Modeling the Impact of Silicon Photonics on Graph Analytics

Nathan R. Tallent, Kevin J. Barker, Daniel Chavarría-Miranda, Antonino Tumeo
Mahantesh Halappanavar, Andrés Márquez, Darren J. Kerbyson, Adolfy Hoisie
Pacific Northwest National Laboratory
{tallent, kevin.barker, daniel.chavarria, antonino.tumeo, hala, andres.marquez, darren.kerbyson, adolfy.hoisie}@pnnl.gov

***Abstract*—Silicon photonics is an emerging technology that delivers higher ratios of bandwidth to power than today's electrical interconnects. This paper explores whether graph-based analytics, increasingly important for high performance computing, can benefit from photonics' energy-efficient bandwidth. We select two contrasting photonically-enhanced systems projected for 2020. We sketch optical and electrical interconnect variants at different points along similar performance-to-power curves. We model applications and graphs that exhibit four distinct workload characteristics: compute-bound, bandwidth-bound all-to-alls, bandwidth-bound neighbor exchange, and latency-bound. We present quantitative results that project execution time and energy on large graphs (1 trillion edges). Our results show that for these workloads, interconnects with efficient optical interconnects can be over-provisioned if their bandwidth is too high. However, interconnects with similar efficiency but lower power present an opportunity for energy savings. We also show that even though optical interconnects do not improve on electrical link latencies, they can substantially increase the performance of latency-bound applications.**

I. Introduction

Graph-based applications — motivated by multiple-domain sciences, national security, and business analytics — are a major emerging application area for high performance computing. The graphs that these applications use can be so large that they must be distributed across multiple memories.

This paper considers graphs that must be mapped to clusters based on aggregate memory capacity rather than computational requirements. Unfortunately, graph applications challenge cluster architectures because they are dominated by irregular and indirect data accesses; and because they have unstructured, though abundant, parallelism. The result is poor performance and energy efficiency. Clearly, graph applications would benefit from improvements in network latency and concurrency. Unfortunately, latency is improving slowly as it nears physical limits; and concurrency, strongly related to topology, is often limited by cost or power. An alternative is to aggregate remote data accesses (reducing parallelism) so that they can be transferred more efficiently on interconnects.

Silicon photonics is an emerging technology that uses optics to deliver higher ratios of bandwidth to power than today's electrical interconnects [1], [2]. Large-scale systems using silicon photonics have recently been announced [3]. This paper uses modeling to explore the impact of silicon photonics networks on applications processing very large graphs.

We consider two photonically-enhanced systems (§II) projected for the 2020 time frame, IBM's TOPS [1] and Oracle's Macrochip [2], [4], [5] systems. The IBM and Oracle architectures represent systems with 'standard' and 'fat' nodes, respectively. Each system has a per-node bandwidth of 8-10 TB/s for 6-8 KW, yielding bandwidth-to-power ratios of more than 30× HDR InfiniBand. We sketch two optical and two electrical variants for each system (for a total of 10 variants) at different points along similar performance-to-power curves. The cited optical systems represent points with high performance; the variants reduce power consumption. The electrical variants compare optical and electrical interconnects by fixing either the power budget or rack count (system size). Electrical systems use HDR InfiniBand [6], scheduled for 2017–18. We briefly consider NDR InfiniBand, projected for post 2020.

We select two applications that, combined with graph topologies, exhibit four distinct workload characteristics: compute-bound, bandwidth-bound all-to-alls, bandwidth-bound neighbor exchange, and latency-bound (§III). Community Detection (graph clustering) creates large messages by packing several remote requests and then floods the network using personalized all-to-alls. Depending on compute and network resources, the workload can be either compute-bound or bandwidth-bound. The second application, Half-Approximate Weighted Matching, inherently performs less work per graph vertex. We select a graph topology that yields two distinct workload phases: bandwidth-bound neighbor exchange and latency-bound point-to-point. The four workloads represent common graph operations.

We model (§IV) each application and graph input. We consider graphs with 1 trillion edges (scale 40) that require a rack's worth of memory. We vary model parameters such as interconnect bandwidth, network power, node concurrency, link concurrency, and graph density.

Our models show (§V) that silicon photonics interconnects can be over-provisioned with bandwidth. However, interconnects with similar efficiency but lower power can deliver time and energy savings. Furthermore, even though optical interconnects do not improve link latencies, they can substantially increase the performance of latency-bound applications. This is the first study considering these contrasting optical and electrical systems at different ratios of performance-to-power, a broad class of graph workloads, and very large graphs.

978-1-5090-3315-7/16/$31.00 ©2016 IEEE

	IBM TOPS (64 nodes)					Oracle Macrochip (32 nodes)				
	Optical			Electrical HDR/NDR		Optical			Electrical HDR/NDR	
	reference	scale nw	fix bw/W	fix size	fix power	reference	scale nw	fix bw/W	fix size	fix power
topology	fully connected	fully connected	fully connected	fat tree	fat tree 5×-plane	fully connected (fully)	fully connected (fully)	fully connected (fully)	fat tree (2d mesh)	fat tree 12×-plane (2d mesh)
BW/link, GB/s	40	40	40	25/50	125/250	16 (2)	16 (2)	16 (2)	25/50 (26)	300/600 (26)
lanes/link	16	16	16	4	4	8 (1)	8 (1)	8 (1)	4 (18)	4 (18)
link/node	256	64	64	1	5	512 (1)	64 (1)	64 (1)	1 (4)	12 (4)
BW/node, GB/s	10,240	2,560	2,560	25/50	125/250	8,192	1,024	1,024	25/50	300/600
latency, μs	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
power, W	6,080	2,240	1,520	1,160	5,800	8,400	1,970	1,050	680	8,160

Fig. 1. Network parameters for optical and electrical system variants. (Parentheses denote intranode.)

II. Strawmen Architectures

We consider two different reference architectures and sketch two optical and two electrical variants for each system. Figure 1 summarizes the key parameters for each of the 10 system variants. The two different *reference* architectures represent contrasting approaches to the integration of photonics into rack-scale computing systems, IBM's TOPS [1] and Oracle's Macrochip [2]. The subsections below first describe each reference system and then explain their variants. The variants only change network (not compute) parameters.

A. IBM TOPS

IBM Throughput Optimized POEM System (TOPS) is a midsize (64-node) system whose nodes are interconnected using high-radix wavelength-division-multiplexed (WDM) switch planes. For a 64-node system, IBM envisions 256 switch planes, each a 64-way crossbar. There is one fiber from each node to each switch plane. Each fiber contains 16 wavelength channels operating at 2.5 GB/s, resulting in 40 GB/s per switch-plane port. Each node has more than 10 TB/s aggregate bandwidth over the 256 planes. Using 4 planes, each node-node pair has an aggregate bandwidth of 160 GB/s and nominal latency of 0.5 μs. The large number of optical switch planes enables significant path diversity, providing one-to-all broadcast with partitioned bandwidth, full bandwidth connections between pairs of nodes, and flexible congestion control with optimized switch scheduling and allocation algorithms.

