Adaptive Tuning of Photonic Devices in a Photonic NoC Through Dynamic Workload Allocation

José L. Abellán, *Member, IEEE*, Ayse K. Coskun, *Member, IEEE*, Anjun Gu, *Student Member, IEEE*, Warren Jin, Ajay Joshi, *Member, IEEE*, Andrew B. Kahng, *Fellow, IEEE*, Jonathan Klamkin, *Senior Member, IEEE*, Cristian Morales, John Recchio, *Student Member, IEEE*, Vaishnav Srinivas, *Member, IEEE*, and Tiansheng Zhang, *Student Member, IEEE*

***Abstract*—Photonic network-on-chip (PNoC) is a promising candidate to replace traditional electrical NoC in manycore systems that require substantial bandwidths. The photonic links in the PNoC comprise laser sources, optical ring resonators, passive waveguides, and photodetectors. Reliable link operation requires laser sources and ring resonators to have matching optical frequencies. However, inherent thermal sensitivity of photonic devices and manufacturing process variations can lead to a frequency mismatch. To avoid this mismatch, micro-heaters are used for thermal trimming and tuning, which can dissipate a significant amount of power. This paper proposes a novel *FreqAlign* workload allocation policy, accompanying an adaptive frequency tuning (*AFT*) policy, that is capable of reducing thermal tuning power of PNoC. *FreqAlign* uses thread allocation and thread migration to control temperature for matching the optical frequencies of ring resonators in each photonic link. The *AFT* policy reduces the remaining optical frequency difference among ring resonators and corresponding on-chip laser sources by hardware tuning methods. We use a full modeling stack of a PNoC that includes a performance simulator, a power simulator, and a thermal simulator with a temperature-dependent laser source power model to design and evaluate our proposed policies. Our experimental results demonstrate that *FreqAlign* reduces the resonant frequency gradient between ring resonators by 50%–60% when compared to existing workload allocation policies. Coupled with *AFT*, *FreqAlign* reduces localized thermal tuning power by 19.28 W on average, and is capable of saving up to 34.57 W when running realistic loads in a 256-core system without any performance degradation.**

***Index Terms*—Optical tuning, silicon photonics, thermal management.**

Manuscript received November 19, 2015; revised March 14, 2016 and June 1, 2016; accepted July 24, 2016. Date of publication August 12, 2016; date of current version April 19, 2017. This work was supported by the NSF under Grant CNS-1149703 and Grant CCF-1149549. This paper was recommended by Associate Editor L. P. Carloni. *(Tiansheng Zhang is the primary student author of this paper.)*

J. L. Abellán is with the Department of Computer Science, Catholic University of Murcia, 30107 Murcia, Spain.

A. K. Coskun, A. Joshi, C. Morales, and T. Zhang are with the Department of Electrical and Computer Engineering, Boston University, Boston, MA 02215 USA.

A. Gu, J. Recchio, and V. Srinivas are with the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093 USA.

W. Jin and J. Klamkin are with the Department of Electrical and Computer Engineering, University of California at Santa Barbara, Santa Barbara, CA 93106 USA.

A. B. Kahng is with the Department of Computer Science and Engineering and the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2016.2600238

I. Introduction

As core count increases in manycore systems to support the ever-increasing thread-level parallelism exhibited by applications, the network-on-chip (NoC) bandwidth (BW) must correspondingly increase to maximize application performance. Future application domains (e.g., cyber-physical and big data) are expected to require even larger NoC BWs. At sufficiently large BWs, photonic links can improve energy-per-bit performance over electrical links with their higher BW density (Gb/s $\cdot$ μm$^-1$), lower global communication latency, and lower data-dependent power [1]–[5].

A typical silicon photonic link consists of: 1) a laser source to emit optical waves; 2) a ring modulator and a ring filter to modulate optical waves at the transmitter and filter them at the receiver, respectively; 3) a passive waveguide to propagate optical waves; and 4) a photodetector to convert optical signals into electrical signals. Silicon photonic links require the optical frequency of the laser source powering that link to match with the resonant frequencies of its associated ring modulators and filters. In a photonic NoC (PNoC), the ring resonators are typically placed close to the cores to reduce the delay and energy of the electrical link connecting the cores to the silicon photonic link transmitter and receiver, but resonant frequencies of these ring resonators are sensitive to temperature. Variations in core power consumption can alter a ring's temperature and introduce data transmission errors (i.e., increasing the link error rates) or even break the link entirely.

In PNoC, thermal tuning via micro-heaters [6] is commonly used to match the resonant frequencies of ring resonators, and the frequencies of ring resonators with those of laser sources. However, this method induces significant power overhead when there are large optical frequency mismatches between ring resonators and corresponding laser sources. Thus, low overhead tuning methods that can match the optical frequencies of both ring resonators and laser sources are required. On-chip laser sources are good candidates for reducing thermal tuning power due to their close proximity to the PNoC control circuits [7]. Their proximity allows for runtime control over their optical frequencies, which allows for more flexible tuning methods for photonic devices to be implemented.[1]

In this paper we propose *FreqAlign*, a new workload allocation policy, and an adaptive frequency tuning (*AFT*) policy, which work together to match the optical frequencies of the on-chip laser sources and ring resonators in a PNoC to minimize tuning power. *FreqAlign* first spatially assigns workloads to cores in a manycore system to achieve an on-die temperature gradient that minimizes the difference among the resonant frequencies of the ring resonators. *AFT* then locally tunes the temperatures of ring resonators and laser sources for the remaining differences in their optical frequencies. The main contributions of this paper are as follows.

1) We provide a full modeling stack of performance, power, and thermal simulations for a manycore system with a PNoC, including a temperature-dependent laser source power model.
2) We propose a novel workload allocation policy, *FreqAlign*, which performs significantly better than our previously proposed workload allocation policy, *RingAware* [8], in matching the resonant frequencies of the ring resonators, even in the presence of process variations and for various PNoC logical topology and physical layout combinations.
3) We propose *AFT*, a tuning policy to control the optical frequencies of on-chip laser sources adaptively based on the temperatures of ring resonators at runtime to reduce the tuning power of manycore systems with PNoC.
4) We demonstrate that when running real workloads, *FreqAlign* reduces the resonant frequency difference between ring resonators in the same photonic link by 50%–60% compared to *RingAware* without any performance degradation. Proposed *AFT* reduces thermal tuning power by as much as 34.57 W when compared to the baseline tuning policy.

The rest of this paper starts with a review of related work in Section II. Section III presents our target manycore system and PNoC design flow, and we introduce our simulation infrastructure in Section IV. Section V describes our proposed workload allocation and tuning policy. Section VI provides experimental evaluation and Section VII concludes this paper.

II. Related Work

Silicon photonics is a promising technology to support the increasing demand for energy-efficient and high-BW on-chip communication in future manycore systems. Compared to an electrical NoC (ENoC), a PNoC can provide higher BW density with lower data-dependent power dissipation. Thus, designing an energy-efficient PNoC has been widely explored [1]–[5].

A major challenge in designing an energy-efficient PNoC is the large thermal tuning power overhead, which adversely affects PNoC energy efficiency. Photonic devices such as ring resonators and laser sources are sensitive to temperature. The optical frequencies of these components need to match to ensure link signal integrity. There has been some effort at the technology level to counter this thermal challenge, including the introduction of negative thermo-optic coefficient materials to compensate for the positive thermo-optic coefficient of silicon [9]. However, the technology surrounding these athermal devices is immature and demonstrates thermal insensitivity over only a limited temperature range. Another method incorporates a heater with a temperature sensor for localized thermal tuning of ring resonators [6]. Other hardware implementations of wavelength locking including balanced homodyne locking [10] and programmable locking and routing using a field programmable gate array (FPGA) [11] have been demonstrated as well. From the chip stack design perspective, inserting an insulation layer between logic and photonic layers can decouple temperatures of these layers [12].

To compensate for the tuning power overhead, one method is to tune a group of ring resonators simultaneously instead of traditional single ring tuning, at the cost of additional hardware support [13]. Our prior *RingAware* workload allocation policy [8] balances the temperature of the ring groups (RGs) without using extra hardware, but this policy does not consider the impact of process variation on the ring resonators, nor does it attempt to match the optical frequencies of the laser sources and ring resonators. Aurora [14] leverages localized tuning and workload allocation techniques and embodies a cross-layer approach at the device, architecture, and OS levels. At the device level, Aurora controls small temperature variations by applying a bias current through the ring resonators [15]. For larger temperature changes, packets are rerouted away from hot regions, and dynamic voltage and frequency scaling is applied to reduce the temperature of hot areas. At the OS level, a job allocation policy prioritizes jobs to the outer cores of the chip.

A common drawback among all these techniques is that there is no focus on matching the optical frequencies between on-chip laser sources and ring resonators under varying system utilizations. Moreover, prior methods do not account for the impact of temporal temperature variations on the thermal tuning power. In this paper, we propose *FreqAlign*, a dynamic workload allocation policy combined with *AFT* policy, to match the resonant frequencies of the ring resonators with the optical frequencies output by their associated laser sources. Compared to solely balancing ring temperatures, our policy requires much lower thermal tuning power under various ring placements, process variation scenarios, and system layouts.

III. Manycore Systems With PNoC

To design a manycore system with PNoC, the requirements and constraints for both electronic components and photonic devices must be considered. In this section, we introduce the architecture of our target manycore system and PNoC design flow. Table I shows the notations used in this paper.

A. Manycore System Architecture

We use a 256-core system designed using a typical 22 nm SOI CMOS process, operating at 1 GHz with 0.9 V supply voltage. For each core, we use an architecture similar to the IA-32 core from Intel Single-chip Cloud Computer (SCC) [16]. Every core consists of a 16 KB I/D $L1$ cache and a 256 KB private $L2$ cache. We scale the core architecture to

[1]Using off-chip laser sources to power silicon photonic links is another option, and may provide better temperature stability and higher operating efficiency than on-chip laser sources. However, the lack of runtime control makes them less flexible for optical frequency tuning.

