Thermal Management of Manycore Systems with Silicon-Photonic Networks

Tiansheng Zhang, José L. Abellán, Ajay Joshi, Ayse K. Coskun
Electrical and Computer Engineering Department, Boston University, Boston, MA, USA
{tszhang, jabellan, joshi, acoskun}@bu.edu

***Abstract*—Silicon-photonic network-on-chips (NoCs) provide high bandwidth density; therefore, they are promising candidates to replace electrical NoCs in manycore systems. The silicon-photonic NoCs, however, are sensitive to the temperature gradients that typically occur on the chip, and hence, require proactive thermal management. This paper first provides a design space exploration of silicon-photonic networks in manycore systems and quantifies the performance impact of the temperature gradients for various network bandwidths. The paper then introduces a novel job allocation technique that minimizes the temperature gradients among the ring modulators/filters to improve the application performance. Experimental results for a single-chip 256-core system demonstrate that our policy is able to maintain the maximum network bandwidth. Compared to existing workload allocation policies, the proposed policy improves system performance by up to 26.1% when running a single application and 18.3% for multi-program scenarios.**

I. Introduction

Silicon-photonic link technology is projected to replace the electrical links in future manycore NoCs. The primary motivation of using silicon-photonic links is that, compared to electrical links, silicon-photonic technology offers an order of magnitude higher bandwidth density. In addition, silicon-photonic NoCs have several times lower data-dependent energy consumption in long global on-chip interconnects, enabling the design of high-radix networks that are easier to program [1], [2]. However, a widespread adoption of the silicon-photonic link technology has not been possible due to high energy consumption in the thermal tuning of these devices and in the laser source that drives the silicon-photonic links.

In this paper, our goal is to minimize the need for localized thermal tuning in silicon-photonic NoCs to enable the adoption of silicon-photonic links in future manycore systems. The photonic devices at the transmitter and receiver side of a silicon-photonic link are highly sensitive to the temperature fluctuations. For reliable data transmission, the ring modulator at the transmitter side and the ring filter at the receiver side of the silicon-photonic link need to be at the same temperature so that their resonant wavelengths match. In a typical multicore chip, it is common to observe on-chip thermal gradients as large as 15-20^oC [3], which may result in a mismatch of the resonant wavelengths of the modulator and filter, and lead to unreliable data transmission. Localized thermal tuning mechanisms have been proposed to align the resonant wavelengths of the rings [4]. However, these mechanisms come with considerable power and performance overhead. Hence, it is critical to proactively manage the thermal gradients across the manycore system to achieve reliable communication at minimal local tuning cost.

A variety of approaches including dynamic voltage and frequency scaling (DVFS) [3], workload scheduling [3], [5] and liquid cooling [6] have been proposed for the thermal gradient management of manycore systems. These techniques aim to maximize performance by mitigating thermal hot spots and large thermal gradients in general; however, the application of these techniques for thermal gradient management of silicon-photonic NoCs have not been explored. We propose a job allocation technique that minimizes the thermal gradients specifically across the photonic devices in a silicon-photonic link, which in turn avoids the need for localized thermal tuning circuits. The contributions of our paper are as follows:

- We conduct a cross-layer design space exploration, where we consider the photonic device parameters, link transceiver circuit parameters, NoC architecture parameters, and software application requirements to determine the optimal design for the photonic devices under thermal constraints.
- We propose a novel thermally-aware job allocation policy that aligns the temperatures of thermally-sensitive ring modulators and filters in a photonic link, and at the same time, balances the overall chip temperature. The proposed policy outperforms existing thermally-aware job allocation policies by reducing the thermal gradients across the photonic devices in a silicon-photonic link to $< 2.2^oC$, which enables aggressive wavelength-division multiplexing. As a result, our method provides large NoC bandwidths and, as a result, improves application performance.
- We evaluate our policy on a single-chip 256-core system with a silicon-photonic Clos topology, running multi-threaded applications from SPLASH-2 [7] and PARSEC [8] suites. We demonstrate that our job allocation policy improves performance by up to 18.3% for multi-program workloads, compared to the best-performing baseline method. We also demonstrate that policies that solely minimize the temperature or the chip-wide gradients cannot sustain high application performance.

II. Related Work

As the number of cores per chip continues to increase, there is a need for a corresponding increase in the on-chip communication bandwidth to enable performance scalability. Silicon-photonic technology is considered as the future technology for manycore NoCs owing to its superior bandwidth density and lower power dissipation compared to conventional electrical NoCs. Recent research has explored a wide spectrum of network topologies for designing efficient silicon-photonic NoC architectures [1], [2], [9], [10].

