Reliability Assessment of Many-Core Dynamic Thermal Management

Alzemiro Silva*, Iaçanã Weber*, André Luís del Mestre Martins†, Fernando Gehm Moraes*
*School of Technology - PUCRS – Av. Ipiranga 6681, 90619-900, Porto Alegre, Brazil
†Sul-Rio-Grandense Federal Institute, IFSul, Charqueadas, Brazil
{alzemiro.silva, iacana.weber}@edu.pucrs.br, almmartins@charqueadas.ifsul.edu.br, fernando.moraes@pucrs.br

2022 IEEE International Symposium on Circuits and Systems (ISCAS) | 978-1-6654-8485-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/ISCAS48785.2022.9937286

***Abstract*—Power density can limit the amount of energy a many-core can consume. A many-core running in its maximum performance can lead to safe temperature violation and reliability issues. The literature presents dynamic thermal management (DTM) techniques that guarantee system operation according to safe temperature restrictions. The state-of-art also targets dynamic reliability management (DRM), aiming for longer lifetime reliability, using the same actuation knobs used to control the temperature. This work assesses the reliability of DTM techniques applied in high and dynamic workloads, showing that DTMs keep the temperature within safe limits and increase system lifetime reliability.**

***Index Terms*—Many-core Systems, Dynamic Thermal Management, Lifetime Reliability**

I. Introduction and Related Work

The increase in power density creates the localized overheating effect known as hotspot. Dynamic temperature behavior can create reliability threats, increase power and increase cooling costs [1]. Recent management approaches assume the system itself executes the adaptation to increase its performance, keeping it within physical operating limits [2].

System temperature monitoring enables the deployment of dynamic thermal management (DTM) techniques to ensure operation within specified temperature limits, increase reliability, reduce energy consumption and extend system lifetime. Due to the many challenges of monitoring performance and temperature in many-core systems, proposals are developed using high-level modeling tools such as GEM5 [3] and McPAT [4].

In dynamic reliability management (DRM) approaches, the main metric used to manage the system is the expected lifetime, extracted from a theoretical model. While some of the DRM proposals consider thermal cycling as a metric to be managed [5, 6], others focus on electromigration and time dependent dielectric breakdown [7], or negative bias temperature instability [8]. We also observed that two works proposed learning techniques to manage the system [5, 8], while [7] proposed an offline MILP (mixed-integer linear programming model) approach and [6] proposed a more traditional heuristic to deal with thermal cycling.

After reviewing the state-of-the-art for DTMs and DRMs, we found the following gaps in the literature:

- Most works focus on high-level architectural simulations with rough power and temperature estimations.
- Some works propose a patterning approach [9], which may limit the system utilization, while others rely on complex heuristics designed to be executed at design time, requiring previous knowledge of the executed workload.
- Reviewed works focus on thermal management or reliability management. None of the works propose thermal management with a lifetime reliability analysis, showing that the proposed thermal management also improves lifetime reliability.

The *goal* of this paper is to demonstrate that DTM techniques, applied in high and dynamic workloads, keep the temperature within safe limits and increases the system lifetime reliability. The main *original contribution* of this work is the reliability assessment of many-cores using state-of-the-art DTMs.

This work fills the above gaps. The architecture presented in Section II has an RTL and an abstract model. The RTL model enables an accurate power and temperature estimation, filling the first gap. Instead of adopting a pattering mapping, we adopt a DTM method based on the observe-decide-act paradigm, enabling better system usage (Section III), filling the second gap. Finally, Sections and IV and V assess the reliability of DTM techniques, our main contribution, filling the last gap. Section VI concludes this paper.

II. Reference Architecture

The many-core adopted in this work [10] has a set of homogeneous processing elements (PEs) and peripherals connected to the borders of the PE region. Each PE contains a processor, a network interface with DMA capabilities [11], local memory, and the NoC router. The many-core contains at least two peripherals: (*i*) application injector (App_{inj}), responsible for the deployment of new applications in the system; (*ii*) TEA (Temperature Estimation Accelerator), responsible for temperature estimation [12].

