Longevity Framework: Leveraging Online Integrated Aging-Aware Hierarchical Mapping and VF-Selection for Lifetime Reliability Optimization in Manycore Processors

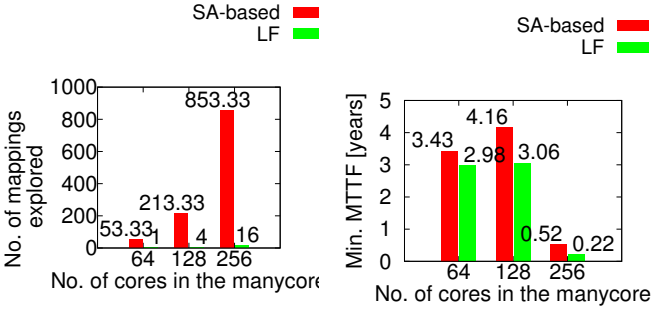
Vijeta Rathore, Vivek Chaturvedi, *Member, IEEE* Amit K. Singh, *Member, IEEE*
Thambipillai Srikanthan, *Senior Member, IEEE* and Muhammad Shafique *Senior Member, IEEE*

Abstract—Rapid device aging in the nano era threatens system lifetime reliability, posing a major intrinsic threat to system functionality. Traditional techniques to overcome the aging-induced device slowdown, such as guardbanding are static and incur performance, power, and area penalties. In a manycore processor, the system-level design abstraction offers dynamic opportunities through the control of task-to-core mappings and per-core operation frequency towards more balanced core aging profile across the chip, optimizing the system lifetime reliability while meeting the application performance requirements. This paper presents *Longevity Framework (LF)* that leverages online integrated aging-aware hierarchical mapping and VF-selection for lifetime reliability optimization in manycore processors. The mapping exploration is hierarchical to achieve scalability. The VF-selection builds on the trade-offs involved between power, performance, and aging as the VF is scaled while leveraging the per-core DVFS capabilities. The methodology takes the chip-wide process variation into account. Extensive experimentation, comparing the proposed approach with two state-of-the-art methods, for 64-core and 256-core systems running applications from PARSEC and SPLASH-2 benchmark suites, show an improvement of up to 3.2 years in the system lifetime reliability and 4× improvement in the average core health.

Index Terms—Lifetime reliability, aging, DVFS, manycore systems, process variation, optimization.

1 Introduction

Advancements in process technology led to an exponential growth of on-chip computation resources. However, there are several design challenges associated, such as high power density, rising chip temperature, and reduced system lifetime reliability. Device aging mechanisms, such as negative bias temperature instability (NBTI), time-dependent dielectric breakdown (TDDB), electromigration (EM), and hot carrier injection (HCI) negatively impact system lifetime reliability [1], [2], [3]. In the deep sub-micron region, the aging mechanisms accelerate due to elevated chip temperature and shrunk dimensions, leading to rapid deterioration of device characteristics including transistor delay degradation and increased metal interconnect resistance. Moreover, manufacturing-induced within-die process variation (PV) further exacerbates the lifetime of the cores [4]. Another factor hampering system performance is dark silicon—as much as 30% chip is envisaged to be dark at 8 nm [5]. The dark silicon constraint however also provides opportunities to mitigate aging and improve lifetime in manycore processors [6], [7]. These factors, consequently, make lifetime reliability a major design concern [8].



(a) Number of mappings explored (normalized with respect to Longevity Framework, for 64-core case).

(b) Minimum MTTF among all the cores for obtained mapping.

Fig. 1: Comparison of SA-based approach and Longevity Framework.

Traditionally, designers use one-time worst-case guardbands to safeguard against future degradation in the form of supply voltage increase, reduced operating frequency, or device oversizing [9], which incur performance, power, or area penalties. Moreover, the process variability (PV) problem makes it inefficient to use the same guardband chip-wide. Overcoming the aforementioned limitations of guardbanding, the system level design abstraction, in case of manycore processors, provides opportunities of affecting the way cores age through task-to-core mappings and per-

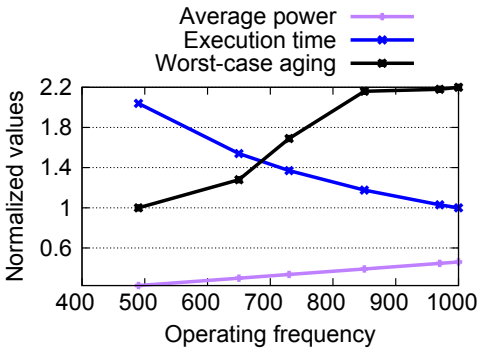
- *V. Rathore and T. Srikanthan are with the School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore Email: vijeta001@ntu.edu.sg, astsrikan@ntu.edu.sg*
- *V. Chaturvedi is with the Department of Computer Science and Engineering, Indian Institute of Technology (IIT) Palakkad, India Email: vivek@iitpkd.ac.in*
- *A. K. Singh is with the School of Computer Science and Electronic Engineering, University of Essex, UK Email: a.k.singh@essex.ac.uk*
- *M. Shafique is with the Institute of Computer Engineering, Technische Universität Wien (TU Wien), Austria Email: muhammad.shafique@tuwien.ac.at*

core VF control. Several mapping-based approaches are proposed in [7], [10], [11], [12], [13] for lifetime/aging optimization. However, they have limited scalability owing the compute intensive approach of mapping determination.

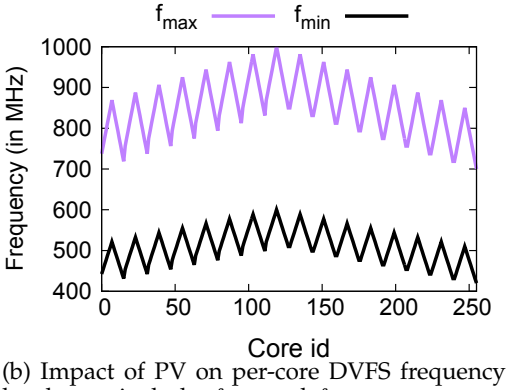
Mintarno et al. [9] use DVFS to alleviate the impact of aging towards maximizing lifetime per unit of power consumed. DVFS is an power management hardware technique that involves scaling of supply voltage to operate at a lower frequency, enabling reduced power consumption at the cost of compromised performance. Quite a few works have delved into using DVFS for lifetime reliability optimization such as in [10], [14], [15], [16]. Das et al. [10] propose genetic algorithm-based combined exploration of mapping and VF-selection to optimize both aging and soft-error susceptibility. Haghbayan et al. [14] propose a combination of power-gating and DVFS along with aging-aware mapping to co-optimize lifetime reliability and performance. VARSHA [15] optimizes system performance and soft-error reliability while meeting the power budget and performance constraints using combined scheduling and DVFS. Contrastingly, Bosaglu et al. [16] propose NBTI-aware control of the supply voltage for lowering energy consumption. These works have certain limitations: Das et al. [10] do not consider process variation, Haghbayan et al. [14] and VARSHA [15] are both restricted to rectangular mapping regions and do not leverage dark cores for thermal mitigation. Longevity Framework (LF) overcomes these challenges by formulating a scalable, non-rectangular region-based mapping cum VF-selection solution that is process variation aware and utilizes dark silicon for thermal mitigation towards lifetime reliability optimization.

Electromigration (EM) is one of the prominent aging mechanisms leading to lifetime reliability reduction for technology nodes smaller than 45 nm [17]. Some of the earliest works on threats to lifetime reliability have also identified EM as a significant aging mechanism as the technology scaling continues [1]. EM affects the lifetime reliability of the metal interconnect, failure of which directly disturbs the system functionality. The proposed method is, however, orthogonal to the aging mechanism considered. Other aging models can be considered in conjunction or alternately. We did not considered BTI in this work as we wanted to study with an aging mechanism affecting the interconnect, specifically the power delivery network. Clock and signal nets do not show significant aging due to the bi-directional current they carry, leading to self-healing [18].

Motivational Analysis and Target Research Problem: To motivate scalable mapping solutions, we analyzed the computational complexity of simulated annealing (SA)-based lifetime optimization mapping-based approach of [13], and compared it with that of our proposed approach, LF. The SA-based approach [13] is the most scalable among the mapping-based related work, including [13], [19], [20], being heuristic-based, and hence we chose to compare against it. Fig. 1(a) plots the size of the mapping space explored, and Fig. 1(b) depicts the achieved lifetime reliability for three systems with 64, 128 and 256 cores, respectively. The workloads for the three cases comprised of 4, 6, and 9 benchmarks from SPLASH-2, respectively. As shown in Fig. 1(a), SA-based approach explored a much larger number of mappings than LF while improving the lifetime



(a) Power, performance and aging trade-offs as frequency varies.



(b) Impact of PV on per-core DVFS frequency levels, particularly, f_{min} and f_{max}.

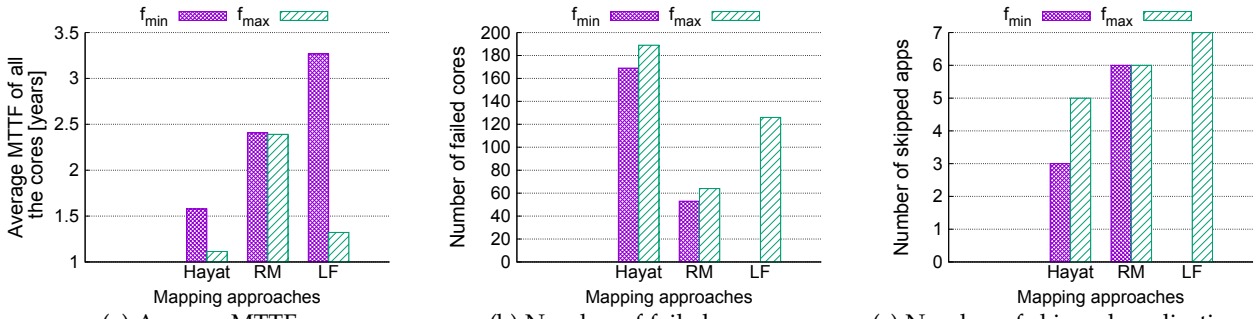
Fig. 2: Motivational analysis.

reliability by nearly a year. For instance, for a 128-core system, SA explored 52× more mappings (Fig. 1(a)) with a lifetime (MTTF) improvement of 1.1 years (Fig. 1(b)). The 52× larger mapping exploration would incur proportionate time and power overhead. We noted that for LF, the number of explored mappings is much smaller as the search space of mappings is greatly reduced (as explained later), while not heavily compromising the lifetime reliability. Hence, LF is more efficient a mapping approach, paving the case of scalable mapping solutions for the manycore systems.