For widespread adoption of silicon-photonic NoC architectures, one of the key challenges is energy-efficient thermal management of the silicon-photonic links. At the hardware level, athermal photonic devices have been proposed to reduce the localized tuning power in modulators/filters. These design-time solutions include using various materials such as cladding to reduce thermal sensitivity [11], using heaters [5] as well as temperature sensors for thermal control [12], and using a combination of ring resonators and Mach-Zehnder interferometers to provide athermal behavior [13]. These device-level techniques are promising; however, they either require costly changes in the manufacturing process or larger device areas that would decrease the network bandwidth density. In addition, such design-time solutions do not consider the run-time workload variations of the manycore system.

To reduce the overhead associated with localized tuning of individual rings, recent work leverages the group drift property of co-located rings and propose a method that trims a group of rings at the same time [14]. Channel re-mapping and calibration through dynamic feedback are other techniques to reduce required ring-tuning and to compensate for resonator thermal variations [15]. A run-time technique using thermal tuning to compensate for the thermal and process variation effects is another effective way of optimizing manycore system performance [16]. These techniques combine system-level and device-level management, but still rely on additional hardware.

There are also a number of techniques such as DVFS [3], workload migration [3], [5] and liquid cooling [6], which reduce the thermal hot spots and gradients on manycore systems. Differently from these techniques that optimize chip-level thermal behavior, our focus is to closely align the temperatures across specific silicon-photonic device locations so as to maximize the run-time NoC bandwidth. In our work, we consider both silicon-photonic device characteristics and the application behavior during thermal management, in addition to optimizing the chip's overall thermal behavior at no extra hardware cost.

III. Experimental Methodology

A. Manycore System Architecture

In this work, we use a 256-core manycore system fabricated using 22 nm SOI CMOS process, operating at 1 GHz with 0.9 V supply voltage. The architecture of each processor core is similar to the IA-32 core used in the Intel Single-Chip Cloud Computer [17]. Each core has 16 KB I/D L1 Cache & 256 KB Private L2 cache. We scale the core power and dimensions from 45 nm to 22 nm technology, resulting in a total chip area of 326.5 mm^2 (0.93 mm^2 per core, including L1, and 0.35 mm^2 for each L2 cache). The average per core power is 1.17 W. The cores are organized into 64 equal tiles. In each tile, four cores are connected via an electronic router. There are 16 memory controllers that are uniformly distributed along the two edges of the chip. We divide the chip into 8 zones with 8 tiles in each zone. We use a symmetric 3-stage Clos network topology for connecting the private L2 caches of the cores and the memory controllers. Our Clos can be described by the triplet (*m*=8, *n*=10, *r*=8), where *m* is the number of middle stage routers, *n* refers to I/O ports on first/last stage routers, and *r* is the number of first/last stage routers. As a result, the Clos is composed of 128 channels. At the center of each zone we place 3 routers, where each router is from a different network stage. Channels between the routers belonging to different zones are implemented through silicon-photonic links.

We map the logical Clos topology to a U-shaped physical layout of the photonic waveguides in the system (see Figure 1a). We use the silicon-photonic link technology described in prior work [18], [19], where photonic devices are monolithically integrated with CMOS devices. The rings and waveguides are made of $mono\ Si$ with SiO_2 as the surrounding material, and the photodetectors' material is Ge. Light waves emitted by an off-chip laser source are coupled into the photonic waveguides. These light waves pass next to a ring modulator that converts data from electrical medium to photonic medium. The modulated light waves travel along the waveguide and can pass through zero or more ring filters. At the receiver side, the light waves are filtered by wavelength matching ring filters and these light waves are incident on a photodetector. The current generated by the photodetector passes through electronic wires and is fed as input to the link receiver circuit.

B. Performance and Power Simulation

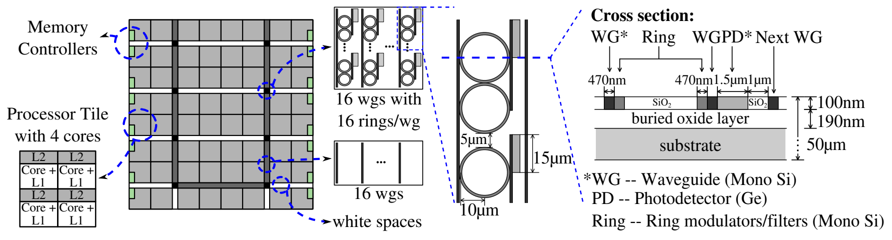
We use the Sniper [20] simulator and run a representative set of multi-threaded benchmarks from SPLASH-2 [7] (*barnes*, *ocean*, *radix*, *lu_contiguous*, *fft* and *water_nsquare*) and PARSEC [8] (*blackscholes*, *canneal* and *swaptions*) suites. We run the benchmarks with *sim_medium* inputs and focus on the parallel phases of their executions. To determine the impact of core thermal variations on the photonic devices under various workload utilization scenarios, we run each benchmark with 32, 64, 96, 128, 156, 180, 206, 230 and 256 threads.