The many-core adopts *cluster* management [13]. Each PE may act as: *SP*: slave PE, execute applications tasks; *MP*: manager PE, execute management actions such as application admission, mapping, remapping, DVFS control. MPs may be local to a given cluster (*CM* – cluster manager), or execute global actions besides the cluster management (*GM* – global manager).

The many-core has three models: (*i*) two clock cycle-accurate models; (*ii*) an instruction-level model. The clock cycle-accurate models have two equivalent descriptions: synthesizable VHDL and SystemC. The instruction-level model uses OVP [14] as the simulation framework, with peripherals and other PE components, like network interface and routers, described in C.

TEA [12] computes periodically, at runtime, the temperature of each PE. The power estimation considers the number of executed instructions, the number of flits traversing the

router, and the local memory accesses. TEA enables fine-grain temperature monitoring, i.e., at the PE level, to provide control decisions to ensure safe operation. MatEx [15] is the reference algorithm implemented in TEA to give the temperature estimation at runtime.

Each PE runs a multi-task operating system, with support for message passing (MPI-like API). Applications are described in C language, modeled as Communicating Task Graphs (CTG).

III. Dynamic Thermal Management – DTM

DTM requires multiple layers of sensors and actuators to provide system self-awareness. Self-aware systems are capable of adapting their behavior and resources to automatically find the best way to accomplish a given goal despite changing environmental conditions and demands [16]. We adopt the Observe-Decide-Act (ODA) paradigm to connect the monitoring to the actuation infrastructure. As illustrated in Figure 1, the system monitors key features, applies a control and decision algorithm, and deploys appropriate actions to adapt to changes in its state.

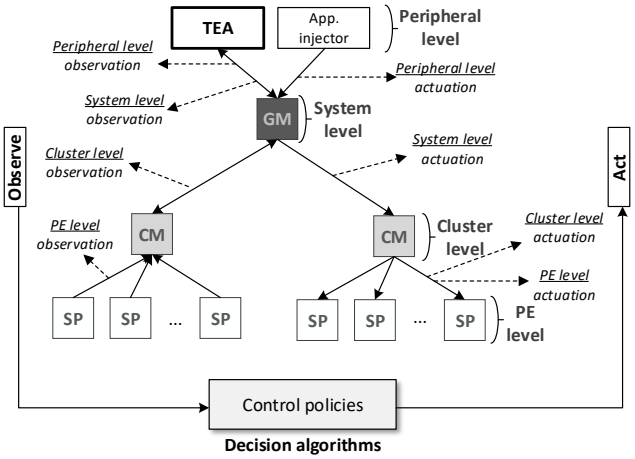

Fig. 1. Observe-Decide-Act paradigm is the methodology for the development of DTM applied to the reference platform.

Each PE monitors its power consumption and sends this information to its cluster manager. The cluster-level observation flows in two directions, the power observation is done from the CM to the GM, and the temperature observation flows in the opposite direction. The system-level observation is the power dissipation information that the GM sends to TEA. The peripheral-level observation is the temperature of all PEs that TEA sends to the GM.

The right side of Figure 1 shows that actuation occurs in the top-down direction. The peripheral-level actuation may occur when a new application enters into the system. The system-level actuation is decided at the GM, which chooses a cluster to run the application. The cluster-level actuation may be an application mapping or a task migration, which is decided at the CM and sent to the SPs. The PE level actuation, as the DVFS, is also decided at the CM and forwarded to the destination PE.

A. Actuation Knobs

The actuation knobs are the tools the decision algorithms have available to manage the system.

