LIBRA: Thermal and Process Variation Aware Reliability Management in Photonic Networks-on-Chip

Sai Vineel Reddy Chittamuru, *Student Member, IEEE*,
Ishan G. Thakkar, *Student Member, IEEE*, and Sudeep Pasricha, *Senior Member, IEEE*

Abstract—Silicon nanophotonics technology is being considered for future networks-on-chip (NoCs) as it can enable high bandwidth density and lower latency with traversal of data at the speed of light. But, the operation of photonic NoCs (PNoCs) is very sensitive to on-chip temperature and process variations. These variations can create significant reliability issues for PNoCs. For example, a microring resonator (MR) may resonate at another wavelength instead of its designated wavelength due to thermal and/or process variations, which can lead to bandwidth wastage and data corruption in PNoCs. This paper proposes a novel run-time framework called *LIBRA* to overcome temperature- and process variation- induced reliability issues in PNoCs. The framework consists of (i) a device-level reactive MR assignment mechanism that dynamically assigns a group of MRs to reliably modulate/receive data in a waveguide based on the chip thermal and process variation characteristics; and (ii) a system-level proactive thread migration technique to avoid on-chip thermal threshold violations and reduce MR tuning/ trimming power by dynamically migrating threads between cores. Our simulation results indicate that *LIBRA* can reliably satisfy on-chip thermal thresholds and maintain high network bandwidth while reducing total power by up to 61.3 percent, and thermal tuning/trimming power by up to 76.2 percent over state-of-the-art thermal and process variation aware solutions.

Index Terms—Thermal variations, process variations, photonic networks-on-chip, reliability

1 Introduction

As the core count in emerging manycore systems increases, traditional on-chip communication fabrics, i.e., electrical networks-on-chip (ENoCs), are unable to scale up to meet aggressive performance goals. To overcome performance bottlenecks with ENoCs, recent advances in CMOS-photonics integration [1] have enabled an exciting solution in the form of photonic NoCs (PNoCs). PNoCs excel over their electrical counterparts in performance due to several advantages, including near light speed transfers, high bandwidth density, and low dynamic power dissipation [2]. Thus, emerging manycore systems may be able to overcome their on-chip communication bottlenecks with PNoCs (e.g., [3], [4], [5], [6], [7], [8]).

Typical PNoC architectures employ several photonic devices such as photonic waveguides, couplers, splitters, and multi-wavelength laser sources, along with microring resonators (MRs) as modulators, detectors, and switches. In a typical high bandwidth PNoC, an off-chip laser source generates multi-wavelength light, which is coupled by an optical coupler to an on-chip photonic waveguide. This waveguide guides the input optical power of multiple dense-wavelength-division-multiplexed (DWDM) wavelengths, via a series of optical power splitters, to the individual nodes (e.g., processing cores) on the chip. Each node in the PNoC can connect and communicate to multiple other nodes through such photonic waveguides that can guide the utilized DWDM wavelengths. A wavelength serves as a carrier to a data signal. Typically, multiple data signals are generated at a source node in the electrical domain as sequences of logical 1 and 0 voltage levels. These input electrical data signals are modulated onto the DWDM wavelengths using a bank of modulator MRs. These data-modulated carrier wavelengths traverse a waveguide to a destination node, where an array of detector MRs filter them and drop them on adjacent photodetectors to regenerate electrical data signals. In general, each node in a PNoC should be able to send and receive data in the optical domain on all of the utilized carrier wavelengths. Therefore, each node has a bank of modulator MRs and a bank of detector MRs. Each MR in a bank is structurally itself a looped waveguide with a small and unique circumference, which makes the MR resonate with and operate on a specific carrier wavelength referred to as the MR's assigned carrier wavelength. Thus, the excellent wavelength selectivity of MRs and DWDM capability of waveguides enable high bandwidth parallel data transfers in PNoCs, under ideal conditions.

As advocated by prior works [3], [4], [5], [6], PNoCs are expected to be 3D-stacked on top of their respective manycore chips. Therefore, the MRs of PNoCs will be placed on top of, and hence in close proximity to, processing cores. Variations in core workloads lead to variations in their power dis4sipation, which in turn can alter the temperatures

• *The authors are with the Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523. E-mail: {sai.chittamuru, ishan.thakkar, sudeep}@colostate.edu.*

Manuscript received 9 Oct. 2017; revised 29 May 2018; accepted 31 May 2018. Date of publication 12 June 2018; date of current version 29 Jan. 2019. (Coressponding author: Sai Vineel Reddy Chittamuru.)
Recommended for acceptance by N. Jha.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMSCS.2018.2846274

of the cores and MRs in their vicinity. For instance, the temperature on a typical manycore chip can easily vary by as much as 90°C [13]. Unfortunately, MRs are very sensitive to these on-chip thermal variations (TV): their effective refractive indices, and hence their resonance wavelengths are altered if their operating temperatures change. Therefore, in a typical PNoC, the resonance wavelengths of the utilized modulator MRs may not align with, and hence may not modulate their assigned carrier wavelengths [9]. This may result in bandwidth wastage, or worse, data corruption when detector MRs are unable to read from their assigned carrier wavelengths [10].

In addition to TV, MRs are also susceptible to fabrication process variations. Process variations (PV) induce variations in the width, thickness, and doping concentration of MRs (see Section 3.2), causing resonance wavelength shifts in MRs [11], [12]. PV measurements of fabricated MR devices indicate a standard deviation (σ) of 1.3 nm in width, which translates to a 0.76nm shift in an MR's resonance wavelength [14]. These PV-induced resonance wavelength shifts in MRs also cause bandwidth wastage and data corruption.

The adverse effects of PV and TV related to resonance shifts in MRs, and their performance and reliability impacts, can be redressed by realigning the resonant wavelengths of MRs with their assigned carrier wavelengths using localized trimming [15] and thermal tuning [9] mechanisms. Trimming alters the free-carrier concentration in an MR core, whereas thermal tuning uses integrated micro-heaters to alter local temperatures at MRs. But these mechanisms come with high power and performance overhead [9]. *Hence, it is essential to intelligently manage thermal and process variations in PNoC-based manycore systems, to achieve reliable communication with minimal trimming and tuning costs.*

In this work, we aim to minimize the need for (and overheads of) localized thermal tuning and trimming in PNoCs while coping with process and thermal variations, thereby easing the adoption of PNoCs in future manycore systems. We propose a novel thermal and process variation aware dynamic reliability management framework called *LIBRA* that integrates adaptive MR assignment at the device-level and dynamic thread migration at the system-level for PNoC-based manycore systems. Our novel contributions as part of the *LIBRA* framework are summarized below:

- We design a novel thermal and process variation aware MR assignment (*TPMA*) mechanism at the device-level, which dynamically assigns a set of MRs to the utilized set of carrier wavelengths at run-time. *TPMA* enables reliable modulation and reception of data with minimal overheads, while maintaining the maximum possible bandwidth;
- We propose a novel PV-aware anti wavelength-shift dynamic thermal management (*VADTM*) mechanism at the system-level, which uses support vector regression (SVR) based temperature prediction and dynamic thread migration to avoid on-chip thermal threshold violations and reduce trimming/tuning power for MRs;
- We evaluate our *LIBRA* ($TPMA + VADTM$) framework on a 64-core chip, and compare it with four state-of-the-art thermal management solutions: an MR-aware thermal management (RATM) framework [16], an MR PV-aware thermal management (FATM) framework [17], a predictive dynamic thermal management (PDTM) framework [18], and an MR-aware thermal management (SPECTRA) framework [10]; and show significant reduction in maximum temperature and trimming/tuning power costs compared to these solutions.

2 Related Work

Traditional electrical NoC communication fabrics are projected to suffer from cripplingly high power dissipation and severely reduced performance in future manycore systems [19]. The higher bandwidth density and lower power dissipation possible with silicon-photonic links, compared to electrical wires, has made them an attractive option for manycore systems. Recent research has thus focused on exploring a wide spectrum of network topologies and protocols to enable efficient PNoC architectures [3], [4], [5], [6], [7], [8], [32].

PV and TV in silicon-photonic links represent important challenges for the widespread adoption of PNoC architectures. Several techniques have been proposed to reduce thermal hotspots and gradients using DVFS [20], [21], workload migration [8], [18], [22] and liquid cooling [27]. A few PV-aware application mapping frameworks have also been proposed [29], [30] that optimize performance and energy in manycore systems. In [29] a run-time application-mapping strategy was presented, which considers the variation profile of a manycore processor to maximize performance and reduce leakage-power for a given fixed power budget. In [30] a framework was presented that integrates reliability and variation-awareness in a run-time variable degree-of-parallelism (DoP) application-scheduling methodology to enhance manycore performance. *However, these techniques do not consider the unique challenges (e.g., MR resonance wavelength shifts) and constraints (e.g., wavelength match between sender and receiver MR pairs) that exist in PNoCs.*

A few prior works have analyzed the impact of TV and PV on PNoCs at the device-level, link-level, and system-level, and proposed solutions to remedy these variations.

The device-level efforts have mainly proposed various athermal photonic devices to reduce localized tuning/trimming power in MRs. These design-time solutions include using materials such as cladding to reduce thermal sensitivity [31] and using heaters and temperature sensors for thermal control [33]. An electrical backend capable of bit reshuffling was proposed in [34] to enhance photonic link robustness against TV and PV with lower MR tuning power. *While these device- and link-level techniques are promising, they either possess a high power overhead or require costly changes in the manufacturing process (e.g., larger device areas) that would decrease bandwidth density and area efficiency.* In addition, there are device and link-level solutions [23], [24], [25], [26], [55] aim to mitigate crosstalk noise in PNoCs.

At the system-level, the overhead associated with localized tuning of MRs was reduced in [9] using the group shift property of co-located MRs as part of a method to trim a group of rings at the same time. In [16], a ring-aware thread scheduling policy was proposed to reduce on-chip thermal gradients in a PNoC. In [35], a thread migration mechanism was proposed to minimize on-chip thermal gradients within a PNoC. In [57], an island of heater based thermal management framework was proposed to adapt groups or islands of MRs within PNoCs to on-chip thermal variations. A few prior works have

also explored the impact of PV on DWDM-based photonic links at the system-level [36], [37], [38]. A reliability-aware design flow to address variation induced reliability issues is proposed in [36], which uses athermal coating at fabrication-level, voltage tuning at device-level, as well as channel hopping at the system-level. In [37], a methodology to salvage network-bandwidth loss due to PV-shifts is proposed, which reorders MRs and trims them to nearby wavelengths. In [38], power-efficient techniques are proposed, based on inter-channel hopping and variation-aware routing to compensate for PV effects at runtime. A few system-level works [14], [17], [39] also consider the impact of both TV and PV on optical links. In [14], a thermal-tuning approach is presented that adjusts chip temperature using DVFS to compensate for chip-wide thermal and process variation induced resonance shifts in MRs and improve system performance. In [17], a PV aware workload allocation policy is presented to reduce the thermal tuning power of PNoCs. In [39], a tunable laser source design is demonstrated, in which the signal power at the source is adapted to compensate for signal losses due to TV and PV across optical interconnects. *None of these system-level solutions for PNoCs considers the impact of the relationship between thermal hotspots and transmission reliability.*

To address these shortcomings of prior work, we proposed the SPECTRA framework in our prior work [10]. SPECTRA is a cross-layer framework that combines two dynamic thermal management mechanisms to reduce maximum on-chip temperature and conserve trimming and tuning power of MRs in DWDM-based PNoC architectures. Our proposed *LIBRA* framework in this paper improves upon SPECTRA, by *(i)* considering the impact of PV on dynamic thermal management; *(ii)* utilizing a new device-level TV and PV aware ring assignment mechanism; and *(iii)* utilizing a new system-level PV-aware thread migration mechanism. Sections 4, 5, 6 describe our proposed framework which is then evaluated in Section 7 against prior work.

3 Impact of TV and PV on DWDM Based PNOCs

In this section, we explain the key impacts of PV and TV on DWDM based PNoCs. Although most silicon-based photonic devices exhibit some susceptibility to temperature and process variations, the high wavelength selectivity of MRs makes them especially susceptible to these variations. Therefore, we primarily focus on the impacts of TV and PV on MRs.

