*electronics*

MDPI

*Article*

# DTaPO: Dynamic Thermal-Aware Performance Optimization for Dark Silicon Many-Core Systems

**Mohammed Sultan Mohammed [1,2,*], Ali A. M. Al-Kubati [2,3], Norlina Paraman [1], Ab Al-Hadi Ab Rahman [1] and M. N. Marsono [1,*]**

[1] Division of Electronic and Computer Engineering, School of Electrical Engineering, Faculty of Engineering, Universiti Teknologi Malaysia, Johor 81310, Malaysia; pnorlina@utm.my (N.P.); hadi@utm.my (A.A.-H.A.R.)
[2] Department of Computer Engineering, Hodeidah University, Hodeidah 3114, Yemen
[3] Department of Information Systems, University of Bisha, Bisha 67714, Saudi Arabia; alikubati@ub.edu.sa
* Correspondence: sammohammed2@live.utm.my (M.S.M.); mnadzir@utm.my (M.N.M.)

Received: 26 October 2020; Accepted: 18 November 2020; Published: 23 November 2020


**Abstract:** Future many-core systems need to handle high power density and chip temperature effectively. Some cores in many-core systems need to be turned off or 'dark' to manage chip power and thermal density. This phenomenon is also known as the dark silicon problem. This problem prevents many-core systems from utilizing and gaining improved performance from a large number of processing cores. This paper presents a dynamic thermal-aware performance optimization of dark silicon many-core systems (DTaPO) technique for optimizing dark silicon a many-core system performance under temperature constraint. The proposed technique utilizes both task migration and dynamic voltage frequency scaling (DVFS) for optimizing the performance of a many-core system while keeping system temperature in a safe operating limit. Task migration puts hot cores in low-power states and moves tasks to cooler dark cores to aggressively reduce chip temperature while maintaining high overall system performance. To reduce task migration overhead due to cold start, the source core (i.e., active core) keeps its L2 cache content during the initial migration phase. The destination core (i.e., dark core) can access it to reduce the impact of cold start misses. Moreover, the proposed technique limits tasks migration among cores that share the last level cache (LLC). In the case of major thermal violation and no cooler cores being available, DVFS is used to reduce the hot cores temperature gradually by reducing their frequency. Experimental results for different threshold temperatures show that DTaPO can keep the average system temperature below the thermal limit. Affirmatively, the execution time penalty is reduced by up to 18% compared with using only DVFS for all thermal thresholds. Moreover, the average peak temperature is reduced by up to 10.8 °C. In addition, the experimental results show that DTaPO improves the system's performance by up to 80% compared to optimal sprinting patterns (OSP) and reduces the temperature by up to 13.6 °C.

**Keywords:** dark silicon; dynamic thermal management (DTM); dynamic voltage frequency scaling (DVFS); many-core system; task migration; power states

## 1. Introduction

Moore's law [1] predicts that the number of transistors on a chip would double every two years, and Dennard scaling [2] predicts that power downscaling is proportional to technology size. These two laws are the key concepts for increasing processor performance. As technology size reduces, it becomes difficult to scale down the supply voltage as it nears the threshold voltage. Thus, further increases in frequency are infeasible due to increasing power densities that directly contribute to increasing chip temperature. As a solution, more cores on a single chip are integrated to increase processing

performance. According to ITRS [3], the number of cores in the future many-core systems will increase to hundreds in mobile devices and thousands in servers.

Although the many-core system is a promising solution for improving processing performance, further reduction of technology size without downscaling the supply voltage would increase the many-core system power density that leads to increased chip temperature. Only some cores can be active (i.e., turned on) while other cores should be dark (i.e., turned off) to ensure safe chip operating temperature. Dynamic thermal management (DTM) manages active cores to run at different voltage/frequency levels. Consequently, turning some cores off will prevent a many-core system from fully utilizing a large number of cores for maximum processing performance. This is the dark silicon problem [4] that is expected to be significant in the future many-core systems.

The authors of [4,5] predicted that, in 8-nm technology, more than half of the cores on a chip would be dark cores. This prediction has prompted researchers to find techniques to maximize multi/many-core system performance for dark silicon while maintaining safe thermal operation. Some of these techniques proposed improvements in system performance under a fixed per-chip power budget called thermal design power (TDP) or variable power budget called thermal safe power (TSP) [6–9]. However, using power-budget based techniques could lead to severe thermal violations [10]. A few techniques have been proposed to optimize multi/many-core system performance under thermal constraints (e.g., [11–13]). However, these techniques may not be suitable for run-time management because of high scheduling time overhead.

DTM techniques should be used to avoid a thermal violation at run-time. DTM is an efficient technique for optimizing cores performance under the thermal constraint [14]. Task migration and dynamic voltage frequency scaling (DVFS) are the two most commonly used DTM techniques for run-time thermal management. The task migration technique moves tasks to a cooler dark core to reduce system temperature and to balance cores processing load such that all cores can operate at the maximum frequency under safe thermal constraints. However, task migration imposes performance overhead due to memory data transfer and cannot guarantee that the average temperature is lower than the critical temperature [8]. On the other hand, using DVFS can guarantee that the average temperature is not higher than the critical core temperature by dynamically controlling chip temperature. However, reducing the frequency of the chip leads to reducing chip performance.

This paper proposes a dynamic thermal-aware optimization technique (DTaPO) for dark silicon many-core systems. The proposed technique uses both task migration and DVFS techniques to optimize many-core system performance while maintaining a safe operating temperature. Our proposed technique's novelty is that it utilizes both active and dark cores to optimize the many-core system's performance. The proposed technique reduces task migration overhead by migrating only in one direction, i.e., from an active core to a dark core. Task migration can aggressively reduce the core temperature by turning a core off and moving tasks to a cooler core dark core. Multiple low-power states are supported to reduce cold start overhead due to L1 and L2 cache misses. During the initial phase of task migration, the source core enters the first low power state where it maintains the contents of L2 caches [15]. Therefore, during cold cache misses, the cache controller can bring data to the destination core from the private source core's L2 cache instead of the main memory. Furthermore, the proposed technique uses a cluster-based task migration by moving tasks inside a core cluster that shares the same last level cache (LLC). Thus, this will reduce the number of memory accesses due to cache misses as task binaries and data are already available in the shared LLC, which reduces fetching them from the main memory. In the case of the dark cores temperature exceeding a safe temperature, DVFS is used to reduce the core temperature by reducing active cores frequency progressively when no cooler cores are available.

The remainder of this paper is organized as follows. Section 2 discusses relevant work for optimizing many-core systems performance under thermal constraint. Section 3 presents the proposed DTaPO technique. An experimental evaluation of the proposed work is discussed in Section 4. Section 5 concludes the paper.

## 2. Related Work

The dark silicon problem is due to the increase in many-core systems' power densities as technology sizes shrink, leading to thermal violations. In recent years, the dark silicon problem has received considerable attention as a real many-core system problem that needs to be carefully addressed. It affects the many-core system from utilizing all available cores for maximum performance [16]. Many techniques have been proposed to optimize the performance of multi/many-core systems in the dark silicon era. These techniques can be classified either as techniques to optimize performance under the power budget constraint or techniques for optimizing performance under thermal constraint.

The power budget techniques in [6,8,9,17] consider a fixed per-chip power budget TDP or per-core power budget TSP. These techniques aim to avoid thermal violations by running the system under the TDP/TSP power budget. However, using only the power budget may lead to a severe thermal violation due to excluding transient temperatures and heat transfer among cores [10]. On the other hand, thermal constraint techniques deal directly with the chip temperature. Thus, both transient temperatures and heat transfer among cores are considered to avoid thermal violations. A DTM technique should be used to prevent thermal violations at run-time. Task migration and DVFS are the two widely used techniques for DTM. The following paragraphs discuss related work that used these DTM techniques to maximize performance under the thermal constraint for dark silicon.

Some approaches use task migration and application mapping to improve dark silicon many-core systems. Shafique et al. [13] proposed a variability-aware dark silicon management technique called DaSiM. DaSiM uses both thread mapping and dark silicon patterning to lower the maximum temperature by modeling core-to-core leakage power variations. Consequently, it can activate or boost more cores. DaSiM aims to avoid a thermal violation by providing a lightweight run-time prediction technique for a steady-state temperature. This technique can predict the temperature distribution for a candidate solution. When a thermal violation occurs, DaSiM uses task migration or power-gating to deal with the thermal violation.

