System-level Evaluation of Chip-Scale Silicon Photonic Networks for Emerging Data-Intensive Applications

Aditya Narayan*, Yvain Thonnart†, Pascal Vivet†, Ajay Joshi* and Ayse K. Coskun*
*Boston University, Boston, MA 02215, USA; E-mail: {adityan, joshi, acoskun}@bu.edu
†Université Grenoble Alpes, CEA-Leti, Grenoble, France; E-mail: {first.last}@cea.fr

***Abstract*—Emerging data-driven applications such as graph processing applications are characterized by their excessive memory footprint and abundant parallelism, resulting in high memory bandwidth demand. As the scale of datasets for applications is reaching orders of TBs, performance limitation due to bandwidth demands is a major concern. Traditional on-chip electrical networks fail to meet such high bandwidth demands due to increased energy-per-bit or physical limitations with pin counts. Silicon photonic networks have emerged as a promising alternative to electrical interconnects, owing to their high bandwidth density and low energy-per-bit communication with negligible data-dependent power. Wide-scale adoption of silicon photonics at chip level, however, is hampered by their high sensitivity to process and thermal variations, high laser power due to losses along the network, and power consumption of the electrical-optical conversion. Device-level technological innovations to mitigate these issues are promising, yet they do not consider the system-level implications of the applications running on manycore systems with photonic networks. This work aims to bridge the gap between the system-level attributes of applications with the underlying architectural and device-level characteristics of silicon photonic networks to achieve energy-efficient computing. We particularly focus on graph applications, which involve unstructured yet abundant parallel memory accesses that stress the on-chip communication networks, and develop a cross-layer framework to evaluate 2.5D systems with silicon photonic networks. We demonstrate 38% power savings through system-level management using wavelength selection policies with only 1% loss in system performance and further evaluate architectural design choices on 2.5D systems with photonic networks.**

I. Introduction

Data-intensive workloads are becoming increasingly widespread in different domains such as physics simulations, biochemistry, image processing, aircraft scheduling, etc. [1]. As the scale of data being processed by applications in these domains is increasing, graph processing is emerging as an important medium for modeling and evaluating the patterns and relationships in interconnected data. There have been major efforts to improve the data organization and representation of graph data [2], with further work on developing graph-specific architectures [3], [4] to improve the performance of graph applications. As current core counts are not sufficient for supporting the ever-increasing datasets of these domains, these data-intensive graph workloads are pushing the need for large manycore systems.

To support graph applications on manycore systems, we need a dense integration of large number of cores with high-bandwidth and low-latency communication networks. Traditional 2D systems are reticle-limited and give rise to high manufacturing costs due to poor yield. Even 3D-integrated technologies, despite their much higher bandwidth densities than 2D system, often suffer from thermal challenges [5] due to dense integration. Therefore, 2.5D manycore systems are materializing as low-cost and energy-efficient alternative [6], [7]. Multiple smaller chiplets are stacked over a large interposer chip in a 2.5D system. Such 2.5D manycore systems provide a favorable computing substrate for data-intensive graph applications that demonstrate high parallelism.

Graph applications, however, are fundamentally characterized by their high orders of random and irregular data accesses. With datasets of real world graphs being on the order of TBs, the performance of graph applications are limited by the memory bandwidth offered by the interconnection networks in manycore systems. To support graph processing applications on 2.5D manycore systems, we need inter-chiplet communication bandwidth on the order of $1Tbps$. Traditional high-speed electrical links fail to provide this required bandwidth due to pin limitations.

With the emergence of CMOS-integrated photonic technology, photonic networks have been demonstrated to provide high-bandwidth and low-latency communication with negligible data-dependent power [8]–[10]. Therefore, photonic networks are a promising alternative to electrical links for inter-chiplet communication in 2.5D manycore systems. Chip-scale photonic communication is conventionally performed using photonic links with microring resonators (MRRs). MRRs are used for modulating the light waves at the transmitter site (Tx) as well as filtering light waves at the receiver site (Rx) of the photonic link. In recent years, photonic networks using MRRs have been explored for 2.5D systems [11]–[13].

