

Received June 19, 2021, accepted August 23, 2021, date of publication September 1, 2021, date of current version September 14, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3109717

PEW: Prediction-Based Early Dark Cores Wake-up Using Online Ridge Regression for Many-Core Systems

MOHAMMED SULTAN MOHAMMED^{1,2}, (Member, IEEE), NORLINA PARAMAN¹,
AB AL-HADI AB RAHMAN¹, FUAD A. GHALEB^{3,4}, AHLAM AL-DHAMARI^{1,2},
AND MUHAMMAD NADZIR MARSONO¹

¹School of Electrical Engineering, Faculty of Engineering, Universiti Teknologi Malaysia, Johor Bahru, Johor 81310, Malaysia

²Department of Computer Engineering, Hodeidah University, Hodeidah, Yemen

³School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, Johor Bahru, Johor 81310, Malaysia

⁴Department of Computer and Electronic Engineering, Sana'a Community College, Sana'a, Yemen

Corresponding authors: Mohammed Sultan Mohammed (sammohammed2@live.utm.my) and Muhammad Nadzir Marsono (mnadzir@utm.my)

ABSTRACT Future many-core systems need to address the dark silicon problem, where some cores would be turned off to control the chip's thermal and power density, which effectively limits the performance gain from having a large number of processing cores. Task migration technique has been previously proposed to improve many-core system performance by moving tasks between active and dark cores. As task migration imposes system performance overhead due to the large wake-up latency of the dark cores, this paper proposes a prediction-based early wake-up (PEW) to reduce the dark cores' wake-up latency during task migration. A window-based online ridge regression (RR) is used as the prediction model. The prediction model uses the past window's thermal, power, and core status (i.e., active or dark) to predict the future core temperatures at run-time. If task migration is predicted in the next control period, the proposed PEW puts the dark cores in a power state with low wake-up latency. Thus, the proposed PEW reduces the time for the dark cores to start executing the tasks. The comparison results show that our proposed PEW reduces the completion time by up to 7.9% and 4.1% compared to non-early wake-up (NoEW) and a fixed threshold wake-up (FEW), respectively. It also shows that the proposed PEW increases the MIPS/Watt by up to 5.5% and 2.3% over NoEW and FEW, respectively. These results show that the proposed PEW improves the many-core system's overall performance in terms of reducing dark cores' wake-up latency and increasing the number of executed instructions per Watt.

INDEX TERMS Dark silicon, many-core systems, task migration, dynamic voltage frequency scaling (DVFS), ridge regression, early wake-up, dark core wake-up latency.

I. INTRODUCTION

The key concept of increasing computing circuits performance was increasing the processor frequency guided by Dennard scaling [1]. However, around 2005, Dennard scaling ended, where the power per transistor could no longer scale down with the scaling of fabrication technology. This led to an end to increasing the frequency of single-core processors due to the high power density. To overcome this problem, many-core systems were introduced by integrating more cores

The associate editor coordinating the review of this manuscript and approving it for publication was Songwen Pei¹.

with lower operating frequencies into the processor's chip to improve the overall computing performance.

Adding more cores by reducing the technology size, according to Moore's law [2], increases the total power of many-core systems that resulted in higher chip temperatures. Thus, only a part of the many-core system can be in an active state (i.e., turned on) while the rest should remain in a dark state (i.e., turned off). Turning off some cores will limit the performance gain from the increasing number of cores in many-core systems. This limitation from using all the processing cores is called the dark silicon problem [3], which is expected to be a major issue in future many-core systems.

According to Ref. [3], [4], over half of the cores in many-core system-on-chip (MCSoc) would be dark cores in 8-nm technologies. This prediction led researchers to identify techniques for the dark silicon problem to improve the performance of many-core systems under either power budget constraints [5]–[14] or thermal constraints [15]–[19]. To avoid a run-time thermal violation, most of these techniques use dynamic thermal management (DTM), such as task migration and dynamic voltage frequency scaling (DVFS). However, the techniques that used task migration avoid migrating tasks to dark cores due to the high wake-up latency of the dark cores, where the dark cores need a longer time to turn on all core components that were previously off in the dark state. Previous studies in Ref. [18], [19] show that migrating tasks from active to dark cores can improve the many-core system performance. However, the dark cores' wake-up latency is significant as dark cores need a longer time to be ready to run the upcoming tasks. In a dark core state, all core components are off and need time to be back to operate normally. Studies in Ref. [20], [21] proposed an early wake-up to address the wake-up latency of the dark cores. However, these studies use a fixed wake-up threshold, which may not suit high thermal fluctuating applications.

This paper proposes a prediction-based early wake-up (PEW) technique to reduce the wake-up latency impact of dark cores during task migration. The proposed PEW consists of two parts: online ridge regression (RR) and early wake-up (EW) algorithm. The online RR is used as a prediction model to predict the future core temperatures at run-time every predefined time called *control period*. Meanwhile, the EW algorithm is used to predict the likelihood of task migration in the next control period based on the predicted cores' temperatures. The proposed PEW sets the dark core power state to the one with a lower wake-up latency if task migration is expected to be used in the next control period. This reduces the time for the cores to start executing the tasks, which collectively improves the many-core system's overall performance. In summary, the contributions of this paper are as follows:

- This paper presents the PEW technique to reduce the dark cores' wake-up latency impact during task migration.
- The online RR is used as a prediction model to predict cores' temperatures in the next control period.
- The EW algorithm is used to put the dark cores in a power state with low wake-up latency based on the predicted temperatures.
- A comprehensive study using compute- and memory-intensive real-world applications has been conducted to validate the proposed PEW technique.

The remainder of this paper is structured as follows. Related works are discussed in Section II. The system model and problem definition are presented in Section III. The methodology of the proposed work is described in Section IV, while the performance of the proposed work is evaluated

in Section V. Finally, the conclusion and future work are presented in Section VI.

II. RELATED WORK

The increased power densities in many-core systems due to technology node shrinking has resulted in the so-called dark silicon problem. The dark silicon problem limits performance gain from using all available cores in a many-core system [22]. This problem has received a lot of attention in recent years as a significant many-core systems issue that requires careful attention. Many techniques for optimizing the performance of dark silicon many-core systems have been proposed in recent years. These performance optimization techniques can be categorized into performance optimization under the power constraint and performance optimization under the thermal constraint.

The power constraints techniques use thermal design power (TDP), which is a fixed per-chip power budget [5], [8]–[10] or thermal safe power (TSP), which is a fixed per-core power budget [6], [7] to avoid thermal violations. However, the use of the power budget can cause chip thermal violations since transient temperature and heat transfer between cores are excluded [23]. In contrast, the thermal constraint techniques consider the transient temperatures and heat transfer between the cores to prevent chip thermal violations. In thermal constraint techniques, task migration is one of the DTM techniques often used to balance the chip's thermal and prevent thermal violations at run-time. However, migrating the task to a dark core imposes an overhead due to the dark core wake-up latency. As our proposed work focuses on reducing the dark cores wake-up latency due to the task migration, the following paragraphs present related works that used task migration to maximize performance for dark silicon many-core systems.

To improve dark silicon many-core systems performance, some techniques use task migration and application mapping. Shafique *et al.* [17] introduced DaSiM, a variability-aware management technique for dark silicon many-core systems. DaSiM models the variations of core-to-core leakage power. It uses thread mapping and dark silicon patterning to activate or boost more cores by reducing the maximum temperature. DaSiM provides a lightweight prediction technique to predict the thermal distribution of a certain mapping and patterning solution at run-time. To handle thermal violation, DaSiM uses power-gating or task migration.

Some studies used a combination of DVFS and task migration for maximizing the dark silicon many-core performance. Hanumaiah *et al.* [24] proposed a run-time scheduling technique to improve many-core system performance. This technique uses task migration to allocate tasks to cores at run-time. During the first period of the task migration, it sets the DVFS levels of cores to a maximum level that does not violate the safe chip temperature. In a similar work, Wang *et al.* [9], [25] introduced a run-time thermal management technique to improve many-core system performance.

Based on model predictive control (MPC) decisions, this technique use task migration to balance the chip's thermal by migrating tasks between active cores. DVFS is used instead if task migration cannot be used.

The aforementioned techniques avoid task migration to dark cores due to the high wake-up latency of dark cores. However, migrating tasks among active cores may increase the migration overhead. For example, if two active cores exchange the tasks between them to balance the temperature, the task migration overhead will be twice. In contrast, migrating tasks from active to dark cores reduces this overhead by half. Some studies used dark cores to migrate the tasks. Studies in Ref. [26]–[29] used a virtual task migration to pattern the active and dark cores for optimizing the communication and computation performance of dark silicon many-core systems. These techniques move the location of dark cores and not the actual tasks. Dark cores are used as bubbles to distribute the active cores' heat. In Ref. [19], a technique for optimizing dark silicon many-core systems called DTaPO was introduced. DTaPO uses task migration to swap the tasks between the active and dark cores to maintain high overall system performance and keep the many-core system temperature within a safe thermal operating range. However, all these studies did not provide a solution for the issue of wake-up latency of dark cores due to task migration.