TABLE I
NOTATIONS USED IN THIS PAPER

Variable	Definition	Unit
i	Ring group index	
j	Core index	
k	Thread index	
l	Laser source index	
s	Simulation step index	
g	Material index	
R	Thermal resistivity	$m \cdot K/W$
R_{joint}	Thermal resistivity of the NoC block	$m \cdot K/W$
$\Delta f_R/\Delta f_{LS}$	Thermal sensitivity of ring resonators / laser sources in frequency domain	GHz/K
$\Delta\lambda_R/\Delta\lambda_{LS}$	Thermal sensitivity of ring resonators / laser sources in wavelength domain	pm/K
η_R/η_{LS}	Thermal tuning efficiency of ring resonators / laser sources	W/K
n_g	Refractive index of the ring resonator material	
r	Ring resonator radius	μm
V	Volume	m^3
M	Total number of ring groups	
H	Total number of rings in a ring group	
N	Total number of cores	
S	Total number of threads	
Q	Total number of laser sources	
n_λ	Number of wavelengths per waveguide	
x	Number of middle stage routers	
y	Number of I/O ports on first or last stage routers	
z	Number of first or last stage routers	
w	Weight factor	K/W
w_{ij}	Weight factor for ring group i and core j for temperature impact	K/W
Δw	Weight factor difference	K/W
t	Time	ms
T	Temperature	^oC
T_{RG_i}	Temperature of ring group i	^oC
T_{LS_l}	Temperature of laser source l	^oC
P_j	Power of core j	W
P_{FT}	Optical frequency tuning power	W
P_{leak}	Leakage power	W
λ	Wavelength	nm
F	Frequency	GHz
F_{RG_i}	Frequency of ring group i	GHz
F_{LS_l}	Frequency of laser source l	GHz

22 nm, resulting in a single core area of 0.93 mm^2 (including the $L1$ cache), and an $L2$ cache area of 0.35 mm^2. Our total chip area[2] is 326.5 mm^2. The average power consumption for each core is 1.17 W. The system is organized into 64 equal tiles. In each tile, four cores are connected via an electrical router. There are 16 memory controllers that are uniformly distributed along two edges of the chip. We use an 8-ary 3-stage Clos network topology to connect the $L2$ caches and memory controllers. Our Clos can be described by the triplet ($x = 8$, $y = 10$, $z = 8$), where x is the number of middle stage routers, y is the number of I/O ports on the first or last stage routers, and z is the number of first or last stage routers. Therefore, the 8-ary 3-stage Clos PNoC has 128 channels in total.

We map the 8-ary 3-stage Clos topology to a U-shaped physical layout of silicon photonic waveguides as shown in Fig. 1(a), where each RG is assigned to the nearest eight tiles and two memory controllers. We apply the silicon photonic link technology described in prior work [18]–[20], where photonic devices are monolithically integrated with CMOS devices. In this system, single crystal *Si* is utilized for waveguides and ring resonators, and *Ge* on *Si* is utilized for photodetectors. Ring resonators are designed in *Si* by ion implantation and are tuned with metal heaters. We combine the ring modulators and filters from one electrical router of each of the three network stages into a RG. The optical waves from laser sources arrive at an RG and are modulated. The modulated optical waves traverse the network and are filtered by the ring filters in the destination RG, where a photodetector converts the optical signal into an electrical current that is fed to the link receiver circuit. Prior work [18]–[20] employs off-chip laser sources. In this paper, we assume on-chip laser sources, which simplify packaging, reduce cost and improve laser source control. Several approaches have been proposed for realizing on-chip laser sources [22]–[24]. Specifically, our discussion below assumes heterogeneous integration to incorporate laser sources above the logic and silicon photonic devices. Such a monolithic approach can be cost effective because it does not require separate fabrication of laser sources and would avoid chip attachment steps that require precise alignment.

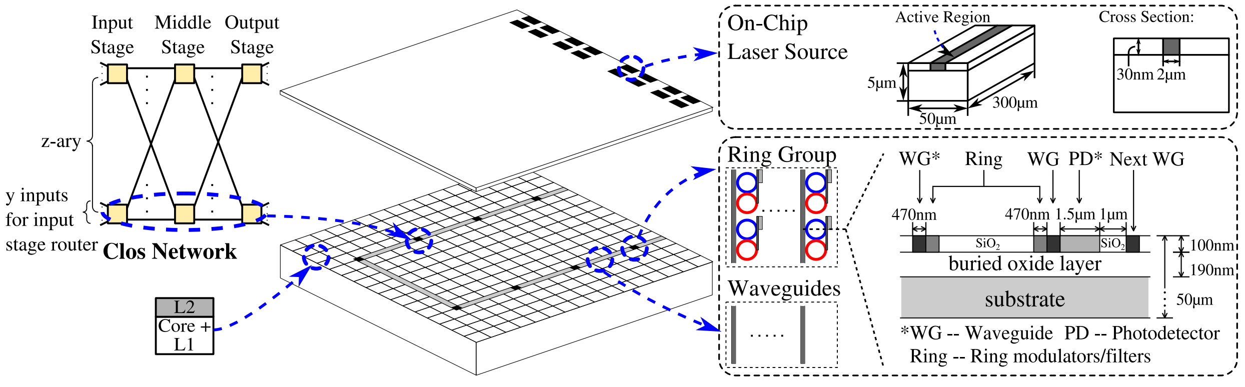
Alternative core and cache architectures may require different logical topology and physical layout combinations, so we also test systems with other layouts and RG locations shown in Fig. 1(b)–(e). Fig. 1(b) and (c) shows two layouts that use the same logical topology but have horizontal and vertical shifts, respectively, in RG locations. Fig. 1(d) presents a system with 16-ary 3-stage logical topology and a W-shaped physical layout. Fig. 1(e) shows a rectangular chip with 8-ary 3-stage logical topology and chain-shape physical layout.

[2]There are commercial products with similar die size and power consumption, e.g., SPARC T4 processor [17].

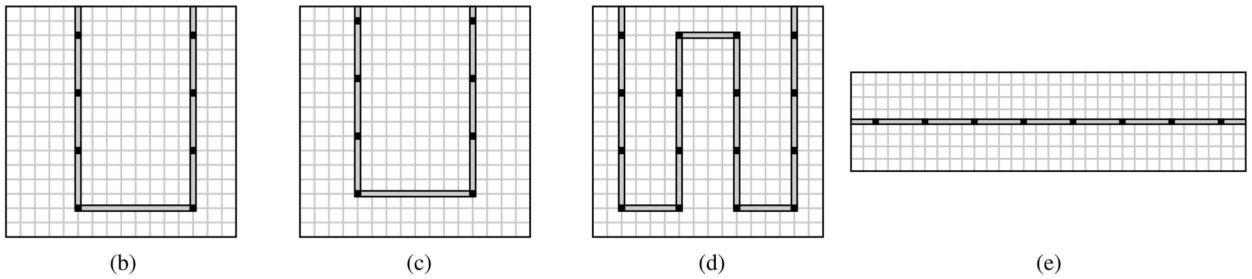
B. PNoC Design Flow

Designing a PNoC for a manycore system has many factors to consider, e.g., BW requirement and area constraints of the target system, data rate of the optical waves as well as the design of ring resonators. To investigate the design space of a PNoC, we adopt a cross-layer approach where we jointly consider the photonic device design and NoC architecture design. Fig. 2 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, resulting in increased manufacturing cost.

The BW requirement of a PNoC depends on targeted applications in a manycore system. In this paper, we simulate selected SPLASH-2 [25], PARSEC [26], and UHPC [27] applications on our manycore system and determine the peak NoC BW requirement to be 512 GB/s, which corresponds to 64 bits/cycle for each photonic channel in our 8-ary 3-stage Clos network. A monolithically integrated silicon photonic link with 2.5 Gb/s $\cdot$ λ^{-1} BW has been demonstrated in prior work [19]. In this paper, we assume a BW of 4 Gb/s $\cdot$ λ^{-1}. This is reasonable considering the performance of current silicon photonic devices that operate beyond 25 Gb/s [28]. The link BW and the required BW of the applications define the total number of wavelengths needed in the PNoC. We constrain PNoC area to be at most 10% of the total die area. This constraint puts an upper limit on the number of waveguides in the system and thus a lower limit on the number of wavelengths that need to be mapped to a waveguide. We ignore the nonlinearity limit on the power that can be injected into a waveguide [2]. However, our proposed policy is applicable



(a)



(b) (c) (d) (e)

Fig. 1. Target manycore system with a (a) U-shape layout of PNoC, (b) and (c) manycore systems with 8-ary 3-stage Clos topology and shifted physical layouts, and (d) and (e) manycore systems with different logical topology and physical layout combinations. (d) is designed with 16-ary 3-stage Clos topology and W-shape physical layout; (e) is designed with 8-ary 3-stage Clos topology and chain-shape physical layout.

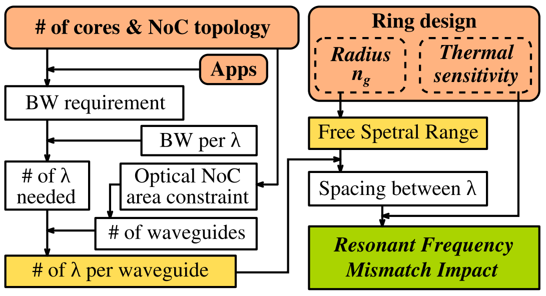


Fig. 2. PNoC design flow chart.

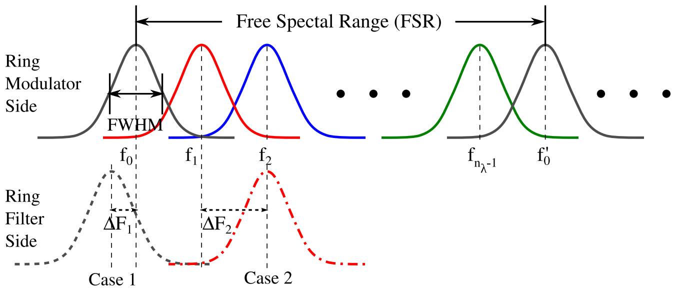


Fig. 3. Impact of resonant frequency mismatch. Case 1: small mismatch reduces the filtered optical power. Case 2: large mismatch may result in a ring to filter the data of its neighboring ring in the frequency domain.

even if we account for waveguide nonlinearity while designing a PNoC.

For the ring resonator design, we choose 10 μm as the radius. The ring resonators are designed around a center wavelength (λ_0) of 1550 nm and have a thermal sensitivity ($\Delta\lambda_R$) of 78 pm/K [18], which translates to a 9.7 GHz/K frequency shift (Δf_R) based on equations (1) and (2).

$$F_0 = \frac{c}{\lambda_0} = 193 \text{ THz} \quad (1)$$

$$\frac{\Delta\lambda_R}{\lambda_0} = \frac{\Delta f_R}{F_0} \quad (2)$$

This means that for every degree of temperature gradient between a ring modulator and ring filter in a link, there is a 9.7 GHz mismatch in resonant frequency.

The spacing between adjacent wavelengths depends on the free spectral range (FSR) of a ring resonator design and the number of wavelengths per waveguide (n_λ), as shown in the following equations:

$$\text{FSR} = \frac{c}{2\pi r n_g} \quad (3)$$

$$F_{\text{spacing}} = \frac{\text{FSR}}{n_\lambda} \quad (4)$$

where n_g is the group index, c is the speed of light, and F_{spacing} is the spacing in resonant frequency for two adjacent wavelengths in a waveguide. The impact of resonant frequency mismatch is shown in Fig. 3, where FWHM represents full width at half maximum. When the mismatch is small, a ring filter receives only a portion of the signal power, resulting in less current from the photodetector and causing data loss (case 1). As the mismatch increases, a ring filter may even filter the optical waves corresponding to its neighboring resonant frequency (case 2).