Another system level handle is per-core DVFS [21]. It has been utilized in [22] and [23] in the context of $Atom^{TM}$ and $Power8^{TM}$ processors, respectively. The per-core DVFS capabilities offer interesting design choices due to the trade-offs among various design parameters including performance, power, and aging. In Fig.1, we demonstrate different trade-offs as the operating frequency is varied across several VF levels with an underlying mapping obtained from the aging-aware mapping approach, *Hayat* [7]. As seen here, increasing the frequency reduces the execution time but incurs higher power and causes more aging. The figure indicates that selecting the least frequency permissible by the performance requirement of the application can achieve maximal power and aging gains.

Fig. 2(b) shows the least (f_{min}) and highest (f_{max}) frequency levels permissible with the existing PV for the cores of a 256-core system. The values depicted are based on the model discussed in Sec. 3.2. PV impacts the VF levels of the cores, which affects the range of exploration of the above-mentioned trade-offs.

To find the extent to which DVFS capabilities can bring lifetime enhancements, we run applications as per ob-



(a) Average MTTF. (b) Number of failed cores. (c) Number of skipped applications.

Fig. 3: A comparison of the impact of operating cores at f_{max} versus f_{min} on aging-related metrics, including average MTTF and number of failed cores, and the number of applications skipped from being serviced, under mappings obtained by different aging-aware approaches.

tained mappings from aging-aware approaches including HiMap [24], Hayat [7] and RM [12], at two frequency extremities, namely f_{min} and f_{max}. We compare the average mean time to failure (MTTF) of all the cores, the number of failed cores at the end of a 10 years simulation, and the number of applications skipped at that time due to the approach not being able to map while meeting their performance requirement.

Fig. 3(a) shows that the lifetime reliability is enhanced at f_{min}, compared to at f_{max}. The improvement is significant for mappings from Hayat and HiMap, while it is only a slight improvement for mappings obtained from RM due to the narrow slack afforded by the rectangular mapping generated by RM. In alignment with it, the number of failed cores at the end of the simulation was consistently higher for f_{max}, than f_{min}; HiMap had the greatest margin. Moreover, the number of skipped applications for f_{min} was also fewer. It proves that opting for f_{min} in case the performance requirements permit, is beneficial for lifetime optimization.

This paper aims to obtain the optimal task-to-core mappings and voltage-frequency (VF) pair assignment for each core running a task. The challenges involved are incorporating the power, performance, and aging trade-offs, and meeting the constraints such as performance requirement of the applications, chip's thermal safe temperature, and power budget.

Our Novel Contributions: To address the above-discussed challenges, we make the following novel contributions.

a) We present VF-selection and hierarchical mapping as a synergistic solution to improve lifetime reliability of the manycore processors. Towards this, this journal submission is partly built on top of our recent work HiMap [24].
b) The proposed methodology obtains the voltage frequency assignment for each core while meeting the application performance requirement.
c) The mapping approach leverages dark cores to mitigate temperature and finds block-based mappings that ensure uniform mapping of the cores.
d) It accounts for the impact of PV and aging on the voltage frequency levels.
e) Results obtained from extensive experimentation with multiple workloads and systems show the effectiveness of the proposed performance constraint-aware framework integrating task mapping and per-core DVFS for lifetime reliability optimization of manycore systems.

Paper Organization: The rest of the paper is organized as follows. Sec. 2 describes the related work. Sec. 3 presents the system model and the preliminaries. Sec. 4 defines the problem statement. Sec. 5 discusses the proposed solution. Sec. 6 explains the experimentation and the results. Sec. 7 concludes.

2 Related Work

In the literature, there are several system-level mapping-based approaches to improve lifetime reliability of multi/many-core systems [12], [13], [19], [20]. However, most do not consider process variation, and dark silicon constraints and are computation intensive making them inadequate to match the scale of the manycore systems. For instance, Das et al. [19] devised a convex optimization-based mapping technique and Wang et al. [20] used sequential quadratic programming to find the optimal processor speeds while meeting the aggregate frequency constraints. Also, Huang et al. [13] have proposed a simulated annealing (SA) based mapping approach to improve lifetime reliability of multi-core systems.

Haghbayan et al. [12] proposed a Reliability-aware Mapping (RM) approach for dark silicon manycore systems; however it does not explicitly consider PV. Gnad et al. [7] have proposed PV- and aging-aware mapping approach for dark silicon manycore systems, namely, Hayat. Wang et al. [25] interspersed dark cores in the region of application mapping, for thermal mitigation and performance optimization, however, they do not form the area based on blocks of cores. Carvalho et al. [26] created rectangular regions of cores or *clusters* to map applications, while LF forms clusters by selecting blocks, which gives it the flexibility to choose cores from irregular locations as well.

DVFS is widely used for power/energy optimization [27], [28]. DVFS-based thermal management is also proposed [29], [30]. Temperature minimization, however, does not necessarily lead to aging/lifetime optimization since aging also depends on other parameters such as supply voltage, frequency, stress time, and process variation [11].

Use of per-core DVFS for lifetime enhancement is explored in [10], [16], [31]. Basoglu et al. [16] reduce dynamic energy consumption by performing a greedy-based mapping for lifetime maximization, and adjust the supply voltage of each core to a lower value of DVFS voltage level considering aging due to NBTI. Das et al. [10] address the twin problem of maximizing lifetime reliability and reliabil-

ity due to transient faults by affecting both mapping and DVFS, using a multi-objective genetic algorithm. As the design space is enormous, the initial phase is made to contain a certain percentage of the feasible solutions, finding which can be a challenge even for systems with tens of cores. Kim et al. [31] proposed a learning-based DVFS and dark core placement solution for dark silicon manycore to minimize energy consumption while meeting reliability, thermal and performance constraints. They considered aging of power grid networks due to EM. However, this Q-learning-based approach lacks scalability as the policy table size grows exponentially with increasing number of cores. [32], [33] investigate interesting power-reliability trade-off for different VF levels.

Distinctions of proposed LF over State-of-the-Art: In summary our work is different from the above state of the art in the following respects:

a) In addition to temperature, we also incorporate impact of PV on aging, unlike other approaches such as in [31].
b) We reduce the design space to be explored significantly, as shown in Fig. 1(a), making it a scalable solution.
c) We do not restrict the mapping regions to be rectangular, as in [12], [26], and are able to flexibly select noncontiguous regions, possibly resulting in favorable mappings.

3 System Model and Preliminaries

3.1 Manycore Processor Architecture

The manycore system is an $L_X \times L_Y$ grid of tiles. A tile consists of a core, a memory (private L1 and L2 caches) and a switch as shown in Fig. 5(a). The set of cores is denoted as $C = \{C_{i,j}, \forall i \epsilon [1, L_X], j \epsilon [1, L_Y]\}$. There is per-core DVFS capability [21]. The unused cores are power-gated, ensuring zero power for the sleeping cores. Each core has a thermal sensor as in [34]. The on-chip network is of the mesh topology.

3.2 Process Variation Model

Process variation impacts parameters such as frequency, metal width and interconnect resistance [35]. We use statistical process variation model presented in [36] that overlays a fine grid of spatially correlated Gaussian random variables, $p_{x,y}, (x \epsilon [1, P], y \epsilon [1, Q])$, on the chip. PV impacts the metal width and interconnect resistance [35]. The physical parameters are related to their nominal values, e.g. wire width (W), height (H) and power grid resistance (Res) at grid point (x, y) are given as:

$$W_{x,y} = \kappa_1 p_{x,y} \tag{1a}$$
$$H_{x,y} = \kappa_2 p_{x,y} \tag{1b}$$
$$Res_{x,y} = \gamma p_{x,y} \tag{1c}$$

where, κ_1, κ_2 and γ are technology specific constants.

The maximum frequency of a core ($f_{i,j}$) is also subject to PV and is given as:

$$f_{i,j} = \beta \min_{s,t \in S_{CP,i,j}} p_{s,t} \tag{2}$$

where, $S_{CP,i,j}$ is the set of grid points containing critical paths and β is a technology specific constant. The leakage and dynamic power are also affected by PV as described in [36].

Eq. 3a expresses the total power of core $C_{i,j}$ running thread τ_q of application A_p. Eqs. 3b and 3c are the dynamic power and leakage power formulations. The dynamic and temperature dependent leakage are affected by PV. The leakage power of the core is found by summing over all the grid points lying on the core.

$$P_{total,i,j} = P_{dyn,i,j} + P_{leak,i,j} \tag{3a}$$
$$P_{dyn,i,j} = \alpha'_{p,q} Cap_{i,j} V^2 f_{i,j} \tag{3b}$$
$$P_{leak,i,j} = \sum_{(u,v) \epsilon C_{i,j}} p^{leak}_{u,v} \times e^{V_{th} p_{u,v}/V_T} \tag{3c}$$

In Eq. 3b, $\alpha'_{p,q}$ is the switching activity of thread τ_q of application A_p, $Cap_{i,j}$ is the effective capacitance of the tile i, j, V is the supply voltage, $f_{i,j}$ is the frequency of core i, j, $V_T = KT_{i,j}/e$ ($T_{i,j}$ is core $C_{i,j}$'s temperature, e is the charge of an electron and K is Boltzmann constant).