We derive dynamic core power values for each benchmark for the corresponding number of threads using McPAT [21]. We calibrate the dynamic power numbers collected from McPAT based on the (scaled) power dissipation data published for Intel SCC [17]. At 70^oC, we assume the leakage power for the cores is 35% of the total average core power. While leakage power is exponentially dependent on temperature, practical studies indicate that linear models are sufficient for the temperature ranges observed on processors [22]. We derive a linear temperature-dependent leakage power model based on the reported data of Intel 22nm commercial processors. The leakage power model is $P_{Leak} = 0.0014T + 0.31$, where T is the temperature in ^oC. We assume idle cores enter into low power $sleep\ states$ that consume close to 0 W.

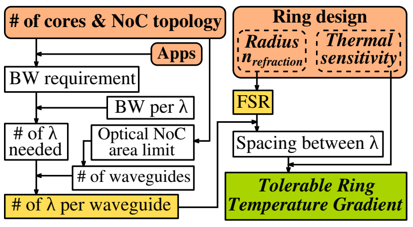
C. Thermal Modeling

We use HotSpot 5.02 [23] for our thermal simulations. We set the ambient temperature at 35^oC and use the default package configurations in HotSpot. The cross-sectional view of the target system is shown in Figure 1a. The photonic part in the system includes waveguides and a large number of ring modulators. Modeling every waveguide and ring modulator leads to long simulation times; thus, we aggregate the photonic devices in larger-sized blocks in the floorplan. Such aggregation methods provide desirable accuracy-simulation time tradeoffs in thermal simulation [6].

Each NoC block in our floorplan contains either only waveguides or both ring modulators/filters and waveguides, as shown in Figure 1a. We compute the joint thermal resistivity for each of these two types of blocks using $R_{joint} = V_{total}/\Sigma(V_i/R_i)$, where R_i and V_i refer to the thermal resistivity and volume of material i in the blocks. The joint thermal resistivity values of waveguide blocks and ring blocks are 0.01004 m-K/W and 0.01006 m-K/W, respectively, which



(a)



(b)

Fig. 1: (a) Target silicon-photonic system design and floorplan; (b) Silicon-photonic network design flow chart.

are approximately the same as the thermal resistivity of Si. Thus, we do not model the thermal resistivity heterogeneity inside the chip and use a single thermal resistivity value of 0.01 m-K/W across the die. The dimensions of our system are shown in Figure 1a. All the thermal results we report in this work are from steady state analysis, as we have not observed notable intra-application power variations.

IV. Design Space Exploration

To investigate the design space of silicon-photonic NoC, we adopt a cross-layer approach where we jointly consider the photonic device design, link transceiver circuit design, NoC architecture design, and performance characteristics of the benchmarks. Figure 1b shows the design flow adopted for jointly choosing the ring dimensions, the number of wavelengths per waveguide, and the number of waveguides for a given thermal gradient and area constraint. We consider area overhead as a constraint in the design flow because monolithic integration increases die area and, in turn, manufacturing cost.

We simulate the selected SPLASH-2 [7] and PARSEC [8] applications on our 256-core system and determine the peak NoC bandwidth (BW) requirement as 64 bits/cycle/channel. A silicon-photonic link with 2.5 Gbps per λ bandwidth has been demonstrated in prior work [18]. Hence, for the silicon-photonic link we consider three different bandwidths: 2 Gbps per λ, 4 Gbps per λ and 8 Gbps per λ. We expect the link bandwidth to increase to 4 Gbps per λ and 8 Gbps per λ following technology scaling and improvements in photonic device design. The link bandwidth and the bandwidth offered by the applications define the total number of wavelengths required in the silicon-photonic NoC.

We constrain the area of the photonic device to be at most 5% of the total die area. This constraint puts a lower limit on the number of wavelengths that need to be mapped to a waveguide. We consider three different radii for the rings: 5 μm, 10 μm and 20 μm. The rings are designed around a center wavelength (λ_0) of 1550 nm and they have a thermal sensitivity of 78 pm/K [24]. Using $\frac{\Delta\lambda}{\lambda_0} = \frac{\Delta f}{f_0}$ and $f_0 = \frac{c}{\lambda_0}$ = 193 THz, we get a sensitivity of a 9.7 GHz/K shift. The ring radii define the free spectral range (FSR). The thermal gradient constraint defines the spacing between the adjacent channels in the FSR, which in turn defines the number of wavelengths that can be mapped to a waveguide.

