hybrid packaging platform

The realization of an optical switch that's scalable to a high bandwidth and port count presents numerous packaging challenges, such as the integration of multiple and varied photonic, electronic, and optical-coupling functions. We opt for a hybrid packaging approach wherein SOAs and electronic driver chips are flip-chip-attached to a silicon carrier with monolithically integrated optical-switch elements and photonic waveguide layers. Although heterogeneous wafer-scale integration of Indium Phosphide (InP) optical-gain elements in a silicon photonic platform is promising as a single-chip solution in the long term,<sup>15</sup> we chose separate SOA chips in this program because they can be independently optimized for performance, reliability, and yield. The carrier incorporates  $32 \ 4 \times 4$  port and four  $16 \times 16$  port optical-switch elements, together with waveguide-crossing regions. A nonoptimized layout results in a carrier area of  $600 \text{ mm}^2$ , which can be fabricated as a  $25 \times 25 \text{ mm}^2$  chip.

### Burst-mode and WDM-capable optical transceivers

We can implement the high-speed optoelectronic components required in the optical datapath between source and destination nodes—namely, modulators and photodetectors—using silicon photonic technology for high density and WDM compatibility.

Although many of the circuits have been demonstrated at high density and low power using advanced CMOS technology, one often overlooked element is a burst-mode clock-data recovery (CDR) unit.

When the photonic-switch network is reconfigured, the phase relationship between the clock and newly arriving data signals at the receiver in the destination node must be quickly established before valid bits can be transferred. Receivers designed to respond quickly to dynamic changes are generally referred to as *burst-mode receivers*. Although 10 Gbps burst-mode receivers are commercially available, existing standards have fallen short of our targeted power efficiencies (pJ/bit-scale) and locking times ( $< 50$  ns) by at least an order of magnitude.<sup>9</sup> We are undertaking to build burst-mode links well in excess of the state of the art in bit rate, synchronization time, and power efficiency, and we are leveraging system architecture features to facilitate the development.

#### Pipeline scheduler

Each network plane is controlled by an independent scheduler that arbitrates among requests for outputs of its associated plane and configures the plane accordingly on a packet-by-packet basis. The scheduler performance has a major impact on end-to-end latency, with fast decisions needed to keep up with the packet rate. To reduce the frequency of network plane reconfigurations and the associated CDR latency penalty, the scheduler operates incrementally, maintaining I/O connections persistently as long as there are requests or a timeout occurs (to ensure fairness). The scheduler's decisions are based solely on the current demand matrix; because of the fast switching times, our architecture does not require predictive demand estimation to establish long-lived optical light paths.

#### Performance evaluation

We implemented a high-level model of our system to evaluate the latency-throughput characteristics of the proposed many-plane optical fabric.

#### Simulation model

We model the system shown in Figure 1 at the packet level. The computing nodes are

represented by traffic sources and sinks that model the workload. Each source generates Bernoulli arrivals with different destination distributions of short (256-byte) fixed-size messages. A hub associated with each computing node is responsible for distributing and aggregating traffic to and from the parallel optical-switch planes. The hub incorporates an input buffer with one queue per plane, and an internal crossbar fabric to connect the processing nodes to the planes, and vice versa. A key function of the hub is the plane-mapping policy that assigns each message to a plane. This decision is made when the message arrives at the hub. Because each pair of nodes can communicate across multiple planes in parallel, the hub delivers messages in the same order in which they arrived. The hub also performs some first-in, first-out I/O buffering in the electronic domain and handles the request-grant control protocol to interface with the controller, delivering a request for each packet reaching its associated buffer's head. Credit-based flow control prevents overflow-induced data losses.

Each plane is configured by its own controller, which receives connection requests from the network interfaces and configures the optical paths correspondingly. The controller employs an asynchronous arbitration algorithm that favors serving requests for traffic flows that are already being served to maximize resource reuse, reducing the impact of reconfiguration delay on throughput performance. The asynchronous aspect means that arbitration is not strictly time slotted and synchronized across all ports: any I/O pair can be connected as soon as it becomes available.

Each plane's photonic datapath is modeled by a bufferless three-stage rearrangeable nonblocking Clos network composed of sixteen  $4 \times 7$  switches in the first stage, seven  $16 \times 16$  switches in the middle stage, and sixteen  $7 \times 4$  switches in the third stage. The reconfiguration time is much longer than the packet duration, so reconfiguring on a packet-by-packet basis would lead to inefficient network usage. Consequently, the controller only reconfigures an I/O pair when needed—that is, when a request for a different port is granted. When a request for the same port is granted, it maintains the existing connection, thereby eliminating both the

switching time and the CDR locking time. We assume that the controller can complete each scheduling decision within the duration of one packet.