A scheduling technique to optimize system performance under thermal constraint by reducing the wake-up time needed for the task migration was proposed by Bashir *et al.* [20]. Based on offline thermal results, the proposed technique estimates the time needed to reach the threshold temperature to put the sleeping cores in the idle mode before performing task migration. However, this technique is not suitable for uncharacterized applications. In another work, Bashir *et al.* [21] proposed an improved technique suitable for run-time performance optimization. In this technique, the temperature is sensed at run-time, and task migration is to move the tasks to dark cores to address the thermal violation. These works use early switching the dark cores to idle mode and depend on a fixed early wake-up threshold. Although the cores in idle mode can run the upcoming tasks immediately, early switching to idle mode may cause more performance degradation due to more frequent DTM calls. Moreover, using a fixed wake-up threshold may not be suitable for applications that have high thermal fluctuation.

This paper provides a solution for dark cores wake-up latency overhead during task migration by proposing a prediction-based early wake-up (PEW) technique. Instead of using a fixed wake-up threshold, the proposed technique uses a prediction model to determine when to wake up the dark cores. An online sliding window-based ridge regression (RR) is used as the prediction model. If task migration is expected to be used in the next control period, the early wake-up (EW) algorithm uses the core's power states to put the dark cores in a power state with low wake-up latency ($\sim 10 \mu\text{s}$). Thus, it reduces the time for the dark cores to start

running the tasks to improve the many-core system overall performance.

III. SYSTEM OVERVIEW AND PROBLEM DEFINITION

This section presents the dark silicon many-core system model, a background on core power states, as well as problem definition and formulation.

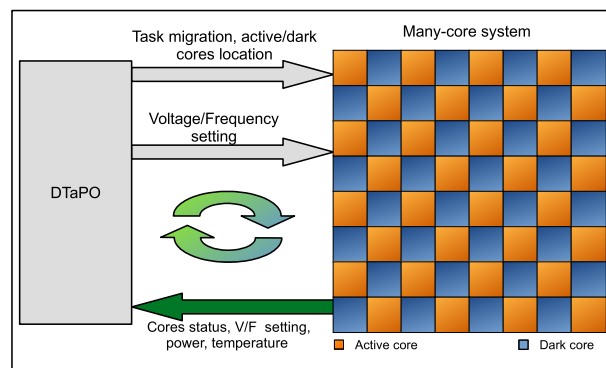


FIGURE 1. An overview of the system model.

A. SYSTEM MODEL

The system model is presented in Fig. 1. The many-core system consists of 64 homogeneous cores. The many-core system supports preemptable tasks so that a task can be stopped and moved to another core to continue the execution. As this study targets the dark silicon many-core system, we assume that only half of the cores can be activated simultaneously. The active and dark cores were patterned like a chessboard so that dark cores surround each active core for better heat dissipation [23]. Despite that the chessboard pattern adds one hop for each active core to the communication latency, it has a low peak chip temperature compared to the contiguous pattern [19].

DTaPO [19] is used to continuously tracks the many-core system status. Specifically, it monitors the active and dark cores' locations, voltage/frequency level, power, and transient temperature. DTaPO swaps the active and dark cores locations using the task migration to manage the thermal violation. In case no thermal headroom is available, it reduces the voltage/frequency level using the DVFS. For more details about DTaPO, refer to Ref. [19].

B. CORE C-STATES

Modern many-core processors are designed to support a set of low-power states called C-states [30] to reduce power consumption. C-states are designated by the letters C0, C1, C2, ..., Cn, where the processor's designer decides the value of n . The active state is C0, in which the core is in active mode. As the C-state progresses, further power-saving steps are taken, such as turning off more core components such as caches.

According to the ACPI standard [30], as shown in Fig. 2, the C1 state lowers the core voltage and turns off the core's

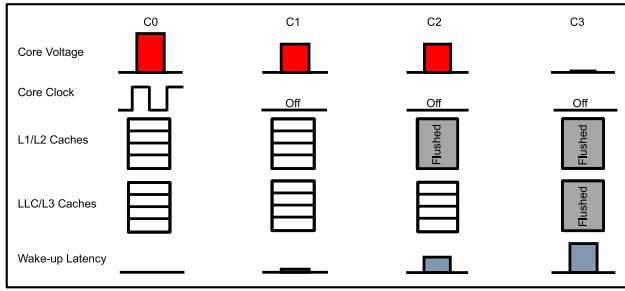


FIGURE 2. Core power states.

clock while preserving the L1/L2 cache contents. In the C2 state, the L1/L2 cache contents are flushed to the last level cache (LLC) cache. The core is completely dark or off in the C3 state. However, turning off more core components will increase the cores' time to return to a fully operational state (C0). The proposed technique assumes that our many-core system supports the C0, C1, and C3 power states.

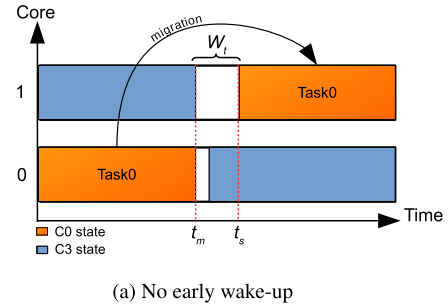
C. PROBLEM DEFINITION

Migrating tasks to dark cores causes performance degradation due to the substantial wake-up latency of the dark core. Fig. 3 illustrates that migrated tasks should wait until the dark cores are ready to execute them. Fig. 3a shows that when the dark core was in C3 state (dark state), the task migrated at time t_m should wait until the starting time t_s . Thus, reducing the task waiting time $W_t = t_s - t_m$ improves the overall system performance. The proposed PEW technique aims to reduce t_s by putting the dark cores in a power state with low wake-up latency, i.e., C1 state, just before the task migration at t_m . Thus, the dark core will start executing the migrated task earlier, as shown in Fig. 3b. This minimizes the W_t of the migrated task and improves the overall performance of a many-core system. Our aim can be mathematically expressed as follows:

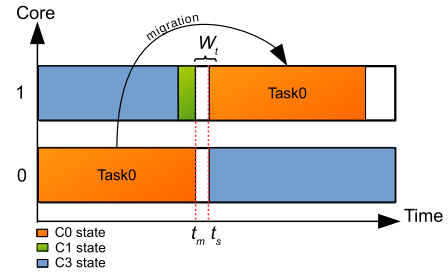
$$\text{Minimize } W_t : |t_s - t_m| \rightarrow 0 \tag{1}$$

IV. PROPOSED TECHNIQUE: PEW

The proposed PEW consists of a prediction model and early wake-up (EW) algorithm. Fig. 4 shows how the proposed technique is integrated into the system model. The proposed PEW uses ridge regression (RR) as a prediction model to predict the core's temperature. The prediction model uses the current core's status (i.e., active/dark), power, and thermal to predict the core's temperature in the next control period. Based on the predicted temperatures, the proposed EW algorithm predicts whether there will be a migration in the next control period. If migration is predicted in the next control period, it will put the dark cores in a power state with a low wake-up latency. Thus, it reduces the waiting time for the dark cores to be ready for new coming tasks and improves the overall performance. On the other hand, if it predicts no



(a) No early wake-up



(b) Early wake-up

FIGURE 3. Illustration of task waiting time when no early wake-up is used and when early wake-up is used.

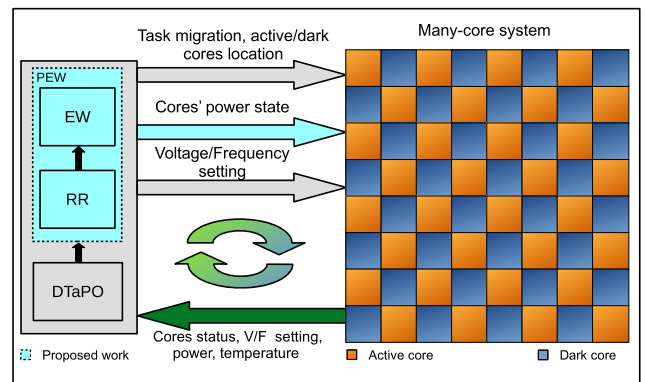


FIGURE 4. Integrating the proposed PEW into the system model.

migration in the next control period, it will leave the dark cores in a power state that saves power.