Though silicon-photonic link technology has shown promise of sub-pJ energy-per-bit communication, the maturity of chip-scale photonic networks is hampered by the high sensitivity of MRRs to thermal and process variations, high power overhead along the network and bandwidth-energy tradeoff for optimal utilization. The thermal sensitivity of MRRs and device-level techniques to mitigate such thermal effects have been studied over the past years [14]. Conventionally, a closed-loop feedback monitoring mechanism detects the MRR resonance shift due to thermal variations and performs controlled local heat injection to tune the MRRs back to resonance. Several such techniques, analog and digital, have been demonstrated to handle large temporal thermal variations [15]–[17]. However, there is a strong diversity among applications with respect to their runtime bandwidth needs and resource utilization that result in highly application-specific power and thermal profiles. We could increase the network bandwidth to the required peak value, but this comes at a considerably high power cost of lasers, electrical-optical conversion circuitry and the thermal tuning of MRRs. Therefore, we strongly argue for application-aware and device-level solution aware solutions to manage the system performance and power.

Our specific contributions are as follows:

1) We demonstrate that graph processing on large datasets has bandwidth requirements of the order of $1 - 2Tbps$ (Sec. II-A). When running these applications on 2.5D manycore systems, we identify photonic networks as a promising solution for inter-chiplet communication as it provides the required high bandwidth density. (Sec. II-B)
2) We observe that graph applications' bandwidth needs are highly diverse. We argue for a need for system-level management policies that caters to application-specific bandwidth needs on top of underlying device-level solutions (Sec. IV). We demonstrate the benefits of our wavelength selection policy, *WAVES*, on graph applications and obtain power savings of 36% on average using minimum required bandwidth for an application (Sec. V-B).
3) We perform a detailed architectural evaluation of 2.5D manycore systems with integrated photonic networks. We observe that large L2 caches do not provide any performance improvements when photonic links are able to meet the required bandwidth of applications. Furthermore, photonic links are also able to provide scalable bandwidth for highly parallel graph applications as the number of chiplets increases (Sec. V-C, V-D).

II. Background and Motivation

A. Graph applications

A graph represents the basic relationship between two vertices. Graphs are rather ubiquitous in real-world applications, such as social network applications, web applications, transportation applications, etc. The graphs in these applications are extremely large with upto a billion of vertices and similar number of edges interconnecting these vertices [18].

A primary bottleneck in the execution of these graph applications is the highly irregular memory access patterns resulting in poor spatial and temporal locality. These irregular access patterns often result in high and frequent memory accesses. In large 2.5D systems, when the last level caches (LLC) are spread over multiple chiplets, the data accesses to LLC on separate chiplets and DRAM accesses constitute a major fraction of the application execution time. This is illustrated in Fig. 1, which shows high fraction of time spent in memory accesses for graph applications for two different memory and LLC access bandwidths. As the network bandwidth increases from $96Gbps$ to $1.5Tbps$, there is an average 61% reduction in the fraction of time spent in memory accesses. Therefore, the memory and LLC bandwidths play a crucial role in influencing the performance of large 2.5D systems running graph applications.

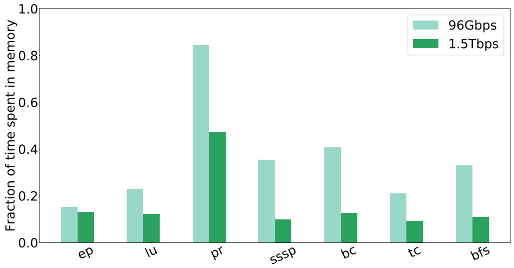


Fig. 1: Fraction of time spent in memory accesses for applications from NAS Parallel Benchmarks [19] (*ep* and *lu*) and graph applications from GAP-BS [20] (*pr, sssp, bc, tc* and *bfs*) for two different memory and LLC access bandwidth.

B. Silicon-photonic links

To support the high bandwidth density of graph applications, electrical links often fall short due to pin limitations. On the other hand, silicon-photonic technology has seen a rapid growth that has promised much higher orders of chip-scale communication bandwidth in 2.5D manycore systems. Several device-level innovations have demonstrated the feasibility of integrating photodiodes [21], low-loss waveguides [22], and MRR modulators and filters [23] through the use of slightly adapted or unmodified CMOS process. This has paved the way for realization of efficient photonic links for communication.

A major obstacle towards attaining sub-pJ per bit energy communication in photonic links stems from the high laser power due to losses along the waveguide [22] and the high thermal tuning power resulting from MRR sensitivity towards manufacturing process [24] and on-chip thermal variations [17]. In large 2.5D systems, high core activity creates large thermal variations and hot spots. These hot spots can reach high temperatures ($>85°C$) for these data-intensive graph workloads. Therefore, MRRs on the interposer experience resonance wavelength shifts. To compensate for resonant wavelength shifts, the MRRs are thermally tuned by controlled local heat injection. When using wavelength division multiplexing, if n laser wavelengths are used within a free-spectral range of FSR, the maximum tuning shift that any MRR has to undergo is FSR/n.