3.1 Impact of TV on DWDM Based PNoCs

In a DWDM PNoC, the temperatures of the individual compute nodes and their associated MR banks follow the workload-dependent temperatures of the processing cores in the nodes. As the application workload of each core in a manycore system usually differs from that of other cores and also varies with time, the temperatures of the cores (and thus nodes) of the system differ from one-another and vary with time. As a result, the temperatures of different MR banks of the PNoC also differ from one another and vary with time.

Typically, the MR banks of each PNoC node are designed to resonate with and operate upon their assigned carrier wavelengths at a specific temperature, e.g., room temperature. But due to the time- and workload-dependent temperature variations, the resonances of different MR banks shift away from their assigned carrier wavelengths by different amounts.

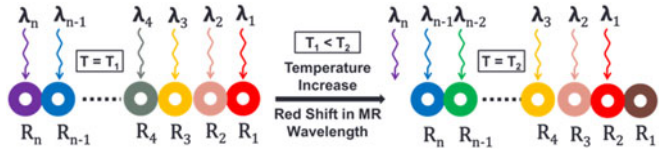

Fig. 1. Impact of temperature increase on an MR bank.

For example, Fig. 1 depicts an MR bank with MRs R_1-R_n that are designed to resonate with their assigned carrier wavelengths λ_1-λ_n, respectively, at temperature T_1. As the temperature increases to T_2, the resonance wavelength of each MR shifts away from its assigned carrier wavelength towards the red end of the spectrum (i.e., *red-shift*). This red-shift is shown in the figure where, at temperature T_2 ($T_2 > T_1$), the resonance wavelength λ_i of MR R_i is in line with the carrier wavelength λ_{i-1}. Consequently, the carrier wavelength λ_n is not assigned to any of the MRs. This results in *bandwidth wastage* if the MR bank is a modulator MR bank, as λ_n cannot be modulated by any modulator MR now. This example scenario can also result in *data corruption* if the MR bank is a detector MR bank, as λ_n cannot be received by any detector MR. Similarly, if $T_2 < T_1$, the resonance wavelength λ_i of each R_i shifts towards the blue end of the spectrum (i.e., *blue-shift*), which may leave λ_1 unassigned, causing bandwidth wastage or data corruption. Thus, during the runtime of a PNoC, an increase in an MR bank's temperature red-shifts the resonances of all its MRs, whereas a decrease in an MR's temperature blue-shifts the resonances of all its MRs.

The amount of shift in an MR's resonance not only depends on the magnitude of temperature change, but also on the MR's structure and geometry manifested as its effective refractive index n_{eff}. Typically, an MR is a looped waveguide with a silicon (Si) core and silicon dioxide (SiO_2) cladding, irrespective of whether it is used as a modulator or a detector. The change $\Delta\lambda_r$ in the resonance wavelength λ_r of an MR due to an arbitrary change ΔT in its local temperature is given by the following equation [40]:

$$\frac{\Delta\lambda_r}{\Delta T} = \frac{\delta n_{eff}}{\delta T}\frac{\lambda_r}{n_g} = \left(\Gamma_{Si}\frac{\delta n_{Si}}{\delta T} + \Gamma_{SiO2}\frac{\delta n_{SiO2}}{\delta T}\right)\frac{\lambda_r}{n_g}, \quad (1)$$

Here, n_g is the group refractive index (ratio of speed of light to group velocity of all wavelengths traversing the waveguide) of the MR waveguide. Γ_{Si} and Γ_{SiO2} are the modal confinement factors of the MR's core (Si) and cladding (SiO_2), respectively. $\delta n_{Si}/\delta T$ and $\delta n_{SiO2}/\delta T$ are the thermo-optic coefficients of *Si* (MR's core) and SiO_2 (MR's cladding) materials, with values of $1.86 \times 10^{-4}\,K^{-1}$ and $1 \times 10^{-5}\,K^{-1}$, respectively [40]. As the thermo-optic coefficient of *Si* is an order of magnitude greater than that of SiO_2, and as Γ_{Si} is also greater than Γ_{SiO2} for a typical MR, the contributions from the MR's cladding (SiO_2) in Eq. 1 can be ignored. Consequently, Eq. (1) reduces to:

$$\Delta\lambda_r = \Gamma_{Si}.\frac{\delta n_{Si}}{\delta T}.\frac{\lambda_r}{n_g}.\Delta T, \quad (2)$$

Note that the MRs used in this study are looped channel waveguides with a cross section of $450\,nm \times 220\,nm$. We model these MRs using a commercial-grade eigenmode solver [54], based on which the values of Γ_{Si} and n_g at 1550 nm are calculated to be 0.78 and 4.16, respectively.

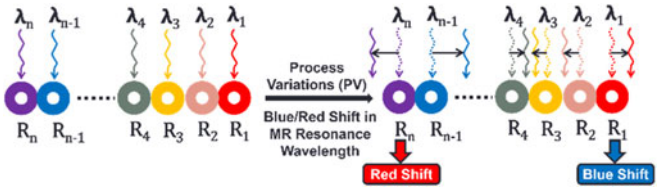

Fig. 2. Impact of PV on DWDM based PNoCs.

3.2 Impact of PV on DWDM Based PNoCs

Ideally, without any fabrication-induced PV, a sender or a receiver node can modulate and detect all of the carrier wavelengths available in the waveguide without any bandwidth loss or error. But in reality, similar to deep submicron electronic devices, photonic devices such as MR modulators, MR detectors, grating couplers, splitters etc. also suffer from significant PV [43]. In this work, we mainly focus on the severe PV effects in MRs. The MR structure is very sensitive to PV, much like it is to TV. Due to PV effects, the widths, heights, and side wall roughness of MRs can deviate from desired values after fabrication. Consequently, the resonance wavelengths (λ_r) of the MRs also deviate from their designed values. For example, 1nm of variation in width and height of an MR can lead to 0.58∼1 nm and ∼2 nm shift in its resonance wavelength, respectively [11].

As discussed earlier, PNoCs employ DWDM-based photonic links with cascaded MRs (i.e., MR banks) in their sending and receiving nodes. Unlike TV that induces systematic red or blue shifts in all the MRs of an MR bank, PV can incur random shifts in the resonance wavelengths of the MRs of a single bank, as shown in Fig. 2. From this figure, MRs R_1, R_4, . . ., R_{n-1} have blue shift in their resonance wavelengths and MRs R_2, R_3, . . ., R_n have red shift in their resonance wavelengths. Much like with TV, PV can also throw the resonances of the MRs out of alignment with their assigned carrier wavelengths, which can ultimately lead to bandwidth wastage and/or data corruption.

In summary, to enable reliable photonic communication, there is a need to mitigate the combined impact of TV and PV on PNoCs. This paper presents a cross-layer framework that uses device-level and system-level enhancements to remedy the combined impact of TV and PV. Before discussing our proposed framework, we present our performance, power, and thermal setup for modeling manycore systems with PNoCs in the next subsection. We also present a characterization of the impact of TV and PV on the MRs of a typical DWDM PNoC based manycore system, in this next subsection.

3.3 Modeling TV and PV in PNoC Architectures

To model and characterize TV and PV in a manycore system with a PNoC, we developed a simulation framework, which integrates performance, power, thermal, and variation simulators, as shown in Fig. 3. We considered a three layered 3D-stacked 64-core system as advocated in existing PNoC architectures [3], [4], [5], [6] with a planar die area footprint of 400 mm^2. Microarchitectural parameters of the manycore system are presented in Table 1. The top layer is the core-cache layer, where each core along with its private L1 cache, private and inclusive L2 cache and network interface spreads across an area of 6.25 mm^2 (i.e., $2.5\ mm \times 2.5\ mm$). For each core, we use an architecture and layout similar to the IA-32 core from Intel's Single-chip Cloud Computer (SCC) [47], scaled to 32 nm. As this core-cache layer dissipates significant portion of total power, therefore heat sink is placed closer to this layer. The middle layer is the conversion layer with digital and analog circuits that support electrical-to-optical (E/O) and optical-to-electrical (O/E) conversion of data. Finally, the bottom layer is the photonic layer with photonic components and devices (e.g., MRs, waveguides, ring heaters, etc.) that comprise a PNoC.

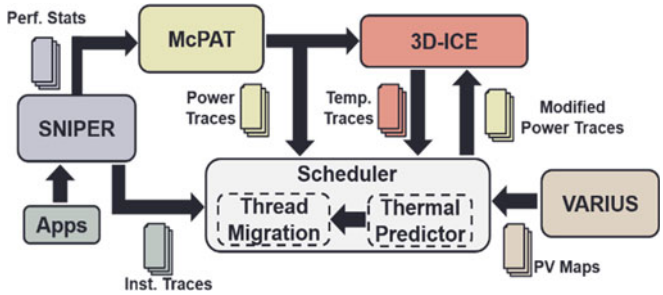

Fig. 3. Simulation framework to analyze TV and PV in a manycore system with a PNoC architectures. The framework integrates performance, power, thermal, and variation simulators.

TABLE 1
Micro-Architectural Parameters for Manycore System

Number of cores	64
Threads per core	1
Per Core:	
L1 I-Cache size/Associativity	32 KB/Direct Mapped Cache
L1 D-Cache size/Associativity	32 KB/Direct Mapped Cache
L2 Cache size/ Associativity	256 KB/ Direct Mapped Cache
L2 Coherence	MOESI
Frequency	2 GHz
Issue Policy	In-order
Memory controllers	8
Main memory	8 GB; DDR5@30 ns

We use Sniper [44] to simulate the performance of the manycore system while it executes multithreaded applications from SPLASH-2 [45] and PARSEC [46] benchmark suites. To factor in the varying system utilizations as a contributor to the dynamic TV in the processing cores and its impact on the associated photonic devices (e.g., MRs), we run each application on a target 64-core system (see Section 6) with 8, 16, 32, 48, and 64 threads. To capture runtime behavior of an application, we generate performance traces using Sniper, which are fed to MCPAT [46] to generate power traces at core-level granularity. We use published power dissipation data from Intel's Single-Chip Cloud Computer (SCC) [47], scaled to 32 nm, to calibrate our dynamic power data. The power traces generated by McPAT are given as inputs to the 3D-ICE tool [48] for transient thermal simulations (see Fig. 3). Some of the key materials used in the construction of the 3D-stack in the 3D-ICE tool and their properties are shown in Table 2.

We analyzed the spatial variation in the peak temperatures of various tiles (at core-level granularity) of the photonic layer. For the 64-core system each tile has an estimated area of 6.25 mm^2 (i.e., $2.5 \times 2.5\,mm^2$). We executed 64-threaded versions of the blackscholes (*BS*), bodytrack (*BT*), vips (*VI*), facesim (*FS*), fluidanimate (*FA*), swaptions (*SW*), barnes (*BA*), fft (*FFT*), radix (*RX*), radiosity (*RD*), and raytrace (*RT*) applications from the PARSEC and SPLASH2 benchmark suites on

TABLE 2
Properties of Materials Used By 3D-ICE Tool [49], [50]

Material	Thermal Conductivity	Volumetric Heat Capacity
Silicon	1.30e-4 W/μm K	1.628e-12 J/μm^3 K
Silicon di oxide	1.46e-6 W/μm K	1.628e-12 J/μm^3 K
BEOL	2.25e-6 W/μm K	2.175e-12 J/μm^3 K
Copper	5.85e-4 W/μm K	3.45e-12 J/μm^3 K

the 64-core system with one application running at a time. We monitored the peak temperature of each part of the photonic layer for every application and plotted the maximum peak temperature of each part across all the applications, as shown in Fig. 4a. From this figure, we can observe the maximum possible temperature-rise (above the room temperature) for any part of the layer, which caps all possible dynamic TV values for that part. From Fig. 4a, higher peak temperatures are obtained at the center of the chip while relatively lower peak temperatures are achieved at the periphery of the chip. The main reason for the higher temperature at the center of the chip is the inefficiency of the heat sink to remove heat from the center of the chip. Furthermore, using Eqs. (1) and (2), we determined the resonance wavelength shifts because of the peak temperature-rises, which are presented as a histogram in Fig. 4b. As evident from this figure, TV can induce up to a 7.4 nm shift in MR resonances.