Within each RG in a PNoC, there are ring resonators with varying resonant frequencies belonging to different silicon photonic links. Each silicon photonic link is multiplexed with other links on a waveguide and has one ring modulator (on the transmitter side) in one RG and a ring filter (on the receiver side) with the same resonant frequency in another RG. For the

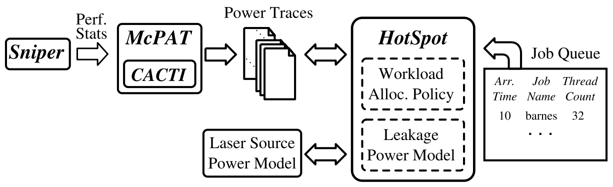


Fig. 4. Our performance [29], power [30], [31], and thermal [32] simulation setup for modeling manycore systems with a PNoC.

sake of convenience, in the rest of this paper, we refer to "resonant frequencies of ring resonators within an RG" as "the resonant frequency of an RG." We also use "resonant frequency difference between two RGs" to represent "resonant frequency difference between a ring modulator in one RG, and the corresponding ring filter in the other RG".

For systems without process variations, a corresponding resonant frequency difference between two RGs can be computed using the temperature gradient (ΔT) between them: $\Delta F = \Delta T \times \Delta f_R$. Due to manufacturing process variations, there are variations in the dimensions of the waveguides across a chip. Since the resonant frequency is very sensitive to these dimensions, there is an initial gradient in frequency across RGs. Thus, temperature alone cannot accurately indicate the frequency difference among RGs.

IV. Experimental Methodology

To investigate thermal conditions and corresponding optical frequency variations of ring resonators and laser sources at runtime when running realistic workloads on a manycore system with PNoC, we set up a simulation infrastructure composed of performance, power and thermal simulators, as shown in Fig. 4. We use Sniper [29] to simulate performance. Sniper comes interfaced with McPAT [30] (integrated with CACTI [31]) to estimate the power consumption of the simulated system. The power traces generated by McPAT are given as inputs to the HotSpot 3-D extension (hereafter, HotSpot) [32], [33] for transient thermal simulations.

A. Performance and Power Simulation

For performance simulations, we simulate the region of interest of a representative set of multithreaded applications from the SPLASH-2 [25] (barnes, lu_cont, and water_nsq), PARSEC [26] (blackscholes and canneal), and UHPC [27] (md and shock) benchmark suites. To investigate the impact of core thermal variations on the photonic devices under varying system utilizations, we run each application on a target manycore system (explained in Section III-A) with 32, 64, 96, and 128 threads.

We use the performance statistics from Sniper as input to McPAT to calculate power for cores and caches. After generating all power traces, we use published power dissipation data from Intel SCC [16], scaled to 22 nm, to calibrate our dynamic power data. HotSpot takes these power traces as inputs, and outputs corresponding temperature traces. We assume that idle cores are put into sleep states and consume 0 W. We also assume that 35% of the average core power (1.17 W) at 70 °C comes from leakage [33]. We calculate the average core power consumption in one core for each application and categorize the applications as shown in Table II. We compose

TABLE II
Classification of Applications

High Power (HP) Apps	md (2.15 W), shock (1.7 W)
Medium Power (MP) Apps	blackscholes (1.46 W), barnes (1.3 W)
Low Power (LP) Apps	canneal (0.9 W), water_nsq (0.7 W), lu_cont (0.75 W)

and evaluate different workload combinations based on this categorization in Section VI.

To simulate thermal behavior of the cores more accurately, we implement a linear leakage power model inside HotSpot. This model is suitable due to the relatively limited range in the operating temperature on our target system [34]. We use published data for Intel 22 nm commercial processors [35] to extract this linear leakage power model as shown in (5). In this equation, $T(t_{s-1})$ is the temperature in °C at time t_{s-1} and $P_{\text{leak}}(t_s)$ is the leakage power in W at time t_s, where s is the thermal simulation step index and t_s is the time at which the leakage power is recalculated. c_1 and c_2 are constant coefficients with values 1.4e−3 and 0.31, respectively. During thermal simulations, we update the leakage power for every core based on its temperature at t_{s-1}.

$$P_{\text{leak}}(t_s) = c_1 \times T(t_{s-1}) + c_2 \qquad (5)$$

A novel part of our simulation infrastructure is modeling the laser source power consumption at runtime as a function of temperature, and including this model in our HotSpot transient thermal simulations. Previous work [21] implemented a temperature-dependent laser source power model for steady state thermal simulations. We put together a similar framework that works with both steady state simulations and transient simulations. In our framework, we generate a lookup table for laser power by employing the theory described in prior work [36]. The laser source power that contributes to heat dissipation is the difference between the required input electrical power and the required output optical power. The required input electrical power depends on the required output optical power and the laser source efficiency. The required output optical power is determined by the optical loss during optical wave transmission in the PNoC, and thus is fixed for a given PNoC design. The laser source efficiency is based on the required output optical power and laser source temperature. Thus, the lookup table takes required output optical power and laser source temperature as inputs, and computes the required input electrical power based on the corresponding laser source efficiency. During transient thermal simulations, we update the laser source power at the beginning of each simulation step for each laser source based on its temperature.

B. Thermal Simulation

To implement dynamic workload allocation policies in our thermal simulations, we enable HotSpot to read the upcoming jobs from a job queue, in which each job entry has an arrival time, an application name, and a required thread count. We also integrate a workload allocation module in HotSpot. When a job arrives, this module allocates the threads to cores. HotSpot assigns a power value for each core at each simulation step based on the specific thread it runs (assigned from a power trace database generated via Sniper-McPAT). Thread migration can be applied to this framework as needed.

TABLE III
PROPERTIES OF THE MATERIALS IN OUR TARGET SYSTEM

	Thickness (mm)	Side (mm)
Heat Sink	6.9	80
Spreader	1	40
Interface Material	0.02	
Laser Source Layer	0.005	
Core Layer	0.05	
	Thermal Conductivity ($W \cdot m^{-1} \cdot K^{-1}$)	Specific Heat ($J \cdot g^{-1} \cdot K^{-1}$)
Spreader	400	
Interface Material	4	
Si	100	0.71
InP	68	0.31
	Photonic Device Dimensions	Material
Laser Source Size	$300\ \mu m \times 50\ \mu m$	InP
Ring Radius	$10\ \mu m$	Si

In our HotSpot setup, we use the default configuration with 35 °C ambient temperature, and the properties of the materials shown in Table III. The floorplans of the target systems are shown in Fig. 1. For our system, we assume monolithic integration of waveguides, ring resonators and photodetectors on the logic layer [18], while laser sources are on a separate layer. On the laser source layer, the laser sources are placed along the upper chip edge, arranged in two groups surrounding the waveguides in a matrix fashion, as shown in Fig. 1(a).

The number of laser sources depends on the design choice of laser source type, sharing degree, and required network BW. Sharing a laser source among multiple waveguides has been shown to improve laser source efficiency and reduce total on chip power [37]. We choose 32 waveguides for our PNoC design to allow for laser source sharing between waveguides while remaining within the 10% area overhead maximum given for the photonic devices in our system.

We aggregate waveguides, ring resonators and photodetectors into larger simulation blocks in our floorplan in HotSpot as in [8]. We calculate the joint thermal resistivity for each PNoC block based on the percentage of each material's volume and thermal resistivity of each material using $R_{\text{joint}} = V_{\text{total}}/\Sigma(V_g/R_g)$, where V_{total} represents the total volume of a PNoC block, R_g refers to the thermal resistivity of material g, and V_g indicates the volume of material g in this PNoC block. R_{joint} of the ring blocks is 1.006e−2 m · K/W, while R_{joint} for the waveguide blocks is 1.004e−2 m · K/W (both are almost identical to the thermal resistivity of *Si*).

Transient thermal simulations are first initialized with a steady state simulation. As we add a temperature-dependent leakage model that changes the power traces, we run each transient thermal simulation for another round to ensure convergence of temperature.

V. OPTICAL FREQUENCY TUNING THROUGH WORKLOAD ALLOCATION AND LOCALIZED TUNING

In a PNoC, the power needed to tune laser sources and ring resonators depends on the optical frequency difference among these devices. This frequency difference is caused by temperature variations and process variations. Process variations depend on the quality of the manufacturing process while the temperature variations are highly dependent on the workload distribution in the manycore system. The thermal dependence between workload distribution and optical device frequency control power is shown in Fig. 5. Our target is to change

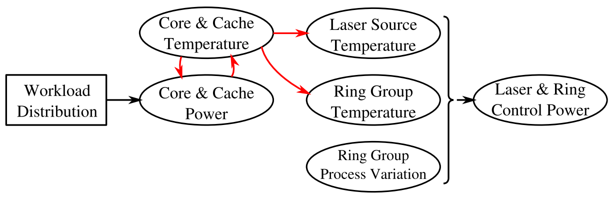


Fig. 5. Thermal dependence between workload distribution and optical device frequency control power.

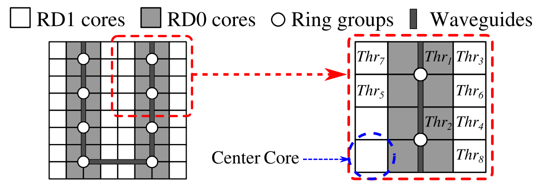


Fig. 6. (a) Classification of *RD*0 cores and (b) an example of *RingAware* allocation in a 64-core system.

the chip thermal map through workload allocation to reduce the resonant frequency difference among all the RGs. On top of this, we propose an *AFT* method to match the remaining optical frequency difference between laser sources and RGs.

In Section V-A, we first describe our previously proposed *RingAware* thermal management policy [8], and then propose an improved RG location aware policy, *FreqAlign*. In Section V-B, we introduce a baseline frequency tuning policy and propose a novel *AFT* policy for manycore systems with on-chip laser sources. We discuss the performance overhead of the proposed workload allocation policy in Section V-C.

A. Workload Allocation Policies

1) RingAware: The *RingAware* workload allocation policy balances the RG temperatures by maintaining similar power profiles around each RG. For a given layout, this policy categorizes cores based on the distance of the core from its closest RG. We use *RD*# notation for each region, where # represents the cores' relative distance to the RG, as shown in Fig. 6. Since *RD*0 cores have the highest impact on an RG's temperature, *RingAware* maintains similar power dissipation across the *RD*0 regions for all RGs to minimize their temperature gradients.