Device-level solutions such as analog thermal control loop for thermal management continuously monitor the MRR resonance shift and supply required heating power to tune the MRRs back to resonance [15]–[17]. The heater aims to maintain a fixed temperature for an MRR, so that the MRR resonance is locked to a laser wavelength. A distinguishing feature of this thermal control loop is that it enables remapping of MRRs to any laser wavelengths at runtime. Therefore, if only k among n laser wavelengths are activated at runtime, depending on the thermal profile of an application, different set of k MRRs can be mapped to the k activated laser wavelengths with the goal of minimizing the overall heating power. We first conduct system-level studies to determine bandwidth needs of an application and then activate the minimum required number of laser wavelengths that can satisfy the average bandwidth needs of applications. The underlying thermal control loop maps the appropriate MRRs to the activated laser wavelengths.

III. Related work

Graph applications have been extensively studied due to their use in a wide variety of fields. Processing-in-memory (PIM) based solutions have been studied for graph applications, as PIM designs provide a high bandwidth density [4]. Ham *et al.* [3] design a specialized hardware pipeline and memory subsystem for graph analytics. These prior works primarily focus on obtaining microarchitectural insights, evaluating tradeoffs and proposing architectural solutions that can address the high parallelism, memory and bandwidth needs for graph applications.

2.5D-integrated systems with chip-scale photonic networks have been extensively studied because of their potential performance and thermal advantages. *Galaxy* [12] is a multi-chip architecture that integrates multiple small chiplets through optical fibers and incorporates photonic waveguides for distant intra-chiplet communication. Grani *et al.* [13] implement a crossbar-based photonic network using arrayed waveguide grating router on a silicon interposer and demonstrate high bisection bandwidth at low energy-per-bit values. Fotouhi *et al.* [11] design a scalable uniform memory architecture with photonic interconnects by moving large LLC from processor chiplet to separate chiplets.

System-level management policies to address power concerns in photonic networks and further improve the energy efficiency have been shown to be effective. *RingAware* [25] and *FreqAlign* [26] employ thread allocation and migration to manage the thermal gradients around communicating MRRs and reduce the thermal tuning power. *Aurora* [27] encompasses a cross-layer approach at the device, system and OS-level to control the thermal tuning power. Chen *et al.* [28], [29] perform dynamic laser management using cache reconfiguration on a manycore system with silicon-photonic crossbar NoC.

As silicon photonic networks provide scalable bandwidth with laser wavelengths and 2.5D systems enable dense integration of chiplets, we observe that such large 2.5D systems with chip-scale photonic networks are energy-efficient solutions to address the high parallelism and bandwidth demands of graph applications. In contrast to earlier works, our system-level wavelength selection encompasses application-specific bandwidth needs and the device-level solutions to address the power-bandwidth tradeoff in photonic networks. We further evaluate graph applications with different architectural parameters and provide insights about their behavior with memory hierarchy and increasing chiplet counts in 2.5D systems.

IV. System architecture and management policy

Our target system is a 2.5D homogeneous manycore system with inter-chiplet photonic network called Processors On Photonic Silicon inTerposer ARchitecture (*POPSTAR*) that was presented in our earlier work [30]. In this section, we briefly detail the *POPSTAR* architecture and our wavelength selection methodology, *WAVES*. We then demonstrate the benefits of *WAVES* on graph workloads running large datasets.

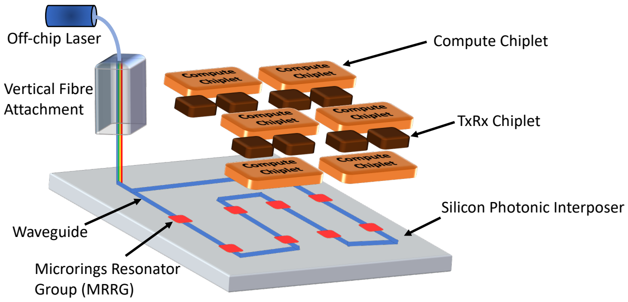


Fig. 2: The POPSTAR architecture.