- Application Admission: selects the cluster with free resources to receive an incoming application.
- Task Mapping corresponds to the allocation of the application tasks to the selected PEs in a cluster.
- *Task migration*: move a task from a source SP to a target SP. The task migration employs a low latency protocol for many-cores with distributed memory.
- *Dynamic voltage and frequency scaling* (DVFS) consists of changing the clock frequency and the supply voltage levels according to the PEs temperature to guarantee that the temperatures are not higher than the critical core temperature. DVFS is the most commonly used DTM mechanism to manage the temperature in many-core systems [17].

B. Decision Heuristics

The present work adopts the DTM method proposed by Silva et al. [18], named Proportional, Integral, and Derivative Temperature Management (**PIDTM**). PIDTM seeks the minimization of hotspots occurrence considering in its cost-function the current temperature of each PE – *Proportional* value, the average temperature at each PE for a predefined number of monitoring windows – *Integral* value, and how the temperature behaves (i.e., if it is increasing or decreasing) – *Derivative* value. PIDTM uses the actuation knobs described in the previous section.

The Temperature Management with Energy Constraint (**TMEC**) heuristic optimizes PIDTM. TMEC includes the energy consumption of each PE in the PIDTM score. The result obtained with TMEC is a better workload distribution, avoiding PEs with no tasks assigned to them.

IV. Flow for Reliability Estimation

Srinivasan et al. [19, 20] proposed a methodology for evaluating processor lifetime reliability called RAMP. RAMP stands for Reliability Aware Micro-Processors and provides a simulation tool to be used together with timing, power, and thermal simulators.

The current version of the tool is RAMP 2.0. We use RAMP 2.0 since it is the most comprehensive simulation tool for many-core systems, modeling five aging mechanisms: electromigration (EM), Time Dependent Dielectric Breakdown (TDDB), Negative Bias Temperature Instability (NBTI), Stress Migration (SM), Thermal Cycling (TC). RAMP is a tool that could be easily adapted to estimate the mean time to failure (MTTF), unlike LifeSim [21], which has its architectural simulator. Finally, RAMP is the tool of choice because even the newer works use the RAMP models, with one exception being the EM model in LifeSim [22], which presents a newer but less conservative model for MTTF estimation due to EM.

The tool initially generates a failure in time (FIT) file containing the FIT for each model in each PE of the system. Then, the FIT file is analyzed by the Monte Carlo tool, which is provided with RAMP 2.0. The output of the Monte Carlo tool is the MTTF measured in years for the whole system,

considering all aging effects in all PEs after the statistical analysis.

RAMP does not provide a tool to analyze the lifetime of systems. Instead, it provides code snippets to be integrated into a simulator to analyze the lifetime at runtime. We analyze the temperature and power logs of the simulations to estimate the lifetime for each test case using the code provided by RAMP. Figure 2 shows the RAMP usage flow used to generate the results presented in Section V.

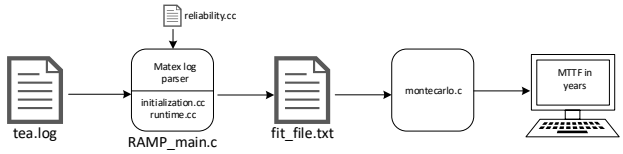

Fig. 2. RAMP 2.0 usage flow. *tea.log* corresponds to temperature log.

The tool *RAMP_main.c* was created to analyze the log provided by TEA during the simulation (*tea.log*) with the power and temperature samples for each simulation. *RAMP_main.c* uses the code snippets *initialization.cc* and *runtime.cc* provided with RAMP to generate the FIT file (fit_file.txt) to be analyzed by the Monte Carlo simulation. The Monte Carlo simulation (*montecarlo.c*) was used without changes, and it outputs on the terminal the expected MTTF in years based on the analysis of the FIT file.