### Plane-mapping policy

We considered three plane-mapping policies: *Random* assigns a plane randomly to each message, *Modulo* assigns a plane on the basis of the message's source and destination port (the planes are effectively partitioned across sources and destinations in a conflict-free manner), and *Backlog* assigns a plane adaptively on the basis of the per-plane backlog. Planes are preferably assigned according to the Modulo assignment, but planes can be reassigned on demand.

### Results

Table 2 lists the most important simulation parameters. The switching and CDR-locking times are conservative estimates of what can be achieved in hardware. The hub-processing delay assumes pipelining and hardware acceleration. The time-of-flight parameter assumes 4 m of optical fiber (rack-scale system). We simulated 10 different input load values, ranging from 10 to 99 percent of the maximum nominal injection rate of 10 TBps per node, with a network size of 64 nodes and 256 planes. Each run represents 100  $\mu$ s of simulated time. We collected the mean throughput and latency numbers across all messages received. The confidence intervals are 1 percent, with a confidence level of 99 percent on the throughput and 95 percent on the delay.

Figure 4 shows the relative throughput per node, and the mean message latency as a function of the traffic load, for uniform traffic (Figures 4a and 4b) and bit-complement traffic (Figures 4c and 4d). Each figure comprises one curve for each of the three plane-mapping policies: Modulo, Random, and Backlog. Figures 4a and 4b clearly demonstrate the Random policy's throughput limitation, which was caused by the overhead of frequent reconfigurations, whereas Modulo sustained close to 100 percent throughput. The Modulo policy, on the other hand, was severely throughput limited when subjected to permutation (bit-complement) traffic, having obtained only 1.6 percent of nominal

**Table 2. Simulation parameters.**

Parameter	Value
No. of nodes	64
No. of planes	256
No. of wavelengths	16
Rate per wavelength	20 Gbps
Aggregated rate per port	320 Gbps
Switching time	30 ns
CDR locking time	100 ns
Hub-processing delay	100 ns
Aggregated time of flight	19.8 ns
Total reconfiguration time	130 ns
Message size	256 bytes
Message duration	25 ps
Aggregated injection rate per node	10 Tbytes/second
Packet size	256 bytes
Packet duration	6.4 ns

throughput. For this traffic pattern, the Random policy performed well.

The Backlog policy, on the contrary, performed equally well for both traffic patterns, achieving full throughput under any load. These results indicate that an adaptive plane-mapping policy is critical to achieving high performance under varying or unpredictable traffic patterns. This policy combines the benefit of exploiting path diversity (as Random does) with that of minimizing reconfiguration overhead (as Modulo does).

An important advantage of this architecture is the flatness of the latency curve throughout most of the load range, which ensures predictable delay and low jitter, greatly benefiting workload performance. The main drawback—besides more complex decision logic in each hub—is a latency penalty compared to the Random policy (see Figures 4b and 5d). The additional latency is attributed to queuing delays caused by this scheme's conservative plane-assignment policy: a connection is assigned to additional planes only when the queues corresponding to the already assigned planes all exceed a threshold. Consequently, each new message is likely to encounter a backlog. Adjusting the threshold can influence this latency penalty. Lowering it lowers the latency at the hub, but could lead to premature plane