Table I: Microarchitectural details of *POPSTAR*

Execution Core	533MHz IA-32 core, x86 ISA with out-of-order execution Dispatch width 4, branch misprediction penalty = 10
On-chip caches	16KB private L1 I and D cache, 4-way, 3 cycle, 64B line size 64KB private L2 cache, 4-way, 8 cycle, 64B line size Shared distributed L3 cache, 8MB per chiplet, 16-way, 20 cycles, 64B line size
Inter-chiplet photonic network	Single-writer multiple reader link 12Gbps datarate per laser wavelength, 1.5Tbps peak aggregate interposer bandwidth

A. System architecture

POPSTAR is a 96-core system with compute chiplets and TxRx chiplets integrated on a photonic interposer as depicted in Fig. 2. There are six compute chiplets, each consisting of 16 IA-32 cores from Intel SCC [31]. The microarchitectural details of *POPSTAR* are detailed in Table. I. There are eight TxRx chiplet, each composed of the electronic circuit for routing, flow control, arbitration, and Electrical-Optical (E-O) and Optical-Electrical (O-E) conversion. Six TxRx chiplets connect to the compute chiplets via a 96-bit wide interface, and two TxRx chiplets connect to external off-interposer main memory. The MRRs and photodiodes for photonic communication are organized in microring resonator groups (MRRG) underneath each TxRx chiplet on the photonic interposer. An off-chip laser emits up to 16 wavelengths that are carried by a vertical fiber attachment and coupled onto the waveguides in the interposer via grating couplers. The 16 wavelengths are evenly spaced in an FSR of 10.8nm around a center wavelength of 1310nm.

B. Simulation framework

We design a simulation framework [30] encompassing a performance simulator, a logic power calculator, a photonic network power model, and a thermal simulator. We use Sniper [32] as our performance simulator. We model the architectural parameters of *POPSTAR* in Sniper and evaluate the system performance. We use widely used graph applications such as PageRank (pr), Breadth First Search (bfs), Single-Source Shortest Paths ($sssp$), Betweenness Centrality (bc) and Triangle Counting (tc) from GAP Benchmark Suite [20]. We evaluate the graph applications on three datasets, two Kronecker graphs with 2^{18} and 2^{20} nodes and a real-world dataset from Google web graph (|V|=875713, |E|=5105039) [18].

We feed the performance statistics from Sniper as input to McPAT [33] and calculate the core and cache power. We collect power traces from all our experiments and use the published data from Intel SCC to calibrate our dynamic power data [31]. Since the leakage power component is strongly dependent on temperature, we implement a linear temperature-dependent leakage power model in our thermal simulator [34].

To calculate the photonic power component, we use our analytical model developed in our earlier work [30]. We calculate the laser and EOE power based on the number of activated laser wavelengths in the system. We use the 3D extension of HotSpot [35], [36] to determine the thermal profile of each MRR. Using the compute chiplet power and TxRx chiplet power as inputs to HotSpot, we determine the thermal profile in the photonic interposer and calculate the temperature of each MRR. We assume the MRR temperatures within a MRRG remain the same due to the small area footprint of a MRRG. Since each MRR is designed to resonate at a specific laser wavelength at a temperature of 300K, we calculate the tuning shift required to tune each MRR back to the desired resonance. Using the MRR temperatures obtained from our thermal simulation, we determine the aggregate heating power to thermally tune all the MRRs to the activated laser wavelengths in the system.

C. Wavelength selection policy (WAVES)

The power consumption along a photonic link consists of the laser power, the EOE power and the heating power to thermally tune the MRRs. The overall photonic power increases as the number of the activated laser wavelengths (λ_{act}) in the system increases. Therefore, even though a higher λ_{act} is desirable for higher performance, it comes at a considerably high power cost. Figure 3 illustrates the normalized execution time of graph applications as we increase the peak aggregate bandwidth in the interposer by activating more laser wavelengths. We observe substantial speedup initially as we increase the inter-chiplet bandwidth, λ_{act}. This speedup corresponds to the increased L2-L3 and L3-DRAM bandwidth. However, the performance saturates at different bandwidth values for different applications. It is, therefore, counter-productive to activate all laser wavelengths in the system for all applications. We argue that it is essential to address this bandwidth-power tradeoff and activate the minimum number of activated laser wavelengths, λ_{min}, for different applications that caters to specific bandwidth requirements of that application.

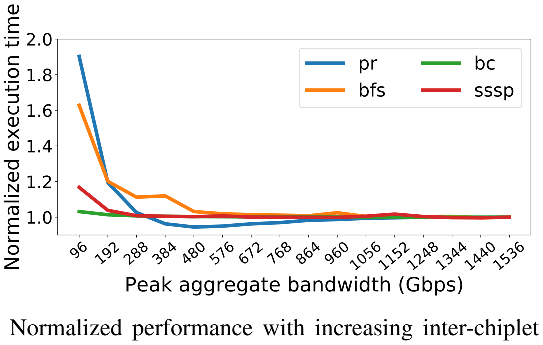


Fig. 3: Normalized performance with increasing inter-chiplet bandwidth for graph applications on Google web graph. The performance is normalized to the performance with peak bandwidth of $1.536Tbps$.