Similarly, Kanduri et al. [18] proposed a dark silicon aware run-time application mapping approach for many-core systems to distribute power density uniformly on the chip by patterning active cores and the dark cores. This approach keeps the active cores' temperature within a safe operating temperature to provide a higher power budget and better use of the resources. Wang et al. [19,20] also proposed a static and dynamic patterning approach for many-core systems to find the location and number of active and dark cores for each application. This approach aims to optimize the system throughput by optimizing performance, communication cost, and waiting time. All of these approaches used task migration only. However, in some cases, using only task migration cannot ensure a safe temperature operation. Thus, some cores should turn off, which leads to higher performance degradation than using DVFS.

Other approaches use DVFS and application mapping to improve dark silicon many-core systems. Khdr et al. [12] proposed a design-time resource management technique called DsRem for maximizing the performance of dark silicon many-core systems. It maps applications among active cores such that the maximum steady-state temperature for all cores is kept lower than the threshold temperature. DsRem schedules the number of active cores and their DVFS levels based on the thread-level parallelism (TLP) and instruction-level parallelism (ILP) characteristics of the running applications. However, DsRem has a high scheduling time overhead that may not be suitable for run-time performance optimization.

Another DVFS technique to improve dark silicon many-core systems' performance is by increasing cores' frequencies for a short period. This technique is called computation sprinting [7,11,21–27]. Raghavan et al. [11] proposed a computational sprinting technique to improve many-core system performance through activating all cores (including dark cores) with maximum frequency for a sub-second period to improve parallel computation and later deactivate them to cool down the chip. During the sprinting mode, power consumption significantly exceeds the TDP budget. Raghavan et al. [21] proposed an adaptive sprint pacing to adapt sprint pattern at run-time dynamically. This technique runs all

cores at maximum speed until half of the thermal constraints and then reduces the cores speed to half. Wang et al. [27] proposed a technique to find the optimal sprinting patterns (OSP) that keep the many-core system sprinting without resting to maximize the system performance.

A few studies used a combination of task migration and DVFS, such as [8,28,29]. Hanumaiah et al. [28] proposed a technique for maximizing system performance by finding the optimal DVFS levels of the cores under a thermal constraint. However, this technique ignores the heat generated from the neighboring cores to reduce the computation overhead. In a many-core system, heat exchange among cores is significant and cannot be ignored. Wang et al. [8] proposed a hierarchical dynamic thermal management technique for many-core systems. This technique uses both task migration and DVFS with model predictive control to maximize many-core system performance. However, this technique did not consider the dark silicon problem. It reduces the voltage/frequency level of cores and does not turn them completely off for aggressive thermal reduction. Wang et al. [29] proposed a technique that considered the dark silicon problem. This technique swaps the tasks among active cores to balance the thermal/power of the chip. However, swapping tasks between two active cores doubles the migration overhead since the tasks migrate in both directions.

Our proposed technique differs from the techniques mentioned above in that it leverages dark silicon by moving a task from a hot core to a dark core. Previous studies avoid migration to dark cores due to cold start cache misses overhead. However, in modern many-core systems, the core goes in multiple low-power states before it completely shut off, such as Intel Xeon Phi [30]. During the first low-power state, its L2 cache stays active, and the destination core can access data from it rather than from the shared L3 cache or the main memory. The proposed technique combines task migration and DVFS to optimize many-core system performance while the system temperature is kept in a safe operating range. The tasks are moved only in one direction after activating the dark core, i.e., a task is only moved from an active core to the dark core. The proposed DTaPO differs from our previous proof-of-concept work [31] as follows:

- A cluster-based task migration is used to reduce the number of memory accesses due to L2 cache misses as migration is only allowed between cores sharing the LLC.
- The cold start overhead is considered as it can significantly affect the system performance.
- The proposed DTaPO's performance is evaluated under several threshold temperatures.
- A mix of four compute- and memory-intensive applications are used with more cores to show the task migration overhead, where using only compute-intensive applications cannot show the task migration overhead.

## 3. Proposed DTaPO Methodology

This section presents the proposed technique methodology. Our proposed technique aims to optimize the performance of many-core systems under thermal constraints dynamically. Therefore, the proposed technique should be computationally light. DTM techniques (i.e., task migration and DVFS) are good candidates to manage chip thermal at run-time. The proposed technique depends on that the many-core system supports multiple low-power states. A short background about the core power states is introduced in the next subsection.

### *3.1. Core Power States*

Modern many-core processors support several low-power states called C-states [32]. The C-states are named C0, C1, C2, ..., C$n$, where the value of $n$ depends on the processor design. C0 state is the active state, where the core in the execution mode. As the C-state goes deeper, more power-reducing actions are taken by shutting down more core components, as illustrated in Figure 1. However, the wake-up latency is increased as the C-state goes deeper. According to the ACPI standard [32], the C1 state reduces the core voltage and turns off the core's clock while maintaining the contents of L1/L2

caches. In the C2 state, the contents of L1/L2 caches are flushed to the LLC cache. In the C3 state, the core is completely off.

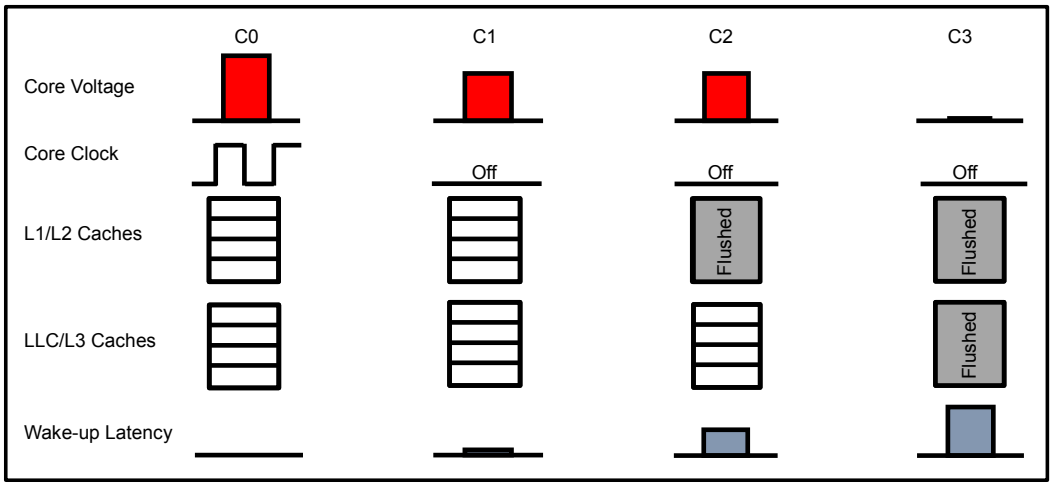
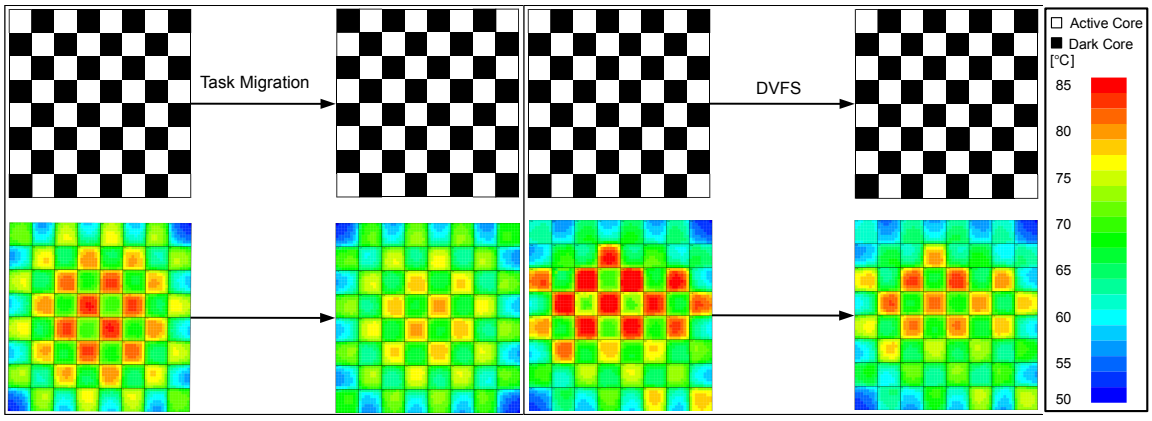



**Figure 1.** Core power states illustration.

### *3.2. Proposed Framework*

The proposed technique uses task migration to lower the temperature of hot active cores aggressively by moving tasks to cooler dark cores after turning them on, as shown in Figure 2a. Moving the tasks to dark cores instead of other active cores reduces task migration overhead, where the tasks move only in one direction from active to dark cores. To reduce the cold start overhead due to L1/L2 cache misses, the proposed technique assumes that the source core goes to C1 state at the initial phase of migration. As mentioned in the previous section, the core maintains the contents of its L1/L2 caches in the C1 power state. Thus, the cache controller can get data for the destination core from the source core's L2 cache instead of the last level cache (LLC) or the main memory. Moreover, the cores are arranged in clusters, where all cores belonging to a cluster share the LLC. Thus, task migration is performed as cluster-based, i.e., the tasks are migrated among the cores that share the same LLC. The cluster-based task migration will reduce the number of memory accesses due to cache misses because the data are already available to all cores that share an LLC without the need to fetch it from the main memory. In the case of no cooler cores available, DVFS is used to reduce hot cores temperature gradually by lowering the voltage/frequency of the active cores to below the threshold temperature, as shown in Figure 2b.