For TOPS, on a timescale of 2019–22, IBM anticipates nodes with 8 TB of memory, 8 Tera-ops (over 4 sockets), and a memory bandwidth of 1 B/op. The assumption is that aggressive technology scaling will allow nodes to have sufficient memory capacity and bandwidth so that they are still network bandwidth bound. To reach the off-chip bandwidth of 10 TB/s, the hub chip inside each node will require new solutions for hardware acceleration, pipelining of memory requests, and message coalescing. IBM proposes a vision [1] for the required hardware developments, including high photonic integration density coupled with advanced 3D-IC packaging, effective switching technology with nanosecond-level reconfiguration times, optical amplification to compensate for switch losses, hybrid packaging approaches, burst-mode and WDM-capable optical transceivers, and efficient pipeline schedulers. Validating whether these hardware solutions are feasible in the projected time frames is out of scope.

		reference (scale nw)
IBM TOPS	switch plane	20 W
	hub chip (per node)	15 W
Oracle Macrochip	network (per node)	66 (37) W
	I/O ports (per node)	197 (25) W
HDR/NDR InfiniBand	switch	200 W
	HCA (per node)	15 W

Fig. 2. Power consumption of network components.

B. Oracle Macrochip System

Oracle's design [2] also targets a rack-scale system, but focuses on 'fat' nodes. To implement a compute node, Oracle proposes a multi-chip platform, called a Macrochip. A Macrochip uses a silicon substrate with embedded silicon photonic links to interconnect up to 64 sites bonded to the top of the substrate. In this study, we consider an 8×8 mesh design where each site has a compute and memory component. The memory is stacked on top of the compute component and communicates electrically with it. Each site has a bridge that converts electrical signals to optical signals. The Macrochip's optical communication layer provides all-to-all connection between the sites, configured as a planar point-to-point network. The proposed design targets a 2020 time frame and assumes aggressive technology scaling and exploitation of technologies such as 3-D stacking.

A full system includes 32 Macrochip nodes housed into 2 racks. Each node is projected to provide 32 TFlops (512 GFlops/site). Each site's compute component has 1 B/Flop of memory bandwidth; the bridge to the optical interconnect provides 0.5 B/Flop of network bandwidth. The total network I/O bandwidth per site is thus 256 GB/s (unidirectional). Half of the bandwidth is used for internal communication in the Macrochip; half for external communication with the other Macrochips. Within a Macrochip, there is one wavelength of 2 GB/s from every site to every other site (64 wavelengths, comprised of 8 waveguides with 8 wavelengths). For external communication, each site has 128 GB/s to another Macrochip. This consists of 8 fibers/site (for each site, another set of 8 waveguides with 8 wavelengths), resulting in 512 exiting fibers (and a corresponding 512 entering). To create a fully connected inter-node network, each Macrochip has 32 I/O

ports partitioning the 512 exiting fibers. Since two sites connect to each I/O port, there is 256 GB/s unidirectional bandwidth between Macrochip pairs. Communicating between any two sites in the system requires a 3-hop routing algorithm from (a) the source site, to (b) the source Macrochip site linked to the target Macrochip, and finally to (c) the target site. We assume a nominal node-node latency of 0.5 μs.

C. Optical Variants

Each *reference* system has very high performance, e.g., an inter-node *per-node* bandwidth of 8-10 TB/s. We introduce two optical variants that reduce power consumption but that maintain similar ratios of bandwidth to power.

The *scale nw* variants conservatively scale down each interconnect but without removing certain fixed costs and without changing topology. The IBM variant uses 64 switch planes, the minimum needed for a fully connected network. Thus, the bandwidth per node is quartered. Although the interconnect power is reduced, it is not quartered. Figure 2, which lists the power of key interconnect components, shows that each hub chip has a fixed power draw. The Oracle variant removes seven of the eight fibers emerging from each site for the inter-node interconnect. Again, the topology remains unchanged. Figure 2 shows how the power is scaled when each fiber is an equal contribution to overall power.

The *fix bw/W* variant begins with the *scale nw* system (i.e., same performance) but optimistically scales down the power. Specifically, this variant maintains a constant ratio of bandwidth/node and network power. Thus for the IBM and Oracle variants, both bandwidth per node and interconnect power is quartered and one-eigthed, respectively.[1]

D. Electrical Variants

We introduce two electrical variants for each reference system. The default variants are based on HDR InfiniBand [6]. We briefly consider NDR InfiniBand, HDR's successor. Based on past performance, we assume an NDR 4× link has twice the HDR bandwidth and the same nominal latency. We (optimistically) assume comparable power consumption for HDR and NDR components.

The *fixed size* variant maintains rack count. This variant replaces each *inter-node* interconnect with an HDR InfiniBand-based non-blocking fat tree. Projecting to the 2020 time frame, we assume a single 64-port HDR switch, where each switch port accepts a 4× HDR link. Each node uses a 4× HDR link of 25 GB/s and 0.5 μs latency. Assuming the switch and HCA power draws in Figure 2, the power for the electrical network is 1/5 to 1/12 its optical counterparts, respectively. However, the connectivity and bandwidth/node are inferior. Observe that the HDR bandwidth-to-power ratios (≈ 0.03 GB/s/W) are much less than for the optical systems (≈ 1 GB/s/W). The NDR bandwidth-to-power ratios double the HDR numbers.

[1]This Oracle fix bw/W result does not separate inter- and intra-node power. Although imprecise, it is desirable to maintain comparable bandwidth-to-power ratios with TOPS, as each system represents different architectural approaches to for assigning similar bandwidth-to-power resources.

The *fixed power* variant equalizes network power by adding additional HDR switches and node adapters to create 5× and 12× multiplane fat trees, respectively. We calculate network power consumption by summing the power consumed by the network switch, link, and interface components. Although the fixed-power variant increases the bandwidth per link substantially, the bandwidth-to-power ratio is constant and the connectivity is still inferior.

For the Macrochip variants, we model the *on-node* electrical interconnect using a 2D mesh based on the Intel Quick Path Interconnect (QPI), where each mesh link is comprised of 18 lanes and has a total bandwidth of 25.6 GB/s.

III. Application Workloads

We select two applications that, combined with graph topologies, exhibit four distinct workload characteristics. Community Detection represents a large class of applications that, for each vertex in parallel, perform an operation along all edges. In contrast, Half-Approximate Weighted Matching represents applications that (for each vertex) perform an operation along *one* edge, resulting in less vertex-local work. We use distributed parallel versions of each application to process large graphs ranging from scale 32–40, where scale corresponds to the logarithm, base 2, of the number of edges in the graph. To represent a production environment, we introduce 'cluster-friendly' optimizations that batch local work and communication. We select graph topologies that create four distinct workloads: compute-bound, bandwidth-bound all-to-alls, bandwidth-bound neighbor exchange, and latency-bound. The four workloads represent common graph operations.

A. Community Detection

Consider a weighted graph $G = (V, E)$ with vertex set V, edge set E, and a weight function $w : E \rightarrow \mathbf{R}$ that associates real-valued non-negative weights to edges. The process of grouping vertices into disjoint sets such that vertices in the same set are highly correlated between themselves, but sparsely correlated with vertices in other sets, is called graph clustering or community detection. Although community detection is an NP-hard problem, several fast and efficient heuristics exist. One such heuristic is the Louvain method [7], which is based on iteratively optimizing a metric called *modularity* [8]. In each iteration, the algorithm processes each vertex and makes a greedy decision, based on gain in modularity, whether the vertex should remain in its current community, or migrate to the community of one of its neighbors.

We created a distributed memory implementation (using the Message Passing Interface) that leverages the Grappolo shared-memory version [9]. The result is a hybrid MPI-OpenMP program. We distribute the graph using a block-oriented 1-D vertex partitioning, where each compute node gets a contiguous block of vertices. For a graph with $n = |V|$ vertices and $m = |E|$ edges, each compute node (out of p) is assigned (at most) $\frac{n}{p}$ contiguous vertices.