Figures 2 (a)-(c) shows the maximum temperature gradient that can be tolerated by the rings with radii 5μm, 10μm and 20μm, respectively. For each ring radius, we vary the link bandwidth per λ and the number of waveguides. However, we ensure that the total NoC bandwidth (*bandwidth per* λ * λ *per waveguide* * *number of waveguides*) is the same for all design points. For a given ring radius and bandwidth per λ, as we increase the number of waveguides, we can reduce the total number of wavelengths that we need to pack in each waveguide, which increases the maximum tolerable thermal gradient. Similarly, for a given ring radius and number of waveguides, as we increase the bandwidth per λ, we can reduce the number of λ per waveguide, which also results in an increase in the maximum tolerable thermal gradient. For fixed bandwidth per λ and number of waveguides, and hence a fixed number of wavelengths per λ, as we increase the ring radius the FSR decreases, which reduces the spacing between adjacent λ and decreases the maximum tolerable thermal gradients.

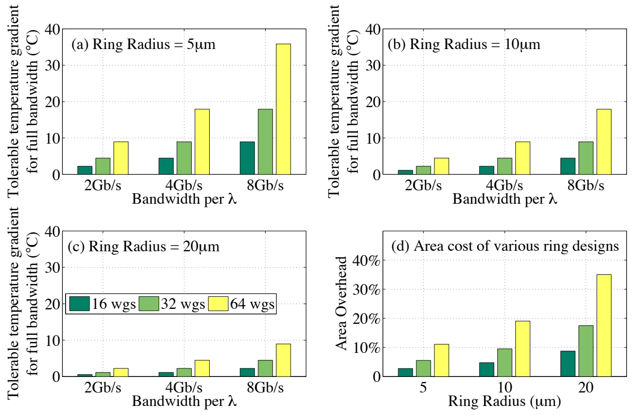


Fig. 2: Silicon-photonic NoC design space exploration

Figure 2(d) presents the area of the silicon-photonic NoC (normalized to the overall die area) for different ring designs and number of waveguides. As the ring radius and number of waveguides increase, the area overhead increases. For our system we use 10 μm as the ring radius, 16 waveguides, 64λ per waveguide and a projected bandwidth of 8 Gbps per λ. Assuming a waveguide loss of less than 2 dB/cm and nominal values for other losses, the laser power per waveguide is within the 30 mW non-linearity limit when we multiplex 64 wavelengths on a waveguide. The 10 μm results in a 1.4 THz Free Spectral Range (FSR), which gives a 21.5 GHz separation between adjacent λ. Hence, for 64λ in the FSR, which corresponds to 64-bit flit size in the network, we can tolerate a maximum of 2.2^oC thermal gradient among the rings. It is possible to perform NoC reconfiguration to decrease the flit size if the thermal gradient threshold is violated. After every NoC reconfiguration, the adjacent rings will be tuned to achieve a wider passband to manage Full Width at Half Maximum (FWHM). In our experiments, we consider four different photonic link widths (flit sizes) for the simulated Clos network: 64, 32, 16 and 8 bits. Note that such link bandwidths tolerate 0-2.2^oC, 2.2-4.4^oC, 4.4-8.8^oC and over 8.8^oC inter-ring temperature gradients, respectively.

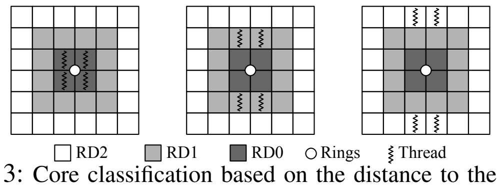


Fig. 3: Core classification based on the distance to the rings.

V. Ring-Aware Thermal Management Policy

We next describe our thermally-aware job allocation policy that minimizes the difference among the ring temperatures and, at the same time, reduces the overall chip temperature. As we specifically want to align the ring temperatures, our policy takes the ring locations into account during job allocation. In a silicon-photonic NoC, temperatures of rings are affected by the temperatures of cores that are in close proximity. To quantify this effect, we measure the ring temperature when four cores at various distances from the ring blocks are active and all the other cores are idle, as shown in Figure 3. We classify the ring block's neighboring cores with *RD#* (Ring Distance), where # represents the cores' relative distance to the ring block. We measure the temperature gradients among the ring blocks across the chip when we assign jobs to cores with different *RD#* values. When all cores in *RD0* of only one ring block are active, the temperature gradient among the rings across the chip increases to 7.5^oC. When four cores in *RD1* or *RD2* are active, the ring temperature gradient is $< 1^oC$. Thus, we propose a job allocation policy that maintains similar power dissipation across the *RD0* regions to minimize the gradients among the rings. We first design a policy that focuses solely on chip temperature minimization, *MinTemp*. We then design a *Ring-Aware* policy based on *MinTemp*; however, *Ring-Aware* explicitly takes the ring locations into account.