WAVES is a static wavelength policy that determines λ_{min} required for an application through offline analysis based on a set performance loss threshold (L_{thr}). The performance loss is calculated from the case where all laser wavelengths in the system are activated, i.e. $\lambda_{act} = \lambda_{tot}$. By setting an L_{thr} that is deemed acceptable for a system, we ensure that we are achieving maximum power savings by meeting the performance requirements. Once we determine the λ_{min} for an application offline, the runtime execution can result in highly application-specific thermal profile. As different MRRs in the system incur different resonance shifts, we determine the aggregate resonance shift of all MRRs. Furthermore, there are $\binom{\lambda_{tot}}{\lambda_{min}}$ combinations to activate λ_{min} laser wavelengths among λ_{tot}, and each combination requires a different tuning range of MRRs to lock on to the laser wavelengths. We determine the optimal combination of λ_{min} that result in the lowest thermal tuning range. The analog thermal control loop continuously monitors the MRR resonance shift and supplies appropriate heating power to tune each MRR to selected laser wavelengths.

V. Evaluation results

In this section, we perform an architectural evaluation of graph applications on 2.5D manycore systems with photonic networks. Furthermore, we evaluate the benefits of our wavelength selection policy, *WAVES*, on graph applications.

A. Different system utilization of graph application

We first explore the parallelism of graph applications in our 96-core *POPSTAR* system. We observe from Fig. 4 that the overall system performance of graph applications improves significantly as we execute them with higher thread counts. We get performance improvement of upto 74% for *pr* and an average improvement of 60% by running these applications with 96 threads compared to 24 threads. This can primarily be attributed to the inherent parallelism of graph applications. As the number of threads increases, the overall LLC and memory accesses also increases, resulting in higher inter-chiplet communication traffic. The high-bandwidth silicon-photonic links are able to meet the high bandwidth demands with increasing thread counts and, therefore, facilitate the execution of these parallel graph applications.

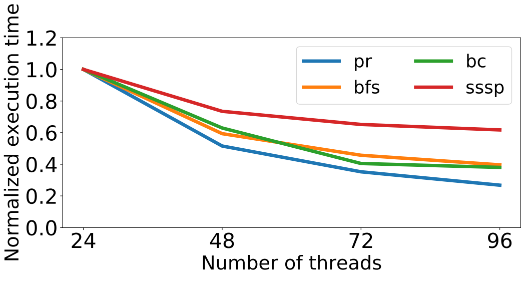


Fig. 4: Normalized execution time with increasing thread counts. The performance is normalized to the execution time with 24 threads.

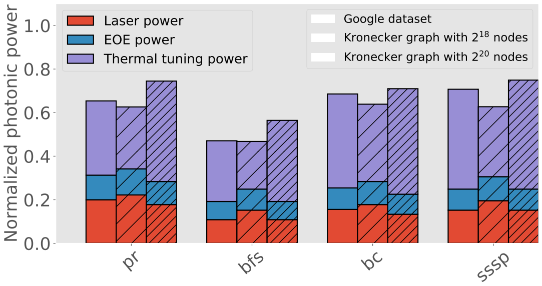


Fig. 5: Power consumption in photonic network for graph applications on three different datasets. Power numbers are normalized to baseline case where all laser wavelengths are activated.

B. WAVES on graph applications

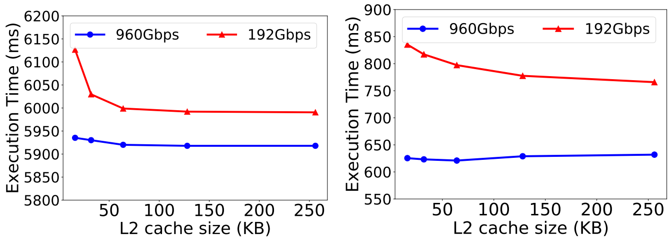
To investigate the power benefits from *WAVES* on graph applications, we compare to a baseline case where we activate all laser wavelengths in the system (λ_{tot}) to achieve peak aggregate bandwidth of $1.5Tbps$. We set the performance loss threshold, L_{thr}, of 1%, and determine the number of laser wavelengths, λ_{min}, that is able to provide an inter-chiplet bandwidth to meet system performance within this threshold. Figure. 5 shows the normalized photonic power with λ_{min}, compared to the power with the highest bandwidth, i.e. λ_{tot}.