In addition to TV, we also analyzed PV in PNoCs with the simulation setup presented in Fig. 3. We adapted the VARIUS tool [51] to model die-to-die (D2D) as well as within-die (WID) process variations in MRs for the PNoC. VARIUS uses a normal distribution to characterize on-chip D2D and WID process variations. The key parameters are mean (μ), variance ($\sigma 2$), and density (α) of a variable that follows the normal distribution. As wavelength variations are approximately linear to the dimension variations of MRs, we assume they follow the same distribution. The mean (μ) of wavelength variation of an MR is its nominal resonance wavelength. For PNoCs, we considered waveguides with 32 DWDM degree sharing the working band 1530–1625 nm (i.e., C and L bands) with a wavelength channel spacing of 1.48 nm. Hence, those wavelengths are the means for each MR modeled. The variance ($\sigma 2$) of wavelength variation is determined based on laboratory fabrication data [11] and our target die size. For a 64-core chip with 400 mm^2 size at 32 nm node, we consider a WID and D2D standard deviations of $\sigma \mathrm{WID} = 0.61$ nm $\sigma \mathrm{D2D} = 1.01$ nm, respectively [37]. We also consider a density (α) of 0.5 [37] for this die size. With these parameters, we use VARIUS to generate 100 process variation maps.

We depict a PV map in Fig. 5a, which shows a spatial variation in PV-induced resonance wavelength shifts on the photonic die. Each PV map contains over one million points indicating the PV-induced shifts in MR resonances. The total number of points picked from these maps equal the number of MRs in the PNoC. We also present these points as a histogram in Fig. 5b. As evident from the histogram, PV can induce resonance wavelength shifts in the range of -1.8 nm to 1.6 nm. However, we observed that this range can increase up to -3 nm to 3 nm for other PV maps.

4 Overcoming PV/TV Induced Resonance Wavelength Shifts

The adverse effects of PV and TV, i.e., resonance shifts in MRs and their performance and reliability impacts, can be

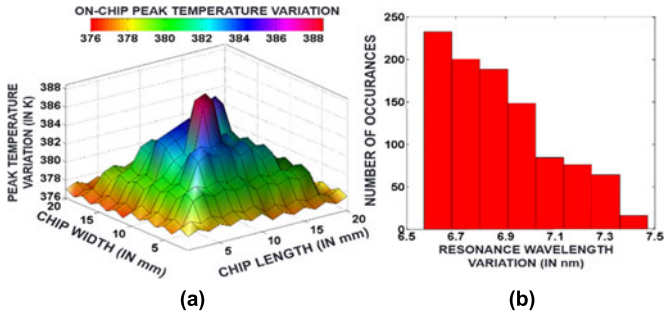

Fig. 4. (a) Spatial variation in peak temperatures. (b) Histogram of peak TV-induced resonance wavelength variation across a chip of size 400 mm^2 using 3D ICE tool while executing 64 threaded PARSEC and SPLASH2 benchmark applications on a 64-core system.

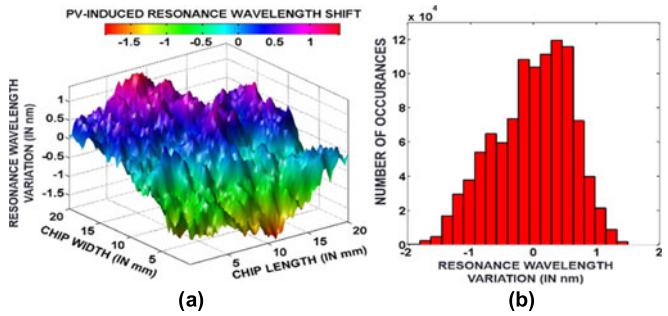

Fig. 5. (a) PV-induced resonance wavelength variation. (b) Histogram of resonance wavelength variation across a chip of size 400 mm^2.

overcome by realigning and locking the resonance wavelengths of the individual MRs with the utilized carrier wavelengths. As PV is a static phenomenon, the PV-induced resonance shifts need to be overcome only once at system initialization. In contrast, due to the dynamic nature of TV, the TV-induced resonance shifts require runtime thermal stabilization of MRs. A stable locking of MR resonances with the utilized carrier wavelengths can be achieved using device-level (MR-level) mechanisms, such as localized trimming [15] and/or thermal tuning [9], with a dithering signal based feedback control [40]. However, the localized trimming and thermal tuning mechanisms proposed in prior work come with several challenges, which must be overcome to ease the adoption of PNoCs for future manycore systems.

First, thermal tuning and localized trimming mechanisms cannot provide sufficient tuning range to remedy PV/TV-induced resonance shifts in MRs. For instance, from Section 3, TV and PV together can induce shifts in MR resonance wavelengths of up to 10.4 nm, i.e., 7.4 nm for TV and ± 3 nm for PV. Therefore, compensating these TV/PV-induced resonance shifts would require a net tuning range of 10.4 nm. But localized trimming can provide a tuning range of only 1.5 nm at most [36]. In contrast, thermal tuning can provide a tuning range of about 6.6 nm corresponding to the temperature range of up to 60K [40] at 0.11 nm/K sensitivity [9]. Thus, even the thermal tuning and localized trimming together (i.e., $6.6\ \mathrm{nm} + 1.5\ \mathrm{nm}$ tuning range) cannot provide the required tuning range of $\sim$10.4 nm. Another challenge for these mechanisms is their significant power overhead. A typical MR may consume 130 μW of trimming power or 240 μW of thermal tuning power to remedy 1 nm shift in its resonance wavelength, depending on its size, structure, and integration feasibility [9]. To remedy a larger

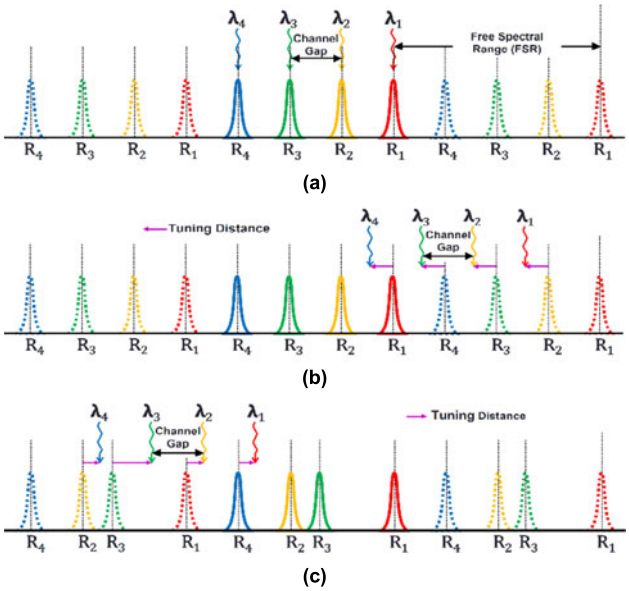

Fig. 6. Periodic resonances $(R_1 - R_4)$ of an example bank of four MRs and their assigned carrier wavelengths $(\lambda_1 - \lambda_4)$ for (a) an ideal case with no resonance shifts, (b) a case with systematic blue-shifts in resonances, and (c) a case with random red-shifts in resonances.

shift of ~10.4 nm, a single MR may consume as much as ~1.35 mW of trimming power or ~2.5 mW of thermal tuning power. As a DWDM PNoC may have thousands of MRs, the total power overhead of PV/TV remedy can easily be in the range of a few tens of watts, which is a prohibitively high power overhead for chip-scale systems and must be minimized to make the total power costs of large-scale DWDM PNoCs manageable.

Fortunately, due to the periodicity of MR resonances, the resonance of none of the MRs in a PNoC needs to be tuned for more than a single channel gap [34]. This makes the required tuning range and the total tuning power more manageable. To understand this, consider Fig. 6. The periodic resonances (R_1-R_4) of an example bank of four MRs and their assigned carrier wavelengths (λ_1-λ_4) for an ideal case with no PV or TV are shown in Fig. 6a. Due to the absence of PV/TV, the resonances of all MRs are aligned with their assigned carrier wavelengths. Fig. 6b shows systematic blue-shifts of over two channel gaps in the resonances of all four MRs. In this case, the MR resonances can be re-aligned to their nearest carrier wavelengths followed by electrical repositioning of bits using backend barrel-shifters or pipelined shift registers [34]. In case the random PV throw the MR resonances out of order (Fig. 6c), use of bit reordering multiplexers at the backend can still allow the MR resonances to be re-aligned to their nearest carrier wavelengths. Thus, due to the periodicity of MR resonances, and the use of bit reordering/repositioning techniques, the necessary tuning distance for the individual MRs reduces to less than one channel gap.

Our previously proposed SPECTRA framework [10] uses a different approach to reduce the required tuning distance and power overhead of PV/TV remedy. It integrates one system-level and two device-level optimizations. At the device-level, the SPECTRA framework utilizes three more MRs than the number of utilized carrier wavelengths, and thus, increases the available tuning range by three channel gaps. This mechanism reassigns the extra MRs to operate on nearby carrier wavelengths in the case when the resonances shift by less than three channel gaps. The need for remedying resonance

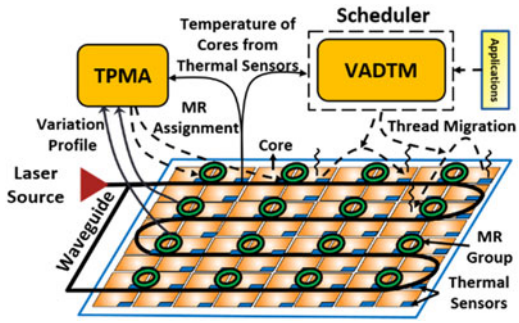

Fig. 7. Overview of LIBRA framework that integrates a device-level thermal and process variation aware microring assignment mechanism *(TPMA)* and a system-level variation aware anti wavelength-shift dynamic thermal management *(VADTM)* technique.

shifts of more than three channel gaps is eliminated by reducing the range of temperature swings of the individual cores below the threshold levels that can induce resonance shifts of greater than three channel gaps. For that, an adaptive thread migration policy is used at the system level, which also eliminates the need of bit-shifting. Moreover, SPECTRA adaptively chooses the least power-consuming method from *thermal tuning* and *localized trimming* as the preferred method for PV/TV remedy. Thus, SPECTRA conserves the total power required for PV/TV remedy with low latency overhead. However, the SPECTRA framework does not deal with PV and its benefits come with the area and power overheads of the extra MRs and bit-reordering multiplexers [34]. *To address these shortcomings of the SPECTRA framework, we propose a new TV and PV aware reliability management framework called LIBRA, which is described next.*

5 LIBRA Framework: Overview

Our *LIBRA* framework enables reliability-aware run-time PNoC management while rectifying TV and PV in MRs by integrating device-level and system-level enhancements. Fig. 7 gives a high-level overview of our framework. The thermal and process variation aware microring assignment (*TPMA*) mechanism dynamically assigns each MR to the nearest available carrier wavelength, which enables reliable modulation and reception of data while maintaining the maximum possible bandwidth. This device-level mechanism also adaptively chooses the least power-consuming method from thermal tuning and localized trimming as the preferred method for PV/TV remedy, and thus, reduces the total power for PV/TV remedy in the PNoC. However, limiting the peak temperature swings below threshold levels is critical to further reduce the total power for PV/TV remedy. To achieve this, we devise a PV-aware anti-wavelength-shift dynamic thermal management (*VADTM*) scheme that uses support vector regression (SVR) based temperature prediction and dynamic thread migration, to avoid on-chip thermal threshold violations, minimize on-chip thermal hotspots, and reduce thermal tuning power for MRs. The next two sections present details of the *TPMA* and *VADTM* schemes.