For the single-application case, we assume that there are n threads, each with the same average power dissipation, to be allocated on an m-core system. In *MinTemp*, we partition the system into four equal quadrants, and then assign $\lfloor \frac{n}{4} \rfloor$ threads to each quadrant. The residual threads, if any, are allocated to the quadrants in a round-robin fashion. In each quadrant, threads are first allocated to alternate cores (i.e., like a chessboard) on the two outer boundaries, starting from the corner core. Then, we continue to allocate threads to alternate cores in the next inner column and inner row of the quadrant. As the cores in the center of the chip are generally hotter compared to the outer cores, starting the job allocation from the outer cores helps reduce the temperature. In addition, allocating threads in a chessboard fashion of active and idle cores spreads the heat from the hotter cores to cooler cores. When the workload has $n > \frac{m}{2}$, then after first $\frac{m}{2}$ threads are assigned to alternate cores, the remaining threads are allocated to the idle cores, again starting from the outer cores. As we use an optical Clos NoC, the communication delay between the L2 caches and the memory controllers is agnostic of the core position. Thus, spreading the active threads across the chip does not introduce performance overhead.

To minimize the ring temperature gradients, we propose a *Ring-Aware* policy. This algorithm first categorizes cores based on their distance from the silicon-photonic rings. It then compares the number of threads we need to allocate against the total number of cores that are neither adjacent to the ring blocks nor are center cores. If the number of threads

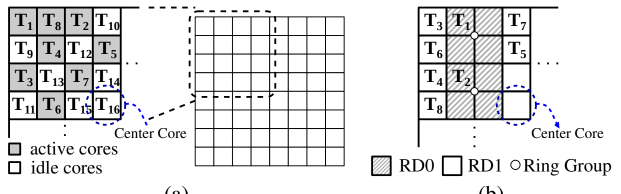


Fig. 4: (a) *MinTemp* and (b) *Ring-Aware* job allocation illustration.

is significantly lower, we keep all the *RD0* cores and center cores idle. If we have to utilize *RD0* and center cores, we maintain the same active core count among *RD0* regions for all ring groups in the system to minimize the ring gradient. After allocating the threads in the *RD0* regions, we allocate the rest of the threads to the other cores in the system according to the *MinTemp* policy, without disturbing the active core count across the *RD0* regions. In this way, the proposed policy minimizes the ring temperature gradient while reducing the temperature in the system. Balancing the absolute temperature of the processor chip with the temperature of the laser source is also necessary, but this is out of our scope.

For multi-program workloads, there are multiple applications running in the system and each application may have a different power level. We assume the applications' relative power levels (i.e., which application has higher power) are known *a priori*. This is a reasonable assumption as most applications run many times over the life-time of a system. We first sort all the threads according to their power dissipation. Then we allocate one application at a time, starting from the high power application, using the *Ring-Aware* policy. We start allocating the high power application first as *Ring-Aware* selects the outer cores initially to reduce system temperature. Our job allocation technique applies the same strategy if there are large power variations among a single application's threads. For a run-time implementation of the policy, performance counters such as number of instructions executed and cache misses can be leveraged as indicators of core power dissipation.

As an example, let us consider a 64-core system with an 8-ary 3-stage Clos optical network. When allocating 32 threads of a single application on this system, each quadrant is assigned 8 threads. Job allocation by *MinTemp* is shown in Figure 4 (a). The numbers in the figure represent the sequence in which the cores are activated. Job allocation for *Ring-Aware* is shown in Figure 4 (b), where striped blocks are the *RD0* cores and white blocks are the *RD1* cores. As there are only 8 threads in each quadrant, when we assign one thread to each *RD0* region, the other threads fit in the quadrant without having to activate the center cores. In this way, *Ring-Aware* maintains the same active core count in both of the *RD0* regions in the quadrant. The remaining threads are allocated using *MinTemp*, without violating the RD0 allocation restrictions. The same principles are repeated for the other quadrants.

VI. Experimental Results

This section evaluates the proposed thermal management policy on the 256-core system while running single-program and multi-program workloads. We first run benchmarks from PARSEC and SPLASH2 for 4 different flit sizes (64, 32, 16, and 8 bits) with various application thread counts (see Figure 5) to determine the impact of flit size and thread count on application performance. For a fixed thread count, the performance of all benchmarks saturates at a link bandwidth of 64

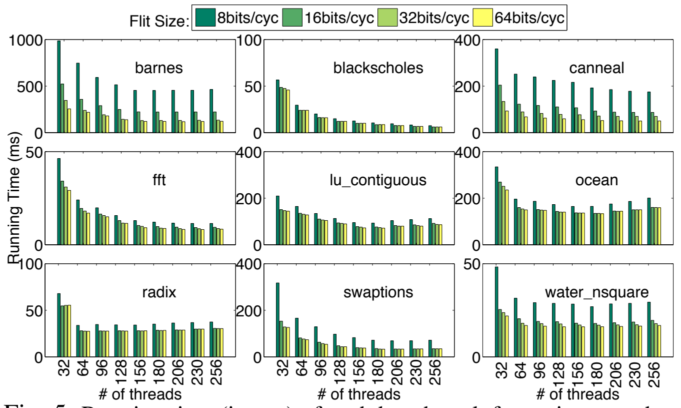


Fig. 5: Running time (in ms) of each benchmark for various number of threads and flit sizes.