3.3 DVFS Voltage and Frequency (VF) levels

The DVFS at each core can be exercised at one of the num_{dvfs} levels of VF. The set of VF levels for core C_i is represented as:

$$\{V_i, F_i\} = \{\{v_1, f_1\}, \{v_2, f_2\}, \ldots \{v_{num_{dvfs}}, f_{num_{dvfs}}\}\} \quad \forall\ i\ \epsilon\ \{1, 2, \ldots, N\} \tag{4}$$

All cores have the same set of available voltage levels given by:

$$\{v_1, v_2, \ldots, v_{num_{dvfs}}\}\ \forall i\ \epsilon\ \{1, 2, \ldots, N\} \tag{5}$$

For a given supply voltage, the operating frequency is a function of both aging and PV [37]. The nominal voltage frequency pairs are given as:

$$\{V, F_{nom}\} = \{\{v_1, f_{nom,1}\}, \{v_2, f_{nom,2}\}, \ldots, \{v_{num_{dvfs}}, f_{nom,num_{dvfs}}\}\} \tag{6}$$

At any moment t, each DVFS frequency level ($f_{i,t}$) is affected by the PV and aging. The frequency levels are thus proportionally related to their nominal value ($f_{nom,i}$) by the same factor as the maximum operation frequency ($f_{max,t}$), given by:

$$f_{i,t} = f_{nom,i} \times \frac{f_{max,t}}{f_{nom,max}} \tag{7}$$

3.4 Application Model

We model workload as periodic multi-threaded applications, $A = \{A_1, A_2, \ldots, A_M\}$ with periods given by $Q = \{Q_1, Q_2, \ldots, Q_M\}$. Application A_p has N_p threads, denoted by $\{\tau_{p,1}, \tau_{p,2}, \ldots, \tau_{i,N_p}\}$. Thread $\tau_{p,q}$ has a deadline given by $t_{d,p,q}$. The frequency requirement of thread $\tau_{p,q}$ is given by $f_{req,p,q}$.

Execution time ($t_{execution}$) is related to the instruction count (IC) and the average number of cycles per instruction (CPI) by the following equation:

$$t_{execution,p,q} = IC \cdot CPI \cdot T_{clk}\,, \tag{8}$$

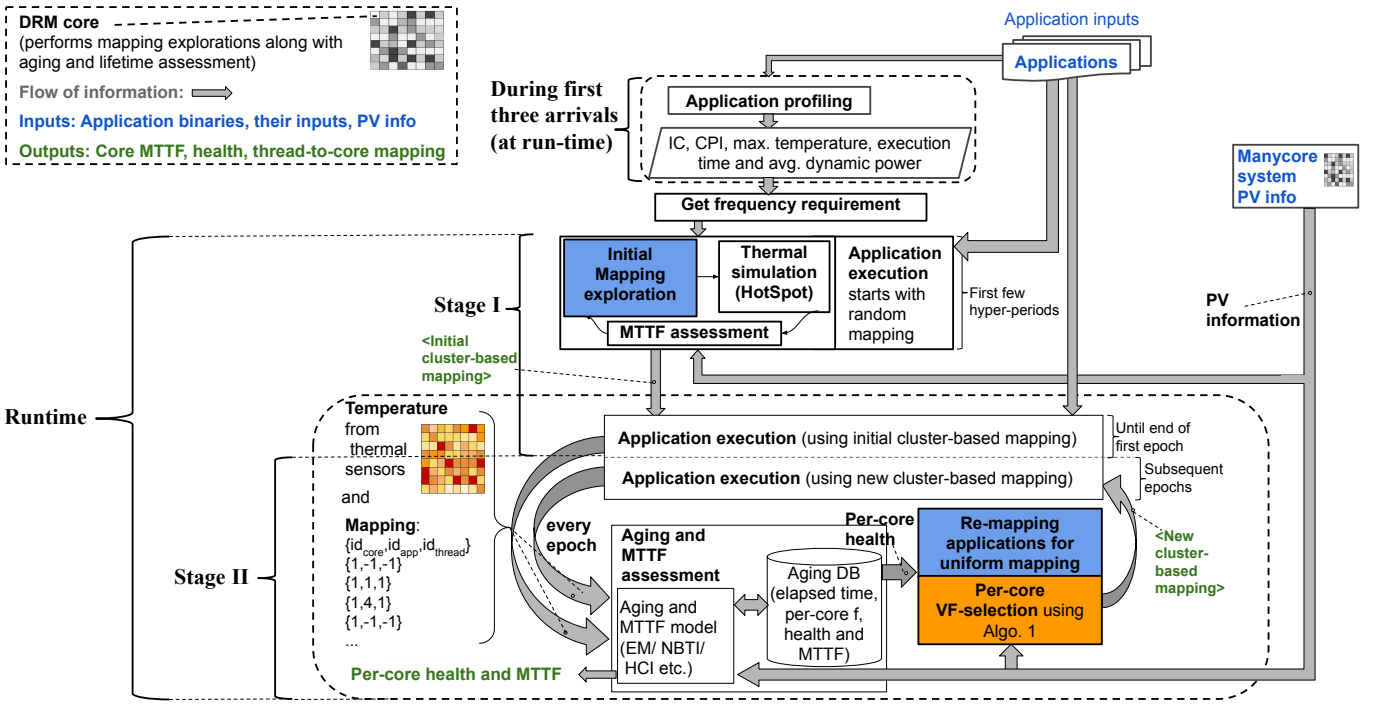


Fig. 4: Longevity Framework schematic involving one-time initial mapping exploration, and repetitive integrated application re-mapping and per-core VF-selection.

where, T_{clk} is the clock period. Substituting execution time with the deadline ($t_{d,p,q}$) and T_{clk} by $1/f_{req,p,q}$ gives frequency requirement as:

$$f_{req,p,q} = \frac{IC \cdot CPI}{t_{d,p,q}}. \tag{9}$$

Mapping of thread $\tau_{p,q}$ (of application A_p) to core $C_{i,j}$ is represented by the mapping function $m : \{\tau_{p,q}, p\epsilon[1,M], q\epsilon[1,N_p]\} \rightarrow \{C_{i,j}, i\epsilon[1,L_X], j\epsilon[1,L_Y]\}$.

3.5 Aging and Lifetime Reliability Assessment

Without the loss of generality, we consider the aging due to EM, since it is one of the major aging phenomena affecting lifetime reliability of the manycore system. However, the proposed approach can work for other aging mechanisms as well by assessing the aging and lifetime reliability of the considered aging phenomena. EM affects power grid networks leading to increase in resistance and larger voltage drop [38]. As a result of which, the maximum operation frequency reduces. Similar to [38], we consider a core as faulty if its supply voltage drops to below a threshold value of V_f.

As explained in detail by Huang et al. [38], nucleation time is taken for the stress in the wire to reach a critical value (σ_{crit}). σ_{crit} is the critical stress needed for the nucleation towards the formation of a void or hillock. The nucleation phase is followed by the growth phase during which the resistance increases.

The lifetime reliability, measured as mean time to failure (MTTF), of a core is formulated as [38]:

$$MTTF = (t_{growth,\Delta V = V_f}) + t_{nuc} \tag{10}$$

where, $t_{growth,\Delta V = V_f}$ is the duration for the worst-case voltage drop to become V_f, and t_{nuc} is the nucleation time. In Eq. 10, nucleation time (t_{nuc}) is added to the growth time (t_{growth}), since this time is spent before the growth phase. Nucleation time is approximated as given by Eq. 11:

$$t_{nuc} \approx \tau^* e^{\frac{E_V}{kT}} e^{-\frac{f\Omega}{kT}(\sigma_{Res}+\sigma)} ln\Big\{\frac{\sigma}{\sigma_{Res}+\sigma-\sigma_{crit}}\Big\} \tag{11}$$

where, $\tau^* = \frac{l^2}{D_0} \exp \frac{E_D}{kT} \frac{kT}{\Omega B}$ and $\sigma = \frac{eZ\rho l}{4\Omega} j$. The rest of the parameters are current density (j), activation energy of vacancy formation (E_V) and diffusion (E_D), ratio of volumes of vacancy and lattice atom f, residual stress (σ_{Res}), atomic volume (Ω), wire segment length l, resistivity of wire metal (ρ), and effective charge of the migrating atoms (eZ).

Eq. 12 approximates the growth of resistance:

$$\Delta Res(t) = v(t - t_{nuc})\Big[\frac{\rho_{Ta}}{h_{Ta}(2H+W)} - \frac{\rho_{Cu}}{HW}\Big] \tag{12}$$

The symbols involved are resistivity of barrier material tantalum (ρ_{Ta}) and line metal copper (ρ_{Cu}), barrier height (h_{Ta}), and drift velocity (v). As the core temperature changes with different mappings over time, we track the aging phase (nucleation or growth) and state (change in resistance) in an aging database. The aging assessment uses this information to determine core aging and MTTF for the ensuing epoch.

MTTF is an instantaneous value which reflects the expected lifetime under existing conditions. Aging, on the other hand, represents cumulative degradation over a period of time. MTTF can increase over time depending on the current temperature and other aging-affecting parameters. For instance, if a task leading to high temperature is first assigned to a core, followed by a task generating lower temperature, then one can observe that the MTTF value first

decreases and then increases. It is due to the fact that under latter conditions the aging rate is slowed down, allowing the core a longer time before it fails. However, the overall aging of the system will continue to degrade during both tasks, at different rates.

We consider the shortest MTTF among all the cores as the MTTF of the system, as also considered in [19], [20], i.e.:

$$MTTF_{sys} = \min_{\forall i \epsilon [1,L_X], j \epsilon [1,L_Y]} \{MTTF_{i,j}\} \quad (13)$$

This definition of system MTTF considers a system to fail when any of the core fails. Thus, system MTTF indicates the expected time to first core failure.

Average MTTF of the system which is given as:

$$MTTF_{avg} = \sum_{\forall i \epsilon N} MTTF_i / N, \quad (14)$$

where, N is the number of cores in the system. Average MTTF gives the expected value of core MTTF.

It is possible to combine aging due to other wearout phenomena such as NBTI and HCI with that due to EM. Unlike EM, BTI and HCI deteriorate the transistor threshold voltage (V_{th}). The aging caused by EM can be combined with that due to BTI and HCI by incorporating the impact of the different aging phenomena on the supply voltage and threshold voltage. The transistor delay depends on the supply voltage and threshold voltage as follows:

$$delay = \frac{K \cdot V_{dd}}{(V_{dd} - V_{th})^2}, \quad (15)$$

where, K is a technology specific constant.

Let a core fail when its frequency reaches a certain value denoted as $failure\text{-}frequency$. Given that the maximum frequency of core $C_{i,j}$ is given by Eq. 2. MTTF can be calculated by determining the time taken for frequency to reach the failure-frequency. It can be solved iteratively, starting with a small time-step, incrementing it each iteration and checking if the frequency has reached the failure-frequency.