A. PREDICTION MODEL

Linear regression is one of the most widely used techniques for predictive modeling. It tries to find a linear relationship between the inputs (independent variables) and the output (dependent variable) according to the following formula:

$$Y = X\beta + \epsilon \tag{2}$$

where Y is the dependent variable and X is an $n \times p$ matrix representing the independent variables, where n is the number of samples and p is the number of features. Vector β represents the regression coefficients. Vector ϵ represents the random errors, which are the residuals that are not explained by the $X\beta$ term. In this work, a type of linear regression called ridge regression [31] is used as a prediction model. The linear

prediction model is used because the changes in the transient temperature are linear in the short control period < 1 ms.

1) RIDGE REGRESSION

Ridge regression (RR) [31] is a type of multiple linear regression represented by Eq. (2). It is used when there is a correlation between the independent variables called the multicollinearity problem [32]. It adds a penalty of a squared magnitude of the coefficients to the loss function to overcome the multicollinearity problem:

$$\min_{\beta} \{(Y - X\beta)^T(Y - X\beta) + \lambda \|\beta\|^2\} \quad (3)$$

where $\|\beta\|^2 = \sum_{j=1}^p \beta_j^2$ is the penalty term, and λ is the regularization parameter that represents the penalty control. Ridge regression becomes ordinary linear regression when $\lambda \rightarrow 0$.

Ridge regression was chosen in our work because it fits our regression problem, where there is a correlation between the independent variables, i.e., the current temperature, power, and cores status (i.e., active/dark). Lasso regression may also be used to eliminate the collinearity in a large number of independent variables by selecting a subset of them. However, it may eliminate some important collinear variables that may affect the prediction accuracy, especially when the number of independents variable is small, as in our prediction system model.

2) ONLINE RIDGE REGRESSION

The ridge regression uses all available data samples to make an accurate prediction. However, using all data samples is computationally intensive and infeasible for online prediction where the ridge regression has $\mathcal{O}(np^2)$ time complexity [33]. In highly fluctuating input data such as the core's temperature, the old data samples may be worthless. Therefore, considering only the last data samples using a sliding window will reduce the time complexity since:

$$n = \min(m, w) \quad (4)$$

where m represents all data samples, and w is the sliding window size. The sliding window starts to move when the data samples are larger than the window size. The time complexity now depends on w and p . This is suitable for our online system model because it has only three independent variables (p) and w is small.

Algorithm 1 shows the pseudo-code of the online ridge regression. It receives the transient temperature (T), power (P), and status (active or dark) (S) of all cores k , where T , P , and S are $1 \times k$ vectors. These vectors are stored in $Temp$, Pow , and $Stat$ buffers (lines 2-4). The length of each buffer represents all samples m (line 5). The samples that are used in regression fitting n are determined based on the window size. If all input samples m are still less than the window size, all samples are used as an input to ridge regression. Otherwise, only the last window samples are used (lines 6). The n samples are used to fit the ridge

regression model (line 7). The fitted model is used the current temperature, power, and core status to predict the future cores temperature (T_p) (line 8). The predicted temperatures are used by the EW algorithm (line 9).

Algorithm 1 Online Ridge Regression

Input: w, T, P , and S
Output: T_p

```

1 while true do
2   Temp ← T
3   Pow ← P
4   Stat ← S
5   m = length(Temp)
6   n = min(m, w)
7   RR ← fit(Tempn×k, Pown×k, Statn×k)
8   Tp ← RR(T, P, S)
9   EW ← Tp
10 end

```

TABLE 1. Definition of the EW algorithm's symbols.

Symbol	Definition
Λ	A set of all active cores
Γ	A set of all dark cores
T_p	A set of predicted transient temperature of all cores
H	A set of all tasks exceeded the threshold temperature
t_i	i^{th} task, $t_i \in H$
a_i	i^{th} active core, $a_i \in \Lambda$
d_i	i^{th} dark core, $d_i \in \Gamma$
T_{th}	Threshold temperature
θ	Safe margin value

B. PROPOSED PREDICTION-BASED EARLY WAKE-UP

This subsection presents the second component of the proposed PEW, which is the EW algorithm. All symbols used in our proposed EW algorithm are defined in Table 1. Algorithm 2 describes the proposed EW algorithm in detail. The proposed algorithm receives the predicted temperatures T_p from the prediction model. Also, it receives the set of all active cores $\Lambda = \{a_0, \dots, a_{k-1}\}$, the set of all dark cores $\Gamma = \{d_0, \dots, d_{k-1}\}$, the threshold temperature T_{th} and the safe margin θ .

In each control period, the proposed algorithm reads the predicted temperature of each active core $T_p[a_i]$. If the predicted temperature of the active core is higher than the threshold temperature, it is most likely that DTaPO will do either task migration or DVFS in the next control period. Thus, the proposed EW algorithm reads the predicted temperature of a destination dark core $T_p[d_i]$. If the predicted temperature of the destination dark core is lower than the threshold temperature by θ or the temperature of dark cores is lower than the active core by θ (this condition statement is the conditions statement used by DTaPO [19] to do task migration), it sets

$H[t_i] = 1$ to indicate that the task exceeded the threshold temperature is a movable task (lines 4-5). Otherwise, it marks the task that exceeded the threshold temperature as a non-movable task by setting $H[t_i] = 0$ (line 8). If all the tasks in H are movable, the proposed algorithm puts the dark cores in the C1 power states to reduce the wake-up latency of the dark cores (lines 12-14). Otherwise, it will leave the dark cores in the C3 power state to save more power (lines 15-17).

Algorithm 2 Early Wake-up (EW) of Dark Cores

```

Input:  $\Lambda, \Gamma, T_p, T_{th}$ , and  $\theta$ 
Output: Power states of the dark cores
1 while true do
2   for  $\forall a_i \in \Lambda$  do
3     if  $T_p[a_i] > T_{th}$  then
4       if  $(T_p[d_i] < T_{th} - \theta) \parallel (T_p[d_i] < T_p[a_i] - \theta)$ 
5         then
6            $H[t_i] = 1$ 
7         end
8       else
9          $H[t_i] = 0$ 
10      end
11    end
12  if  $H[t_i] = 1 \forall t_i \in H$  then
13    | Put all darks cores in C1 state
14  end
15  else
16    | Leave all dark cores in C3 state
17  end
18 end

```

C. COMPLEXITY ANALYSIS

The time complexity of ridge regression is $\mathcal{O}(np^2)$ [33]. The proposed PEW uses sliding ridge regression, where $p = 3$ is kept constant, whereas n is a small sample represented by window size (w). Therefore, time complexity in our online ridge regression is $\mathcal{O}(w)$. For the EW algorithm, it needs to check the predicted temperature of all cores. Therefore, the time complexity depends on the number of cores (k). Thus, the time complexity is $\mathcal{O}(k)$.

The space complexity of ridge regression is $\mathcal{O}(wp + w)$. As ridge regression needs to store matrices X and Y as $w \times p$ and $n \times 1$ matrix, respectively, to find β according to Eq.(3). Therefore, the online ridge regression space complexity is also $\mathcal{O}(w)$. For the space complexity of EW, the indices of active and dark cores and a $1 \times k$ vector of predicted temperature needs to be stored. Therefore, the space complexity of the EW algorithm is $\mathcal{O}(k)$.

The overall complexity of the proposed PEW is the summation of online ridge regression and EW algorithm complexities that are computed one after another. Hence, the time

and space complexity of the proposed PEW technique are linear.

V. EXPERIMENTAL EVALUATION

Many experiments were conducted to evaluate our proposed work. The following subsections show how the experiments are set up, the comparison results, and the discussion of the comparison results.

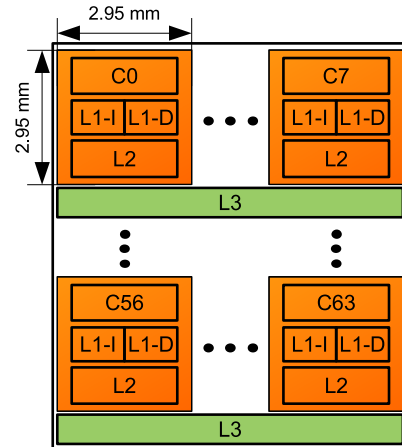


FIGURE 5. The simulated many-core system floorplan.

A. EXPERIMENTAL SETUP

Our proposed work was evaluated on a many-core system that consists of 64-core, where 32-core are active cores, and 32-core are dark cores. These cores are connected using an 8×8 mesh network-on-chip (NoC). All the cores share the same instruction set architecture (ISA) (homogeneous microarchitecture) and can operate at different frequencies (heterogeneous frequency). Every core can run at a maximum frequency of 4 GHz. The floorplan of the simulated system is shown in Fig. 5. Every core has a size of $2.95 \text{ mm} \times 2.95 \text{ mm}$ according to McPAT [34] modeling for 22-nm technology. Every core has a 32 KB private L1 data cache, 32 KB private L1 instruction cache, and 512 KB private L2 cache. All eight cores share an 8 MB L3 cache. Table 2 shows the summary of the system setup.