6 Thermal and Process Variation Aware Microring Assignment (TPMA)

6.1 Thermal Variation Aware MR Assignment (TMA)

As discussed in Section 3.1, TV shifts MR resonances, which can prevent MRs from reading or writing to their assigned

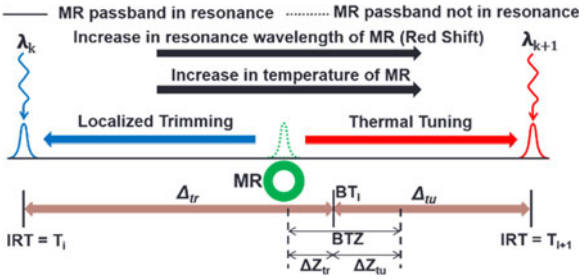

Fig. 8. Red shift of MR with increase in temperature from IRTs T_i to T_{i+1} with trimming and tuning range of temperatures between these IRTs.

carrier wavelengths. Fortunately, there is a linear dependency between temperature increase and resonance wavelength shift [36], which we exploit in our TV-aware microring assignment (*TMA*) mechanism that dynamically assigns each MR to the nearest available carrier wavelength.

Fig. 8 shows how at temperatures T_i and T_{i+1} ($T_{i+1} > T_i$), an MR resonance is in exact alignment with the available wavelengths λ_k and λ_{k+1}, respectively. These temperatures are called ideal resonant temperatures (IRTs). When the MR temperature is in between IRTs T_i and T_{i+1}, as shown in Fig. 8, the MR needs to be either *trimmed* to resonate to λ_k (which is the resonance wavelength of an MR at temperature T_i) or thermally *tuned* to resonate to λ_{k+1} (which is the resonance wavelength of an MR at temperature T_{i+1}). To adaptively choose the least power consuming method from trimming and thermal tuning, we divide the temperature range between IRTs T_i and T_{i+1} into two parts: trimming temperature range (Δ_{tr}) and tuning temperature range (Δ_{tu}). For an MR at temperature T, if $(T_i + \Delta_{tr}) > T > T_i$ we perform trimming as it takes the least power, else if $(T_i + \Delta_{tr}) < T < T_{i+1}$ we perform tuning as it takes the least power (see Fig. 8). At the boundary of the trimming and tuning temperature ranges, where $T_{i+1} - \Delta_{tu} = T_i + \Delta_{tr}$, both trimming and tuning consume equal power, and hence, an MR can be either trimmed or tuned. This temperature is called the boundary temperature (BT_i). It has been shown that for a small resonance wavelength shift (<1 nm), thermal tuning power is higher compared to trimming power to mitigate the same amount of TV-induced shift [9]. Thus, our *TMA* approach considers a higher trimming temperature range compared to tuning temperature range ($\Delta_{tr} > \Delta_{tu}$), to minimize total trimming and tuning power.

In *TMA*, MRs are dynamically shifted (trimmed or tuned) to an appropriate IRT for correct operation based on their current temperature. Figs. 9a, 9b, 9c, 9d show four different MR wavelength assignment configurations at successive IRTs T_1, T_2, T_3, and T_4, where $T_4 > T_3 > T_2 > T_1$. If the MR group temperature T is such that $(T_1 - \Delta_{tu}) < T < (T_1 + \Delta_{tr})$ then the assignment in Fig. 9a is chosen, otherwise if $(T_2 - \Delta_{tu}) < T < (T_2 + \Delta_{tr}), (T_3 - \Delta_{tu}) < T < (T_3 + \Delta_{tr})$, $or (T_4 - \Delta_{tu}) < T < (T_4 + \Delta_{tr})$ then the assignment in Figs. 9b, 9c, or 9d is chosen, respectively. One critical observation in the assignment shown in Fig. 9a is that MRs R_1-R_n are in resonance with $\lambda_1 - \lambda_n$ within the same Free Spectral Range (FSR_i), whereas, in Fig. 9b at IRT T_2, MRs $R_2 - R_n$ are in resonance with $\lambda_1 - \lambda_{n-1}$, respectively in FSR_i and MR R_1 is in resonance with λ_n of the next FSR (i.e., FSR_{i+1}). In this assignment and the ones shown in Figs. 9c and 9d, as explained in Section 4, there is a need to reposition bits in electrical domain using backend barrel-shifters or pipelined shift registers. The assignments shown in Figs. 9b, 9c, and 9d require one, two, and three bit shifts, respectively, to retrieve the original data. In addition, our TMA mechanism can tolerate a range of on-chip temperature variations by employing trimming/tuning for smaller variations and MR reassignment for larger variations.

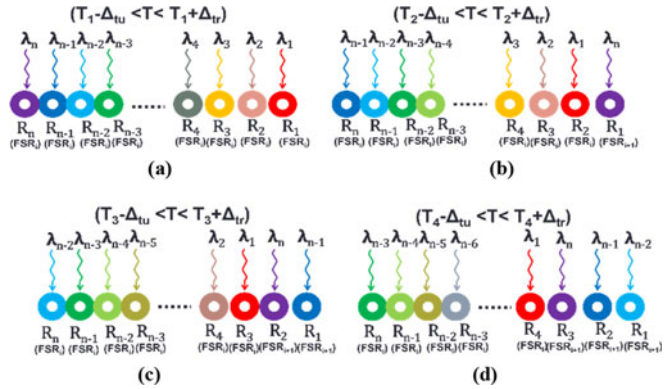

Fig. 9. Thermal aware assignment of microrings (R_{1-n}) to wavelengths (λ_{1-n}) at four successive IRTs T_1, T_2, T_3, and T_4 in *TMA* mechanism.

TMA represents a powerful reactive technique to adapt to on-die thermal variations with low overhead while ensuring reliable and high-bandwidth communication in MR based PNoCs. But there is scope for three further enhancements. First, *TMA* does not consider the impact of PV on MRs, thus there is a need to readapt *TMA* to address the impact of PV on MRs, which is discussed in Section 6.2. Second, there is a need to proactively control the peak on-chip temperature to reduce the range of on-chip temperature swings, which ultimately limits the number of required bit shifts (this work caps the number bit shifts to three as shown Fig. 9d) and reduces the latency to retrieve the original data. Third, at the BT temperature (Fig. 8), maximum trimming or tuning power is required to realign the MR resonances to their nearest carrier wavelengths. Thus, avoiding BT temperatures at MRs can reduce trimming and tuning power overhead. As shown in Fig. 8, we define a boundary temperature zone (BTZ) around each BT_i. This zone includes temperatures T such that $BT_i - \Delta Z_{tr} < T < BT_i + \Delta Z_{tu}$ where ΔZ_{tr} and ΔZ_{tu} are designer specified parameters. Cores with corresponding MR bank temperatures that are within BTZs are called boundary temperature cores (BTCs). As BTCs possess the highest trimming and tuning power overhead for their corresponding MR bank, a mechanism that reduces the number of BTCs can save trimming and tuning power. Section 7 describes such a mechanism, which also controls the range of on-chip temperature swings within allowable limits.

6.2 Readapting TMA for Process Variations (PMA)

In this subsection, we readapt the *TMA* mechanism to address the impact of PV on MR resonances. When using the *TMA* mechanism, PV-induced red or blue shift ($\Delta\lambda_{PV}$) alters the resonance wavelength (λ_{BTi}) of an MR at BT_i to λ_{BTR} or λ_{BTB}, respectively, as shown in Fig. 10. This violates the actual definition of BT, which is the temperature from which either trimming to λ_k (which is the resonance wavelength of an MR at temperature T_i) or tuning to λ_{k+1} (which is the resonance wavelength of an MR at temperature T_{i+1}) dissipates equal power. For example, in case of a PV-induced blue shift, tuning λ_{BTB} to λ_{k+1} would consume more power than trimming it to λ_k, as λ_{BTB} is shifted towards λ_k from λ_{BTi}. Therefore, in the *TMA* that is readapted for process variations

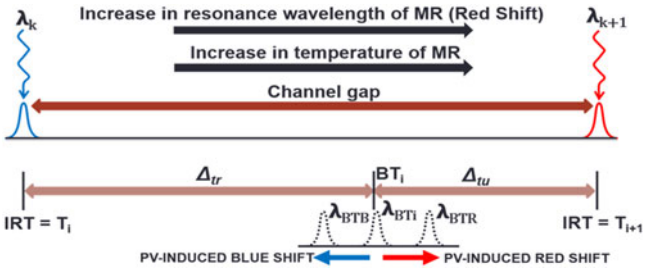

Fig. 10. Impact of PV-induced red and blue shift on boundary temperature on *TMA*.

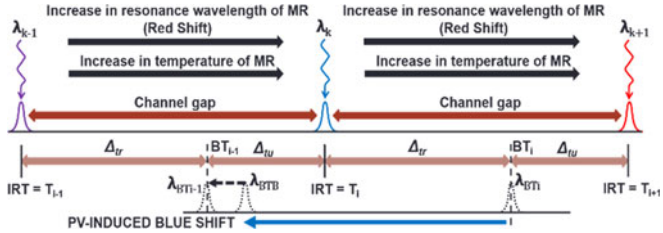

Fig. 11. Boundary temperature adaptation for larger PV-induced blue shifts in *PMA*.

(*PMA* mechanism), we propose to either increase or decrease the BTs in line with the PV-induced red or blue shifts in MR resonances, respectively.

In our *PMA* mechanism, first, the PV-induced resonance shifts in MRs are gauged in situ at initialization by using a dithering signal based control system [40]. The overhead of this in-situ PV detection technique is considered in our results section as dithering power. In our analysis, we model and estimate PV in MRs using the VARIUS tool [51], a description of which is already given in Section 3.3. Once PV-induced red or blue shifts of MRs are determined, we estimate the average resonance shift (in nm) across all MRs of each MR bank. We use each average shift value ($\Delta\lambda_{PV,ave}$) to determine the shift in BT (i.e., ΔBT_i) for all the MRs of the corresponding MR bank using Eq. (3), where TS is the MR thermal sensitivity obtained from Eq. (2) as $\Delta\lambda_r/\Delta T$.

$$\Delta BT_i = \frac{\Delta\lambda_{PV,\,ave}}{TS}, \tag{3}$$

Once the ΔBT_i values for all MR banks of the PNoC are obtained, we revise the BTs of each MR bank by either adding or subtracting the corresponding ΔBT_i value from the original BT. Similar to the *TMA* mechanism, we then build BTZs around these updated BTs. Note that we cannot shift the original BT beyond a particular temperature range (i.e., $\Delta BT_i > \Delta_{tu}$ and $\Delta BT_i < -\Delta_{tr}$), especially when the PV-induced resonance wavelength shifts are greater than one channel gap (CG). Unfortunately, for state-of-the-art fabrication processes, the maximum PV-induced wavelength shifts are around $\pm$ 3 nm ($>$ one channel gap of 1.48 nm). Shifting BT beyond a certain range to compensate for larger PV-induced shifts will also lead to higher tuning and trimming power dissipation.

Fig. 11 shows an example of a larger PV-induced blue shift, which alters the resonance wavelength (λ_{BTi}) of an MR at BT to λ_{BTB}. One possible solution is to bring back the resonance wavelength to λ_{BTi}. But this is not always possible especially when the chip is operating at lower temperatures. Therefore, we propose to shift this λ_{BTB} to $\lambda_{\mathrm{BTi\text{-}1}}$ instead of λ_{BTi}, i.e., instead of decreasing BT by a larger amount here we increase BT by a smaller amount. In order to facilitate this shifting, similar to *TMA*, we perform ring assignment along with extra bit shifts. At a channel spacing of 1.48 nm, to compensate for peak PV-induced resonance shift of $\pm$ 3 nm, two extra bit shifts (forward and backward bit shifts to compensate positive and negative PV induced resonance shift) are needed.

Overheads: Our proposed *TPMA* scheme is a combination of the two previously proposed techniques: *TMA* (Section 6.1) and *PMA* (this subsection). *TPMA* requires a maximum of five bit shifts, which include three for *TMA* and two for *PMA*. These additional bit shifts in *TPMA* incur latency overhead. This latency overhead is quantified in more detail in Section 8. Furthermore, with *TPMA* each MR bank requires a Read Only Memory (ROM) to store its corresponding three BT values, which are determined using PV profiling at design time, as discussed earlier. This ROM also stores beginning and ending temperatures of three BTZs in each MR bank. We have considered 16-bits to store each temperature value. As there is a need to store nine different temperature values (three BTs, three BTZ start temperatures, three BTZ end temperatures) for each MR bank, we need a ROM that can store 144-bits. Moreover, a 16-bit comparator circuit is needed for each MR bank to determine the range of operation of MRs (i.e., trimming or tuning temperature range). This comparator is also used to determine whether an MR bank is in BTZ or not. Therefore, one input for this comparator comes from a thermal sensor (i.e., information on current temperature) and the other input is from the ROM. The area and power overhead of the ROM and comparator is quantified in detail in Section 8.