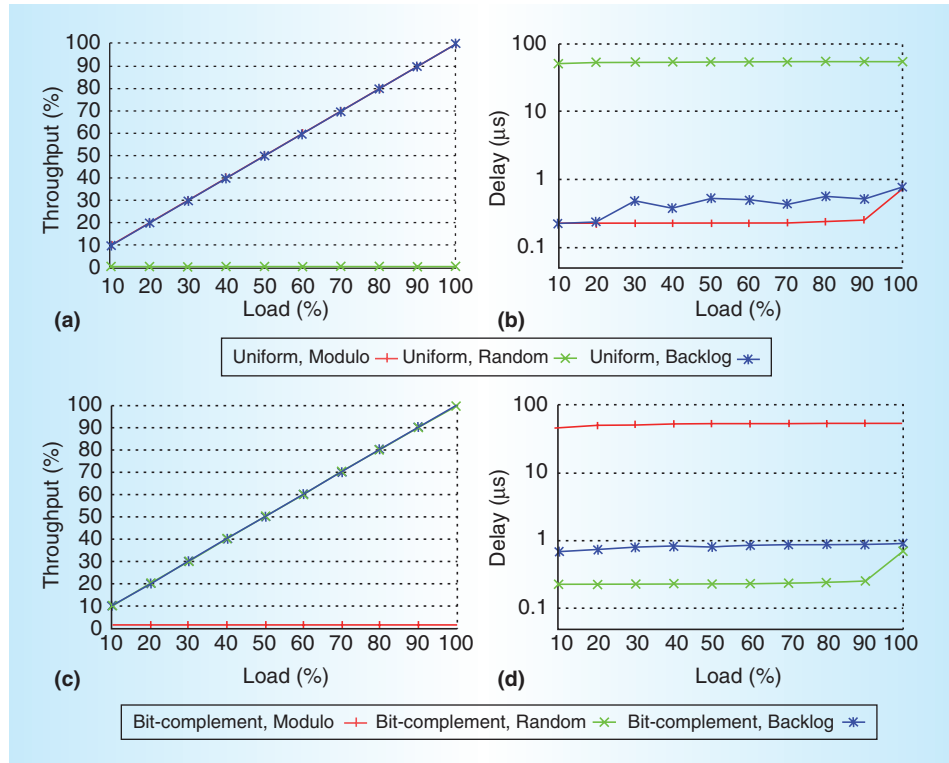


Figure 4. Throughput and latency versus load for a 64-node cluster: throughput load, uniform (a); delay load, uniform (b); throughput load, bit complement (c); delay load, bit complement (d). Backlog is the only policy that performed well for uniform as well as bit-complement traffic; although the other policies slightly outperformed Backlog on one of the traffic patterns, they were drastically worse for the other.

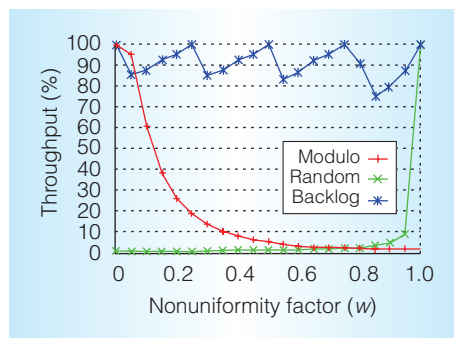


Figure 5. Throughput as a function of nonuniformity ( $w$ ) for three plane-mapping policies: Modulo, Random, and Backlog. Backlog achieved throughput in excess of 80 percent throughout almost the entire range of  $w$ , which shows that it can effectively adapt to traffic conditions. Although Modulo and Random performed well at one of the extremes, their performance quickly degraded when increasing (Modulo) or decreasing (Random) the value of  $w$ .

(re)assignments that cause unnecessary switch reconfigurations with associated utilization penalties. Hence, setting the threshold involves a careful tradeoff between latency and utilization that could be adjusted according to workload requirements.

In addition to the extreme cases of uniform and permutation traffic, we also studied the plane-mapping policy's impact on intermediate degrees of nonuniformity. To this end, we adopted a nonuniform traffic model characterized by a single parameter  $w$ , which represents the nonuniformity factor:  $w = 0$  corresponds to uniform traffic, and  $w = 1$  represents permutations. We vary the value of  $w$  from 0 to 1 and measure the throughput achieved at an offered load of 100 percent. Figure 5 shows the results for a system with 64 nodes.

As we expected, the Modulo mapping policy achieved excellent throughput for uniform traffic, but poor throughput for permutation traffic. Throughput dropped rapidly as

$w$  increased. The results for Random mapping were the exact opposite: high throughput for permutation traffic, low throughput for uniform traffic. The Backlog policy achieved throughput in excess of 80 percent throughout almost the entire range of  $w$ , confirming its capability to effectively adapt to traffic conditions. Backlog's behavior was better than Random and Modulo in almost all cases. This behavior would be more stable when increasing granularity in plane distribution, with a greater number of planes associated with a given source-destination pair than the current value of  $256/64 = 4$ . This coarser granularity makes it more difficult to precisely match the individual flow rates. This is also the cause of the zigzag pattern with increasing  $w$ .