TABLE 2. Summary of the system setup.

Parameter	Value
Number of cores	64
Maximum frequency	4 GHz
L1-I	32 KB
L1-D	32 KB
L2	512 KB
L3	8 MB/8 cores
Technology	22 nm
Connection type	8×8 mesh NoC

Fig. 6 shows the experimental setup of the proposed work. LifeSim simulation tool [35] for many-core systems was

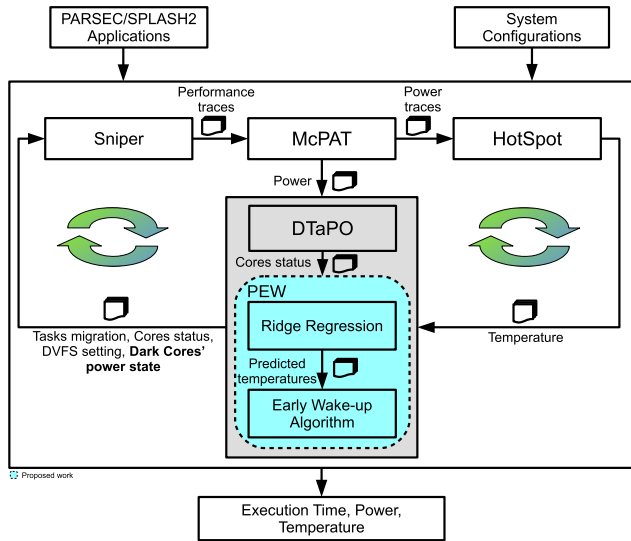


FIGURE 6. Experimental setup of the proposed technique.

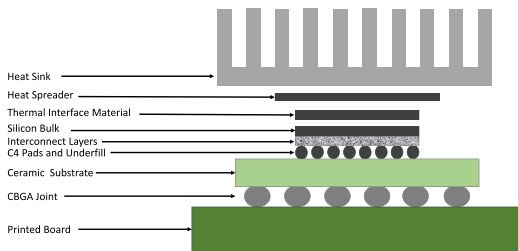


FIGURE 7. Package layers of a ceramic ball grid array (CBGA)(adapted from Fig. 2 in Ref. [38]).

used. LifeSim is a tool that integrates Sniper [36] with HotSpot [37] thermal simulator. Sniper is an architectural x86-64 many-core simulator (including the power framework McPAT). It is faster than cycle-accurate simulations with a 25% average performance error compared to actual hardware. McPAT is commonly used for modeling integrated power, area, and timing because it provides comprehensive design space exploration for multi/many-core processor configurations. HotSpot is the most widely used thermal simulator. It is built on the widely used stacked-layer packaging scheme used in modern very-large-scale integration (VLSI) systems, as shown in Fig. 7.

As Sniper does not support core power-gating, Sniper’s scheduler was modified to assign the tasks to the active cores only using the core mask pattern. Also, McPAT was modified to estimate only the power of caches and memory management unit for the C1 power state as only the caches are active in the C1 state. The power of the dark state (C3 state) is not considered. The wake-up latency for C1 and C3 are assumed to be 10 μ s and 200 μ s, respectively. These values are chosen according to Linux’s intel_idle driver for Nehalem microarchitecture as our simulation’s cores are Nehalem-based. Thus, only 10 μ s is added to the execution time for every migration that was predicted correctly, i.e., wake-up

TABLE 3. HotSpot thermal configuration.

Parameter	Value
Thickness of the chip	0.15 mm
Silicon thermal conductivity	100 W/(m·K)
Silicon specific heat	1.75×10^6 J/(m ³ ·K)
Heat sink convection capacitance	140.4 J/K
Heat sink convection resistance	0.1 K/W
Heat spreader thermal conductivity	400 W/(m·K)
Heat sink spreader specific heat	3.55×10^6 J/(m ³ ·K)
Interface material thickness	20 μ m
Interface material thermal conductivity	4 W/(m·K)
Interface material specific heat	4×10^6 J/(m ³ ·K)
Ambient temperature	318.15 K

m: Meter, W: Watt, J: Joule, K: Kelvin.

from C1 state. On the other hand, 200 μ s is added to the execution time for every migration that was mispredicted, i.e., wake-up from C3 state.

System configurations, such as the number of cores, floor-plan, caches, are used to configure the simulated system. To generate performance traces, Sniper runs applications from the SPLASH-2 [39] and PARSEC [40] benchmark suites. These traces are used by McPAT to estimate each core’s power consumption. HotSpot estimates the transient temperature using the estimated power traces. The HotSpot configuration parameters are listed in Table 3. In each control period, DTaPO is used to schedule the tasks and do thermal management based on the transient temperature generated from the Hotspot. To predict the temperature of each core in the next control period, the ridge regression uses the transient temperature generated from HotSpot, the power generated from McPAT, and the core states from DTaPO. Based on the predicted temperature, the early wake-up algorithm decides whether to wake up the dark cores.

In our experiment, compute- and memory-intensive applications from SPLASH-2 and PARSEC benchmark suites are used to evaluate the efficiency of the proposed technique. High-temperature tasks in compute-intensive applications can rapidly increase the core temperature, making them good candidates for validating our proposed algorithm. In contrast, memory-intensive applications have a large number of memory accesses, showing the task migration overhead due to cache misses. The experimental evaluation was done in two phases; preliminary study and comprehensive study. In both studies, the value of θ is set to 5% of the threshold temperature, and the control period interval length is set to 1 ms. To eliminate the experiment results’ randomness, the results reported in this study are the average results of conducting the experiment ten times.

A preliminary study was carried out by executing a mix of four 8-thread applications: *Bodytrack*, *Ocean*, *Radix*, and *Blackscholes*. The threshold temperature was set to 70 $^{\circ}$ C. The threshold temperature was chosen based on the temperature profile of the studied application on the target platform. There may be no migration when threshold temperature is

too high. Also, there may not be any cold cores to migrate to when threshold temperature is too low. For more details on the impact of threshold temperature on task migration, please refer to Ref. [19].

The preliminary study was carried out to determine the best-fixed threshold for fixed threshold early wake-up (FEW) [21] technique and evaluate the performance of the proposed PEW technique under various prediction accuracy scenarios. In this preliminary study, a pre-known future temperature generated from the Hotspot simulator was used as an input to the proposed EW algorithm. These pre-known future temperatures represent the prediction model with 100% accuracy, which is the best-case scenario. In the other scenarios, this accuracy was reduced 10% each time by introducing a uniformly distributed random error.

In the second phase, a comprehensive study was carried out by running eight 32-thread applications: *Fluidanimate*, *Bodytrack*, *Cholesky*, *Blackscholes*, *Raytrace*, *FFT*, *Ocean*, and *Swaptions* individually. The threshold temperature was lowered to 65 °C to show the efficiency of the proposed work. The comprehensive study used RR as a prediction model to predict the future temperature. This predicted temperature was used as input to the proposed EW algorithm. There is a trade-off between the prediction accuracy and the window size. The bigger the window size, the better prediction accuracy. However, the prediction overhead will increase as the window size increase. Therefore, the value of window size (w) is set to 30, which gives a good prediction accuracy and low prediction overhead.

The value of the regularization parameter (λ) for RR is set to 0.2. This value was empirically determined by conducting experiments and choosing the value with a low average mean absolute error (MAE). Table 4 shows the MAE for the studied applications at different regularization parameter (λ) values. It can be seen that when $\lambda = 0.2$, the average MAE for all the studied applications is the lowest. In the comprehensive study, the computation efficiency in terms of completion time, power efficiency in terms of million instructions per second/Watt (MIPS/W), and average temperature were reported.

TABLE 4. The MAE for the studied applications at different values of λ .

Benchmark	Regularization parameter (λ)				
	0.1	0.2	0.3	0.4	0.5
Fluidanimate	2.46	2.46	2.46	2.47	2.47
Bodytrack	3.99	3.89	3.86	3.86	3.86
Cholesky	2.19	2.17	2.20	2.23	2.26
Blackscholes	1.18	1.19	1.20	1.20	1.21
Raytrace	0.90	0.90	0.91	0.91	0.92
FFT	0.67	0.67	0.67	0.67	0.68
Ocean	2.84	2.85	2.87	2.90	2.92
Swaptions	1.79	1.74	1.73	1.73	1.73
Average	2.00	1.98	1.99	2.00	2.00

B. COMPARATIVE RESULTS AND ANALYSIS

The proposed prediction-based early wake-up (PEW) was compared with our previous work that uses a non-early

wake-up technique (NoEW) [19] and with the state-of-the-art fixed threshold early wake-up (FEW) [21] that uses a fixed threshold to wake up the dark cores. Moreover, for a fair comparison, the dark cores are switched to a C1 power state instead of an idle state in FEW.