7 Variation Aware Anti Wavelength-Shift Dynamic Thermal Management (VADTM)

To proactively reduce thermal hotspots (which in turn will limit the maximum number of bit shifts to five) and control on-die temperature (to reduce the number of BTCs), we propose a system-level variation aware anti wavelength-shift dynamic thermal management (*VADTM*) technique, described below.

7.1 Objective

The primary goals with *VADTM* is to maintain the temperature of all of the cores on a die below a specified thermal threshold, i.e., for all cores $1 \leq i \leq N, T_i < T_t$ where T_i is the temperature of core *i* and T_t is threshold temperature. We utilize support vector based regression (SVR) to predict the future temperature of a core, as SVR achieves better accuracy compared to several previously proposed statistical, learning-based, and regression-based algorithms by ultimately reducing number of false positives [52]. This predicted temperature is compared with a thermal threshold to determine the potential for a thermal emergency. If such a potential exists, threads are migrated to available BTCs. These BTCs are determined based on the PV profile of MRs and ring blocks that are used to send and receive data from these cores. Migration to a BTC has a twofold benefit. First by moving the thread away from a core that could suffer a thermal emergency, we limit the required bit shifts and related performance overhead in the MR groups of that core. Second, by moving the thread to a BTC, the temperature of the BTC will increase resulting in that core no longer being a BTC (consequently the temperature of the core's MR groups will also increase, taking them

TABLE 3
List of VADTM Parameters and Their Definitions

Symbol	Definition
IPC_i	Instructions per cycle of ith core
CT_i	Current temperature of ith core
TN_i	Average temperature of immediate neighboring cores of i^{th} core; if this core is on chip periphery and missing neighbors, then we consider virtual neighbor cores at ambient temperature in lieu of the missing cores
PT_i	Predicted temperature of i^{th} core
T_t	Thermal threshold
$BTCs$	Boundary temperature cores
$NBTCs$	Non-boundary temperature cores
C	Regularization parameter
W	Weight vector for regression
x_i and y_i	Input and outputs in training and test data
ξ_i	Slack variables
E	Error function
B	Bias for cost function

outside of their BTZ and closer to IRTs, which will reduce trimming/tuning power). The parameters used to describe *VADTM* are shown in Table 3.

7.2 Temperature Prediction Model

We designed a support vector regression (SVR) based temperature predictor that accepts input parameters reflecting the workload for the core under consideration, in terms of instructions per cycle (IPC_i), as well as the current core temperature (CT_i) and temperatures of surrounding cores (TN_i). Once these parameter values are obtained at runtime, the future temperature for the core can be predicted.

Architecture: A typical SVR [52] relies on a prediction model that ignores errors that are situated within ε distance of the true value. This type of a prediction model is called an ε-insensitive prediction model. Fig. 12 shows an example of a one-dimensional non-linear SVR based prediction model with an ε-insensitive band. The variables ***(ξ and ε)*** measure the cost of the errors on the training points. These are zero for all points that are inside the ε-insensitive band.

SVR is primarily designed to perform linear regression. To handle non-linearity in data, SVR first maps the input x_i onto an m-dimensional space using some fixed (non-linear) mapping notated as Φ, and then a linear model is constructed in this high-dimensional space as shown in eqs. (5) and (6) below. Thus, it overcomes drawbacks of linear and logistic regression towards handling non-linearity in data. This class of SVRs is called *kernel based SVRs* which use kernel κ as shown in Eq. (7) for implicit mapping of non-linear training data (as shown in Fig. 12) into a higher dimensional space.

$$CF = min\ \frac{1}{2}\ W^T.W + C\sum_{i=1}^{n}(\xi_i + \xi_i{}^*). \tag{4}$$

Subject to:

$$y_i - W^T\Phi(\ x_i) - b\ \leq \varepsilon +\ \xi_i\ \ (\xi_i \geq 0, i = 1,\ 2,\ \ldots, n) \tag{5}$$

$$W^T\Phi(\ x_i) + b -\ y_i \leq \varepsilon +\ \xi_i{}^*\ (\xi_i{}^* \geq 0,\ i = 1,\ 2,\ \ldots, n) \tag{6}$$

$$\kappa(\ x_i,\ x_j) = \Phi(x_i)^T\Phi(x_j). \tag{7}$$

SVR performs linear regression in this high-dimension space using ε-insensitive loss and, at the same time, tries to reduce model complexity by minimizing $W^T.W$. This can be described by introducing (non-negative) slack variables ξ_i and $\xi_i^*\ (i = 1\ to\ n)$, to measure the deviation of training samples outside the ε-insensitive band. Thus SVR is formulated with the goal of minimizing the cost function in Eq. (4) with the constraints shown in eqs. (5) and (6).

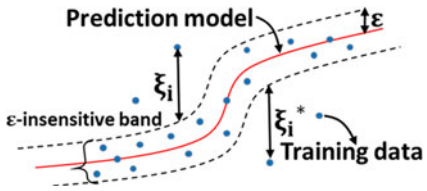

Fig. 12. Non-linear support vector based regression prediction model.

As on-chip temperature variation data is non-linear in the original space, our SVR model employs a kernel based regression which uses a Radial Basis Function (RBF) [25] (Gaussian kernel) as shown in Eq. (8). The RBF kernel improves the accuracy of SVR when data has non-linearity in the original space. We performed a sensitivity analysis (SA) to determine regularization parameter (C) and 'gamma' (γ) values of the kernel based SVR (see Section 7.1 for chosen values). This SA overcomes the possibility of over fitting of training data and improves accuracy further.

$$\kappa(\ x_i,\ x_j) = \exp\left(-\gamma\left|\ x_i -\ x_j\right|^2\right). \tag{8}$$

Training and Accuracy: We trained our SVR model using a set of multi-threaded applications from the SPLASH-2 [45] and PARSEC [46] benchmark suites, specifically: blackscholes (BS), bodytrack (BT), vips (VI), facesim (FS), fluidanimate (FA), swaptions (SW), barnes (BA), fft (FFT), radix (RX), radiosity (RD) and raytrace (RT) with different thread counts: 2, 4 and 8. We considered different combinations of thread mappings on a 9-core (3×3) floorplan, to train our predictor to determine the temperature of the center (target) core. As the future temperature of a target core is dependent on the average temperature of its immediate neighboring cores, we trained our SVR model with temperature inputs from the target core running a single thread, as well as its surrounding cores running a variable number of threads. Simulations for each of these floorplans allowed us to obtain data to train our SVR model. This data included temperature for the target core and its neighboring core temperatures, as well as instructions per cycle (IPC) for the target core. IPC is very useful to determine if there is a phase change in an application and plays a crucial role in maintaining future temperature prediction accuracy, especially when temperatures of a target core and its neighbors are similar at a given time. Our training algorithm involved an iterative process that adjusts the weights and bias values in the SVR shown in eqs. (4), (5) and (6) to fit the training set.

We verified the accuracy of our SVR model for multi-threaded benchmark workloads (we considered 6000 floorplans, with 70 percent of input data for training and 30 percent for testing) and found that it has an accuracy of over 95 percent. Figs. 13a and 13b show actual and predicted on-chip temperature variations for a 64-core platform executing 32 threads of the FA and RD benchmarks. From these figures it can be seen that our temperature predictor tracks temperature quite accurately. When predicted temperature exceeds the thermal threshold our thread migration mechanism (which is discussed next) migrates threads from hotter cores

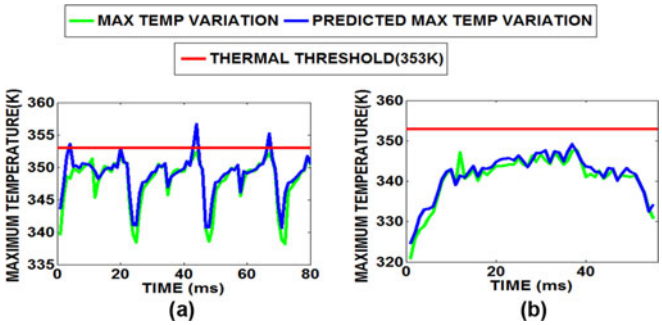

Fig. 13. Actual and predicted maximum temperature variation with execution time for (a) fluidanimate *(FA)* and (b) radiosity *(RD)* benchmark applications executed on 64-core platform executing 32 threads.

to cooler cores to keep overall maximum temperature below the threshold.

7.3 Thermal Management Framework

Fig. 14 illustrates the entire *VADTM* technique. For each core, we periodically monitor the IPC value from performance counters and temperature of each of its MR banks from on-chip thermal sensors. A unique digital thermal sensor (DTS) is used to monitor temperature of every MR bank. Furthermore, the monitored temperature value of each MR bank is stored in a bank specific register. In addition, each core employs an 8-bit adder circuit and a bit shifter, along with a mechanism of accessing the current temperature values of all MR banks of neighboring cores from their specific registers, to determine the average temperature of all MR banks of neighboring cores. We have considered the latency and power overheads of these circuits in our results section (i.e., Section 8.2). If a thermal emergency is predicted for a core by the SVR predictor, then *VADTM* initiates a thread migration procedure, otherwise no action is taken.

Algorithm 1 shows the pseudo-code for the *VADTM* thread migration procedure. First, the future temperature (PT_i) of the ith core is predicted using the SVR based predictor with inputs: core temperature (CT_i), core IPC (IPC_i), and temperature of neighboring cores (TN_i) in steps **1-3**. The list of available BTCs (i.e., those that are not currently executing any thread) and available NBTCs is obtained in steps **4-10**. In steps **11-12**, a loop iterates over all cores and checks for possible thread migration conditions (i.e., thermal emergency cases where current core predicted temperature (PT_i) is greater than thermal threshold (T_t)). If a thread migration is required, then in steps **13-21**, we check for free BTCs, and if they are available then we migrate the thread from the current core to the BTC with lowest temperature, else we migrate the thread to a free NBTC with lowest temperature. This *VADTM* thread migration procedure is invoked at every epoch (1ms).

8 Experiments

8.1 Experimental Setup

We target a 64-core manycore system to evaluate our *LIBRA* $(TPMA + VADTM)$ framework. Microarchitecural parametrs of the many core system is presented in Table 1. We evaluate LIBRA on two well-known PNoC architectures: Corona [3] and Flexishare [5]. Corona uses a 64×64 multiple write single read (MWSR) crossbar with token slot arbitration. Flexishare uses 32 multiple write multiple read (MWMR) waveguide groups with 2-pass token stream

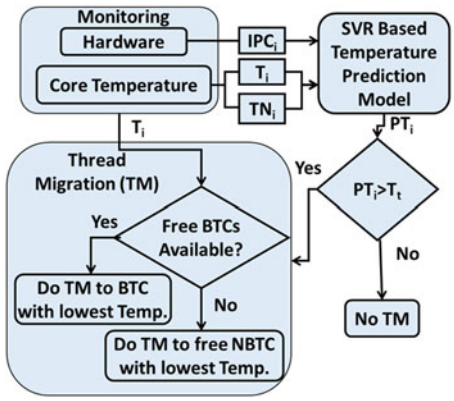

Fig. 14. Overview of *VADTM* in *LIBRA* framework with support vector regression (SVR) based temperature prediction model.

arbitration. Each MWSR waveguide in Corona and MWMR waveguide in Flexishare is capable of transferring 512 bits of data from a source node to a destination node. As each core in our considered manycore system has private L1 and L2 caches, electrical interconnects are used for communications between the L1 and L2 caches at each core, and a photonic network is used for communications between the L2 caches of cores and memory controllers.