On average, we obtain 36% reduction in power with λ_{min} than using the peak aggregate bandwidth with λ_{tot} using our *WAVES* policy across applications. For the real-world Google web graph, we obtain power savings of 38% with λ_{min}. Our policy also accounts for the thermal profile of applications and MRR process variations, and selectively activates the λ_{min} that result in lowest thermal tuning power. The underlying thermal control loop [17] remaps the MRR to the activated λ_{min} laser wavelengths. We also observe that graphs with larger datasets consume higher photonic power. This is due to the increased bandwidth needs and higher inter-chiplet communication traffic as the scale of input dataset increases. Our WAVES policy, therefore, addresses the power-performance tradeoffs of applications executing on 2.5D manycore systems with photonic links and provides an energy-efficient execution.

C. Graph bandwidth needs with increasing L2 size

We evaluate the performance of graph applications with varying private L2 cache sizes for two different inter-chiplet bandwidth. For this experiment, we use the Google web graph dataset from SNAP [18]. We observe that the application performance improves as we increase the L2 cache size for a low inter-chiplet bandwidth of $192Gbps$ (see Fig. 6). However, a higher inter-chiplet bandwidth of $960Gbps$ shows minimal execution time variations with increasing L2 cache size.

For lower inter-chiplet bandwidth and smaller L2 cache sizes, the execution time due to L2 misses also includes the high fraction of queue latency in the photonic link. Increasing the L2 cache size improves the hit rate and we observe a speedup in the performance. However, the L2 miss latency is still dominated by the queue latency in the photonic link. When we increase the inter-chiplet bandwidth to meet the bandwidth requirements of graph applications, we significantly reduce the queue latency. As a result, the L2 cache misses for the same L2 cache size is serviced faster with a high-bandwidth link. Due to irregular memory accesses in graph applications, we do not observe performance improvement with increasing L2 cache when the bandwidth requirements are met.



(a) *bc*

(b) *pr*

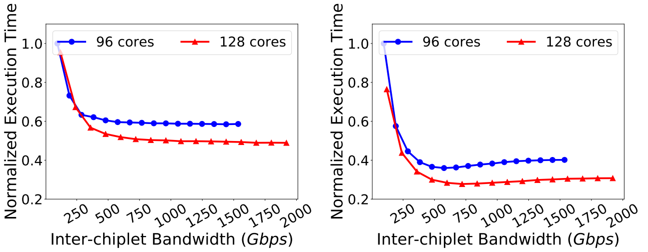
Fig. 6: Performance of *bfs* and *pr* with different inter-chiplet bandwidth, when executed on 2 systems with different L2 cache sizes.

As photonic links for inter-chiplet communication in 2.5D manycore systems are able to meet the high bandwidth demands of applications, there is an opportunity to incorporate a smaller L2 cache per core and per chiplet.

D. Graph bandwidth needs with higher chiplet counts

In this section, we evaluate the performance scaling of graph applications with increasing core counts. As 2.5D systems enable modularity, we integrate more chiplets on the interposer, keeping the same number of cores per chiplet. For this experiment, we use our largest data graph, the Kronecker graph with 2^{20} vertices. As the number of chiplets increases from six compute chiplets in a 96-core system to eight chiplets in a 128-core system, the peak aggregate bandwidth on the interposer increases for same number of activated laser wavelengths. The maximum bandwidth with $\lambda_{act} = 16$ increases from $1.5Tbps$ in 96-core system to $1.9Tbps$ in 128-core system.

We observe a performance improvement of 21% on average for a 128-core system compared to a 96-core system for the same number of activated laser wavelengths (see Fig. 7). It is interesting to note that the system performance saturates at a higher inter-chiplet bandwidth for the 128-core system than the 96-core system. For example, in *bfs*, we obtain a system performance within 1% of peak performance for an inter-chiplet bandwidth of $864Gbps$ ($\lambda_{act} = 9$) in a 96-core system, while in the 128-core system, we obtain 1% of peak performance for an inter-chiplet bandwidth of $1.56Tbps$ ($\lambda_{act} = 13$). Similarly, in *pr*, we obtain the peak performance for $\lambda_{act} = 6$ for both systems. However, the aggregate bandwidth corresponds to $576Gbps$ in a 96-core system and $720Gbps$ in a 128-core system.



(a) *bfs*

(b) *pr*

Fig. 7: Performance of *bfs* and *pr* with different inter-chiplet bandwidth, when executed on 2 systems with different core counts.