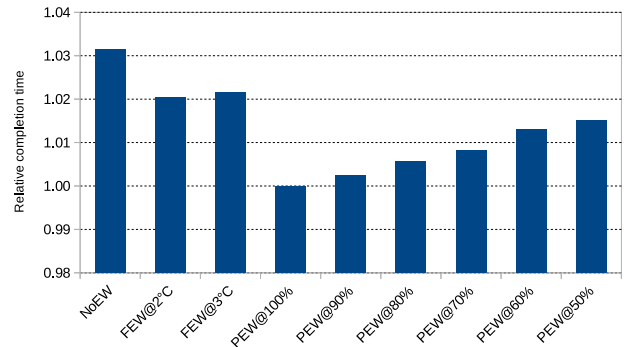


FIGURE 8. Relative completion time of the proposed PEW at different accuracy level, FEW at the different wake-up threshold, and NoEW.

1) PRELIMINARY RESULTS

The results from the preliminary study are shown in Fig. 8. These results are the relative completion time of executing the mix of four multi-threaded applications: *Bodytrack*, *Ocean*, *Radix*, and *Blackscholes*. The completion time is plotted relative to the proposed PEW with 100% prediction accuracy (the best case). The proposed algorithm was evaluated using different accuracy levels, starting from 100% to 50%. FEW was also evaluated using two fixed early wake-up thresholds, 2 °C (FEW@2 °C) and 3 °C (FEW@3 °C) under the temperature threshold. For the studied application, it is obvious that the proposed techniques outperform the NoEW by 3.1% and 1.5% at 100% and 50% accuracy, respectively. Thus, even at low prediction accuracy, using prediction-based early weak-up still performs better than without early wake-up. Moreover, using prediction-based early wake-up at 100% accuracy outperforms the fixed early wake-up threshold FEW@2 °C and FEW@3 °C by 2% and 2.2%, respectively. In addition, FEW@2 °C reduces the completion time by 0.2% compared to FEW@3 °C. Thus, in the comprehensive study, the fixed early wake-up threshold in FEW was set to 2 °C below the threshold temperature.

2) COMPREHENSIVE RESULTS

In the comprehensive study, RR is used as a prediction model. Table 5 illustrates the average number of task migrations and the percentage of wake-up accuracy (i.e., the percentage of task migration predicted accurately using the EW algorithm). It also shows the RR prediction model accuracy in terms of MAE and root mean square error (RMSE) for the cores temperatures. It is obvious that using the prediction model gives better wake-up accuracy than using a fixed wake-up threshold. On average, the proposed PEW predicts 91.42% of the task migration accurately compare to 76.62% using a fixed wake-up threshold. Fig. 9 shows the actual and predicted

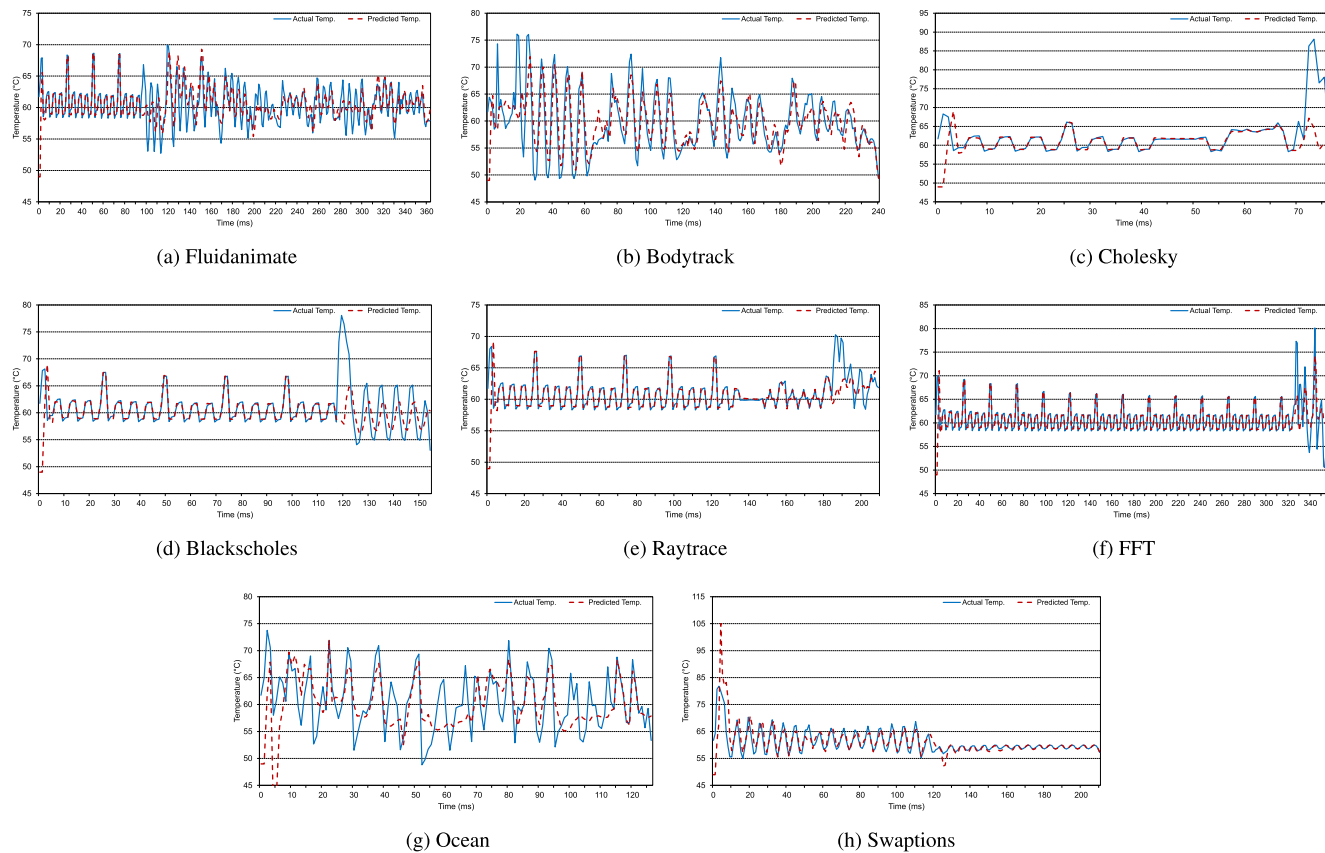


FIGURE 9. Actual and predicted temperature of core 0 for all studied applications.

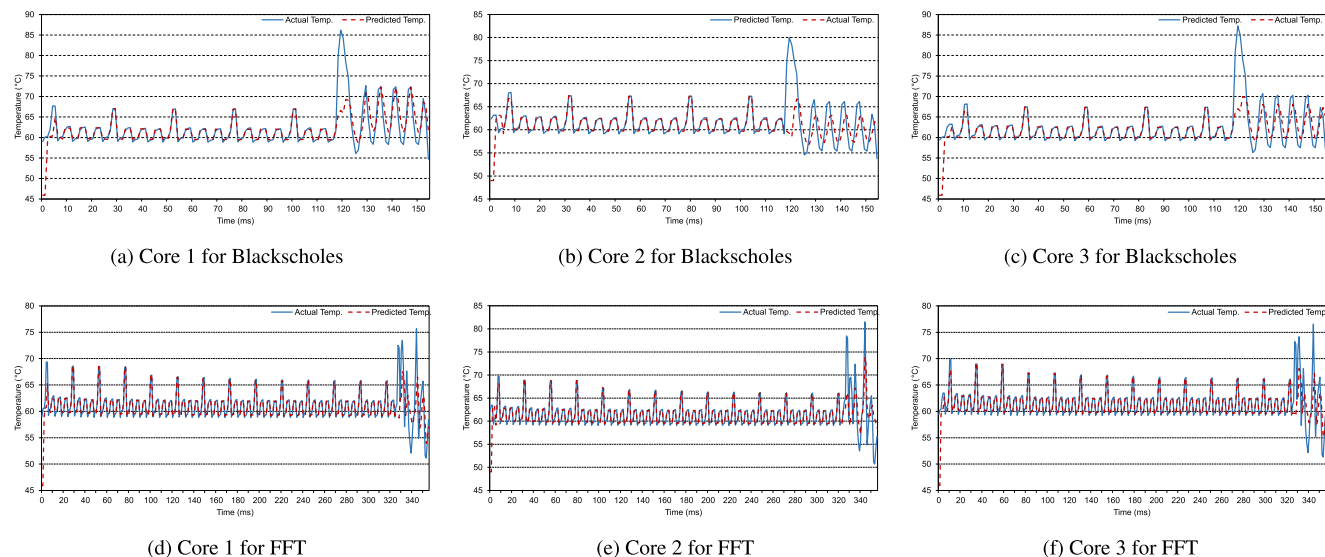


FIGURE 10. Actual and predicted temperature of cores 1-3 for Blackscholes (a-c) and FFT (d-f) application.

temperature of core 0 for all studied applications. The results of all cores are not presented because they show similar trend as shown in Fig. 10, which shows the results of three cores (core 1-3) for *Blackscholes* and *FFT*. Although the prediction

model does not fit well for some applications, the prediction model can predict well when the temperature exceeded the threshold temperature (65 °C), which is important for the EW algorithm to make the early wake-up decision.