(**a**) Task migration (**b**) DVFS

**Figure 2.** DTaPO technique: (**a**) perform task migration; and (**b**) perform DVFS.

As the proposed technique optimizes the performance of dark silicon many-core systems, we assume that only 50% of cores can be activated simultaneously. Figure 3 shows an overview of the proposed technique. The active and dark cores are arranged in the form of a chessboard pattern so that each active core is surrounded by dark cores to dissipate the heat better [10]. Although the chessboard pattern increases the communication latency by one hop for each active core, its peak chip temperature is lower than for the contiguous pattern. To map the upcoming task on the chessboard pattern, the following formula is used.

$$l = \begin{cases} c+1 & when\ w\ mod\ c = w-1 \\ c+3 & when\ w\ mod\ c = w-2 \\ c+2 & otherwise \end{cases} \quad (1)$$

where $l$ is the location of new task, $c$ is the location of current task, and $w$ is the width of the network-on-chip (NoC) mesh. Equation (1) ensures that all tasks are mapped to the active cores, which are surrounded by the dark cores. To ensure that the tasks are migrated in a clustered-based, Equation (2) is used.

$$d = \begin{cases} s-(w-1) & when\ s\ mod\ w = w-1 \\ s+1 & otherwise \end{cases} \quad (2)$$

where $d$ is the location of destination core and $s$ is the location of source core.

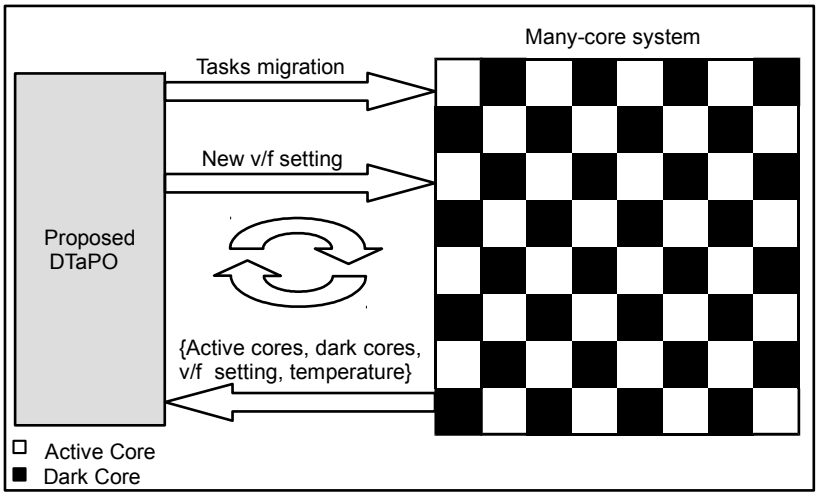



**Figure 3.** Overview of the proposed framework.

Periodically, the proposed technique monitors the many-core system parameters such as the number of active cores, the number of dark cores, the current setting of frequency, and the transient temperature. Transient temperature is sensed instead of steady-state temperature to allow the thermal management to capture actual core temperature, which might exceed the steady-state temperature [33]. Suppose the transient temperature exceeded a predefined threshold temperature, and the temperature of the dark cores was below the threshold temperature by a safe margin. In that case, the proposed technique moves tasks from active cores to dark cores. The safe margin prevents the tasks from moving if the dark cores temperature just below the threshold temperature.

### *3.3. Proposed Algorithm*

Given $M$ multi-threaded applications running on a many-core system that consists of $N$ cores where only half of the cores can be active, the proposed technique aims at minimizing the total

completion time or makespan time ($M_t$), while keeping the chip temperature below a predefined threshold temperature $T_{thr}$. This aim can be mathematically expressed as follows:

$$\text{Minimize } M_t \qquad \text{s.t. } T_i < T_{thr} \text{ for all } i = 0, 1, \cdots, N-1 \tag{3}$$

where $T_i$ is the transient temperature of core $i$, where the transient temperatures of all active and dark cores are checked. Table 1 describes all symbols that are used in our algorithm.

**Table 1.** The used symbols description.

| Symbol | Description |
|---|---|
| $\boldsymbol{A}$ | Set of all active cores |
| $\boldsymbol{D}$ | Set of all dark cores |
| $\boldsymbol{\mathcal{F}}$ | Set of frequencies of all cores |
| $\mathbf{T}$ | Set of transient temperature of all cores |
| $\mathbb{Q}$ | Set of all tasks |
| $a_i$ | Active core $i \in \boldsymbol{A}$ |
| $d_i$ | Dark core $i \in \boldsymbol{D}$ |
| $t_i$ | Task on core $i \in \mathbb{Q}$ |
| $T_{thr}$ | Threshold temperature |
| $\mathcal{F}_{thr}$ | Threshold frequency |
| $\varepsilon$ | Safe margin value |
| $\triangle f$ | Frequency level step |

The proposed algorithm (DTaPO) is detailed in Algorithm 1. DTaPO monitors the following parameters: the transient temperature of all cores $\mathbf{T}$, a set of all active cores $\boldsymbol{A} = \{a_0, ..., a_n\}$, a set of all dark cores $\boldsymbol{D} = \{d_0, ..., d_n\}$, and a set of current frequency level for all cores $\boldsymbol{\mathcal{F}}$. In addition, DTaPO reads a threshold temperature $T_{thr}$ and threshold frequency $\mathcal{F}_{thr}$ from a configuration file. Periodically, the proposed technique monitors the transient temperature for each active core. If the temperature exceeded the threshold temperature, DTaPO checks the temperature of a destination dark core. If the temperature of dark cores is less than the threshold temperature by a safe margin $\varepsilon$, DTaPO marks a task as a movable task by setting $\mathbb{Q}[t_i] = 1$ (Lines 4–6). If the temperature of dark cores is not less than the threshold temperature, but it is less than the active core temperature by $\varepsilon$, DTaPO reduces the frequency of dark core by one frequency level step called $\zeta$, and marks a task as a movable task (Lines 7–10). Otherwise, it reduces only the frequency level of high-temperature cores by $\zeta$, and marks a task as an unmovable task by setting $\mathbb{Q}[t_i] = 0$ (Lines 11–14). In the case the temperature is below the threshold temperature by the safe margin $\varepsilon$, and the frequency does not exceed the threshold frequency, the proposed technique increases the frequency level by one step to enhance the performance (Lines 16–19). Lastly, if all tasks are movable tasks, DTaPO performs a task migration by moving tasks from the active cores to the dark cores after activating the dark cores within the same cluster and putting the active cores in a low-power state (Lines 20–24).

**Algorithm 1** DTaPO algorithm

**Input:** $\boldsymbol{A}$, $\boldsymbol{D}$, $\boldsymbol{\mathcal{F}}$, $\boldsymbol{T}$, $T_{thr}$, $\mathcal{F}_{thr}$, $\varepsilon$, and $\zeta$
**Output:** New sets of $\boldsymbol{A}$, $\boldsymbol{D}$, and $\boldsymbol{\mathcal{F}}$

```
while true do
    for ∀ a_i ∈ A do
        if T[a_i] > T_thr then
            if T[d_i] < T_thr − ε then
                ℚ[t_i] = 1
            end
            else if T[d_i] < T[a_i] − ε then
                Reduce ℱ[d_i] by ζ
                ℚ[t_i] = 1
            end
            else
                Reduce ℱ[a_i] by ζ
                ℚ[t_i] = 0
            end
        end
        else if ℱ[a_i] <ℱ_thr and T[a_i] < T_thr − ε then
            Increase ℱ[a_i] by ζ
            ℚ[t_i] = 0
        end
        if M[t_i] = 1 ∀ t_i ∈ ℚ then
            Activate all cores in D
            Move the tasks from A to D within the same cluster
            Put all cores in A at low-power state
        end
    end
end
```

## 4. Experimental Evaluation

This section presents the experimental setup, results, and discussion of the proposed work.