4 Problem Formulation

Optimization Goal: The objective of this work is to optimize system lifetime reliability, given by Eq. 16, by finding the task-to-core mappings and ascertaining per-core VF levels from amongst the DVFS levels available for the core, while satisfying the constraints of performance, maximum permissible temperature, and thermal design power (TDP).

$$\max \min_{\forall i \epsilon [1,L_X], j \epsilon [1,L_Y]} MTTF_{i,j} \quad (16)$$

Constraints: The thermal constraint is given by Eq. 17a, and TDP is given by Eq. 17b. The performance requirement constraint of an application A_p's ($p \epsilon [1, M]$) thread $\tau_{p,q}$ ($q \epsilon [1, N_p]$), states that the assigned frequency should meet the task's performance requirement (Eq. 17c).

$$T_{i,j} \leq T_{safe} \forall i \epsilon [1, L_X], j \epsilon [1, L_Y] \quad (17a)$$

$$\sum_{\forall i \epsilon [1,L_X], j \epsilon [1,L_Y]} P_{total,i,j} \leq TDP \quad (17b)$$

$$f_{i,j} \geq f_{req,p,q} \quad (17c)$$

where, $T_{i,j}$ is the steady-state temperature of core $C_{i,j}$, and $f_{i,j}$ is core $C_{i,j}$'s frequency of operation.

5 Proposed Longevity Framework (LF)

Fig. 4 shows the steps involved in the proposed online lifetime reliability optimizing framework, namely Longevity Framework. LF integrates our previously proposed hierarchical mapping approach, HiMap, and per-core VF-selection.

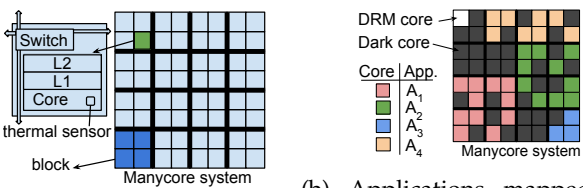
We first briefly introduce HiMap since it provides a foundation for this work. Followed by that we discuss the per-core VF selection proposed in this paper.

5.1 HiMap: Hierarchical Mapping Approach for Lifetime Reliability Optimization

The concepts central to the proposed approach are:

- **Blocks**: To manage the complexity and scale of mapping exploration, we group cores into *blocks* of equal size. For example, Fig. 5(a) shows a 64-core system with 2×2 sized blocks.[1]
- **Clusters**: A cluster is a region of cores, formed by selecting some blocks, to which threads of an application are mapped. A cluster can have some dark cores as well, e.g., Fig. 5(b) shows 4 clusters with 4 applications mapped.
- **Inclusion of dark cores in the cluster:** Dark cores are included in each cluster. The steps to obtain the number of dark cores are explained in [24].
- **Hyperperiod:** It is the least common multiple of all the application periods.
- **Epoch:** It is the period for the aging assessment and mapping intervention.
- **Health of a core:** A core's health is inversely related to the aging state. For EM we quantify it as the increment in the power grid resistance (ΔRes). We formulate health of a core as $Health = 1/(1 + \Delta Res)$.

1. The best block size is found as described ahead in this section.



(a) Cores grouped in blocks. (b) Applications mapped to clusters.

Fig. 5: Illustration of blocks and application clusters.

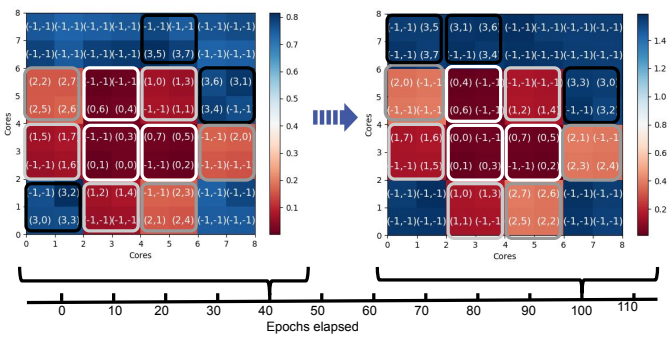


Fig. 6: Aging heatmaps illustrating mappings at epoch number 40 and 100, obtained through the simulation of 4 PARSEC applications running on a 64-core system affected by PV. The numbers in paranthesis indicate the application id and the task id, respectively. (-1,-1) means that it is a dark core.

In order to account for the memory and network interference from other workloads on application metrics such as execution time, HiMap performs profiling during runtime. The profiling is done during the first three[2] executions of the applications. Out of those executions, we use the average case (with respect to execution time) as considering the worst-case might be too pessimistic. HiMap can thus handle newly introduced applications by profiling them at runtime.

The profiled data includes the application IC and CPI, along with the maximum temperature, execution time and average dynamic power. The data related to application threads is associated with the application identifier and thread identifier by tagging it with the tuple $\{app\_id, thread\_id\}$. Instead of logging values at each profiling tick, it logs average or steady state values. Out of the profiled data some are per-thread such as average instruction per cycle (IPC) and instruction count (IC). Other data is stored per application, including average total power, maximum temperature among all the cores, and execution time. The memory required to store the profiled data is very small. The estimated storage needed for an application with 16 threads is 140 B (64 B for IC, 64 B for CPI, and 4 B each for average power, execution time and maximum temperature.). Estimated storage requirement for 10 applications is thus only 1.36 kB.

Data is associated with its application and thread (where applicable) by tagging it with the corresponding application identifier (app_id) and thread identifier ($thread\_id$). A thread of an application is uniquely identified by $\{app\_id, thread\_id\}$.

Using the IPC and IC obtained from profiling, HiMap determines per-thread frequency requirement by using Eq. 9. HiMap determines task mapping every epoch, at point of time estimated to be the end of the hyperperiod occurring at the epoch end. It finds mappings at runtime in parallel to the application execution on a dedicated core called dynamic lifetime reliability manager or *DRM core*. Post the profiling, the operating system (OS) performs the thread-to-core mapping as per the mapping generated by HiMap task mapper. As the workload is periodic, it is required to map the same set of threads each time. Within (between) epochs, it follows the mapping obtained at the beginning of the epoch.

The first design point of HiMap is the order in which it maps the given applications. Its second design point is that it intersperses sleeping cores in a block of cores assigned to an application for temperature lowering and lifetime reliability gain. The third design point is assignment of threads to cores from the assigned blocks of cores, which is done during Stage I using simulated annealing-based mapping exploration, and in Stage II using a heuristic of assigning threads with higher average power to faster cores.

HiMap differs from the state-of-the-art aging-aware mapping methods such as Hayat [7] and RM [12]. HiMap proceeds mapping applications in a specific order which is the most favorable order, in terms of enhancing sysytem lifetime reliability, obtained from the initial mapping exploration, i.e., mapping higher power applications to faster blocks.

The mapping exploration consists of two stages: initial cluster-based mapping exploration (*stage I*) and finding an alternate cluster-based mapping to achieve uniform aging across the cores (*stage II*). In stage I, DRM core finds an initial cluster-based mapping (shown in a blue-colored box in Fig. 4) to maximize $MTTF_{sys}$ while taking several hyperperiods. Concurrently, applications execute on cores as per a random mapping. After the mapping exploration, the applications run as per the obtained mapping until the end of the first epoch. After that, stage II continues in which DRM finds a new mapping every epoch (also shown in a blue-colored box in Fig. 4). Notably, both Stage I and II use the profiled application average dynamic power and assign higher power applications to faster blocks of cores. Stage II also determines the most favorable VF level meeting the performance requirement of the mapped task as explained in Sec. 5.2 (shown in an orange-colored box in Fig. 4). As core-level aging profile varies over the epoch, the mapping step of stage II identifies the set of healthy cores and maps workload to these to ensure uniform aging across the cores. Fig. 6 shows the mappings superpositioned on the aging heatmaps at epoch number 40 and epoch 100 obtained through the simulation of 4 PARSEC applications on a 64-core system. The blocks occupied by the applications are marked color-wise. As seen here, applications get migrated to different blocks to keep the overall aging profile uniform so as to achieve maximization of the system lifetime reliability. Our paper [24] explains both the stages in detail.

Core Aging and MTTF assessment: To evaluate a mapping in terms of lifetime reliability, the DRM core performs per-core aging and MTTF assessment, for the given PV, core temperatures and existing aging (if any). For stage I, it returns per-core MTTF corresponding to the explored mapping. During stage II, it assesses per-core aging for the duration of an epoch (as described in Sec. 3 for EM) and maintains an aging database with per-core aging, frequency, and health.

5.2 Per-core VF-Selection for Lifetime Reliability Optimization

We derive from the motivational analysis that the least frequency level meeting the performance requirement of the task being mapped to a core is the most beneficial in terms of minimizing aging and hence maximizing lifetime reliability. Therefore, the per-core VF level selection is a greedy frequency selection, as described in Algo. 1.

Algo. 1 first initializes each core frequency as zero, which corresponds to a sleeping core (ln. 1 to ln. 3). Next, for all the tasks of each application (ln. 4), it gets the core it is mapped to (ln 5). For that task, it begins from the lowest frequency level available at the core (ln. 6), and checks if it meets the task's performance requirement (ln. 7). If the frequency requirement is met that frequency level is saved (ln. 8), otherwise it moves to the next higher frequency and repeats the process (ln. 6 to ln. 13). The core frequency is set as the saved frequency level (ln. 14).

Lifetime reliability gain obtained through the per-core VF-scaling depends on the mapping approach. It depends

2. Three executions were observed to be sufficient to overcome the cold start of the cache.

on the exection slack offered by the mapping: a mapping approach that offers greater execution slack will result in a lower frequency level, leading to lower temperature and a higher lifetime reliability.

Algorithm 1: Greedy per-core frequency selection.

Input: A, F_{req}, m, V, F
Result: $F_{running}$
for $\forall f \epsilon F$ **do**
  $f = 0$
end for
for All tasks $\tau_{p,q}$ of the application set A **do**
  Get the core C_j it is mapped to i.e. $m(\tau_{p,q})$
  for $\forall f \epsilon F$, initializing $f = f_{min}$ **do**
    if $f_{req,p,q} < f$ **then**
      $f_{set} = f$
      break
    else
      $f = f_{next\ higher}$
    end if
  end for
  $f_{running,j} = f_{set}$
end for

5.3 Intervention overhead and scalability

As the workload is periodic in nature, the mapping decisions are made ahead of time on a separate DRM core, so that there is no delay owing the mapping and VF determination. Moreover, the intervention is done once an epoch, which is of the order of hours or months. Hence, during both the stages, there is no performance penalty.