We use single-threaded cores and each workload is composed of S threads. For an N-core system with M RGs, if there are S threads to allocate, we first compare S with the total number of *non-RD0* cores. If S is larger, the *RD0* cores need to be utilized to run all the threads and we assign $\lceil (S - (N - \#RD0\text{Cores}))/M \rceil$ threads to each *RD0* region. The *RD0* regions of all RGs need to have the same active core count to minimize the RG temperature gradient. Then, we partition the system into four quadrants and then assign the rest of the threads evenly in each quadrant. The residual threads, if any, are allocated to the quadrants in a round-robin fashion. For each quadrant, *RingAware* activates *non-RD0* cores alternately from the outer boundaries to the inner part of the chip (i.e., to reduce chip temperature) until all threads are allocated, starting from the corner core, as shown in Algorithm 1. If there are power variations among threads, we rank the threads according to their power consumptions at the beginning and

Algorithm 1: Pseudocode for *RingAware* [8] Policy

```
Identify RD0 cores $\rightarrow$ RD0 core list;
Partition the system into 4 quadrants;
Sort all threads based on their power consumption;
if $S > N - \#RD0Cores$ then
    foreach ring group in the system do
        assign $\lceil \frac{S-(N-\#RD0Cores)}{M} \rceil$ threads;
    end
    Each quadrant $\leftarrow \frac{S-\lceil \frac{S-(N-\#RD0Cores)}{M} \rceil * M}{4}$ threads;
else
    Each quadrant $\leftarrow \frac{S}{4}$ threads;
end
foreach quadrant in the system do
    foreach thread left in queue do
        $allocatedCore \leftarrow 0$;
        $nextThread \leftarrow 0$;
        foreach alternative core j on boundary do
            if core j is idle & core j $\notin$ RD0 core list then
                $allocatedCore \leftarrow j$;
                $nextThread \leftarrow 1$;
                break;
            end
        end
        if $nextThread == 0$ then
            foreach alternative core j in inner area do
                if core j is idle & core j $\notin$ RD0 core list
                then
                    $allocatedCore \leftarrow j$;
                    $nextThread \leftarrow 1$;
                    break;
                end
            end
        end
    end
end
```

start the allocation process with the high-power threads in the order (Thr_1–Thr_8) shown in Fig. 6(b).

RingAware allocation effectively reduces the RG temperature gradient, which results in a low resonant frequency gradient when the system does not have process variations. For systems with process variations, only balancing the temperatures of RGs is not sufficient to reduce the resonant frequency difference among RGs. Also, when the RGs are not symmetrically placed on the chip, *RingAware* starts to require larger thermal tuning power. Hence, we now propose an improved policy that jointly accounts for thermal variations and process variations. This policy works even for asymmetric placement of RGs.

2) FreqAlign (Proposed Workload Allocation Policy): The goal of *FreqAlign* workload allocation policy is to reduce the resonant frequency difference among RGs. To do this, we estimate the RG resonant frequency for every potential workload allocation decision by estimating the RG temperatures. In a system that has M RGs and N cores, for each RG, we use an $M \times N$ weight matrix of w_{ij} that contains the steady state temperature impact per unit of power of core j on RG i. This weight matrix is used to estimate the temperature of RGs.

Algorithm 2: Pseudocode for *FreqAlign* Policy

```
Sort all threads based on their power consumption;
foreach thread in queue do
    $\Delta w_{min} \leftarrow -1$;
    $allocatedCore \leftarrow 0$;
    foreach available core j in manycore system do
        foreach ring group i in manycore system do
            $w_{est} \leftarrow w_{curr} + core\ j\ impact\ on\ RG\ i$;
        end
        $\Delta w_{est} \leftarrow \max(w_{est}) - \min(w_{est})$;
        if $\Delta w_{est} < \Delta w_{min}$ or $\Delta w_{min} == -1$ then
        $\Delta w_{min} \leftarrow \Delta w_{est}$;
        $allocatedCore \leftarrow j$;
        else continue
    end
    $allocatedCore\ in\ coreArray \leftarrow active$;
    $w_{curr} \leftarrow w_{curr} + core(allocatedCore)\ impact\ on\ RGs$;
end
```

For example, if core j has a weight factor of 0.5 for RG i, it means at steady state, RG i's temperature increases by 0.5 K when core j consumes 1 W. This weight matrix can be obtained using HotSpot for a given physical layout.

When calculating the shift in the resonant frequencies of all ring resonators in RG i due to temperature change, we use (6) and (7), where ${F_{\text{RG}_i}}^{\text{post}}$ and ${T_{\text{RG}_i}}^{\text{post}}$ are the resonant frequency and temperature, respectively, of RG i after the updated workload allocation. Correspondingly, ${F_{\text{RG}_i}}^{\text{pre}}$ and ${T_{\text{RG}_i}}^{\text{pre}}$ are the resonant frequency and temperature of RG i before the updated workload allocation. Δf_R is 9.7 GHz/K, P_j is the power value of core j, and w_{ij} is the weight factor of core j to RG i.

$$ {F_{\text{RG}_i}}^{\text{post}} = {F_{\text{RG}_i}}^{\text{pre}} - \Delta f_R \times \left({T_{\text{RG}_i}}^{\text{post}} - {T_{\text{RG}_i}}^{\text{pre}}\right) \quad (6) $$

$$ {T_{\text{RG}_i}}^{\text{post}} = \sum_{j=1}^{N} w_{ij} \times P_j \quad (7) $$

Algorithm 2 shows the pseudocode of the proposed *FreqAlign* policy. Here, we define a job as an application with a number of threads to be allocated (our target system has single-threaded cores, so we can only assign one thread per core), and we put the threads into a queue and allocate them to the available cores in the manycore system. The objective function of the optimization is to minimize the sum of the absolute differences in resonant frequencies of all the RGs ($\sum_{i=1}^{M-1} \sum_{i'=i+1}^{M} |F_{\text{RG}_i} - F_{\text{RG}_{i'}}|$).

Every RG has a designed resonant frequency value F_i^0. Due to process variations, this value varies depending on the RG location. The variations in the resonant frequency values could be diagnosed after the chip is manufactured. We maintain a resonant frequency array (w_{curr} in Algorithm 2) for the RGs during the system operation. This array contains the initial values of w which depend on the resonant frequency shift of each RG caused by its process variations. For example, an initial array of $[5, 0, 0, 0, 0, 0, 0, -5]$, means that RG_1 has a resonant frequency $5\ \text{K} \times 9.7\ \text{GHz/K} = 48.5$ GHz lower than the designed frequency while RG_8 has a resonant frequency 48.5 GHz higher than the designed frequency. Every time a core is activated, we update this array based on the impact of the core on these RGs. Our target in workload allocation is to equalize the values in this array.

During the system operation, when an application with S threads arrives, we rank the threads based on their power consumption (which can be estimated through previous runs or performance counters history) and assign them to the cores while balancing the resonant frequency of the RGs. After all S threads are allocated to the corresponding cores, the system starts to run. As Algorithm 2 shows, when assigning the threads, we go through all the available cores in the system. For each available core, we calculate the expected resonant frequency difference among all RGs if a thread is assigned to that core. For each thread, we select the core that results in the smallest resonant frequency variation among all RGs ($\Delta w_{\min}$). After assigning a thread, we update the estimated resonant frequency values for all RGs. We iterate this process until all threads are assigned. If there are jobs currently running on the manycore system and a new job arrives, we rank the new threads and the existing threads together according to their power consumption and redo the workload allocation. The potential workload migration when redoing the allocation induces context switch overhead and cache cold start effect to the system. *FreqAlign* can be integrated with the operating system scheduler and run on any available core in the system. We discuss the performance overhead of *FreqAlign* in Section V-C.

B. Frequency Tuning Methods

1) Baseline Frequency Tuning: Workload allocation can help decrease the resonant frequency difference among the RGs. We use localized tuning to compensate for the remaining resonant frequency difference as well as the optical frequency difference between RGs and laser sources. Resonant frequencies of ring resonators can be controlled through thermal tuning devices such as micro-heaters. As for on-chip laser sources, their optical frequencies can be controlled in a number of ways, depending on the laser source type. For example, multisection distributed Bragg reflector laser sources comprise of wavelength tuning control elements such as mirrors and a phase section. The wavelengths of distributed feedback lasers, which we use in this paper, are controlled by injecting current. More advanced laser sources on silicon photonic platforms may comprise of extra ring filters within the laser cavity that can be also used for tuning.

Our baseline frequency tuning method is *Target Frequency Tuning (TFT)*. In this tuning method, at any given time during system operation, all RGs and laser sources are first tuned to their optical frequencies at the temperature threshold of the target manycore system (90 °C in our case), and then are individually tuned further to compensate for process variations to match their optical frequencies. We also assume that all the ring resonators within an RG share the same temperature. Since the material used for the laser sources and ring resonators have different thermo-optic coefficients, their respective tuning efficiencies (8 mW/nm [38] for the laser sources and 2.6 mW/nm [18] for the ring resonators) also differ. The temperature sensitivity values for laser sources and ring resonators are 12.5 GHz/K [39] and 9.7 GHz/K [40], respectively. For a fixed target optical frequency, the amount of frequency tuning power (P_{FT}) required is shown in (8),

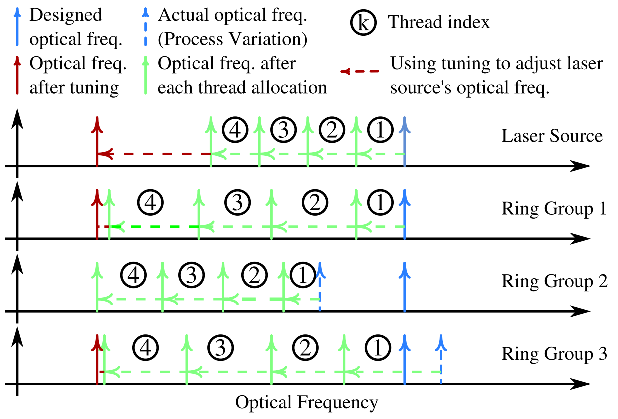

Designed optical freq.
Actual optical freq. (Process Variation)
k Thread index
Optical freq. after tuning
Optical freq. after each thread allocation
Using tuning to adjust laser source's optical freq.
Laser Source
Ring Group 1
Ring Group 2
Ring Group 3
4
3
2
1
Optical Frequency

Fig. 7. Illustration of *FreqAlign* workload allocation policy and *AFT* policy. Every thread allocated by *FreqAlign* increases the temperatures of RGs and causes a downward shift in their frequencies. When all threads are allocated, thermal tuning is used to bring all RGs to the lowest common resonant frequency. Above, RGs 1 and 3, as well as the laser source, are tuned to match the resonant frequency of RG 2.