### *4.1. Experimental Setup*

A 64-core NoC-based many-core system with shared memory was used to evaluate our proposed work. The simulated cores have a homogeneous microarchitecture, i.e., they have the same instruction set architecture (ISA), and a heterogeneous frequency, i.e., each core can run on a different frequency. The maximum frequency of each core is set to 4 GHz. Figure 4 shows the floorplan of the simulated system. The size of each core is 2.95 mm $\times$ 2.95 mm, where its power profile is calculated using McPAT [34] for 22-nm technology. Every core has a private 32 KB L1 data cache, 32 KB L1 instruction cache and 512 KB L2 cache. Every eight cores share 8 MB L3 cache. The 64-core are connected through 8 $\times$ 8 mesh NoC. A summary of the system configuration is presented in Table 2.

The experimental setup of the proposed work is shown in Figure 5. We used LifeSim simulation tool [35] for many-core systems. LifeSim is a tool that contains Sniper [36], McPAT [34], and HotSpot [37] simulators. Sniper is a parallel x86-64 multi/many-core simulator. It offers fast simulation with an average performance error of 25% compared to real hardware. McPAT is widely and recently used [38,39] for integrated power, area, and timing modeling as it provides comprehensive low-level configuration details for multi/many-core processors. Other power models such as Wattch [40] and PowerTrain [41] can also be used to estimate the cores' power profiles as our proposed algorithm deals

with the cores' transient temperatures, which do not change immediately with the power changes. HotSpot is the most widely used thermal simulator. It is based on the popular stacked-layer packaging scheme of modern very-large-scale integration (VLSI) systems (see Figure 6). We modified Sniper to support dark silicon modeling. Specifically, we modified Sniper's scheduler to schedule the tasks only on the active cores using a core mask pattern. In addition, we modified McPAT to report only the static power of the dark cores. Moreover, we modeled wake-up latency from a dark state to an active state by adding 100 µs, which is near the practical wake-up latency for real processors as reported in [42].

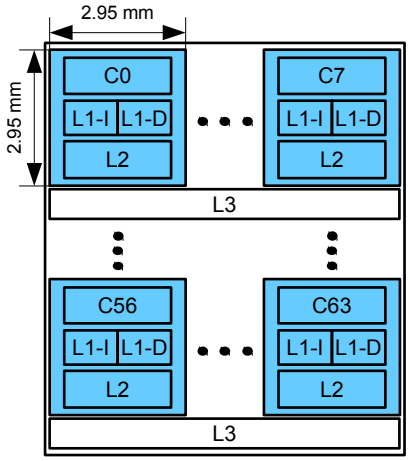



**Figure 4.** Floorplan of the simulated many-core system.

**Table 2.** System configuration.

| 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 |

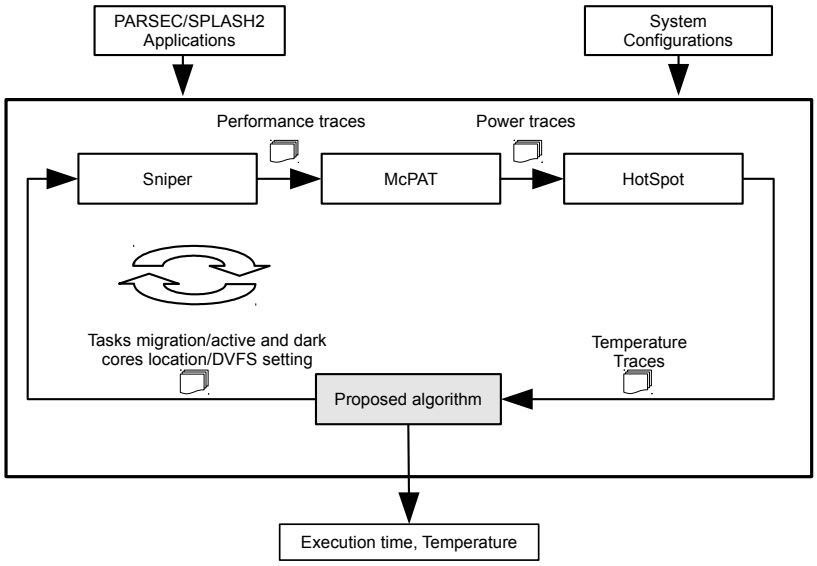



**Figure 5.** Experimental setup of the proposed technique.

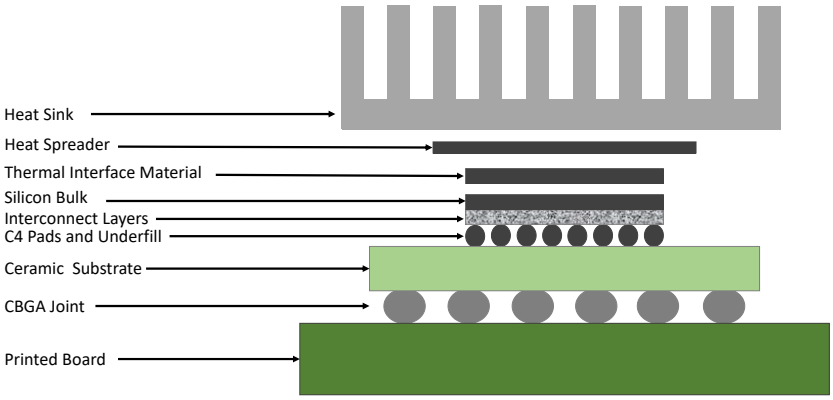



**Figure 6.** Stacked layers in a typical ceramic ball grid array (CBGA) package (adapted from Figure 2 in [37]).

Based on the system configurations that are provided to the simulation tool (i.e., number of cores and floorplan), Sniper generates performance traces by run multi-threaded applications from the Splash-2 benchmark suite [43] and PARSEC benchmark suite [44]. McPAT uses these traces to estimate the power consumption of each core. As the whole core will turn off, the total power of the core (including L1 and L2 power) is used as an input to HotSpot. The L3 cache's power was excluded as it is sited outside the core, and it is shared among the cluster's cores. The processing core generates the most heat, and, thus, the core's power density is not uniformly distributed. As the temperature is measured for each core, the non-uniform power density will not affect our estimation. After a specified period that is termed the *control period*, HotSpot uses both the power traces and the cores' steady-state temperature for the previous control period (i.e., the cores' initial temperature for the current control period) to estimate the transient temperature. Based on the generated transient temperature, the proposed DTaPO algorithm determines either to migrate the tasks or to reduce the voltage/frequency level.

In our experiment, a mix of four compute- and memory-intensive applications were used to evaluate the proposed technique's efficiency. *Radix* and *ocean* are from Splash-2, and *blackscholes* and *bodytrack* are from PARSEC. Compute-intensive applications are good candidates to validate our DTaPO algorithm because the tasks for these applications are high-temperature tasks, which can show an increase in the core temperature [45]. On the other hand, using memory-intensive applications can show task migration overhead of the proposed technique. Table 3 shows the number of instructions and cycles (in millions) and the number of memory accesses of the used applications, where each application has eight threads. It can be noticed that *ocean* and *bodytrack* have the highest number of memory accesses.

**Table 3.** Applications characteristics.

| Benchmark | Threads | Instructions (M) | Cycles (M) | Memory Accesses |
|---|---|---|---|---|
| Radix | 8 | 187.4 | 53.94 | 136,802 |
| Ocean | 8 | 1165 | 203.6 | 10,886,925 |
| Blackscholes | 8 | 1238 | 229.9 | 18,570 |
| Bodytrack | 8 | 3830 | 658.2 | 287,250 |

The four applications were simulated on 64 cores, where 32 cores are active and 32 cores are dark cores. The values of $\varepsilon$, $\zeta$, and $\mathcal{F}_{thr}$ are empirically determined through parameterization of their values until the system achieves optimal switching frequency (i.e., the system does not keep switching between the active and dark cores). In our experiment, the values of $\varepsilon$, $\triangle f$, and $\mathcal{F}_{thr}$ are set to 5 °C, 200 MHz and 3.6 GHz, respectively. The configuration parameters of HotSpot are detailed in Table 4. The proposed DTaPO technique was evaluated using several threshold temperatures, which

was ranged from 60 to 85 °C to analyze the influence of threshold temperature. Besides, to get more accurate results, the experiments were conducted ten times, and the average results are reported.

**Table 4.** Configuration parameters of HotSpot.

| Parameter | Value |
|---|---|
| Thickness of the chip | 0.15 mm |
| Silicon thermal conductivity | 100 W/(m·K) |
| Silicon specific heat | $1.75 \times 10^6$ 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 | $3.55 \times 10^6$ J/(m$^3$·K) |
| Interface material thickness | 20 µm |
| Interface material thermal conductivity | 4 W/(m·K) |
| Interface material specific heat | $4 \times 10^6$ J/(m$^3$·K) |
| Ambient temperature | 318.15 K |

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

### *4.2. Experimental Results and Discussion*

Our previous paper [31] shows that the use of only task migration cannot keep the average chip temperature below the threshold temperature. Therefore, in this paper, the proposed technique and DVFS are evaluated at several threshold temperatures.