Our implementation of the distributed Louvain phase is summarized in Figure 3. There are five communication steps.

```
1  DistributedLouvainPhase(G_i = (V_i, E_i), C)
2    exchangeVertexRequests()                          // all-to-all
3    while Δmodularity > threshold do
4      map' ← recvRemoteEdgeInfo()                     // all-to-all
5      map ← fillRemoteCommunityMap(map')              // all-to-all
6      parallel foreach v ∈ V_i do
7        Louvain iteration
8      newModularity ← globalModularity(map)
9      Δmodularity ← newModularity − oldModularity
10     updateLocalCommunities(map)
11     updateRemoteCommunities(map)                    // all-to-all
```

Fig. 3. Distributed Louvain; node i owns subgraph G_i.

```
1  DistributedMatching(G_i = (V_i, E_i), M)
   // Phase 1: Aggregated work
2  For each vertex, identify locally dominant cross-edge
3  Try matching with aggregated (per-node) request
   // Phase 2: Fine-grain work
4  while a vertex can be matched do
5    Receive message and process based on type
6    Try matching along locally dominant cross-edge
```

Fig. 4. Distributed Matching; node i owns subgraph G_i.

First, line 2 (executed once) prepares for several Louvain iterations. Each node determines edges crossing its inter-node boundaries and the nodes owning the target of each edge. This information is stored and used to aggregate remote data requests. Second, line 4 communicates (via personalized all-to-all) the community identity of remote target vertices to the owner of the source vertex. Third, line 5 determines which community identities are not *local* to the compute node (ownership of communities exactly mirrors ownership of vertices) and requests the owner of those communities to send the *size* and *degree* of those communities to the local compute node (two personalized all-to-alls). Fourth, line 8 computes global modularity using an allreduce collective. Fifth, line 11 writes back the increment (or decrement) in community size and degree for any non-local communities (personalized all-to-all).

B. Half-Approximate Weighted Matching

Consider a weighted graph $G = (V, E)$ with a weight function $w : E \rightarrow \mathbf{R}^+$ that associates real-valued positive weights to edges. A matching M is a subset of edges such that no two edges in M are incident on the same vertex. The weight of a matching M is the sum of weights of the matched edges. A maximum weighted matching is a matching of maximum weight among all possible matchings. Half-approximate weighted matching (Matching) computes a solution within a factor of half the optimal value. We use the locally dominant algorithm [10], [11]; cf. [12].

An edge (v, w) is *locally dominant* if it is heavier than all edges incident on its endpoints v and w. One way to identify locally dominant edges is to sort edges by weight. However, such an order serializes the execution. To introduce parallelism, this algorithm does the following. For each vertex v, finds its heaviest neighbor w and sets a pointer from v to w. If two neighbors point to each other, then the corresponding edge is locally dominant. Such an edge is added to the matching set and removed from the graph. The algorithm iterates until there are no more matched edges.

In a distributed implementation (Figure 4), each compute node gets a subgraph G_i using a 1-D or 2-D vertex partitioning. Communication occurs along cross edges (edges with a remote vertex). The algorithm requires 2–3 messages for each vertex. Since *the number cannot be known in advance*, the execution is separated into two phases. Phase 1 (lines 2–3) aggregates work and communication (since each vertex will exchange at least 2 messages). Phase 2 (line 4) uses an asynchronous message engine to complete the protocol, exchanging 1–2 messages per vertex, in no particular order. Each message is very small (a 3-tuple). The phase terminates when matching is complete.

C. Summary

For Community Detection, we select input graphs so that most vertices on a node have cross-edges to all other nodes. This generates communication dominated by bandwidth-bound personalized all-to-all messages that flood interconnects. If Community Detection is *also* network-bound (instead of compute-bound), it makes the best usage of silicon photonics' high bandwidth-to-power ratio. Our models explore both the compute-bound and network-bound cases. We make the workload network-bound by (artificially) using perfect thread scaling. In reality, Community-Detection-style applications often require significant compute time, resulting in varying ratios of compute to network time. Consequently, our final results do not assume perfect thread scaling. For silicon photonics to make sense, its must yield an overall energy win even when the network cannot be fully utilized.

For Matching, we select an input graph so that each node's subgraph has a sparse neighbor set. As a result, Matching's two phases represent two new workloads: bandwidth-bound neighbor exchange and latency-bound where there are far fewer opportunities for batching local work and communication. Although Phase 1 is network-bound (requiring limited compute time), each node much exchange bandwidth-bound messages with at most 4 *neighbor* nodes. Phase 2 is is dominated by small asynchronous messages that stress a network's point-to-point concurrency and latency.

IV. Analytical Models

To predict the impact of silicon photonics technologies on analytics workloads, we developed analytical models of each application. Our models account for the first-order effects of computation and communication. For communication, we use LogGP-style [13] models that account for communication patterns and contention. We account for various interconnect inefficiencies using latency and bandwidth de-rating factors extrapolated from current electrical networks.

We validated Community Detection up to scale 35 (logarithmic in edges) on an FDR InfiniBand cluster (one switch, 4× links) with two 10-core Intel Xeon processors and 768 GB memory/node. We validated Matching up to scale 32 on our InfiniBand cluster. Given the properties of the selected input

graphs, we can accurately extrapolate the behavior for a scale 40 graph, a scale of interest for large-scale graph analytics. Section V gives predictions for scale 40 graphs.

A. Community Detection

Community Detection (Figure 3) takes as input a weighted graph $G = (V, E)$ and produces a set of communities C that partitions the vertices. The input graph has $n = |V|$ vertices and $m = |E|$ directed edges. (An undirected edge is a pair of directed edges.) The result is a set of $c = |C|$ communities. A vertex, edge, and community consume β_v, β_e, and β_c bytes of data, respectively. The application executes on p nodes.

1) Input graphs: We considered both uniform graphs and power law graphs. Although power law graphs are more representative topologically, a uniform graph eases modeling without sacrificing accuracy. That is, given a power-law graph, replicate (local) graph density along the critical path to create a corresponding uniform graph. This is reasonable because each iteration synchronizes; and message sizes are not proportional to remote edges (m) but to *unique* remote edge targets and their *unique* communities, both of which are capped by n.

We use a scalable random graph generator that can produce graphs of a given density and scale. Because graph partitioners are costly for such large graphs, we assume one is not used.

2) Observations: We exploit properties of uniform random graphs to simplify the model. First, a graph vertex has average out-degree $d = \frac{m}{n}$. Because every out-edge is an in-edge at the target, out-degree equals in-degree. Because of the uniform distribution of degrees, average out-degree for a node's vertices is also d. Second, we can characterize the probability π_e that a vertex has an edge whose target is remote. Since an edge's target will be resolved by any vertex (and node) with equal probability, we have $\pi_e = (p-1)/p$.

3) Model: The application is modeled by a sequence of iterations where the time for one iteration is dictated by lines 4, 5, 7, 8, and 11 of Figure 3.

Receive remote edge community ids (line 4) An iteration begins by receiving community ids for remote edge targets. A node has $\frac{n}{p}$ vertices of degree d, yeilding $\pi_e(\frac{n}{p}d)$ remote edges. The remote edges have $\hat{n}$ *unique* targets. When, $d \geq p$, $\hat{n} \approx \frac{n}{p}(p-1)$. Target community ids are received using a personalized all-to-all: $p-1$ messages, each of similar size. Since ids use β_v bytes, each message is $(\hat{n}\beta_v)/(p-1)$ bytes.