These observations enforce the scalability of graph applications with number of coures due to their inherent parallelism. There is a significant increase in inter-chiplet traffic with increasing LLC and memory accesses with higher chiplet counts. Therefore, 2.5D manycore systems with photonic links are able to meet the required bandwidths for graph applications. Furthermore, as application's bandwidth needs increase with larger chiplet counts as seen in Fig. 7, our proposed *WAVES* policy can adapt to meet these changing bandwidth needs.

VI. Conclusion

Graph applications form a domain of emerging workloads that demand high bandwidth, due to increased data footprint of real-world graphs and abundant parallel memory accesses. 2.5D integration provide the opportunity for modular integration of a large number of chiplets to support these graph applications. The inter-chiplet bandwidth demands of emerging data-centric applications can reach as high as $1-2Tbps$. Silicon-photonic links, despite their capability to meet the high bandwidth demands of graph applications, often suffer from high power cost. In this work, we demonstrate the benefits of wavelength selection, *WAVES*, that enables power-efficient execution of graph applications on a 2.5D manycore system with photonic links.

Furthermore, as silicon-photonic links are able to meet the high bandwidth demands, there lies a favorable premise to move to larger 2.5D systems that are beneficial to highly-parallel applications. We also demonstrate a study showing the redundancy of large L2 caches, as photonic links provide an opportunity to rethink the conventional cache hierarchy.

Acknowledgement

This work was funded partly by the CARNOT institute, and by NSF CCF-1716352.

References

[1] S. Shirinivas, S. Vetrivel, and N. Elango, “Applications of graph theory in computer science an overview,” *Int. Journal of Engineering Science and Technology*, vol. 2, no. 9, pp. 4610–4621, 2010.

[2] G. Malewicz *et al.*, “Pregel: A system for large-scale graph processing,” in *Proc. Int. Conf. on Management of data*, 2010, pp. 135–146.

[3] T. J. Ham *et al.*, “Graphicionado: A high-performance and energy-efficient accelerator for graph analytics,” in *Proc. Int. Symp. on Microarchitecture (MICRO)*, 2016, pp. 1–13.

[4] J. Ahn *et al.*, “A scalable processing-in-memory accelerator for parallel graph processing,” in *Proc. Int. Symp. on Computer Architecture (ISCA)*, 2016, pp. 105–117.

[5] T. Zhang, Y. Zhan, and S. S. Sapatnekar, “Temperature-aware routing in 3D ICs,” in *Proc. Asia and South Pacific Design Automation Conf.*, 2006, pp. 309–314.

[6] I. Akgun, J. Zhan, Y. Wang, and Y. Xie, “Scalable memory fabric for silicon interposer-based multi-core systems,” in *Proc. Int. Conf. on Computer Design*, 2016, pp. 33–40.

[7] J. Macri, “AMD's next generation GPU and high bandwidth memory architecture: FURY,” in *Proc. Hot Chips Symp. (HCS)*, 2015, pp. 1–26.

[8] P. Dong *et al.*, “Reconfigurable 100 Gb/s silicon photonic network-on-chip,” *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 1, pp. A37–A43, 2015.

[9] G. A. Fish and D. K. Sparacin, “Enabling flexible datacenter interconnect networks with WDM silicon photonics,” in *Proc. Custom Integrated Circuits Conf.*, 2014, pp. 1–6.

[10] Z. Wang *et al.*, “CAMON: Low-cost silicon photonic chiplet for manycore processors,” *IEEE Trans. on Computer Aided Design (TCAD)*, 2019.

[11] P. Fotouhi, S. Werner, J. Lowe-Power, and S. Yoo, “Enabling scalable chiplet-based uniform memory architectures with silicon photonics,” in *Proc. Int. Symp. on Memory Systems*, 2019, pp. 222–334.

[12] Y. Demir *et al.*, “Galaxy: A high-performance energy-efficient multi-chip architecture using photonic interconnects,” in *Proc. Int. Conf. on Supercomputing*, 2014, pp. 303–312.

[13] P. Grani, R. Proietti, V. Akella, and S. B. Yoo, “Design and evaluation of AWGR-based photonic NoC architectures for 2.5 D integrated high performance computing systems,” in *Proc. Int. Symp. on High Performance Computer Architecture (HPCA)*, 2017, pp. 289–300.

[14] K. Padmaraju and K. Bergman, “Resolving the thermal challenges for silicon microring resonator devices,” *Nanophotonics*, vol. 3, no. 4-5, pp. 269–281, 2014.