V. Evaluation of the DTM heuristics

A set of benchmarks were used to evaluate the DTM heuristics: (*i*) Dijkstra (DIJ), with 7 tasks; (*ii*) audio and video application (AV), with 7 tasks, that implements a video and audio decoding pipeline in parallel; (*iii*) AES encoder, with 5 tasks; (*vi*) MPEG decoder (5 tasks); (*v*) DTW - *Digital Time Warping* (6 tasks); communication intensive synthetic application (SYN) (6 tasks); (*vi*) sort application, with 4 tasks, that implements a parallel quick sort algorithm.

The evaluation of the DTM approaches considers five different scenarios. Table I presents the applications used in each scenario.

TABLE I
Thermal evaluation scenarios.

Scenario		Applications							#Tasks
		MPEG	DTW	AES	SYN	DIJ	Sort	AV	
HW	High Workload	1	2	0	1	1	2	1	47
LW	Low Workload	1	1	0	0	1	0	0	18
AW	Average Workload	1	1	0	1	1	1	1	36
DW1	Dynamic workload 1	1	2	10	4	4	2	2	142
DW2	Dynamic workload 2	2	6	10	4	4	6	2	192

The first three scenarios are HW – high workload, AW – average workload, LW – low workload. In these scenarios, the workload remains constant throughout the simulation, corresponding to 75%–50%–33% of system resources for HW, AW, and LW, respectively. The last two scenarios execute dynamic workloads (DW1 and DW2) with high and low workloads, executing more tasks than scenarios with a fixed workload. The goal in the dynamic workload scenarios is to produce peaks and valleys of system utilization.

These scenarios were executed in an 8x8 system, partitioned in 4 4x4 clusters using the reference architecture (Section II). We compare PIDTM and TMEC (state-of-the-art DTMs) with a patterning mapping [23] and a Multi-Objective Resource Management (MORM) approach [24]. Recent works on DTM use the patterning approach in a known set of applications, without the support of remapping by task migration. MORM manages applications without considering temperature (it uses power dissipation and performance as cost functions), using task migration dynamically.

A. Effect of the DTM in the Temperature

Figure 3 presents a peak temperature graph for the HW scenario. The peak temperature is the highest temperature that one core achieved during execution.

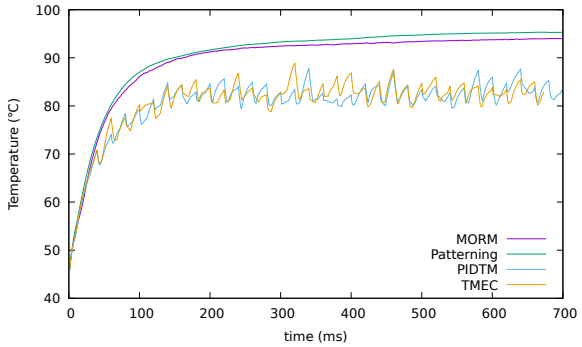

Fig. 3. MORM, Patterning, PIDTM, TMEC peak temperature graph for the high workload (HW) scenario.

Similar average and peak temperatures were observed for MORM/Patterning and PIDTM/TMEC. Using PIDTM/TMEC the peak temperature reduced by 7%, and the average temperature by 10.3%.

PIDTM tends to present better average temperature results than TMEC. However, in clusters with a low workload, some PEs are not used by PIDTM. Despite the advantage related to the average temperature observed in the PIDTM, the TMEC presents a better workload distribution among PES. Thus, the TMEC heuristic is a good candidate when the focus is to reduce the probability of failures induced by the temperature stress in the same PE.

B. Effect of DTM on the Lifetime

This section presents the expected lifetime results for the reference platform.

1) Lifetime evaluation in an clock-cycle RTL model: The evaluation of lifetime reliability requires long simulations to capture the steady-state temperature behavior of the selected test case. Long simulations using the RTL description of the reference platform are time-consuming, usually about two days of simulation for each second of results. Thus, we used only the high workload (HW) test case to evaluate the MTTF in the RTL platform. The test case was executed in an 8x8 system, and Figure 4 presents the expected MTTF results for this scenario.