where F_{LS_l} is the frequency of laser source l, $F_{\mathrm{target}_{i/l}}$ is the desired target optical frequency of a photonic device i/l at the target temperature, Δf_{LS} is the thermal sensitivity of the laser source, η_{LS} is the tuning efficiency of the laser sources, F_{RG_i} is the frequency of RG i, Δf_R is the thermal sensitivity of the ring resonators, η_R is the tuning efficiency of a single ring resonator, Q is the total number of laser sources, M is the total number of RGs, and H is the number of ring resonators in an RG.

$$P_{\mathrm{FT}} = \sum_{l=1}^{Q} \frac{\left|F_{\mathrm{LS}_l} - F_{\mathrm{target}_l}\right|}{\Delta f_{\mathrm{LS}}} \times \eta_{\mathrm{LS}} + \sum_{i=1}^{M} \frac{F_{\mathrm{RG}_i} - F_{\mathrm{target}_i}}{\Delta f_R} \times \eta_R \times H \tag{8}$$

2) Adaptive Frequency Tuning (Proposed Tuning Policy): TFT tunes all RGs and laser sources in PNoC to a target optical frequency. Using this method, the total tuning power depends directly on the sum of differences between the optical frequency of optical devices and the target frequency. When the system is underutilized, the tuning power becomes significant due to the low average temperature. Since in this paper we use on-chip laser sources, which provide a much shorter control loop compared to off-chip laser sources, we propose a new tuning method to match the optical frequency of laser sources and ring resonators, called *AFT*. In this tuning method, we set the lowest frequency among the RGs as the target frequency and tune all the other devices to this target frequency, because RGs' resonant frequencies change with their temperatures, the target frequency is chosen adaptively based on the current lowest resonant frequency among all RGs. Therefore, the tuning power depends on the combination of relative differences between the lowest resonant frequency among the RGs and the optical frequencies of other optical devices. As a result, *FreqAlign* requires lower power consumption for optical frequency tuning. Fig. 7 shows an example of *FreqAlign* workload allocation and *AFT*.

C. Performance Overhead Analysis

The performance overhead of *FreqAlign* policy is composed of two parts: 1) the execution time of *FreqAlign* and 2) the potential thread migration overhead in a manycore system. To evaluate the execution time of *FreqAlign*, we carried out an offline experimental analysis considering the worst-case scenario of allocating one thread to each core in the target 256-core system. For our analysis, we implement *FreqAlign* in C programming language, compile it using *gcc* with −O3 flag, and run it on Sniper. The simulation results show that the allocation of 256 threads to 256 cores takes a total of 192 μs.

Whenever a new job enters the system, thread allocation in *FreqAlign* also involves a reallocation process, in which we migrate the existing threads if necessary. Thread migration may hurt application performance due to the context switch and the cache cold start effect. We use Sniper along with its hardware thread migration scheme [41] to investigate the impact of thread migration overhead on our target system's performance. Once the pipeline of the core a thread is originally running on has been drained, its architectural state is transferred to the destination core. The destination core then starts running the thread and endures the cache cold start effect. The overhead from migrating a thread from one core to another core includes three major components: 1) a fixed penalty of 1000 cycles for storing and restoring the core's architectural state [41]; 2) the time to drain the source core's pipeline prior to migration; and 3) the cache cold start effect. As quantified in several previous studies [41], [42], cache cold start effect is the dominant component in migration overhead and can be two orders of magnitude larger than the other two components combined. In our thread migration scheme, there is no flushing of the source core's caches. Every cache miss in the destination core sends a memory request to the source core's $L2$ cache instead of memory. This lowers the number of memory accesses. Any source core's $L2$ cache block that is in shared/modified state triggers a writeback/invalidation using the normal cache coherency protocol.

As thread migration overhead varies with both application workload and the number of threads needed to be migrated, we carry out a comprehensive experimental evaluation that considers all applications under varying number of threads used in this paper (further details in Section VI). We configure Sniper with multiple thread migration intervals (time slice after which a thread migrating process occurs): 500 μs, and 1 and 10 ms. For each interval case, we configure the migration percentage, i.e., the number of threads that actually migrate: 0.0 (baseline case without thread migration), 0.05 (5% of the threads migrate), 0.1, 0.25, 0.5, and 1.0 (all threads migrate). We compare the migration cases with the baseline cases to calculate the average cycle count per migration for each of the combinations. From the results, we observe a maximum of 147.3 μs (14.3 μs on average) increment in running time due to thread migration. The running times of our applications with native input size vary from hundreds of milliseconds to seconds [43]. Real-life running times for similar scientific applications vary from minutes to hours. As *FreqAlign* executes only when a new job arrives at the system, we conclude that it entails negligible performance overhead.

TABLE IV
WORKLOAD COMBINATIONS. HP: HIGH-POWER; MP: MEDIUM-POWER; AND LP: LOW-POWER

Workload	Job 1	Job 2
HPHP	md	shock
HPMP	md	blackscholes
HPLP	shock	lu_cont
MPMP	barnes	blackscholes
MPLP	barnes	water_nsq
LPLP	lu_cont	canneal

Algorithm 3: Pseudocode for *Clustered* Policy

```
Sort all threads based on their power consumption;
foreach thread in queue do
    allocatedCore ← 0;
    for j = 1 to N do
        if core j is available then
            allocatedCore ← j;
            break;
        end
    end
end
```

TABLE V
RUNNING TIMES OF JOBS (UNIT: MS)

Thread count	md	shock	black-scholes	lu_cont	barnes	water_nsq	canneal
32	320	397	30	140	237	17	72
64	295	309	24	128	218	16	68
96	221	205	16	104	158	16	62
128	167	188	12	89	139	16	59

VI. EXPERIMENTAL RESULTS

To demonstrate the benefits and scalability of *FreqAlign*, we conduct experiments using systems with different logical topology/physical layout combinations and process variations and compare *FreqAlign* with two other policies: 1) assignment of threads starting from the lowest indexed core (*Clustered*, shown in Algorithm 3) and 2) *RingAware* (shown in Algorithm 1). In our target system, the cores are indexed from left to right and from bottom to top. There are 32 waveguides in the PNoC, and the data rate for each wavelength is 4 Gb/s. Our design of experiments contains the following cases.

1) Six workload combination cases using two different jobs (see Table IV) at the same time: HPHP, HPMP, HPLP, MPMP, MPLP, and LPLP; job 1 arrives at 1 ms and job 2 arrives at 2 ms after the start of each simulation. Jobs have different running times (see Table V), and the shorter job repeats itself until the longer job finishes execution.
2) Six utilization cases: 25% (job 1: 32 cores + job 2: 32 cores), 50% ($32 + 96$, $64 + 64$, $96 + 32$ cores), 75% ($96 + 96$ cores), and 100% ($128 + 128$ cores).
3) One case without process variations and four cases with process variations in different directions.
4) Five logical topology and physical layout combinations as shown in Fig. 1(a)–(e).

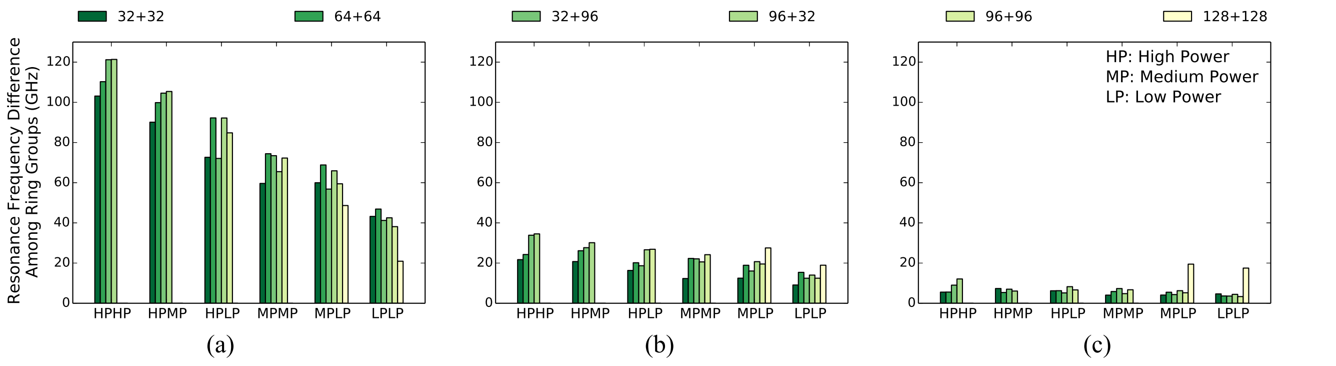


Fig. 8. Average resonant frequency differences when using (a) *Clustered*, (b) *RingAware*, and (c) *FreqAlign* workload allocation policy for U-shape layout with 8-ary 3-stage Clos topology shown in Fig. 1(a). Each bar represents a workload and utilization combination case.

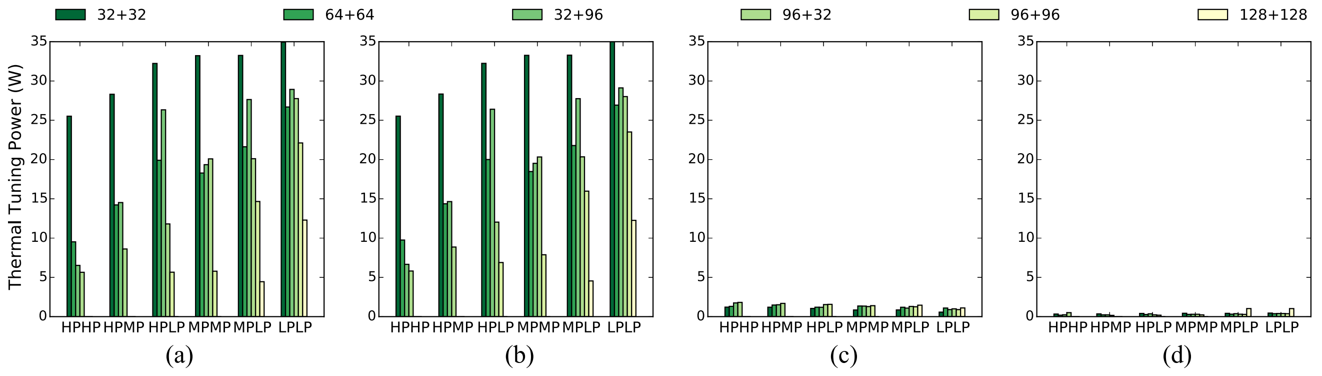


Fig. 9. Average optical tuning power when using localized tuning for *Clustered*, *RingAware*, and *FreqAlign* workload allocation policy for 8-ary 3-stage Clos topology with U-shape layout in Fig. 1(a). (a) Clustered + TFT. (b) RingAware + TFT. (c) RingAware + *AFT*. (d) FreqAlign + *AFT*

A. Optical Frequency Tuning Evaluation

We compare the optical frequency difference and required tuning power for the three policies. Fig. 8 shows the resonant frequency differences among the RGs for the three policies using a U-shape 8-ary 3-stage Clos PNoC [as shown in Fig. 1(a)]. We can see from this figure that *Clustered* results in highest resonant frequency gradient among all three policies. *FreqAlign* achieves a 60.6% reduction in the resonant frequency difference on average as compared to *RingAware*. This is because *RingAware* only focuses on cores that are closest to RGs (RD0 cores), but the aggregation of non-RD0 cores still has a huge impact on RG temperature. *FreqAlign* estimates the impact of allocating a thread to a core on all RG temperatures, which reduces the resonant frequency difference among all RGs.