Fill remote community map (line 5) Because communities change, each iteration sends a request and response to gather community metadata from each community's head vertex. Each request and response is a personalized all-to-all. Requests and responses are composed of vertices and communities, respectively, each of size β_v and β_c bytes.

The size of each message is proportional to $\hat{c}$, the number of (unique) remote communities incident to a node's remote edges. The value of $\hat{c}$ is dependent on the rate at which communities are formed. We say that $\hat{c} = \pi_c\hat{n}$, where π_c is the probability that a vertex has an edge incident to a (unique) remote community. The value of π_c changes as the computation proceeds. At the beginning each vertex is its own community, so initially $\pi_c = 1$. Since the π_c probability distribution is similar for graphs of same structure (independent of scale), we model it empirically. Accordingly, each request has volume $(\hat{c}\beta_v)/(p-1)$ bytes; and each response has volume $(\hat{c}\beta_c)/(p-1)$ bytes.

Process local vertices (line 7) The amount of work per node is proportional to the node's edges, $\frac{n}{p}d$. The work per edge ω_e is modeled by a count of indirect memory accesses and an empirical value for processing a community. Because all edges on a node can be processed independently, with large graphs there is sufficient parallelism for many threads.

Global modularity (line 8) Model as two allreduces.

Update remote communities (line 11) The final step is to update each *changed* remote community by providing new community metadata to its head vertex (personalized all-to-all). Each message has elements of size β_c bytes. Let $\pi_{c'}$ be the probability that remote communities change. Since the $\pi_{c'}$ probability distribution is similar for graphs of same structure (independent of scale), we model it empirically.

B. Half-Approximate Weighted Matching

Matching (Figure 4) takes a weighted graph $G = (V, E)$ and produces a set of matched edges M that pairs the vertices. The input graph has $n = |V|$ vertices and $m = |E|$ directed edges. (An undirected edge is a pair of directed edges.) The remote matching protocol is based on receiving a tuple of size β bytes. The application executes on p nodes.

1) Input graphs: To generate two new distinct workloads, we select a two-dimensional mesh with randomly weighted edges. The mesh input graph is distributed across nodes with a 2D block distribution. Thus, each node's subgraph has both low degree and a sparse neighbor set. Matching's two phases result in the following two respective workloads: bandwidth-bound neighbor exchange and latency-bound. As an additional benefit, this topology ensures stable scaling behavior.

2) Observations: We exploit properties of the input graph to simplify the model. First, an interior vertex has average out-degree $d = 4$. Second, each node has $\frac{n}{p}$ vertices. Third, the vertices mapped to a compute node form a mesh, where an interior node has four boundaries. We model the number of vertices z on one boundary. Assume for simplicity that p is a perfect square. Then, each node has $z^2 = \frac{n}{p}$ vertices.

3) Model: The analytical model has two pieces, corresponding to the two phases of Figure 4. In the first phase, remote matching requests for each boundary are aggregated. In the second phase, individual requests are sent.

Phase 1 (line 3) Phase 1 aggregates all match requests for a boundary (node). Because of the protocol, each boundary exchanges three messages in sequence. The volume of each message is βz bytes.

Phase 2 (line 4) We view Phase 2 as a series of *rounds*, where a round processes all vertices unmatched at the round's beginning. Thus, a round depends on the number of remaining

locally dominant cross edges. Characterizing remaining cross edges analytically is difficult because matches are not independent. Extensive empirical results for random mesh graphs [14] show that about $\frac{1}{2}$ of the vertices are matched per round. Hence, $\frac{1}{2}$ of the z vertices on a boundary are matched and $\frac{1}{2}$ remain. Although each vertex's request can proceed in parallel, the rate is limited by network link concurrency.

V. Analysis and Results

This section predicts the total performance and network energy of Community Detection and Matching on the IBM TOPS and Oracle Macrochip system variants of Figure 1. Since the optical system variants have strictly superior interconnects, we generally expect a performance advantage. The primary question, therefore, is whether photonics' performance-to-power ratio also improves energy consumption for graph workloads that either are not network bound (Community Detection) or do not use interconnect bandwidth efficiently (Matching).

This analysis focuses on *network* energy because we assume the electrical and optical system variants have similar compute and memory components. To present a conservative estimate, we assume the network is fully powered for the duration of an execution. Therefore, network energy is the product of total execution time and the network power draw listed in Figure 1.

To represent rack-scale graphs, we use use scale 40 input graphs that require order $2^{40}\beta_e$ bytes, where β_e represents bytes/edge. Recall that the input graphs are described in Section IV. A simple weighted edge has $\beta_e = 16$, giving 16 TB. With nodes of order 1 TB memory, each graph and some metadata can be distributed across either system. Because the Macrochip-based system has half the nodes, twice as many vertices and edges are mapped to each node.

A. Community Detection

Recall that Community Detection exhibits compute-bound and bandwidth-bound personalized all-to-all workloads. (A personalized all-to-all sends distinct data to all other $p' = p-1$ nodes.) To show workloads that have a decreasing ratio of compute-to-network time, we vary concurrency per node, expressed as hardware thread count, where processor cores use multiple hardware threads to hide memory latency. To show these effects clearly, we assume *ideal scaling* with respect to hardware threads — an assumption that is revisited below.

Figure 5 shows Community Detection's absolute and relative execution time on a scale 40 uniform graph. Figure 6 shows relative energy. The figures are discussed in more detail below. We use a graph with an average degree of 160, which corresponds to moderate connectivity. Each aggregated per-node message is a few gigabytes in size.

Each of Figure 5's charts varies node thread count between 128–4096. Since TOPS variants have four sockets/node, 4096 threads/node corresponds to 1024 threads/socket. For Macrochip variants, these threads are distributed over the 64 compute sites. Note that the execution time charts omit 'Optical: fix bw/W' variants because their times are identical to 'Optical: scale nw'.

1) Execution Time: Figure 5(b) shows relative execution time of the photonically-enhanced variants relative to an electrical HDR counterpart. (Again, 'Optical: scale nw' and 'Optical: fix bw/W' have the same times.) As expected, the execution time of the optical variants is better (or nearly the same) as the electrical variants.

Observe that the predicted speedups are much smaller than the maximum speedups possible. To see this, consider that the maximum optical speedup for personalized all-to-all is essentially the ratio of per-node bandwidths for the optical and electrical interconnects. For example, the maximum speedup of the TOPS reference system is 400× and 80× over the corresponding fixed-size and power electrical variants. In contrast, the best relative execution time of the TOPS system is 25% of the electrical fixed size system, or a speedup of 4×.

The reason for the difference between maximum and actual speedups is that, even with ideal thread scaling, it is hard to make the full application network bound. Community Detection, as with many applications that on each step inspect each edge of each vertex, has a substantial amount of on-node memory-bound work. Thus, although each personalized all-to-all can fully exploit the optical networks' bandwidth and topology, the aggregated message itself depends on many small independent tasks where the network is not utilized. One way to reduce the ratio of compute to network time would be to send substantial amounts of metadata with each request. However, that is not typical of this workload.

Figure 5(b) shows the importance of reducing the compute-to-network ratio. At at 128 threads, the best time relative to an electrical fixed size variant is 90%; and the best relative to an electrical fixed power variant is 97%. At 4096 threads, the best times are 25% and 40%, respectively.