TABLE 5. Average number of tasks migration and wake-up accuracy.

Benchmark	Avg no. of task migration		Wake-up accuracy (%)		Regression accuracy (°C)	
	PEW	FEW	PEW	FEW	MAE	RMSE
Fluidanimate	48.90	50.40	95.91	83.73	2.46	11.11
Bodytrack	17.30	20.00	65.90	55.00	3.86	32.95
Cholesky	18.00	20.00	88.89	100.00	2.17	41.24
Blackscholes	39.00	39.00	94.87	100.00	1.30	11.12
Raytrace	61.10	61.60	96.73	95.29	0.90	6.10
FFT	111.00	111.60	99.01	98.03	0.68	4.42
Ocean	13.10	12.80	98.47	46.09	2.85	14.33
Swaptions	29.60	29.00	91.55	34.83	2.28	8.69
Average	42.25	43.05	91.42	76.62	1.86	14.48

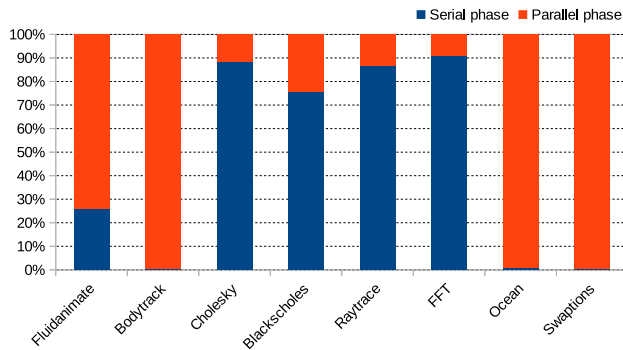
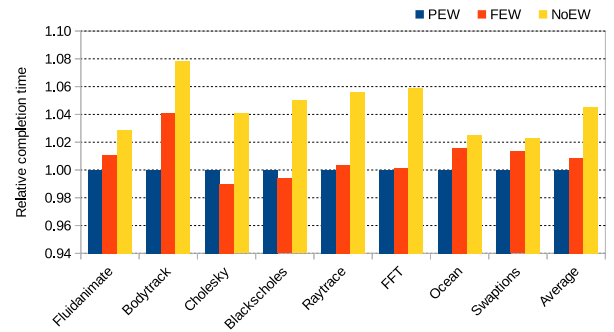


FIGURE 11. The ratio of serial and parallel phases of the studied application.

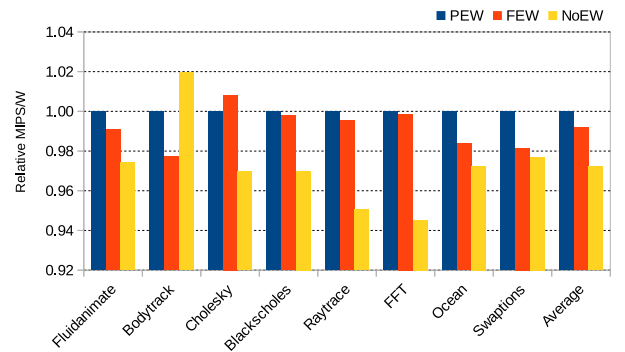
For the *Cholesky* and *Blackscholes* applications, using a fixed wake-up threshold resulted in a higher wake-up accuracy than the prediction model. This is because these applications have a large percentage of serial phase, as shown in Fig. 11. For more characteristics of these applications, refer to Ref. [41]. In the serial phase, these applications run only one cool thread with a small number of task migrations so that our prediction model cannot fit well. Although *Raytrace* and *FFT* also have a large percentage of serial phase, the serial phase of these applications has a high number of task migrations. Thus, our prediction model fitted well with these applications.

The comparison results of the computational and power efficiency shown in Fig. 12 are relative to the proposed PEW technique. These results show the comparative results when executing the nine multi-threaded applications individually with the proposed PEW, NoEW, and FEW techniques. The efficiency of computation in terms of relative completion time is shown in Fig. 12a. In all the studied applications, the proposed prediction-based early wake-up (PEW) reduces the completion time by 4.2% on average over NoEW. Specifically, it reduces the completion time by 2.9% for *Fluidanimate*, 7.9% for *Bodytrack*, 4.1% for *Cholesky*, 5% for *Blackscholes*, 5.6% for *Raytrace*, 5.9% for *FFT*, 2.5% for *Ocean*, and 2.3% for *Swaptions*.

Moreover, the proposed PEW reduces the completion time in most studied applications by 0.8% on average over FEW. Specifically, it shortens the task completion time by 1.1% for *Fluidanimate*, 4.1% for *Bodytrack*, 0.4% for *Raytrace*, 0.1% for *FFT*, 1.6% for *Ocean*, and 1.3% for *Swaptions*. FEW reduces the completion time over the proposed PEW by 1%



(a) Completion time



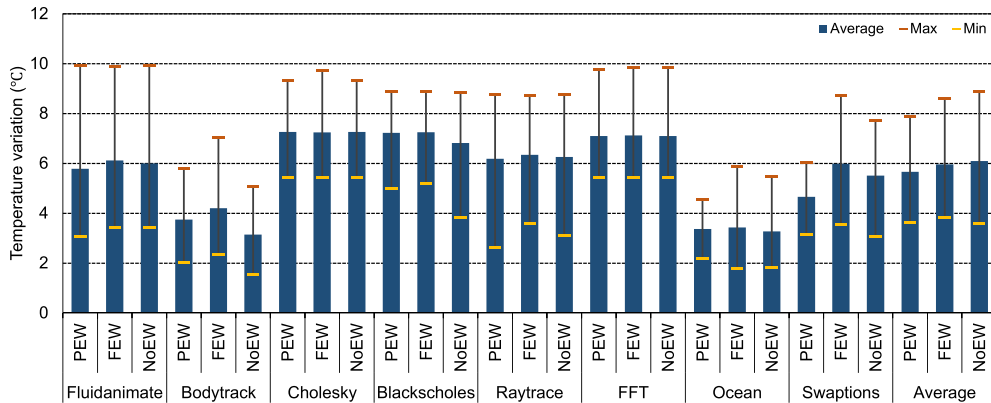
(b) MIPS/Watt

FIGURE 12. Relative performance efficiency in terms of completion time and MIPS/Watt.

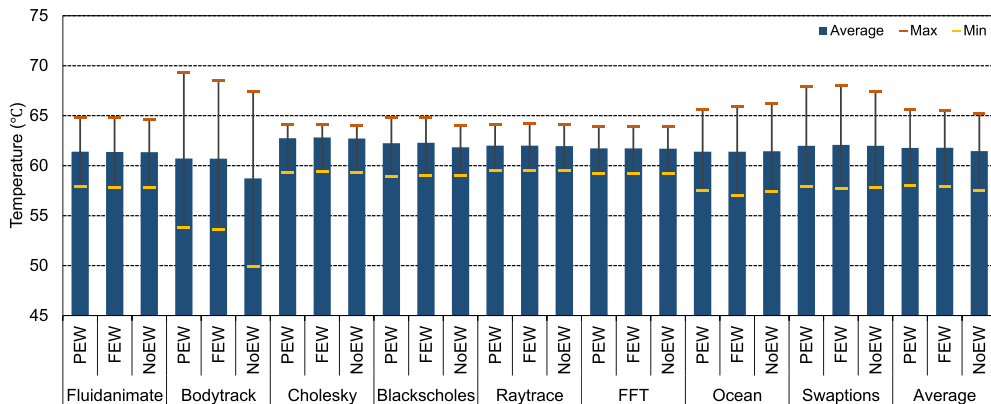
for *Cholesky* and 0.6% for *Blackscholes* because these applications have a large percentage of serial phase, as mentioned previously. In general, the overall completion time of the studied application is improved because the waiting time (W_l) of the tasks is reduced. It is worth mentioning that all the comparison results are based on assuming the wake-up latency for dark state 200 μs according to Linux's intel_driver. However, if the wake-up latency for the dark state is longer like in the LEAT processor (261.77 ms), the improvement is expected to be much better.