Results show that the expected lifetime when using MORM in the high workload scenario is about 19 years of operation,

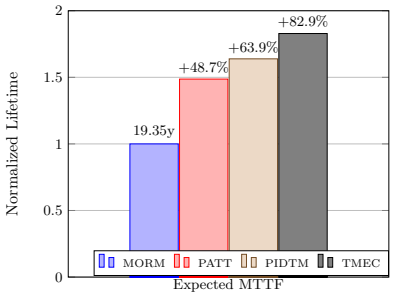

Fig. 4. Expected MTTF comparison between MORM, patterning, PIDTM and TMEC for the HW scenario (clock-cycle RTL model).

while the patterning approach may increase the lifetime by 48.7%, even achieving a slightly higher overall peak temperature. This observation shows that reducing only the peak temperature is not enough to provide a higher MTTF. The average temperature is also important to consider when the objective is to increase the lifetime.

PIDTM and TMEC heuristics both get a lifetime improvement when comparing with MORM and patterning approaches. TMEC got the best MTTF result, even getting a higher peak temperature than PIDTM. This is due to the workload distribution provided by the TMEC heuristic, reinforcing that the overall peak temperature is not the most important metric to improve the MTTF.

2) Lifetime evaluation in an abstract model (OVP): The OVP platform enables faster simulations than the RTL platform. Therefore, we executed four scenarios to evaluate the expected lifetime in the OVP platform: AW, HW, DW1, DW2. All scenarios were executed using: (*i*) spiral mapping, which is the initial mapping used in MORM; (*ii*) patterning mapping; (*iii*) PIDTM; (*iv*) TMEC.

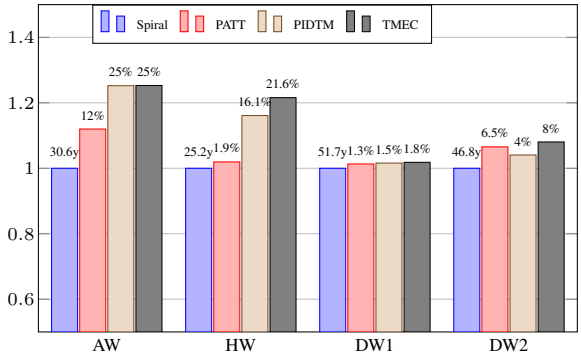

Fig. 5. Spiral, Patterning, PIDTM and TMEC normalized MTTF results (OVP model).

Figure 5 shows the results of the expected MTTF for the four executed scenarios. Considering the spiral mapping as a baseline, it is possible to observe that the higher the workload is, the lower is the expected MTTF. In HW, the spiral mapping got an average MTTF of 25.2 years, while in DW1, which is the lightest workload, the expected lifetime was about 51.7 years. The improvements obtained with the PIDTM and TMEC heuristics were also proportional to the workload. Although the AW shows the greater improvements when using the PIDTM and TMEC heuristics (25%), the DW1 shows that in a lighter workload, the improvements are negligible. In DW2, which has a slightly higher workload than DW1, we got a maximum improvement of 8% in the expected lifetime when using TMEC. For the HW, which is the worst-case scenario for lifetime reliability, TMEC got an improvement of 21.6% in the lifetime, while PIDTM got 16.1%.

DW1 is the lightest workload of all executed scenarios. There are peaks and valleys of processor usage in this scenario. Thus there is no average temperature advantage for the PIDTM and TMEC heuristics since there is a cooldown period during the valleys of workload. As the results provided by RAMP consider the temperature during the whole simulation, and the average temperatures are similar in all simulations, there are no significant differences in lifetime reliability for this scenario.