Of course, the assumption of ideal scaling requires sufficient on-node memory bandwidth. Each independent task performs several irregular memory accesses, making it memory bound. Without sufficient memory bandwidth, threads will stall waiting for memory operations to complete. The TOPS and Macrochip based systems are projected to provide 1 B/Flop of peak memory bandwidth or 8–32 TB/s peak. Based on experiments [15], [16], we estimate that irregular memory accesses can utilize 10% of this bandwidth. Assume 20% [17] of a 1 GHz thread's operations are memory operations. Then scaling plateaus around 500–2000 threads. At 2048 threads, the time benefits of the TOPS and Macrochip systems are at most 2.5× (40%) and 1.8× (55%), respectively, the electrical fixed size.

There are two reasons for the differences between the TOPS and Macrochip systems. First, the TOPS system has twice the node count. Because the graph is distributed over more nodes, messages sizes are smaller. Second, the Macrochip's optical network has less effective per-node bandwidth because of the intranode bottleneck in the three-hop routing algorithm (§II-B). For all-to-alls, the routing algorithm experiences an intranode bottleneck within the Macrochip's site-to-site links. The bottleneck could be overcome by dedicating more bandwidth to the intranode network.

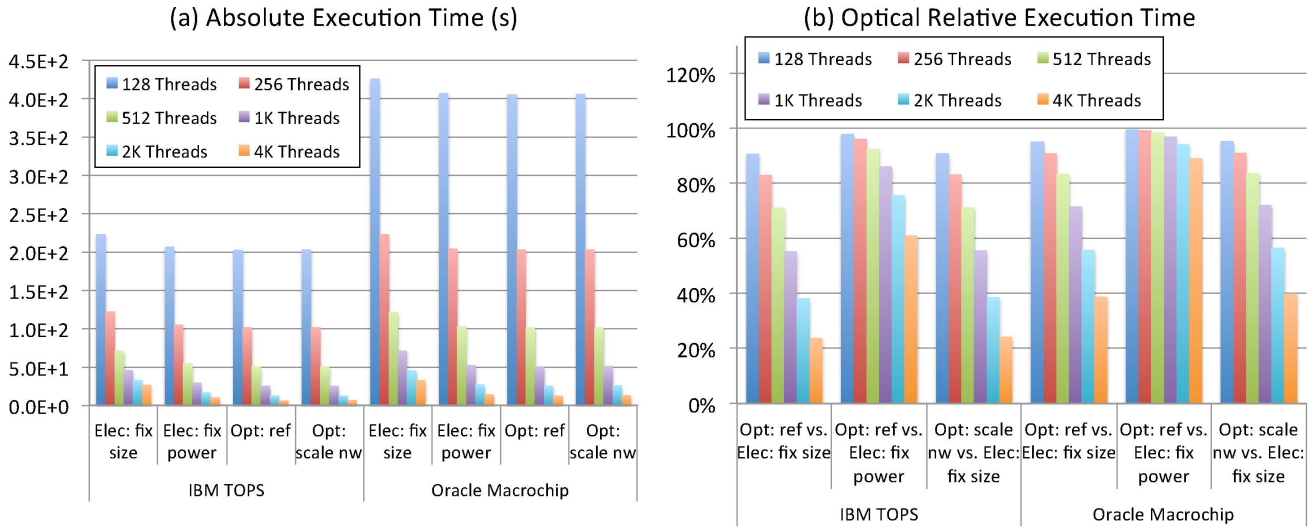

Fig. 5. Community Detection/Scale 40 execution times for 128–4096 hardware threads/node.

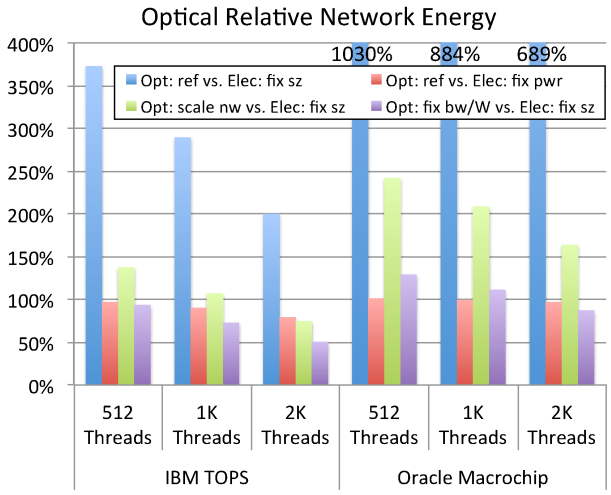

Fig. 6. Community Detection/Scale 40 network energy.

If we keep the scale of the graph (edges) constant and increase the average degree, the benefits of the optical network decrease. Suppose we increase the average degree of the graph by a factor of 4×, meaning that the number of vertices decreases by 4×. Since the number of communities is related to the number of vertices, message sizes decrease by roughly 4×. The per-node work, which is proportional to edges, does not change. Thus, the ratio of compute to network time increases, creating less opportunity to exploit optical interconnects.

2) Energy: For a bandwidth-bound *and* network-bound workload, there are obvious benefits of photonics' performance-to-power ratio. As highlighted above, the key question is whether photonics' also improves energy consumption on this realistic graph workload. Figure 6 summarizes the potential improvement on network energy efficiency for expected effective concurrency levels. The figure assumes the network is fully powered for the duration of an execution. We make several observations. Again, the electrical variants are based on HDR InfiniBand.

First, for these workloads, the reference systems are probably over-provisioned. Most notably, the reference optical systems show significantly *degraded* energy relative to fixed size electrical variants. Even though the reference systems improve performance, their power consumption is much higher than the fixed size electrical variant. As a result, the reference system's power costs cannot be amortized over periods when the interconnect is under utilized. Observe that relative to the fixed power variants, reference systems show modest benefits. That is, when network power costs are equal, the optical networks can provide limited speedup.

Second, there is limited benefit of the optical scale nw over electrical fixed size. The reason is that the fixed power costs are large enough that the improved performance cannot be amortized during periods of compute-bound work.

Third, is the benefit of the optical fix bw/W over electrical fixed size. The reason is that power has reduced enough. Thus, for these workloads, photonics makes the most sense when it delivers modest bandwidth — which is still competitive with electrical bandwidths — for very low power.

3) Summary: Optical networks can improve the execution time of Community Detection's bandwidth-bound all-to-all communication. If the workload is highly compute-bound, the overall time speedup is minimal. However, for expected compute node performance, the total time speedup is about 2× the electrical fixed size. Although much smaller than the theoretical maximum, this is still a notable improvement. Expected speedups over the fixed-power variant are smaller.

Silicon photonics can deliver energy improvements when the interconnect focuses on low power rather than high performance. The optical TOPS and Macrochip fix bw/W variants deliver an improvement of 2× and 1.25× over electrical fixed-size (i.e., relative energy of 50% and 80%).

Community Detection illustrates a common 'compute-

bound' problem with graph applications that use dense bandwidth bound communication. One way to make the application network bound is to send substantial amounts of metadata with each request. However this is not typically necessary. Therefore, to ensure energy benefits, it is necessary to design an interconnect's performance relative to both the expected workload and node performance. In particular, it must be possible to amortize power draw during compute-bound periods.

Clearly, technology to temporarily power down the network during could be beneficial. However, such technology would likely have an electrical counterpart, so we ignore it.

B. Matching

Matching's two phases represent two distinct workloads: bandwidth-bound neighbor exchange and latency-bound messages. Unlike Community Detection, both phases are network-bound (and thus on-node concurrency is not an interesting parameter). Phase 1, like Community Detection, uses bandwidth-bound messages. However, each node exchanges messages messages with at most four nodes, all of which are logical neighbors. Phase 2 uses small asynchronous messages that stress a network's point-to-point concurrency and latency.