We could analyze other workloads, such as one-to-many and many-to-one patterns, using suitable plane-mapping policies in our massively multipath architecture. As a next step, we will use trace-driven simulation models to confirm the results of the synthetic workload models. To facilitate that, we have implemented a complete software framework to collect timing information from existing Graph500 applications on actual high-performance computing systems such as BlueGene/Q, and to couple these obtained traces to our network simulator. With this capability, we can evaluate the architecture and network performance with realistic scenarios.

This work opens the door to further scalability assessments of optically switched systems at the network, control, node, and software levels. We believe that our architecture will become important for workload-optimized real-world computing, which increasingly involves data movements at the scale of the entire machine, requiring quick and efficient processing of massive petabyte-scale small-message datasets.

MICRO

## Acknowledgments

We thank Jagdeep Shah and Nicole DiLello for many useful discussions and valuable guidance. This work was supported by DARPA/ARL under contract W911NF-11-2-0059. The views, opinions, and/or

findings contained in this article are those of the authors and should not be interpreted as representing the official views or policies, either expressed or implied, of DARPA or the Department of Defense. Approved for public release, distribution unlimited.

## References

1. F.T. Leighton, *Introduction to Parallel Algorithms and Architectures*, Morgan Kaufmann, 1992.
2. J. Kim et al., "Technology-Driven, Highly Scalable Dragonfly Topology," *Proc. 35th Ann. Int'l Symp. Computer Architecture (ISCA 08)*, 2008, pp. 77-88.
3. R.G. Beausoleil, M. McLaren, and N.P. Jouppi, "Photonic Architectures for High-Performance Data Centers," *IEEE J. Selected Topics Quantum Electronics*, vol. 19, no. 2, 2013, doi:10.1109/JSTQE.2012.2236080.
4. B.G. Lee et al., "Monolithic Silicon Integration of Scaled Photonic Switch Fabrics, CMOS Logic, and Device Driver Circuits," *J. Lightwave Technology*, vol. 32, no. 4, 2014, pp. 743-751.
5. R. Stabile et al., "Monolithic Active-Passive  $16 \times 16$  Optoelectronic Switch," *Optics Letters*, vol. 37, no. 22, 2012, pp. 4666-4668.
6. Q. Cheng et al., "Modular Hybrid Dilated Mach-Zehnder Switch with Integrated SOAs for Large Port-Count Switches," *Proc. Optical Fiber Communication Conf.*, 2014, paper W4C.6.
7. L. Chen et al., "Electronic and Photonic Integrated Circuits for Fast Data Center Optical Circuit Switches," *IEEE Comm.*, vol. 51, no. 9, 2013, pp. 53-59.
8. K. Barker et al., "On the Feasibility of Optical Circuit Switching for High Performance Computing Systems," *Proc. ACM/IEEE Supercomputing Conf. (SC 05)*, 2005, pp. 16-37.
9. X-Z. Qiu et al., "Fast Synchronization 3R Burst-Mode Receivers for Passive Optical Networks," *J. Lightwave Technology*, vol. 32, no. 4, 2014, pp. 644-659.
10. A.V. Krishnamoorthy et al., "Computer Systems Based on Silicon Photonic Interconnects," *Proc. IEEE*, vol. 97, no. 7, 2009, pp. 1337-1361.

11. C. Batten et al., "Building Many-Core Processor-to-DRAM Networks with Monolithic CMOS Silicon Photonics," *IEEE Micro*, vol. 29, no. 4, 2009, pp. 8-21.
12. F. Checcoli et al., "Traversing Trillions of Edges in Real-Time: Graph Exploration on Large-Scale Parallel Machines," *Proc. Int'l Parallel and Distributed Processing Symp. (IPDPS 14)*, 2014, pp. 425-434.
13. D. Chen et al., "The IBM Blue-Gene/Q Interconnection Fabric," *IEEE Micro*, vol. 32, no. 1, 2012, pp. 32-43.
14. B. Arimilli et al., "The PERCS High-Performance Interconnect," *Proc. 18th IEEE Symp. High Performance Interconnects*, 2010, pp. 75-82.
15. B.R. Koch et al., "Integrated Silicon Photonic Laser Sources for Telecom and Datacom," *Optical Fiber Communications Conf.*, 2013, paper PDP5C.8.

**Laurent Schares** is a researcher at the IBM T.J. Watson Research Center. His research focuses on integrating optical technologies into future datacenter and supercomputer networks. Schares has a PhD in physics from the Swiss Federal Institute of Technology (ETH) Zurich.