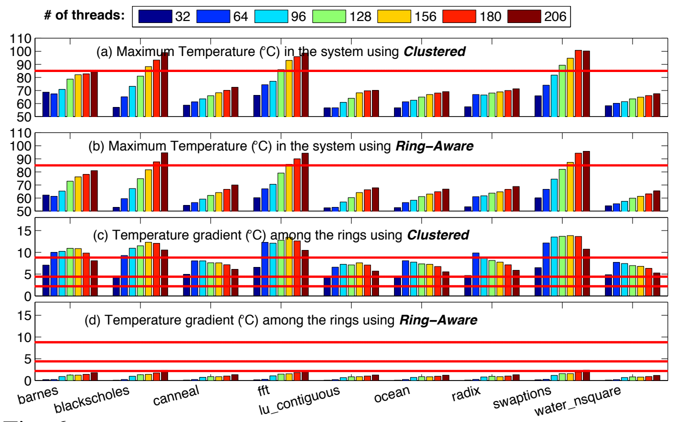


Fig. 6: System maximum temperature and ring temperature gradient using *Clustered* and *Ring-Aware*.

$bits/cycle$. However, the performance of *canneal* and *barnes* is highly sensitive to flit size, i.e., NoC bandwidth; hence, for these benchmarks it is desirable to have the ring temperature gradient minimized as much as possible. The figure also shows that the running times of *ocean*, *water_nsquare*, *radix* do not scale well with a higher number of threads. *lu_contiguous*, *swaptions* and *blackscholes* are more sensitive to the number of threads than the NoC bandwidth as they are CPU-bounded. As these benchmarks can effectively utilize a larger number of cores, they can substantially benefit from minimizing the maximum system temperature so that they can operate at the highest performance level without violating thermal thresholds. *barnes* and *fft* benefit both from higher NoC bandwidth and a larger number of active cores, motivating minimizing the ring thermal gradients and system temperature at the same time. *Ring-Aware* policy achieves these goals simultaneously.

A. Evaluation of Single-Application Workloads

We compare our *Ring-Aware* policy against three other allocation policies: *Clustered*, *Chessboard*, and *MinTemp*. The *Clustered* policy allocates the threads to the cores starting from one side of the chip and activates the cores in each column without leaving any idle cores. *Chessboard* policy allocates the threads to alternate cores starting from two opposite sides of the chip. For systems with more than 50% utilization, the threads are first allocated to alternate cores starting from two opposite chip sides, and then the additional threads are allocated starting from two chip sides to the idle cores.

Figure 6 (a)-(d) shows the maximum system temperature and ring temperature gradients for *Clustered* and *Ring-Aware* policies. We set the temperature threshold as 85^oC. Figure 6(a) shows that for the *Clustered* policy, 11 cases exceed the threshold, while 8 cases exceed the maximum temperature threshold for *Ring-Aware* policy. *Ring-Aware* reduces the system maximum temperature by 4.64 ^oC on average compared to *Clustered*. As illustrated in Figure 6(c), *Clustered* results in ring thermal gradients larger than 2.2^oC for all cases. Moreover, 49/50 cases have the ring temperature gradient larger than 4.4^oC and 24 of them result in gradients larger than 8.8^oC. In contrast, our proposed policy always maintains the thermal gradient below 2.2^oC for all the cases that do not violate the maximum temperature threshold (see Figure 6 (d)) and, as a result, enables the silicon-photonic NoC to operate at its full bandwidth.

TABLE I: Multi-Program Workloads

LL	water_nsquare (L), lu_contiguous (L)	**HH**	barnes (H), fft (H)
LH	barnes (H), lu_contiguous (L)	**LM**	canneal (M), ocean (L)
MM	radix (M), blackscholes (M)	**MH**	radix (M), swaptions (H)

Figure 5 and Figure 6 demonstrate that when using more than 50% of the cores, *canneal*, *barnes*, *water_nsquare*, *fft*, *swaptions* and *blackscholes* experience performance improvements of 84.7%, 269.2%, 17%, 36.6%, 123.4% and 23.4%, respectively, when using *Ring-Aware* instead of *Clustered*. This improvement is a result of maintaining ring gradients below 2.2^oC and system temperatures below 85^oC, thus, enabling using a larger number of cores and a larger NoC bandwidth. *ocean*, *lu_contiguous* and *radix* experience less than 10% performance improvement as these benchmarks are not highly sensitive to NoC bandwidth or thread count.