The comparative results of the power efficiency in terms of relative MIPS/Watt are shown in Fig. 12b. On average, our proposed PEW performs better than NoEW and FEW by 3% and 1%, respectively. In all the studied applications except for *Bodytrack* and *Cholesky*, our proposed PEW increased the MIPS/Watt by up to 5.5% and 2.3% over NoEW and FEW, respectively. The lower MIPS/Watt in *Bodytrack* and *Cholesky* is due to the high prediction RMSE for these applications, as shown in Table 5.

The thermal efficiency was also evaluated, as shown in Fig. 13. Fig. 13a shows the average, max, and min of the variation between the coldest and the hottest core. Our proposed PEW, on average, exhibits less temperature variation than FEW and NoEW. On the other hand, Fig. 13b shows the average, max, and min of the cores' transient temperature. It can be noted that the average temperatures for the three



(a) Temperature variation between the coldest and hottest core



(b) Transient temperature of all cores

FIGURE 13. The average, max, and min of temperature variation between the coldest and hottest core and transient temperature of all cores.

techniques are identical as all these techniques use identical thermal management.

3) SIGNIFICANCE TEST

A significance test (t-test) was performed to verify the significance of the performance improvement in terms of task completion time. A paired t-test was conducted for the completion time of our proposed PEW against FEW and NoEW. The significant level (α) is set to the standard value of 0.05. The null hypothesis H_0 is tested against the alternative hypothesis H_a . H_0 assumes that the improvement is not significant, and H_a assumes that the improvement is significant. The null hypothesis H_0 is rejected if $p\text{-value} < \alpha$. Table 6 shows the significance test for the proposed technique’s completion time against FEW and NoEW. It shows that the improvement when using our proposed PEW against FEW is statistically significant for most of the studied applications. The improvement is not significant for *Cholesky* and *Blackscholes* that suggests that the prediction model may need to be tuned to fit these applications, which is beyond the scope of this paper. The overall improvement when using our proposed PEW against NoEW is statistically significant for all studied applications.

TABLE 6. Significant test (t-test) for the proposed technique against FEW and NoEW.

Benchmark	FEW			NoEW		
	t-value	p-value	Significant?	t-value	p-value	Significant?
Fluidanimate	4.22	1.13E-03	Yes	13.82	1.15E-07	Yes
Bodytrack	3.41	3.89E-03	Yes	2.43	1.91E-02	Yes
Cholesky	-15.96	3.28E-08	No	52.97	7.66E-13	Yes
Blackscholes	-8.13	9.76E-06	No	20.27	4.03E-09	Yes
Raytrace	3.58	2.98E-03	Yes	47.03	2.22E-12	Yes
FFT	2.98	7.68E-03	Yes	164.26	2.92E-17	Yes
Ocean	2.84	9.78E-03	Yes	4.82	4.71E-04	Yes
Swaptions	7.16	2.65E-05	Yes	22.32	1.72E-09	Yes

VI. CONCLUSION

This paper proposes a prediction-based early wake-up (PEW) for the dark cores technique that utilizes an online sliding window-based ridge regression (RR) to reduce the dark cores wake-up latency during the task migration. RR predicts the future’s core temperatures based on the previous thermal, power, and core status. Based on these predicted temperatures, the proposed early wake-up (EW) algorithm puts the dark cores in a power state with low wake-up latency if task migration is expected in the next control period. Thus, our proposed PEW reduces the time for the dark cores to start

running the tasks, which improves the many-core system's overall performance. The comparison results show that using our proposed PEW reduces the task completion time by up to 7.9% and 4.1% compared to non-early wake-up (NoEW) and using a fixed threshold wake-up (FEW), respectively. It also shows that using our proposed PEW increases the MIPS/Watt by up to 5.5% and 2.3% over NoEW and FEW, respectively. Moreover, a significance test shows that our improvements are statistically significant for all studied applications except those that cannot fit well in our prediction model. For future work, we plan to propose a technique that dynamically tunes the prediction model parameters (window size and regularization parameter) according to the running application and to evaluate the impact of chip floorplan on the temperature.