During stage I, when the initial cluster-based mapping is found the scheduler maps threads as per the obtained mapping from the next hyperperiod onwards, for the rest of the epoch. For a number of block dimensions explored ($num\text{-}block\text{-}dim\text{-}options$), stage I has a runtime complexity of $O(num\text{-}block\text{-}dim\text{-}options * M * Num_{rounds} * Num_{steps})$, where, M is the number of applications, and Num_{rounds} and Num_{steps} are, respectively, the number of rounds and steps in simulated annealing. Thus, stage I is scalable w.r.t. number of cores and applications.

The runtime complexity of stage II is $O(M.k.logk)$, where M is the number of applications, and k is the maximum number of threads among the applications. Thus, stage II is scalable to the number of applications and the size of the manycore system. Stage II takes place on DRM core to get the other mapping just before the end of the epoch. The VF level selection involves parsing the DVFS frequency levels to find the least frequency meeting the performance requirement, with a linear complexity of $O(num_{dvfs})$. The scheduler maps threads to cores as per the obtained mapping for the length of an entire epoch, starting with the time stipulated for the next hyper-period. Since runtime interferences could lead to application execution not finishing in the estimated time, the unfinished applications need to be migrated. A performance overhead is associated with the context switching of the unfinished application threads including their suspension and restart, application mapping, data migration, and cache flushing.

6 Experimental Setup and Results

6.1 Experimental Setup

Our experimental setup is as shown in Fig. 7. It consists of manycore simulator Snipersim [39], which is interfaced with power simulator McPAT [40]. The power output from it is fed to thermal simulator HotSpot [41]. After that, steady-state temperature output is passed to an aging and MTTF assessment unit. The aging and MTTF values for each core are sent to the dynamic resource manager (DRM) module, which interacts with the manycore simulator to control the mapping and per-core VF level. We updated the leakage model in HotSpot with a PV-aware adaptation of temperature dependent leakage model from [42]. The technology node is 22 nm. Tile size is 0.7 mm $\times$ 0.8 mm and comprises of Nehalem core, L1 cache (256 kB), and L2 cache (512 kB). The nominal frequency is 1GHz. In the architecture we have used, there is a memory controller for each core, with a three-channel connection to main memory and per-controller bandwidth of 7.6 GB/s. There is memory interleaving of 4 banks.

It is noteworthy that we could not measure the context switching overhead since manycore system simulation platform, Snipersim, simulates user-space only [39] and does not simulate the operating system (OS) and the context switch managed by it. Typically, context switch requires 5-7 μsec but since in case of LF it may occur only once an epoch, which is of the order of weeks or months, it does not hamper the average performance significantly.

We conducted experiments with a medium-sized system of 64 cores and a larger system with 256 cores. Similar to state-of-the-art works [7], [43], [44], we consider workload comprising of a number of applications. For both the systems the workload comprised of a mix of applications from SPLASH-2 and PARSEC benchmark suites—while SPLASH-2 is a classic benchmark suite, PARSEC contains emerging applications. Since aging is a long-term phenomena, and applications do not run long enough to cause noticeable aging, we consider periodic workloads. In order to capture the long-term effect of aging, we have used accelerated aging simulation, similar to Hayat [7], to obtain the effective aging due to the multiple executions of the periodic workload during an intervention interval i.e. epoch, which is set according to the experimental conditions and is typically of the order of weeks or months. The workload for 64-core

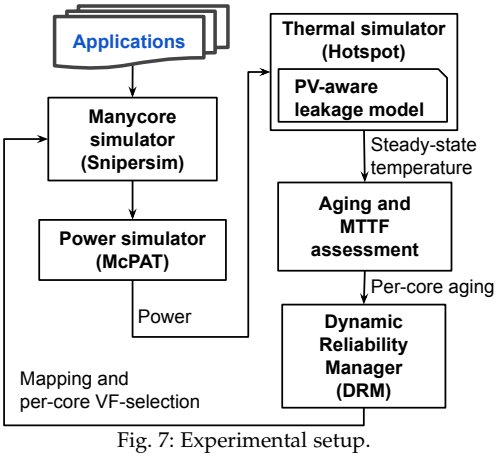


Fig. 7: Experimental setup.

system consisted of two applications each from PARSEC and SPLASH-2, and that for the 256-core system comprised of five applications from SPLASH-2 and 4 applications from PARSEC.

We chose a mix of applications from SPLASH-2 and PARSEC benchmark suites so that the power budget can be met, and also the IPC of the applications varies over a wide range—particularly from under 0.5 (low) to above 1.0 (high). In Table I (workload used for 64-core system), there are two SPLASH-2 applications—one with an IPC of 0.4 and another with an IPC of 1.3—and, two PARSEC applications—one with an IPC of 0.52 and the other with an IPC of 1.14. Table II (workload used for 256-core system) comprises of five applications from SPLASH-2 such that three applications have IPC up to 0.6 and two have IPC above 1.0. Similarly, the applications chosen from PARSEC benchmark suite have IPC varying from 0.5 to above 1.0.

A different combination of applications would give similar results with a varying degree of gain. It due to the distinctive design differences of our framework, LF, to the state-of-the-art works compared. LF sorts applications according to average power and maps them (this heuristic was chosen as it was found to be better among the explored application selection criteria) to cores in that order. The mapping involves a block-based mapping region selection, followed by thread mapping. On the other hand, the state-of-the-art mapping approach Hayat [7] does not order applications in a certain way before mapping them which makes it less in-control and hence it loses the opportunity to save cores from fast aging. Besides, the other comparison partner, i.e., reliability-aware mapper (RM) [12] selects contiguous rectangular region of cores to map threads of an application which creates thermal hotspots of greater intensity as compared to a dispersed mapping (with sleeping cores in-between) achieved by our framework, LF.

We considered T_{safe} of 90°C, and frequency to vary by 10% in the 64-core system and 30% in the 256-core system reflecting larger impact of PV in larger chips. We considered 5 DVFS VF levels, as shown for a core with 1 GHz maximum operation frequency in Table 3. It is based on the observation that frequency is proportional to voltage [45]. The simulation epoch is subject to the conditions such as the rate of aging, which in turn depends on the performance and power requirements. While choosing too small an epoch is inefficient, choosing a very long epoch will make the intervention less effective as much aging would have occurred already. It was decided to take epoch as one month for the evaluations. The corresponding observed worst-case resistance increment was 4%, which is not too drastic and was capable of avoiding unmanageable core failures. We simulation period was ten years (120 epochs).

We took the ratio of running to dark cores in a cluster (R) as two, to keep a moderate number of dark cores in the cluster. The $max\text{-}hop\text{-}count$ was taken as two since a larger value would incur significant communication overhead. We performed HiMap's stage I with block dimensions: 2×1, 1×2, 2×2, 4×1 and 1×4. For the particular PV maps, we found the block dimension of 2×2, that packs the cores of the block the closest, to be the most favorable for improving $MTTF_{sys}$, in both manycore systems.

TABLE 3: DVFS VF levels.

Voltage (in V)	Frequency (in MHz)
1	1000
0.95	900
0.9	800
0.85	700
0.8	600

We compared LF with two state-of-the-art aging-aware mapping approaches, namely Hayat [7] and RM [12]. To assess the impact of DVFS on these three approaches, we compared with and without integration of DVFS. In the following the DVFS integrated versions are denoted as prefixing 'Di-' to the name of the method. For a fair comparison, we compared lifetime reliability as well as performance.

6.2 Results

6.2.1 Evaluation on a 64-core system

Fig. 8(a) illustrates the system MTTF, which is the minimum MTTF among all the cores of the system. It shows that LF achieves lesser reduction in system MTTF from 2.5 years to 1.5 years at the end of 120 epochs, which corresponds to a 38% reduced system lifetime at the end of 10 years simulation period. On the other hand, HiMap, Hayat, and RM, as well as, the DVFS-integrated versions of Hayat and RM (Di-Hayat and Di-RM) resulted in 100% degradation in system MTTF. Thus, LF led to 62% less decline in the system MTTF.

Fig. 8(b) compares the average MTTF among all the cores as the simulation progressed. It shows that Di-RM and RM both led to a slightly improved average MTTF compared to LF, this is due to up to three applications being skipped from being mapped as shown in Fig. 8(d). Both Hayat and Di-Hayat both resulted in a steadily degrading average MTTF. LF extended the average MTTF by 3× compared to Di-Hayat. Fig. 8(c) shows the number of failed cores resultant due to the different mapping methods. It is observed that Hayat resulted in the greatest number of failed cores with 47 failed cores at the end of 120 epochs. LF led to no failed cores. HiMap resulted in 15 failed cores at the end of the simulation period, while both RM and Di-RM, respectively, led to only 3 and 2 failed cores due to skipping of up to 3 applications causing lesser aging. Thus, LF led to the least number of failed cores. The figure indicates that Hayat led to more rapid aging, than both LF and HiMap from epoch 40 onwards. It is, essentially, because Hayat does not follow any particular order of mapping applications. Unlike Hayat, the HiMap mapper (which is also the mapper used in LF), maps higher power applications on faster blocks of cores, which is found to be beneficial in extending system lifetime reliability during the initial mapping exploration.