For the same workloads, *Chessboard* results in larger ring gradients compared to *Ring-Aware* for low utilization cases. For example, the ring gradient exceeds 2.2^oC for *barnes* at 32 threads, and for *fft* and *swaptions* at 64 threads. Consequently, there are performance losses of 26.1%, 6% and 1.6%, respectively. *MinTemp* keeps the ring gradients under 2.2^oC for the cases where the temperature threshold is not violated.

B. Evaluation of Multi-Program Workloads

We also evaluate our *Ring-Aware* policy for multi-program workloads on the 256-core system. As *Clustered* performs considerably worse than the other allocation methods, we focus on *Chessboard* ($Chess$), *MinTemp*, and *Ring-Aware* ($Ring$). In multi-program workloads, as there is higher variability in the power dissipated by various threads, how the specific threads are *mapped* to cores changes the thermal behavior. We implement the following thread mapping policies: in-order left ($Inorder$, which maps one application at a time, from left to right, onto the active cores), random mapping ($Rand$), and the multi-program support we design for our *Ring-Aware* policy ($Proposed$). Among various in-order mapping schemes, we select in-order left as it performs better on average compared to others. Results are shown in Figure 7.

Figure 7a shows the average ring gradients for a multi-program workload composed of two benchmarks for various thread allocation and mapping policies. The total system utilization is 50%, and the ratio of the active threads belonging

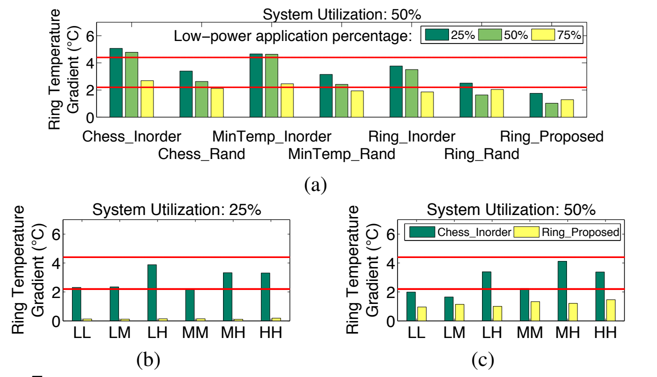


Fig. 7: Ring temperature gradients under various system utilizations for multi-program workloads.

to the lower power application varies as: 25%, 50% and 75%. We use *lu_contiguous* as the low-power application as it gives the lowest power in all these cases, and *barnes*, *fft*, *swaptions* as the high-power applications, as these are the highest power benchmarks for the three cases (i.e., for the corresponding thread counts), respectively. We run 200 cases of each $Rand$ mapping and report the average result. $Ring$ outperforms $Chess$ and $MinTemp$ for all mapping algorithms. $Chess$ has a gradient of more than 2.2^oC for all cases and $MinTemp$ achieves a gradient of less than 2.2^oC using $Rand$ only when the low-power benchmark is using 75% of the active cores. Both $Ring\_Rand$ and $Ring\_Proposed$ have gradients of less than 2.2^oC. However, $Rand$ mapping cannot provide guarantees, e.g., for $Ring\_Rand$ over 60% of the 200 runs exceed 2.2^oC constraint.

We also conduct thermal simulations for a diverse set of multi-program workloads as shown in Table I, and compare $Chess\_Inorder$ and $Ring\_Proposed$. Based on the average power dissipated (for 64-bit flit size), we categorize benchmarks as: low-power (L), medium-power (M), and high-power (H). We then create various combinations of L, M, and H. In this experiment, each application in a multi-program load has the same number of threads with the co-runner application. The results are shown in Figure 7b and 7c. When utilization is 50%, for 3/6 of the multi-program workloads the $Chess\_Inorder$ mapping results in a gradient $> 2.2^oC$, i.e., a lower NoC bandwidth. Thus, $Chess\_Inorder$ results in 6.1%, 13.8% and 8% lower performance compared to $Ring\_Proposed$ for LH, MH and HH, respectively. Our policy provides larger benefits when at least one of the application is high-power, as such applications create larger gradients.

VII. Conclusion

This paper has proposed a low-cost thermal management method for manycore systems with silicon-photonic NoCs. Using a 256-core system running multi-threaded applications, we have first quantified the impact of thermal gradients on the silicon-photonic NoC bandwidth and the application performance. We have then presented a novel job allocation policy that explicitly accounts for the physical locations of the photonic modulator/filter rings to minimize thermal gradients across those photonic devices. Our proposed method reduces the thermal gradients across the photonic modulator/filter rings to less than 2.2^oC and achieves the full NoC bandwidth, which improves performance by up to 26.1% and 18.3% in single-application and multi-program workloads, respectively, compared to existing policies.