In Fig. 9, we present the thermal tuning power when applying different workload allocation policies and tuning mechanisms. Here, we do not show the cases (e.g., HPHP with 128+128 threads) in which the maximum on-chip temperature is higher than the temperature threshold, 90 °C. This rule also applies to all the other figures in this section. Since TFT requires every RG and laser source to be tuned to the resonant frequency at 90 °C, the required tuning power only depends on the absolute operating temperatures of the photonic devices. Under such scenarios, temperature balancing techniques without proper tuning strategies do not show advantage on reducing thermal tuning power. Thus *Clustered* and *RingAware* have similar required tuning power. *AFT*, on the other hand, tunes the laser source frequency to align with the lowest of the current resonant frequencies of RGs, which is balanced through the proposed workload allocation policy. *FreqAlign+AFT* saves 19.28 W thermal tuning power on average and up to 34.57 W compared to *RingAware+TFT*. This result demonstrates that there is a need for proper control of the on-chip laser source and ring resonator optical frequency tuning mechanism.

Apart from edge-placed laser sources, we also test the proposed policy and baseline policies for the same manycore chip, but this time with locally-placed laser sources, where on-chip laser sources are placed around the RGs along the U-shape waveguide. We observe similar percentages of thermal tuning power reduction for *FreqAlign*. For off-chip laser sources, due to the lack of runtime control, *AFT* is not applicable in this scenario. Thus, systems with off-chip laser sources have similar ring resonator thermal tuning power as the cases with TFT [as shown in Fig. 9(a) and (b)].

B. Case Study on Process Variation

Ring resonators are sensitive to process variations, and the resonant frequency can vary approximately linearly with distance on the scale of waveguide length [44]. To study the impact of process variations, we consider a wavelength variation of 400 pm/cm due to process variations by considering a linear process variation of 0.76 nm [45] over our approximately 1.9 cm long chip.[3] In this case study, we consider four process variation directions as shown in Fig. 11: 1) horizontal; 2) vertical; 3) diagonal which results in largest process variations among RGs; and 4) diagonal which results in largest process variations across the chip. We evaluate both

[3]While we are using linear process variation for our case study, *FreqAlign* comprehends unique process variation parameters for each RG in a system, so any pattern of process variation between RGs could be considered.

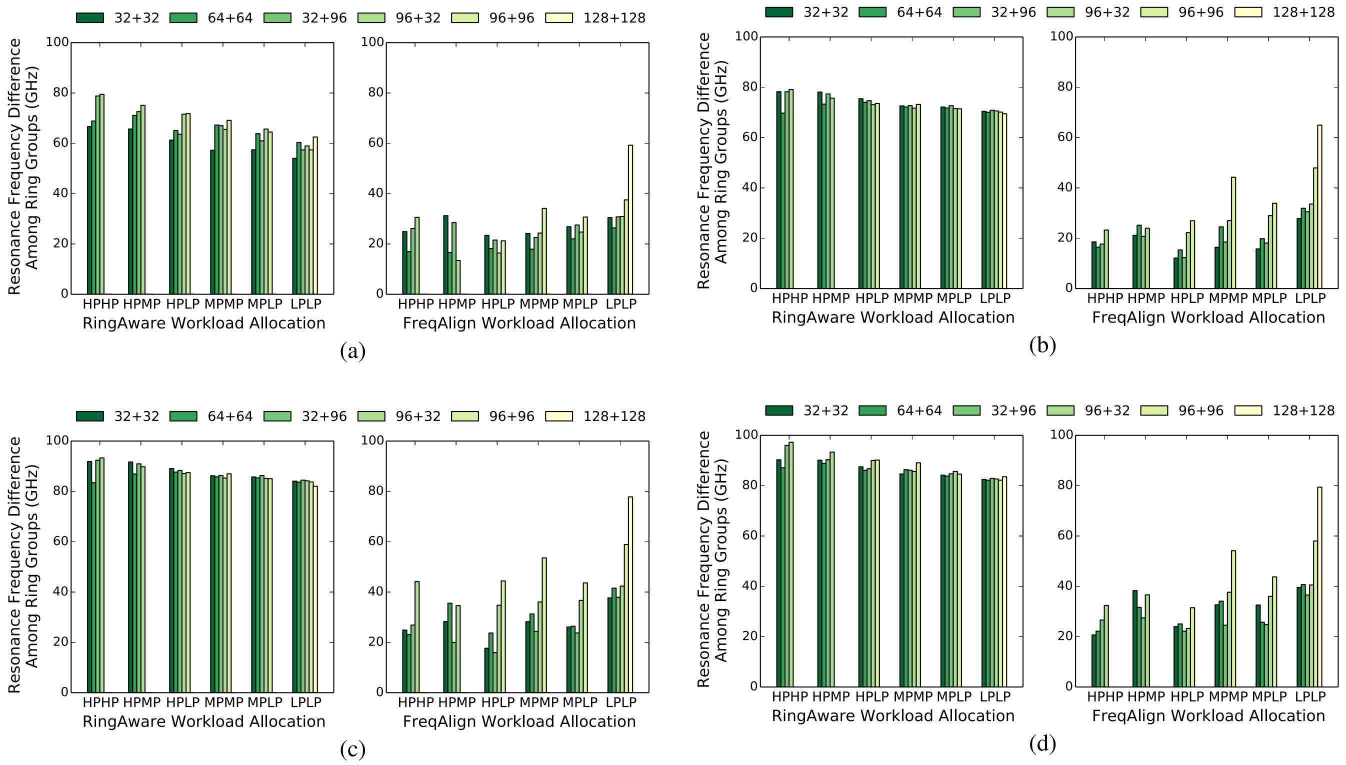


Fig. 10. Average resonant frequency difference when comparing *RingAware* and *FreqAlign* workload allocation policy for U-shape layout with 8-ary 3-stage Clos topology and a wavelength variation of 400 pm/cm in multiple directions due to process variations. The process variation cases and tuning power reduction of *FreqAlign+AFT* compared to *RingAware+AFT* are shown in the captions of subfigures. (a) Horizontal process variation gradient. Average (maximum) power reduction is 1.83 W (2.77 W), 62.9% (82.5%). (b) Vertical process variation gradient. Average (maximum) power reduction is 1.81 W (2.91 W), 64.7% (84.3%). (c) Maximum process variation among RGs. Average (maximum) power reduction is 2.04 W (3.26 W), 60.1% (80.8%). (d) Maximum on-chip process variation. Average (maximum) power reduction is 2.12 W (3.13 W), 61.4% (76.2%).

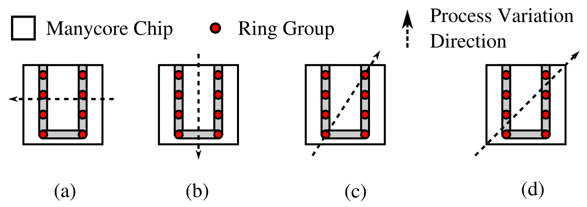


Fig. 11. Process variation directions considered for case study. (a) Horizontal. (b) Vertical. Max. process variations (c) among RGs and (d) across the chip.

RingAware and *FreqAlign* policies, with the assumed process variation directions. The results in Fig. 10 show that *FreqAlign* reduces the resonant frequency difference by 52.7% on average compared to *RingAware*. This is because *RingAware* is designed to balance the operating temperatures among the RGs, and does not account for process variations. On the other hand, *FreqAlign* considers both temperature and process variations, so it significantly reduces the average resonant frequency difference.[4]

C. Case Study on Layout Sensitivity

In addition to process variations, we also evaluate *FreqAlign* for various PNoC layouts. We conduct this case study in two aspects: 1) using the same PNoC logical topology and physical layout combination but slightly shifted RG locations and 2) using different PNoC logical topology and physical layout combinations.

For case (1), we test the two layouts with shifted RG placements in Fig. 1(b) and (c). When using these two different layouts, *FreqAlign* has 60% and 50.7% reduction in resonant frequency difference, respectively, compared to *RingAware* [see Fig. 12(a) and (b)] since it considers the estimated resonant frequencies of RGs and their placement, when allocating the jobs. This case study demonstrates that *FreqAlign* is adaptive to minor changes in system layout.

For case (2), we use two other logical topology and physical layout combinations shown in Fig. 1(d) and (e). In Fig. 1(d), the system uses 16-ary 3-stage Clos topology and a W-shape physical layout and Fig. 1(e) shows a system with a different chip aspect ratio, to which we map an 8-ary 3-stage topology. The comparisons of average resonant frequency difference among RGs for these two cases between *RingAware* and *FreqAlign* are shown in Fig. 12(c) and (d), respectively. Compared to *RingAware*, *FreqAlign* reduces the average resonant frequency difference by 42.8% and 59.2% for W-shape 16-ary 3-stage PNoC and chain-shape 8-ary 3-stage PNoC, respectively. Thus, *FreqAlign* is more adaptive to different PNoC logical topology and physical layout combinations.

D. Transient Resonance Frequency Investigation

To compare the transient behavior of *RingAware* and *FreqAlign*, we conduct a testcase where the target system has

[4]We also investigate systematic within-die (WID) process variations for both ring resonators and cores. We assume 50% leakage power variations across the chip from top to bottom due to WID process variations in a decreasing gradient [46]. Our results show that *FreqAlign* reduces the resonant frequency difference by 57.3% compared to *RingAware*. Without WID process variations for cores, the reduction is 55.6%.