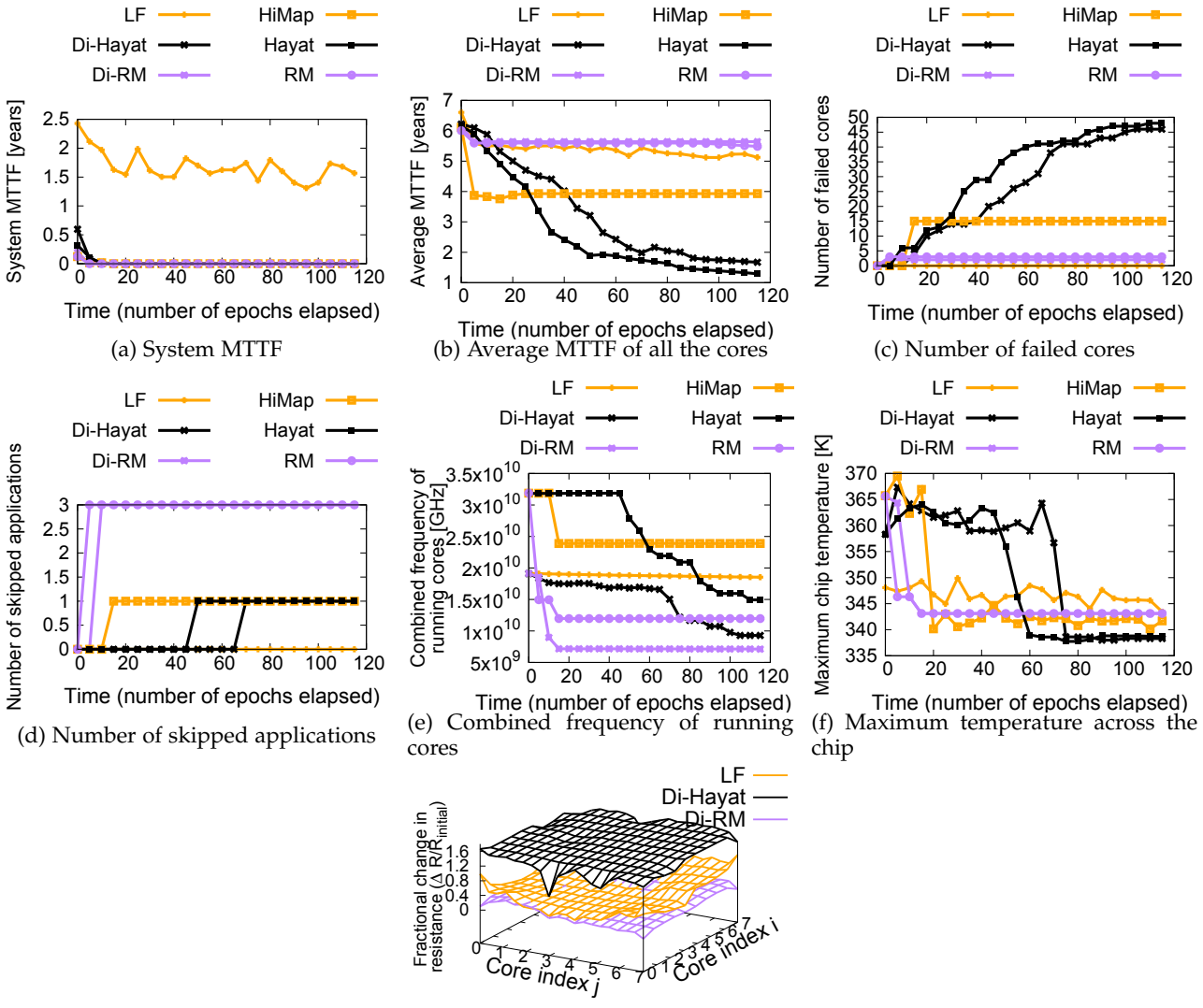
Fig. 8(d) depicts the number of applications skipped from mapping due to not being able to meet the tasks' frequency requirements. As seen here, LF did not skip any application. Both RM and Di-RM led to three applications being skipped by the 10th epoch. HiMap led to one skipped application at epoch 15. Hayat led to a skipped application from epoch 49, and Di-Hayat resulted the same a bit later, from epoch 69 onwards. The reason for applications being skipped by RM is that it could not find cores meeting the

TABLE 1: Workload used for the 64-core system.

App. ID	App. Name	Benchmark	IPC (least value)	IC (max. value)	Thread count
0	cholesky	SPLASH-2	1.31	66653024	8
1	radix	SPLASH-2	0.4	5459539	8
2	blackscholes	PARSEC	1.14	16164994	8
3	bodytrack	PARSEC	0.52	76373790	8

TABLE 2: Workload used for the 256-core system.

App. ID	App. Name	Benchmark	IPC (least value)	IC (max. value)	Thread count
0	barnes	SPLASH-2	0.48	110087840	16
1	cholesky	SPLASH-2	1.31	66653024	8
2	fft	SPLASH-2	1.07	7680429	8
3	radix	SPLASH-2	0.4	5459539	8
4	raytrace	SPLASH-2	0.6	67360133	8
5	blackscholes	PARSEC	1.14	16164994	8
6	bodytrack	PARSEC	0.52	76373790	8
7	swaptions	PARSEC	1.27	56345245	8
8	vips	PARSEC	1.5	112551751	8



(a) System MTTF
(b) Average MTTF of all the cores
(c) Number of failed cores
(d) Number of skipped applications
(e) Combined frequency of running cores
(f) Maximum temperature across the chip
(g) Core aging expressed as fractional change in resistance

Fig. 8: Comparison of lifetime reliability and some related metrics obtained for a 64-core system.

performance requirements for all the threads involved. It highlights the design difference of LF and RM, i.e., while LF maps thread of an application in a distributed manner, across disjointed blocks, RM does it contiguously in a rectangular region of cores.

In accordance to the skipped applications count, the sum of the frequency of running cores varied as shown in Fig. 8(e). The sum denotes the throughput. It is observed that LF maintains a steady throughput, while HiMap, Di-Hayat, Hayat, RM, and DiRM show decline over time due to applications being skipped from being mapped. The maximum temperature across the chip varied as shown in Fig. 8(f) in agreement to the number of applications mapped each epoch by the different methods. Fig. 8(f) shows that LF results in the largest drop (of 10 K) in the maximum chip temperature compared to HiMap, compared to the drop achieved by both Di-Hayat and Di-RM over Hayat and RM, respectively, for the most of the simulation (the difference was only 3 K in both the cases). It indicates that the execution slack available to LF is more than how much is available to both Di-Hayat and Di-RM.

Fig. 8(g) illustrates the core aging in terms of fractional change in resistance. The aging profile compares only the DVFS-integrated methods. It shows that LF led to less aging for all the cores, compared to Hayat. RM resulted in the lowest aging compared to all the methods as it skipped certain applications causing less wearout.

6.2.2 Evaluation on a 256-core system

Fig. 9(a) shows the minimum MTTF among all the cores. It is seen here that LF results in a 63% reduced system MTTF at the end of the simulation compared to that at the beginning. While the rest of the methods resulted in 100% reduction in the system MTTF. Thus, LF led to a 36% less degradation of the system MTTF at the end of simulation period of ten years. Fig. 9(b) compares the average MTTF of all the cores. It shows that among all the methods, LF achieves the best average MTTF, which is 1.87× that of Di-Hayat. HiMap, Hayat, and Di-Hayat show steady decline in the average MTTF. RM and Di-RM, however, show decline and then increment in the average MTTF. Since MTTF depends on the instantaneous aging as well as temperature, it may happen that a reduced temperature owing skipping of certain applications leads to an improved MTTF. As shown in Fig. 9(d), both RM and Di-RM result in applications getting skipped resulting in lowering maximum chip temperature, as shown in Fig. 9(f).

Fig. 9(c) compares the number of cores failed. Up to epoch 27, RM caused highest number of failed cores. After that, Hayat resulted in more failed cores due the fact that it does not manage mapping applications according to their characteristics. Moreover, epoch 40 onwards RM was unable to map at least one application, as seen in Fig. 9(d) due to its contiguous, rectangular mapping style. It was noted that RM skipped more applications than Di-RM. There is a decrease in the number of applications skipped by RM at epoch 102, as before that epoch RM was able to map a 16-threaded application and later had to drop it and in return it was able to map two 8-threaded applications. Per-contra, LF which assigns higher power applications to faster blocks of cores, led to no core failures throughout the simulation.

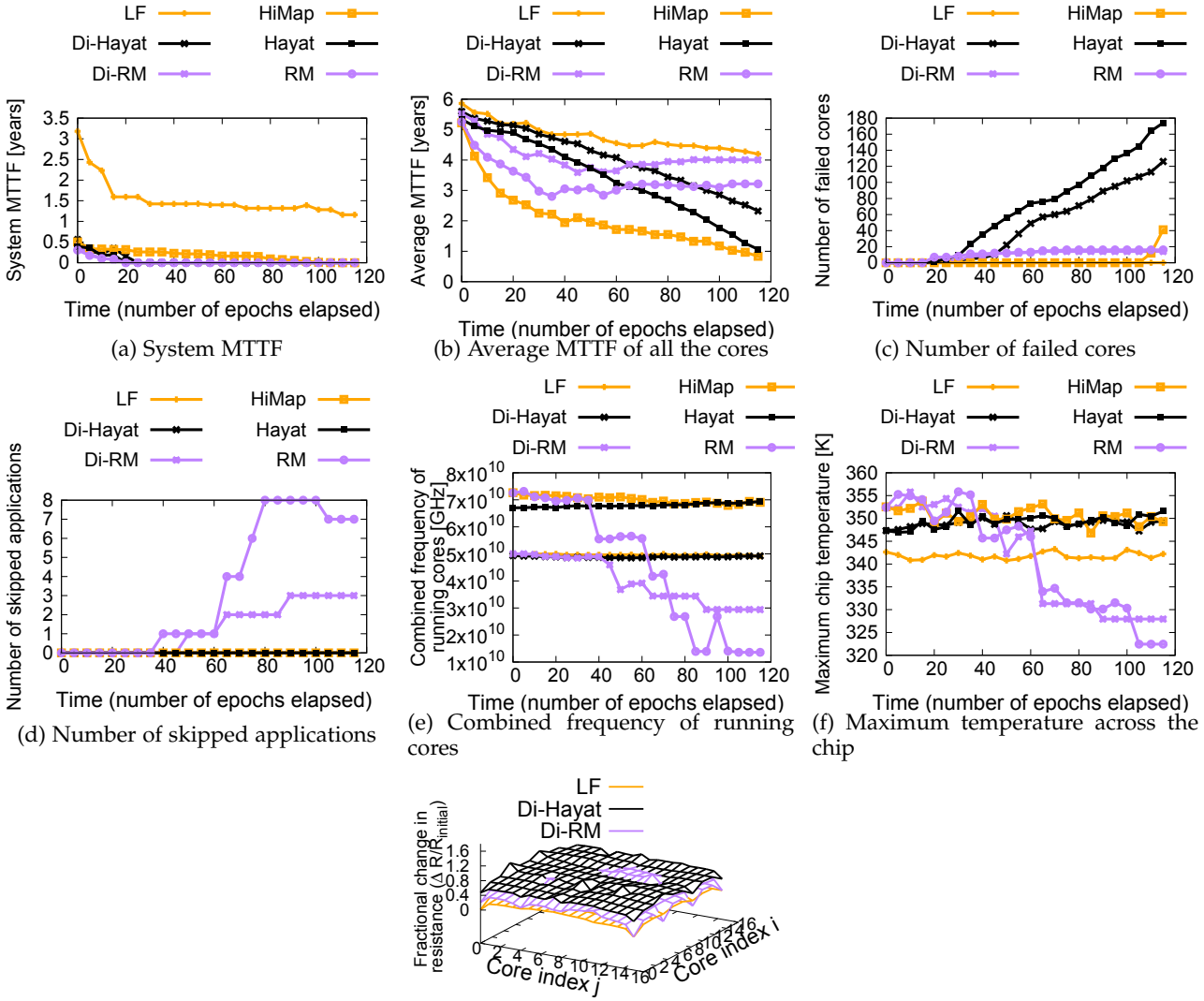
Fig. 9(e) illustrates the sum of the frequency of the running cores. It is observed that LF, HiMap, Di-Hayat, and Hayat maintain a steady throughput all throughout. As expected, in general, the DVFS-integrated versions lead to less throughput than the basic mapping methods, such as LF less than HiMap, and Di-Hayat less than Hayat. Di-RM too shows less throughput than RM up to epoch 71. However, from epoch 72 onwards as RM skips another application, it gives a less throughput than Di-RM.