Using the task migration in the dark silicon era can utilize all cores, i.e., active and dark cores, by swapping tasks among active and dark cores, as illustrated in Figure 7. However, task migration incurs system performance overhead due to three factors [46]:

1. A fixed time is spent on storing and restoring the state of architecture.
2. Time is spent draining and refilling a core's pipeline.
3. Time is spent restoring the data from memory due to cache misses.

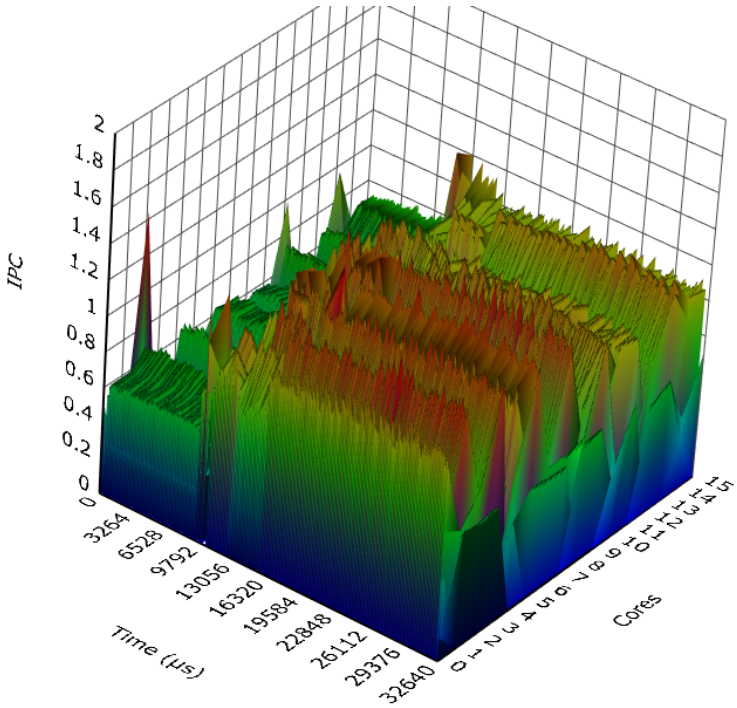



**Figure 7.** Illustration of cores utilization when using task migration.

The last factor is considered very large compared to the first two factors. In our work, 1000 cycles are added as a penalty of the first factor for each task migration, while the last two factors are already modeled in Sniper as mention in [46].

Our proposed technique uses cluster-based task migration to reduce the task migration overhead so that tasks are migrated only among cores that shared an L3 cache. Figure 8 shows the normalized memory accesses when running the four applications using the proposed technique with and without clustering. Using cluster-based task migration reduces the number of memory accesses by 8% over no clustering. In addition, the increase in the number of memory accesses using the proposed technique is only 1% compared with using no DTM. Table 5 shows the average number of times that task migration and DVFS have been used in DTaPO at different thermal thresholds. It can be seen that the number of used task migration and DVFS is decreased by increasing the threshold temperature.

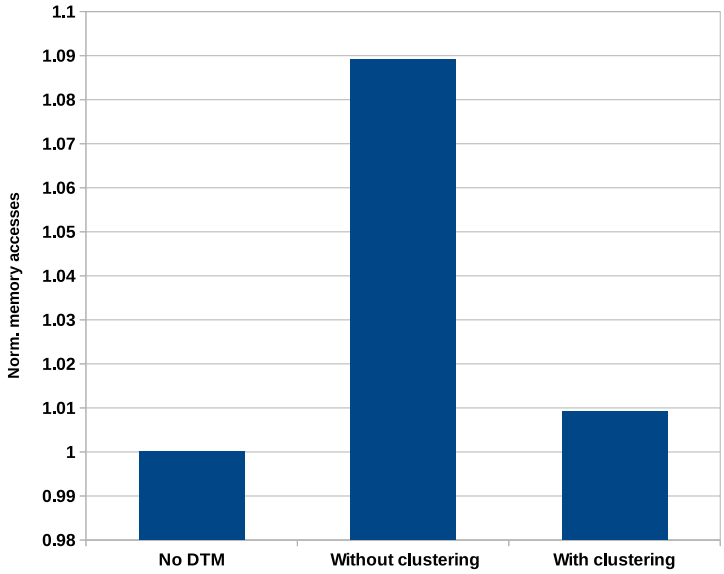



**Figure 8.** Normalized memory accesses with and without clustering at 60 °C threshold temperature.

**Table 5.** The average number of times task migration and DVFS have been used in DTaPO at different thermal thresholds.

| $T_{thr}$ | Avg. Number of Migrations | Avg. Number of DVFS |
|---|---|---|
| 60 °C | 98.7 | 36.4 |
| 65 °C | 74.3 | 27.4 |
| 70 °C | 67.7 | 19.2 |
| 75 °C | 28.7 | 3.7 |
| 80 °C | 18.9 | 0 |
| 85 °C | 0 | 0 |

Figure 9 shows that using the chessboard pattern keeps the active cores cooler than using a contiguous pattern. The peak chip temperature in the chessboard pattern is by about 5 °C lower compared to using the contiguous pattern. In addition, the performance degradation due to increasing the communication latency by one hop for each active core is insignificant by only 1%. Figure 10 shows the penalty of using the proposed technique and DVFS only compared with using no DTM, as well as the maximum, average, and minimum temperature of running the four applications, which are radix, ocean, blackscholes, and bodytrack. The 10th and 90th percentiles are used to represent the minimum and maximum temperature because of the highly fluctuating of transient temperatures, as shown in Figure 11.

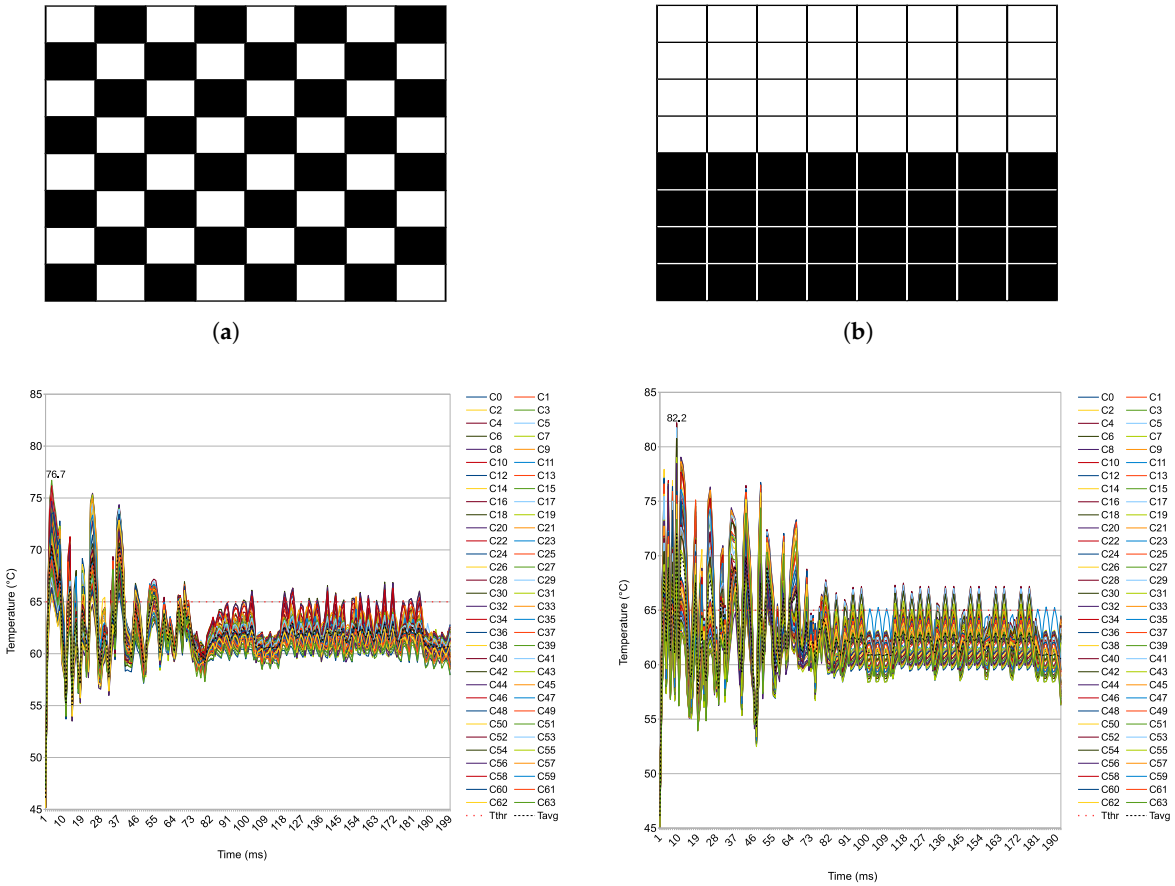



**Figure 9.** Transient temperature and active and dark cores patterns: (**a**) chessboard pattern; (**b**) contiguous pattern; (**c**) transient temperature for chessboard pattern; and (**d**) transient temperature for contiguous pattern.