[15] C. Sun *et al.*, “A 45nm CMOS-SOI monolithic photonics platform with bit-statistics-based resonant microring thermal tuning,” *IEEE Journal of Solid-State Circuits*, vol. 51, no. 4, pp. 893–907, 2016.

[16] P. Dong *et al.*, “Thermally tunable silicon racetrack resonators with ultralow tuning power,” *Optics express*, vol. 18, no. 19, pp. 20 298–20 304, 2010.

[17] Y. Thonnart *et al.*, “A 10Gb/s Si-photonic transceiver with 150μW 120μs-lock-time digitally supervised analog microring wavelength stabilization for $1Tb/s/mm^2$ die-to-die optical networks,” in *Proc. International Solid-State Circuits Conference (ISSCC)*, 2018, pp. 350–352.

[18] J. Leskovec and A. Krevl, “SNAP Datasets: Stanford large network dataset collection,” http://snap.stanford.edu/data, Jun. 2014.

[19] D. H. Bailey *et al.*, “The NAS parallel benchmarks,” *International Journal of Supercomputing Applications*, vol. 5, no. 3, pp. 63–73, 1991.

[20] S. Beamer, K. Asanović, and D. Patterson, “The gap benchmark suite,” *arXiv preprint arXiv:1508.03619*, 2015.

[21] L. Virot *et al.*, “Germanium avalanche receiver for low power interconnects,” *Nature communications*, vol. 5, p. 4957, 2014.

[22] J. Cardenas, C. B. Poitras, J. T. Robinson, K. Preston, L. Chen, and M. Lipson, “Low loss etchless silicon photonic waveguides,” *Optics express*, vol. 17, no. 6, pp. 4752–4757, 2009.

[23] W. Bogaerts *et al.*, “Silicon microring resonators,” *Laser & Photonics Reviews*, vol. 6, no. 1, pp. 47–73, 2012.

[24] A. V. Krishnamoorthy *et al.*, “Exploiting CMOS manufacturing to reduce tuning requirements for resonant optical devices,” *IEEE Photonics Journal*, vol. 3, no. 3, pp. 567–579, 2011.

[25] T. Zhang, J. L. Abellán, A. Joshi, and A. K. Coskun, “Thermal management of manycore systems with silicon-photonic networks,” in *Proc. Design, Automation & Test in Europe Conf. (DATE)*, 2014, p. 307.

[26] J. L. Abellán *et al.*, “Adaptive tuning of photonic devices in a photonic NoC through dynamic workload allocation,” *IEEE TCAD*, vol. 36, no. 5, pp. 801–814, 2017.

[27] Z. Li *et al.*, “Aurora: A cross-layer solution for thermally resilient photonic network-on-chip,” *IEEE Trans. on Very Large Scale Integration Systems*, vol. 23, no. 1, pp. 170–183, 2015.

[28] C. Chen and A. Joshi, “Runtime management of laser power in silicon-photonic multibus noc architecture,” *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 19, no. 2, pp. 3 700 713–3 700 713, 2013.

[29] C. Chen, J. L. Abellán, and A. Joshi, “Managing laser power in silicon-photonic NoC through cache and NoC reconfiguration,” *IEEE TCAD*, vol. 34, no. 6, pp. 972–985, 2015.

[30] A. Narayan *et al.*, “WAVES: Wavelength selection for power-efficient 2.5D integrated photonic NoCs,” in *Proc. DATE*, 2019, pp. 516–521.

[31] J. Howard *et al.*, “A 48-core IA-32 message-passing processor with DVFS in 45nm CMOS,” in *Proc. ISSCC*, 2010, pp. 108–109.

[32] T. E. Carlson, W. Heirman, and L. Eeckhout, “Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulation,” in *Proc. Int. Conf. for High Performance Computing, Networking, Storage and Analysis*, 2011, p. 52.

[33] S. Li *et al.*, “McPAT: an integrated power, area, and timing modeling framework for multicore and manycore architectures,” in *Proc. MICRO*, 2009, pp. 469–480.

[34] H. Wong. A comparison of Intel's 32nm and 22nm core i5 CPUs: Power, voltage, temperature, and frequency. [Online]. Available: http://blog.stuffedcow.net/2012/10/intel32nm-22nm-core-i5-comparison/

[35] K. Skadron *et al.*, “Temperature-aware microarchitecture,” in *Proc. ISCA*, 2003, pp. 2–13.

[36] J. Meng, K. Kawakami, and A. K. Coskun, “Optimizing energy efficiency of 3D multicore systems with stacked DRAM under power and thermal constraints,” in *Proc. Design Automation Conference*, 2012, pp. 648–655.