Acknowledgment

We thank J. Klamkin for helpful discussions on photonic device design and K. Kawakami for his contributions to thermal modeling. This work has been partially funded by the NSF grants CNS-1149703 and CCF-1149549.

References

[1] A. Shacham, K. Bergman, and L. P. Carloni, "On the design of a photonic Network-on-Chip," in *Proc. NOCS*, 2007, pp. 53–64.

[2] A. Joshi *et al.*, "Silicon-photonic clos networks for global on-chip communication," in *Proc. NOCS*, 2009, pp. 124–133.

[3] A. K. Coskun, T. S. Rosing, K. A. Whisnant, and K. C. Gross, "Static and dynamic temperature-aware scheduling for multiprocessor SoCs," *Proc. IEEE Trans. on VLSI*, vol. 16, no. 9, pp. 1127–1140, 2008.

[4] P. Amberg *et al.*, "A sub-400 fJ/bit thermal tuner for optical resonant ring modulators in 40 nm CMOS," in *Proc. IEEE ASSCC*, 2012, pp. 29–32.

[5] X. Zhou, J. Yang, M. Chrobak, and Y. Zhang, "Performance-aware thermal management via task scheduling," *ACM Transactions on Architecture and Code Optimization*, vol. 7, no. 1, pp. 5:1–5:31, 2010.

[6] A. K. Coskun, J. L. Ayala, D. Atienza, and T. S. Rosing, "Modeling and dynamic management of 3D multicore systems with liquid cooling," in *IFIP/IEEE International Conference on VLSI-SoC*, 2009, pp. 35–40.

[7] S. C. Woo *et al.*, "The SPLASH-2 programs: characterization and methodological considerations," in *Proc. ISCA*, 1995, pp. 24–36.

[8] C. Bienia *et al.*, "The PARSEC benchmark suite: Characterization and Architectural Implications," in *Proc. PACT*, 2008, pp. 72–81.

[9] Y. Pan *et al.*, "Firefly: Illuminating future Network-on-Chip with nanophotonics," in *Proc. ISCA*, 2009, pp. 429–440.

[10] L. Ramini, D. Bertozzi, and L. Carloni, "Engineering a bandwidth-scalable optical layer for a 3d multi-core processor with awareness of layout constraints," in *Proc. NOCS*, 2012, pp. 185–192.

[11] S. S. Djordjevic *et al.*, "CMOS-compatible, athermal silicon ring modulators clad with titanium dioxide," *Optics Express*, vol. 21, no. 12, pp. 13 958–13 968, 2013.

[12] C. T. DeRose *et al.*, "Silicon microring modulator with integrated heater and temperature sensor for thermal control," in *Proc. CLEO*, 2010, pp. 1–2.

[13] G. Biswajeet *et al.*, "CMOS-compatible athermal silicon microring resonators," *Optics Express*, vol. 18, no. 4, pp. 3487–3493, 2010.

[14] C. Nitta, M. Farrens, and V. Akella, "Addressing system-level trimming issues in on-chip nanophotonic networks," in *Proc. HPCA*, 2011, pp. 122–131.

[15] Y. Zheng *et al.*, "Power-efficient calibration and reconfiguration for on-chip optical communication," in *DATE*, 2012, pp. 1501–1506.

[16] Z. Li *et al.*, "Reliability modeling and management of nanophotonic on-chip networks," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 1, pp. 98–111, 2012.

[17] J. Howard *et al.*, "A 48-core IA-32 processor in 45 nm CMOS using on-die message-passing and DVFS for performance and power scaling," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 1, pp. 173–183, 2011.

[18] B. Moss *et al.*, "A 1.23pJ/b 2.5Gb/s monolithically integrated optical carrier-injection ring modulator and all-digital driver circuit in commercial 45nm SOI," in *ISSCC*, 2013, pp. 126–127.

[19] J. S. Orcutt *et al.*, "Open foundry platform for high-performance electronic-photonic integration," *Opt. Express*, vol. 20, no. 11, pp. 12 222–12 232, May 2012.

[20] T. E. Carlson *et al.*, "Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulations," in *Proc. SC*, 2011, pp. 1–12.

[21] S. Li *et al.*, "McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures," in *Proc. MICRO-42*, 2009, pp. 469–480.

[22] H. Su *et al.*, "Full chip leakage-estimation considering power supply and temperature variations," in *Proc. ISLPED*, 2003, pp. 78–83.

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

[24] P. Dong *et al.*, "Wavelength-tunable silicon microring modulator," *Optics Express*, vol. 18, no. 11, pp. 10 941–10 946, May 2010.