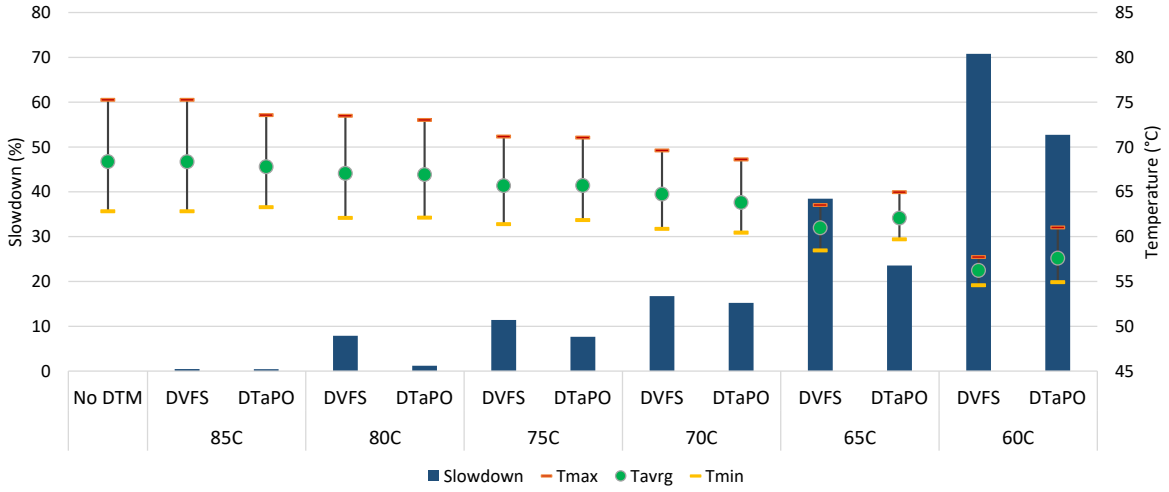
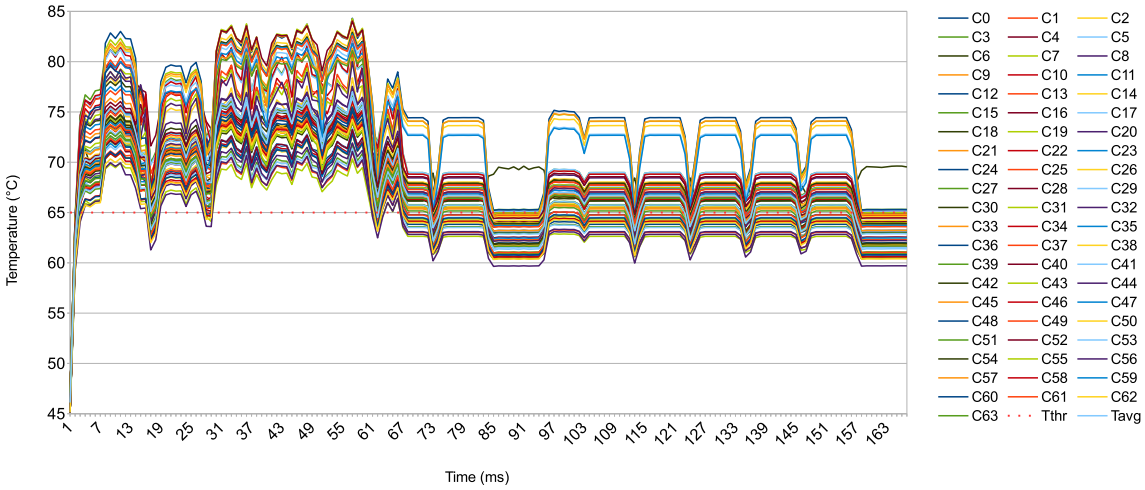
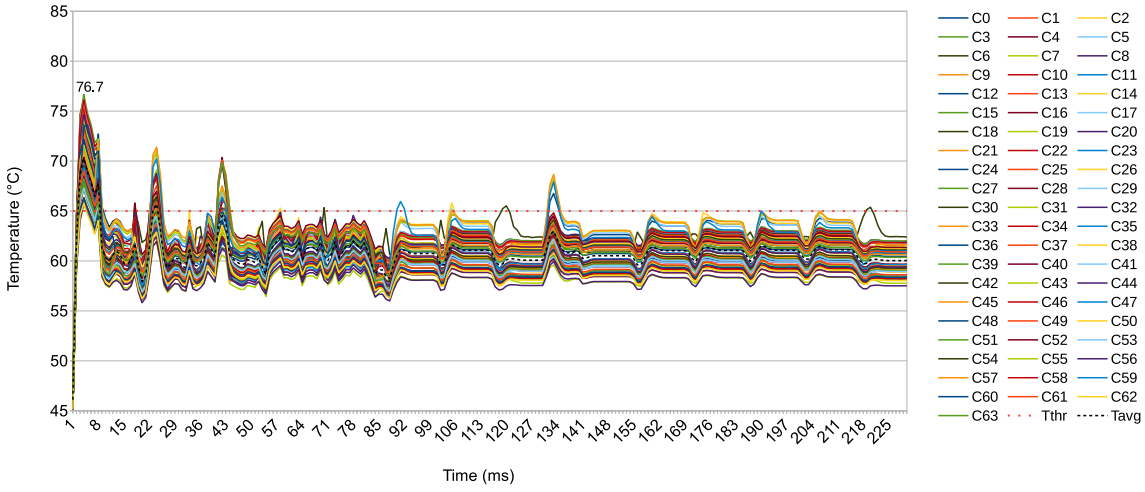


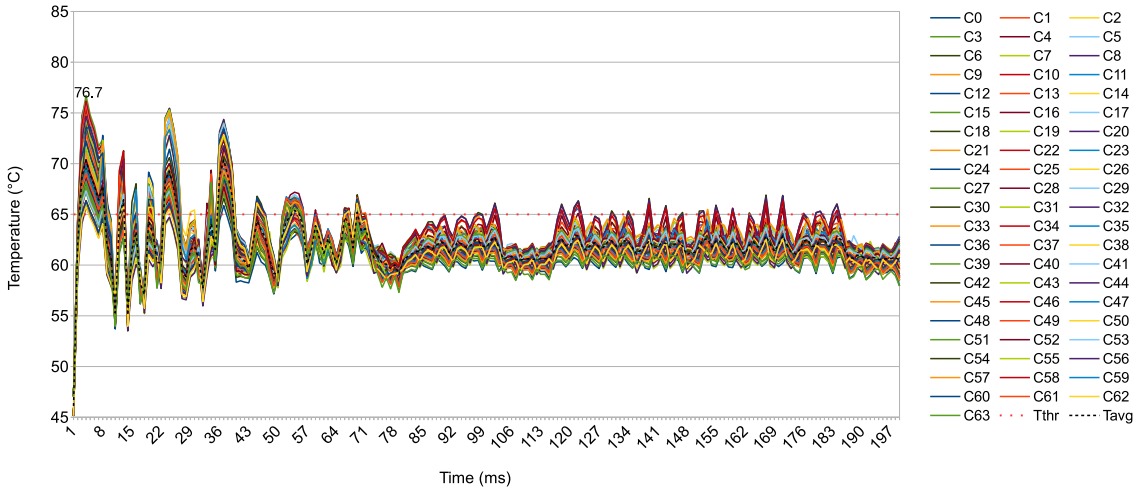

**Figure 10.** Performance slowdown and maximum, average, and minimum temperatures using different DTM techniques.




(**a**)




(**b**)



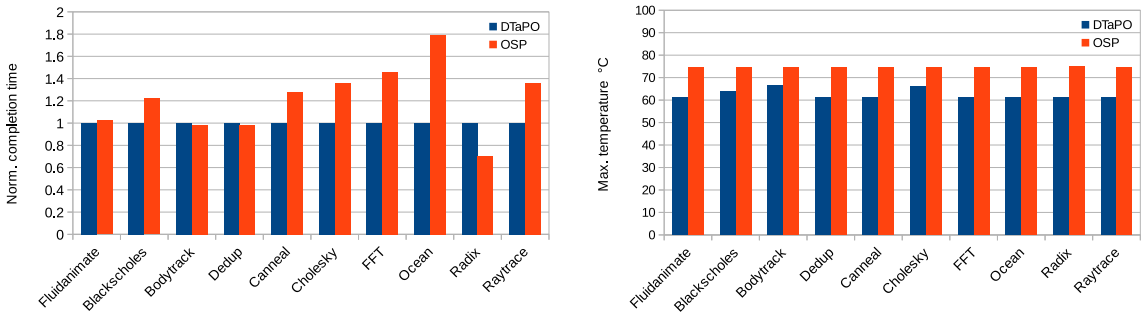

(**c**)

**Figure 11.** Transient temperature of running four applications on 64-core with different DTM techniques: (**a**) without DTM; (**b**) DVFS; and (**c**) DTaPO.