Comparing the HW results of Figure 5 to the results obtained with the RTL simulations from Figure 4, it is possible to observe that the baseline for MORM, compared to the spiral mapping in the OVP simulation, is significantly smaller. This happens for two main reasons: (*i*) the architectures are different; (*ii*) tasks in the quantum-based simulation wait for longer periods for data to compute. While in the RTL model, we model a MIPS-like CPU, with a subset of the MIPS instructions, the OVP CPU is an OR1K instance, which uses Open RISC instructions, which may cause different computation times for each architecture. Furthermore, the communication latency induced by the quantum-based simulation in OVP makes computation-intensive tasks wait more time for the required data. So, these tasks might generate less heat in the OVP model, leading to a longer lifetime, especially with the spiral mapping, where the computation-intensive tasks may be mapped close to each other.

VI. Conclusions

This paper presented a study related to the lifetime reliability for state-of-the-art DTMs targeting NoC-based many-cores. We choose RAMP 2.0 to estimate MTTF using the DTM heuristics since this tool considers a rich set of aging effects. The results confirmed our hypothesis that managing the system temperature is important to improve the lifetime reliability of the system. We observed that the peak temperature reduction is relevant, but the average temperature also plays a major role in improving the expected system lifetime. We also concluded that choosing the best PE to reduce the temperature is not ideal for improving the system lifetime. Balancing the workload between all PEs is essential to distribute the aging accumulation in each PE.

Directions for future works include adding reliability in DTM heuristics. The TMEC heuristic provided significant lifetime improvements, but improvements are expected if we include in this heuristic parameters affecting the system lifetime, as thermal cycling.

Acknowledgment

The authors would like to thank Imperas Software and Open Virtual Platforms for their support and access to their models and simulator. This work was financed in part by CNPq , grant 309605/2020-2, and CAPES, Finance Code 001.

REFERENCES

[1] M. El Ahmad, M. Najem, P. Benoit, G. Sassatelli, and L. Torres, "PoETE: A Method to Design Temperature-Aware Integrated Systems," *Journal of Low Power Electronics*, vol. 14, no. 1, pp. 1–7, 2018.

[2] A. Rahmani, M. Haghbayan, A. Kanduri, A. Y. Weldezion, P. Liljeberg, J. Plosila, A. Jantsch, and H. Tenhunen, "Dynamic power management for many-core platforms in the dark silicon era: A multi-objective control approach," in *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED)*, 2015, pp. 219–224.

[3] N. Binkert *et al.*, "The Gem5 Simulator," *SIGARCH Comput. Archit. News*, vol. 39, no. 2, pp. 1–7, 2011.

[4] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi, "McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures," in *Proceedings of the ACM/IEEE International Symposium on Microarchitecture (MICRO)*, 2009, pp. 469–480.

[5] A. Das, B. M. Al-Hashimi, and G. V. Merrett, "Adaptive and hierarchical runtime manager for energy-aware thermal management of embedded systems," *ACM Transactions on Embedded Computing Systems*, vol. 15, no. 2, pp. 24:1–24:25, 2016.

[6] M.-H. Haghbayan, A. Miele, Z. Zou, H. Tenhunen, and J. Plosila, "Thermal-Cycling-aware Dynamic Reliability Management in Many-Core System-on-Chip," in *Proceedings of the Design, Automation Test in Europe Conference (DATE)*, 2020, pp. 1229–1234.

[7] W. Liu, J. Yi, M. Li, P. Chen, and L. Yang, "Energy-Efficient Application Mapping and Scheduling for Lifetime Guaranteed MPSoCs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 1, pp. 1–14, 2019.

[8] V. Rathore, V. Chaturvedi, A. K. Singh, T. Srikanthan, and M. Shafique, "LifeGuard: A Reinforcement Learning-Based Task Mapping Strategy for Performance-Centric Aging Management," in *Proceedings of the ACM/IEEE Design Automation Conference (DAC)*, 2019.

[9] A. Kanduri, M.-H. Haghbayan, A.-M. Rahmani, P. Liljeberg, A. Jantsch, and H. Tenhunen, "Dark silicon aware runtime mapping for many-core systems: A patterning approach," in *ICCD*, 2015, pp. 573–580.