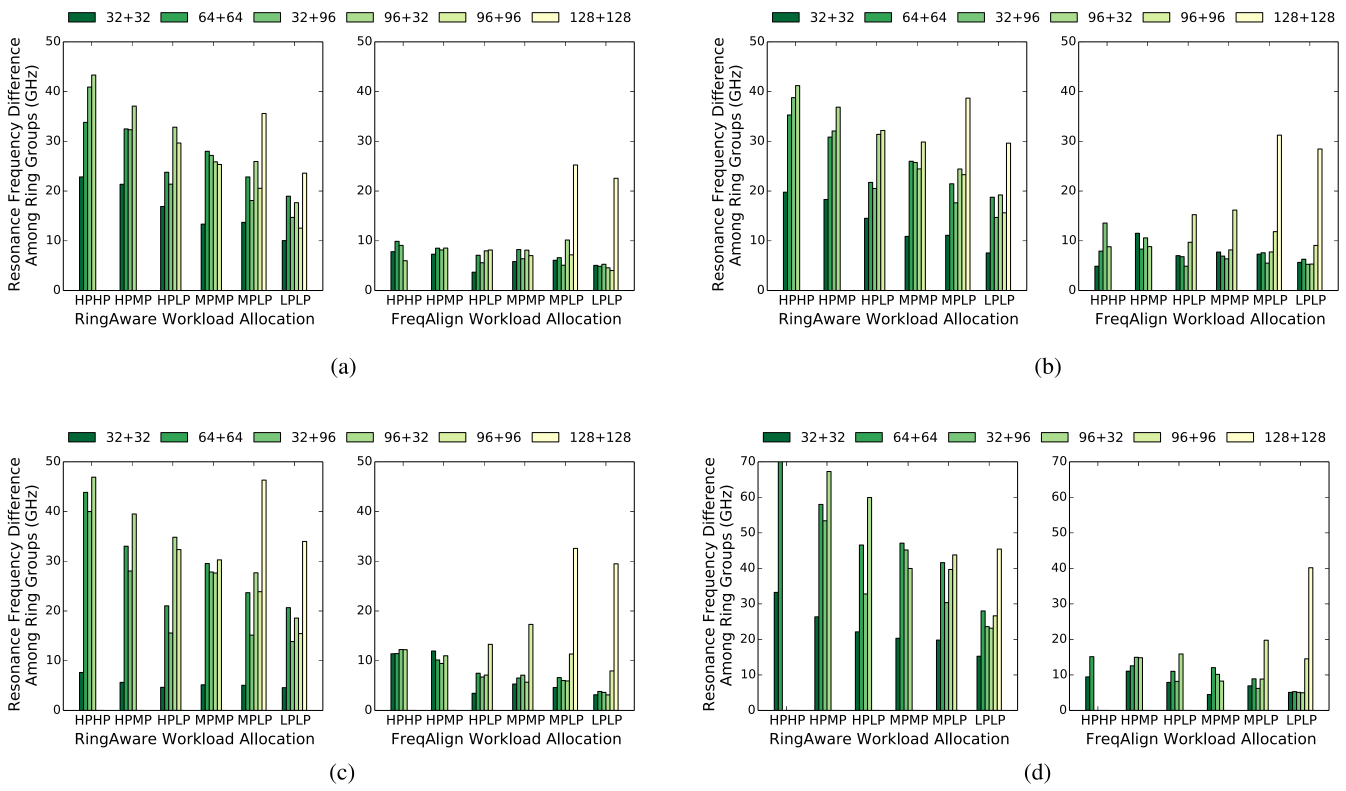


Fig. 12. Comparison of average resonant frequency difference among RGs for different PNoC logical topology and physical layout combinations between *RingAware* and *FreqAlign*. The PNoC cases and tuning power reduction of *FreqAlign+AFT* compared to *RingAware+AFT* are shown in captions of subfigures. (a) Horizontally shifted U-shape layout with 8-ary 3-stage Clos topology [Fig. 1(b)]. Average (maximum) power reduction is 1.03 W (2.05 W), 72.7% (96.3%). (b) Vertically shifted U-shape layout with 8-ary 3-stage Clos topology [Fig. 1(c)]. Average (maximum) power reduction is 0.80 W (1.47 W), 65.8% (90.3%). (c) W-shape layout with 16-ary 3-stage Clos topology [Fig. 1(d)]. Average (maximum) power reduction is 1.63 W (3.47 W), 30.1% (82.7%). (d) Chain-shape layout with 8-ary 3-stage Clos topology [Fig. 1(e)]. Average (maximum) power reduction is 0.92 W (1.8 W), 70.3% (77.9%).

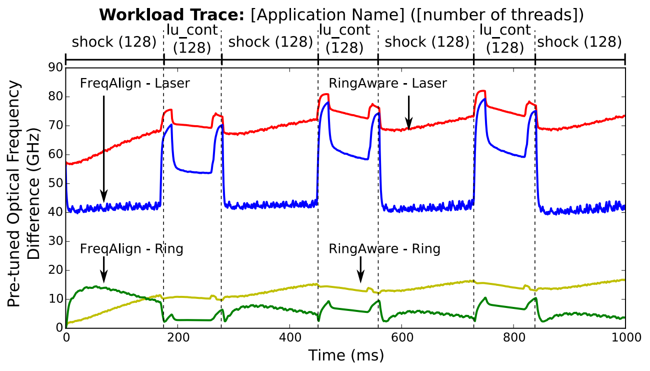


Fig. 13. Runtime optical frequency difference trace for transient case with large temporal on-chip temperature gradients prior to applying tuning methods.

a large temporal temperature gradient during operation. We use HPLP (shock + lu_cont) as the jobs and let them run alternately. We use the U-shape layout and 8-ary 3-stage Clos topology for the manycore system and set the initial temperatures of all units at 65 °C, so *RingAware* and *FreqAlign* have the same starting point. We also limit the running time to be 1000 ms. Fig. 13 shows the maximum optical frequency difference between the target frequency and the optical frequencies of RGs and on-chip laser sources for this testcase prior to applying tuning methods. We observe that at first *RingAware* results in lower frequency difference than *FreqAlign*. This is because *RingAware* has a more even workload distribution than *FreqAlign*. However, such a distribution causes the inner RGs to be hotter than the outer RGs. As the time progresses and the system enters its steady state, the resonant frequency difference of *RingAware* increases. On the other hand, *FreqAlign* achieves much lower resonant frequency difference. The jittering during shock (128) and variation during lu_cont (128) in the traces are caused by applications going through different phases during execution. *FreqAlign* reduces the amount of optical frequency that the on-chip laser sources need to be tuned compared to *RingAware*. Since scientific applications usually have long running times (minutes or hours), which is sufficient for the system to enter steady state, using *FreqAlign* can achieve lower resonant frequency difference than *RingAware* in general.

E. Closeness to Optimal Workload Allocation Solution

An important issue, which is difficult to address in the context of heuristics for NP-hard optimizations, is the degree of suboptimality of heuristic solutions. To offer some insight into the quality of *FreqAlign* workload allocation solutions relative to optimal solutions, we have studied the various heuristics on a smaller chip architecture (a 2×4-core system with two RGs placed on the two shorter opposite edges). This is because finding the optimal solution is practically infeasible for larger systems ($N!$ workload allocation solutions for a fully utilized N-core system). For the 2×4 system, we run all 10 080 workload allocation solutions ($10080 = 8!/4$ after accounting for horizontal and vertical symmetries) for ten power profiles generated by picking power numbers randomly between 0.4 and 2.8 W for each core. On average, *FreqAlign* results in lower resonant frequency difference between two RGs than 87.7% of all workload allocation solutions, while *RingAware* outperforms 69.3% of all workload allocation solutions. This suggests that *FreqAlign* returns solutions that are substantially closer to optimal than those of *RingAware*. We leave the closing of this suboptimality gap for future research.

VII. Conclusion

PNoC is a promising replacement for ENoC in manycore systems. Adoption of PNoC relies on developing techniques that efficiently manage the optical frequencies of the optical devices. In this paper, we propose a novel workload allocation policy accompanied by an adaptive tuning technique to align the optical frequencies of on-chip laser sources and ring resonators. Our policy, *FreqAlign* with *AFT*, aligns the resonant frequency of the RGs instead of aligning the temperature, and hence it can jointly compensate for the difference in the optical frequency due to thermal and process variations, which in turn reduces the power consumed in localized thermal tuning. This is the first time resonant frequency matching of on-chip laser sources and ring resonators has been investigated, and their transient impact considered with dynamic workload allocation. The experimental results demonstrate that the proposed *FreqAlign* policy performs significantly better than our previously proposed workload allocation policy *RingAware*, especially for systems with process variation in ring resonators. We also demonstrate that *FreqAlign* achieves similar benefits across systems with different PNoC logical topology and physical layout combinations. More specifically, adaptively tuning the on-chip optical devices reduces the localized tuning power by 19.28 W on average and by as large as 34.57 W.

References

[1] A. Shacham, K. Bergman, and L. P. Carloni, "On the design of a photonic network-on-chip," in *Proc. Int. Symp. Netw. Chip*, Princeton, NJ, USA, 2007, pp. 53–64.

[2] A. Joshi *et al.*, "Silicon-photonic clos networks for global on-chip communication," in *Proc. Int. Symp. Netw. Chip*, San Diego, CA, USA, 2009, pp. 124–133.

[3] Y. Pan *et al.*, "Firefly: Illuminating future network-on-chip with nanophotonics," in *Proc. Int. Symp. Comput. Archit.*, Austin, TX, USA, 2009, pp. 429–440.

[4] D. Vantrease *et al.*, "Corona: System implications of emerging nanophotonic technology," in *Proc. Int. Symp. Comput. Architect.*, Beijing, China, 2008, pp. 153–164.

[5] M. J. Cianchetti, J. C. Kerekes, and D. H. Albonesi, "Phastlane: A rapid transit optical routing network," in *Proc. Int. Symp. Comput. Architect.*, Austin, TX, USA, 2009, pp. 441–450.

[6] C. T. DeRose *et al.*, "Silicon microring modulator with integrated heater and temperature sensor for thermal control," in *Proc. Conf. Lasers Electro Opt. Quant. Electron. Laser Sci.*, San Jose, CA, USA, 2010, pp. 1–2.

[7] M. J. R. Heck and J. E. Bowers, "Energy efficient and energy proportional optical interconnects for multi-core processors: Driving the need for on-chip sources," *IEEE J. Sel. Topics Quantum Electron.*, vol. 20, no. 4, pp. 1–12, Jul./Aug. 2014.

[8] T. Zhang, J. L. Abellan, A. Joshi, and A. K. Coskun, "Thermal management of manycore systems with silicon-photonic networks," in *Proc. Design Autom. Test Europe Conf. Exhibit.*, Dresden, Germany, 2014, pp. 1–6.

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

[10] J. A. Cox, D. C. Trotter, and A. L. Starbuck, "Integrated control of silicon-photonic micro-resonator wavelength via balanced homodyne locking," in *Proc. IEEE Opt. Interconnects Conf.*, Santa Fe, NM, USA, 2013, pp. 52–53.

[11] D. M. Calhoun *et al.*, "Programmable wavelength locking and routing in a silicon-photonic interconnection network implementation," in *Proc. OSA Opt. Fiber Commun. Conf.*, Los Angeles, CA, USA, 2015, pp. 1–3.