Algorithm 1. *VADTM* thread migration algorithm

Inputs: Current core temperature (CT_i), average neighboring core temperature (TN_i), current core IPC (IPC_i)

1: **for** each core i **do** // *Loop that predicts future temperature*
2: PT_i = SVR_predict_future_temperature (CT_i, TN_i, IPC_i)
3: **end for**
4: **for** each core i **do** // *Loop that checks for free BTCs and NBTCs*
5: **if** CT_i in BTZ **and** $IPC_i == 0$ **then**
6: List_BTC = Push i //*add core to BTC list*
7: **else if** $IPC_i == 0$ **then**
8: List_NBTC = Push i //*add core to NBTC list*
9: **end if**
10: **end for**
11: **for** each core i **do** // *Loop that performs thread migration*
12: **if** $PT_i \geq T_t$ **then**
13: **if** List_BTC $\neq \{\}$ **then**
14: Migrated_core = Find_min_temperature_core(List_BTC)
15: Do_thread_migration(core_i $\rightarrow$ Migrated_core)
16: List_BTC = Pop i
17: **else if** List_NBTC $\neq \{\}$ **then**
18: Migrated_core = Find_min_temperature_core(List_NBTC)
19: Do_thread_migration(core_i $\rightarrow$ Migrated_core)
20: List_NBTC = Pop i
21: **end if**
22: **end if**
23: **end for**

Output: Thread migration to BTC or NBTC cores

We modeled and simulated these architectures with the *LIBRA* framework for multi-threaded applications from the SPLASH-2 [45] and PARSEC [46] benchmark suites as explained in Section 3. Simulations were performed for entire regions-of-interest (ROIs) where parallelized version of these applications were executing on the 64-core system. Power and instruction traces for the benchmark applications were generated using the Sniper 6.0 [44] simulator and McPAT [47]. We used the 3D-ICE tool [49] for thermal

analysis. The ambient temperature was set to 303K and the thermal threshold (T_t) was set to 353K.

We model and consider area, power, and performance overheads for our framework in our analysis. *LIBRA* considers the power overhead of the dithering signal based control mechanism to be 385μW per MR [59]. *LIBRA* with both Corona and Flexishare PNoCs has an electrical area overhead of 0.34 mm^2 and a power overhead of 57 mW using gate-level analysis and the CACTI 6.5 [56] tool for memory and comparators. The MR trimming power is set to 130 μW/nm [15] for current injection (blue shift) and tuning power is set to 240 μW/nm [9] for heating (red shift). From [58], tuning/trimming power values are linear with resonance shifts of up to 3.5 nm for various sizes of MRs. Because of this, and as our TPMA technique requires the use of localized trimming and thermal tuning methods to compensate for resonance shifts of less than one channel gap (<1.5 nm) only, we assume that the tuning/trimming power values are linear with resonance shifts. To compute laser power, we considered detector responsivity as 0.8 A/W [53], MR through loss as 0.02 dB, waveguide propagation loss as 0.274 dB/cm, waveguide bending loss as 0.005 dB/90^0, and waveguide coupler/splitter loss as 0.5 dB [28], [53]. We calculated photonic loss in components using these values, which sets the photonic laser power budget and correspondingly the electrical laser power. For energy consumption of photonic devices, we adapt parameters from [53], with 0.42 pJ/bit for every modulation and detection event, and 0.18 pJ/bit for the driver circuits of MR modulators and photodetectors. Based on our sensitivity analysis we get the best accuracy with our SVR-based temperature predictor when parameters C and γ are set to 1000 and 0.1 respectively.

Thread migration model and overhead analysis: We have considered the overheads of transferring architectural register values and L1-L2 cache misses incurred due to thread migration. In our thread migration model, the register values (ten $32\text{-bit} + \text{six } 16\text{-bit} + \text{eight } 80\text{-bit registers} = 1056$-bit register values) of the source core are packed in three 512-bit packets and then these packets are transferred to the migrated core on the PNoC. At the migrated core, the register values in these packets are unpacked before being used to populate the local registers. We assume that packing and unpacking of register values takes 16 cycles each. After this transition phase, during which the register values are transferred between the cores, thread execution begins at the migrated core. The initial phase of execution at the migrated core results in L1 and L2 cache misses due to cold start effects. During this cold-start phase, the source core keeps its cache contents intact and the corresponding migrated core accesses the source core's L2 cache upon cold cache misses. Accessing the source core's L2 cache is possible through the PNoC using the source core's microrings, as we do not turn off or lose control of the microrings of the inactive source core. In case of a cache miss at the source core during the cold-start phase, the requested line is fetched from the off-chip memory via a memory controller. Thus, upon an L2 cache miss at the migrated core during the cold-start phase, the requested cache line is served either by the L2 cache of the source core or the off-chip memory. The overheads of such cold cache misses at the migrated core depend on several factors, including the underlying PNoC architecture, cache microarchitecture, locations of the source and migrated cores, locations of memory controllers, network traffic conditions (or program/application type), and program/application phase. The overheads of transferring the register values from the source core to the migrated core depend on factors such as the PNoC architecture, locations of cores, and network traffic conditions. Our trace-driven simulator, which we use to model the manycore system operation, captures the effect of all these dependence factors in a cycle-accurate manner and accounts for the overheads of thread migration.

Closed-loop thermal simulation framework: We use a closed-loop thermal simulation framework for implementing, evaluating, and comparing our LIBRA framework with various thermal management techniques from prior work. Our closed-loop thermal simulation framework utilizes coarse-grained core's activity traces extracted from the interval simulation of the 64-core system based on the Sniper simulator [44]. These coarse-grained cores' activity traces are fed into McPAT [47] to generate power traces, which in turn are fed into a 3D-ICE [49] based model of the manycore system with the PNoC to generate thermal/temperature traces. In addition to these traces, our framework employs a cycle-accurate network-centric simulator that models the behavior and operation of our considered manycore system with the PNoC of interest and evaluates the overhead of thread migration events. These core's activity traces and related temperature traces engage with our SVR-based thermal predictor and thread migration scheduler in a closed loop. If the threads are migrated due to thermal emergencies, our scheduler updates the activity traces based on the revised threads-to-cores mapping and incurred overheads that are evaluated using our cycle-accurate simulator. Then, based on the revised activity traces, new power and temperature traces are generated. These traces are used to evaluate total power dissipation, energy consumption, and total performance overhead in terms of execution time.

As presented in Section 5, to minimize trimming and tuning power consumption for a fixed channel gap of 1.48 nm, trimming temperature range (Δ_{tr}) and tuning temperature range (Δ_{tu}) for *TPMA* are calculated as 8.73K and 4.72K, respectively.

8.2 Comparison Results

We compared the performance of our *LIBRA* framework with four prior works on manycore thermal management: a ring aware policy (RATM) [16], frequency align policy (FATM) [17], a predictive dynamic thermal management (PDTM) framework [18], and the SPECTRA framework from our prior work [10]. RATM distributes threads uniformly across cores that are closer to PNoC nodes first and then distributes the remaining threads in a regular pattern from outer cores to inner cores. FATM distributes threads across cores based on the process variation profile of ring blocks that are in the proximity of these cores. PDTM uses a least square based regression temperature predictor to determine if the predicted temperature of a core exceeds a thermal threshold, and if so then thread migration is performed from that core to the coolest core that is not executing any threads. SPECTRA performs ring assignment at the device-level and SVR prediction based proactive thread migration at the system-level for thermal reliability management in PNOCs.

Figs. 15a-15b show the maximum temperature obtained with the five frameworks across eleven applications from the PARSEC and SPLASH-2 benchmarks suites with 48 and 32 thread counts executing on a 64-core system with the Corona PNoC [3] architecture. As *LIBRA* and FATM perform thread

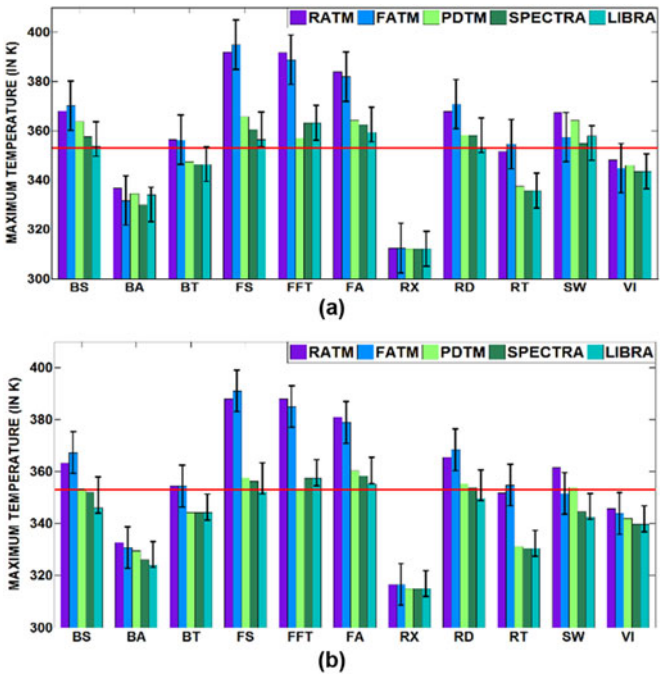

Fig. 15. Maximum temperature comparison for *LIBRA* with RATM [16], FATM [17], PDTM [18], and SPECTRA [10], for (a) 48 thread, and (b) 32 thread PARSEC and SPLASH-2 benchmarks executing on 64-core manycore system with Corona PNoC. Bars show mean values of maximum temperature across 100 PV maps; confidence intervals show variation in maximum temperature.

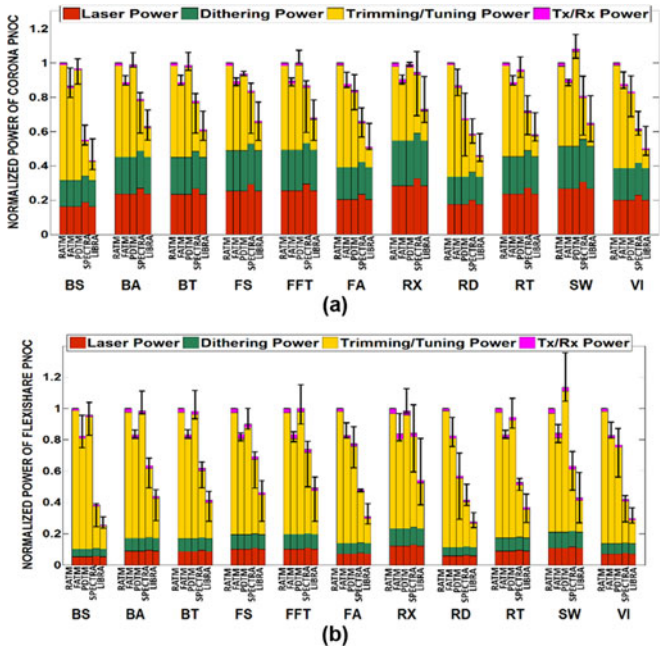

Fig. 16. Normalized power dissipation (Laser Power, Dithering Power, Trimming/Tuning power, and Modulating and Detecting (Tx/Rx) Power) comparison for *LIBRA* with RATM [16], FATM [17], PDTM [18], and SPECTRA [10] for 48 threaded applications of PARSEC and SPLASH-2 suites executed on (a) Corona and (b) Flexishare PNoC architectures for a 64-core manycore system. Results shown are normalized w.r.t RATM, therefore, RATM does not have confidence intervals. Bars show mean values of power dissipation across 100 PV maps; confidence intervals show variation in power dissipation.

management based on the PV profile of MRs, only these frameworks have confidence intervals in Fig. 15. From Fig. 15a it can be observed that some applications (e.g., *FA*, *SW*) with 48 threads exceed the threshold (353K) for all frameworks, as there are insufficient number of free cores on the chip whose temperature is below the thermal threshold to migrate threads. However, with a more manageable number of threads, the situation improves. In Fig. 15b, for the case with 32 threads, our *LIBRA* framework avoids violating thermal thresholds for very small number of benchmark applications with 32 threads. On average, *LIBRA* has 14.6K and 17.5K lower maximum temperature compared to the RATM policy for 48 and 32 threads, respectively. In addition, on average *LIBRA* has 13.5K and 15.9K lower maximum temperature compared to the FATM policy for 48 and 32 threads, respectively. *LIBRA* migrates threads from hotter cores to cooler cores to control maximum temperature, whereas no thread migration is performed in both RATM and FATM when the on-chip thermal threshold temperature (i.e., 353K) is reached, as these mechanisms are simple thread allocation policies without control on peak temperature. For most of the benchmarks, maximum temperatures with PDTM, SPECTRA, and *LIBRA* are below the thermal threshold. However, on average *LIBRA* has 3.2K and 3.5K lower maximum temperature compared to PDTM for 48 and 32 threads, respectively. This is because LIBRA employs a more accurate (less false-positives) SVR based prediction approach, which reduces the number of mispredictions compared to the low accuracy (more false-positives) of the least square regression mechanism in PDTM. Lastly, *LIBRA* has a 0.8K and 1.9K lower maximum temperature compared to SPECTRA for 48 and 32 threads, respectively. Even though both *LIBRA* and SPECTRA prefer to migrate threads to BTCs, the maximum temperatures with *LIBRA* are sometimes lower compared to SPECTRA, as *LIBRA* is able to perform thread migrations more often to lower temperature BTCs compared to SPECTRA.