Fig. 9(f) shows the maximum temperature across the chip achieved by the mapping methods. It is observed that among the basic versions i.e. HiMap, Hayat, and RM, up to epoch 35 RM shows the highest temperature while Hayat shows the lowest. It is due to the fact that RM maps tasks in a contiguous set of cores, while HiMap intersperses sleeping cores among blocks of running cores and Hayat performs a locality-oblivious mapping. From epoch 35 to 39, RM achieved less temperature as the aging lowered the maximum operating frequency. After epoch 39, RM skipped applications and as a result achieved a lower temperature. Similarly, in the case of DVFS-integrated versions, Di-RM leads to the highest temperature up to epoch 48 and LF achieves the lowest temperature. After epoch 48, Di-RM achieves temperature lower than Di-Hayat as it skips an application. Comparing the DVFS integrated versions to the basic versions, LF achieves up to 20 K drop in the maximum chip temperature than HiMap, while Di-Hayat achieved only up to 4 K reduction than Hayat, and Di-RM attained maximum chip temperature similar as that of RM.

Fig. 9(g) compares the core aging across the chip. It is observed that LF achieved the lowest aging for all the cores. Di-Hayat caused the largest amount of aging for 67% of the cores. For the remaining 33% Di-RM caused greater aging.

Comparing the case of 64-core and 256-core systems: With an increase in the number of cores in the system from 64 to 256, the process variation becomes more pronounced, as also reflected by variation considered—10% variation in the former and 30% in the latter. It can be seen from Fig. 8(c) and 9(c), that in the case of 64-cores system, Hayat resulted in more core failures than HiMap—by 50% of the system size. In the case of 256-cores system, also, Hayat led to more core failures than HiMap—by 54.5% of the system size. In both the cases LF led to zero failures throughout. It indicates that HiMap mapper, which is also the mapping logic used by LF, exercises increasingly better mapping decisions than Hayat as the system size increases. HiMap leverages on its ordered mapping process which maps higher power applications to faster blocks of cores. On the other hand, with an increased system size, Hayat performed more poorly compared to HiMap. LF directly benefits from HiMap's ordered mapping strategy. The other comparison partner, RM, caused skipped applications and thus caused lesser number of failed cores than HiMap, in both the cases, due to less wearout. In summary, LF is able to leverage on the application characterisitcs-aware mapping to a greater extent for larger systems, as compared to the state-of-the-art works.

In summary, for both 64-core and 256-core systems, LF has been able to achieve an extended system lifetime reliability while meeting the performance requirements of the workload by successfully harnessing the per-core DVFS capabilities towards reduced



(a) System MTTF
(b) Average MTTF of all the cores
(c) Number of failed cores
(d) Number of skipped applications
(e) Combined frequency of running cores
(f) Maximum temperature across the chip
(g) Core aging expressed as fractional change in resistance

Fig. 9: Comparison of lifetime reliability and some related metrics obtained for a 256-core system.

aging.

7 Conclusions

This paper presents LF, an efficient and scalable hierarchical mapping cum VF-selection approach for improving the lifetime reliability of dark silicon manycore systems while satisfying performance, power, and temperature constraints. LF ensures uniform aging and defers core failure. It contains inter-thread communication distances with aging- and PV-aware cluster-based mapping of threads to healthy cores keeping weaker cores as dark for thermal mitigation. The VF-selection ensures that it meets the performance requirements, and translates the power savings to core health improvements. Experimental results for 64- and 256-core systems validate LF's effectiveness and show significant improvement over the state-of-the-art. LF leverages an online integrated framework encompassing a hierarchical mapping approach and per-core VF selection. It exhibits that such a framework can enable efficient management of the complex space of mapping and per-core DVFS for dark silicon manycore systems towards lifetime reliability enhancement.

8 Acknowledgement

The coauthor Dr. Shafique's contributions in this work are supported in parts by the German Research Foundation (DFG) as part of the GetSURE project in the scope of SPP-1500 priority program "Dependable Embedded Systems".

References

[1] Jayanth Srinivasan, Sarita V Adve, Pradip Bose, and Jude A Rivers. The impact of technology scaling on lifetime reliability. In *International Conference on Dependable Systems and Networks, 2004*, pages 177–186. IEEE, 2004.

[2] Jörg Henkel, Lars Bauer, Nikil Dutt, Puneet Gupta, Sani Nassif, Muhammad Shafique, Mehdi Tahoori, and Norbert Wehn. Reliable on-chip systems in the nano-era: Lessons learnt and future trends. In *Proceedings of the 50th Annual Design Automation Conference*, page 99. ACM, 2013.

[3] Muhammad Shafique and Jörg Henkel. Mitigating the power density and temperature problems in the nano-era. In *2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 176–177. IEEE, 2015.

[4] Xin Fu, Tao Li, et al. Nbti tolerant microarchitecture design in the presence of process variation. In *Proceedings of the 41st annual IEEE/ACM International Symposium on Microarchitecture*, pages 399–410. IEEE Computer Society, 2008.

[5] Jörg Henkel, Heba Khdr, Santiago Pagani, and Muhammad Shafique. New trends in dark silicon. In *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2015.

[6] Muhammad Shafique, Siddharth Garg, Jörg Henkel, and Diana Marculescu. The eda challenges in the dark silicon era: Temperature, reliability, and variability perspectives. In *Proceedings of the 51st Annual Design Automation Conference*, pages 1–6. ACM, 2014.

[7] Dennis Gnad, Muhammad Shafique, Florian Kriebel, Semeen Rehman, Duo Sun, and Jörg Henkel. Hayat: Harnessing dark silicon and variability for aging deceleration and balancing. In *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2015.

[8] Jayanth Srinivasan, Sarita V Adve, Pradip Bose, and Jude A Rivers. The case for lifetime reliability-aware microprocessors. In *ACM SIGARCH Computer Architecture News*, volume 32, page 276. IEEE Computer Society, 2004.

[9] Evelyn Mintarno, Joëlle Skaf, Rui Zheng, Jyothi Bhaskar Velamala, Yu Cao, Stephen Boyd, Robert W Dutton, and Subhasish Mitra. Self-tuning for maximized lifetime energy-efficiency in the presence of circuit aging. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 30(5):760–773, 2011.

[10] Anup Das, Akash Kumar, Bharadwaj Veeravalli, Cristiana Bolchini, and Antonio Miele. Combined dvfs and mapping exploration for lifetime and soft-error susceptibility improvement in mpsocs. In *Proceedings of the conference on Design, Automation & Test in Europe*, page 61. European Design and Automation Association, 2014.

[11] Vijeta Rathore, Vivek Chaturvedi, and Thambipillai Srikanthan. Performance constraint-aware task mapping to optimize lifetime reliability of manycore systems. In *Proceedings of the 26th edition on Great Lakes Symposium on VLSI*, pages 377–380. ACM, 2016.

[12] Mohammad-Hashem Haghbayan, Antonio Miele, Amir M Rahmani, Pasi Liljeberg, and Hannu Tenhunen. A lifetime-aware runtime mapping approach for many-core systems in the dark silicon era. In *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 854–857. IEEE, 2016.

[13] Lin Huang, Feng Yuan, and Qiang Xu. On task allocation and scheduling for lifetime extension of platform-based mpsoc designs. *IEEE Transactions on Parallel and Distributed Systems*, 22(12):2088–2099, 2011.

[14] Mohammad-Hashem Haghbayan, Antonio Miele, Amir M Rahmani, Pasi Liljeberg, and Hannu Tenhunen. Performance/reliability-aware resource management for many-cores in dark silicon era. *IEEE Transactions on Computers*, 66(9):1599–1612, 2017.

[15] Nishit Kapadia and Sudeep Pasricha. Varsha: Variation and reliability-aware application scheduling with adaptive parallelism in the dark-silicon era. In *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*, pages 1060–1065. EDA Consortium, 2015.

[16] Mehmet Basoglu, Michael Orshansky, and Mattan Erez. Nbti-aware dvfs: A new approach to saving energy and increasing processor lifetime. In *Proceedings of the 16th ACM/IEEE international symposium on Low power electronics and design*, pages 253–258. ACM, 2010.

[17] Andrew B Kahng, Siddhartha Nath, and Tajana S Rosing. On potential design impacts of electromigration awareness. In *2013 18th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 527–532. IEEE, 2013.

[18] Jens Lienig. Electromigration and its impact on physical design in future technologies. In *Proceedings of the 2013 ACM International symposium on Physical Design*, pages 33–40, 2013.

[19] Anup Das, Akash Kumar, and Bharadwaj Veeravalli. Reliability-driven task mapping for lifetime extension of networks-on-chip based multiprocessor systems. In *2013 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 689–694. IEEE, 2013.

[20] Shengquan Wang and Jian-Jia Chen. Thermal-aware lifetime reliability in multicore systems. In *2010 11th International Symposium on Quality Electronic Design (ISQED)*, pages 399–405. IEEE, 2010.

[21] Wonyoung Kim, Meeta S Gupta, Gu-Yeon Wei, and David Brooks. System level analysis of fast, per-core dvfs using on-chip switching regulators. In *2008 IEEE 14th International Symposium on High Performance Computer Architecture*, pages 123–134. IEEE, 2008.