Figure 7 shows Matching's absolute and relative execution time on a scale 40 mesh graph. Figure 8(a) shows relative energy. To show characteristics of each workload, the above charts distinguish results for Phase 1 and Phase 2. Figure 8(b) compares the execution time of Phases 1 and 2. Note that the execution time charts omit 'Optical: fix bw/W' variants because their times are identical to 'Optical: scale nw'. The figures are discussed in more detail below.

1) Execution Time: Figure 7(b) shows relative execution time of the photonically-enhanced variants relative to an electrical HDR counterpart. (Again, 'Optical: scale nw' and 'Optical: fix bw/W' have the same times.) Even though Matching is network-bound, unlike Community Detection, the execution time of the optical variants is *not* always better. The results depend on both workload and system. All TOPS system variants have speedups for both of Matching's phases. Most Macrochip variants show speedup, but there can be degradation for Phase 1. Many of the speedups are substantial.

Phase 1. Phase 1 is network-bound. During the phase, each node exchanges bandwidth-bound messages with at most four neighbor nodes. Thus, compared to all-to-all, the neighbor exchange results in a comparatively sparse and inefficient use of interconnect links. For this phase, the speedup of the optical variants is approximately the ratio of optical and electrical bandwidths at each system's bottleneck.

For the TOPS optical variants, each node utilizes $\frac{1}{16}$th of the outgoing links, i.e., a ratio of 4 neighbors to 63. We assume it is unrealistic to reconfigure the switch planes to perfectly align the logical and physical topologies. Although many optical links are underutilized, the available per-node bandwidth for the optical system is still high: 640 GB/s and 160 GB/s for the reference and scale nw variants. As a result, the optical reference execution times are 4% and 20% the fixed-size and fixed-power electrical variants, translating to speedups of 25× and 5×. The optical scale nw has a speedup of 6×, using 16% the time of the electrical fixed-size.

For the Macrochip systems, the bottleneck shifts depending on system. For the optical variants, the intranode exchange is the bottleneck. The optical intranode (site-to-site) neighbor exchange uses only four (of 63) links to exchange data in cardinal directions. The total available bandwidth for one site face is 2 GB/s. It is conceivable to use an indirect routing scheme that exploits many more of the available intranode links. However, we assume this is not practical.

For the electrical variants, the inter-node exchange is the bottleneck, which must share the inter-node bandwidth among 32 site faces. The per-site-face bandwidth for the fixed-sized variant is $25/32$ GB/s, resulting in a 2.6× speedup. The per-site-face bandwidth for the fixed-power variant is $300/32$ GB/s, resulting in a notable slowdown.

Phase 2. Phase 2 is also network-bound. However, because the small messages during Phase 2 are latency-bound, there are two sources of inefficiency. First, as with Phase 1, links are utilized sparsely. Second, even when the small messages fully utilize a link's concurrency, the links bandwidth is not fully utilized. For this phase, the speedup of the optical variants is approximately the ratio of effective link concurrency between the optical and electrical variants.

The TOPS system again utilizes only $\frac{1}{16}$th (ratio of 4:63) of the outgoing links. However, there are 16 lanes (wavelengths) per link. Thus, when using 4 links, 64 messages may proceed concurrently compared to only 4 and 20 for the electrical fixed size and fixed power variants, respectively. The result is a speedup of 64× and 13× for the reference optical system over the electrial fix sized and fixed power.

For the Macrochip optical and electrical systems, the bottlenecks are again at the intra-node and inter-node exchanges, respectively. For the optical site-to-site exchange, each site uses links corresponding to communication in the cardinal directions. The result is a per site-face concurrency of 1 lane. Again, it is conceivable to use indirect routing (e.g., scatter-gather) to exploit the Macrochip's fully connected topology, but we consider it unlikely.

For the electrical variants, the bottleneck is the inter-node InfiniBand concurrency. Because InfiniBand HDR has 4 lanes per link, the per-node concurrency of the fixed-size and fixed-power variants are 4 and 48. The node concurrency is shared over 32 site faces, resulting in average concurrencies of $1/8$ and 1.5 per site face. As a result the speedup of the optical reference system over the electrical fixed size is 8×. In contrast, there is a degradation compared to the fixed power variant.

Total. Figure 8(b) shows that the performance of Matching is largely determined by Phase 2. As shown in the rightmost chart, Phase 2's small latency-bound messages dominate the time taken by Phase 1's larger aggregated messages. When the optical networks provide increased effective concurrency — via topology and link concurrency — the performance of Phase 2 substantially improves.

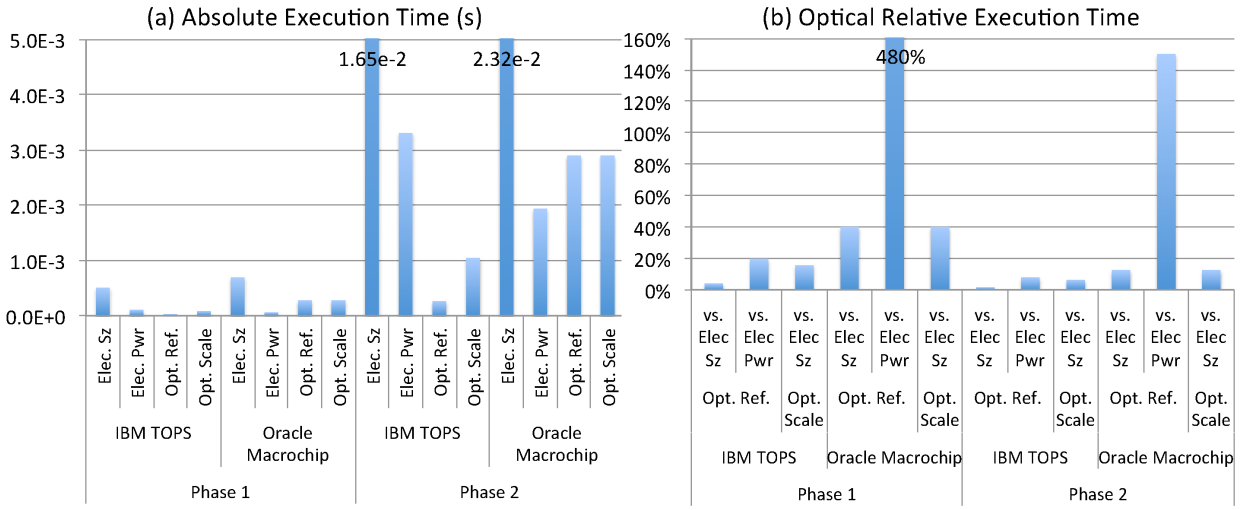

Fig. 7. Matching/Scale 40 execution times.