In the interest of brevity, we do not show maximum temperature results for the Flexishare PNoC architecture. We observed a similar trend in maximum temperature variations for Flexishare as we did for Corona (Fig. 15).

Fig. 16 shows the power dissipation comparison for the five frameworks across multiple 48-threaded applications for the Corona [3] and Flexishare [5] PNoC architectures, respectively. One of the main reasons why *LIBRA* has lower power dissipation than RATM, FATM, and PDTM is that it more aggressively reduces trimming and tuning power in both Corona and Flexishare PNoCs. From Fig. 16a, *LIBRA* has 74.5, 67.4, and 70.8 percent lower trimming and tuning power on average compared to RATM, FATM, and PDTM for Corona. Furthermore, from Fig. 16a, *LIBRA* also has 76.2, 68.3, and 72.5 percent lower trimming and tuning power on average compared to RATM, FATM, and PDTM for Flexishare. The *TPMA* technique in *LIBRA* intelligently conserves trimming and tuning power compared to RATM, FATM, and PDTM by performing process variation aware MR reassignment, while our *VADTM* further improves trimming and tuning power savings with its intelligent thread migration to BTCs. Lastly, the *TPMA* mechanism in *LIBRA* adapts intelligently to the PV profiles of MRs, reducing it's trimming and tuning power dissipation by 46.3 and 48.1 percent, compared to SPECTRA for the Corona and Flexishare architectures, respectively.

Fig. 16 also shows the laser power comparison of the five frameworks for the Corona and Flexishare architectures. It can be observed that Corona and Flexishare with *LIBRA* need similar laser power as Corona and Flexishare

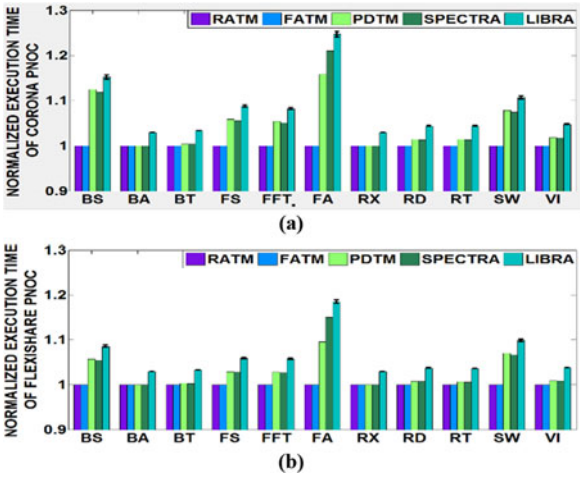

Fig. 17. Normalized average execution time comparison of *LIBRA* with RATM [16], FATM [17], PDTM [18], and SPECTRA [10] for (a) Corona and (b) Flexishare PNoCs for 48 threaded applications from PARSEC and SPLASH-2 suites executed on 64-core system. Results shown are normalized wrt RATM. Bars show mean values of execution time across 100 PV maps; confidence intervals show variation in execution time.

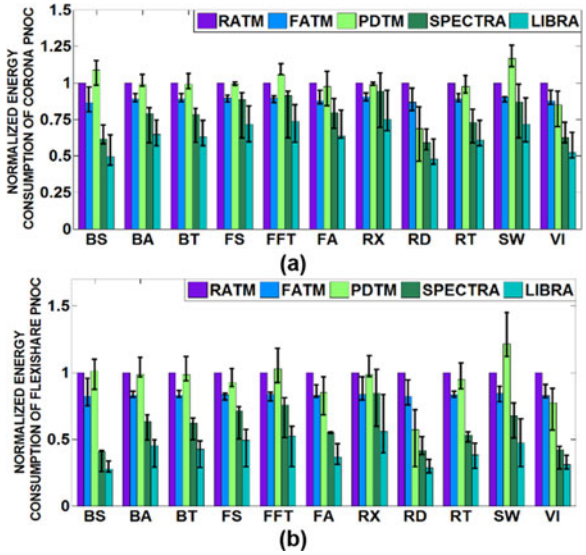

Fig. 18. Normalized energy consumption comparison of LIBRA with RATM [16], FATM [17], PDTM [18], and SPECTRA [10] for (a) Corona and (b) Flexishare PNoCs for 48 threaded applications from PARSEC and SPLASH-2 suites executed on a 64-core system. Results shown are normalized wrt RATM, therefore, RATM does not have confidence intervals. Bars show mean values of energy consumption across 100 PV maps; confidence intervals show variation in energy consumption.

architectures with RATM, FATM, and PDTM. Furthermore, *LIBRA* requires 12.9 and 6.4 percent lesser laser power compared to SPECTRA for Corona and Flexishare. The extra MRs used in SPECTRA to compensate for TV-induced resonance shifts contribute to the increase in laser power compared to *LIBRA* for both architectures. From these results it can also be observed that the laser power saving in Corona is higher than for the better performance optimized architecture of Flexishare.

In summary, *LIBRA* saves considerable trimming/tuning power to ultimately achieve overall power reduction. From the power analysis in Fig. 16a, *LIBRA* with Corona has 40.8, 34.1, 37.2, and 21.4 percent lower total power dissipation compared to Corona with RATM, FATM, PDTM, and SPECTRA, respectively. Further from Fig. 16b it can be seen that Flexishare with *LIBRA* has 61.3, 52.9, 57.4, and 32.8 percent lower power dissipation compared to Flexishare with RATM, FATM, PDTM, and SPECTRA, respectively.

Fig. 17 shows the average execution time comparison between the five frameworks across the 11 48-threaded applications from PARSEC and SPLASH-2 suites, for the Corona and Flexishare PNoC architectures, respectively. As only *LIBRA* performs thread migration based on the PV profile of MRs, therefore, this framework has confidence intervals on execution time shown in Fig. 17. From Fig. 17a it can be seen that Corona with *LIBRA* has 8.3 percent higher execution time compared to Corona with RATM and FATM. Corona with *LIBRA* needs extra execution time to migrate threads between cores and to reorder bits using shift registers whereas the RATM and FATM policies simply schedule threads without any thread migration and bit reorder, and thus do not possess such overheads. Further, Corona with *LIBRA* has 3.3 percent higher execution time compared to PDTM. Despite *LIBRA* using a faster SVR based temperature predictor compared to a more complex least square based regression predictor in PDTM, the higher number of thread migrations (to adapt to PV profiles of MRs) and bit reordering operations in *LIBRA* contribute to an increase in execution time. Similarly, from Fig. 17b Flexishare architecture with *LIBRA* has 6.3 percent higher execution time compared to Flexishare with RATM and FATM. In addition, *LIBRA* also has 3.1 percent higher execution time compared to Flexishare with PDTM. The figures also indicate that the execution time overheads for *LIBRA* are lower when utilizing the faster Flexishare architecture compared to the slower Corona architecture. Moreover, the bit-shifting overhead in *LIBRA* increases its execution time by 3.9 and 3 percent compared to the SPECTRA framework with Corona and Flexishare PNoCs, respectively. From the execution time results, it can be summarized that the significant power benefits achieved with *LIBRA* come at some cost: an increase in execution time.

Lastly, from the power dissipation and execution time results, we obtain energy consumption results for the five frameworks, as shown in Fig. 18. On average, for Corona, energy consumption with *LIBRA* is 36.8, 28.8, 35.3, and 18.6 percent lower compared to RATM, FATM, PDTM and SPECTRA, respectively. For the Flexishare architecture, *LIBRA* has 58.7, 50.3, 55.8, and 30.8 percent lower energy consumption compared to RATM, FATM, PDTM, and SPECTRA respectively.

In summary, from the above results, it is apparent that our proposed PV-aware *LIBRA* framework outperforms previously proposed approaches for thermal management in manycore systems with PNoCs by combining a novel reactive device-level technique (*TPMA*) that improves waveguide channel utilization with a novel system-level proactive thread migration technique (*VADTM*). The excellent power and energy savings compared to previous approaches strongly motivate the use of our LIBRA framework in future PNoC based manycore architectures.

9 Conclusion

In this paper, we have presented the *LIBRA* framework that combines two novel dynamic thermal management mechanisms for the reduction of maximum on-chip temperature and conservation of trimming and tuning power of MRs in DWDM-based PNoC architectures. These techniques (*TPMA*

at the device-level, *VADTM* at the system-level) constitute a hybrid reactive-proactive management framework that demonstrated interesting trade-offs between performance and power/energy across two different state-of-the-art crossbar-based PNoC architectures. Our experimental analysis on the well-known Corona and Flexishare PNoC architectures has shown that *LIBRA* can notably conserve total power by up to 61.3 percent (trimming and tuning power by up to 76.2 percent) and total energy by up to 57.3 percent.

Acknowledgments

This research is supported in part by grants from SRC, NSF (CCF-1252500), and AFOSR (FA9550-13-1-0110).