[22] Ramnarayanan Muthukaruppan, Tarun Mahajan, Harish K Krishnamurthy, Sumedha Mangal, Am Dhanashekar, Rupak Ghayal, and Vivek De. A digitally controlled linear regulator for per-core wide-range dvfs of atom™ cores in 14nm tri-gate cmos featuring non-linear control, adaptive gain and code roaming. In *ESSCIRC 2017-43rd IEEE European Solid State Circuits Conference*, pages 275–278. IEEE, 2017.

[23] Zeynep Toprak-Deniz, Michael Sperling, John Bulzacchelli, Gregory Still, Ryan Kruse, Seongwon Kim, David Boerstler, Tilman Gloekler, Raphael Robertazzi, Kevin Stawiasz, et al. 5.2 distributed system of digitally controlled microregulators enabling per-core dvfs for the power8 tm microprocessor. In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pages 98–99. IEEE, 2014.

[24] Vijeta Rathore, Vivek Chaturvedi, Amit K Singh, Thambipillai Srikanthan, R Rohith, Siew-Kei Lam, and Muhammad Shaflque. Himap: A hierarchical mapping approach for enhancing lifetime reliability of dark silicon manycore systems. In *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 991–996. IEEE, 2018.

[25] Xiaohang Wang, Amit Kumar Singh, Bing Li, Yang Yang, Hong Li, and Terrence Mak. Bubble budgeting: throughput optimization for dynamic workloads by exploiting dark cores in many core systems. *IEEE Transactions on Computers*, 67(2):178–192, 2017.

[26] Ewerson Carvalho, Ney Calazans, and Fernando Moraes. Heuristics for dynamic task mapping in noc-based heterogeneous mpsocs. In *18th IEEE/IFIP International Workshop on Rapid System Prototyping (RSP'07)*, pages 34–40. IEEE, 2007.

[27] Tejaswini Kolpe, Antonia Zhai, and Sachin S Sapatnekar. Enabling improved power management in multicore processors through clustered dvfs. In *2011 Design, Automation & Test in Europe*, pages 1–6. IEEE, 2011.

[28] Canturk Isci, Gilberto Contreras, and Margaret Martonosi. Live, runtime phase monitoring and prediction on real systems with application to dynamic power management. In *2006 39th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'06)*, pages 359–370. IEEE, 2006.

[29] Yongpan Liu, Huazhong Yang, Robert P Dick, Hui Wang, and Li Shang. Thermal vs energy optimization for dvfs-enabled processors in embedded systems. In *8th International Symposium on Quality Electronic Design (ISQED'07)*, pages 204–209. IEEE, 2007.

[30] Ayse K Coskun, Jose L Ayala, David Atienza, Tajana Simunic Rosing, and Yusuf Leblebici. Dynamic thermal management in 3d multicore architectures. In *Proceedings of the Conference on Design, Automation and Test in Europe*, pages 1410–1415. European Design and Automation Association, 2009.

[31] Taeyoung Kim, Xin Huang, Hai-Bao Chen, Valeriy Sukharev, and Sheldon X-D Tan. Learning-based dynamic reliability management for dark silicon processor considering em effects. In *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 463–468. IEEE, 2016.

[32] Mohammad Salehi, Muhammad Shafique, Florian Kriebel, Semeen Rehman, Mohammad Khavari Tavana, Alireza Ejlali, and Jörg Henkel. dsrelim: Power-constrained reliability management in dark-silicon many-core chips under process variations. In *2015 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ ISSS)*, pages 75–82. IEEE, 2015.

[33] Mohammad Salehi, Mohammad Khavari Tavana, Semeen Rehman, Florian Kriebel, Muhammad Shafique, Alireza Ejlali, and Jörg Henkel. Drvs: Power-efficient reliability management through dynamic redundancy and voltage scaling under variations. In *2015 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, pages 225–230. IEEE, 2015.

[34] Stefan Rusu, Simon Tam, Harry Muljono, Jason Stinson, David Ayers, Jonathan Chang, Raj Varada, Matt Ratta, Sailesh Kottapalli, and Sujal Vora. A 45 nm 8-core enterprise xeon processor. *IEEE Journal of Solid-State Circuits*, 45(1):7–14, 2009.

[35] Vikas Mehrotra, Shiou Lin Sam, Duane Boning, Anantha Chandrakasan, Rakesh Vallishayee, and Sani Nassif. A methodology for modeling the effects of systematic within-die interconnect and device variation on circuit performance. In *Proceedings of the 37th Annual Design Automation Conference*, pages 172–175. ACM, 2000.

[36] Bharathwaj Raghunathan, Yatish Turakhia, Siddharth Garg, and Diana Marculescu. Cherry-picking: exploiting process variations in dark-silicon homogeneous chip multi-processors. In *2013 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 39–44. IEEE, 2013.

[37] Sebastian Herbert and Diana Marculescu. Variation-aware dynamic voltage/frequency scaling. In *2009 IEEE 15th International Symposium on High Performance Computer Architecture*, pages 301–312. IEEE, 2009.

[38] Xin Huang, Tan Yu, Valeriy Sukharev, and Sheldon X-D Tan. Physics-based electromigration assessment for power grid networks. In *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2014.

[39] Wim Heirman, Trevor Carlson, and Lieven Eeckhout. Sniper: Scalable and accurate parallel multi-core simulation. In *8th International Summer School on Advanced Computer Architecture and Compilation for High-Performance and Embedded Systems (ACACES-2012)*, pages 91–94. High-Performance and Embedded Architecture and Compilation Network of Excellence (HiPEAC), 2012.

[40] Sheng Li, Jung Ho Ahn, Richard D Strong, Jay B Brockman, Dean M Tullsen, and Norman P Jouppi. Mcpat: an integrated power, area, and timing modeling framework for multicore and manycore architectures. In *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*, pages 469–480. ACM, 2009.

[41] Wei Huang, Shougata Ghosh, Sivakumar Velusamy, Karthik Sankaranarayanan, Kevin Skadron, and Mircea R Stan. Hotspot: A compact thermal modeling methodology for early-stage vlsi design. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 14(5):501–513, 2006.

[42] Vivek Chaturvedi, Huang Huang, Shangping Ren, and Gang Quan. On the fundamentals of leakage aware real-time dvs scheduling for peak temperature minimization. *Journal of Systems Architecture*, 58(10):387–397, 2012.

[43] Marcelo Mandelli, Luciano Ost, Gilles Sassatelli, and Fernando Moraes. Trading-off system load and communication in mapping heuristics for improving noc-based mpsocs reliability. In *Sixteenth International Symposium on Quality Electronic Design*, pages 392–396. IEEE, 2015.

[44] Cristiana Bolchini, Matteo Carminati, Antonio Miele, Anup Das, Akash Kumar, and Bharadwaj Veeravalli. Run-time mapping for reliable many-cores based on energy/performance trade-offs. In *2013 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS)*, pages 58–64. IEEE, 2013.

[45] Vivek Chaturvedi Gang Quan. Leakage conscious dvs scheduling for peak temperature minimization. In *Asia and South Pacific Design Automation Conference*, pages 135–140.

Vijeta Rathore is a Ph. D. student at the School of Computer Science and Engineering, Nanyang Technologiocal University (NTU). She earned a Masters in Electrical Engineering and Computer Science from Seoul National University (SNU), South Korea in 2011. She received a Bachelor of Technology in Electical Engineering from Indian Institute of Technology (IIT) Delhi, India in 2008. Her research interests include aging and lifetime optimization in manycore processors, embedded software and network virtualization.

Vivek Chaturvedi (M'09) is currently working as Assistant Professor in the Department of Computer Science and Engineering at Indian Institute of Technology, Palakkad, India. Previously, he was a Research Scientist in the School of computer science and engineering at Nanyang Technological University, Singapore. He received his M.S. from Syracuse University, NY in 2008 and PhD from Florida International University, Miami in 2013. He also worked in Sun Microsystems as a Student intern in 2007. His current research interest includes power and thermal optimization in manycore processors including both 2D and 3D architectures. He is also actively working on Hardware security and reliability. He is a reviewer and has served as TPC for several IEEE/ACM Journals and Conferences.

Amit K. Singh is a Lecturer (Assistant Professor) at University of Essex, UK. He received the B.Tech. degree in Electronics Engineering from Indian Institute of Technology (Indian School of Mines), Dhanbad, India, in 2006, and the Ph.D. degree from the School of Computer Engineering, Nanyang Technological University (NTU), Singapore, in 2013. He was with HCL Technologies, India for a year and half until 2008. He has a post-doctoral research experience for over five years at several reputed universities. His current research interests are system level design-time and run-time optimization of 2D/3D multi-core systems for performance, energy, temperature, reliability and security. He has published over 80 papers in reputed journals/conferences, and received several best paper awards, e.g. IEEE TC February 2018 Featured Paper, ICCES 2017, ISORC 2016, PDP 2015, HiPEAC 2013 and GLSVLSI 2014 runner up. He has served on the TPC of IEEE/ACM conferences like NoCArc, DATE, CASES and CODES+ISSS.

Thambipillai Srikanthan joined Nanyang Technological University, Singapore, in 1991. He founded the Centre for High Performance Embedded Systems (CHiPES) in 1998 and elevated it to a University Level Research Centre in 2000. He currently holds a full professor and joint appointments as the Director of CHiPES which is a 100 strong center and also the Director of the Intelligent Devices and Systems (IDeAS) Cluster. He has authored over 250 technical papers. His research interests include design methodologies for complex embedded systems, architectural translations of compute intensive algorithms, computer arithmetic, high-speed techniques for image processing, and dynamic routing.

Muhammad Shafique (M'11 - SM'16) received the Ph.D. degree in computer science from the Karlsruhe Institute of Technology, Germany, in 2011. He is currently a Full Professor with the Department of Informatics, Institute of Computer Engineering, TU Wien, Austria, where he is directing the group on Computer Architecture and Robust, Energy-Efficient Technologies. He holds one U.S. patent and over 200 papers in premier journals and conferences. His research interests include computer architecture, energy-efficient systems, robust computing, hardware security, brain-inspired computing, emerging technologies, and embedded systems. His research has a special focus on cross-layer analysis, the modeling, design, and optimization of computing and memory systems, and their integration in the Internet of Things and smart cyber-physical systems. He is a Senior Member of the IEEE and a member of the ACM, SIGARCH, SIGDA, SIGBED, and HiPEAC. He received the 2015 ACM/SIGDA Outstanding New Faculty Award, six gold medals, and several best paper awards and nominations at prestigious conferences. He served on the program committees of several conferences and gave several invited talks, tutorials, and keynotes.