It is observed that the proposed technique (DTaPO) always keeps the average temperature below the threshold temperature with up to 18% lower penalty compared with using only DVFS for any threshold temperatures. Moreover, the average peak temperature is reduced by 10.8 °C. Besides that, the results show that the slowdown is highly dependent on the threshold temperature. The slowdown increases as the threshold decreases due to the increase in the number of calls for DTaPO to retain the chip temperature in a safe operating range.

To compare our work with state-of-the-art, the OSP algorithm [27], which shares the same goal of maximizing the many-core system's performance under thermal constraint, was re-implemented in this experiment. The concept of OSP is to run the many-core system with an optimal frequency that keeps the system sprinting without violating the thermal limit and going to the resting mode. To make a fair comparison, TDP is added as a constraint to the DTaPO algorithm in which the total power consumption does not exceed the TDP. The value of TDP was set to 42 W as in [27]. The values of $\triangle f$ and $\varepsilon$ were chosen to ensure that the system does not enter the resting mode. In our experiments, $\triangle f$ and $\varepsilon$ were set to 200 and 0.5, respectively.

The comparative results when running ten applications from SPLASH2 and PARSEC on the simulation framework described in Section 4 are shown in Figure 12. Figure 12a shows the computational efficiency in terms of normalized completion time. In most of the studied applications, our DTaPO algorithm outperforms the OSP in computational efficiency. It reduces the completion time over OSP by 80% in *Occean*, 46% in *FFT*, 36% in *Raytrace*, 36% in *Cholesky*, 28% in *Canneal*, 21% in *Blackscholes*, and 2% in *Fluidanimate*. These applications have high instruction-level parallelism. Thus, they benefit from running 50% of the cores (32 cores) with a high frequency in DTaPO compared to running 100% of the cores (64 cores) with low frequency in OSP. Although OSP reduces the completion time over DTaPO by 29% in *Radix*, 2% in *Dedup*, and 1.5 in *Bodytrack*, our algorithm reduces the peak temperature in all studied applications by up to 13.6 °C, as shown in Figure 12b. This is because DTaPO uses a chessboard pattern that gives better heat dissipation.




(**a**) (**b**)

**Figure 12.** Comparison results of running 10 applications from SPLASH2 and PARSEC: (**a**) comparison of the computational efficiency in term of completion time; and (**b**) comparison of thermal efficiency in term of peak temperature.

## 5. Conclusions

This paper proposes DTaPO, a dynamic thermal-aware performance optimization technique for dark silicon many-core systems. The proposed technique utilizes both task migration and DVFS to optimize many-core system performance under thermal constraint. Task migration aggressively reduces many-core system temperature by putting the hot cores in low-power states and moves tasks to dark cores that share the LLC. Putting the source core in low-power reduces overhead due to the cold start cache misses. Moreover, limiting task migration among cores that share LLC reduces cache misses for memory-intensive applications. Task migration enhances the cores' utilization in the dark silicon many-core system by swapping tasks among active and dark cores. When cooler cores are unavailable, DVFS is used to keep the core temperature in a safe operating range. The simulation results show

that the average system operating temperature is kept below the threshold temperature with up to 18% less penalty than using only DVFS for all threshold temperatures. In addition, simulation results show that the slowdown is highly dependent on the threshold temperature. The slowdown increases as the threshold decreases because of the increase in the number of calls for DTaPO to keep the chip temperature in a safe operating range. Moreover, the average peak temperature is reduced by 10.8 °C. Compared with the state-of-the-art, our algorithm improves the system's performance by up to 80% in computational efficiency, especially for tasks with high instruction-level parallelism. In addition, it improves the temperature efficiency by up to 13.6 °C. In future work, we plan to integrate a patterning model to our technique to increase the number of active cores without violating the thermal limit.

**Author Contributions:** Conceptualization, M.S.M.; Methodology, M.S.M.; Supervision, A.A.M.A.-K., N.P., A.A.-H.A.R. and M.N.M.; Validation, M.S.M. and M.N.M.; Writing—original draft, M.S.M.; Writing—review and editing, N.P., A.A.-H.A.R. and M.N.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Moore, G.E. Cramming more components onto integrated circuits. *Electronics* **1965**, *38*, 114–117. [CrossRef]
2. Dennard, R.H.; Gaensslen, F.H.; Rideout, V.L.; Bassous, E.; LeBlanc, A.R. Design of ion-implanted MOSFET's with very small physical dimensions. *IEEE J. Solid-State Circuits* **1974**, *9*, 256–268. [CrossRef]
3. ITRS. International Technology Roadmap for Semiconductors 2.0. Available online: https://www.semiconductors.org/wp-content/uploads/2018/06/0_2015-ITRS-2.0-Executive-Report-1.pdf (accessed on 14 October 2019).
4. Esmaeilzadeh, H.; Blem, E.; Amant, R.S.; Sankaralingam, K.; Burger, D. Dark silicon and the end of multicore scaling. In Proceedings of the 38th Annual International Symposium on Computer Architecture (ISCA), San Jose, CA, USA, 4–8 June 2011; pp. 365–376.
5. Henkel, J.; Khdr, H.; Pagani, S.; Shafique, M. New trends in dark silicon. In Proceedings of the 2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 8–12 June 2015; pp. 1–6.
6. Raghunathan, B.; Turakhia, Y.; Garg, S.; Marculescu, D. Cherry-picking: Exploiting process variations in dark-silicon homogeneous chip multi-processors. In Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, 18–22 March 2013; pp. 39–44.
7. Rezaei, A.; Zhao, D.; Daneshtalab, M.; Wu, H. Shift sprinting: Fine-grained temperature-aware NoC-based MCSoC architecture in dark silicon age. In Proceedings of the 53rd Annual Design Automation Conference (DAC), Austin, TX, USA, 5–9 June 2016; pp. 1–6.
8. Wang, H.; Ma, J.; Tan, S.X.D.; Zhang, C.; Tang, H.; Huang, K.; Zhang, Z. Hierarchical dynamic thermal management method for high-performance many-core microprocessors. *ACM Trans. Des. Autom. Electron. Syst. (TODAES)* **2016**, *22*, 1–21. [CrossRef]
9. Pagani, S.; Khdr, H.; Munawar, W.; Chen, J.J.; Shafique, M.; Li, M.; Henkel, J. TSP: Thermal safe power: Efficient power budgeting for many-core systems in dark silicon. In Proceedings of the 2014 International Conference on Hardware/Software Codesign and System Synthesis, New Delhi, India, 12–17 October 2014; pp. 1–10.
10. Shafique, M.; Garg, S.; Henkel, J.; Marculescu, D. The EDA challenges in the dark silicon era: Temperature, reliability, and variability perspectives. In Proceedings of the 51st Annual Design Automation Conference, Francisco, CA, USA, 2–5 June 2014; pp. 1–6.
11. Raghavan, A.; Luo, Y.; Chandawalla, A.; Papaefthymiou, M.; Pipe, K.P.; Wenisch, T.F.; Martin, M.M. Computational sprinting. In Proceedings of the 18th International Symposium on High-Performance Computer Architecture, New Orleans, LA, USA, 25–29 February 2012; pp. 1–12.
12. Khdr, H.; Pagani, S.; Shafique, M.; Henkel, J. Thermal constrained resource management for mixed ILP-TLP workloads in dark silicon chips. In Proceedings of the the 52nd ACM/EDAC/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 8–12 June 2015; pp. 1–6.