REFERENCES

- [1] R. H. Dennard, F. H. Gaensslen, V. L. Rideout, E. Bassous, and A. R. LeBlanc, "Design of ion-implanted MOSFET's with very small physical dimensions," *IEEE J. Solid-State Circuits*, vol. JSSC-9, no. 5, pp. 256–268, Oct. 1974.
- [2] G. E. Moore, "Cramming more components onto integrated circuits," *Electronics*, vol. 38, no. 8, pp. 114–117, Apr. 1965.
- [3] H. Esmailzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," in *Proc. 38th Annu. Int. Symp. Comput. Archit. (ISCA)*, 2011, pp. 365–376.
- [4] J. Henkel, H. Khdr, S. Pagani, and M. Shafique, "New trends in dark silicon," in *Proc. 52nd ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Jun. 2015, pp. 1–6.
- [5] B. Raghunathan, Y. Turakhia, S. Garg, and D. Marculescu, "Cherry-picking: Exploiting process variations in dark-silicon homogeneous chip multi-processors," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2013, pp. 39–44.
- [6] S. Pagani, H. Khdr, W. Munawar, J.-J. Chen, M. Shafique, M. Li, and J. Henkel, "TSP: Thermal safe power: Efficient power budgeting for many-core systems in dark silicon," in *Proc. Int. Conf. Hardw./Softw. Codesign Syst. Synth.*, 2014, pp. 1–10.
- [7] 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 *Proc. 33rd IEEE Int. Conf. Comput. Design (ICCD)*, New York, NY, USA, Oct. 2015, pp. 573–580.
- [8] A. Rezaei, D. Zhao, M. Daneshmand, and H. Wu, "Shift sprinting: Fine-grained temperature-aware NoC-based MCMSoC architecture in dark silicon age," in *Proc. 53rd Annu. Design Autom. Conf.*, Jun. 2016, p. 155.
- [9] H. Wang, J. Ma, S. X.-D. Tan, C. Zhang, H. Tang, K. Huang, and Z. Zhang, "Hierarchical dynamic thermal management method for high-performance many-core microprocessors," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 22, no. 1, pp. 1–21, Dec. 2016.
- [10] M. Ansari, M. Salehi, S. Safari, A. Ejlali, and M. Shafique, "Peak-power-aware primary-backup technique for efficient fault-tolerance in multicore embedded systems," *IEEE Access*, vol. 8, pp. 142843–142857, 2020.
- [11] R. Devaraj, A. Sarkar, and S. Biswas, "Supervisory control approach and its symbolic computation for power-aware RT scheduling," *IEEE Trans. Ind. Informat.*, vol. 15, no. 2, pp. 787–799, Feb. 2019.
- [12] R. Devaraj and A. Sarkar, "Resource-optimal fault-tolerant scheduler design for task graphs using supervisory control," *IEEE Trans. Ind. Informat.*, vol. 17, no. 11, pp. 7325–7337, Nov. 2021.
- [13] J. Zhou, J. Sun, M. Zhang, and Y. Ma, "Dependable scheduling for real-time workflows on cyber-physical cloud systems," *IEEE Trans. Ind. Informat.*, vol. 17, no. 11, pp. 7820–7829, Nov. 2021.
- [14] J. Teich, B. Pourmohseni, O. Kesocze, J. Spieck, and S. Wildermann, "Run-time enforcement of non-functional application requirements in heterogeneous many-core systems," in *Proc. 25th Asia South Pacific Design Autom. Conf. (ASP-DAC)*, 2020, pp. 629–636.
- [15] A. Raghavan, Y. Luo, A. Chandawalla, M. Papaefthymiou, K. P. Pipe, T. F. Wensisch, and M. M. K. Martin, "Computational sprinting," in *Proc. IEEE Int. Symp. High-Perform. Comp. Archit.*, Feb. 2012, pp. 1–12.
- [16] H. Khdr, S. Pagani, M. Shafique, and J. Henkel, "Thermal constrained resource management for mixed ILP-TLP workloads in dark silicon chips," in *Proc. 52nd Annu. Design Autom. Conf.*, 2015, p. 179.
- [17] M. Shafique, D. Gnad, S. Garg, and J. Henkel, "Variability-aware dark silicon management in on-chip many-core systems," in *Proc. Design, Autom. Test Eur. Conf. Exhib.* San Jose, CA, USA: EDA Consortium, 2015, pp. 387–392.
- [18] M. S. Mohammed, A. K. Al-Dhamari, A. A.-H. ab Rahman, N. Paraman, A. A. M. Al-Kubati, and M. N. Marsono, "Temperature-aware task scheduling for dark silicon many-core system-on-chip," in *Proc. 8th Int. Conf. Modeling Simulation Appl. Optim.*, Apr. 2019, pp. 1–5.
- [19] 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, p. 1980, Nov. 2020.
- [20] Q. Bashir, M. N. Shehzad, M. N. Awais, U. Farooq, M. T. Hamayun, and I. Ali, "A scheduling based energy-aware core switching technique to avoid thermal threshold values in multi-core processing systems," *Microprocessors Microsyst.*, vol. 61, pp. 296–305, Sep. 2018.
- [21] Q. Bashir, M. N. Shehzad, M. N. Awais, S. Baig, M. G. Dogar, and A. Rashid, "An online temperature-aware scheduling technique to avoid thermal emergencies in multiprocessor systems," *Comput. Electr. Eng.*, vol. 70, pp. 83–98, Aug. 2018.
- [22] M. Shafique and S. Garg, "Computing in the dark silicon era: Current trends and research challenges," *IEEE Design Test*, vol. 34, no. 2, pp. 8–23, Apr. 2017.
- [23] M. Shafique, S. Garg, J. Henkel, and D. Marculescu, "The EDA challenges in the dark silicon era: Temperature, reliability, and variability perspectives," in *Proc. 51st Annu. Design Autom. Conf.*, 2014, pp. 1–6.
- [24] V. Hanumaiah, S. Vrudhula, and K. S. Chatha, "Performance optimal online DVFS and task migration techniques for thermally constrained multi-core processors," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 11, pp. 1677–1690, Nov. 2011.
- [25] H. Wang, M. Zhang, S. X.-D. Tan, C. Zhang, Y. Yuan, K. Huang, and Z. Zhang, "New power budgeting and thermal management scheme for multi-core systems in dark silicon," in *Proc. 29th IEEE Int. System-on-Chip Conf. (SOCC)*, Sep. 2016, pp. 344–349.
- [26] X. Wang, A. K. Singh, and S. Wen, "Exploiting dark cores for performance optimization via patterning for many-core chips in the dark silicon era," in *Proc. 12th IEEE/ACM Int. Symp. Netw. Chip (NOCS)*, Oct. 2018, pp. 1–8.
- [27] X. Wang, A. K. Singh, B. Li, Y. Yang, H. Li, and T. Mak, "Bubble budgeting: Throughput optimization for dynamic workloads by exploiting dark cores in many core systems," *IEEE Trans. Comput.*, vol. 67, no. 2, pp. 178–192, Feb. 2018.
- [28] S. Wen, X. Wang, A. Singh, Y. Jiang, and M. Yang, "Performance optimization of many-core systems by exploiting task migration and dark core allocation," *IEEE Trans. Comput.*, early access, Dec. 4, 2021, doi: 10.1109/TC.2020.3042663.
- [29] X. Huang, X. Wang, Y. Jiang, A. K. Singh, and M. Yang, "Dynamic allocation/reallocation of dark cores in many-core systems for improved system performance," *IEEE Access*, vol. 8, pp. 165693–165707, 2020.
- [30] Unified Extensible Firmware Interface Forum. *Advanced Configuration and Power Interface (ACPI) Specification*. Accessed: May 1, 2021. [Online]. Available: <https://www.uefi.org/specifications>
- [31] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.
- [32] D. E. Farrar and R. R. Glauber, "Multicollinearity in regression analysis: The problem revisited," *Rev. Econ. Statist.*, vol. 4, pp. 92–107, Feb. 1967.
- [33] T. Hastie, "Efficient quadratic regularization for expression arrays," *Biostatistics*, vol. 5, no. 3, pp. 329–340, Jul. 2004.
- [34] 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 *Proc. 42nd Annu. IEEE/ACM Int. Symp. Microarchitecture*, 2009, pp. 469–480.
- [35] R. Rohith, V. Rathore, V. Chaturvedi, A. K. Singh, S. Thambipillai, and S.-K. Lam, "LifeSim: A lifetime reliability simulator for manycore systems," in *Proc. IEEE 8th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2018, pp. 375–381.
- [36] T. E. Carlson, W. Heirman, and L. Eeckhout, "Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulation," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal.*, 2011, pp. 1–12.
- [37] R. Zhang, M. R. Stan, and K. Skadron, "HotSpot 6.0: Validation, acceleration and extension," Univ. Virginia, Charlottesville, VA, USA, Tech. Rep. CS-2015-04, 2015.

[38] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan, "HotSpot: A compact thermal modeling methodology for early-stage VLSI design," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 5, pp. 501–513, May 2006.

[39] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta, "The SPLASH-2 programs: Characterization and methodological considerations," in *Proc. 22nd Annu. Int. Symp. Comput. Archit.*, 1995, pp. 24–36.

[40] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The PARSEC benchmark suite: Characterization and architectural implications," in *Proc. Int. Conf. Parallel Archit. Compilation Techn. (PACT)*, 2008, pp. 72–81.

[41] M. S. Mohammed and G. A. Abandah, "Communication characteristics of parallel shared-memory multicore applications," in *Proc. IEEE Jordan Conf. Appl. Electr. Eng. Comput. Technol. (AEECT)*, Nov. 2015, pp. 1–6.



(MCSoc), network-on-chip (NoC), thermal management, and parallel processing.

MOHAMMED SULTAN MOHAMMED (Member, IEEE) received the B.Sc. degree in computer engineering from Hodeidah University, Yemen, in 2005, and the M.Sc. degree in computer engineering and networks from The University of Jordan, Jordan, in 2015. He is currently pursuing the Ph.D. degree in electronic and computer engineering with Universiti Teknologi Malaysia, Malaysia. His research interests include computer architectures, many-core system-on-chip



NORLINA PARAMAN received the B.E. degree in computer engineering, the M.E. degree in electronic and telecommunication engineering, and the Ph.D. degree in electrical engineering from Universiti Teknologi Malaysia, in 2003, 2006, and 2017, respectively. She is currently a Senior Lecturer with the Faculty of Engineering, UTM. Her research interests include digital IC test and digital design.



interests include optimization methods and design automation for applications in video coding and deep learning.

AB AL-HADI AB RAHMAN received the B.S. degree from the University of Wisconsin-Madison, USA, in 2004, the M.Eng. degree from Universiti Teknologi Malaysia, in 2008, and the Ph.D. degree from École polytechnique fédérale de Lausanne, Switzerland, in 2013. He is currently a Senior Lecturer at Universiti Teknologi Malaysia. He has authored or coauthored more than 30 journals and conference papers, mainly with contributions in developing new design methodologies and techniques for high-performance and low-power systems. His current research



the Faculty of Engineering, School of Computing, Universiti Teknologi Malaysia. He is the author of 35 articles related to information and network security. His research interests include vehicular network security, cyber threat intelligence, intrusion detection, data science, data mining, and knowledge discovery. He was a recipient of many awards and recognitions, such as the Postdoctoral Fellowship Award, the Best Postgraduate Student Award, and the Best Presenter Award from the Faculty of Engineering, School of Computing, UTM; and the Best Papers Awards from IICIST, Kuala Lumpur, Malaysia, and Effat University, Jeddah, Saudi Arabia.

FUAD A. GHAIEB received the B.Sc. degree in computer engineering from Sana'a University, Sana'a, Yemen, in 2003, and the M.Sc. and Ph.D. degrees in computer science and information security from the Faculty of Computing, Universiti Teknologi Malaysia, Johor, Malaysia, in 2014 and 2018, respectively. From 2004 to 2012, he was a Lecturer of network and computer engineering at Sana'a Community College, Sana'a. He is currently a Senior Lecturer of cybersecurity with

AHLAM AL-DHAMARI received the B.Sc. degree in computer engineering from Hodeidah University, Yemen, the M.Sc. degree in computer engineering and networks from The University of Jordan, Jordan, and the Ph.D. degree in electrical engineering from Universiti Teknologi Malaysia (UTM), Malaysia. Her research interests include image and video processing, computer vision, machine learning, deep learning, computer architectures, and crowd analysis and management.



many-core system-on-chips, network-on-chip interconnects, domain-specific computer architectures, and network algorithmics accelerators.

MUHAMMAD NADZIR MARSONO received the B.Eng. degree in computer engineering and the M.Eng. degree in electrical engineering from Universiti Teknologi Malaysia, Malaysia, in 1999 and 2001, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Victoria, Victoria, BC, Canada, in 2007. He is currently an Associate Professor with the Department of Electronics and Computer Engineering, Faculty of Engineering, School of Electrical Engineering, Universiti Teknologi Malaysia. His research interests include

...