**Benjamin G. Lee** is a research staff member at the IBM T.J. Watson Research Center. His research interests include photonic devices, integrated optical-switch fabrics, and highly parallel transceivers. Lee has a PhD in electrical engineering from Columbia University.

**Fabio Checcoli** is a research staff member in the High Performance Analytics Group at the IBM T.J. Watson Research Center. His research interests include distributed graph processing, graph algorithms, and real-time operating systems. Checcoli has a PhD in computer engineering from Scuola Superiore S. Anna in Pisa.

**Russell Budd** is a senior technical staff member at the IBM T.J. Watson Research Center. His research interests include wafer-debonding technologies, and silicon photonic-device design and packaging for data-center optical interconnects. Budd has a BS

in mechanical engineering from Michigan State University.

**Alexander Rylyakov** is a research staff member at the IBM T.J. Watson Research Center. His research focuses on digital phase-locked loops and integrated circuits for wireline and optical communication. Rylyakov has a PhD in physics from the State University of New York at Stony Brook.

**Nicolas Dupuis** is a postdoctoral researcher at the IBM T.J. Watson Research Center. His research interests include silicon photonics and optical-link modeling. Dupuis has a PhD in physics from Université de Lorraine.

**Fabrizio Petrini** is a researcher and manager of the High Performance Analytics Group at the IBM T.J. Watson Research Center. His research interests include data-intensive algorithms, exascale computing, and high-performance interconnection networks. Petrini has a PhD in computer science from the University of Pisa.

**Clint L. Schow** manages the Optical Link and System Design Group at the IBM T.J. Watson Research Center. His research focuses on developing high-speed and power-efficient multimode and single-mode optical links, nanophotonic switches, and new system architectures based on low-latency photonic-switching fabrics. Schow has a PhD in electrical engineering from the University of Texas at Austin. He is a senior member of IEEE and the Optical Society of America.

**Pablo Fuentes** is a PhD student in computer science at the University of Cantabria, Spain. His research focuses on the characterization and performance evaluation of system interconnection networks. Fuentes has an MS in telecommunications engineering from the University of Cantabria. He completed the work for this article while working as a research intern at IBM Research-Zurich.


**Oliver Mattes** is a PhD student in computer engineering at the Karlsruhe Institute

of Technology. His research interests include adaptive computer architectures, memory systems, simulation, parallelization, and high-performance computing. Mattes has a Diploma (MS) in computer science from the Karlsruhe Institute of Technology. He completed the work for this article while working at IBM Research-Zurich.

**Cyriel Minkenberg** is a research staff member in the Cloud and Computing Infrastructure Department at IBM Research-Zurich. His research interests include the architecture, design, and performance evaluation of large-scale interconnection networks for

cloud datacenters and high-performance computing systems. Minkenberg has a PhD in electrical engineering from the Eindhoven University of Technology.

Direct questions and comments about this article to Laurent Schares, IBM T.J. Watson Research Center, Route 134, Yorktown Heights, NY 10598; schares@us.ibm.com.

 Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.

## ADVERTISER INFORMATION

### Advertising Personnel

Marian Anderson: Sr. Advertising Coordinator  
Email: manderson@computer.org  
Phone: +1 714 816 2139 | Fax: +1 714 821 4010

Sandy Brown: Sr. Business Development Mgr.  
Email: sbrown@computer.org  
Phone: +1 714 816 2144 | Fax: +1 714 821 4010

### Advertising Sales Representatives (display)

Central, Northwest, Far East:  
Eric Kincaid  
Email: e.kincaid@computer.org  
Phone: +1 214 673 3742  
Fax: +1 888 886 8599

Northeast, Midwest, Europe, Middle East:  
Ann & David Schissler  
Email: a.schissler@computer.org, d.schissler@computer.org  
Phone: +1 508 394 4026  
Fax: +1 508 394 1707

Southwest, California:  
Mike Hughes  
Email: mikehughes@computer.org  
Phone: +1 805 529 6790

Southeast:  
Heather Buonadies  
Email: h.buonadies@computer.org  
Phone: +1 973 304 4123  
Fax: +1 973 585 7071

### Advertising Sales Representatives (Classified Line)

Heather Buonadies  
Email: h.buonadies@computer.org  
Phone: +1 973 304 4123  
Fax: +1 973 585 7071

### Advertising Sales Representatives (Jobs Board)

Heather Buonadies  
Email: h.buonadies@computer.org  
Phone: +1 973 304 4123  
Fax: +1 973 585 7071