13. Shafique, M.; Gnad, D.; Garg, S.; Henkel, J. Variability-aware dark silicon management in on-chip many-core systems. In Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, 9–13 March 2015; pp. 387–392.
14. Donald, J.; Martonosi, M. Techniques for Multicore Thermal Management: Classification and New Exploration. In Proceedings of the 33rd International Symposium on Computer Architecture (ISCA'06), Boston, MA, USA, 17–21 June 2006; pp. 78–88.
15. Thakkar, I.G.; Pasricha, S. LIBRA: Thermal and process variation aware reliability management in photonic networks-on-chip. *IEEE Trans. Multi-Scale Comput. Syst.* **2018**, *4*, 758–772.
16. Shafique, M.; Garg, S. Computing in the Dark Silicon Era: Current Trends and Research Challenges. *IEEE Des. Test.* **2017**, *34*, 8–23. [CrossRef]
17. Muthukaruppan, T.S.; Pricopi, M.; Venkataramani, V.; Mitra, T.; Vishin, S. Hierarchical power management for asymmetric multi-core in dark silicon era. In Proceedings of the 50th ACM/EDAC/IEEE Design Automation Conference (DAC), Austin, TX, USA, 29 May–7 June 2013; pp. 1–9.
18. Kanduri, A.; Haghbayan, M.H.; Rahmani, A.M.; Liljeberg, P.; Jantsch, A.; Tenhunen, H. Dark Silicon Aware Runtime Mapping for Many-core Systems: A Patterning Approach. In Proceedings of the 33rd IEEE International Conference on Computer Design (ICCD), New York, NY, USA, 18–21 October 2015; pp. 573–580.
19. Wang, X.; Singh, A.K.; Wen, S. Exploiting dark cores for performance optimization via patterning for many-core chips in the dark silicon era. In Proceedings of the 2018 Twelfth IEEE/ACM International Symposium on Networks-on-Chip (NOCS), Turin, Italy, 4–5 October 2018; pp. 1–8.
20. Wang, X.; Singh, A.; Li, B.; Yang, Y.; Li, H.; Mak, T. Bubble budgeting: Throughput optimization for dynamic workloads by exploiting dark cores in many core systems. *IEEE Trans. Comput.* **2018**, *67*, 178–192. [CrossRef]
21. Raghavan, A.; Emurian, L.; Shao, L.; Papaefthymiou, M.; Pipe, K.P.; Wenisch, T.F.; Martin, M.M. Utilizing dark silicon to save energy with computational sprinting. *IEEE Micro* **2013**, *33*, 20–28. [CrossRef]
22. Zhan, J.; Xie, Y.; Sun, G. NoC-sprinting: Interconnect for fine-grained sprinting in the dark silicon era. In Proceedings of the 51st Annual Design Automation Conference, Francisco, CA, USA, 2–5 June 2014; pp. 1–6.
23. Shao, L.; Raghavan, A.; Emurian, L.; Papaefthymiou, M.C.; Wenisch, T.F.; Martin, M.M.; Pipe, K.P. On-chip phase change heat sinks designed for computational sprinting. In Proceedings of the 2014 Semiconductor Thermal Measurement and Management Symposium (SEMI-THERM), San Jose, CA, USA, 9–13 March 2014; pp. 29–34.
24. Kaplan, F.; Coskun, A.K. Adaptive sprinting: How to get the most out of Phase Change based passive cooling. In Proceedings of the 2015 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), Rome, Italy, 22–24 July 2015; pp. 37–42.
25. Morris, N.; Stewart, C.; Birke, R.; Chen, L.; Kelley, J. Early work on modeling computational sprinting. In Proceedings of the 2017 Symposium on Cloud Computing, Santa Clara, CA, USA, 24–27 September 2017; p. 661.
26. Morris, N.; Stewart, C.; Chen, L.; Birke, R.; Kelley, J. Model-driven computational sprinting. In Proceedings of the Thirteenth EuroSys Conference, Porto, Portugal, 23–26 April 2018; pp. 1–13.
27. Wang, J.; Chen, Z.; Guo, S.; Li, Y.B.; Lu, Z. Optimal Sprinting Pattern in Thermal Constrained CMPs. *IEEE Trans. Emerg. Top. Comput.* **2019**. [CrossRef]
28. Hanumaiah, V.; Vrudhula, S.; Chatha, K.S. Performance optimal online DVFS and task migration techniques for thermally constrained multi-core processors. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2011**, *30*, 1677–1690. [CrossRef]
29. Wang, H.; Zhang, M.; Tan, S.X.D.; Zhang, C.; Yuan, Y.; Huang, K.; Zhang, Z. New power budgeting and thermal management scheme for multi-core systems in dark silicon. In Proceedings of the 2016 29th IEEE International System-on-Chip Conference (SOCC), Seattle, WA, USA, 6–9 September 2016; pp. 344–349.
30. Intel. Intel® Xeon Phi Processor x200 Product Family Datasheet, Volume 1. Available online: https://www.intel.com/content/www/us/en/processors/xeon/xeon-phi-processor-x200-product-family-datasheet.html (accessed on 12 February 2020).

31. Mohammed, M.S.; Al-Dhamari, A.K.; Rahman, A.A.H.A.; Paraman, N.; Al-Kubati, A.A.M.; Marsono, M.N. Temperature-Aware Task Scheduling for Dark Silicon Many-Core System-on-Chip. In Proceedings of the 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO), Manama, Bahrain, 15–17 April 2019; pp. 1–5.
32. Forum, U.E.F.I. Advanced Configuration and Power Interface (ACPI) Specification. Available online: https://www.uefi.org/specifications (accessed on 12 February 2020).
33. Pagani, S.; Chen, J.J.; Shafique, M.; Henkel, J. MatEx: Efficient transient and peak temperature computation for compact thermal models. In Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, 9–13 March 2015; pp. 1515–1520.
34. Li, S.; Ahn, J.H.; Strong, R.D.; Brockman, J.B.; Tullsen, D.M.; Jouppi, N.P. 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, New York, NY, USA, 12–16 December 2009; pp. 469–480.
35. Rohith, R.; Rathore, V.; Chaturvedi, V.; Singh, A.K.; Thambipillai, S.; Lam, S.K. LifeSim: A lifetime reliability simulator for manycore systems. In Proceedings of the 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 8–10 January 2018; pp. 375–381.
36. Heirman, W.; Carlson, T.; Eeckhout, L. Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulation. In Proceedings of the 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, Seatle, WA, USA, 12–18 November 2011; pp. 1–12. [CrossRef]
37. Zhang, R.; Stan, M.R.; Skadron, K. *HotSpot 6.0: Validation, Acceleration and Extension*; University of Virginia: Charlottesville, VA, USA, 2015.
38. Gao, D.; Reis, D.; Hu, X.S.; Zhuo, C. Eva-CiM: A System-Level Performance and Energy Evaluation Framework for Computing-in-Memory Architectures. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2020**, *39*, 5011–5024. [CrossRef]
39. Shen, L.; Wu, N.; Yan, G. Fuzzy-Based Thermal Management Scheme for 3D Chip Multicores with Stacked Caches. *Electronics* **2020**, *9*, 346. [CrossRef]
40. Brooks, D.; Tiwari, V.; Martonosi, M. Wattch: A framework for architectural-level power analysis and optimizations. *ACM Sigarch Comput. Archit. News* **2000**, *28*, 83–94. [CrossRef]
41. Lee, W.; Kim, Y.; Ryoo, J.H.; Sunwoo, D.; Gerstlauer, A.; John, L.K. PowerTrain: A learning-based calibration of McPAT power models. In Proceedings of the 2015 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), Rome, Italy, 22–24 July 2015; pp. 189–194.
42. Schöne, R.; Molka, D.; Werner, M. Wake-up latencies for processor idle states on current x86 processors. *Comput.-Sci.-Res. Dev.* **2015**, *30*, 219–227. [CrossRef]
43. Woo, S.C.; Ohara, M.; Torrie, E.; Singh, J.P.; Gupta, A. The SPLASH-2 programs: Characterization and methodological considerations. In Proceedings of the 22nd Annual International Symposium on Computer Architecture, Santa Margherita Ligure, Italy, 22–24 June 1995; pp. 24–36.
44. Bienia, C.; Kumar, S.; Singh, J.P.; Li, K. The PARSEC benchmark suite: Characterization and architectural implications. In Proceedings of the 2008 International Conference on Parallel Architectures and Compilation Techniques (PACT), Toronto, ON, Canada, 25–29 October 2008; pp. 72–81.
45. Yoon, C.; Shim, J.H.; Moon, B.; Kong, J. 3D die-stacked DRAM thermal management via task allocation and core pipeline control. *IEICE Electron. Express* **2018**, *15*, 1–12. [CrossRef]
46. Van Craeynest, K.; Akram, S.; Heirman, W.; Jaleel, A.; Eeckhout, L. Fairness-aware scheduling on single-ISA heterogeneous multi-cores. In Proceedings of the 22nd International Conference on Parallel Architectures and Compilation Techniques, Edinburgh, UK, 7–11 September 2013; pp. 177–187.