[12] Y. Demir and N. Hardavellas, "Parka: Thermally insulated nanophotonic interconnects," in *Proc. Int. Symp. Netw. Chip*, Vancouver, BC, Canada, 2015, pp. 1–8.

[13] C. Nitta, M. Farrens, and V. Akella, "Addressing system-level trimming issues in on-chip nanophotonic networks," in *Proc. Int. Symp. High Perform. Comput. Architect.*, San Antonio, TX, USA, 2011, pp. 122–131.

[14] Z. Li *et al.*, "Aurora: A cross-layer solution for thermally resilient photonic network-on-chip," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 1, pp. 170–183, Jan. 2015.

[15] S. Manipatruni *et al.*, "Wide temperature range operation of micrometer-scale silicon electro-optic modulators," *Opt. Lett.*, vol. 33, no. 19, pp. 2185–2187, 2008.

[16] 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 J. Solid-State Circuits*, vol. 46, no. 1, pp. 173–183, Jan. 2011.

[17] *SPARC T4 Processor Data Sheet*. Accessed on May 5, 2016. [Online]. Available: http://www.oracle.com/us/products/servers-storage/servers/sparc-enterprise/t-series/sparc-t4-processor-ds-497205.pdf

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

[19] B. R. 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 *Proc. IEEE Int. Solid State Circuits Conf.*, San Francisco, CA, USA, 2013, pp. 126–127.

[20] M. Georgas *et al.*, "A monolithically-integrated optical transmitter and receiver in a zero-change 45nm SOI process," in *Symp. VLSI Circuits Dig. Tech. Papers*, Honolulu, HI, USA, 2014, pp. 1–2.

[21] H. Li *et al.*, "Thermal aware design method for VCSEL-based on-chip optical interconnect," in *Proc. Design Autom. Test Europe Conf. Exhibit.*, Grenoble, France, 2015, pp. 1120–1125.

[22] B. Song *et al.*, "3D integrated hybrid silicon laser," in *Proc. Eur. Conf. Opt. Commun.*, Valencia, Spain, 2015, pp. 1–3.

[23] T. Wang, H. Liu, A. Lee, F. Pozzi, and A. Seeds, "1.3-μm InAs/GaAs quantum-dot lasers monolithically grown on Si substrates," *Opt. Express*, vol. 19, no. 12, pp. 11381–11386, 2011.

[24] S. Lourdudoss, "Heteroepitaxy and selective area heteroepitaxy for silicon photonics," *Current Opin. Solid State Mater. Sci.*, vol. 16, no. 2, pp. 91–99, 2012.

[25] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta, "The SPLASH-2 programs: Characterization and methodological considerations," in *Proc. Int. Symp. Comput. Architect.*, Santa Margherita Ligure, Italy, 1995, pp. 24–36.

[26] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The PARSEC benchmark suite: Characterization and architectural implications," in *Proc. Int. Conf. Parallel Architect. Compilation Tech.*, Toronto, ON, Canada, 2008, pp. 72–81.

[27] D. Campbell *et al.*, "Ubiquitous high performance computing: Challenge problems specification," Georgia Tech. Res. Inst., Atlanta, GA, USA, Tech. Rep. HR0011-10-C-0145, 2012.

[28] T. Baehr-Jones *et al.*, "A 25 Gb/s silicon photonics platform," arXiv:1203.0767 [physics.optics], Mar. 2012.

[29] 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. High Perform. Comput. Netw. Stor. Anal.*, Seattle, WA, USA, 2011, pp. 1–12.

[30] S. Li *et al.*, "McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures," in *Proc. Int. Symp. Microarchitect.*, New York, NY, USA, 2009, pp. 469–480.

[31] D. Tarjan, S. Thoziyoor, and N. P. Jouppi, "CACTI 4.0," HP Lab., Palo Alto, CA, USA, Tech. Rep. HPL-2006-86, 2006.

[32] K. Skadron *et al.*, "Temperature-aware microarchitecture," in *Proc. Int. Symp. Comput. Architect.*, San Diego, CA, USA, 2003, pp. 2–13.

[33] J. Meng, K. Kawakami, and A. K. Coskun, "Optimizing energy efficiency of 3-D multicore systems with stacked DRAM under power and thermal constraints," in *Proc. Design Autom. Conf.*, San Francisco, CA, USA, 2012, pp. 648–655.

[34] H. Su, F. Liu, A. Devgan, E. Acar, and S. Nassif, "Full chip leakage-estimation considering power supply and temperature variations," in *Proc. Int. Symp. Low Power Electron. Design*, Seoul, South Korea, 2003, pp. 78–83.

[35] H. Wong. *A Comparison of Intel's 32nm and 22nm Core i5 CPUs: Power, Voltage, Temperature, and Frequency*. Accessed on Sep. 3, 2013. [Online]. Available: http://blog.stuffedcow.net/2012/10/intel32nm-22nm-core-i5-comparison/

[36] L. A. Coldren, S. W. Corzine, and M. L. Mashanovitch, *Diode Lasers and Photonic Integrated Circuits*. Hoboken, NJ, USA: Wiley, 2012.

[37] C. Chen *et al.*, "Sharing and placement of on-chip laser sources in silicon-photonic NoCs," in *Proc. Int. Symp. Netw. Chip*, Ferrara, Italy, 2014, pp. 88–95.

[38] M. C. Larson *et al.*, "Narrow linewidth sampled-grating distributed Bragg reflector laser with enhanced side-mode suppression," in *Proc. Opt. Fiber Commun. Conf. Exhibit.*, Los Angeles, CA, USA, 2015, pp. 1–3.

[39] T. Kimoto *et al.*, "Highly reliable 40-mW 25-GHz× 20-ch thermally tunable DFB laser module integrated with wavelength monitor," *Furukawa Rev.*, vol. 24, pp. 1–5, Oct. 2003.

[40] P. Dong *et al.*, "Wavelength-tunable silicon microring modulator," *Opt. Express*, vol. 18, no. 11, pp. 10941–10946, 2010.

[41] K. Van Craeynest, S. Akram, W. Heirman, A. Jaleel, and L. Eeckhout, "Fairness-aware scheduling on single-ISA heterogeneous multi-cores," in *Proc. Int. Conf. Parallel Architect. Compilation Tech.*, Edinburgh, U.K., 2013, pp. 177–188.

[42] K. Van Craeynest, A. Jaleel, L. Eeckhout, P. Narvaez, and J. Emer, "Scheduling heterogeneous multi-cores through performance impact estimation (PIE)," in *Proc. Int. Symp. Comput. Architect.*, Portland, OR, USA, 2012, pp. 213–224.

[43] *Benchmark Native Execution*. Accessed on Feb. 25, 2016. [Online]. Available: http://snipersim.org/documents/gainestown-native.html

[44] W. A. Zortman, D. C. Trotter, and M. Watts, "Silicon photonics manufacturing," *Opt. Express*, vol. 18, no. 23, pp. 23598–23607, 2010.

[45] M. Mohamed *et al.*, "Power-efficient variation-aware photonic on-chip network management," in *Proc. Int. Symp. Low Power Electron. Design*, Austin, TX, USA, 2010, pp. 31–36.

[46] S. Dighe *et al.*, "Within-die variation-aware dynamic-voltage-frequency-scaling with optimal core allocation and thread hopping for the 80-core teraFLOPS processor," *IEEE J. Solid-State Circuits*, vol. 46, no. 1, pp. 184–193, Jan. 2011.

José L. Abellán (S'07–M'08) received the Ph.D. degree in computer science and engineering from the University of Murcia, Murcia, Spain.

He is an Assistant Professor with the Computer Science Department, Catholic University of Murcia, Murcia. His current research interests include GPU architectures, networks-on-chips, high performance computing, and emerging silicon photonics technology.

Ayse K. Coskun (M'06) received the Ph.D. degree in computer science and engineering from the University of California at San Diego, La Jolla, CA, USA.

She is an Associate Professor with the Electrical and Computer Engineering Department, Boston University, Boston, MA, USA. Her current research interests include energy-efficient computing, 3-D stack architectures, and embedded systems and software.

Anjun Gu (S'12) received the B.S. degree in electrical and computer engineering from the University of Texas at Austin, Austin, TX, USA. He is currently pursuing the Ph.D. degree with the Electrical and Computer Engineering Department, University of California at San Diego, La Jolla, CA, USA.

His current research interests include 3-D IC physical implementation flow and design margin reduction and signoff.

Warren Jin received the B.S. degree in computer engineering from Brown University, Providence, RI, USA. He is currently pursuing the Ph.D. degree with the Electrical Engineering Department, University of California at Santa Barbara, Santa Barbara, CA, USA.

His current research interests include optoelectronics and photonics.

Ajay Joshi (S'99–M'07) received the Ph.D. degree in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, GA, USA.

He is an Associate Professor with the Electrical and Computer Engineering Department, Boston University, Boston, MA, USA. His current research interests include very large-scale integration design and emerging device technologies such as silicon photonics and memristors.

Andrew B. Kahng (M'03–SM'07–F'10) received the Ph.D. degree in computer science from the University of California at San Diego, La Jolla, CA, USA.

He is a Professor with the Computer Science Engineering Department and the Electrical and Computer Engineering Department, University of California at San Diego. His current research interests include IC physical design, the design-manufacturing interface, combinatorial optimization, and technology roadmapping.

Jonathan Klamkin (SM'14) received the Ph.D. degree in electronic materials from the University of California at Santa Barbara, Santa Barbara, Ca, USA.

He is an Associate Professor with the Electrical and Computer Engineering Department, University of California at Santa Barbara. His current research interests include photonic integrated circuits and devices, widely-tunable lasers, and semiconductor optical amplifiers.

Cristian Morales received the B.S. degree in electrical engineering from Boston University, Boston, MA, USA, where he is currently pursuing the Ph.D. degree with the Electrical and Computer Engineering Department.

His current research interests include thermal modeling and management of systems with silicon photonic networks.

John Recchio (S'11) received the B.S. degree in electrical and computer engineering from the University of California at San Diego, La Jolla, CA, USA, where he is currently pursuing the Ph.D. degree.

His current research interests include photonic integrated circuits and networks.

Vaishnav Srinivas (M'02) received the M.S. degree from the University of California at Los Angeles, Los Angeles, CA, USA. He is currently pursuing the Ph.D. degree with the Electrical and Computer Engineering Department, University of California at San Diego, La Jolla, CA, USA.

His current research interests include low-power interfaces, IO and memory architecture, and roadmapping interface technology.

Tiansheng Zhang (S'12) received the B.S. degree in electrical engineering from the Harbin Institute of Technology, Harbin, China. He is currently pursuing the Ph.D. degree with the Electrical and Computer Engineering Department, Boston University, Boston, MA, USA.

His current research interests include 3-D stack architectures, silicon photonic NoCs, and resource and thermal management on manycore systems.