References

[1] Y. A. Vlasov, "Silicon CMOS-integrated nano-photonics for computer and data communications beyond 100G," *IEEE Comm. Mag.*, vol. 50, no. 2, pp. 67–72, Feb. 2012.
[2] J. D. Owens, et al., "Research challenges for on-chip interconnection networks," *IEEE Micro*, vol. 27, no. 5, pp. 96–108, Sep./Oct. 2007.
[3] D. Vantrease, et al., "Light speed arbitration and flow control for nanophotonic interconnects," in *Proc. 42nd Annu. IEEE/ACM Int. Symp. Microarchitecture*, 2009, pp. 304–315.
[4] S. V. R. Chittamuru, D. Dang, S. Pasricha, and R. Mahapatra, "BiGNoC: Accelerating big data computing with application-specific photonic network-on-chip architectures," *IEEE Trans. Parallel Distrib. Syst.*, Early Access, p. 1, 2018.
[5] Y. Pan, J. Kim, and G. Memik, "Flexishare: Channel sharing for an energy efficient nanophotonic crossbar," in *Proc. 16th Int. Symp. High-Perform. Comput. Archit.*, 2010, pp. 1–12.
[6] S. V. R. Chittamuru, S. Desai, and S. Pasricha, "A reconfigurable silicon-photonic network with improved channel sharing for multicore architectures," in *Proc. 25th Edition Great Lakes Symp. VLSI*, May 2015, pp. 63–68.
[7] I. Thakkar, S. V. R. Chittamuru, and S. Pasricha, "Improving the reliability and energy-efficiency of high-bandwidth photonic NoC architectures with multilevel signaling," in *Proc. 11th IEEE/ACM Int. Symp. Netw.-on-Chip*, Oct. 2017, pp. 1–8.
[8] S. Bahirat and S. Pasricha, "OPAL: A multi-layer hybrid photonic NoC for 3D ICs," in *Proc. 16th Asia South Pacific Des. Autom. Conf.*, Jan. 2011, pp. 345–350.
[9] C. Sun, et al., "Single-chip microprocessor that communicates directly using light," *Nature*, vol. 528, pp. 24–31, 2015.
[10] S. V. R. Chittamuru and S. Pasricha, "SPECTRA: A framework for thermal reliability management in silicon-photonic networks-on-chip," in *Proc. 29th Int. Con. VLSI Des./15th Int. Conf. Embedded Syst.*, 2016, pp. 86–91.
[11] S. K. Selvaraja, "Wafer-scale fabrication technology for silicon photonic integrated circuits," PhD thesis, Ghent University, 2011.
[12] C. Batten, et al., "Building manycore processor-to-DRAM networks with monolithic CMOS silicon photonics," *IEEE Micro*, vol. 29, no. 4, pp. 8–21, 2009.
[13] A. Kumar, et al., "HybDTM: A coordinated hardware-software approach for dynamic thermal management," in *Proc 43rd Annu. Des. Autom. Conf.*, 2006, pp. 548–553.
[14] Z. Li, et al., "Reliability modeling and management of nanophotonic on-chip networks," *IEEE Trans. Very Large Scale Integration Syst.*, vol. 20, no. 1, pp. 98–111, 2012.
[15] C. Li, et al., "Silicon photonic transceiver circuits with microring resonator bias-based wavelength stabilization in 65 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 49, no. 6, pp. 1419–1436, Jun. 2014.
[16] T. Zhang, et al., "Thermal management of manycore systems with silicon-photonic networks," in *Proc. Des. Autom. Test Europe Conf. Exhib.*, 2014, pp. 1–6.
[17] J. L. Abellan, et al., "Adaptive tuning of photonic devices in a photonic NoC through dynamic workload allocation," *IEEE Trans. Comput.-Aided Des. Integrated Circuits Syst.*, vol. 36, no. 5, pp. 801–814, May 2017.
[18] I. Yeo, C. Liu, and E. Kim, "Predictive dynamic thermal management for multicore systems," in *Proc. 45th ACM/IEEE Des. Autom. Conf.*, 2008, pp. 734–739.
[19] L. Zhou and A. K. Kodi, "PROBE: Prediction-based optical bandwidth scaling for energy-efficient NoCs," in *Proc. 17th IEEE/ACM Int. Symp. Netw.-on-Chip*, 2013, pp. 1–8.
[20] A. K. Coskun, et al., "Static and dynamic temperature-aware scheduling for multiprocessor SoCs," *IEEE Trans. Very Large Scale Integration Syst.*, vol. 16, no. 9, Sep. 2008.
[21] H. Hanson, et al., "Thermal response to DVFS: Analysis with an Intel Pentium M," in *Proc. Int. Symp. Low Power Electron. Des.*, 2007, pp. 219–224.
[22] E. Ipek, et al., "Core fusion: Accommodating software diversity in chip multiprocessors," in *Proc. 34th Annu. Int. Symp. Comput. Archit.*, 2007, pp, 186–197.
[23] S. V. R. Chittamuru, I. Thakkar, and S. Pasricha, "Process variation aware crosstalk mitigation for DWDM based photonic NoC architectures," in *Proc. 17th Int. Symp. Quality Elect. Des.*, Mar. 2016, pp. 57–62.
[24] I. Thakkar, S. V. R. Chittamuru, and S. Pasricha, "Mitigation of homodyne crosstalk noise in silicon photonic NoC architectures with tunable decoupling," in *Proc. Int. Conf. Hardware/Softw. Codesign Syst. Synthesis*, Oct. 2016, pp. 1–10.
[25] S. V. R. Chittamuru and S. Pasricha, "Improving crosstalk resilience with wavelength spacing in photonic crossbar-based network-on-chip architectures," in *Proc. IEEE 58th Int. Midwest Symp. Circuits Syst.*, Aug. 2015, pp. 1–4.
[26] S. V. R. Chittamuru, I. Thakkar, and S. Pasricha, "HYDRA: Heterodyne crosstalk mitigation with double microring resonators and data encoding for photonic NoCs," *IEEE Trans. Very Large Scale Integration Syst.*, vo1. 26, no. 1, pp. 168–181, Jan. 2018.
[27] M. M. Sabry, A. Sridhar, and D. Atienza, "Thermal balancing of liquid-cooled 3D-MPSoCs using channel modulation," in *Proc. Des. Autom. Test Europe Conf. Exhib.*, 2012, pp. 599–604.
[28] S. V. R. Chittamuru, I. Thakkar, and S. Pasricha, "Analyzing volt-age bias and temperature induced aging effects in photonic interconnects for manycore computing," in *Proc. ACM/IEEE Int. Workshop Syst. Level Interconnect Prediction*, Jun. 2017.
[29] B. Raghunathan, et al., "Cherry-picking: Exploiting process variations in dark-silicon homogeneous chip multi-processors," in *Proc. Des. Autom. Test Europe Conf. Exhib.*, 2013, pp. 39–44.
[30] N. Kapadia and S. Pasricha, "VARSHA: Variation and reliability-aware application scheduling with adaptive parallelism in the dark-silicon era," in *Proc. Des. Autom. Test Europe Conf. Exhib.*, 2015, pp. 1060–1065.
[31] B. Guha, et al., "Cmos-compatible athermal silicon microring resonators," in *Opt. Express*, vol. 18, no. 4, pp. 3487–3493, 2010.
[32] S. V. R. Chittamuru, et al., "SWIFTNoC: A reconfigurable silicon-photonic network with multicast enabled channel sharing for multicore architectures," *ACM J. Emerging Technol. Comput. Syst.*, vol. 13, no. 4, 2017, Art. no. 58.
[33] C. T. DeRose, et al., "Silicon microring modulator with integrated heater and temperature sensor for thermal control," *CLEO/QELS: Laser Sci. Photonic Appl.*, 2010, pp. 1–2.
[34] M. Georgas, et al., "Addressing link-level design tradeoffs for integrated photonic interconnects," in *Proc. IEEE Custom Integrated Circuits Conf.*, 2011, pp. 1–8.
[35] M. V. Beigi and G. Memik, "Therma: Thermal-aware run-time thread migration for nanophotonic interconnects," in *Proc. Int. Symp. Low Power Electron. Des.*, 2016, pp. 230–235.
[36] M. Mohamed, et al., "Reliability-aware design flow for silicon photonics on-chip interconnect," *IEEE Trans. Very Large Scale Integration Syst.*, vol. 22, no. 8, pp. 1763–1776, Aug. 2014.
[37] Y. Xu, J. Yang, and R. Melhem, "Tolerating process variations in nanophotonic on-chip networks," in *Proc. 39th Annu. Int. Symp. Comput. Archit.*, 2012, pp. 142–152.
[38] M. Mohamed, et al., "Power-efficient variation-aware photonic on-chip network management", in *Proc. ACM/IEEE Int. Symp. Low-Power Electron. Des.*, 2010, pp. 31–36.
[39] R. Wu, et al., "Variation-aware adaptive tuning for nanophotonic interconnects," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, 2015, pp. 487–493.
[40] K. Padmaraju and K. Bergman, "Resolving the thermal challenges for silicon microring resonator devices," *Nanophotonics*, vol. 3, no. 4-5, pp. 269–281, 2013.
[41] M. A. Popovic, "Theory and design of high-index-contrast microphotonic circuits," Ph.D. dissertation, MIT, Cambridge, 2008.
[42] W. Bogaerts, et al., "Silicon microring resonators," *Laser Photon. Rev.*, vol. 6, no. 1, pp. 47–73, 2012.
[43] P. P. Absil, et al., "Silicon photonics integrated circuits: a manufacturing platform for high density, low power optical i/os," *Opt. Express*, vol. 23, no. 7, pp. 9369–9378, 2015.

[44] T. E. Carlson, et al., "Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulations," in *Proc. Int. Conf. High Perform. Comput. Netw. Storage Anal.*, 2011, pp. 1–12.
[45] S. C. Woo, et al., "The SPLASH-2 programs: Characterization and methodological considerations," in *Proc. 22nd Annu. Int. Symp. Comput. Archit.*, 1995, pp. 24–36.
[46] C. Bienia, et al., "The PARSEC benchmark suite: Characterization and architectural implications," in *Proc. Int. Conf. Parallel Archit. Compilation Techn.*, 2008, pp. 72–81.
[47] S. Li, et al., "McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures," in *Proc. 42nd Annu. IEEE/ACM Int. Symp. Microarchitecture*, 2009, pp. 469–480.
[48] 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.
[49] A. Sridhar, et al., "3D-ICE: Fast compact transient thermal modeling for 3d ICs with inter-tier liquid cooling," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, 2010, pp. 463–470.
[50] Crystal Quartz. (2016). [Online]. Available: http://www.mt-berlin.com/frames_cryst/crystals_frameset1.htm
[51] S. R. Sarangi, et al., "Varius: A model of process variation and resulting timing errors for microarchitects," *IEEE Trans. Semicond. Manuf.*, vol. 21, no. 1, Feb. 2008, pp. 3–13.
[52] D. Harris, et al., "Support vector regression machines," in *Proc. 9th Int. Conf. Neural Inf. Process. Syst.*, 1996, pp. 155–161.
[53] P. Grani and S. Bartolini, "Design options for optical ring interconnect in future client devices," *ACM J. Emerging Technol. in Comput. Syst.*, vol. 10, no. 4, 2014, Art. no. 30.
[54] Lumerical Solutions. (2017). [Online]. Available: http://www.lumerical.com/tcad-products/mode/
[55] S. V. R. Chittamuru, I. Thakkar, S. Pasricha, "PICO: Mitigating heterodyne crosstalk due to process variations and intermodulation effects in photonic NoCs," in *Proc. 53nd ACM/EDAC/IEEE Des. Autom. Conf.*, 2016, pp. 1–6.
[56] CACTI 6.5. (2017). [Online]. Available: http://www.hpl.hp.com/research/cacti/
[57] D. Dang, S. V. R. Chittamuru, R. Mahapatra, and S. Pasricha, "Islands of heaters: A novel thermal management framework for photonic NoCs," in *Proc. 22nd Asia South Pacific Des. Autom. Conf.*, 2017, pp. 306–311.
[58] J. E. Cunningham, et al., "Highly-efficient thermally-tuned resonant optical filters," *Opt. Express*, vol. 18, no. 18, pp. 19055–19063, 2010.
[59] K. Padmaraju, et al., "Wavelength locking and thermally stabilizing microring resonators using dithering signals," in *J. Lightw. Technol.*, vol. 2, no. 3, pp. 505–512, Feb. 2014.

Sai Vineel Reddy Chittamuru (S'14) received the BTech degree in electrical engineering from the Indian Institute of Technology Kharagpur (IIT-KGP), in 2011, and the PhD degree in electrical and computer engineering, from Colorado State University, Fort Collins, CO, USA, in 2018. His research interests include memory system design, embedded system, systems-on-chip, and optical networks-on-chip. He is the recipient of the best paper awards from IEEE/ACM SLIP 2016 and ACM GLSVLSI 2015 conferences, and a best paper nomination from the IEEE ISQED 2016 conference for his research contributions. He is a student member of the IEEE.

Ishan Thakkar (S'14) received the MS degree in electrical engineering from Colorado State University, Fort Collins, CO, USA. He is currently working toward the the PhD degree in electrical engineering from Colorado State University, Fort Collins, CO, USA. His research interests include DRAM architectures, non-volatile memories, high-speed optical interfaces, and optical networks-on-chip. He is the recipient of the best paper award from IEEE/ACM SLIP 2016 conference and a best paper nomination from the IEEE ISQED 2016 conference for his research contributions. He is a student member of the IEEE.

Sudeep Pasricha (M'02-SM'13) received the PhD degree in computer science from the University of California, Irvine, in 2008. He is currently a Monfort and Rockwell-Anderson Professor of Electrical and Computer Engineering at Colorado State University. His research interests include energy efficiency and fault tolerance for multicore computing. He is on the editorial board of various journals such as the *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *IEEE Transactions on Multi-Scale Computing Systems*, *TECS*, etc. He currently is or has been on the organizing committee and TPC of various conferences such as DAC, DATE, NOCS, ESWEEK, VLSID, etc. He has received Best Paper Awards at SLIP'16, GLSVLSI'15, AICCSA'11, ISQED'10, and ASPDAC'06.