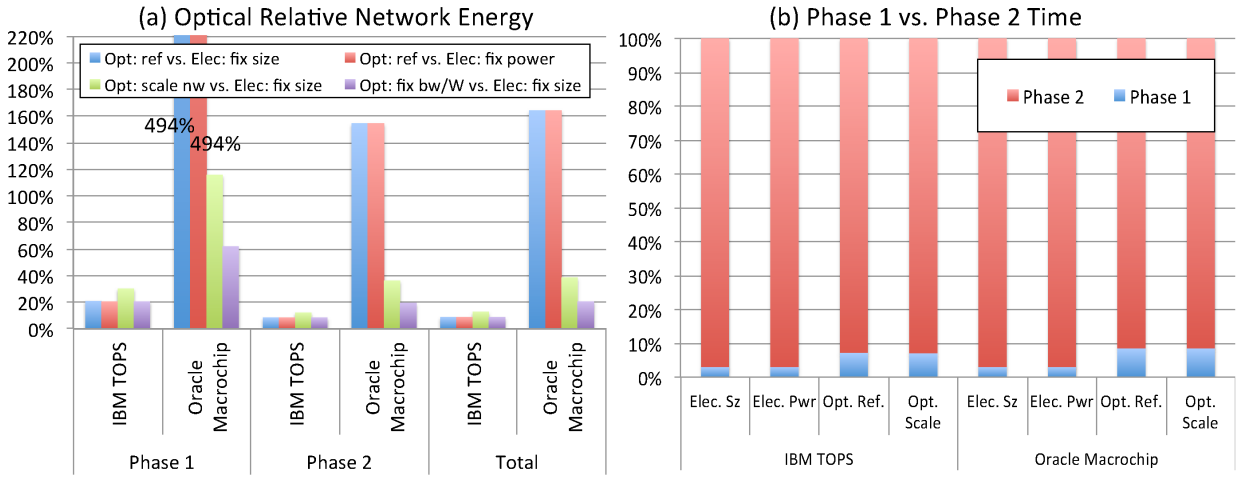

Fig. 8. Matching/Scale 40 (a) network energy and (b) ratio of Phase 1 to Phase 2.

2) Energy: Figure 8(a) summarizes the potential improvement of silicon photonics technology on network energy efficiency relative to HDR electrical systems. The results are broken down by Phase 1, Phase 2 and total time. In both cases, most total savings are due to the performance of Phase 2, as made clear by Figure 8(b).

Observe that *all* TOPS optical variants — even the reference systems — show substantial energy reduction relative to their electrical counterparts. Even though Matching's communication is sparse, the relative improved bandwidth and concurrency increases performance enough to offset higher power consumption. However, for Macrochip, only the down-scaled variants (scale nw and fix bw/W) show savings. The reason is that the Macrochip's intranode bottleneck leaves the inter-node network underutilized but drawing power.

3) Summary: Optical networks can improve the execution time and energy of Matching's bandwidth-bound neighbor exchange and latency-bound messages. Because Phase 2 dominates execution time, the interconnect's effective concurrency — a function of links/node and topology — is paramount. The TOPS systems, both reference and down-scaled variants offer benefits. Overall energy improvement is at least an order of magnitude; and time improvements can be larger. For the Macrochip systems, the intranode bottleneck means only the down-scaled optical variants offer both time and energy benefits.

Interestingly, although silicon photonics does not improve on electrical link latencies, the rich optical topologies means that latency-bound applications (like Phase 2) can achieve *more* speedup and energy savings than bandwidth-bound applications. Matching's Phase 1 results show that bandwidth-bound applications with sparse communication (as opposed to Community Detection's dense communication) may modestly benefit in both time and energy.

VI. Related Work

Modeling of Large-scale Systems: Optical networking technologies have been explored for use in large-scale high-

performance systems. Modeling techniques have been applied to scientific workloads to assess the performance potential of Optical Circuit-Switched technologies [18], [19] to construct interconnection network fabrics. The workloads we use to drive this analysis differ from this previous work in that graph-based applications exhibit very little computation for each graph vertex; in essence application performance is determined by communication and memory access performance. Modeling techniques have also been applied to analyze large-scale systems with multi-level fully-connected network topologies [20], [21]. The work presented here leverages this work but extends it through the use of graph-based workloads to study the impact of silicon-photonic technology.

Application modeling for irregular & graph workloads: Zhang *et al.* [22] develop a performance modeling methodology for Pig programs executing on multi-node cluster. Their focus is on modeling the execution of collections of MapReduce phases that compose a Pig workflow. They are able to model sequences of phases as well as DAGs. The methodology is applied to workloads derived from Internet data corresponding to web proxies and transactional-like input sets. Their models operate at the coarser levels of granularity needed to accurately model I/O-based workloads such as MapReduce.

Gharaibeh *et al.* [23] develop a performance model for graph kernels and a software framework to map graph applications to heterogeneous CPU-GPU nodes. Their model focuses on how to partition the graph between the GPU and CPU sides in order to minimize data movement and time-to-completion, while addressing the different parallelism granularities and particulars of each platform. However, their work is limited to a single node and a BSP execution style.

Other graph architectures: The Cray XMT system [17] was designed to tolerate latency at full distributed system scale by using highly concurrent multithreaded processors. Each Threadstorm processor provided 128 hardware thread contexts with the capacity of switching context on every clock cycle at 500 MHz. This processor design coupled with Cray's Seastar network enabled effective processing of highly irregular applications at system scale (128 nodes) with a significant fraction of the exposed latency hidden by the system's capabilities. In contrast to its Cray MTA-2 predecessor system [24], [25], exposed latency was not fully covered due to the mismatch between the level of hardware threading and the XMT's network capabilities. A study by Secchi *et al.* [26] proposes solutions to increase throughput and network utilization via increased threading and a multicore design.

VII. Conclusions

This paper presents an analytical and quantitative performance evaluation of the impact of silicon photonics networks on very large graph-based workloads. We selected a 'standard'-node and 'fat'-node system utilizing different optical interconnects projected for the 2020 time frame. We sketched optical and electrical variants of each system to represent different points along similar performance to power curves. We selected application and input graphs to represent four distinct workloads: compute-bound, bandwidth-bound all-to-alls, bandwidth-bound neighbor exchange, and latency-bound. We analyzed the results from the perspective of both performance and energy efficiency. Our conclusions are as follows.

First, although silicon photonics has bandwidth-to-power ratios of more than 30× HDR InfiniBand, it is not always easy to exploit high bandwidths. Our results show that for these applications and workloads, interconnects with efficient optical interconnects can be over-provisioned if their bandwidth is too high. That is, Community Detection and Matching not only illustrate four common workloads, but common application tensions. Matching is network-bound — well suited for silicon photonics — but its communication is sparse and latency-bound. Community Detection has bandwidth-bound and dense communication — well suited for silicon photonics — but it is difficult to make the application network-bound. Furthermore, in this latter case, the high bandwidth of optical systems can convert a modestly network-bound workload into a compute-bound one.

Second, it is critical that systems have balanced compute and network resources, relative to workloads. More specifically, interconnects with similar efficiency but lower power present a substantial opportunity for network energy savings. (Total system energy savings must consider node power.) This conclusion is confirmed by the consistent benefit of the optical fix bw/W variants over the electrical counterparts. In fact, our results show the optical networks could be downscaled in two ways. Community Detection shows it is possible to downscale the aggregate bandwidth (links) per node and still achieve benefits. Matching (and especially the Macrochip's intra-node bottleneck) shows it is possible to downscale topology (connections between nodes) and still achieve benefits. (Community Detection also supports this observation.) This latter observation is especially pertinant because fully connected networks are not scalable. Thus, if a technology like optical circuit switching can be implemented well, there is still reason to expect benefits.

Third, even though optical interconnects do not improve the latencies of electrical links, Matching's results show photonics can improve the time and energy of latency-bound applications. In this case, network energy savings can be an order of magnitude. Optical interconnects improve Matching because both lanes/link and topology (connectivity) enable concurrent messages. Assuming that fully connected topologies will be rare, the lanes per link parameter is important.