[10] M. Ruaro, L. Caimi, V. Fochi, and F. G. Moraes, "Memphis: a Framework for Heterogeneous Many-core SoCs Generation and Validation," *Design Automation for Embedded Systems*, vol. 23, no. 3-4, pp. 113–122, 2019.

[11] M. Ruaro, F. B. Lazzarotto, C. A. Marcon, and F. G. Moraes, "DMNI: A specialized network interface for NoC-based MPSoCs," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, 2016, pp. 1202–1205.

[12] A. H. L. da Silva, I. I. Weber, A. L. d. M. Martins, and F. G. Moraes, "Hardware Accelerator for Runtime Temperature Estimation in Many-cores," *IEEE Design Test of Computers*, vol. *preprint*, pp. 1–7, 2021.

[13] G. Castilhos, M. Mandelli, G. Madalozzo, and F. G. Moraes, "Distributed resource management in NoC-based MPSoCs with dynamic cluster sizes," in *Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, 2013, pp. 153–158.

[14] Imperas, "Open Virtual Platforms - OVP," 2019, source: http://www.ovpworld.org/. [Online]. Available: http://www.ovpworld.org/

[15] S. Pagani, H. Khdr, W. Munawar, J. Chen, M. Shafique, M. Li, and J. Henkel, "MatEx: Efficient transient and peak temperature computation for compact thermal models," in *Proceedings of the Design, Automation Test in Europe Conference (DATE)*, 2015, pp. 1515–1520.

[16] N. Dutt, A. Jantsch, and S. Sarma, "Self-Aware Cyber-Physical Systems-on-Chip," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2015.

[17] M. S. Mohammed, A. A. M. Al-Kubati, N. Paraman, A. A.-H. Ab Rahman, and M. N. Marsono, "DTaPO: Dynamic Thermal-Aware Performance Optimization for Dark Silicon Many-Core Systems," *Electronics*, vol. 9, no. 11, 2020.

[18] A. H. L. da Silva, A. L. d. M. Martins, and F. G. Moraes, "Mapping and Migration Strategies for Thermal Management in Many-Core Systems," in *Proceedings of the Symposium on Integrated Circuits and Systems Design (SBCCI)*, 2020, pp. 1–6.

[19] J. Srinivasan, S. Adve, P. Bose, J. Rivers, C.-K. Hu, P. Emma, B. Linder, and E. Wu, "Ramp: A model for reliability aware microprocessor design," *RC23048 Computer Science*, vol. 29, pp. 312–122, 2004.

[20] J. Srinivasan, "Lifetime Reliability Aware Microprocessors," Ph.D. dissertation, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA, 2006.

[21] R. Rohith, V. Rathore, V. Chaturvedi, A. K. Singh, S. Thambipillai, and S.-K. Lam, "LifeSim: A lifetime reliability simulator for manycore systems," in *Proceedings of the IEEE Computing and Communication Workshop and Conference (CCWC)*, 2018, pp. 375–381.

[22] X. Huang, T. Yu, V. Sukharev, and S. X.-D. Tan, "Physics-based electromigration assessment for power grid networks," in *Proceedings of the ACM/IEEE Design Automation Conference (DAC)*, 2014, pp. 1–6.

[23] L. Yang, W. Liu, W. Jiang, M. Li, P. Chen, and E. H.-M. Sha, "Fotonoc: A folded torus-like network-on-chip based many-core systems-on-chip in the dark silicon era," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 7, pp. 1905–1918, 2017.

[24] A. L. d. M. Martins, A. H. L. da Silva, A. M. Rahmani, N. Dutt, and F. G. Moraes, "Hierarchical adaptive Multi-objective resource management for many-core systems," *Journal of Systems Architecture*, vol. 97, pp. 416–427, 2019.