Finally, considering the potential of NDR InfiniBand reinforces the conclusions above. Assume NDR InfiniBand doubles HDR's performance for similar power draw. For Community Detection, the benefits of the TOPS downscaled optical systems degrade. Using Figure 6 as a reference point, the TOPS optical fix bw/W variant has relative network energy of 110%, 94%, and 74% the NDR electrical fix size. Thus, the 512 thread variant no longer has a benefit over NDR. Moreover, none of the figure's Macrochip variants would benefit over NDR. For Matching, whose overall results are

dominated by Phase 2's small messages, NDR offers very little benefit because the lanes/link remains the same.

REFERENCES

[1] L. Schares, B. Lee, F. Checconi, R. Budd, A. Rylyakov, N. Dupuis, F. Petrini, C. Schow, P. Fuentes, O. Mattes, and C. Minkenberg, "A throughput-optimized optical network for data-intensive computing," *IEEE Micro*, vol. 34, no. 5, pp. 52–63, Sept 2014.

[2] A. V. Krishnamoorthy, H. Schwetman, X. Zheng, and R. Ho, "Energy-efficient photonics in future high-connectivity computing systems," *J. Lightwave Technol.*, vol. 33, no. 4, pp. 889–900, Feb 2015.

[3] Intel Corporation, "U.S. Department of Energy selects Intel to deliver nation's most powerful supercomputer at Argonne National Laboratory," newsroom.intel.com/community/intel_newsroom/blog/2015/04/09/us-department-of-energy-selects-intel-to-deliver-nations-most-powerful-supercomputer-at-argonne-national-laboratory, April 2015.

[4] A. Zulfiqar, P. Koka, H. Schwetman, M. Lipasti, X. Zheng, and A. Krishnamoorthy, "Wavelength stealing: An opportunistic approach to channel sharing in multi-chip photonic interconnects," in *Proc. of the 46th Annual IEEE/ACM Intl. Symp. on Microarchitecture*. New York, NY, USA: ACM, 2013, pp. 222–233.

[5] P. Koka, M. O. McCracken, H. Schwetman, X. Zheng, R. Ho, and A. V. Krishnamoorthy, "Silicon-photonic network architectures for scalable, power-efficient multi-chip systems," *SIGARCH Comput. Archit. News*, vol. 38, no. 3, pp. 117–128, Jun. 2010.

[6] InfiniBand Trade Association, "Infiniband roadmap," www.infinibandta.org/content/pages.php?pg=technology_overview, October 2015.

[7] V. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, p. P10008, 2008.

[8] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, p. 026113, 2004.

[9] D. Chavarría-Miranda, M. Halappanavar, and A. Kalyanaraman, "Scaling Graph Community Detection on the Tilera Many-core Architecture," in *Proc. IEEE Intl. Conf. on High Performance Computing*, 2014.

[10] M. Halappanavar, "Algorithms for vertex-weighted matching in graphs," Ph.D. dissertation, Old Dominion University, Norfolk, VA, USA, 2009.

[11] U. V. Catalyurek, F. Dobrian, A. Gebremedhin, M. Halappanavar, and A. Pothen, "Distributed-memory parallel algorithms for matching and coloring," in *Proc. of the 2011 IEEE IPDPS Workshops and PhD Forum*. Washington, DC, USA: IEEE Computer Society, 2011, pp. 1971–1980.

[12] F. Manne and R. H. Bisseling, "A parallel approximation algorithm for the weighted maximum matching problem," in *Proc. of the 7th Intl. Conf. on Parallel Processing and Applied Mathematics*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 708–717.

[13] A. Alexandrov, M. F. Ionescu, K. E. Schauser, and C. Scheiman, "LogGP: Incorporating long messages into the LogP model — One step closer towards a realistic model for parallel computation," in *Proc. of the 7th ACM Symp. on Parallel Algorithms and Architectures*. New York, NY, USA: ACM, 1995, pp. 95–105.

[14] M. Halappanavar, J. Feo, O. Villa, A. Tumeo, and A. Pothen, "Approximate weighted matching on emerging manycore and multithreaded architectures," *Int. J. High Perform. Comput. Appl.*, vol. 26, no. 4, pp. 413–430, Nov. 2012.

[15] A. Morari, A. Tumeo, O. Villa, S. Secchi, and M. Valero, "Efficient sorting on the tilera manycore architecture," in *IEEE 24th Intl. Symp. on Computer Architecture and High Performance Computing*, New York, NY, October 2012, pp. 171–178.

[16] C. S. Guiang, K. F. Milfeld, A. Purkayastha, and J. R. Boisseau, "Memory performance of dual-processor nodes: comparison of Intel Xeon and AMD Opteron memory subsystem architectures," in *Proc. for ClusterWorld Conference and Expo*, 2003.

[17] J. Feo, D. Harper, S. Kahan, and P. Konecny, "Eldorado," in *Proc. of the 2nd Conf. on Computing Frontiers*. New York, NY, USA: ACM, 2005, pp. 28–34.

[18] K. J. Barker, A. Benner, R. Hoare, A. Hoisie, A. K. Jones, D. J. Kerbyson, D. Li, R. Melhem, R. Rajamony, E. Schenfeld, S. Shao, C. Stunkel, and P. Walker, "On the feasibility of optical circuit switching for high performance computing systems," in *Proc. of the 2005 ACM/IEEE Conf. on Supercomputing*, November 2005.

[19] K. J. Barker and D. J. Kerbyson, "Circuit switched network for peta-scale systems," in *Proc. of 13th International Euro-Par Conference*, Rennes, France, August 2007, pp. 858–867.

[20] D. J. Kerbyson and K. J. Barker, "Analyzing the performance bottlenecks of POWER7-IH network," in *Proc. of IEEE Conference on Cluster Computing*, Austin, TX, USA, September 2011.

[21] K. J. Barker, D. J. Kerbyson, and A. Hoisie, "An early performance analysis of POWER7-IH systems," in *Proc. of the 2011 Intl. Conf. for High Performance Computing, Network, Storage, and Analysis*, Seattle, WA, USA, November 2011.

[22] Z. Zhang, L. Cherkasova, A. Verma, and B. T. Loo, "Performance modeling and optimization of deadline-driven Pig programs," *ACM Trans. Auton. Adapt. Syst.*, vol. 8, no. 3, pp. 14:1–14:28, Sep. 2013.

[23] A. Gharaibeh, L. Beltrão Costa, E. Santos-Neto, and M. Ripeanu, "A Yoke of Oxen and a Thousand Chickens for Heavy Lifting Graph Processing," in *Proc. of the 21st Intl. Conf. on Parallel Architectures and Compilation Techniques*. New York, NY, USA: ACM, 2012, pp. 345–354.

[24] W. Anderson, P. Briggs, C. S. Hellberg, D. W. Hess, A. Khokhlov, M. Lanzagorta, and R. Rosenberg, "Early Experience with Scientific Programs on the Cray MTA-2," in *Proc. of the 2003 ACM/IEEE Conf. on Supercomputing*. New York, NY, USA: ACM, 2003, pp. 46–.

[25] J. Nieplocha, A. Márquez, J. Feo, D. Chavarría-Miranda, G. Chin, C. Scherrer, and N. Beagley, "Evaluating the Potential of Multithreaded Platforms for Irregular Scientific Computations," in *Proc. of the 4th Intl. Conf. on Computing Frontiers*. New York, NY, USA: ACM, 2007, pp. 47–58.

[26] S. Secchi, A. Tumeo, and O. Villa, "A Bandwidth-Optimized Multi-core Architecture for Irregular Applications," in *Proc. of the 12th IEEE/ACM Intl. Symp. on Cluster, Cloud and Grid Computing*, Washington, DC, USA, 2012, pp. 580–587.